Первая часть этой книги охватывает основы, Postfix начиная с подготовки операционной системы. И рассказывает как настроить почтовый сервер для одного домена. Эти главы помогут Вам познакомится с синтаксисом конфигурационного файла и некоторыми из составляющих программ Постфикса и утилитами.

***краткий обзор четырех глав в этой части книги:***

## Подготовка сервера и Окружения

Перед установкой Постфикса, Вы должны всегда проверять, что ваш сервер может обращаться к SMTP серверу. Глава 2 показывает Вам, как подготовить операционную систему к установке Постфикса.

## Настройка почтового сервера для домена

Первый шаг в любой новой установке Постфикса создание конфигурации, которая может получать почту для одного домена. В Главе 3, Вы увидите, как проверить, что система работает и как создать основу для более сложных случаев.

### Dial-up почтовый сервер для домена

Вы должны незначительно изменить установку для одного домена , чтобы получить Dial-up почтового сервера для домена. Глава 4 показывает Вам эти маленькие, но важные изменения.

## Анатомия Постфикса

Витс Венема говорит, что "Постфикс - фактически маршрутизатор, " маршрутизирующий сообщения вместо IP пакетов. В Главе 5, показывается картина того, как взаимодействуют внутренние части Постфикса.

# Подготовка сервера и Окружения

*В начале ничего не было ничто. Бог сказал, " Пусть быдет свет! " Тогда все еще ничего не было , но Вы могли видеть это.*
*Ignacio Schwartz*

Вы вероятно сильно возбуждены, так как только что получили эту книгу, и не можете дождаться, начала работы с Постфиксом Однако, есть одна вещь, которую Вы должны знать прежде, чем Вы начнете.

Постфикс был создан Витсом Венемой, он хорошо знает Unix, и Постфикс не включает функциональные возможности, которые Unix обеспечивает по умолчанию. Поэтому, Постфикс ожидает, что ваша система будет настроена должным образом и его работа зависит от работы основной системы.
Не пропускайте эту главу, считая материал детским. Потратьте время, на прохождение следующих разделов, это гарантирует, что ваша система -, чтобы. Постфикс вознаградит Вас за эти усилия быстрой, надежной, и безопасной работой. Вот контрольный список для проверки системы перед установкой Постфикса:

- Правильное имя хоста
- Проверка возможности соединения с вашим хостом
- Системное время
- Удостоверьтесь, что служба syslog может записывать диагностическую информацию Постфикса
- Настройте службу разрешения имен для клиента
- Настройте записи DNS для почтового сервера

Почтовый сервер должен иметь доменное имя в формате FQDN (FQDN; см. RFC 821, ftp://ftp.rfc-editor.org/in-notes/rfc821.txt ) например mail.example.com для надежного взаимодействия с другими системами. Постфикс автоматически использует назначенное имя хоста при приветствии почтовых клиентов и серверов, если Вы вручную не определите другое имя.
Имя в формате FQDN важно, потому что Постфикс, не только принимает почту от клиентов - в режиме клиента, Постфик также передает сообщения другим почтовым серверам. Многие почтовые серверы проверяют имя хоста, который объявляет клиент и не принимают сообщения, если имя клиента не в формате FQDN, некоторые серверы даже проверяют, что FQDN соответствует записи DNS.

Ваша операционная система устанавливает имя хоста во время загрузки. Чтобы узнать, имеет ли ваша система уже, FQDN, загрузитесь войдите в систему и выполните:

```
$ hostname-f mail.example.com
```

Если эта команда не возвращает имя домена в формате FQDN, узнайте, как ваша система назначает имя хоста и переопределите его. Однако, если ваша система уже имеет FQDN имя, но Вы хотели бы, чтобы Постфик использовал другое имя, без изменения установок вашей системы,Вы можете назначить используемое имя с помощью параметра myhostname.

*ОБРАТИТЕ ВНИМАНИЕ, что опция -f  команды hostname не работает на Solaris, с командой GNU hostname, и в некоторых других средах. Если ваша команда hostname не работает как описано выше, пробуйте опустить опцию -f (справедливо для FreeBSD). Если это не помогает смотрите руководство,*

## Возможность соединения

Проверите, что ваша машина может соединятся с  сетью и что другие хосты в сети могут взаимодействовать с ней. Первая часть проста - если ваша машина может открывать веб страницы в онлайн браузере значит она  соединяется с сетью. Входящие соединения более сложные. Чтобы проверять их, Вы нуждаетесь в другой машине, подключенной к сети. Если Постфикс предоставляет услуги в сети Интернет, Вы должны проверить возможность соединения хоста, который полностью независит от вашего сервера.

### TCP Порт 25

Удостоверьтесь, что ничто не блокирует TCP соединения на 25 порту вашего сервера. Если у вас есть брандмауэр, удостоверьтесь, что политика брандмауэра разрешает входящие и исходящие соединения на 25 порту. Имейте в виду, что некоторые Интернет провайдеры ISP блокируют исходящие соединения на 25  порту в  Интернет на своих маршрутизаторах, если Вы не просите, чтобы они сняли это ограничение. Некоторые ISP могут отказаться снимать ограничения, предпочитая чтобы Вы пересылали почту через их почтовые серверы, используя систему типа SMTP аунтефикации, описанной в Главе 16.

## Системное Время  и временных записей

Наличие правильного системного времени  важно, когда вы производите тонкие настройки системы и устраняете проблемы. Когда Вы должны выйти за пределы вашей системы, для решения проблем почты с другими  администраторами почтовых серверов, правильные  временные записи в лог необходим  Вам для сопоставления событий на ваших почтовых серверах  с теми  серверами, которыми Вы не управляете.
Постфикс четко отображает  действия в заголовках почтовых заголовках. Например, взгляните на этот заголовок:

Received: from mail.example.net: (mail.example.net [192.0.34.166])

by mail.example.com (Postfix) with ESMTP id 6ED90E1C65

for <recipient@example.com>; **Sat,  7 Feb 2004 10:40:55 +0100 (CET)**

Reply-To:  sender@example.net

From:  Sender <sendergexample.net>

To:  Recipient <recipient@example.com>

Subject:  Keep correct system time

**Date: Sat,  7 Feb 2004 10:42:01 +0100**

Постфикс также делает  датированные записи  в лог файле (*maillog*) . Вот -

некоторые типовые записи:

**Feb 7 2004 10:40:55** mail postfix/pickup[326iO]: 6ED90E1C65: uid=50l from=<sender>

**Feb 7 2004 10:40:55** mail postfix/cleanup[398]: 6ED90E1C65:

message-id=<20O4O416O2O2O9.7D62343F3O@mail.example.com>

Поэтому, Вы должны быть уверены, что Вы получаете наиболее точное время, которое возможно. Не доверяйте встроенному таймеру вашей системы; не только время, сохраненное ядром Unix дрейфует с течением времени, но и чипы, которые изготовители материнских платах используют в часах питаемых батареей, дешевы и также дрейфуют. Вы не можете быть уверены, что источник местного времени находится в синхронизации с временем на других почтовых серверах.

Существует два способа получения точного времени. Вы можете использовать NTP (Сетевой временной протокол), чтобы синхронизировать время по сети, или использовать GPRS (во всем мире) или DCF-77 (в большинстве Европы) устройство, для синхронизации времени по радио. Однако, если Вы не имеете доступа к этим решениям, Вы можете пробовать использовать clockspeed (<http://cr.yp.to/clockspeed.html>) как последнею возможность. Эта утилита использует тактовые импульсы аппаратных средств ЭВМ для сравнения , и компенсации спешащие или отстающих часы. Учитывая измерения нескольких времен из надежных источников, вычисляет и компенсирует уход времени.

*ОБРАТИТЕ ВНИМАНИЕ, для использования NTP сервера, Вы должны настроить NTP клиент на вашей системе (такой клиент идет фактически с каждой операционной системой). Чтобы использовать NTP, Вы должны разрешить входящие и исходящие UDP пакеты на 123 порту в вашем брандмауэре. Если Вы не знаете, как сконфигурировать ваш NTP клиента, посетите вебсайт NTP (http://www.ntp.org ) для дополнительной информации.*

# Syslog

Одно из самых важных мест, для поиска диагностических сообщений – лог файл почты. Постфикс использует стандартный Unix демон, для ведения журналов, syslogd. Вы обычно конфегурируете syslogd через конфигурационный фаил */etc/syslog.conf*. Вот - типовая конфигурация:

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none -/var/log/messages
# The authpriv file has restricted access.
authpriv.* -/var/log/secure
# Log all the mail messages in one place.
mail.*          -/vax/log/maillog
# Log cron stuff
cron.*   -/var/log/cron
# Everybody gets emergency messages, plus log them on another
# machine. *.emerg   *
```

```
#  Save mail and news errors of level err and higher in a
#  special file.
uucpjnews.crit        -/var/log/spooler
#  Save boot messages also to boot.log local7.*
 /var/log/boot.log
```

Сначала, посмотрите на первую строчку, содержащую параметр **mail.none**, он указывает не помещать сообщения почтовой службы в */var/log/messages.* Это важно, потому что Вы не хотите, чтобы сообщения почтовой службы загромоздили ваши общие системные сообщения. Вы можете видеть, что почтовая служба имеет свою собственную запись и файл (*/var/log/maillog*). Дефис перед именем файла указывает, что **syslogd** должен записывать сообщения в файл асинхронно, а не пробовать осуществлять запись на диск каждый раз, при регистрации нового сообщения.

К сожалению, есть несколько вещей, которые могут идти не так, как надо с **syslogd.** Если Вы, не получаете регистрационные сообщения, самая первая вещь которую Вы должны сделать - удостоверяются, что syslogd запущен. Следующий пример показывает, как пользоваться командой **ps**, для поиска демона.

```
# ps auxwww | grep syslog
root     15540   0.0   0.0   1444   524  ?      S   May21   18:20 syslogd -m  0
root     22616   0.0   0.0   1444   452  pts/o  R   18:09    0:00  grep syslog
```

Первая строка вывода показывает, что **syslogd** выполняется с 21 мая.
Кроме того, удостоверьтесь, что файлы системного журнала существуют и перезаписываемы прежде, чем Вы указываете **syslogd** записывать в них. Некоторые реализации **syslogd** не создает файлы автоматически и не сообщают об ошибке, если есть проблема с файлами системного журнала. Solaris **syslogd** печально известен за это.
Очень часто ошибка состоит в использовании пробела вместо табуляции, чтобы разделять тип регистрации и файл системного журнала в */etc/syslog.conf* файле. Ваш *syslog.conf* должен иметь примерно такой вид:

```
mail.*<TAB>-/uar/log/maillog
```

Еще одна возможная проблема в **syslogd.conf** запись в лог файлы на другом хосте в сети. Не упустите строчку подобную этой:

```
mail.* @loghost
```

В этом случае, **syslogd** посылает все его регистрационные сообщения на **loghost**, так что Вы должны искать логи на этом хосте вместо почтового сервера . Удостоверьтесь, что Вы фактически имеете такой хост. Слишком часто регистрационная информация идет не к тому хосту (или в никуда) из-за неправильной fde строки в *syslogd.conf*.

## Разрешение Имен (DNS)

Прежде, чем почтовый сервер типа Постфикса, может передать сообщение к

удаленному адресату, он должен определить местонахождение адресата в Интернете, Вы находите удаленные ресурсы в Интернет  с помощью сервера доменных имен (DNS). Сервер имен возвращает IP адрес хоста, и наоборот имя хоста, которое соответствует IP адресу.
Хорошо функционирующий DNS является критическим для работы MTA . Чем скорее Постфикс, может определить IP адрес, тем скорее он сможет начать связываться с удаленным почтовым сервером и передавать сообщение.

*ОБРАТИТЕ ВНИМАНИЕ, что плохое выполнение поиска может стать главным узким местом на крупных почтовых серверах. Если ваш сервер сталкивается с проблемами, Вам поможет кэширующий ДНС сервер. Настройте кэширующий ДНС сервер для больших почтовых систем. Знайте, что антиспам меры могут увеличить число запросов DNS, вашего почтового сервера.*

Прежде, чем Вы попытаетесь улучшить работу ДНС вашей системы, убедитесь, что ваша операционная система правильно определяет удаленный почтовый сервер, запрашивая MX запись (см раздел MX записи долее в этой главе) postfix-book.com. у ДНС сервера.
Попробуйте выполнить следующую команду:

```
$ dig postfix-book.com MX
```

В результате получите примерно следующий вывод:

```
; <<>> DiG 9.2.2-P3 <<>> postfix-book.com MX
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23929
;; flags: qr rd ra; QUERY: l, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
;; QUESTION SECTION:
;postfix-book,com,                IN      MX
;; ANSWER SECTION:
   postfix-book.com.     86400  IN      MX     10mail.postfix-book.com.    1
;; AUTHORITY SECTION:
postfix-book.com.        86400  IN      NS     ns3.ray.net.                2
postfix-book.com.        86400  IN      NS     ns.state-of.
;; ADDITIONAL SECTION:
mail.postfix-book.com.   86400  IN       A     212.14.92.89
ns.state-of-mind.de.     81566  IN       A     212.14.92.88
;; Query time: 58 msec
;; SERVER: 212.18.0.5#53(212.18.0.5)
;; WHEN: Sat Apr 17 03:56:47 2004
;; MSG SIZE rcvd: 145
```

1 Эта строчка указывает, что mail.postfix-book.com – почтовый сервер, который принимает почту для адресатов в пределах домена postfix-book.com.
2 Эти две линии показывают, что ns3.ray.net и ns.state-of-mind.de  - авторитетные

серверы имен для postfix-book.com.

*ОБРАТИТЕ ВНИМАНИЕ команда dig не стандартна для некоторых устаревших платформ. Вы можете получить dig с сайта ISC (<http://www.isc.org>). Если Вы не можете установить dig, , Вы можете вероятно воспользоваться nslookop; данная команда теперь критикуется .*

Если запрос выполнен успешно, Постфикс, может (теоретически) определять имена хостов правильно. Если запрос не выполнен, и вы не можете выполнить запрос для любого другого хоста, Вы должны немедленно заняться DNS.

Одна обычная проблема при проблемах с работой ДНС недоступность серверов имен к которым обращается сервер . Проверьте ваш */etc/resolv.conf* файл. Допустим он выглядит примерно так: сервер сначала обращается к серверу имен на **localhost** (127.0.0.1), и в сучае неудачи опрашивает сервер 134.169.9.107:

```
nameserver 127.0.0.1
nameserver 134.169.9.107
```

Прекрасная идея опрашивать **localhost**, если у Вас настроен кэширующий сервер имен. Однако, если у Вас его нет, этот запрос отнимет некоторое время.

Как Вы узнаете позже, если запросы к серверу имен выполняются а Постфикс, не может найти хост (например, если Вы видите no route to host, в лог файле), то вероятно,Вы установили Постфикс в chroot окружении, и поэтому, он использует другие конфигурационные файлы, для обращения к серверам ДНС. Например, если ваша chroot директория - /var/spoll/postfix/, Постфикс, будет использовать файл /var/spool/postfix/etc/resolv.conf. Удостоверьтесь, что файл существует и верен, выполнив команду cp -p /etc/resolv.conf /var/spool/postfix/etc/resolv.con-f, и затем перезапустите Постфикс.

## DNS для Почтовых Серверов

Вы должны ваш сервер ДНС, чтобы остальная часть мира знала, что ваш сервер является почтовым сервером вашего домена. Попросите системного администратора, обслуживающего ДНС сервер вашего домена настроить следующие записи:

### A запись

Ваш почтовый сервер должен иметь полностью определенное имя так, чтобы клиенты могли узнать, адрес вашего сервера. А рекорд сопоставляет FQDN имя и IP адрес сервера.

### PTR запись

Имя хоста вашей системы должно быть обратно-разрешимым. Почтовые сервера, которые узнают, что имя вашего сервера из SMTP соединения должен быть способен узнать, является ли ваш сервер действительно тем, с кем он общается .

## MX запись

MX запись позволяют клиентам знать, что ваш сервер ответственен за доставку почты для домена или некоторого хоста.

Отчеты{Рекорды}
Система доменных имен имеет различные типы записей, для сообщения хостам о ресурсах в сети. Одина из наиболее важных – A запись, которая сопоставляет имя хоста с IP адресом. Клиент, который посылает имя хоста серверу имен, должен получить IP адрес хоста в качестве ответа. Следующий пример сессии показывает, что www.example.com сопоставлен с IP адресом 192.0.34.166.

```
$ dig www.example.com A
; <<>> DiG 9.2.1 <<>> www.example.com
;; global options:    printcmd
;; Got answer:
;; ->>HEADER«- opcode: QUERY, status: NOERROR, id: 30122
;; -flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL 0
;; QUESTION SECTION:
;www.example.com.                    IN              A
;; ANSWER SECTION:
www.example.com.          172627    IN              A          192.0.34.166
;; AUTHORITY SECTION:
example.com.              21427     IN              NS         b.iana-servers.net.
example.com,              21427      IN             NS         a.iana-servers.net.
;; Query time:  l msec
;;  SERVER:  I27.0.0.1#53(127.0.0.1)
;; WHEN: Sat Apr 17 16:43:40 2004
;; MSC SIZE rcvd: 97
```

## *RTR запись*

Партнер A записи - запись RTR, которая сопоставляет адреса к имени хоста. Когда клиент посылает IP адрес ДНС серверу, ответ должен быть именем хоста,переданного адреса, как в этом примере:

```
$ dig -x 192.0.34.166
; <<>> DiG 9.2.1 <<>> -x 192.0.34.166
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37949
;; "flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 0
;; QUESTION SECTION:
;166.34.0.192.in-addr.arpa.   IN    PTR
;; ANSWER SECTION:
```

166.34.0.192.in-addr.arpa. 21374 IN PTR www.example.coin.

;; AUTHORITY SECTION:

34.0.192.in-addr.arpa. 21374 IN NS ns.icann.org.

34.0.192.in-addr.arpa. 21374 IN NS svcOO.apnic.net.

34.0.192.in-addr.arpa. 21374 IN NS a.iana-servers.net.

34.0.192.in-addr.arpa. 21374 IN NS b.iana-servers.org.

34.0.192.in-addr.arpa. 21374 IN NS c.iana-servers.net.

;; Query time: 1 msec

;; SERVER: 127.0.O.l#53(l27.0.O.l)

;; WHEN: Sat Apr 17 16:44:39 2004

;; MSG SIZE rcvd: 201

*ВНИМАНИЕ Теперь, когда спамеры чума Интернета, обратное разрешение имен с помощью RTR записи наиболее важны чем когда-либо. Много администраторов почтовых серверов настраивают их серверы так, чтобы принимать почту, только если обратный поиск для соединяющегося клиента удачен.*
*Однако, только, потому что другие почтовые серверы отклоняют почту, основываясь на обратных запросах я не утверждаю, что Вы должны это делать. Это часто причиняет проблемы, потому что много провайдеров не делегируют RTR запись их клиентов и не обеспечивают надлежащую информацию относительно их сервера,*

## MX Записи

ДНС сервер делает больше чем предоставляет разрешения имен; он также говорит клиентам о сервисах, доступных в домене. Почтовый сервер, ответственный за домен - один из этих сервисов. Вы можете сформировать запись MX, чтобы указать на A запись вашего почтового.

*ПРЕДОСТЕРЕЖЕНИЕ, ЧТО DNS также имеет CNAME, псевдоним, который может указывать на A запись. Например, Вы могли сформировать запись CNAME, который преобразует www.example.com в srv01.example.com . Клиенты, которые запрашивают www.example.com , перенаправились бы srv01.example.com в ответ. Не формируйте запись MX к одному из этих псевдонимов. SMTP протокол требует, чтобы имя домена в адресе электронной почты было или в A записи или в MX записи. В предыдущем примере, Вы не могли указать в записи MX на www.example.com , но так srv01.example.com имеет A запись, Вы могли указать его там.*

Вы так же можете определить больше чем одну записи MX, и Вы можете также расположить по приоритетам почтовые серверы так, чтобы клиенты обращались к серверам в определенном порядке. Например:

$ dig m-net.de MX

; <<>> DiG 9.2.1 <<>> m-net.de MX

;; global options: printcmd

;; Got answer:

;; ->>HEADER«- opcode: QUERY, status: NOERROR, id: 3133

```
;; flags: qr rd ra; QUERY; 1,  ANSWER: 3, AUTHORITY; 2, ADDITIONAL; 0
;; QUESTION SECTION:
;m-net.de.                          IN      MX
;; ANSWER SECTION:
m-net.de.           7200    IN      MX   50 mail-in.m-online.net.   1
m-net.de.           7200    IN      MX   100 mxoi.m-online.net.     2
m-net.de.           7200    IN      MX   100 mx02.m-online.net.
;; AUTHORITY SECTION:
m-net.de.           7200    IN      NS   ns2.m-online.net.
m-net.de.           7200    IN      NS   nsl.m-online.net.
;; Query time: 27 msec
;; SERVER: 127.0.O.l#53(l27.O.O.l)
;; WHEN: Sat Apr 17 17:07:05 2004
;; MSG SIZE rcvd: 140
```

**1** mail-in.m-online.net имеет самый высокий приоритет, потому что он имеет самое низкое число (50). Клиенты будут пробовать доставить почту сначала этому серверу.

**2** mx01.m-online.net и mx02.m-online.net имейте второй по старшинству приоритет с номером (100) . Клиенты будут обращаться к любому из них, если почтовый сервер самого высокого приоритета недоступен.

# Почтовый сервер для домена

Конфигурирование Постфикса для отдельного домена, занимает несколько минут. Независимо от того, какую конфигурацию Вы планируете настраивать, ваш первый шаг всегда должен быть - старт со следующей конфигурации для отдельного домена; она докажет что Постфикс работает в самом простейшем случае.

Эта глава познакомит Вас с минимальным параметрами конфигурации, которые необходимы Постфиксу , для запуска, и это покажет Вам, как отобразить длинные адреса электронной почты в короткие пользовательские имена в случае простого домена.

**Минимальная Конфигурация**

Мы настроим, Постфикс, для приема электронной почты для отдельного домена. Постфикс должен доставлять электронную почту с различных почтовых адресов в пределах этого домена в различные почтовые ящики.
Постфикс должен обрабатывать почту только этого домена, и мы покажем минимальный набор изменений конфигурации которые потребуются внести, в первоначальную конфигурацию после установки . Типичная архитектура сети для минимальной конфигурации показана на рисунке 3-1.

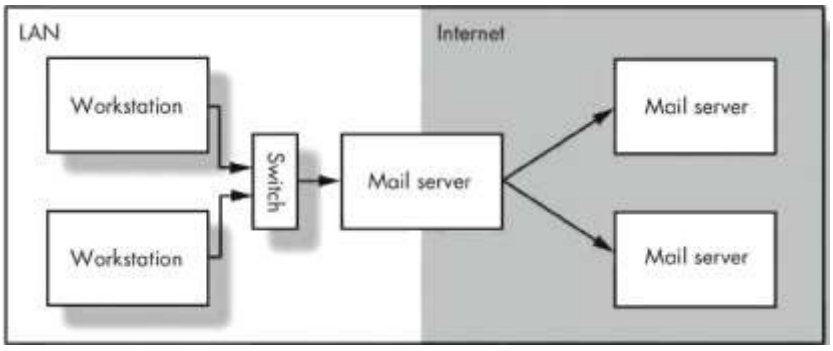



Рис 3.1 Архитектура сети одиночного домена

Почтовый сервер связан постоянно с Интернетом и имеет статический IP адрес. Прямая и обратная записи DNS, которые соответствуют, IP адресу почтового сервера настроены.

Базовая установка имеет основные требования. Удостоверьтесь, что Вы должным образом формировали ваш хост, как описано в Главе 2.

## Конфигурирование Постфикса

В этой главе, мы будем настраивать, Постфикс, чтобы получать почту для отдельного домена. Нашу машина будут иметь имя mail.example.com, а наш домен - example.com. Мы будем следовать за этими шагами:

1. Настроим, Постфикс, для приветствия почтовых клиентов с правильным

именем хоста.
2. Настроим, Постфикс, для принятия почты домена example.com.
3. Настроим, Постфикс чтобы он добавлял имя домена example.com к почте отправленной только с именем пользователя,
4. Настроим, Постфикс, чтобы он доставлял почту,адресованную root, в различные почтовые ящики.
5. Настроим Постфикс так, чтобы он доставлял почту, посланную на адрес электронной почты, соответствующим пользователям.
6. Установим разрешения для пересылки Постфиксом электронной почты из вашей сети.

## Установка имени хоста в smtpd баннере

Когда почтовые клиенты и серверы встречаются, они приветствуют друг друга с их DNS именами. Первую вещь, которую мы должны сделать,определить имя, которое Постфикс будет использовать, при приветствии почтового клиента. Если имя вашего хоста тот же самое, как то которое вы хотите чтобы Постфикс использовал при приветствии, значит вам повезло и вам не надо ничего менять.

С другой стороны, если имя хоста вашей системы установлено ка www.example.com , и Вы запускаете, Постфикс на той же самой машине и хотите, чтобы он приветствовал почтовых клиентв используя mail.example.com в качестве имени хоста, Вы можете легко достигнуть этого.

*ПРЕДОСТЕРЕЖЕНИЕ, когда Постфикс передает сообщения другим почтовым серверам, он работает как почтовый клиент. При приветствии почтового сервера почты, использует имя указанное в myhostname в качестве HELO имени. Некоторые почтовые серверы, настроены так чтобы отклонить почту если имя HELO и обратноразрешимое FQDN имя сервера не соответствуют друг. Или удостоверьтесь, что имя хоста, которое Вы устанавливаете для Постфикса соответствует имени хоста IP адресу вашего сервера, или установите параметр* ***smtp_helo_name*** *в соответствии с вашим официальным FQDN в DNS именном пространстве.*

Есть два способа получить различные имени хоста, или установив параметр ***myhostname*** или использовать параметр ***mydomain*** параметр.

## Установка *myhostname*

Установка параметра ***myhostname*** производится редактированием фаила */etc/postfix/main.cf.* Используйте ваш любимый редактор, для открытия файл и поиска параметра ***myhostname.*** Затем установите параметр hostname равным вашему FQDN имени:

***myhostname*** = mail.example.com

Как только Вы установили ***myhostname***, Постфикс, способен автоматически получить ***mydomain***. Постустфикс просто отбросит все ненужное до первой точки. Поскольку мы установили myhostname как mail.example.com, Постфикс получит ***mydomain***, равный example.com, как раз то что нас нужно.

## Использование ***mydomain***

Вместо того, чтобы использовать myhostname, Вы можете установить только ***mydomain***. Эта альтернатива может быть очень удобна, если Вы имеете конфигурацию, которая должна быть скопирована на несколько машин.

***mydomain*** = example.com

Как только Вы установили ***mydomain***, Постустфикс, способен получить myhostname, добавляя вывод команды uname-n данного хоста к ***mydomain***. Это означает, что, если ваш *main.cf* устанавливает mydomain явно, и Вы копируете файл на другой хост в пределах того же домена (example.com в нашем примере), Постфикс, опредеит правильный ***hostname*** самостоятельно.

## Установка Домена для которого принимается почта

Постфикс будет рэлеем локальных клиентов, это означает, что он примет почту для доменов, для которых он не сконфигурирован как конченое предназначение или рэлей. В установке для одного домена, все, что Вы должны сделать, установить параметр ***mydestination***. (Процедуры по настройке Постфикса в качестве почтового рэлея для более чем одного домена, обсуждаются в Главах 13 и 14,)

*ОБРАТИТЕ ВНИМАНИЕ, когда Вы устанавливаете.* ***mydestination****, Вы можете жестко задать предназначение (например,* ***mydestination =*** *mail.exomple.com, или Вы можете использовать значения параметров, которые были уже определенны, Постфикс использует запись вида $параметр . Жесткое определение параметров не удобно, при изменении конфигурации, потому что есть много параметров, чтобы редактировать,опечатки и другие потенциальные человеческий ошибки, могут испортить настройки. Мы не рекомендуем жесткое задание.*

Наша цель в этой главе состоит в том, чтобы, Постфикс, принимал любую почту, которая предназначена для example.com . Поскольку мы уже использовали это значение для параметра ***mydomain***, мы можем просто обратиться к нему, когда мы устанавливаем ***mydestination*** в *main.cf:*

***mydestination = $mydomain***

Если в дальнейшем, Вы хотите чтобы, Постфикс, принимал почту для хоста, который Вы определили в параметре ***myhostname***, то Вы просто добавляете этот параметр к ***mydestination***:

***mydestination = $mydomain, $myhostname***

Как Вы можете видеть, значения добавляются перечислением через запятую и пробел, в конце списка запятая отсутствует. Для дальнейшего использования , Вы можете также добавить wqw.example.com и ftp.example.com , дополняя список сочетанием типа хоста и переменной ***$mydomain***

```
mydestination =
        $mydomain,
        $myhostname,
        www.$mydomain,
        f t p.$mydomain
```

Этот пример также показывает другую форму записи. Если Вы добавляете много значений к параметру, Вы можете записывать каждое значение на отдельной строке, но каждая последующая строка должна начаться с табуляции (иначе Постфикс, не будет воспринимать значения). Вы можете проверить, это выполнив просмотрев вывод команды postconf mydestination, чтобы убедиться.
Этот формат может использоваться для любого параметра Постфикса который имеет несколько значений.

## Установка значения доменной части которая будет добавятся к исходящим сообщениям

Когда локальная служба, типа cron или at, или консольного почтового клиента отправляет почту, она обычно не использует полный адрес отправителя или получателя, а использует только имена пользователей. Хотя это хорошо, когда получатель является локальным, это становится проблемой, когда сообщение отсылается другому хосту. Это отнимает некоторое время, для отслеживания, хоста от которого поступает почта, и принимающий почтовый сервер, не будет способен возвратить сообщение назад, если получатель почты не существует на целевом хосте.
Постфикс имеет параметр, значение которого добавляется в конец адреса отправителя или получателя, которые определены в не FQDN форме: ***myorigin***. Мы снова можем повторно использовать параметры, которые уже были определены в пределах *main.cf*:

```
myorigin = $mydomain
```

Как только Вы произвели эту настройку, Постфикс, добавит значение ***mydomain*** к любому адресу, который не был задан полностью. Например, сообщение, сформированное демоном cron и посланное root было бы преобразовано в root@$mdomain, который в нашем случае примет значение root@example.com .

Если Вы не определите ***myorigin*** вручную, он примет значение myhostname, что очень удобно, если Вы управляете различными хостами, сообщения для root которых нужно доставлять на один ящик на центральном сервере. Этим путем Вы будете всегда знать, имя хоста от которого поступило сообщение; адрес отправителя сообщения посланного cron root`у, например, была бы изменено, Постфиксом, на root@$myhostname, в нашем случае на root@mail.example.com.

## Пересылка почты root`а на другой почтовый ящик

Постфикс доставит почту любому локальному пользователю непосредственно, даже root, но Постфикс, не даст привилегии root внешним программам в течение доставки. Это означает, что Вы не можете использовать местных агентов доставки (LDAs) типа procmail или maildrop, чтобы доставить почту для root, потому что Постфикс не запустит эти программы с привилегиями root, вместо этого он запустит

их с привилегиями пользователя nobody по умолчанию. Это - предостережение безопасности, предназначенная для того, чтобы никогда не ставить под угрозу учетную запись суперпользователя, запуская уязвимую внешнею программу от имени root`a. Это не означает, что невозможно доставить почту, которая предназначена для суперпользователя. Решение состоит в том, чтобы создать другого пользователя на вашей машине с нормальными, низкими привилегиями, и пересылать почту, предназначенную для root, этому пользователю.

В наших примерах мы используем admin[1] как учетную запись , с которой мы начинаем администрирование нашего хоста. Для того чтобы Постфикс, доставлял почту для root на почтовый ящик admin, просто откроем файл */etc/postfix/aliases*, который создается Постфиксом при установке[2], и заменим *postfix* на *admin* так, чтобы строка выглядела следующим образом:

root: admin

*ОБРАТИТЕ ВНИМАНИЕ Если Вы выберете учетную запись admin, для использования. для этой цели, Вы должны также удалить строчку файла псевдонимов, который отсылает почту admin`a root`y. Иначе Вы создадите петлю.*

Как только Вы отредактировали файл */etc/postfix/aliases*[3] и добавили имя пользователя, Вы должны создать индексированную версию файла, обычно */etc/postfix/aliases.db*, чтобы ускорять процесс поиска Постфиксом. Это делается, применением команды postalias к файлу /etc/postfix/aliases или запуском команды newaliases без параметров. Для использования стандартных утилит постфикса используйте следующую команду:

# postalias hash:/etc/postfix/aliases

*ПРЕДОСТЕРЕЖЕНИЕ Постфикс, не будет использовать никакие изменения в вашем файле псевдонимов, пока Вы не обновили индексированную версию, поскольку он производит чтение только этого файла.*

## Запуск Постфикса и проверка доставки почты, root`y

Пришло время провести первые испытаниями. В предыдущих разделах мы добавили или изменили множество значений, и если продолжим, без проверки, что все хорошо работает в данной точке, мы вероятно имели бы неприятности, если бы мы пропустили ошибку.

Прежде, чем мы начнем посылать почте, мы должны запустить, Постфикс. Все, в что Вы должны сделать ввести postfix start и Постфикс, должен ответить следующим сообщением:

# postfix start

1 так как недавно несколько вирусов -червей использовали учетную запись admin для распространения себя через интернет, использование этой учетной записи не очень хорошая идея.

2 файл псевдонимов который создается Постфиксом при установке содержит псевдонимы, которые требуются различными RFC стандартами. Там же Вы можете найти намек на то где искать более подробную информацию об этих требования

3 Входной и выходной файлы имеют формат, совместимы с Sendmail 8 версии и подходят для использования как N1S карт.

postfix/postfix-script:  starting the Postfix mail system

Если Вы получаете следующее сообщение, то Постфикс, был уже запущен:

```
#  postfix start
postfix/postfix-script: fatal: the Postfix mail system is already running
```

Если Постфикс был запущен, когда Вы сделали изменения в его конфигурации, то изменения не будут применены Постфиксом, Вы могли остановить, и  стартовать Постфикс, чтобы заставить его перечитать конфигурацию, но есть намного более изящный способ сделать это. Просто выполните postfix reload:

```
# postfix reload
postfix/postfix-script: refreshing the Postfix mail system
```

В этом случае, Постфикс, только пречитает конфигурацию, которая отнимает меньше времени и не будет прерывать обслуживание клиентов.

## Отправка тестового письма

Теперь, когда Постфикс запущен, мы можем провести первое испытание: доставка письма, посланного root в его почтовый ящик. Есть два очень простых способа сделать это: послать письмо из командной строки, или послать письмо из telnet сессии. Оба подхода имеют преимущество исключения влияния других приложений, типа почтовых клиентов GUI , и позволяют Вам сосредоточиться на Постфиксе, в случае,  появления ошибки.

## Использование утилиты sendmail

Самое простое, надежное испытание использование sendmail, для проверки основных функциональных возможностей, потому что никакие компоненты вне Постфикса не будут использоваться . Эту полезную утилиту командной строки называют sendmail для обратной совместимости – многие приложения  на Unix системах, посылают электронную почту, имеют привязку к утилите sendmail  в, */usr/sbin/sendmail* или /*usr/lib/sendmai*l,  жестко закодированную в них. Постфикс также помещает туда собственную утилиту sendmail, чтобы обеспечить гладкий переход от Sendmail к Постфиксу.[4]
Введите следующую команду, чтобы послать письмо пользователю root:

```
# echo foo | /usr/sbin/sendmail -f root root && tail -f /var/log/maillog
```

Она пошлет текст foo root'y, от  root в качестве отправителя , и откроет ваш почтовый лог файл, чтобы проверить статус доставки:

```
Aug 20 21:56:42 mail postfix/pickup[5l6o]: 848AD7247:  uid=0 from=<root>
Aug 20 21:56:42 mail postfix/cleanup[5340]: 848AD7247:
message-id=<2O03082O195642.848AD7247@mail. example. com>
```

4Есть один нюанс: Если Вы мигрируете от, Sendmail до Постфикса, Вы, можете получить две утилиты sendmail: те, которую поставил  Постфикс и ту, что осталась от реального Sendmail, Вы должны только использовать ту что устанавил Постфикс.

Aug 20 21:56:42 mail postfix/nqmgr[516l]: 848AD7247:

from=<root@mail.example.com>, size=306,, nrcpt=l (queue active)

Aug 20 21:56:42 mail postfix/local[5343]: 848AD7247:

to=<admin@mail.example.com>, orig_to=<root>, relay=local, delay=0,

status=sent (mailbox)

Как Вы можете видеть из лог файла, Постфикс был способен послать сообщение в почтовый ящик. Вы можете проверить это, выполнив **less /var/mail/admin**

from root@mail.example.com Wed Aug 20 21:56:42 2003

Return-Path: <root@mail.example.com>

X-Original-To: root

Delivered-To: admin@mail.example.com

Received: by mail.example.com (Postfix, from userid 0)

id 848AD7247; Wed, 20 Aug 2003 21:56:42 +0200 (CEST)

Message-Id: <20030820195642.848AD7247@mail.example.com>

Date: Wed, 20 Aug 2003 21:56:42 +0200 (CEST)

From: root@mail.example.com (root)

To; undisclosed-recipients:;

foo

*Если Вы неуверены, где искать почтовый ящик, введите* **postconf mail_spool_directory**. *Эта команда покажет вам, куда Постфикс доставляет почту.*

## Отправка письма из Командной строки

Затем мы проверим, что мы способны послать почту root с помошью. MUA на локальном хосте. Это второй самый простой тестовый случай:

# mail admin

Subject: Test from command line

This is a test mail from command line.

***Совет*** *В случае, если Вы не знакомы с программой mail, вот как ею пользоваться*

1. *Введите* ***mail*** *в командной строке.*
2. *Введите название аккаунта, которому Вы хотите послать почту, и нажмите Return.*
3. *После запроса введите тему сообщения и нажмите Return*
4. *Введите текст сообщения.*
5. *Чтобы посылать сообщение, начните новую чистую строку наберите одну точку, и нажмите Return.*

Для проверки доставки почты введите **less /var/mail/admin** еще раз:

```
# less /var/mail/admin
From root@mail.example.com   Wed Aug 20 20:55:11 2003
Return-Path:  <root@mail.example.com>
X-0riginal-To:  admin
Delivered-To:  admin@mail.example.com
Received:  by mail.example.com (Postfix,,  from userid 0)
        id  37DE07247;  Wed,   20 Aug 2003  20:55:11 +0200 (CEST)
To: admin@mail.example.com
Subject: Test from command line
Message-Id:  <2003O820i8ssii.37DEO7247@mail.example.com>
Date: Wed,  20 Aug 2003 20:55:11 +0200 (CEST)
 From:  root@mail.example.com (root)

This is a test mail from command line.
```

Сообщение доставлено, и мы доказали, что локальные пользователи могут послать почту другим локальны пользователям. Теперь пришло время проверять, можно ли посылать почту admin`у от удаленного пользователя.

## Отправка почты через Telnet Сессию

Самый простой почтовый клиент - telnet клиент, который соединяется с SMTP портом (порт 25). Мы проделаем этот трудный путь, потому что мы хотим исключить побочные эффекты, которые могли бы быть введены другим более удобным (и содержащим больше ошибок) почтовым клиентом. Вот, как Вы посылаете почтовое сообщение с помощью telnet:

```
# telnet mail.example.com 25
Trying 172.16.0.1...
 Connected to mail.example.com. Escape character is ,'^]'.
220 mail.example.com ESMTP Postfix
HELO client.example.com
250 mail.example.com
MAIL FROM: <test@client.example.com>
250 Ok
RCPT TO: <root@example.com>
250 Ok
DATA
354 End data with <CRxLF>.<CR><LF>
```

**Test mail from a telnet session**

.

250 Ok: queued as 69F1A7247

**QUIT**

221 Bye

И, в завершение, проверим доставку почты с помощью **less /var/mail/admin**:

From test@client.example.com Wed Aug 20 21:25:16 2003

Return-Path: <test@client.example.com>

X-Original-To: root@example.com

Delivered-To: admin@mail.example.com

Received: from client.example.com (mail.example.com [172.16.0.1])

by mail.example.com (Postfix) with SMTP id 2D89A72S1

for <root@example.com>; Wed, 20 Aug 2003 21:24:59 +0200 (CEST)

Message-Id: <20030820192459.2D89A7251@mail.example.com>

Date: Wed, 20 Aug 2003 21:24:59 +0200 (CEST)

From: test@client.example.com

To: undisclosed-recipients:;

Test mail from a telnet session.

## Сопоставление адресов электронной почты к именам пользователя

Теперь, когда мы успешно настроили базовую конфигурацию, пришло время сформировать адреса электронной почты, что обычно немного более сложно. По умолчанию, Постфикс доставляет электронную почту только локальным пользователям на вашем почтовом сервере. Однако, имена пользователя (типа y0OOO247), которые также часто используются для аунтефикации, когда пользователи хотят получить почту, редко соответствуют именам которые используют люди для связи между собой (типа john.doe@example.com). Заставим Постфикс получают и доставлять электронную почту для имен, используемых в реальном мире к существующим акаунтам, Вы нуждаетесь в создании псевдонимов, которые указывают на адресатов, которым Постфикс должен доставлять сообщения.

## Создание Псевдонимов

Предположим, что Вы имеете нового коллегу в Example Inc, имя которого - Джон До, и - ваша задача, обеспечить его адресом электронной почты. Джон работает в коммерческом отделе, и он, предполагает, получать почту, которую адресуют на john@example.com, john.doe@example.com, и doe@example.com в один почтовый ящик, так же как как почту отправленную на sales@example.com, Он работает вместе с Сильвией и Карол, которые обе получают любую почту, которая поступает на sales@example.com. Джон уже имеет акаунт Jhon, с которым он имеет доступ к своим файлам.

Вы должны сопоставит имена псевдонимов (john@example.com, sales@example.com,

и так далее) с его локальным именем пользователя. Это делается, созданием записей в */etc/postfix/aliases*. В случае Джона, Вы должны создать только три записи, хотя требуются четыре сопоставления. Вы не должны создавать запись - john@example.com так, как любая почта, которую посылают по этому адресу, будет доставляться на локальную учетную запись Джона. Вы должны были добавить следующие строки в */etc/postfix/aliases*:

```
# users
John.doe:        John
doe:             John
# groups
sales:           silvia, karol, John
```

Чтобы закончить вашу задачу, Вы будете должны выполнить, или postalias hash:/etc/postfix/aliases или newaliases, чтобы обновит ваш фаил aliases.db.

*ОБРАТИТЕ ВНИМАНИЕ Из предыдущего внесения в список, Вы можете видеть, что Вы должны определить локальную часть с левой стороны, заканчивая двоеточием и именем пользователя на правой стороне (локальная часть адреса электронной почты – все до знака @ ). Каждая строка псевдонима может состоять из одной или более значений разделенных запятыми и пробелом. Вы можете определить или имена пользователей или адреса электронной почты. Адреса электронной почты могут указывать на других пользователей на различных хостах, что означает, что Вы могли принять почту для пользователя на вашем сервере почты и переслать его на совершенно другой адресу. Дальнейшая информация может быть найдена непосредственно в файле aliases, или Вы можете обратится к справочной системе* ***man 5 aliases****.*

Как только Вы добавили столько псевдонимов, сколько Вам нужно, пришло время протестировать почтовые ящики, точно так же как мы делали раньше.

## Установка разрешения на пересылку постфиксом электронной почты из вашей сети

*Открытые релеи* кошмар администратора почтового сервера. Любая установка Постфикса - по умолчанию защищенный релей. В его первоначальной конфигурации, Постфикс будет пересылать только сообщения от IP адресов в вашей сети. Постфикс определяет IP адреса вашей сети, проверяя все сетевые интерфейсы, которые Вы сконфигурировали для вашего сервера.

*ОБРАТИТЕ ВНИМАНИЕ На Linux сервере, Постфикс будет доверять всем подсетям, в которых находятся сетевые интерфейсы. Выполните* ifconfig *на Linux, чтобы получить список всех подсетей которым Постфикс будет доверять по умолчанию.*

Установки по умолчанию работают пока ваш сервер и хосты, использующие Постфикс находится в пределах одного и того же самого диапазона сети. Возможно, что Вы будете должны изменить эти настройки, когда ваша сеть вырастет или станет более сложной. Вы могли, например, решить запустить Постфикс в

демилитаризованной зоне в пределах IP диапазона, который отличается от того что используют ваши внутренние хосты. В той ситуации, Постфикс вероятно не разрешил бы вашим клиентам предавать почту к инородному предназначению, и Вы должны будите исправит это, установив правильные разрешения для пересылки.

Расширение или ограничение разрешений пресылки могут быть сделаны или в общем виде, выбирая значение параметров ***mynetworks_style***, который удовлетворяет топологии вашей сети, или индивидуально, вручную определяя список IP адресов или рангов в (CIDR) выражении (смотри Приложение C) для параметров ***mynetworks***.

Оба метода требуют, чтобы Вы изменили конфигурацию в main.cf вручную. Усилие администраторов разумно для статических IP диапазонов, потому что они не изменяются часто.

*ОБРАТИТЕ ВНИМАНИЕ что ручное редактирование не имеет смысла, если Вы хотите разрешить передавать почту для хостов с динамическими IP адресами, которые изменяют их IP адрес регулярно. Внесение изменений вручную быстро становится утомительной задачей. Глава 16 объясняет , как автоматизировать этот процесс.*

## Общие сетевые разрешения пересылки

Общие разрешения установлливаются параметром ***mynetworksstyle***, выбором опций *class*, *subnet*, или *host*

### *class*

Выбор значения *class* заставит Постфикс расширить разрешения для к целому IP классу A/B/C классу сети, для которых сервер сконфигурирован. Например, если Вы запустили Постфикс на машине с IP адресом 192.0.34.166, и Вы задали ***mynetwork_sstyle =class***, Постфикс будет доверять целому классу сети, 192.0.34.0/24, и разрешил бы передавать почту для хостов в пределах этого диапазона.

### *subnet*

Выбор опция *subnet* заставит Постфикс ограничить разрешения только теми подсетями, для которых Вы сконфигурировали сетивые интерфейсы сервера. Например, если Вы запустили Постфикс на машине с IP адресом 192.0.34.166/30, и Вы задали mynetworks_style =subnet , Постфикс будет доверять всем хостам точно в пределах этого диапазона.

### host

Выбор опции host заставит Постфикс ограничить разрешения передачи сервером, на котором Вы запустили Постфикс. Например, если Вы запустили, Постфикс на машине с IP адресом 192. 0.34. I66/30 и 127.0.0.1, и Вы устаоновили mynetworks_style = host. Постфикс доверял бы, только хосту. (IP адреса 127.0.0.1 и 192.0.34.166).

## Индивидуальные разрешения пересылки

Индивидуальные разрешения пересылки устанавливаются параметром ***mynetworks***, перечислением через запятую всех хостов и сетей, в CIDR записи, для которых Постфикс служит релеем.

Например, если Вы запустили Постфикс в сети, которая соединила два местоположения (192.168.100.0/24 и 192.168.200.0/24), и Вы хотели, чтобы было разрешено передавать почту для всех хостов демилитаризованной зоны, которая определена (10.0.0.0/30), и также для любого из его собственных локальных интерфейсов (127.0.0.0/8), Вы определили бы список так:

mynetworks = 127.0.0.0/8, 192.168.100.0/24, 192.168.200.0/24, 10.0.0.0/30

ОБРАТИТЕ ВНИМАНИЕ*, Если у вас много IP адресов и диапазонов, этот вид внесения в список может стать весьма сложным в пределах main.cf. Альтернативно Вы можете указать mynetworks в отдельном файле (mynetworks -* ***hash:/etc/postfix/mynetworks****) и поместить полный список там. Этот файл не может содержать сети в CIDR записи. Если Вы нуждаетесь в CIDR примечании, используете* ***mynetworks = cidr:/etc/postfix/ntynetworks,***

# DIAL-UP ПОЧТОВЫЙ СЕРВЕР ДЛЯ ОДИНОЧНОГО ДОМЕНА

Настройка почтового сервера, использующего модемное соединение требует незначительных изменений базовой конфигурации Постфикса. Модемное соединение стоит денег (особенно в Европе, где есть плата за соединение ), так что Вы можете не хотеть запустить почтовый сервер, который устанавливает связь для каждого исходящего сообщения. Вместо этого, Вы можете сделать так, чтобы сервер накапливал некоторое количество сообщений перед посылкой их, делая процесс передачи более рентабельным.

Когда модемное соединение будет активным, Вы захотите чтобы, Постфикс пересылал сообщения очередями через хост вашего ISP провайдера. Кроме того, Вы, возможно, должны поддержать SMTP аунтификации. Вы должны также автоматически найти сообщения, которые нельзя было доставить локальным пользователям, в то время как сервер был без связи.

Различия между dial-up сервером и базовой конфигурацией Постфикса следующие:

**Связь**

Поскольку почтовый сервер только временно связан с Интернетом, его IP адрес вероятно изменяется с каждой новой связью.

**DNS разрешения**

Сервер не может выполнять ДНС запросы, когда он автономен. Также, собственные DNS данные сервера изменяются с каждой новой связью, так что правильное обратное разрешение не может быть доступным.

**Ограничения доставки**

Ваш ISP требует, чтобы Вы использовали его релей, и кроме релей может пересылать сообщения только для авторизованных пользователей.

**Поиск почты**

Внешние почтовые сервера, не могут доставлять сообщения непосредственно вашему серверу, потому что ваш сервер обычно вне сети. Ваш ISP должен обращаться к почтовым серверам, у которых есть почта для вас. Когда Вам посылают сообщения, почтовый сервер вашего ISP принимает и хранит ее, пока Вы не используете или POP/IMAP клиента или fetchmail, чтобы отыскать почту и передать ее вашему местному MTA.

*ОБРАТИТЕ ВНИМАНИЕ, что доставка Почты с POP/IMAP и fetchmail (http://catb.org / ~ esr/fetchmail) не описывается в этой книге.*

Рисунок 4-1 изображает типичную модемную сеть. Одна или более машин находятся в частной сети, и любая машина, которая нуждается в услугах Интернет доступа, использует ваш модемный гейт,на котором также запущен ваш Постфикс сервер.

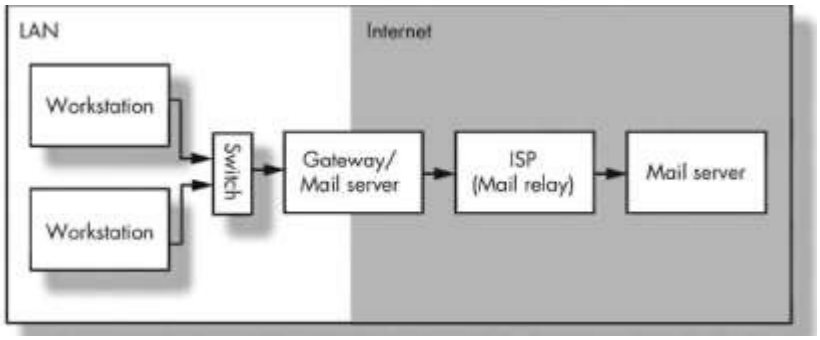



Рис 4.1 модемная сеть.

Вы должны будете выполнить, следующую последовательность шагов, чтобы сконфигурировать, Постфикс в качестве модемного почтового сервера для одиночного домена. Эта последовательность шагов, описана в следующих разделах.

1. Отключить DNS.
2. Проверить разрешения пересылки.
3. Установить хост для пересылки почты.
4. Отсрочить передачу сообщения.
5. Определить механизм запуска доставки сообщений.
6. Сконфигурировать разрешение для релея.

*ОБРАТИТЕ ВНИМАНИЕ, что Этот сценарий основывается на установке описанной в Главе 3. Вы должны сконфигурировать и проверять ваш сервер как описано в этой главе. Кроме того, Вы должны иметь доступно dial up соединение, настроив ваш сервер* *http://www.ibiblio.org/pub/Linux/docs/HOWTO/ other-formats/html_single/PPP-HOWTO.html.*

## Отключение DNS запросов.

Когда Постфикс, получает сообщение, которое будет доставлено удаленному домену, он должно искать MX записи домена предназначения. Поиски имени на DNS серверах обычно формируют запрос, покидающий вашу сеть, что означает, что сервер должен соединиться с Интернетом.

Поскольку Вы хотите свести число модемных соединений к минимуму, Вы должны инструктировать, Постфикс, не выполнять DNS запрос, пока сервер не имеет соединения. Фактически, Постфикс, никогда не должен искать удаленные домены, потому что Вы хотите, чтобы он послало сообщения через релей вашего ISP, который непосредственно может выяснить, куда посылать сообщение.

Для предотвращения выполнения Постфиксом DNS запросов, установите параметр ***disabie_dns_lookups*** в вашем файле *main.cf*:

```
disable_dns_lookups = yes
```

Это подавляет DNS MX/A запросы smtp(8) клиента, и поиски lmtp (8) клиента; в обоих случаях вместо этого используется gethostbyname(), Вы должны будите помнить это, когда Вы будете определять релей, далее в этой главе.

После установки параметра ***disabie_dns_lookups***, перезапустите, Постфикс, чтобы активизировать изменение.

*ОБРАТИТЕ ВНИМАНИЕ, что эта установка не отключит DNS для* smtpd *сервера. Параметры типа reject_unknown_sender_domain и permit_mx_backup (см. Главу 8} все еще, работают, независимо от значения параметра* ***disabie_dns_lookups***

## Настройка релей разрешений

Dial Up сервер обычно имеет динамический IP адрес, который изменяется всякий раз, когда сервер соединяется с Интернетом. Поэтому, Вы не можете управлять релей разрешениями для сетевого интерфейса Dial UP сервера, если Вы вручную не устанавливаете разрешения каждый раз, когда ваш сервер соединяется с Интернетом. Также, кто в Интернете хотел бы пересылать посту через другой Dial UP сервер кроме спамеров?

*ОБРАТИТЕ ВНИМАНИЕ, даже если ваш хост имеет только периодическую возможность соединения, Вы никогда не должны позволять релей доступ для всего Интернета. Один из Dial UP почтовых серверов, автора получил 56 (неудавшихся) попыток релея в течении 30-дневного периода. Это - примерно два день, а машина даже была в сети постоянно! К счастью ничто не случилось, потому что это был защищенный релей.*

Если Вы не хотите позволить пользователям из Интернета, использовать ваш Постфикс сервер как релей по некоторой причудливой причине (см. Главу 16), Вы должны ограничить передачу вашим локальным сетевым интерфейсом и интерфейсом loopback в файле main.cf. Например, Вы могли бы сделать это так, если ваша частная сеть 192.168.0.0/24:

```
mynetworks = 192.168.0.0/24, 127.0.0.1/8
```

*ПРЕДОСТЕРЕЖЕНИЕ не использует* ***mynetworks_style = class****, чтобы управлять разрешениями реле для Dial UP сервера. Это значение использует все IP диапазоны адресов, формируемые для ваших сетевых интерфейсов, включая сеть, с которой ваш сервер соединяется по модему . Поэтому, каждый клиент в сети вашего ISP провайдера был бы способен использовать ваш почтовый сервер для пересылки сообщений!*

Как и прежде, используйте postfix reload, чтобы перезагрузить конфигурацию.

## Установка релей хоста провайдера.

Прежде, чем Вы сделайте этот специфический шаг конфигурации, Вы должны определить почтовый реле хост вашего ISP. Много ISP провайдеров блокируют исходящие связи на 25 TCP порту (SMTP порт) для для модемных клиентов, потому что спаммеры используют модемные соединения для рассылки.

*ОБРАТИТЕ ВНИМАНИЕ В дополнение к собственным требованиям вашего ISP, есть множество серьезных оснований, чтобы не Постфикс, не отпровлял сообщения непосредственно конечному предназначению. Например, из за того что существенное количество спама происходит от модемных соединений, в*

*черные списки начали вносить целые блоки Dial UP сетей (аналогично, ISDN, и ADSL) которые использовались спамерами. Даже если ваше сообщение - не спам, удаленный MTA мог бы отклонить его на основе DUL (список модемных пользователей), просто потому что ваша почта происходит из диапазона IP адресов, принадлежащего модемному пулу.*

Например, если релей хост relay.example.com, Вы использовали бы запись:

relayhost = [relay.example.com]

Размещение имени или адреса релей хоста в квадратных скобках отменяет запрос MX записи для этого хоста.
После перезапуска Постфикса, Вы готовы идти дальше.

## Отсрочка передачи Сообщений

В данной точке, Постфикс, имеет конфигурацию, отправляющую почту релей хост без DNS запросов, избегая любых проблем открытого релея. Однако, сервер все еще набирает номер ISP провайдера каждый раз, когда получает исходящую почту, предназначенную для удаленных сетей. Чтобы остановить такое поведение и указать Постфиксу ставить в очередь, исходящие сообщения, отредактируйте main.cf и укажите, Постфиксу, отсрочить доставку методом SMTP с помощью параметра ***defer_transports*** , как показано в следующем примере:

defer_transports = smtp

*ОБРАТИТЕ ВНИМАНИЕ, если вы используете UUCP вместо SMTP, Вы можете заменить smtp на uucp.*

Как обычно, выполните, перезагрузку Постфикса после того, как внесли эти изменения. После перезагрузки, Постфикс, больше не будет доставлять сообщения через SMTP, пока ***defer_transports*** параметр не изменяется или не исчезает. Следующий раздел показывает, как использовать эту особенность, чтобы доставлять сообщения, когда ваш сервер соединяется с ISP.

## Вызов доставки сообщений

Единственная оставшаяся задача состоит в том, чтобы инструктировать, Постфикс, доставлять всю почте из очереди через SMTP, когда сервер соединяется с Интернетом. Все, что Вы должны сделать - автоматически переконфигурировать, Постфикс, когда сервер соединен с Интернетом и вернуть первоначальную конфигурацию позже. Вы можете сделать это с помощью сценария, который система выполняет при установлением связи. На Linux системе, с запущеноой PPP, эти сценарии часто находятся в /etc/ppp/ip-up.d.

Создайте сценарий, с именем postfix в этой директории, чтобы он выполнялся *за сценарием*, который устанавливает resolv.conf. Сценарий postfix примерно такой:

```
## start or reload Postfix as needed
#  if Postfix is running chrooted, copy resolv.conf to the resolv.conf Postfix
uses
cp -p /etc/resolv.conf  'postconf -h queue_directory `/etc/resolv.conf
```

```
#  unset defer_transports and make Postfix note it
postconf -e "defer_transports ="
postfix reload
#  Force a queue run to unload any mail that is hanging around
postfix flush
```

1 Строка, содержащая resolv.conf уместна, только если Постустфикс, запущен в chroot окружении. Она предполагает, что сервер изменяет resolv.conf файл при установке связи. Постфиксу также необхдимо знать текущие nameservers, так что эта команда копирует новую версию в chroot окружение, где Постфикс, может найти его.

Точно так же, когда машина автономна, Вы хотите восстановить старое поведение организации очереди. Создайте сценарий названный, postfix в /etc/ppp/ip-down.d, который будет выполнятся, при разрыве связи (снова, ситрока с resolv.conf необходима только в chroot jail)

```
# start or reload Postfix as needed
#  copy resolv.conf to the resolv.conf Postfix uses (only if Postfix is chrooted)
cp -p /etc/resolv.conf 'postconf -h queue_directory' /etc/resolv.conf
#  set defer_transports and make Postfix note it
postconf -e "defer_transports = smtp"
postfix reload
```

## Конфигурирование разрешений пересылки для релей хоста

Много бесплатных поставщиков почты, особенно те, кто предоставляют SMTP доступ клиенту наряду с почтовым веб интерфейсом, требуют дополнительной авторизации прежде, чем они разрешают вашему клиенту использовать их релей хост. Это необходимо, потому что большинство их пользователей соединяется от других организаторов доступа (и, поэтому, их IP диапазон отличается от их собственных), так что они не могут установить разрешения пересылки, основанные на IP адресах. Если бы поставщики почты открыли свои почтовые серверы широкому диапазону IP адресов, они фактически стали бы открытыми релеями, и потребуется всего несколько минут прежде, чем спамеры начали использовать их. Поэтому, поставщики почты требуют - POP-перед-SMTP или SMTP установление подлинности.

## POP-перед-SMTP авторизация.

Поставщик, который требует POP-before-SMTP (смотри Главу 15) принимает исходящие сообщения, только если Вы получаете входящую почту перед посылкой любых новых сообщений. Другими словами, ваша машина должна подтвердить подлинность с POP3 сервером или IMAP4 сервером перед посылкой чего - нибудь. Когда ваш хост подтверждает подлинность, поставщик определяет и запоминает, ваш текущий IP , и позволяет, данному IP, послать сообщения через его релей в пределах некоторого временного окна.

Постфикс- MTA ; он не общается с POP3 и IMAP 4. Поэтому, Постфикс, не

может исполнить POP-before-SMTP самостоятельно. Это - не проблема, потому что Вы можете легко настроить fetchmail (<http:// catb.org / ~ esr/fetchmail>), чтобы сделать это для Вас. Fetchmail - маленькая утилита , которая получает почту от практически любого вида почтовой системы в Интернете. Чтобы использовать ее в POP-before-SMTP авторизации, выполните эти шаги:

1. Сконфигурируйте, постфикс как описано в этой главе.
2. Следуйте инструкциям в документации fetchmail, чтобы создать рабочую конфигурацию.
3. Добавите триггер, который вызывает fetchmail перед переконфигурированием Постфикса в вашем /etc/ppp/ Dial UP сценарии.

Таким образом, ваш сервер запускает fetchmail (POP3/IMAP4 клиент) по крайней мере однажды перед запуском Постфикса (SMTP) стадия, так что ваш поставщик почты примет ваши исходящие сообщения.

## SMTP установление подлинности

Поставщик, который требует SMTP установление подлинности, позволяет вашему клиенту или серверу пересылать сообщения через их релей хост, только если он аунтифицировал себя в течение SMTP сеанса. Для использования SMTP установление подлинности в Постфиксе, Вы не нуждаетесь ни в каких дополнительных утилитах или программах, так что это предпочтительней чем POP-before-SMTP (особенно в случаях, где Вы хотите послать сообщения, но не получать ничего).

Вы можете найти, обширную информация относительно того, как формировать клиентскую часть SMTP установление подлинности для Постфикса в Главе 16.

# ГЛАВА 5

# Анатомия Постфикса

Эта глава описывает, как работает Постфикс, что делает каждая часть системы, и как эти компоненты взаимодействуют друг с другом. После прохождения этого материала, Вы будите понимать, Постфикс в целом, и сможете сосредоточится на индивидуальных задачах.

Постфикс состоит из небольшого количества программ, которые взаимодействуют с пользовательскими процессами (*sendmail, postqueue, postsuper*, и так далее) и большая этих программ работают в фоновом режиме. программы которые работают в фоновом режиме управляются демоном master , . Работа демона master состоит в определении, какое действие нужно выполнить и в вызове соответствующей программы, для выполнения работы. Такая модульная конструкция обеспечивает более высокий уровень безопасности, потому что каждая программа запускается с самым низким набором привилегии необходимым для выполнять ее задач.
Вы можете думать в целом, о Постфикс системе как о маршрутизаторе. Это может звучать странно сначала, но вспомните, что работа маршрутизатора состоит получении IP пакета, определении IP адреса предназначения (и возможно источника), и затем в выборе правильного интерфейса для отправки пакета к месту предназначения. Постфикс делает ту же самую вещь с почтой (см. Рисунок 5-1), смотрит на предназначение сообщения (получатель письма); и на источник (отправитель письма), для определения приложения, которое подвинет сообщение поближе к его заключительному предназначению.

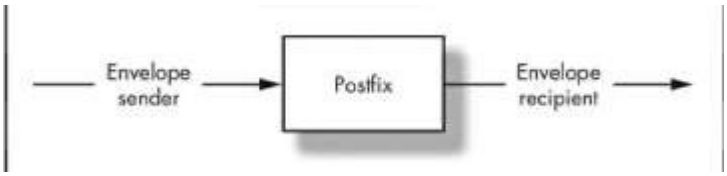


Рис 5-1 работа Постфикса как маршрутизатора

Теперь давайте более близко рассмотрим систему. Реальный маршрутизатор обычно принимает IP пакеты из множества интерфейсов, и направление их обратно через интерфейсы. То же самое верно для, Постфикса; он принимает сообщения от множества источников и затем передает почту к множеству предназначений. Сообщение может происходить от локальной утилиты sendmail или из SMTP или QMQP соединения. Предназначением может быть локальный почтовый ящик, исходящие SMTP или LMTP соединение, передача в программу, и дальше. Рисунок 5-2 иллюстрирует это.

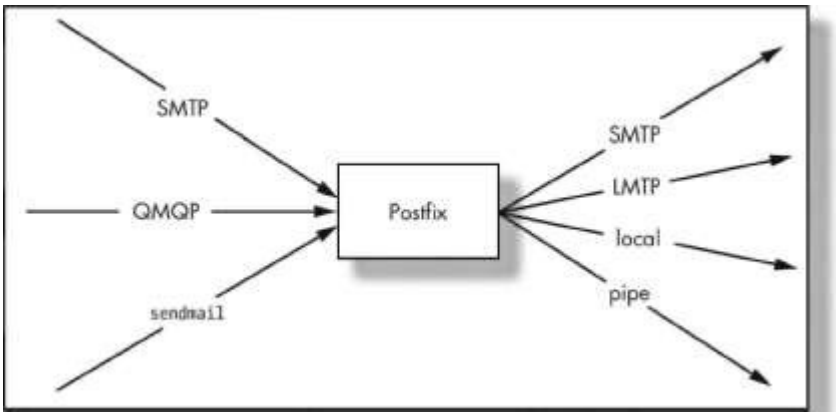



Рис 5-2 Постфикс «маршрутизатор» принимает и устанавливает все типы соединений

Происхождение и предназначение сообщения кажутся достаточно ясными, но *как* действительно Постфикс выбирает метод доставки для, данного предназначения? Маршрутизатор использует таблицы маршрутов, которые соответствуют IP адресам и сетям, чтобы определить путь. Постфикс делает ту же самую процедуру с адресами электронной почты.

Постфикс,таблицы запросов называются *картами*. Посфикс использует карты не только, чтобы узнать, куда послать почту, но также и для накладывания ограничений на клиентов, отправителей, и получателей, и проверки некоторых разделов в содержании электронной почты. На рисуноке 5-3 показаны, где карты *псевдонимы, виртуальные, транспортировка* - сходятся.

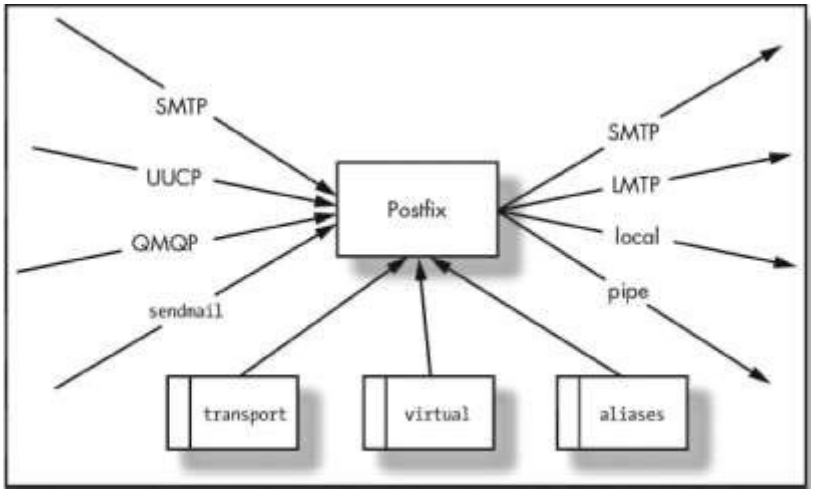



Рис 5-3 карты таблиц запросов «маршрутизатора» Постфикс

Демоны постфикс

Рисунок 5-4 показывает краткий обзор Постфикс демонов и как они взаимодействуют вместе.

ПРИМЕЧАНИЕ Постфикс, постоянно развивается. Список демоны базируется на, версии Постфикс 2.1.

**master**

Демон мастера – основной демон Постфикса, и он наблюдает за весь другими

демонами Постфикса. master ждет входящие задания, которые будут делегированы, соответствующим демонам. Если много заданий, для выполнения, master может вызвать несколько копий демона. Вы можете задавать число одновременных процессов демона, как часто Постфикс, может повторно использовать их, и период бездеятельности, которая должн протечь перед остановкой копии.
Если Вы когда-либо работали с inetd сервером на машине Unix, Вы найдете много сходств между ним и демоном master.

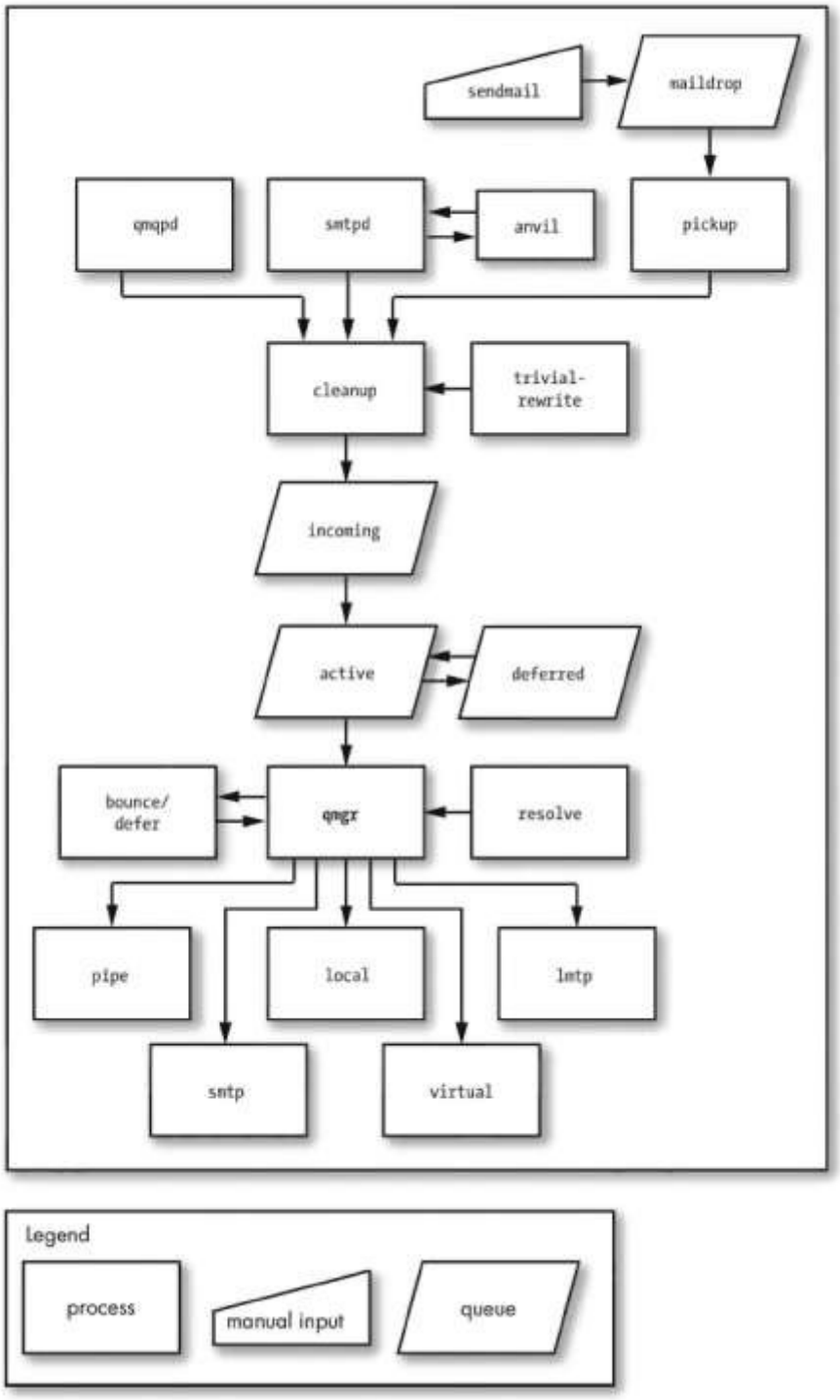



Рис 5-4 Взаимодействие между демонами Постфикса

***bounce (возврат) and defer (отсрочка)***

Агент передачи почты должен уведомить отправителя о не подлежащей доставке почте. В Постфиксе, демоны **bounce** и **defer** , отвечают за эту задачу, которая вызывается менеджером очереди (**qmgr**). Определенны, два типа случая, в которых происходит уведомление отправителя - не восстанавливаемые ошибки и недоступность места предназначения, в течение длительного периода времени. В последнем случае отправитель получает предупреждение о задержке доставки.

***error (ошибка)***

Демон error агент доставки почты подобный ***local*** или *smtp*. Это - агент lоставки, который всегда отклоняет почту . Обычно Вы не используете его, если Вы не конфигурируете домен как не подлежащий доставке, направляя почту агенту доставки error . Если почта посылается демону ***error*** , он сообщит демону **bounce**, сделать запись что, получатель недоступен.

***trivial-rewrite***

***trivial-rewrite*** демон действует после запроса демона ***cleanup***, чтобы переписать нестандартные адреса в стандартную, user@fqdn форму. Этот демон также определяет предназначение после запроса от менеджера очереди (qmgr). По умолчанию, **trivial-rewrite** , различает только локальное удаленным предназначение.

***showq***

***showq*** демон отображает очередь Постфикса , и он выполняется после *mailq* (sendmail -bp ) командs. Этот демон необходим, потому что очередь Постфикса не доступной для чтения всем; и пользовательская программа не может отобразаить очередь .

***flush***

Демон ***flush*** используется, для очистки очередь почты от незаконченных и отсроченных сообщений. Использование списка предназначения почты с очередью, улучшает работу SMTP (ETRN) запрос и его эквивалент командной строки, sendmail-qR предназначение. Вы можете определить список предназначения с помощью параметра fast_flush_domains в main.cf файле.

***qmgr***

qmgr демон управляет очередями Постфикс; это - сердце почтовой системы Постфикс. Он распределяет задачи доставки демонам l***ocal, smtp, lmtp***, и демону ***pipe***. После делегирования работы, это представляет информацию о имени и пути файла очереди, адрес отправителя сообщения целевом хосте, (если предназначение удаленное), и один или более адресов получателя сообщения демону которому он делегировал задачу доставки.
Конструкция qmgr - хороший пример того, как Постфикс управляет работой, чтобы избежать недостатка ресурсов и поддерживать стабильность. Выделим две вещи:
• qmgr обслуживает маленькую активную очередь, только с несколькими сообщениями, ожидаемыми для доставки. Эта очередь фактически действует как ограниченное окно для потенциально больших входящей и отсроченных очередей, и

препятствует исчерпыванию памяти qmgr из за большой нагрузки.
• если Постфикс, не может немедленно доставить сообщение, qmgr перемещает сообщение в очередь отсроченных сообщений. Хранение, временно не подлежащих доставке сообщений в отдельной очереди гарантируют что, большое количество отложенных сообщений не замедлит нормальный доступ к очереди.
qmgr использует демоны ***bounce*** и ***error*** чтобы вернуть почту для получателей, внесенных в список в таблице перемещенных, который содержит информацию о контактах для пользователей или доменов, которые больше не существуют на системе.

***proxymap***

Клиентские процессы Постфикс, могут получить доступ на чтение карт только через демон ***proxymap*** . Разделяя отдельную открытую карту среди многих Постфикс демонов, proxymap обходит chroot ограничения и уменьшает число открытых таблиц поиска.

***spawn***

Процесс ***spawn*** пораждает не Постфикс процессы по запросу. Он слушает на TCP порту, сокете Unix, или FIFO с стандартным потоком ввода вывода, и потоками ошибок. Единственное использование для ***spawn***, обсуждаемое в этой книге - использование его для внешней программы системы фильтрования содержимого Постфикс в Главе 13.

***local***

Как и указывает название, демон ***local*** ответственен за доставку почтового сообщения в локальные ящики. Постфикс демон ***local*** может осуществлять доставку в почтовые ящики в форматах Maildir и mbox. Кроме того, ***local*** может получить доступ к данным в базе данных псевдонимов формата sendmail и пользовательским файлам .forward.

*ОБРАТИТЕ ВНИМАНИЕ, эти способности делают **local** коллегой агента по отправке почты Sendmail, и они оба поддерживают тот же самый пользовательский интерфейс.*

В качестве альтернативы, ***local*** может делегировать доставку в почтовые ящики местному агенту доставки (LDA), который обеспечивает более продвинутые функции, типа фильтрования. Два очень популярных LDA - procmail (http://www.procrriail.org) и maildrop (http://www.flounder.net/ ~mrsam/maildrop). Постфикс может управлять несколькими копиями ***local***.

***virtual***

Демон ***virtual***, иногда называемый виртуальным агентом доставки, является упрощенной версией ***local***, который обслуживает исключительно mailbox . Это самый безопасный, агент доставки Постфикс; он не поддерживает псевдонимы и .forward файлы.
Этот, агент может доставлять почту для нескольких доменом, это особенно подходит для того, чтобы доставлять почту для нескольких маленьких доменов на единственной машине (так называемый POP тостер ) без необходимости аккаунтов в

реальной системе.

***smtp***

smtp клиент Постфикса - клиентская программа, которая транспортирует исходящие сообщения к удаленному предназначению. Он ищет почтовые сервера места предназначения, сортирует список согласно приоритетам, и пробует каждый адрес, пока он не находит сервер, который отвечает. Занятая Постфикс система, обычно имеет несколько smtp демонов, запущенных одновременно.

***lmtp***

lmtp клиент связывается с местными и отдаленными почтовыми хранилищами с помощью Локального Протокола Доставки Почты (LMTP) определенного в RFC 2033 (ftp://ftp.rfc-editor.org/in-notes/rfc2033.txt ). Он часто используется с Cyrus IMAP сервером (<http: // asg.web.cmij.edu/cyrijs/imapd>).
Преимущество, использования lmtp клиента Постфикса, одна машина с Постфиксом отвечающая за управление всеми очередями, может обслуживать множество почтовых хранилищ (которые должны иметь LMTP демон) по LMTP. Противоположное тоже верно: несколько Постфикс машин, может обслуживать одно почтовое хранилище через lmtp. Это почтовое хранилище может, например, управляться Cyrus IMAP.

***pipe***

Почтовый клиент ***pipe*** - граничный интерфейс для других механизмов транспортировки почты. Он вызывает программы с параметрами и передает тело сообщения в их стандартный ввод.

***pickup***

Демон ***pickup*** собирает сообщения, помещенные в очередь maildrop в локальной пользовательской клиентской программой sendmail. После выполнения нескольких проверок, ***pickup*** передает сообщения к демону ***cleanup***.

***smtpd***

Демон ***smtpd*** отвечает за связь с сетевыми почтовыми клиентами , которые поставляют сообщения, Постфиксу через SMTP, ***smtpd*** выполняет множество проверок, которые защищают остальную часть Постфикс системы, и может быть настроен, для контроля за спамом (локальные или сетевые черные списки , DNS запросы, другие клиентские запросы, и так далее).
После принятия сообщения, ***smtpd*** помещает его в очередь входящих сообщений, где его принимает qmgr.

***cleanup***

Демон ***cleanup*** финальная, стадия обработки новых сообщений. Он добавляет любые требуемые, но отсутствующие заголовки, принимает меры к переписыванию адреса, и (опционально) извлекает адреса получателя из заголовков сообщений. Демон ***cleanup*** вставляет обработанное сообщение в очередь входящих сообщений и затем уведомляет менеджера очереди, о то что новая почта получина.

***sendmail***

sendmail – Постфикс команда, которая заменяет и эмулирует MTA Эрика Оллмана Sendmail. Его цель состоит в том, чтобы обеспечить Sendmail-совместимый интерфейс к приложениям, которые обращаются толлько к */usr/sbin/sendmail.* Она взаимодействует с ***postdrop*** , чтобы поместить почту в maildrop очередь для демона ***pickup***.

*ПРИМЕЧАНИЕ sendmail – самый медленный способ ввести почту в систему очередей Постфикс. Если Вы посылаете большое количество почты одновременно, используйте SMTP .*

***qmqpd***

Постфикс QMQP сервер осуществляет Быстрый Протокол Организации очереди Почты (QMQP; см. http://cr.yp.to/proto/qmqp.html), чтобы обеспечить совместимость, Постфикса с с менеджером очереди qmail, и ezmlm .

***anvil***

Постфикс ***anvil*** - предварительная защита против SMTP клиентов и нападений типа отказ в обслуживании (DoS), которые загружают SMTP сервер со слишком многими одновременными или последовательными попытками соединения. Он идет с поддержкой белых списков для отмены ограничений к авторизованным клиентам. ***anvil*** не включен в Постфикс 2.1, но доступен экспериментальных релизах Постфикс 2.2 , ***anvil*** останется экспериментальным до накопления достаточно опыта и статистической информации.

## Очереди Постфикса

Постфикс помещает все очереди в директории, указанной в параметре ***queue_directory*** в вашем main.cf файле. Обычно /var/spool/postfix. Каждая очередь имеет ее собственную. поддиректорию с именем очереди. Все сообщения Постфикс держит в этих директориях до тех пор пока, Постфикс, не доставит их. Вы можете определить статус сообщения по его очереди: входящие, maildrop, отсроченный, активные, удерживаемые, испорченные исходящие.

***incoming (Входящие)***

Все новые сообщения, входящие в Постфикс попадают в очередь ***incoming (Входящие),*** обслуживаемой демоном ***cleanup***. Новые файлы в очереди создаются с пользователем Постфикс как владельцем и правами доступа 0600. Как только файл в очереди готов к дальнейшей обработке, служба ***cleanup*** изменяет права доступа файла в очереди на 0700 и уведомляет менеджера очереди, что новая почта поступила. Менеджер очереди игнорирует неполные файлы в очереди, маска которых – 0600.
Менеджер очереди просматривает очередь ***incoming (Входящие)*** при перемещении новых сообщений в ***активную*** очередь и удостоверяется, что пределы ресурсов ***активной*** очереди не были превышены. По умолчанию, активная очередь имеет предел 20 000 сообщений.

*ПРЕДОСТЕРЕЖЕНИЕ, как только предел сообщений в активной очереди*

*достигнут, менеджер очереди прекращает просматривать очереди **incoming (Входящие)** и **отсроченных** сообщений.*

***maildrop***

Сообщения, переданные с помощью команды sendmail , которые не были переданы в первичную очередь Постфикса, службой ***pickup***, ждут обработку в maildrop очереди. Вы можете добавлять сообщения в maildrop очередь, даже когда Постфикс, не запущен; Постфикс будет просматривать их, как только будет запущен.

Служба ***pickup*** просматривает и очищает maildrop очередь как периодически, так и после уведомления программой ***postdrop*** . Программа ***postdrop*** - setgid помощник, который позволяет непривилегированной sendmail программе помещать почту в maildrop очередь и уведомлять службу ***pickup*** о прибытии сообщений. (Все сообщения, поступают в главную, очередь Постфикс так же, через службу ***cleanup***.)

***deferred (отсрочка)***

Если сообщение все еще имеет получателей, для которых доставка неудалась по некоторым временным причинам, и сообщение было доставлено всем доступным получателям, Постфикс помещает сообщение в очередь ***deferred (отсрочка)***. Менеджер очереди просматривает отсроченную очередь периодически, чтобы переместить отсроченные сообщения в активную очередь. Интервал просмотра определяется параметром ***queue_run_delay*** в конфигурации. Если просмотр отсроченной и поступающей очереди, происходит одновременно, менеджер очереди чередует обращения к этим двум очередям через сообщение.

***active (активные)***

Активная очередь несколько походит на процесс операционной системы поставленный в очередь. Сообщения в активной очереди готовы быть посланными, но - не процесс доставки не обязательно уже идет.
Менеджер очереди - планировщик агента доставки, работа которого - гарантировать быструю и справедливую доставку почты всему предназначениям в пределах выделенных ресурсов.

*ОБРАТИТЕ ВНИМАНИЕ, хотя большинство администраторов Постфикса, думают об активной очереди как о директории на диске, активная очередь - набор структур данных в памяти.*

***hold (удерживаемые)***

Администратор может определить smtpd (5) политику доступа и отсева почты проверкой тела и заголовков (см. Главу 10), по этой причине сообщения, которые будут автоматически отклонены от нормальной обработки и помещены на неопределенное время в очередь ***hold (удерживаемые)*** . Сообщения, помещенные в очередь ***hold (удерживаемые)*** пребывают там до вмешательства администратора. Никакие периодические попытки доставки не делаются для сообщений в очереди ***hold (удерживаемые)***. Вы можете использовать команду postsuper, чтобы вручную поместить сообщения, или выпускать сообщения из ***hold (удерживаемые)*** очереди в очередь ***deferred (отсрочка).***

Сообщения могут потенциально оставаться в очереди ***deferred (отсрочка)*** какое-то время, не превышающее время жизни файла очереди, установленную

параметром ***maximalqueuelifetime*** (после чего,не доставленные сообщения вернутся отправителю). Если старшие сообщения были выпущены из очереди, Вы можете использовать команду postsuper -r , чтобы переместить их в maildrop очередь, так, чтобы сообщение получило новую временную отметку и таким образом дать больше одной возможности для доставки.

*ОБРАТИТЕ ВНИМАНИЕ, что очередь **hold (удерживаемые)** не играет, большой роли в работе Постфикс; контроль очереди **hold (удерживаемые)** - типичен для отслеживая спама, а не проблем с работой Постфикса.*

***corrupt (испорченные)***

Директория ***corrupt (испорченные)*** , содержит поврежденные файлы из очередей. Вместо того, чтобы удалить их, Постфикс, хранит их здесь , чтобы администратор почтового сервера мог просмотреть их используя *postcat.*

Постфикс регистрирует предупреждение о любых испорченных файлах после запуска.

## Карты

Карты - файлы, и базы данных используются Постфиксом для поиска информации. Карты имеют много различных целей, однако они имеют один общий формат: левая сторона (LHS) является ключевой а правая сторона (RHS) содержит значение

Вот несколько примеров значений и ключей:

| Key | Value |
|---|---|
| postmaster: | John |
| postmasterfexample.com | John |
| 192.168.254-12 | REJECT |
| spamrner@example.com | REJECT |
| /^A Subject: your account {25}[a-z]{8}/ | REJECT Mmail Virus Detected |

Используя карту, Вы определяете ключ и в результате получаете связанное с ним значение.

*ОБРАТИТЕ ВНИМАНИЕ Ключи и значения здесь, берутся из различных файлов и не имеют смысл при указании в одном файле. Предыдущий список - только иллюстрация, чтобы показать, что все записи в картах имеют одну и ту же каноническую форму.*

**Типы Карт**

Постфикс может использовать много различных видов карт. Доступные форматы зависят от параметров с которыми скомпилирован Постфикс на вашей системе. Чтобы узнавать какие форматы поддерживает ваш Постфикс, выполните команду postconf-m в командной строке. Вы должны получить список типов карт

```
# postcon-f -m
btree
cdh
```

cidr

environ

hash

ldap

mysql

nis

pcre

proxy

regexp

sdbm

static

tcp

## Индексированные Карты (hash, btree, dbm, и так далее)

Индексированные карты - бинарные базы данных, построенные из упорядоченных текстовых файлов командами типа newaliases, postalias, и postmap. Бинарные карты имеют индексированный формат так, что Постфикс, может быстро найти значение, связанное с ключом. Для дальнейшего улучшения быстродействия, демоны Постфикса открывают эти карты при запуске, и они не перечитывают их, если не обнаруживают изменения в файлах карт в файловой системе. Чтобы перезагрузить карту, работа демона завершается , и новая копия демона запускается демоном master.

*ОБРАТИТЕ ВНИМАНИЕ, если у Вас есть карты, которые часто изменяются , демоны, использующие эти карты должны рестартовать также часто. При большой нагрузке, это может привести к проблемам быстродействия.*

Наиболее часто индексированные карты построены из текстовых файлов aliases (псевдонимы), virtual (виртуальные), transport (транспорт), relocated (перемещенные), и sasl_passwd (sasl пароли). Вы можете идентифицировать файл карты, потому что его название - имя оригинального файла с суффиксом, который обозначает формат индексированного файла. Например, файл карты псевдонимов, построенный с помощью команды postalias имеет имя aliases.db.

*ОБРАТИТЕ ВНИМАНИЕ, когда Вы создаете файл, для построения из него индексированной карты, Вы не должны помещать ключи в определенном порядоке. Конверсионные инструменты и программы, используемые для индексации карты, не требуют определенного порядка входного файла. Фактически, процесс преобразования разрушает порядок.*

Постфикс выполняет запросы в предопределенном порядке, указанном в мане access(5). Другими словами, каждый запрос карты фактически состоит из ряда отдельных запросов (полученный из первоначального запроса) по отдельным ключам в индексированной карте.

## Линейные Карты (PCRE, regexp, CIDR, и Flat Files )

Линейные карты – это упорядоченные текстовые файлы. Постфикс читает эти файлы сверху в низ, в отличие от индексированных карт. Это различие весьма важно, потому что первое соответствие в файле определяет действие, которое выполнит Постфикс. Постфикс игнорирует любые более поздние строки, не зависимо от того соответствуют ли они запросу или нет.
Рассмотрим следующую карту regexp, где запрос john.doe@example.com возвращает ОК, чему соответствует первая запись.

```
/john\.doe@example\.com/ OK
/example\.com/           REJECT
```

Однако, если Вы поменяете местами записи в карте regexp, тот же самый запрос john.doe@example.com, возвратит REJECT

```
/example.\com/           REJECT
/john\.doe@exampie\.com/ OK
```

Вы не должны преобразовать линейные карты в бинарную форму (фактически, Вы не можете сделать этого). Постфикс демоны читают их при запуске и не замечают никаких изменений в карте, пока они не будут перезапущены. Типичные примеры линейные карт Постфикса, включают header_checks, body_checks, и mime_header_checks (см. Главу 9).

*ПРЕДУПРЕЖДЕНИЕ Поскольку ваши линейные карты растут, требуется больше времени для их обработки Постфикс демонами . Это особенно касается проверки тела или заголовков, потому что демон уборки должен проверить каждую строку тела письма (число строк подлежащих проверке определяется параметром body_checks_size_limit) и заголовков сопоставив их с каждой строкой карты. Это может вызвать существенное замедление, особенно если Вы сделали так, чтобы обширная *_checks проверка ,которые используют - regexp или PCRE (Perl-совместимое регулярное выражение) карты раньше чем отсев спама. Когда это происходит лучше проводить комплексную проверку на спам внешними программами.*

Для того чтобы Постфикс демоны приняли изменения в линейных картах, Постфикс необходимо перезапустить. Если время принятия изменений не является критическим, Вы можете установить max_use параметр определяющий время жизни для демонов. Как только демон обработал число задач, указанных в этом параметре, он завершается и повторно запускается демоном master. При старте, он перечитывает все требуемые карты.

## Базы данных (MySQL, PostgreSQL, LDAP)

Постфикс использует базы данных точно так же как индексированные карта. Результат запроса базы данных - соответствие (наряду со значениеем, возвращенной в соответствии с запросом) или отсутствие соответствия. Основное

различие между картой базы данных и индексированной картой - то, что Вы не должны повторно перезапускать демона, после внесения изменений в базу данных, Постфикс, не предполагает, что администратор почтового сервера - единственный человек, который может изменить базу данных.

Недостаток этого подхода состоит в том, что база данных не может быть способна обращаться с номером запроса, потому что Постфикс требует выполнения самое меньшее три запроса для каждого поиска в карте (см., "Как Постфикс опрашивает карты далее). Под большой нагрузкой, база данных может прекратить работать, и ваша служба почты будет уязвима к DOS атакам. Эта возможность не должна препятствовать Вам использовать базу данных, но Вы должны знать о риске.

Поиск в базе данных может быть проблемой для систем с большой нагрузкой, но это не единственная проблема. Время ожидания, может быть другой проблемой. Запросы базы данных имеют более высокую латентность чем индексированные карты, потому что Постфикс, должен соединиться с базой данных, посылать запрос, и ждать результата. С индексированной картой, Постфикс, должен только обращатся с данными, которые уже загружены в память.

Если ваша база данных становится узким местом, и Вы не имеете большой карты, Вы можете вставить карту между базой данных и Постфиксом. То есть Вы можете создать индексированную карту из полного запроса базы данных, и затем запустить, Постфикс с использованием этой карты. Вы должны не забывать обновлять карту по мере необходимости, но proxymap демон может использоваться, чтобы значительно уменьшить число параллельных связей.

## Определение числа одновременных подключений к базе данных

Постфикс демоны (smtpd, smtp, и так далее) запускаются с пределом (установленный параметром defaultprocesslimit) 100 одновременных процессов. Прии пиковой нагрузке, было бы 100 параллельных smtpd демонов, каждый выполнял бы запрос к базе данных для просмотра access(5) (например, потому что мы используем карту для того, чтобы проверить, находится ли клиент в нашем личном черном списке и если да тогда отклоняем получение почты от него),

Помните, что один просмотр состоит по крайней мере из трех запросов, так что число одновременных запросов к базе данных будут равны по крайней мере default_process_limit * 3 ( в конфигурации по умолчанию 300 запросов), в то время как число одновременных подключений - default_process_limit. Это - только число запросов и подключений для smtpd демонов; другие демоны, типа local и qmgr, могут тоже обращатся к базе данных , увеличивая число одновременных подключений и запросов.

## Как Постфикс выполняет поиск в картах

Карты могут использоваться для различных задач. Постфикс имеет механизмы использующие карты (см. access (5), aliases (5), canonical(5), и transport (5)). Эти карты могут использовать различные механизмы поиска (LDAP, NIS, SQL, btree, hash, regexp, cdb, cidr, pcre, и так далее).

1. localpart@domainpart соответствует указанному дословному адресу почты.

2. domainpart Соответствует доменной части области адреса электронной почты. значение domainpart также соответствует поддоменам, но только когда smtpd_access_maps, указана в параметре patten_domain_matches_subdomains файла конфигурации. Иначе, определите .domainpart (обратите внимание на

начальную точку), для указания на поддомены.

3. localpart@ Соответствует всем адресам почты с указанной пользовательской частью (localpart), независимо от того какому домену они принадлежат

4. Fail, если поиск не имеет соответствия , Постфикс, возвратит no match, и запрос завершится ошибкой.

.

*ПРИМЕЧАНИЕ Не возможно искать пустой адрес отправителя в некоторых типах таблиц поиска. По умолчанию, Постфикс использует <> как ключ поиска для пустого адреса отправителя. Значение определяется параметром smtpd_null_access_lookup_key в файле main.cf.*

Этот порядок запросов подразумевает, что Постфикс, выполняет несколько запросов для каждого поиска, что действительно не проблема, если Вы не используете карты с высокой латентностью подобно SQL или картам LDAP (и, конечно, Вы должны ожидать, что много поисков будет нуждаться в многократных заросах). Это - только одна вещь, которую Вы должны помнить прежде, чем Вы поместите все ваши карты в LDAP и затем жалуетесь в списке рассылки пользователей Постфикса, что "Постфикс, медленен.. .. "

## Внешние Источники

Постфикс поддерживает источники информации, которые не встроены в сам Постфикс и даже не находятся под вашим прямым контролем, типа черных списков (DNSBL и списков RHSBL), списков основанных на DLS, и других внешних источников. Черные списки почти исключительно используются в параметре smtpd_*_restrictions , чтобы отклонить почту, поступающую от клиентов или отправителей, внесенных в список DNSBL-или RHSBF (см. Главу 7).

Как и любые внешние запросы, эти поиски могут терпеть неудачу из-за проблем соединения, DOS атаки против серверов черных списков, и других проблем. В случае перерыва или другого отказа, Постфикс, будет продолжать принимать почту (пропуская возможные ограничения), но будет регистрировать соответствующее предупреждение в логах.

## Утилиты командной строки

Постфикс содержит множество утилит командной строки, чтобы помочь Вам с задачами администрирования. Хотя они исполняют различные функции (типа выполнения запросов карт, исследование файлов очереди, удаление сообщений из очереди реорганизации очереди, и изменения конфигурации), все они все имеют одну общую часть в своем названии. Они начинаются с "post".

*ОБРАТИТЕ ВНИМАНИЕ, что Эти команды могут сделать намного больше чем, то что описано здесь. Мы сосредоточимся на вариантах, которые Вы используете в ежедневной работе. Если Вы не находите здесь то, что Вам нужно, первое место, куда следует смотреть - онлайн руководство.*

**postfix**

Команда **postfix** , запускает, останавливает или перезагружает конфигурацию. Это зависит от опций с которыми она выполняется: start, stop, или reload.

**postalias**

Команда **postalias** создает индексированную карту псевдонимов из файла псевдонимов. Она работает точно так же как команда postmap (описанная ниже), но обращает особое внимание на примечание в файле псевдонима (где двоеточие отделяет ключ и значение), **postalias** должен использоваться для файлов псевдонимов.

**postcat**

Команда **postcat** показывает содержание сообщения в очереди почты.
Чтобы читать сообщение в очереди почты, Вы нуждаетесь в его ID. Выполните **mailq** для вывода списка ID очереди . Например, ID следующего сообщения - F2B9715C0B3:

```
#  mailq
F2B9715C0B3    2464 Mori Oct 13 15:29:39   markus.herrmann@example.com
        (connect to mail.example.com[217.6.113.151]:  Connection timed out)
                                                torsten.hecke@example.net
-- 2 Kbytes in 1 Requests.
```

После получения ID, используйте его как опцию к **postcat**, чтобы просмотреть содержание файла:

**#  postcat -q F2B9715COB3**

**postmap**

Первичная цель команда **postmap** построение индексированных карт из плоских файлов. Например, чтобы посстроить /etc/postfix/virtual.db из/etc/postfix/virtual, выполните следующую команду:

```
# postmap hash:/etc/postfix/virtual
```

Команда **postmap** может сделать больше. Среди его самых полезных особенностей - способность проверить любой вид карты, с поддержкой которой установлен Ваш Постфикс. Это чрезвычайно полезно при отладке конфигурации, где поиски к картам, кажется, терпят неудачу, и Вы неуверены, фактически видит ли Постфикс ключ и значение.

## Отладка значений в таблице поиска

Для определения, может ли Постфикс найти запись в карте, используйте **postmap -q**. Например, следующая команда возвращает значение, соответствующее ключю sender@example.com в карте/etc/postfix/sender_access (**hash** типа):

```
# postmap -q sender@example.com hash:/etc/postfix/sender_access
OK
```

Важно обратить внимание, что **postmap** *не* ищет условия <sender@>, < example.

com>, и <com>, даже при том, что эти условия находятся на странице руководства access(5). Вы должны выполнить эти поиски вручную:

# postmap -q sender^ hash:/etc/postfix/sender_access

# postmap -q example.com hash:/etc/postfix/sendei_access

# postmap -q com hash:/etc/postfix/sender_access

***postdrop***

Команда ***postdrop*** читает почту от стандартного потока ввода и помещает результат в директорию maildrop. Эта программа работает в соединении с утилитой **sendmail** .

**postkick**

Команда **postkick** посылает запрос Постфикс демону через локальный транспортный канал, создавая связь между процессами Постфикс, доступную, для командных сценариев и других программ.

*ОБРАТИТЕ ВНИМАНИЕ, что команда **postkick** посылает сообщения, процессам Постфикс демонов. Она требует чтобы Постфикс был запущен.*

## Переорганизация очереди Сообщений

Следующий продвинутый пример использования ***postkick*** показывает как переместить сообщение из очереди для немедленной передоставки.

**# cat queueidlist | postsuper – r -**
**postkick public pickup W**

Эта последовательность команд перемещает все отобранные сообщения, внесенные в список в queueidlist в maildrop очередь с помощью команды postsuper -r - , где демон pickup обработал бы их подобно любой другой почте. Делая это, Вы сбрасываете фильтр содержания к установкам, соответствующим локальной доставке и добавляете дополнительный заголовок Received.
Команда **postkick** запрашивает немедленный просмотр очереди maildrop. Иначе, сообщения находились бы в maildrop очереди максимум 60 секунд. Демон pickup отправляет сообщение демону cleanup, где оно получает новый id очереди и передается во входящую очередь. Следующий пункт: перемещение сообщения в активную очередь настолько быстро насколько возможно.

**postlock**

Команда **postlock** дает Вам исключительный доступ mbox файлам, которые пишет Постфикс и затем он запускает команду при удерживании блокировки. Блокировка, которую Вы получаете от **postlock**, совместима с локальным агентом доставки Постфикса. Постфикс не работает с файлом, во время выполнения вашей команды. Вот – пример:

**# postlock /var/mail/user from**

*ПРЕДОСТЕРЕЖЕНИЕ избегайте любых команд, которые могли бы требовать CTRL-C для завершения работы. Прерывание **postlock** не гарантирует, что блокировка будет снята; Вам, возможно,потребуется, удаление заблокированного файла, для возобновления доставки в mailbox. Для просмотра если есть заблокированный файл, выполните **postlock** без команды. Если он приводит к зависанию на некоторое время, Вы вероятно имеете оставшуюся блокировку.*

**postlog**

Команда **postlog** позволяет внешним программам, типа сценариев оболочки, писать сообщения в лог файл почты. Это - Postfix-совместимый интерфейс для записи лог файлов по умолчанию, он пишет текст из командной строки как простую запись. Вот простой пример:

# postlog This is a test

postlog: This is a test

# grep "This is a test" /var/log/mail.log

Feb 20 11:50:16 mail postlog: This is a test

**postqueue**

Команда **postqueue** - пользовательский интерфейс, для Постфикс очереди, дающий Вам функциональные возможности, которые являются традиционно доступными с командой sendmail

- -f параметр заставляет **postqueue** запросить у менеджера очереди доставку всей почте в очереди (поток), независимо от предназначения. Это эквивалентно, чтобы postfix -flush или sendmail-q:

**# postqueue -f**

- -p параметр заставляет **postqueue** печатать содержание очереди. Это эквивалентно mailq:

**# postqueue -p**

- s параметр домен заставляет **postqueue** попытаться доставить всю почту в очереди, предназначенную для домена. Это эквивалентно sendmail-q домен:

**# postqueue -s example.com**

*ВНИМАНИЕ Команда **postqueue** посылает сообщения, процессам демонов Постфикс и требует, чтобы Постфикс был запущен.*

**postsuper**

Команда **postsuper** выполняет работу, внутри очереди Постфикс. В отличие от **postqueue** эта команда ограничена суперпользователем, и это может выполнятся в то время как Постфикс не запущен. Некоторые возможности **postqueue** необходимы, чтобы проверить очередь прежде, чем процессы демонов будут запущены. Таблица

5-1 показывает, что команда **postqueue** может делать.

Таблица 5.1 Возможности команды **postsuper**

| Опция | Действие |
|---|---|
| -d | Удалите сообщение с ID из названной очереди почты |
| -h | Поместить сообщение в очередь hold никакая попытка доставки не будет сделана. |
| -H | Выпустить почту из очереди hold |
| -p | Переместить сообщения из очереди a в очередь b |
| -r | Проверить и исправить структуру очереди |
| -s | |

Одно из самых частых использований **postqueue** удаляет сообщение из очереди с помощью команды **postqueue -d** queueid. Выполнение этого вручную утомительно, особенно при удалении многих файлов. Следующий Perl сценарий (delete_from_mailq) облегчает выполнение данной процедуры:

```
} }
#open(POSTSUPER,"|cat") || die "couldn't open postsuper" ;
open(POSTSUPER,"|postsuper -d -") || die "couldn't open postsuper" ;
foreach (keys %0) {
print POSTSUPER "$_\n";
};
close(POSTSUPER);
```

Вот как его использовать:

```
# mailq
C73AO15C095          7509 Mon Oct 13 14:56:17    MAILER-DAEMON
(connect to mx5.ancientaward.com[64.156.l66.21l]:  Connection refused)
National_Nosepicking_M0nth@mx5.ancientaward.com
```

Заметьте, что отправитель идентифицирован как <MAILER-DAEMON>. Чтобы удалять эти сообщения, запускайте delete-from-mailq как суперпользователь.

```
# delete-from-mailq MAILER-DAEMON
postsuper:  C73A015C095: removed
postsuper:  Deleted:  1 message
```

# ЧАСТЬ 2

## КОНТРОЛЬ СОДЕРЖИМОГО

Постфикс идет с тремя наборами параметров, которые управляют тем, как сообщения могут попадать и оставлять почтовую систему. С этими параметрами, Вы можете управлять потоком сообщений, основанным на SMTP диалоге и содержании сообщения, или Вы можете делегировать контроль за содержимым внешним программам. Эти три типа контроля падают в три различных группы параметров конфигурации: ограничения, проверки, и фильтры.

### Учебник для начинающих , чтобы Послать по электронной почте

Контроль за содержимым требует знания о содержании. Вы должны знать все, что нужно и может быть в электронной почте, чтобы применять ограничения, проверки, и фильтры эффективно. Читайте Главу 6, чтобы получить понятие о содержании электронной почты.

### Как работают ограничения на передачу сообщений

Ограничения управляют SMTP связью. Глава 7 объяснит, как ограничения работают. Не торопитесь читая ее; она поможет сделать применение ограничений более легкой задачей.

### Использование ограничений на передачу сообщений

В Главе 8 мы покажем Вам**, как воплотить** ограничения в жизнь. Все из них могут использоваться почти немедленно.

### Как работают встроенные фильтры содержимого

Работа, фильтров основана на содержании сообщений. Но как они работают? Глава 9 представляет Вам проверки и рассказывает Вам всем о теории проверок.

### Использование встроенных фильтров содержимого

**Глава 10** содержит различные примеры, чтобы показать Вам правильное использование.

### Как внешние фильтры содержимого работает

Внешние фильтры содержимого делегируют управление SMTP связью и контроль содержимого внешним приложениям. Чтобы понять, как сообщения Постфикс процессов, проходят через внешние фильтры содержимого, читайте **Главу 11.**

### Использование внешних фильтров содержимого

Вам нужны примеры того, как использовать внешние фильтры содержимого? Читайте Главу 12, чтобы найти примеры, которые Вы можете использовать.

Термин к*онверт, заголовок, тело,* и *вложение* все касаются некоторой части обмена данных с MTA. Если Вы знаете то, что они означают, Вы поймете части сообщений, которые затрагивают параметры контроля за содержимым Постфикс. Также удобно, что названия Постфикс параметров и их синтаксис получены из RFCs.

Эта глава - учебник для начинающих, по контролю за содержимым. Читайте ее тщательно, и потратьте некоторое время, для усвоения понятий и терминологии. После того, как Вы усвоите основы, у вас не возникнет неприятности, при достижении эффективного контроля за содержимым.

## Основы транспортировки сообщения

Транспортировка сообщения содержит две главных части: SMTP связь, которая отвечает за передачу и данные, которые передаются (которые большинство людей, именует как "электронная почта" или "сообщение"). Термины которые обычно используются для описания передачи сообщения, не были изобретены внезапно; они были приняты из древней, но известной и установленной системы, которую люди в более ранних столетиях называли "почта."
Когда мы имеем дело с обычной почтовой системой, термины: п*осыльный конверт, заголовок, тело*, и *вложение* все имеют известные значения. Эти термины - технические термины которые используются при обращении, с электронной почтой. Рисунок 6-1 сравнивает обычное письмо с электронной почтой, и Вы можете увидить следующие части:

### Посыльный

В обычной почте, посыльного называют *почтальоном*. В электронной почте, посыльный - *клиент*.

### Конверт

В электронной почте, также, как с обычными письмами, конверт служит оберткой, которая объясняет, как нужно доставить содержимое. На конверте, Вы находите *отправителя конверта* и *получателя конверта*.

### Заголовок

Заголовок дает Вам метаданные (информация) о сообщении. Также, как в реальном письме, заголовок дает Вам информацию об отправителе (заголовок От:), получателе (Кому:), дате и времени (Дата:), и теме (Тема:). Кроме того заголовок, Получено: в сообщении электронной почты говорят Вам о пути сообщения и как долго оно передавалось.

### Тело

Тело сообщения электронной почты содержит фактическое содержание, также, как в письме.

### Вложения

Если есть вложения в сообщении электронной почты, этот факт будет отмечен в теле письма; также, как это было бы в реальном письме. Приложения являются дополнительными и могут быть разнообразных форматов.

## Почему Вы должны знать это?

Это может звучать несколько теоретически, какое отношение это имеет к управлению Постфиксом? Для начала, есть  больше информации в сообщении электронной почты чем в письме. Вы должны знать то, чем являются дополнительные части, а так же как в какой части сообщения эти части появляются. Также, Постфикс, имеет три различных группы параметров для контроля содержимого, которые имеют отношение непосредственно к различными частями сообщений:

**smtpd_*_restrictions**

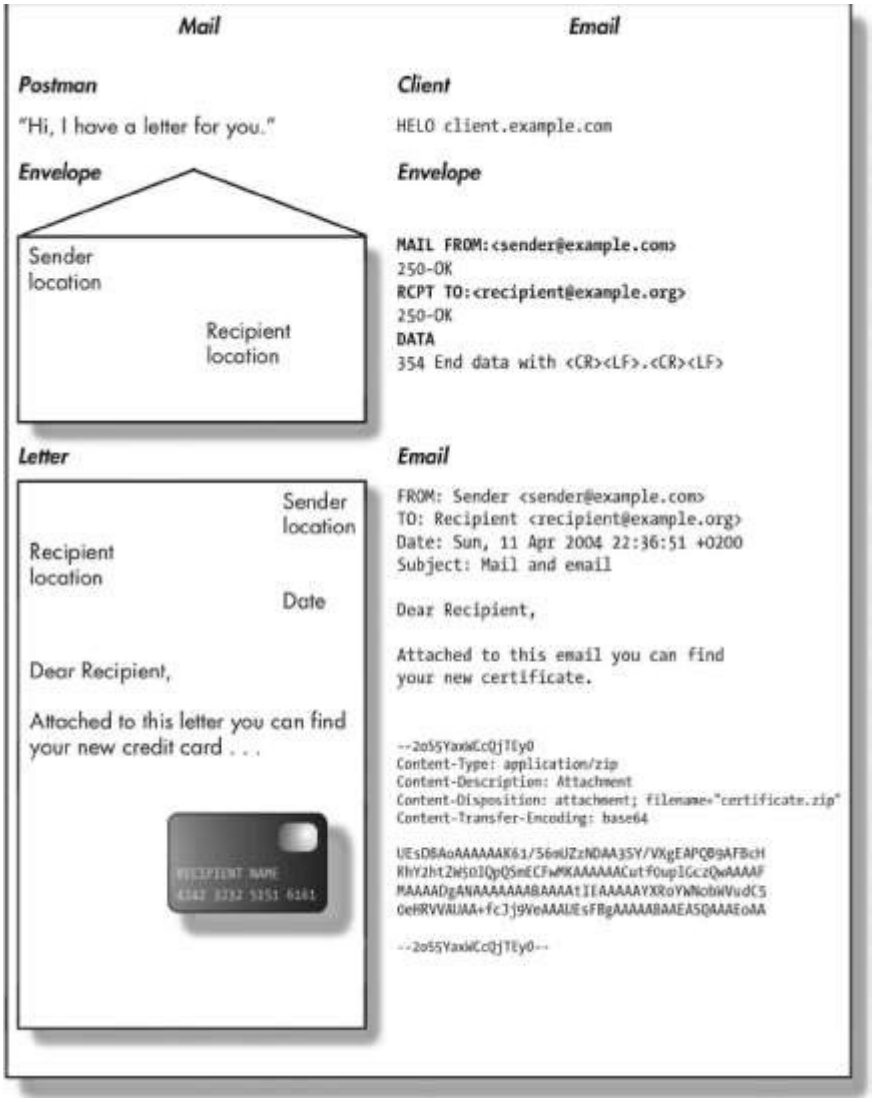
Параметры smtpd_*_restrictions контролируют соединение клиентов и свойства конверта в процессе передачи сообщения.

***_checks**

Параметры *_checks наблюдают за заголовком}, телом письма и вложениями.

**Filters**

Постфикс использует фильтры, чтобы делегировать задачи по контролю  другим внешним приложения. Фильтры имеют общее назначение; они могут контролировать каждую часть сообщения, от конверта до приложений.



Каждый из этих параметров имеет большое число опций; если Вы не знаете, какая часть сообщения вызывает срабатывание специфического параметра, ваш контроль за содержимым не будет работать.

## Контроль за SMTP соединением (Конверт)

SMTP соединение состоит из двух компонентов: клиента ( машина которая соединяется с SMTP сервером), и конверт, который клиент передает. Наиболее легко наблюдать это, используя telnet клиент на вашей машине, для соединения с вашим сервером.
Вот – пример типичной связи. Начните, соединение с 25 портом вашего почтового сервера выполнив в командной строке:

**$ telnet mail.example.com 25**

220 mail.example.com

Код 220 возвращенный сервером подтверждает имя хоста сервера. Теперь, представьтесь серверу например так:

**HELO client.example.com**

250 mail.example.com

Вы можете выполнить приветствие командой HELO (для SMTP) или ELHO (для ESMTP) командой, с именем хоста вашего клиента как параметр. Если команда успешна, Вы должны получить в ответ код 250 , сопровождаемый именем хоста сервера.
Теперь давайте пошлем некоторую почту. Команда MAIL строит конверт, начинающийся с отправителя конверта. Если сервер принимает отправителя, Вы получите другие 250 коды возврата:

**MAIL FROM:<sender@example.com>**

250 Ok

Следующий шаг в построении конверта: использование команды RCPT, чтобы определить получателя конверта. Вы можете указать больше чем одного получателя:

**RCPT TO:<recipient@example.com>**

250 Ok

**RCPT T0:<recipient_2@example.com>**

250 Ok

*ПРИМЕЧАНИЕ Имейте в виду, что отправитель конверта и получатель конверта часто отличаются от отправителя и получателя, указанных в заголовке сообщения (которые определяются как часть последовательности команды DATA, которую Вы увидите далее). Если Вы путаете различных отправителей, и получателей, ваш контроль за содержимым терпит неудачу.*

Для отправки фактического сообщения (включая все дополнительные заголовки, типа Тема, Кому, и Даты), используют команду DATA:

**DATA**

354 End data with <CRxLF>.<CRxLF>

Subject: message

.....

**This is the message**

.....

250 Ok: queued as 92933E1C66

**QUIT**

Вот – сокращенный вариант того, что Вы только что видели, определенный в RFC об электронной почте:

**Клиент**

Клиент - машина посылающая почту; Постфикс будет или регистрировать имя хоста и IP, или обозначать как "неизвестный" (если имя хоста не может быть определено DNS запросом). Постфикс получает IP адрес клиента из TCP/IP стека ядра, и получает имя из DNS или /etc/hosts прежде, чем SMTP связь имеет место. Это позволяет, Постфиксу, накладывать ограничения, если нет соответствия IP адреса клиента и имени в течение SMTP связи.

Постфикс всегда регистрирует IP адрес клиента и его имя (если доступно) в лог файле почты, и также включает эту информацию в заключительный заголовок сообщения.

**HELO/EHLO приветствие**

Клиент должен представиться почтовому серверу с двумя частями информации: тип обслуживания и имя хоста.
Первая часть приветствия - тип службы, который клиент запрашивает. HELO определяет обычную службу, определенную в RFC 821 (ftp://ftp.rfc-editor.org/in-notes/rfc821.txt ), а запрос EHLO расширенную службу, определенную в RFC 2821 (ftp://ftp.rfc-editor.org/in-notes/rfc282l.txt).

После типа службы - индетификация клиента. Клиент, как предполагается, представляется его полностью определенным доменным именем.

**Конверт**

Конверт должен содержать по крайней мере два различных пункта: одиного отправителя конверта и по крайней мере одного получателя конверта. Клиент посылает конверт, передавая сначала адрес отправителя конверта а затем, добавляет получателей конверта.

Если существует более одного получателя конверта, клиент должен представить их один за другим, начиная каждого получателя конверта с новой строки

и ожидая ответа сервера после каждой передачи[1] . Работа сервера – разрешить доставку нескольким или всем получателям.

**Отправитель конверта**

Отправитель конверта - отправитель, которому Постфикс посылает уведомление в случае ошибки, например задержки или возврата сообщения.

**Получатель конверта**

Поле получатель конверта определяет получателя (ей) которому предназначено сообщение. Отдельное сообщение может иметь несколько получателей конверта (например, сообщение нескольким подписчикам списка рассылки).
Почтовый сервер требует по крайней мере одного получателя конверта (иначе ему некому будут доставить сообщение). Поэтому, клиент не может использовать пустого получателя конверта (<>).

*ПРЕДОСТЕРЕЖЕНИЕ, Не используйте адрес получателя определенного в заголовке Кому, когда Вы хотите ввести ограничения по получателям сообщения. Сообщения идут к получателям, определенным в конверте, а не заголовке сообщения.*

Почти все данные из предыдущего списка могут быть подделаны, так Постфикс предлагает способы ограничения подделки с помощью параметров smtpd_*_restriction, которые обращаются к следующим вопросам:

1. Откуда клиент прибывает?
2. Кем клиент представляется?
3. Клиент имеет специальные привилегии?
4. Кто - отправитель?
5. Кто получатели?

Постфикс,также пытается получить ответы на более трудные вопросы:

1. Клиент обеспечивает, Постфикс информацией в соответствующем стиле?
2. Клиент обеспечивает информацию в соответствующем порядке?
3. Клиент обеспечивает всю информацию?
4. Если клиент не обеспечивает всю соответствующую информацию, клиент
5. будет пытаться отправить сообщение?
6. Возможно сказать, правильна ли информация?
7. Если это возможно, клиент лжет?

Постфикс может получить ответы на эти вопросы, осматривая конверт сообщения и то как проходил SMTP диалог. Когда Постфикс, отклоняет сообщение в соответствии с SMTP ограничениями конверта, он отклоняет сообщение прежде, чем оно получено. Поэтому, Постфикс, не будет посылать уведомление о "не подлежащей доставке почте" по адресу отправителя. Это ответственность клиента.

*ПРИМЕЧАНИЕ, если Постфикс отклонил сообщение, основанное на SMTP ограничениях конверта, Постфикс, не должен возвращать его, потому что Постфикс, зарезервировал клиента. Это помогает экономить ресурсы системы,*

1 SMTP команда **pipelining** исключение из этого правила

*трафик, и может быть особенно удобно, если бы Постфикс, находится под сильным нападением spam`a, которое требовало бы тысяч возвратов, если сообщения были первоначально приняты для дальнейшей транспортировки.*

Вы можете изучить ограничения, которые имеет Постфикс и то, как они работают в Главе 7. Глава 8 содержит несколько примеров, которые Вы можете использовать в своей конфигурации.

## Контроль содержимого сообщения

Сообщение электронной почты состоит из заголовка и тела. Тело может также содержать одно или более вложений в виде файла или другого сообщения, открытого в пределах главного сообщения. Рисунок 6-2 показывает представление высокого уровня простого сообщения с вложением.


Рис 6-2 Письмо с вложением

Рисунок 6-3 показывает сообщение с другим письмом в качестве вложения.


Рис 6-2 сообщение с другим письмом в качестве вложения

Вы можете идентифицировать эти части, просмотрев сообщение в обычном текстовом редакторе. Например, вот - сообщение с вложенным файлом:

```
Return-Path:  <sender@example.com>                1
X-Original-To: recipient@example.com
Delivered-To:  recipient@example.com
Received:  by mail.example.com (Postfix)
        id 9F71443F50; Mon,  26 Apr 2004 01:32:59 +0200 (CEST)
Delivered-To:  recipient@example.com
Received:  by mail.example.com (Postfix,  from userid 500)
        id 2F23043F4F; Mon,  26 Apr 2004 01:32:59 +0200 (CEST)
Date: Mon,  26 Apr 2004 01:32:58 +0200
From:  Sender <sender@example.com>
To:  Recipient <recipient@example.com>
 Subject:  Elements of email content
Message-ID:  <2004O425233258.GA22383@mail.example.com>
Mime-Version:  1.0
Content-Type: multipart/mixed;  boundary="/9DWx/yDrRhgMJTb
 Content-Disposition:  inline
User-Agent:  Mutt/1.5.4i

--/9DWx/yDrRhgMjTb         2
Content-Type:  text/plain;  charset=us-ascii
Content-Disposition:  inline
```

Пустые линии отделяют тело сообщения от заголовков. MIME-кодированный текст  и MIME-кодированное вложение могут появлятся в теле письма.

Вы можете присоединять несколько файлов включая другое сообщение. Другое сообщение содержит свое собственное тело и заголовки поэтому вы можете иметь вложенные заголовки.

Надеюсь это поможет,

отправитель

```
--/9DWx/yDrRhgMJTb         3
Content-Type: application/x-zip-compressed
Content-Disposition: attachment; filename="attachment.zip"
Content-Transfer-Encoding: base64

UEsDBAoAAAAAAIILmjBOMxluCwAAAAsAMAOAAAAYXROYWNobWudC50eHRhdHRh
Y2htZW50 CIB LAOIUAAoAAAAAAIILmj BOMxluCwAAAAsAC2g0AAAABhdH RhY2 ht
ZW50L nR4dF L BOYAAAAAAOABADwAAAA3AAAAAAA=

--/9DWx/yDrRhgMJTb--
```

Части электронной почты следующие:

1 Заголовки электронной почты
2 Начало тела письма
3 Начало вложения

Постфикс может выполнить, проверку каждой из этих частей (header_checks, body_cheks, mime_header_checks) по отдельности. Чтобы выполнять проверку эффективно, Вы должны знать требуемые, рекомендованные, и дополнительные

части, которые сообщение может содержать.

### *Заголовки*

Заголовок  несет метеданные о теле сообщения, например, типа кодировки и даты передачи. RFC 2822 (ftp://ftp.rfc-editor.org/in-notes/rfc2822.txt разделяет элементы заголовка на требуемые и рекомендованные.

*ОБРАТИТЕ ВНИМАНИЕ, что поля заголовка не обязаны следовать в некотором специфическом порядке. Тем не менее рекомендуется, чтобы, они следовали в следующем порядке: Return-Path, Received, Date, From, Subject, Sender, To, Cc, и так далее. Вы можете наити дополнительную информацию о заголовках электроной почты здесь: http://www.stopspam.org/email/headeis.html*

## Требуемые Заголовки

**Date**

Поле Date обычно определяет дату и время, когда сообщение было составлено и послано. Если клиент отправителя опускает этот заголовок, Постфикс, добавляет его.

**From**

Это поле содержит данные о том , кто послал это сообщение. Если клиент отправителя опускает этот удар заголовок, Постфикс, добавляет его.

**Рекомендованные Заголовки**

Вот рекомендованные элементы заголовка:

**Message-Id**

Эта поле содержит уникальный идентификатор, который относится к текущей версии сообщения. Клиент генерирует ID и гарантирует его уникальность. Кроме того, сообщение ID предназначено, для машины, и оно может не обязательно означать что нибудь для людей. Поскольку сообщение ID соответствует только специфическому сообщению, любые последующие пересмотры сообщения должны получать новое ID.
Если клиент отправителя опускает этот заголовок, Постфикс, добавляет его.

**To**

Это поле определяет первичных получателей сообщения. Если клиент отправителя опускает этот заголовок, Постфикс, добавляет значение параметра undisclosed_recipients_header конфигурации.

**Subject**

Это поле должно содержать очень краткое описание сообщения.

**Cc**

Это поле определяет любых вторичных получателей сообщения.

**Reply-To**

Это поле указывает, куда клиент получателя должен послать ответы на сообщение.

**Content-type**

Это поле определено в RFC 1049 (ftp://ftp.rfc-editor.org/in-notes/rfc1049.txt), и это указывает структуру тела сообщения.

**MIME-Version**

Если это поле заголовка присутствует, тело сообщения было (возможно) составлено в согласии с RFC 1521 (ftp://ftp.rfc-editor.org/in-notes/rfc1521.txt

**Received**

Каждый транспортный агент, который сталкивается с сообщением, добавляет одну строчку в заголовок, чтобы указать, куда, когда, и как сообщение прибыло. Информация в этих полях может быть полезна при, прослеживании проблем с транспортировкой.

**Return-Path**

Этот удар головой  указывает отправителя конверта и используется, чтобы идентифицировать обратный путь к отправителю. Почтовый сервер вставляет данное поле при доставке от локального агента доставки, типа демона local.

**Дополнительные заголовки (X-заголовки)**

X-headers общее название для дополнительных полей заголовков  с именем, которое начинается с буквы X и дефиса. X-заголовки предназначены,  для нестандартной информации только, и наоборот, любая нестандартная информация  должна содержатся в X-заголовке.
Вот - несколько примеров X- заголовков (есть, конечно, миллионы других):

X-Mailer: Ximian Evolution 1.4.3

X-Priority:  3

X-Spam-Checker-Version: SpamAssassin 2.53 (l.174.2.15-2O03-O3-30-exp)

X-Original-To: recipient@example.com

## Тело

Тело несет собственно сообщение и должно идти после секции заголовков. Тело может быть в обычным текстом или кодированным. Тело может также содержать вложения кодированные в форме, которая не  искажается при транспортировке через Интернет (Раньше, много MTAs не отбрасывали восьмой бит; отбрасывание

восьмого бита от двойчного файла портит его).

## Вложения

Вложения - файлы, преобразованные текстовый вид (только печатные символы) подходящие для передачи, по электронной почте. Есть несколько частей в разделе приложения, они описаны в следующих подразделах.

### MIME Encodings

MIME сокращение от Многоцелевые Расширения Почты Интернета, это - система для преобразования формат сообщений, как описано в RFC 2045 (<http: //> www.rfc-editor.org/rfc/rfc2045.txt ). Два типа MIME шифрования для двоичных файлов обычно применяются - печатаные символы и base64:

#### печатаные символы

Кодирование печатными символами предназначено, для представления данных, который в значительной степени состоит из октетов, которые соответствуют печатаным символам в наборе АМЕРИКАНСКОГО ASCII. Оно кодирует данные таким способом, что результирующие октеты, вряд ли, будут изменены при передаче сообщения.

#### base64

Base64 – схема шифрования данных, определенная в RFC 1421 (<http://editor.org/in-notes/rfc1421.txt ) и RFC 2045 (ftp://ftp.rfc-editor.org/in-notes/rfc2045.txt ), для преобразования бинарных кодируемых данных в печатные символы ASCII. Это - по существу МИМЕ шифрование передаваемого содержимого в электронной почте Интернет, использующее только буквенно-цифровые символы (A-Z, a-z, цифры 0-9) и символы "+" и "/" символы, а также с символом "=" являющимся специальным суффиксом кода. Утилиты командной строки для ручного кодирования и расшифровки методом base64 включают mpack, munpack, и uudeview.
Все современные MUA поддерживают MIME, и вложения обычно посылаются только base64-кодируемыми.

### Кодирующий процессор

MUA выполняет задачу шифрования бинарного вложения, и это также автоматически создает структуру MIME, необходимую для включения в текст почты и кодируемое вложение в форме, понятой другим MIME MUAs. Эта форма требует следующих заголовков в сообщении:

### MIME-Version

Присутствие этого заголовка указывает, что сообщение MIME форматировано. Значение - обычно 1.0, так что заголовок обычно напоминает этот:

MIME-Version: 1.0

**Content-type**

Этот удар заголовок указывает тип и подтип содержания сообщения. Вот - пример:

Content-type: text/plain

Комбинацию типа (text, в этом примере) и подтипа (plain) обобшенно называют MIME типом, так что MIME тип в этом примере text/plain.
Большое количество файловых форматов имеют зарегистрированные MIME типы. IANA содержит архив, внося в список зарегистрированные типы (ftp://ftp.isi.edu/in-notes/iana/assignrnents/media-types ). Кроме того, все текстовые типы имеют дополнительный параметр charset, который указывает кодировку. Очень большое количество типов кодировок имеют зарегистрированные MIME charset имена.

**Content Types**

Этот раздел описывает некоторые из MIME типов с которыми Вы, вероятно, столкнетесь. Кроме того, millti-part-mime-message MIME тип позволяет сообщениям состоять из нескольких различных частей, имеющих древовидную структуру , где узловые листы имеют не многослойный тип, а не узловые листы - любое разнообразие многослойных типов. MIME механизм поддерживает следующие типы (среди других):

**text/plain**

Простые текстовые сообщения текста используют значение text/plain ; это - значение по умолчанию заголовка тип содержимого.

**multipart/mixed**

Этот тип обозначает текст плюс вложения (многослойный/смешанный с текстовыми / простыми частями и другими не текстовыми частями). MIME сообщение с вложенным файлом в общем случае указывает оригинальное имя файла с заголовком размещения содержимого, так что тип файла обозначен и типом содержимого и (обычно определяемым ОС) расширением имени файла.
Вирусы часто посылают себя как файлы, где тип содержимого и заголовок размещения содержимого указывают на различные типы файла.

**message/rfc822**

Это - ответ с вложенным оригинальным сообщением (многослойное/смешанное с простой/текстовой частью и с оригинальным сообщением как частью message/rfc822). Постфикс осуществляет возврат сообщений, таким образом (вложение типа message/rfc822 является первоначальным сообщением, которое было отброшено).

**multipart/alternative**

Этот тип указывает на содержимое с двумя альтернативными, методами просмотра, типа сообщения, посланного и в обычном тексте и в другой формате, типа HTML (то же самое содержимое в форматах text/plain и text/HTML). Outlook Express использует этот тип содержимого по умолчанию, потому что он посылает почту и HTML и

обычный текстовом формате одновременно.

**Структура кодированного сообщения**

MIME многослойное сообщение содержит границу, отмеченную как границу в почте, в заголовке типа содержимого, и эта граница, не должна встречаться ни в одной из частей. Вместо этого, она должна появиться между частями, и в начале и конце тела сообщения. Следующий пример иллюстрирует типичное многослойное сообщение:

Return-Path; <sender@example.com>

X-Original-To: recipient@example.com

Delivered-To: recipient@example.com

Received: by mail.example.com (Postfix)

id 9F71443F50; Mon, 26 Apr 2004 01:32:59 +0200 (CEST)

Delivered-To: root@example.com

Received: by mail.example.com (Postfix, from userid 500)

id 2F23043F4F; Mon, 26 Apr 2004 01:32:59 +0200 (CEST)

Date: Mon, 26 Apr 2004 01:32:58 +0200

From: Sender <sender@example.com>

To: Recipient <recipient@example.com>

Subject: Elements of email content

Message-ID: <2004O425233258.GA22383@mail.example.com>

Mime-Version: 1.0 **1**

Content-Type: multipart/mixed; boundary='79DWx/yDrRhgM3Tb" **2**

Content-Disposition: inline

User-Agent: Mutt/i.5-4i

--/9DWx/yDrRhgMJTb **3**

Content-Type: text/plain; charset=us-ascii **4**

Content-Disposition: inline

Пустая строка отделяет тело сообщения от ударов заголовков. MIME кодированный текст и MIME кодированные приложения могут появиться в теле.

Вы можете вложить один или более файлов, включая другое сообщение электронной почты.

Сообщение в пределах другого сообщения включает свои собственные заголовки и тело.

Поэтому, Вы, возможно, имеете вложенные заголовки.

Надеемся, это поможет.

--/9DWx/yDrRhgMJTb **5**

Content-Type: application/x-zip-compressed **6**

Content-Disposition: attachment; filename="attachnent.zip"

Content-Transfer-Encoding: base64 **7**

UEsDBAoAAAAAAIILmjBOMxlijCwAAAAsAAAAOAAAAYXROYWNobWVijdC50eHRhdH

RhY2htZW50C1BLAQIUAAoAAAAAAIILmBOMxluCwAAAAsAAAAOAAAAAAAAAAAEAI

AC2 gQAAAABhdHRhY2 ht ZW50L nR4dF L BOYAAAAAAOABADwAAAA3AAAAAAA=

--/9DWx/yDrRhgMJTb-- **8**

Части сообщения следующие:

1. Это заголовок версии MIME.
2. Это заголовок, содержащий тип содержимого, и граничную строку которая отделяет различные части сообщения.
3. Первое появление граничной строки. Новая часть многослойного сообщения начинается здесь.
4. Это обычная текстовая часть.
5. Это - второе появление граничной строки, которое указывает на то, что предыдущая часть закончилась, и новая часть многослойного сообщения начинается здесь.
6. Новая часть – файл в формате zip.
7. Zip файл закодирован в формате base-64.
8. Это - заключительное использование граничной строки, указывающее конец части и сообщения.

# Как работают ограничения на передачу сообщений.

Знать, что может быть ограничено, необходимо знать
чем "что" является и, как это должно быть.. .. -
Патрик, в попытке понимать Ральфа, в то время как он объяснял ограничения

Эта глава объясняет теорию ограничений. Ограничения позволяют вашему почтовому серверу принимать или отклонять поступающие сообщения, просматривая SMTP связь, которая имеет место между клиентом и сервером. Информация, полученная от этого диалога позволяет, Постфиксу, накладывать или снимать ограничения клиентов, отправителя и получателя.

Хотя слово "ограничение" обычно означает, что Вы ограничиваете что либо, термин "ограничение" может также подразумевать, полную противоположность в Постфиксе; Вы можете сформировать ограничения так, чтобы явно позволить что либо.

## «Спусковые механизмы» ограничений

Ограничения - мощный инструмент. Чтобы использовать их эффективно, Вы должны понимать SMTP связи и ее особенности, которые Постфикс, используют для анализа этой связи. Вы уже видели, как SMTP связь происходит в Главе 6. Мы будем рассматривать ее здесь под различной перспективой; на сей раз мы интересуемся стадиями SMTP связи которые определяются командами, отдаваемыми клиентом. Рисунок 7-1 выделяет эти стадии.

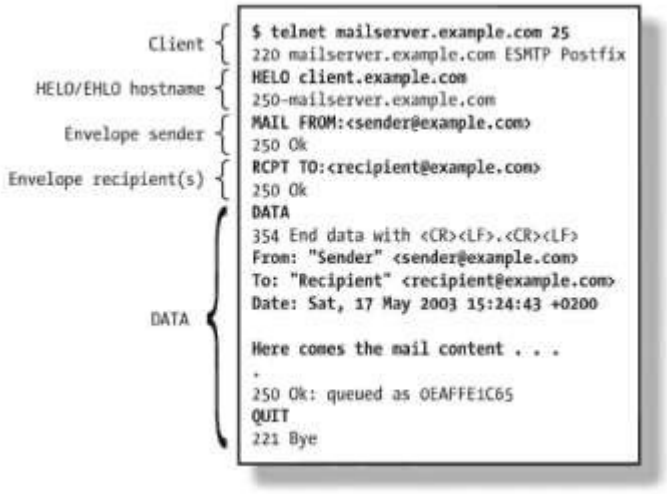



рис 7-1 стадии SMTP связи и типичный ввод клиента.

Каждая новая стадия на рисунке 7-1 помечает момент, когда smtpd демон Постфикса получает новую частицу информации о клиенте и сообщении, которое он хочет передать. Постфикс использует, эти стадии, для вызова ограничения, и каждая стадия имеет свои собственные параметры ограничения, названные по имени активного демона, названии стадии, и цели. Именно поэтому «спусковые механизмы» ограничений следуют этому шаблону: Smtpd_stagename_restrictions.

Вот - список всех механизмов ограничений и их поведение по умолчанию:

**smtpd_client _restrictions**

Эти ограничения относится к IP адресу клиента или его имени или обоим.
По умолчанию, Постфикс, позволяет соединятся любому клиенту.

**smtpd_helo_restrictions**

Эти ограничения относится к HELO/EHLO аргументу клиента, IP адресу, имени или обоим. По умолчанию разрешен любой HELO/EHLO аргумент.

**smtpd_sender _restrictions**

Это - первый набор ограничений, который относится к конверту. Постфикс применяет эти ограничения к отправителю конверта, HELO/EHLO аргументу, и клиенту. По умолчанию разрешено любому отправителю конверта посылать сообщения.

**smtpd_recipient_restrictions**

Эти ограничения относятся к получателю(ям) конверта, отправителю конверта, HELO/EHLO аргументу, и IP адресу клиента, имени хоста клиента или обоим. Установки Постфикса по умолчанию, должны разрешить отправлять письма любому получателю, клиентам, которые принадлежат сети указанной в параметре mynetworks файла конфигурации, для других клиентов разрешено получение писем для адресатов только домена указанного в параметре relay_domains или адресатам доменов указанных в mydomains. Это защищает, Постфикс от использования его в качестве открытого релея.

**smtpd_data_restrictions**

Это ограничения обнаруживает клиентов, которые посылают содержание почты прежде, чем Постфикс, ответил на команду DATA. Постфикс делает это, отслеживая команду DATA, когда клиент посылает команду серверу. Нет никакого ограничения по умолчанию.

**smtpd_etrn_restrictions**

Этот специальное ограничение, ограничивает клиентов, которые могут просить Постфикс, очистить очередь почты. По умолчанию любому клиенту разрешено выполнять команду ETRN.

Каждое ограничение соответствует набору ограничений; Вы можете думать о ограничениях как о пустых коробках. Чтобы использовать любое из них, Вы должны поместить правила (ограничения) внутри.

## Типы ограничений

Постфикс имеет несколько видов ограничений, которые могут быть разделены на четыре различных группы:

- Общие ограничения
- Переключаемые ограничения
- Настраиваемые ограничения
- Дополнительный параметры контроля UCE (нежелательной почты)

## Общие ограничения

Первая группа ограничений ничего не проверяет в SMTP диалоге; они просто выполняют команду:

**permit**

Позволяет запрос.

**defer**

Отсрочивает (задерживает) запрос.

**reject**

Отклоняет запрос.

**warn_if_reject**

Помогает настройку  более поздних ограничений; если ограничение после warn_if_reject, решает отклонить запрос, Постфикс, фактически не отклоняет сообщение, а скорее, печатает сообщение reject_warning в лог файле.

**reject_unauth_pipelining**

Отклоняет запрос, когда клиент посылает команды SMTP раньше срока, не зная, что Постфикс фактически поддерживает конвейерную обработку команд ESMTP. Это останавливает программное обеспечение, которое ненадлежащим образом использует конвейерную обработку команды ESMTP  для ускорения доставки.

## Переключаемые ограничения

Второй вид ограничений работает подобно выключателям. Вы включаете их или выключаете, и когда они активизированы, они отслеживают, было ли некоторое условие выполнено. Вот - неполный список:

**smtpd_helo_required**

Это ограничение требует, чтобы клиенты послали HELO (или EHLO) команда в начале SMTP сессии. Оба RFC821 и RFC 2821 требуют HELO/EHLO.

**strict_rfc821_envelopes**

Это ограничение регулирует устойчивость Постфикса терпимость к ошибкам в адресах, сделанных в  командах MAIL FROM или RCPT TO  . К сожалению, широко используемая  программа Sendmail разрешает весьма маленькое нестандартное поведение, и в результате, есть много программного обеспечения, которое ожидает

не встретить неприятностей здесь. Данное требование отклоняет некоторую нежелательную почту, но также может блокировать законную почту от плохо написанных клиентов.

**disable_vrfy _command**

SMTP VRFY команда позволяет клиентам проверять, что получатель существует. Это ограничение позволяет Вам отключить команду VRFY.

**allow_percent_hack**

Это ограничение контролирует преобразование из формы user%domain в user@domain.

**swap_bangpath**

Это ограничение контролирует преобразование из site!user в user@site. Это необходимо, если ваша машина связана с UUCP сетью.

## Настраиваемые ограничения

Настраиваемые ограничения - карты работающие подобно фильтрам. В каждой строке карты, ключ - фильтр, а значение - действие, которое выполняется если есть совпадение с фильтром (обратитесь к разделу "Общие ограничения" для списка возможных действий). Вот - несколько видов настраиваемых ограничений:

**HELO (EHLO) hostname restrictions**

Эти ограничения ограничивают имена, которые клиенты могут послать с командой EHLO или HELO.

**Client hostname /address restrictions**

Ограничивают клиентов, которые могут устанавливать SMTP соединение с почтовым сервером.

**Sender address restrictions**

Ограничивают адреса отправителей (отправители конверта), которые Постфикс, принимает в команде MAIL FROM.

**Recipient address restrictions**

Эти ограничения ограничивают адреса получателя (получатели конверта), которые Постфикс, принимает в команде RCPT TO.

**ETRN command restrictions**

Ограничивает клиентов, которые могут выполнять команду ETRN.

**Header filtering**

Это фильтрование ограничивает то, что позволяется в заголовках сообщения. Образцы применены к полным логическим заголовкам сообщения, даже когда логические заголовки занимают несколько строк.

**Body filtering**

Это фильтрование ограничивает текст, который может появиться в теле сообщения.

**DNSBL-style blacklists**

Эти черные списки ограничивают связи от IP адресов, которые появляются в DNSBl, черных списках.

**RHSBL style blacklists**

Эти черные списки отвергают домены отправителя (как часть отправителя конверта), которые появляются в черных списках

## Дополнительные параметры контроля UCE

Набор дополнительных параметров контроля UCE поддерживает другие ограничения или особенности, которые - не являются частью функциональных возможностей Постфикса по умолчанию. Вот - только несколько из доступных ограничений:

**default_rbl_reply**

Создает шаблон ответа, который используется, когда SMTP запрос клиента блокирован ограничением reject_rbl_client или reject_rhsbl_sender .

**permit_mx_backup_networks**

Ограничивает использование permit_mx_backup параметра контролиюлищего реле к предназначению, первичные MX хосты которого соответствуют списку блоков в сети.

**rbl_reply_maps**

Определяет таблицы поиска с шаблонами ответа DNSBL, индексированными по имени DNSBL домена . Если никакой шаблон не найден, Постфикс использует **default_rbl_reply** шаблон вместо этого.

**relay_domains**

Инструктирует Постфикс принимать почту для этих доменов, даже при том, что этот сервер не является заключительным предназначением.

**smtpd_sender_login_maps**

Определяет пользователя, которому позволяют использовать определенный MAIL FROM адрес (отправитель конверта). Чтобы использовать это ограничение,

Постфикс должен знать имя пользователя, так что клиент должен идентифицировать себя с SMTP установлением подлинности.

## Диапазон Применений

Ключ к правильному использованию ограничений понимание, к какой стадии связи Вы можете применить их. Некоторые ограничения не имеют смысл в некоторых стадиях. Таблица 7-1 показывает соответствие ограничений и стадий.

Таблица 7-1 Диапазон применений.

| Стадия | Ограничения |
|---|---|
| Client (IP address and/or hostname) | check_c1ient_access<br>reject_rbl_client<br>reject_rhsbl_client<br>reject _unknown_c1ient |
| HELO/EHLO hostname | check_helo_access<br>permit_naked_ip_address<br>reject_invalid_hostname<br>re j e ct _non_fqdn_hostname<br>reject_unknown_ hostname |
| Envelope sender | check_serider_access<br>reject _non_fqdn_sender<br>reject _rhsbl_sender<br>reject _unknown_ sender_domain<br>reject _unverified_sender |
| Envelope recipient | check_recipient_access<br>permit _auth_destiriation permit_mx_backup<br>reject_nonfqdn_recipient<br>reject_unauth_destination<br>reject_unknown_ recipient_domain<br>reject _unverified_recipient |
| DATA | reject_unauth_pipelining |

## Построение ограничений

Ограничения могут стать весьма сложными, и Вы можете сломать ваш почтовый сервер на этом тонком (и не настолько легком) пути, пробуя настроить их, не зная, что Вы делаете. Держите следующие правила в памяти при построении ваших ограничений:

- Неправильное написание сделает ваше ограничение бесполезным.
- Появляются различия в более поздних версиях
- Порядок написания ограничений важен. Предыдущие действия влияют, на прохождение дальнейших ограничений.

Как мы упоминали ранее, ограничения походят на пустые коробки. Однако, заполнение их не подразумевает, что Вы только добавляете ограничения.

*restriction_trigger = conditional_restriction, customizable_restriction *

*maptype:/path/to/the/map, general_restriction*

Поскольку отдельное ограничение может легко превысить разумную ширину линии, Вы можете добавить пробелы в началу каждой строки, которая продолжает предыдущую так, чтобы Постфикс, считал строки как один параметр.
Кроме того, запятые, отделяющие предыдущие ограничения являются опциональными. Поэтому, следующая запись эквивалентна предыдущему примеру (и гораздо легче читаемая).

restriction trigger =

*conditional_* re*striction*

*customizable_restriction maptype:/path/to/the/map*

*general_restriction*

## Мгновенная оценка

Вообще, Постфиск, не оценивает и выполняет ограничения, немедленно после того, как выполнилась некая стадия SMTP связи. Вместо этого, Постфикс, ждет, пока клиент не пошлет сообщение первому получателю конверта. Эта задержка существует, потому что некоторые клиенты почты продолжают пробовать передать их сообщение, если сервер отклоняет команду прежде, чем они закончили посылать сообщение по крайней мере одному получателю конверта.

Вы можете переопределить это значение по умолчанию, установив smtpd_delay_reject параметр как no.

Однако,даже при том, что второй возможно отследить этих клиентов и построить список исключения, чтобы гарантировать, что они не будут прерваны, лучше , подождать, пока все стадии не будут закончены и установленные ограничения вступят в силу после этого. Тем самым, Вы не только уменьшите сложность вашей почтовой системы, но Вы также соберете больше данных о попытке поставки почты.

Чтобы понять идею того, как smtpd_delay_reject влияет на оценку ограничений, взгляните на Рисунок 7-2.

## Влияние действий на оценку ограничений

Как описано в разделе "Настраиваемые Ограничения, " настраиваемые ограничения используют карты. Когда Постфикс, ищет ключ в карте ограничения, Постфикс, выполняет действие, которое соответствует этому ключу. Карта может напоминать эту:

| | |
|---|---|
| 10.0.0.1 | PERMIT Private IP from VPN transfer tunnel |
| 172.16.0 | REJECT Private IP address cannot come from outside |
| 168.100.1.3 | DUNNO |
| 192.0.34.166 | OK |


Рис 7-2 Влияние параметра smtpd_delay_reject на ограничения.

Предыдущая карта содержит четыре различных действия для настраиваемых ограничений: PERMIT, REJECT, DUNNO, и OK. Эти значения говорят, Постфиксу, что делать с клиентом, отправителем, или получателем. Хотя есть несколько действий (см. man access(5) ), эти значения наиболее широко используемые:

**OK**

Нет никаких возражений против клиента и сообщения. Постфикс останавливает, оценку ограничений в текущем наборе ограничений и перемещается в следующий.

**PERMIT**

Эквивалентно OK.

**REJECT**

Отклонить сообщение немедленно, игнорируя любые дальнейшие ограничения. Сообщение в конечном счете будет отклонено.

**DUNNO**

Прекратить оценивать текущее ограничение и перейти к следующему ограничению в текущем наборе ограничений.

Порядок ограничений в пределах набора важен, в случае возврата OK или REJECT, немедленно останавливается, оценка ограничений в текущем наборе (при значении REJECT, клиент, отправитель, или получатель в конечном счете будут отклонены), Постфикс, читает и применяет ограничения сверху донизу, или слева направо если Вы написали их в одной строке. Это - то, почему более легко использовать много строчную запись для сложных ограничений. Вообразите каково читать эти ограничения, если бы все было записано в одной строке.

```
smtpd_recipient_restrictions =
        check_recipient_access hash:/etc/postfix/recipients_restrictions,
        permit_sasl_authenticated,
        prermit_mynetwors,
        reject_unauth_destination,
        reject_unauth_pipelining,
        reject_rbl_client relays.ordb.org
        permit
```

Рисунок 7-3 иллюстрирует процесс оценки ограничения и показывает действие каждой из четырех значений.

## Замедление Плохих Клиентов

Любой клиент, который совершает множество ошибок, при связи smtpd (например, вызывая действие REJECT в ограничении или совершая синтактическую ошибку в аргументах) заставляет демон smtpd делать короткую паузу перед принятием дальнейших команд в этой сессии. Это служит защитой против не контролируемого программного обеспечения клиента.
Вы можете настроить это с помощью нескольких параметров. Параметр smtpd_error_sleep_time определяет число секунд, паузы после каждой ошибке (по умолчанию - одна секунда). Параметр smtpd_soft_error_limit служит своего рода ограничительным механизмом; когда удаленный SMTP клиент делает несколько ошибок, Постфикс SMTP сервер может вставить дополнительные задержки перед ответом. Наконец, Вы можете прервать сессию, основываясь на параметре smtpd_hard_error_limit.
Эти три параметра работают вместе следующим образом:

- Если клиент совершает, ошибки и общее количество ошибок в потоке SMTP сессии - ниже значения smtpd_soft_error_limit, каждая ошибка вызывает задержку smtpd_error_sleep_time.
- Если клиент совершает, ошибки и общее количество ошибок в SMTP сессии превышают значение smtpd_soft_error_limit, каждая ошибка вызывает увеличивает задержку на smtpd_soft_error_limit секунд.
- если число ошибок клиента превышает, значение smtpd_hard_error_limit, Постфикс, завершает сессию.

Например, скажем, то, что Вы установили следующие параметры:

```
smtpd_soft_error_limit = 5
sьtpd_hard_error_limit = 10
smtpd_error_sleep_tiьe = 1s
```

Если клиент совершит 11 ошибок в одной сессии, Постфикс сделает паузы равными 1,1, 1, 1, 1, 2, 3, 4, 5, и 6 секундам, соответственно, и на 11-ую ошибку, разъединит сессию.

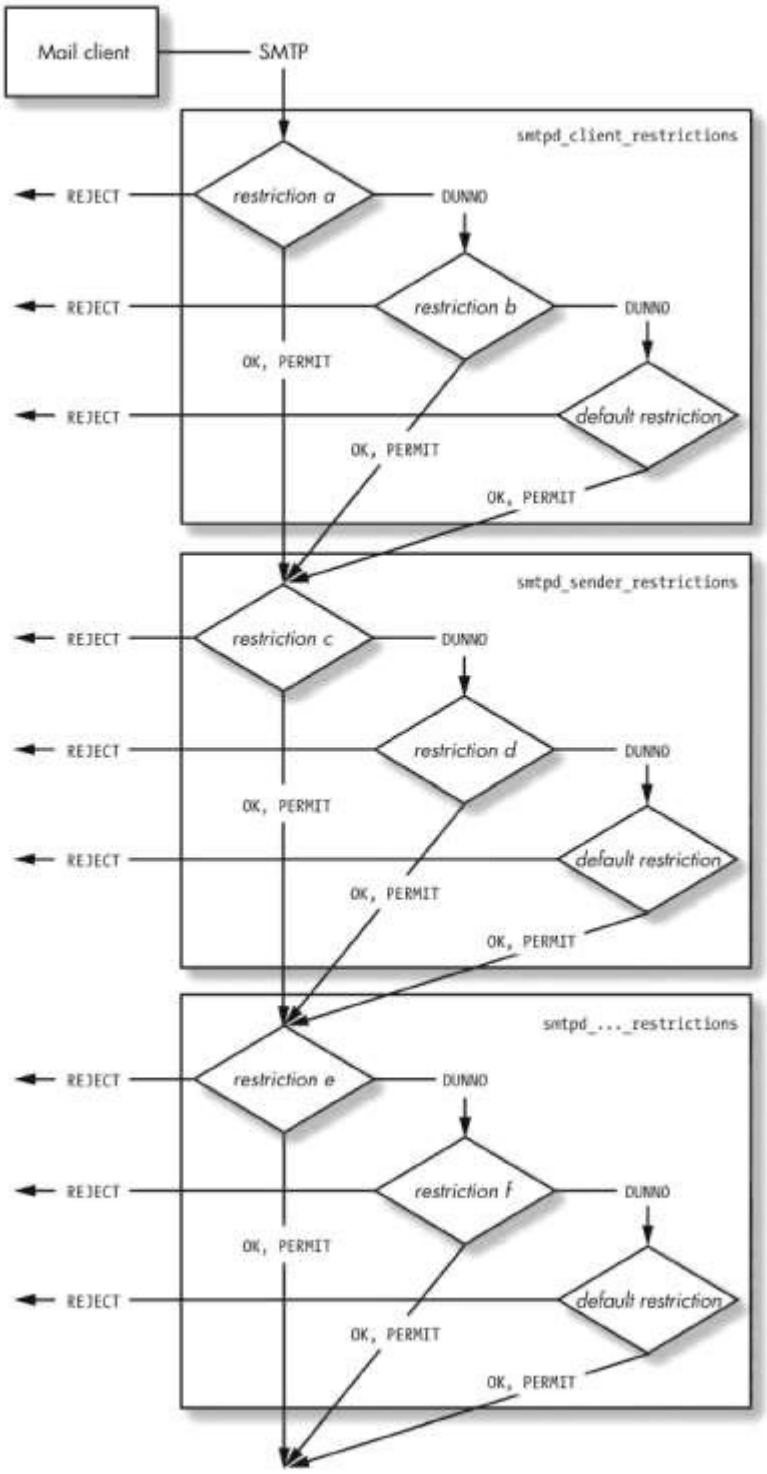



Рис 7-3 процесс оценки ограничений.

## Классы ограничений

Класс ограничения - специальная форма ограничения, котороя не предопределен или связана ни с какой стадией SMTP связи. Вы определяете их, поскольку Вы нуждаетесь в них, и Вы вызываете их, обращаясь к ним в картах настраиваемых ограничений.
Например,  Вы имеете карту настраиваемых ограничений для того, чтобы проверить адреса отправителя конверта, и Вы хотите вызвать другой набор ограничений, если

отправитель конверта соответствует example.com. В этом случае, Вы хотите поместить этот новый набор ограничений в новом классе, названном check_if_example.com_sender. Сначала, объявите новый класс в вашем файле main.cf.

smtpd_restriction_classes = check_if_example.com_sender

Теперь, также в main.cf, добавьте некоторые ограничения к вашему новому классу:

check_if_example.com_sender =

check_sender_access hash:/etc/postfix/bounces

check_sender_access hash:/etc/postfix/valid_example.com_senders

check_sender_access regexp:/etc/postfix/nice_reject.regexp

Как Вы можете видеть, эти новые ограничения исследуют отправителя конверта (хотя они могли быть чем - нибудь соответствующим текущей стадии SMTP связи).
Не волнуйтесь о картах в этих ограничениях. Вы увидите, как определить их позже.
Мы все еще пропускаем важную часть. Как Вы можете активизировать check_if_example.com_sender?
Чтобы сделать это, Вы нуждаетесь в check_sender_access ограничении в вашем наборе srntpd_*_restrictions. Скажем, то, что Вы уже имеете этот набор в следующей карте, которая принимает отправителей от foo.com и отклоняет от bar.org (см. раздел "Влияние действий на оценку ограничений" ранее в этой главе для соответствующих действий):

| | |
|---|---|
| foo.con | OK |
| bar.org | REJECT |

Чтобы добавить новый класс ограничений, дополните карту следующим образом:

| | |
|---|---|
| foo.com | OK |
| bar.org | REJECT |
| example.com | check_if_example.com_sender |

Как Вы можете видеть, ключ к использованию классов ограничения находит, и вставляет их в карту настраиваемого ограничения.

# ИСПОЛЬЗОВАНИЕ ОГРАНИЧЕНИЙ НА ПЕРЕДАЧУ СООБЩЕНИЯ

Ограничения управляют потоком сообщения, принимая решения основанные на том, что клиент передает в течение SMTP связи. Число ситуаций, в которых ограничения могут использоваться, по-видимому неизмеримо, так вместо того, чтобы внести в список все ограничения и все возможные доступные варианты для этих ограничений, эта глава описывает сценарии, которые часто обсуждаются в списке рассылке пользователей Постфикса и ежедневно используются. Для каждого сценария, мы обсудим ограничения и опции так глубоко, чтобы показать Вам, как осуществить их применение и помочь Вам понять, почему они применяются так как описано.

## Как строить и проверять ограничения

Прежде, чем Вы начнете изменять ограничения по умолчанию, Вы должны знать точно, что Вы хотите ограничить. Это не очень трудно, когда Вы изменяете Булевые ограничения на on или off, но может стать более сложным, если Вы хотите ограничить электронную почту от хостов, которые пытаются маскировать их происхождение.
Обычная поговорка списков рассылки Постфикса - " Лог - ваш друг. " Может быть трудно вообразить почтовый лог в качестве друга, но лог файлы действительно очень удобны при сборе информации для того, чтобы ограничить поток электронной почты. Проще говоря, в почтовых логах содержится, большинство информации необходимой для построения эффективных ограничений. Посмотрите на этот ряд строк в лог файле для входящего сообщения:

Apr 14 21:14:48 mail postfix/smtpd[3l840]: 4F2A643F3O:

client=unknown[l72.16.0.l] ***1***

Apr 14 21:14:48 mail postfix/cleanup[31842]: 4F2A643F30:

message-id=<OO2l0lc42254S792c253OS0i00iOac@stateofmind.de> ***2***

Apr 14 21:14:48 mail post-fix/nqmgr[31836]: 4F2A643F30:

from=<test@exanple.com>, ***3***

size=666, nrcpt=l ***4*** (queue active)

Apr 14 21:14:48 nail postfix/smtpd[3l840]: disconnect from unknown[l72.l6.0.l]

Apr 14 21:14:48 mail postfix/smtp[3i844]: 4F2A643F30: to=<p@state-of-mind.de>, ***5***

relay=mail.state-of-mind.de[2l2.14.92.89], ***6***

delay=0, status=sent (250 Ok: queued as 97E70E1C65) ***7***

Части сообщения следующие:

- IP адрес клиента ( и hostname), который доставил сообщение
- Заголовок Message-Id
- Отправитель конверта ( MAIL FROM команда в SMTP связи)
- Число получателей

- Получатель(и) конверта (RCPT TO команда в SMTP связи)
- Куда сообщение пошло
- ID очереди, которое удаленный Постфикс сервер, назначил сообщению

Если ваша работа состоит в том, чтобы ограничить транспортировку сообщения, и Вы нуждаетесь в еще большом количестве информации, чтобы выяснить то, с чем Вы имеете дело, лог файл – то место, где Вы можете узнать вашего "противника",

## Моделирование воздействия ограничений

Хороший набор ограничений редко получается с первой попытки. Чтобы получить то, в чем Вы нуждаетесь, Вы должны провести несколько испытаний для отлавливания ошибок. Чтобы проверять ваши ограничения, Вы будете нуждаться в сообщениях, против которых применяются ваши ограничения , и возможно , что Вы не имеете их на испытательной машине и должны развить ваши ограничения на рабочем сервере. К сожалению, это представляет риск наличия ложного срабатывания и потери важной электронной почты.
Чтобы решить эту проблему, Постфикс, имеет параметр warn if reject parameter для проверки ограничений, который подобен действию WARN в проверках. Поставьте этот параметр перед ограничением, которое Вы хотите проверить, Постфикс только регистрирует, эффект ограничения в лог файле, но не отклоняет почту. Например Вы могли бы использовать это, чтобы проверить reject_unknown_sender_domain.

```
smtpd_recipient_restrictions = permit_mynetworks
                               reject_unauth_destination
                               warn_if_reject reject_unknown_sender_domain
                               permit
```

Как только Вы установите этот параметрах в лог файле зарегистрируется "моделируемое" отклонение подобно этому:

Jun 25 16:10:52 mail postfix/smtpd[325n]: 8O75015CO2F: rejectwarning: RCPT from sccrmhcll.comcast.net[204.127.202.55]: 550 <DickinsL@newfaces.gr>: Sender address rejected: Domain not found; from=<DickinsL@nevrfaces.gr> to=<example@charite.de> proto=ESMTP helo=<sccrmhcll.attbi.com>

После того, как Вы уверены, что ограничения работают, вы можите удалить warn_if_reject параметр из вашего ограничения. Дальнейшие записи в лог файле сообщат Вам, что ограничение было применено успешно, регистрируя отклоненные сообщения:

Jun 25 16:11:23 mail postfix/smtpd[3251l]: 8O75015CO2F: reject: RCPT from sccrmhcll.comcast.net[204.127.202.55]: 550 <DickinsL@newfaces.gr>: Sender address rejected: Domain not found; from=<DickinsL@nevrfaces.gr> to=<recipient@example.com> proto=ESMTP helo=<sccrmhcll.attbi.com>

## Создание ограничений, применяемых немедленно

Постфикс состоит из нескольких различных демонов, которые загружают данные своей конфигурации во время запуска. Некоторые из демонов, выполняются только в течение короткого времени и заканчиваются, чтобы избежать дополнительного использования ресурсов. Однако, другие демоны не перезапучкаются, если Вы не скажите, Постфиксу, сделать это.

Эти демоны длительного действия, qmgr и nqmgr (демон nqmgr только в старых версиях, новые версии Постфикс используют новый менеджер очереди по умолчанию, в этом случает он носит имя qmgr в случае использования старого менеджера очереди - oqmgr ), играющие важную роль в ограничении потока электронной почты, не будут замечать изменения в конфигурации до тех пор пока они не будут перезапущены системой, или Вы не вмешиваетесь вручную. Поэтому, Вы должны помнить, что всякий раз, когда Вы вносите изменения в main.cf, или master.cf, Вы должны выполнить команду **postfix reload**, чтобы заставить менеджера очереди перезагрузить конфигурацию.

*ОБРАТИТЕ ВНИМАНИЕ Теоретически, изменения будут применены через какое-то время, потому что демоны умрут и будут перерождены после использования некоторого числа раз, определяемого параметром max_use. Кроме, конечно, qmgr, который никогда не умирает. Изменения для qmgr всегда требуют выполнения команды postfix reload. Применение изменений через какое-то время, однако, может привести к тому, что некоторые демоны будут использовать старую конфигурацию, в то время как другие используют новую конфигурацию, которая может быть не идеальна.*

## Ограничения по умолчанию

Постфикс идет с безопасным набором ограничений по умолчанию, которые препятствуют вашей машине становиться открытым релеем (или релеем почты третьего лица). Вы можете узнать какие ограничения по умолчанию используются выполнив postconf, чтобы вывести установки по умолчанию для smtpd_recipient ограничений:

**# postconf -d smtpd_recipient_restrictions**

smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination

Постфикс оценивает ограничения в порядке, в котором они внесены в список. В этом случае, если клиент хочет переслать, сообщение, Постфикс проверяет, исходит ли связь от хоста в пределах mynetworks. Если это так (если оценка возвращает permit_mynetworks OK), Постфикс, принимает сообщение для доставки.

Если клиент прибывает не из mynetworks, Посту оценивает reject_unauth_destination. Это правило, ограничивает попытки пересылки почты проверяя, находится ли получатель сообщения в заключительном предназначении и домене релея, которые Вы сконфигурировали для Постфикса. Если получатель - не в пределах этих доменов, reject_unauth_destination возвращает REJECT, и Постфикс, сообщает клиенту, что он не может переслать сообщение.
Если Постфикс, несет ответственность за место предназначение сообщения, reject_unauth_destination возвращает OK, и Постфикс, оценивает следующее ограничение. Однако, так как ограничений в списке больше нет, Постфикс ,

предполагает, что по умолчанию подразумевается premit , и принимает сообщение.

Эти два ограничения - основы, которые защищают ваш сервер от того, чтобы быть открытым релеем, но они не защищают ваших пользователей от spam`a, и при этом они не говорят клиентам, соединяющимся с вашим сервером вести себя должным образом. Остальная часть этой главы покажет Вам, как сделать ограничения более жесткими.

**Требование соответствию RCF**

Требование надлежащего поведения (соответствие RFC) от локальных и удаленных клиентов - первый шаг большого плаванья. Мало того, что это гарантирует, что ваш почтовый сервер распространяет правильные сообщения другим почтовым серверам, но это также требует, чтобы удаленные клиенты вели себя должным образом. Это может быть полезно в защите против спамеров, которые всегда спешат, нарушают правила, и маскируют, идентификационную информацию.

Этот раздел покажет Вам, как наложить ограничения на имя хоста, отправителя конверта, и получателя конверта, чтобы достигнуть RFC соответствия.

*ОБРАТИТЕ ВНИМАНИЕ Ограничения, которых Вы здесь увидите, не будут использоваться в main.cf в порядке, их объяснения. Это намеренно, и Вы увидите почему в разделе "Порядок обработки RFC Ограничений " позже в этой главе. Просто добавьте ограничения, в порядке в котором они появляются в примере.*

## Ограничение Hostname в HELO/EHLO

Хорошее место, для начала, заставить Постфикс настаивать, чтобы клиенты представились должным образом, когда они хотят послать сообщения или через ваш сервер. Есть множество ограничений, которые Вы можете наложить на HELO/ELHO часть SMTP диалога, от простого требования клиентов посылать имя хоста, до требования, чтобы они послали действительное имя хоста.

## Требование Hostname

Постфикс параметр smtpd_helo_required требует, чтобы все клиенты выпустили любой HELO или EHLO при старте SMTP связи. Оба RFC 821 (ftp://ftp.rfc-editor.org/in-notes/rfc821.txt ) и RFC 2821 (ftp://ftp.rfc-editor.org/in-notes/rfc282l.txt ) обязывают делать это, но Постфикс, устанавливает параметр no по умолчанию. Задействуйте это, добавив следующую строку в ваш main.cf файл:

smtpd_ helo_required = yes

После того, как конфигурация будет перезагружена, Постфикс, откажется от сообщений любого клиента, который не представляется должным образом. Вы можете проверить это, соединяясь с вашим сервером и пробуя начать передачу сообщения без команды HELO . Вот как Постфикс, должен взаимодействовать при требовании hostname:

$ telnet mail.example.com 25

220 mail.example.com ESMTP Postfix

MAIL FROM: <sender@example.com>

503 Error:  send HELO/EHLO first QUIT

221  Bye

## Требование FQDN

Требование команды HELO/EHLO это хорошо, но клиенты также обязаны представлять их полное имя хоста наряду с рукопажатием (например, HELO client.example.com). Кроме того, RFC, требуют чтобы имя хоста, было полностью определенным доменным именем (FQDN).

*ОБРАТИТЕ ВНИМАНИЕ, что FQDN не обязательно существует в  записи ДНС службы.*

Постфикс откажется от сообщений  любого клиента, который не представляет FQDN имя, если Вы установите параметр reject_non_fqdn_hostname внутри smtpd_recipient_restrictions.

*ПРЕДОСТЕРИЖЕНИЕ Будьте осторожны с этим ограничением. Некоторые клиенты почты, типа Microsoft Outlook,, используют только локальную часть имени (например, client) по умолчанию, если Вы не сконфигурируете операционную систему предоставлять FQDN имя для приложений.*

Когда Вы добавляете reject_non_fqdn_hostname к вашему списку smtpd_recipient ограничений в main.cf файл, это должно выглядить примерно так:

smtpd_recipient_restrict ions =

permit_mynetworks

reject_unauth_destination

**reject_non_fqdn_hostname**

permit

Проверьте ограничение, соединяясь с вашим почтовым сервером и давая простое имя хоста, как в этом примере:
**$ telnet nail.exaniple.com 25**
220  mail.example.com ESMTP Postfix

**HELO client**

250 mail.example.com

**MAIL FROM:  <sender@example.com>**

250 Ok

**RCPT TO:  <recipient@example.com>**

504 <client>: Helo command rejected: need fully-qualified hostname

**QUIT**

221  Bye

## Отклонение недействительных знаков в имени хоста

RFC говорят, что имя хоста, посылаемое с командой HELO/EHLO должно быть не только FQDN, но и знаки из которых построено имя хоста, должены также повиноваться требованиям системы доменных имен. Действительное имя домена

- должно содержать по крайней мере следующие элементы:
- сокращение доменного уровня (TLD), типа "com"
- имя домена, типа "example"
- точки отделяющий (.) от TLD и имени домена

Любое другое имя хоста, вряд ли, разрешится должным образом, делая взаимодействие между клиентом и сервером трудным, если не невозможный. Вы можете сказать, Постфиксу, не говорить с такими клиентами, используя reject_invalid_hostname опцию в smtpd_recipient_restrictions. Вот - пример того, куда Вы могли бы поместить ее:

smtpd_recipient_restrictions =

permit_mynetworks

reject_unauth_destination

reject_non_fqdn_hostname

**reject_ invalid_hostname**

permit

Как и прежде, испытаем это ограничение, соединяясь от удаленного хоста с вашим почтовым сервером и давая недействительное имя хоста. Клиент представляется как "." в следующем примере сессии:

**$ telnet mail.example.com 25**
220 mail.example.com ESMTP Postfix
**HELO .**
250-mail.example.com
**MAIL FROM:<sender@example.com>**
250 Ok
**RCPT TO:<recipient@example,com>**
501 <•>: Helo command rejected: Invalid name
**QUIT**
221 Bye

## Ограничение отправителя конверта

Отправитель конверта должен также содержать FQDN в доменной части, и конверт должен принадлежать существующему домену. Отправители конверта типа sender и *sender@example* не содержат FQDN доменной части. Пример полного конверта сервера - sender@example.com . Недействительные адреса могут причинить большой замешательство, потому что адрес отправителя сообщения выглядит, как будто оно пришло от сервера. Есть две вещи, которые могут идти не так, как надо:

- MTA, который должен возвратить, сообщение с неполным отправителем конверта возвратил бы его локальным пользователям. Сообщение не будет возвращено оригинальному отправителю.
- Постфикс, мог пробовать "установить" недействительный адрес, создавая еще худшую ситуацию. Поскольку Постфикс, знает, что имя отправителя конверта должно быть FQDN, он бы поручил демону trivial-rewrite преобразовать эти адреса электронной почты, добавляя $myorigin отправителю (результат sender@$myorigin) и $mydomain к sender@example (результат sender@example.$mydomain). Поэтому, отправитель конверта для сообщений, прибывающих от удаленного сервера был бы полностью неправилен.

Чтобы предотвратить это, добавьте опцию reject_non_fqdn_sender к smtpd_recipient ограничениям, как в этом примере:

smtpd_recipient_restrictions =

reject_non_fqdn_sender

permit_mynetworks

reject_unauth_destination

reject_non_fqdn_hostname

reject_invalid_hostname

permit

Проверьте это, соединившись от удаленной машины к вашему почтовому серверу и задавая неправильного отправителя конверта. Этот пример показывает, как ограничение будет работать, Постфикс, отклоняют сообщения от такого отправителя:

**$ telnet mail.example.com 25**

220 mail.example.com ESMTP Postfix

**HELO client.example.com**

250 mail.example.com

**MAIL FROM: <sender>**

250 Ok

**RCPT TO: <recipient@example.com>**

**504 <sender>: Sender address rejected: need fully-qualified address**

## Почта от несуществующих доменов

Ответственный почтовый сервер не принимает сообщения от доменов отправителя, которые не существуют, потому что это не может войти в контакт с отправителем в несуществующем домене, если доставка неудачна. Другие конфигурации вызвали бы двойное возвращение, как только MTA попробовал уведомить отправителя, и сообщение с несуществующем домене отправителя появилось в почтовом ящике postmaster`a.

*ОБРАТИТЕ ВНИМАНИЕ, что почтовые серверы должны иметь дело с несуществующими доменами, потому что пользователи иногда неправильно набирают их почтовые адреса при конфигурировании почтовых клиентов; spammer`ы также используют несуществующие домены, чтобы скрыть свое происхождение.*

Чтобы защитить получателей и postmaster`ов от двойных возвратов и плохо сформированных сообщений, добавьте опцию reject_unknown_sender_domain к вашее smtpd_ recipient_restrictions конфигураций. Например, Вы можете разместить ее следующим образом:

smtpd_recipient_restrictions =

reject_ unknovtn_ sender_ dotnain

permit_mynetworks

reject_unauth_destination

reject_non_fqdn_hostname

reject_invalid_ hostname

permit

Следующий пример показывает, как Вы могли бы проверить ограничение (Вы увидите код ошибки 450, которые Постфикс, посылает как ответ на команду MAIL TO):

**$ telnet mail.example.com 25**

220 mail.example.com ESMTP Postfix

**HELO client.example.com**

250 mail.example.com

MAIL FROM: <sender@domain.invalid>

250 Ok

**RCPT TO: <recipient@example.com>**

**450 <sender$domain,invalid>: Sender address rejected: Domain not found**

## Ограничение получателя конверта

Как заключительный шаг в том, чтобы вынуждать поступающие связи придерживаться RFCs, Вы можете отклонить любое сообщение, которое имеет несуществующего домена или пользователя в получателе конверта.

Почтовый сервер не должен принять никакого сообщения для домена, который не существует, потому что нет никакого способа доставить такое сообщение. Если сервер почты принимает сообщение и возвращает, его назад позже, пользователь может думать, что что-то не так с почтовым сервером, потому что он первоначально принял сообщение.

Настройте ваш сервер, чтобы он отклонял сообщения для несуществующих доменов

возвращая проблему назад клиенту или пользователю, где это произошло. Чтобы настроить это в Постфиксе, используйте опцию reject_unknown_recipient_domain в вашем наборе smtpd_recipient_restrictions, подобно этому:

```
smtpd_recipient_restrictions =
        reject_unknown_recipient_ domain
        permit_mynetworks
        reject_unauth_destiriation
        reject_non_fqdn_hostname
        reject_invalid_hostname
        permit
```

Как обычно, Вы можете проверить это, посылая несуществующий домен получателя в ручной связи с сервером. Вот - пример, где Постфикс, отклоняет сообщение, потому что invalid.domain - несуществующий домен:

```
$ telnet mail.example.com 25
220 mail.example.com ESMTP Postfix
HELO client.example.com
250 mail.example.com
MAIL FROM;  <sender@example.com>
250 Ok
RCPT TO: <recipient@domain.invalid>
450 <recipient@domain.invalid>: Recipient address rejected: Domain not found
```

## Почта неизвестным получателям

Вы можете сконфигурировать, Постфикс, чтобы доставить сообщения для неизвестного пользователя в вашем домене postmaster`у. На первый взгляд, это походит на хорошую идею, потому что postmaster может исследовать и вручную доставить эти сообщения всякий раз, когда возможно.

Хотя это теоретически составило бы превосходное обслуживание клиента, настраивая доставку несуществующим получателям вероятно закончится ДОС атакой на ваш почтовый сервер, как только или нападению по словарю или червя. В таком нападении, нападавший пытается доставлять сообщение существующим получателям, посылая сообщения к адресам, используя все возможные комбинации букв. Например, нападавший мог начать с aa@yourdomain.com, затем пробовать ab@yourdomain.com , и продолжаться через все комбинации с двумя буквами до достижения zz@yourdomain.com.

Мало того, что трудно отделить, существующие сообщения от беспорядка, созданного этим видом нападения, но сервер также подвергнут риску потребления слишком большого количества полосы пропускания, времени центрального процессора, памяти, и места на диске, пока ваш сервер наконец не рухнет и не прекратит обслуживать запросы передачи сообщения. Например, вирус Sobig. F перегрузил много почтовых серверов в августе 2003.

Имейте в виду, что Постфикс пытается обеспечить самое надежное возможное обслуживание. Надежность подразумевает последовательность, и именно поэтому Постфикс отклоняет почту, которая адресована неизвестным пользователям по умолчанию, без любого ручного вмешательства. Это хорошо для автономной установки Постфикса, но это также полезно для Постфикс сервера, запущенного на большом сервере, который защищает другие почтовые серверы.

Постфикс определяет валидность получателей, консультируясь с картами. Есть два параметра конфигурации, которые говорят, Постфиксу, где найти эту информацию: local_recipient_maps и relay_recipient_maps. Оба параметра ожидают одну или более карт, которые содержат действительных получателей. local_recipient_ maps параметр, определяет валидных локальных получателей, как показано в этом примере, который определяет получателей в файле паролей Unix и картах псевдонимов:

**# postconf -d local_recipient_maps**

local_recipient_maps = proxy:unix:passwd.byname $alias_maps

С другой стороны, relay_recipient_maps определяет получателей в случае, когда Постфикс, передает сообщения заключительному предназначению (типа сервера почтовых ящиков):

**# postconf -d relay_recipient_maps**

relay_recipient_maps = hash:/etc/postfix/relay_recipients

При использовании relay_recipient_maps, обратите особое внимание на то, чтобы Постфикс, знал всех действительных получателей для релея. Если конечным предназначением является сервер Microsoft Exchange, обратитесь к главе 13 для того чтобы узнать, как извлечь пользовательскую карту.

*ПРЕДОСТЕРЕЖЕНИЕ: использование luserelay отменяет использование параметра local_recipient_maps , потому что делает всех локальных получателей валидными. Аналогично, catchall запись в вашем virtual_alias_ maps, отменяет отклонение почты несуществующих получателей, потому делает всех получателей валидными, Например, следующая запись карты делает всех получателей домена example.com действительными.*

@example.com catchall@localhost

## Почта для не определенным полностью получателя

Адрес, который полностью не определен, типа получателя, содержит локальную часть адреса электронной почты. Хорошо принимать эти адреса для локальных получателей на машине, которая получает почту для единственного домена, но это - большая проблема, как только ваш почтовый сервер получает сообщения и для другого домена.

Это происходит, потому что локальная часть оставляет слишком много места для интерпретации при использовании без доменной части.

Скажем, то, что Вы - ISP и для example.com и example.net , двух конкурирующих видов коммерческой деятельности. Если Вы получаете сообщение для sales, куда oyj идет? Оно должно идти в sales@example.com или в sales@example.net , если сервис

электронной почты для обоих доменов находятся  на одной машине?

Вот почему Вы должны отклонять адреса, которые не полностью определены. Не принемайте ответственность  за то, что не иметь к Вам  какое-либо отношение. Это работа отправителя - подготовить сообщение к надлежащей передаче, и это означает определять уникального получателя.

*ОБРАТИТЕ ВНИМАНИЕ, что есть только одно исключение: Вы должны принять почту для postmaster`a в не FQDN форме.*

Постфикс отклоняет сообщения для не FQDN получателей, как только Вы добавляете правило reject_non_fqdn_recipient  к ограничениям smtpd_recipierrts_restrictions, как в этом примере:

```
smtpd_recipient_restrictions =
    reject_non_fqdn_recipient
    reject _unknown_recipient_domain
    permit_mynetworks
    reject_unauth_destination
    reject_non_fqdn_hostname
    reject_invalid_hostname
    permit
```

Проверьте это, соединяясь с вашим почтовым сервером от удаленного хоста и предлагая неполного получателя конверта. Сессия подобная следующей достаточна для, проверки того, что ограничение работает:

```
$ telnet mail.example.com 25
220 mail.example.com ESMTP Postfix
HELO client.example.com
250 mail.example.com
MAIL FROM:   <sender@example.com>
250 Ok
RCPT To: <recipient>
504 <recipient>: Recipient address rejected: need fully-qualified address
```

## Поддерживание соответствию RFC

Вы вероятно заметили в этой главе, что ограничения могут стать довольно сложными. Обратная сторона ограничений - то, что чем более сложными они становятся, тем выше шанс, что Вы определите то, которое вызовет  сбой Вашей почтовой системы (, не выполняясь оно полностью бесполезно), отклоняя содержание, которое Вы должны принять при всех обстоятельствах. Следующие

разделы показывают Вам, как избежать неосторожного отклонения некоторых или всех отправителей. Это важно, потому что Вы можете случайно исключить отправителей, которые могли бы бы способны сказать Вам, что кое-что является неправильным в вашей конфигурации.

## Пустой отправитель конверта

Сначала, никогда не блокируйте пустого отправителя конверта (<>). Этот адрес принадлежит ДЕМОНУ ОТПРАВКИ , и сервер почты использует его при посылке уведомлений о статусе сообщений или для возврата сообщения. Если Вы заблокируете его, удаленные серверы не могут сообщить вашим пользователям, если кое-что пойдет не так, как надо с сообщениями, котрорые они посылают,

*ПРЕДОСТЕРЕЖЕНИЕ Черные списки, типа dsn.rfc-ignorant.org вносят в список почтовые сервера, категорически блокирующие пустых отправителей конверта, так сервера, использующие эти черные списки не будут принимать почту от таких серверов. (Это обсуждено позже в разделе "Отклонение, доменов из черного списка " Областей Отправителя).*

Все, что Вы должны сделать - обрабатывать пустых отправителей конверта как любого другого действительного получателя и строить хорошие (антиспам) ограничения, чтобы защитить ваших получателей. Позвольте ограничениям делать работу, и если сообщение с пустым отправителем конверта приходит, примите его. В конце концов, любой адрес отправителя может быть фальшивым....

## Специальные почтовые адреса

Есть два адреса, для которых Вы должны всегда принимать сообщения на почтовом сервере; они обязательны для RFC-СОВМЕСТИМОГО почтового сервера:

**postmaster**

Всегда примите почту для postmaster'а это – это центр сообщений о связанных с почтой проблемах. Пользователи должны быть способны войти в контакт с администратором почтового сервера, если они нуждаются в помощи с почтой (см. RFC 2821 http://www.rfc-editor.org/rfc/rfc2821.txt).

**abuse**

Принятие почты для abuse гарантирует, что пользователи могут сказать Вам о потенциальном злоупотреблении почтой, исходящем от вашего почтового сервера (см. RFC 2142 http:// www.rfc- editor.org/rfc/rfc2142.txt ).
Опционально, Вы должны принять сообщения для следующих адресов, если Вы управляете некоторыми серверами (см. RFC 2142; http://www.rfc-editor.org/rfc/rfc2142.txt):

**webmaster**

Принимайте почту для webmaster, если Вы управляете Веб сервером.

**hostmaster**

Принимайте почту для hostmaster, если Вы управляете ДНС сервером.

Вы можете принять сообщения для этих получателей, используя параметр **check_ recipient_access** в комбинации с картой, типа /etc/postfix/role_account_exceptions, которая содержит список получателей, которые должны принять сообщения. Карта могжет быть подобна этой (значение OK для каждого ключа карты говорит, Постфиксу что, необходимо принять сообщения для этого получателя независимо от ограничений получателя):

```
# addresses that you must always accept
postmaster@       OK
abused@           OK
# addresses that you should accept  if you run DNS and WWW servers
hostmaster@       OK
webmaster@        OK
```

После настройки этого файла, преобразуйте его в карту с помощью команды **postmap hash:/etc/postfix/role_account_exceptions**. Затем определите карту как параметр в правиле check_recipient_access в вашем списке ограничений в main.cf. Вот пример настройки:

```
smtpd_recipient_restrictions =
        reject_non_fqdn_recipient
        reject_non_fqdn_sender
        reject_unknown_sender_domain
        reject_unknown_recipient_domain
        check_recipient_access hash:/etc/postfix/role_accouttt_exceptions
        permit_mynetworks
        reject_unauth_destination
        permit
```

После перезагрузки Постфикса, Вы без опасения можете перейти к построению более сложных правил. Карта с исключениями проверке, после того как, Постфикс, проверил правила для неправомочной передачи; таким образом можно безопасно определить postmaster@.

### Порядок прохождения RFC ограничений

Вы, возможно, заметили к настоящему времени, что опции, которые добавляются к smtpd_recipient_ restrictions в предыдущих разделах были определены не в том же самом как непосредственно разделы. Это – сделано, потому что правила ограничений могут конфликтовать друг с другом, если они не находятся в надлежащем порядке. Взгляните на следующей список:

```
smtpd_recipient_restrictions =
        reject_non_fqdn_recipient
        reject_nonfqdn_sender
        reject_unknown_sender_domain
        reject_unknown_recipient_domain
        permit my_networks
        reject_unauth_destination
        check_recipient_access hash:/etc/postfix/role_account_exceptions
        reject_multi_recipient_bounce
        reject_non_fqdn_hostname
        reject_invalid_hostname
        permit
```

Правило permit_mynetworks обозначает важную границу между клиентами из вашей внутренней сети и клиентами извне . Правила, которые появляются до и включительно этот пункт, относятся и к внутренним и внешним клиентам, но те что ниже permit_mynetworks относятся только к внешним клиентам.
Правила, которые предшествуют permit_mynetworks, требуют базового соответствия RFC от всех клиентов, являются ли они внутренними или внешними.

Правило reject_unauth_destination препятствует вашему серверу становиться открытым релеем. Лучше всего не определить никаких правил, которые позволят сообщениям проходить прежде, чем Вы определите permit_mynetworks. После этого хорошо ввести правило reject_unauth_destination как можно скорее, чтобы удостовериться, что там не никакого способа, спомошью которого неправомочный хост может использовать ваш сервер как открытый релей.
Проверка SMTP авторизации должна идти между reject_unaut_destination и permit_mynetworks. Тогда, перед определений дальнейших правил отклонения, используйте check_recipient_access, чтобы позволить безоговорочную доставку на специальные почтовые ящики в вашей системе.
Наконец, после прохождения правил reject_multi_recipient_bounce, reject_non_fqdn_hostname, reject_invalid_hostname, Вы можете принять сообщения правилом premit.

## Антиспам меры

Спамеры должен маскировать происхождение их сообщений, если они не хотят быть привлеченными к суду. Обычно они будут фальсифицировать отправителя конверта или пробовать убаюкать сервер получателя, говоря что, их клиенту нужно доверять - что он принадлежит локальной сети. Ограничения могут проверить и отклонить такие сообщения. Кроме того, они могут просмотреть черные списки, куда занесены спамеры и другие стороны, от которых Вы не хотите получить сообщения. Следующий раздел покажет Вам, как эфиктивно использовать такие ограничения.

## Предотвращение Очевидных Подделок

Некоторое спам программное обеспечение пробует маскировать происхождение

сообщения, используя имя хоста вашего сервера почты как его собственный в приветствии HELO/EHLO. Для Постфикса, это выглядит как парадокс, потому что единственный хост, которому позволено использовать имя хоста вашего сервера - непосредственно сам сервер. Однако, Постфикс, никогда не соединяются с его smtpd демоном, чтобы послать почту самому себе, если это не результат не правильного конфигурирования, и создана петля почты.
Добавление этих ограничений после правила premit_mynetworks, заставит их применяться только к внешним клиентам а не к прокси фильтрам или локальным клиентам с неполной SMTP реализацией.
Поэтому, Вы можете благополучно отклонить SMTP связь с любым клиентом, который приветствует ваш почтовый сервер с именем хоста вашего сервера. Чтобы сделать это, сначала создайте файл карты, названный /etc/postfix/helo_checks, который содержит разновидности имени вашего хоста. Вот некоторые примеры, которые охватывают имя хоста, IP адрес хоста, и заключенный в скобки IP адрес, который клиенты вне сервера почты не должны использовать:

/^mail\.example\.com$/ 550 Don't use my hostname

/^192\.0\.34\.166$/ 550 Don't use my IP address

/^\[192\.0\.34\.166\]$/ 550 Don't use my IP address

Согласно RFC 2821, IP адрес совершенно отдельный - не действительный аргумент HELO команды. IP адрес позволяется, если он используется в виде [ipv4address] (заключен в квадратные скобки) или как адрес IPv6, [ipv6:ipv6address], также заключенный в квадратные скобки. Для отказа в обслуживании любого клиента, который посылает не заключенный в скобки IP адрес, добавьте эту строку:

/A[0-9.]+$/ 550 Your client is not RFC 2821 compliant

Чтобы поместить карту в действие, определите ее (и его тип) как аргумент check_helo_access правила в вашем smtpd_recipient_restrictions наборе ограничений. Вот, как это могло бы выглядеть:

smtpd_recipient_restrictions =

reject_non_fqdn_recipient

reject_non_fqdn_sender

reject_unknown_sender_domain

rej ect_unknown_recipient_domain

permit_mynetworks

reject_unauth_destination

check_recipient_access hash:/etc/postfix/role_account_exceptions reject_nonfqdn_hostname

reject_invalid_hostname

**check_helo_access pcre:/etc/postfix/helo_checks**

permit

Чтобы проверять его, соединитесь с вашим почтовым сервером , и введите ваше собственное имя в приветствии HELO. Вы должны получить отклонение, как показано в этом примере сессии:

**$ telnet mail.example.com 25**

220 mail.example.com ESMTP Postfix

**HELO mail.example.com**

250 mail.example.com

**MAIL FROM: <sender@example.com>**

250 Ok

**RCPT TO: <recipient@example.com>**

550 «mail.example.com>: Helo command rejected: Don't use my hostname

**QUIT**

221 Bye

## Поддельные ответы сервера имен

Постфикс может отклонить сообщения, если есть свидетельства того, что ответы сервера имен для HELO домена, домена отправителя и домена получателя подделаны или не позволяют правильную транспортировку сообщения. Вот - некоторые сомнительные вещи, которые Вы можете увидеть в ответах DNS:

### Поддельные сети

Некоторые почтовые сервера утверждают, что были от сетей, с которыми Постфикс, не может иметь связи, включая частные IP сети, которые Вы не используете (см. RFC 1918, ftp://ftp.rfc-editor.org/in-notes/rfcl918.txt), интерфейс обратной сети, широковещательные адреса, и сети.

### Spam приюты

Spam приюты - сети, известные, как принадлежащие спамерам или тем, кто предаставляет им услуги. Возможно отклонять все сообщения от таких домнов. Вы можете найти спам приюты или spam операторов на ROKSO (Регистр Известных Спам Операторов, http://www.spamhaus.org/rokso/index.lasso.

### Подставные MTA

Подставной MTA утверждает, что он ответственный за любой домен, даже за тот, который не существует. Это не было бы проблемой, потому что Вы можете отказаться от доступа, из неизвестного домена отправителя или для неизвестного домена получателя. К сожалению, некоторые регистраторы доменов нашли яркую идею, что они могут переадресовывают неизвестные имена доменов к их собственному домену. Это обеспечивает действительный ответ на неизвестные домены, и поэтому делает ограничения reject_unknown_sender_domain и reject_unknown_recipient_domain бесполезными.

*ОБРАТИТЕ ВНИМАНИЕ, что первая регистрирующая организация доменов, которая начала переадресовать неизвестные домены была VeriSign (http: // www.verisign.com ) в 2003. VeriSign злоупотреблял ее властью над .net и .com адресным пространством и переадресовывал все несуществующие .com и .net домены к своему собственному сайту (http://sitefindei.veiisign.com). Кроме того, VeriSign настроила свою собственную почтовую службу для неизвестных доменов, что лишило возможности отклонять сообщения от неизвестных доменов. Это - прямое приглашение для спамеров, и Вы можете отклонить сообщения от подставного MTA блокировав хозяина MX в подставном домене.*

Все перечисленные методы или обеспечивают поддельные отчеты ДНС серверов или поддерживают спамеров. Чтобы отклонять почту от таких доменов и сетей, Вы можете создать файл карты, названный /etc/postfix/bogus_mx, который держит IP адреса в ответе ДНС наряду с типом ответа, который Вы хотите дать им (см. Приложение C для полного списка ответов). Вот - пример файла карты:

```
#  bogus networks
0.0.0.0/8                   550 Mail server in broadcast network
10.0.0.0/8                  550 No route to your RFC 1918 network
127.0.0.0/8                 550 Mail server in loopback network
255.0.0.0/4                 550 Mail server in class D multicast network
193.168.0.0/16              550 No route to your RFC 1918 network
# spam havens
69-6.0.0/18       550 RE1ECT Listed on Register Of Known Spam Operations      1
#  wild-card
64.94.110.11/32           550 REJECT VeriSign Domain wildcard                 2
```

1 Эта сеть была внесена в список на spamhaus.org (<http: // www.spamhaus.org/sbl/> sbl.lasso? query=SBL6636) как сеть, известная как пораждающая спам spam, когда мы писали книгу.
2 Этот хост, как известно, действовал как подставной MTA во время написания.

Так как мы редактируем карту типа CIDR, которая является последовательным типом карты (см. Главу 5), Вы не нуждаетесь и можете не конвертировать ее используя команду postmap. Постфикс будет использовать файл, как он есть. Просто добавьте check_sender_mx_access опцию с картой в качестве аргумента нашему smtpd_recipient_restrictions параметру. Например так:

```
smtpd_recipient_restrictions =
                        reject_nonfqdn_recipient
                        reject_nonfqdn_sender
                        reject_unknown_sender_domain
                        reject_unknown_recipient_dornain
                        permit_mynetworks
```

reject_unauth_destination

check_recipient_access hash:/etc/postfix/role_account_exceptions

reject_non_fqdn_hostname

reject_invalid_hostname

check_helo_access pcre:/etc/postfix/helo_checks

check_sender_mx_access cidr:/etc/postfix/bogus_mx

permit

Ограничение вступят в силу после перезагрузки Постфикса. Вы можете видеть эффект ограничения в лог файле:

Sep 17 12:19:23 mail postfix/smtpd[3323]: A003D15C021: reject: RCPT from

unknown[61.238.134.162]:

554 <recipient@example.com>: Sender address rejected: Verisign Domain

wildcard;

from=<alli.k_lacey_mq@joymail.com> to=<recipient@example.com> proto=ESMTP

helo=<example.corn>

Вы можете проверить IP адрес с помощью команды host:

**# host -t mx joymail.com**

**# host -t a joymail.com**

joymail.com has address 64.94.110.11

*ОБРАТИТЕ ВНИМАНИЕ, что этот домен реально существует теперь; запись показана такой какой она была в октябре 2003.*

## Отклонение почты многократным получателям

В разделе пустые отправители конверта, Вы узнали, что Вы не должны блокировать пустых отправителей конверта. Есть одно исключение к этому правилу - Вы должны блокировать почту с пустым отправителем конверта, посланным нескольким получателям, потому что нет в настоящее время никакого известного законного использования уведомлений о статусе сообщения посланным нескольким получателям, так что любые такие сообщения вероятно, незаконны.

Чтобы отклонять сообщения от пустого отправителя конверта многократным получателям, добавьте reject_multi_recipient_bounce правила к вашему списку smtpd_recipient_ restrictions. Оно может быть записано хоть где в списке ограничений, но следующем примере, оно появляется после permit_mynetworks.

smtpd_data_restrictions =

**reject_mu1ti_recipient_bounce**

Как заявлено в документации, reject_multi_recipient_bounce может надежно использоваться только в smtpd_data_restrictions, когда известны все получатели. Вы можете проверить это с помошью ручной связи, также, как Вы делали раннее для других ограничений. Передача от пустого отправителя конверта для многократных получателей должна кончиться отказом в обслуживании, как показано в следующей сессии:

**$ telnet localhost 25**

220 mail.example.com ESMTP Postfix

**EHLO client.example.com**

250-mail.example.com

250-PIPELINING

250-SIZE 10240000

250-VRFY

250-ETRN

250 8BITMIME

**MAIL FR0M:<>**

250 Ok

**RCPT TO: <recipienti@example.com>**

**RCPT TO: <recipient2@example.com>**

**550 : Recipient address rejected: Multi-recipient bounce**

**QUIT**

221 Bye

## Использование черных списков DNS

Сервер черных списков ДНС - сервер, который говорит Вам о ресурсах (типа IP адресов, отправителей конверта, и доменов), которые являются вероятно ненадежными. Черные списки могут быть очень полезны чтобы блокировать почту, посланную от клиентов вашему серверу, если Вы выберете правильный черный список. Однако, выбрав неправильный черный список вы можете закончиться отказом сервера для получателей, которую Вы можете счесть законной. Убедитесь, что проверили политику черного списка перед его использованием. Любой сайт, управляющий черным списком должен внести в список критерии, которые он применяет при помещении в черный список ресурс, и он должн издать и обеспечить прямую процедуру для того, чтобы удалить ресурсы, которые больше не должны быть помещены в черный список.

Если Вы ищете черный список, хорошее место, чтобы начаться - dmoz.org (http: // dmoz.org/Computers/Internet/Abuse/Spam/Blacklists ).

*ПРЕДОСТЕРЕЖЕНИЕ, Все черные списки базируются на службе ДНС, это означает, что Постфикс, должен выполнять DNS запросы, Не кешируемые DNS запросы могут занимать до секунды, и частота с которой сервер может принимать сообщения, значительно понизятся. Поэтому, проверки черных списков относительно дороги в смысле времени ожидания. Вы должны всегда*

*использовать их как последние правила в ваших списках ограничений.*

## Отклонение клиентов помещенных в черный список.

Вы можете отклонить помещенных в черный список клиентов, используя DNSBL (Основанные на ДНС черные Списки) черные списки. Постфикс имеет reject_rbl_client опцию ограничений, который берет FQDN имя хоста в качестве аргумента сервера черного списка. Вот - пример использования опции:

```
smtpd_recipient_restrictions =
        reject_non_fqdn_recipient
        reject_non_fqdn_sender
        reject_unknown_sender_domain
        reject_unknown_recipient_domain
        permit_mynetworks
        reject_unauth_destination
        check_recipient_access hash:/etc/postfix/rolea_ccount_exceptions
        reject_non_fqdn_ hostname
        reject_invalid_hostname
        check_helo_access pcre:/etc/postfix/helo_checks
        reject_rbl_client reloys.ordb.org
        permit
```

*ПРИМЕЧАНИЕ, чтобы увидеть, внесен ли клиент в список DNSBL, замените на обратный порядок четыре октета IP адреса клиента (то есть замените a.b.c. d на d.c.b.a), добавьте в конец rbl.domain (типа relays.ordb.org ), и ищите результат. Если хост помещен в черный список, Вы получите ответ, указывающий на оригинальный IP адрес, как в этом примере:*

```
$ host 2.0.0.127.relays.ordb.org
2.0.0.127.relays.ordb.org A 127.0.0.2
```

*Результаты с несколькими значениями*

Постфикс может использовать эту дополнительную информацию (хост не только внесен в список, но возвращенный IP адрес позволяет различить, почему оу внесен в список). Например, следующая конфигурация отклоняет сообщения от любого хост, который соответствует A записи 127.0.0.2 в нашей воображаемом домене, tld черном списке:

```
reject_rbl_client domain.tld=127.0.0.2
```

## Отклонение помещенных в черный список доменов отправителя

В дополнение к ограничению почты от IP адресов, Вы можете блокировать почту от помещенных в черный список доменов отправителя. Этот вид черного списка называют черным списком правой стороны (RHSBL). Конфигурирование Постфикса для RHSBL, подобно конфигурированию для DNSBL . Пример в этом разделе использует специальный черный список dsn.rfc-ignorant.org :

**www.rfc-ignorant.org миссия:**

Мы обслуживаем множество списков (в настоящее время dsn, abuse, postmaster, bogusmx и whois), которые содержат ljvtys, администраторы которых хотят не повиноваться RFCs, стандартный блок "правил" сети.
Важно обратить внимание, что НИЧТО не требует, ЧТОБЫ НИКТО исполнил RFC (дословно "Запрос Комментариях"), однако, " способность к взаимодействию" сети, базируется на том что каждый имеет ту же самую "книгу правил" и следование им. Внесение в список просто подразумевает, что сайт не хотел не осуществлять условия, описанные в специфических RFC. Это, конечно, дело других сайтов, решить для себя, действительно ли они желают связаться с сайтами, которые не хотели выполнять, скажем, RFC2142, и имеют рабочий <abuse@domain> адресс.
-dredd, www.rfc-ignorant.org

Есть много MTAs, которые не принимают почту, к которой обязывает RFCs (например, они могли бы отказаться от пустых отправителей конверта), по множеству ошибочных причин, включая эти:

- поддельная почта от анонимных отправителей, не разрешена
- отвергают пустого отправителя (чтобы сразиться с spam проблемой)

Комментарий к этим ошибочным сообщениям не-RFC- совметимым почтовым серверам:
Любой может подделать чей - то адрес электронной почты. Вы могли бы отсылать электронную почту так president@whitehouse.gov , и это будет точно таким же анонимным как пустой отправитель конверта.
Спам можно послать с произвольными отправителями, но отвергать можно только послать с пустым отправителем конверта.
Любой почтовый сервер, который блокирует пустых отправителей конверта, лишает его пользователей от знаний, что их почта, возможно, была отклонена другим почтовым, потому что их сервер блокирует уведомления, посланные другим RFC-ПОСЛУШНЫМ сервером, который использует пустого отправителя конверта, так как описано в RFC:
RFC 2821 явно требует того, что MTA должен, принимают почту с пустым отправителем конверта, потому что "использование пустого отправителя при посылке уведомления предотвращает петлю при доставке уведомлений между двумя системами.

Постфикс имеет опцию reject_rhsbl_sender ограничений, который отделяет локальную часть от любого адреса электронной почты и использует домен, для опроса черного списка (типа dsn.rfc-ignorant.org ). Если домен отправителя конверта находится в черном списке, Постфикс, отклоняет входящие сообщение. Подобно другим опциям черных списков, Вы должны разместить эту опцию в конце списка, как в этом примере:

smtpd_recipient_restrictions =

reject_non_fqdn_recipient

reject_non_fqdn_sender

reject_unknown_sender_domain

reject_unknown_recipient_domain

permit_mynynetworks

reject_unauth_destination

reject_multi_recipient_bounce

reject_rbl_client relays.ordb.org

check_recipient_access hash:/etc/post-fix/role_accoijnt_exceptions

reject_non_fqdn_hostname

reject_invalid_hostname

check_helo_access pcre:/etc/postfix/helo_checks

**reject_rhsbl_sender dsn.rfc-ignorant.org**

permit

Перезагрузка сделает изменения действительными, и Вы можете проверить их, соединяясь с вашим сервером и используя отправителя конверта от домена, внесенного в список в dsn.rfc-ignorant.org <http://ignorant.org>, как в этом примере (sender@exanple.com – официальный тестовый адрес):

**$ telnet localhost 25**

220 mail.example.com ESMTP Postfix

**EHLO client.example.com**

250-mail.example.com

250-PIPELINING

250-SIZE 10240000

250-VRFY

250-ETRN

250 8BITMIME

**MAIL FROM:<sender@example.com>**

250 Ok

**RCPT TO: <recipientgexample.com>**

**554 Service unavailable; Sender address [sender@example.com] blocked **

**using dsn.rfc-ignorant.org; Not supporting null originator (DSN)**

**QUIT**

221 Bye

## Ручная проверка черного списка

Проверка домена в RHSBL подобна процедуре проверки IP адреса, за исключением того, что Вы не должны полностью изменять порядок элементов. Просто добавьте в конец имени домена который Вы хотите проверить сервер черного списка, и выполните DNS запрос.
Вот результат проверки домена, который не находится в черном списке:

**$ host postfix-book.com.dsn.rfc-ignorarit.org**

Host postfix-book.com.dsn.rfc-ignorant.org not found:  3(NXDOMAIN)

Результат для домена внесенного в черный список будет выглядеть так:

**$ host example.com.dsn.rfc-ignorant.org**

example.com.dsn.rfc-ignorant.org has address 127.0.0.2

## Исключения для доменов отправителей помещенных в черный список

Если Вы хотите отклонять почту от почтовых серверов, которые не следуют  rfc правилам, но Вы должны обеспечивать связь со специфическим доменом, который была бы отклонен в соответствии с вашими ограничениями, Вы можете создать список исключений. Используйте check_sender_access правило с картой доменов, для которых необходимо сделать исключение,
Сначала, создайте файл типа /etc/postfix/rhsbl_sender_exceptions содержащие пользователей и домены, от которых Вы хотите принять сообщения. Например, следующий файл разрешает почту от всех пользователей домена example.com  и для единственного пользователя sender@example.org:

example.com OK

sender@example.org OK

После создания файла, используйте postmap hash:/etc/postfix/rhsbl_sender_exceptions , чтобы построить карту. Затем добавьте правило check_sender_access непосредственно перед reject_rhsbl_sender, как в этом примере:

```
smtpd_recipient_restrictions =
        reject_non_fqdn_recipient
        reject_non_fqdn_sender
        reject_unknown_sender_domain
        rejec t _unknown_recipient_domain
        permit_mynetworks
        reject_unauth_destination
        reject_multi_recipient_bounce
```

reject_rbl_client relays.ordb.org

check_recipient_access hash:/etc/postfix/role_account_exceptions

reject_non_fqdn_hostname

reject_invalid_hostname

check_helo_access pcre:/etc/postfix/helo_checks

***check_sender_access hash:/etc/postfix/rhsbl_sender_exceptions***

reject_rhsbl_sender dsn.rfc-ignorant.org

permit

*ОБРАТИТЕ ВНИМАНИЕ что этот пример использует check_sender_access, но вот - полный список вариантов исключений:*

- check_sender_access
- check_client_access
- check_helo_access
- check_recipient_access

Вы уже видели пример использования check_helo_access в разделе "Предотвращение очевидных подделок. " Документация Постфикс объясняет оставшиеся два.

## Подтверждение Отправителя

Венец антиспам инструментов Постфикса - проверка адреса отправителя, которая проверяет, что отправитель конверта существует в домене отправителя: если отправитель не существует, Постфикс, не принимает сообщение.

К сожалению, эта способность дорога, потому что процесс проверки занимает время и потребляет дополнительные ресурсы системы. Он состоит из следующих шагов:

- Клиент представляет отправителя конверта.
- Постфикс, генерирует и добавляет в очередь сообщение исследования отправителя конверта.
- Постфикс, ищет MX запись домена отправителя конверта.
- Постфикс пытается соединиться с почтовым сервером отправителя. Если он не может соединиться с удаленным сервером, smtpd отсрочивает решение принять сообщение, возвращая временный код ошибки 450 клиенту. Тем временем, Постфикс, продолжает пробовать проверить адрес.
- Постфикс инициирует SMTP сессию с удаленным сервером.
- Постфикс, использует отправителя конверта в качестве получателя конверта, чтобы соединится с удаленным почтовым сервером.

- В зависимости от ответа удаленного сервера, Постфикс, может сделать одну из двух вещей:
- если удаленный почтовый сервер принимает, данного получателя (первоначального отправителя конверта), Постфикс, разъединяет связь, и уничтожает сообщение исследования, и принимает сообщение от первоначального клиента.
- Если удаленный почтовый сервер отклоняет получателя (первоначального отправителя конверта). Постфикс разъединяет связь, уничтожает сообщение

исследования, и отклоняет сообщение от клиента

При активизации проверки адреса, нормальная почта получит задержку до девяти секунд, в то время как Постфикс проверит адрес первый раз . Однако, Постфикс кеширует статус адреса, так что последующие сообщения не имеют никакой задержки.

Если процесс проверки длится дольше, чем девять секунд, smtpd отклоняют почту от клиента (посылая машине код 450 в ответ). Нормальные почтовые клиенты соединятся снова после некоторой задержки. Однако они не получат доступ так как, они только повторяют команды, и администратор сервера вряд ли захочет терять время на проверку адреса.

## Конфигурация Проверки Адреса отправителя

Чтобы задействовать проверку адреса отправителя, добавьте правило reject_unverified_sender к вашим smtpd_recipient_restrictions ограничениям, как показано в этом примере:

```
smtpd_recipient_restrictions =
        reject_non_fqdn_recipient
        reject_non_fqdn_sender
        reject_unknown_sender_domain
        reject_unknown_recipient_domain
        permit_mynetworks
        reject_unauth_destination
        reject_multi_recipient._bounce
        reject_rbl_client relays.ordb.org
        check_recipient_access hash:/etc/post-fix/role_account_exceptions
        reject_non_fqdn_hostname
        reject_invalid_hostname
        check_helo_access pcre:/etc/postfix/helo_checks
        check_sender_access hash:/etc/postfix/rhsbl_sender_exceptions
        reject_rhsbl_sender dsn.rfc-ignorant.org
        reject_unverified_sender
        permit
```

Есть несколько других опций кроме reject_unverified_sender, который Вы можете добавить к вашим ограничениям. Однако, параметры идут с разумными установками по умолчанию, и они служат для, настройки проверке адреса отправителя, а не для его формирования. Следующие подразделы описывают самые общие изменения поведения проверки адреса отправителя. Вы можете найти дополнительные параметры настроек в файле ADDRESS_VERIFICATION_README файле, который идет с Постфиксом.

## Отправитель Конверта Исследования

Когда Постфикс, генерирует сообщение исследования, чтобы проверить рассматриваемого отправителя, он должен представиться удаленному серверу с собственным адресом отправителя конверта. Вы можете формировать этот адрес с помощью параметра address_verify_sender По умолчанию - postmaster@$myorigin.
Если Вы хотели установить другого отправителя конверта исследования, добавьте address_ verify_sender параметр в файл main.cf , как в этом примере:

address_verify_sender = sender@example.com.

Конечно, этот отправитель должен существовать, потому что другие серверы могут использовать ту же самую проверку адреса отправителя против Вас.

*ОБРАТИТЕ ВНИМАНИЕ На адрес  указанный адресс получателя, , поскольку параметр для address_verify_sender освобожден от любых ограничений.*

## Кэширование

По умолчанию, Постфикс, хранит проверенных отправителей в памяти. Если Вы перезагружаете, или перезапускаете Постфикс, Вы потеряете их, если Вы не определите дополнительную базу данных, чтобы постоянно хранить адреса. Чтобы использовать базу данных, установите address_verify_map параметр как путь к файлу базы данных (удостоверьтесь, что Вы выбираете раздел диска, который имеет много свободного места). Вот – пример:

address_verify_map = btree:/var/spool/postfix/verified_senders

После перезагрузки, Постфикс, создаст базу данных и продолжет добавлять и положительные и отрицательные результаты проверки. Если Вы хотите исключить кеширование отрицательных результатов, устанавливаете address_verify_negative_cache параметр в main.cf следующим образом:

address_verify_negative_cache = no

## Выборочная проверка адреса отправителя

При увеличении нагрузки на Ваш почтовый сервер, проверка адреса отправителя, наиболее вероятно, станет узким местом. В этом случае, Вы должны перейти к выборочной проверке адреса отправителя.
Работа выборочной проверки адреса отправителя состоит в создании карты доменов отправителей конверта которые обычно используют спамеры. Если  домен отправителя конверта находится в карте, Постфикс проверяет отправителя, но если домена нет в списке проверка не выполняется. Создайте файл карты типа /etc/postfix/common_spam_sender_domains, и установите reject_unverified_sender параметр как действие  которое выполняется если отправитель конверта соответствует домену. Вот - пример того, как выглядит такой файл:

hotmail.com reject_unverified_sender

web.de reject_unverified_sender

msn.com reject_unverified_sender

mail.ru reject_unverified_sender

Access (5) man страница объясняет, что правая сторона этой карты - имя применяемого ограничения или класса ограничений. В этом примере, Постфикс, выполняет одно из двух действий, когда клиент начинает передачу сообщения:

- если домен отправителя соответствует, записи в common_spam_sender_domains, поиск в карте возвращает reject_unverified_sender, и Постфикс, проверяет отправителя конверта. Если он существует, reject_unverified-sender возвращает DUNNO, и Постфикс, оценивает следующее ограничение. Если адрес недействителен, Постфикс, отклоняет сообщение.
- если домен отправителя не соответствует, записи в common_spam_ sender_domains, поиск в карте терпит неудачу, выборочная оценка возвращает DUNNO, и Постфикс, оценивает следующее ограничение, не проверяя адрес отправителя.

После создания карты, конвертируйте ее в базу данных, используя команду postmap hash:/etc/postfix/common_spam_sender_domains . Наконец, замените существующее правило reject_unverified_sender на check_sender_access с картой в качестве аргумента. Вот - пример использующий карту hash:/etc/postfix/common_spam_ sender_domains :

smtpd_recipient_restrictions =

reject_non_fqdn_recipient

reject_non_fqdn_sender

reject_unknown_sender_domain

reject_unknown_recipient_domain

permit_mynetworks

reject_unauth_destination

reject_multi_recipient_bounce

check_recipient_access hash:/etc/postfix/role_account_exceptions

reject_non_fqdn_hostname

reject_invalid_hostname

check_helo_access pcre:/etc/postfix/helo_checks

check_sender_access hash:/etc/postfix/rhsbl_sender_exceptions

reject_rhsbl_sender dsn.rfc-ignorant.org

***check_sender_access hash;/etc/postfix/comon_spam_sender_domains***

permit

Вы можете развить этот шаг далее и ввести другие критерии кроме отправителя конверта, например содержание. Создайте другую карту, названную common_spam_ sender_domain_keywords, который включает ключевые слова доменного имени,

вызывающие проверку адреса отправителя,  например:

/sex/ reject_unverified_sender

/girl/ reject_unverified_sender

/sell/ reject_unverified_sender

/sale/ reject_unverified_sender

/offer/ reject_unverified_sender

/power/ reject_unverified_sender

Тогда добавьте другое правило check_sender_access  указывающее на карту:

smtpd_recipient_restrictions =

reject_non_fqdn_recipient

reject_non_fqdn_sender

reject_unknown_sender_domain

reject_unknown_recipient_domain

permit_mynetworks

reject_unauth_destination

reject_multi_recipient_bounce

check_recipient_access hash:/etc/postfix/role_account_exceptions

reject_non_fqdn_hostname

reject_invalid_hostname

check_helo_access pcre:/etc/postfix/helo_checks

check_sender_access hash:/etc/postfix/rhsbl_sender_exceptions

reject_rhsbl_sender dsn.rfc-ignorant.org

check_sender_access hash:/etc/postfix/common_spam_sender_domains

***check_sender_access regexp:/etc/postfix/common_spam_sender_domain_keywords***

permit

## Порядок прохождения ограничений

Антиспам защита дорога с точки зрения ресурсов системы. Следующий список ограничений показывает, в каком порядке использовать антиспам правила.

smtpd_recipient_restrictions =

reject_non_fqdn_recipient

reject_non_fqdn-sender

reject_unknown_sender_domain

```
        reject_unknown_recipient_domain
        permit_mynetworks                                   1
        (permit_sasl_authenticated)
        (pop-before-smtp)
        reject_unauth_destination
        check_recipient_access hash:/etc/postfix/role_account_exceptions
        reject_multi_recipient_bounce
        check_helo_access pcre:/etc/postfix/helo_checks     2
        reject_non_fqdn_hostname
        reject_invalid_hostname
        check_sender_mx_access /etc/postfix/verisign_mx_access   3
        check_sender_access hash:/etc/postfix/rhsbl_sender_exceptions   4
        reject_rhsbl_sender dsn.rfc-ignorant.org            5
check_sender_access hash:/etc/postfix/common_spam_sender_domains   6
check_sender_access regexp:/etc/postfix/common_spam_sender_domain_keywords
        permit
```

Общее правило:  указывать "дешевые" ограничения перед "дорогими" :

**1** Разместите все антиспам правила после permit_mynetworks так, чтобы они относились только к внешним клиентам (клиенты, не внесенные в список в mynetworks).
**2** Вы может отклонить любого клиента который использует имя вашего сервера без любого дальнейшего исследования. Не имеет значения, испойзуют ли они используют non-FQDN имя  или недействительное имя.
**3** Эта опция отмечает начало дорогих ограничений. check_sender_mx_access требует одного или двух DNS запросов. Если Вы управляете кэширующим ДНС сервером, Вы можете выполнять DNS запросы локально.
**4** Эта карта идет перед опцией черного списка, потому, что она содержит исключения для пользователей и доменов, которые могли бы иначе отклонены.
**5** Эта опция дорога, требуя выполнения запроса к удаленной системе (DNS сервер для dns.rfc-ignorant.org ), который может быть под тяжелой нагрузкой или временно не доступен. Именно поэтому она находится близко к концу  списка ограничений.
**6** два самых дорогих действия находятся последними. Если они срабатывают, Постфикс, должен создать фиктивное сообщение, попытаться доставить его, и зарегистрировать результат. Это. Поэтому, они – последние в списке.

### Использование классов ограничений

Пример, приведенный в этом разделе ограничивает отправителей конверта двумя способами. Сначала, это требует, чтобы почта из вне *не имела* отправителя в вашем домене ; и , во вторых требует, чтобы почта от внутренних клиентов *имела* отправителя находящегося в вашем домене.

Идея состоит в выполнении, Постфиксом сначала проверке, для определения, происходит ли входящая связь от клиента из вашей сети:

1. Если клиент находится в вашей сети, Постфикс, применяет к клиенту класс ограничений. Данный класс содержит проверку адреса отправителя конверта
   - если отправитель конверта соответствует вашему доменному имени, проверка возвращает *OK.* Постфикс прекращает, проверку ограничений и позволяет обработку клиента.
   - если отправитель конверта не соответствует, домену, ограничения возвращает- *reject,* таким образом Постфикс отказывает в обслуживание клиента

2. Если клиент не принадлежит вашей сети, Постфикс, не использует класс ограничений. Вместо этого, он перемещается вперед к следующему правилу ограничений, которое проверяет адрес отправителя конверта.
   - если клиент использует ваш домен как, часть адреса отправителя конверта, Постфикс отказывает в обслуживание.
   - если клиент не использует ваш домен как часть отправителя конверта, он проходит испытание и перемещается в следующее ограничение.

Чтобы осуществить то, о чем мы только что написали, создайте файл карты, содержащий список IP адресов и подсетей в вашей сети. Вы можете назвать файл /etc/postfix/internal_networks; он должен иметь вид:

```
192.0.34        has_our_domain_as_sender
192.168         has_our_domain_as_sender
192.168.1       has_our_domain_as_sender
```

Затем, создайте другой файл карты, названный /etc/postfix/our_domain_as_sender содержащий ваше доменное имя и пустого отправителя конверта (помните, что ваш сервер должен принять письма для него без вопросов); это - список доменов отправителя конверта, которые могут использовать внутренние клиенты . Файл карты имеет вид:

```
example.com     OK
<>              OK
```

Теперь, создайте файл карты, который содержит ljvtys, которые внешние клиенты не могут использовать в их отправителе конверта. Для этого примера, мы будем использовать имя файла /etc/postfix/not_our_domain_as_sender, содержащий только:

```
example.com     554 Do not use my domain in your envelope sender
```

После создания карт для этих двух файлов с помощью команды postmap, установите класс ограничений и требуемые правила ограничений в вашем main.cf файле:

```
smtpd_restriction_classes =
        has_our_domain_as_sender
has_our_domain_as_sender =
        check_sender-acccss hash:/etc/postfix/our_domain_as_sender
```

reject smtpd_recipient_restrictions =

check_client_access hash:/etc/postfix/internal_networks

check_sender_access hash:/etc/post-fix/not_our_domain_as_sender

reject_unauth_destination

...

permit

Как обычно, Вы должны будете перезагрузить конфигурацию Постфикса , чтобы изменения имели эффект.

# КАК РАБОТАЮТ ВСТРОЕННЫЕ ФИЛЬТРЫ СОДЕРЖИМОГО

Фильтры исследуют содержание сообщения и выполняют предопределенное действие, основанное на содержании. Эта глава показывает Вам фильтры, которые доступны и что действия которые обеспечивает Постфикс, для контроля содержания сообщения.

Ограничения дополняются проверками. Принимая во внимание, что ограничения контролируют SMTP связь,проверки контролируют содержание сообщения. Сначала, тем не менее, Вы могли бы увидеть что проверки очень сильно отличаются от ограничений. Проверку легко задействовать, но синтаксис используемый обычно для создания образцов поиска, может показаться сложным, потому что проверки используют регулярные выражения, для определения образцов поиска.

Поскольку Постфикс является прежде всего MTA, встроенные фильтры, не предназначены, для замены полноценного сканера содержимого; скорее, они обеспечивают средства для выполнения простых задач. Вот - некоторые из вещей, которые Вы можете сделать с помощью фильтров:

- Ложные сообщения сгенерированные некоторыми программы, типа SAP почтового шлюза.
- Блокировка сообщений с определенными заголовками.
- Вылавливание сообщений, содержащих потенциально опасные вложения.
- Удаление части информации из заголовков

*ОБРАТИТЕ ВНИМАНИЕ, если Вы никогда не работали с регулярными выражениями прежде, Вы, возможно, должны потратить некоторое время, на изучение основ. Mastering Regular Expressions, 2nd Edition Jeffrey E.F. Friedl - превосходная книга по предмету.*

## Как Работают проверки?

Проверки просматривают сообщения для поиска набора образцов. Если образец соответствует любому содержанию в сообщении, будет выполнено действие. Постфикс может применять отдельные фильтры к различным частям сообщения; разделы, поддерживаемые Постфиксом в настоящее время следующие:

- заголовки сообщений
- MIME заголовки
- части тела сообщений, включая вложения
- заголовки вложенных сообщений

Чтобы создавать набор фильтров, Вы определяете отдельные образцы поиска, используя отдельные карты, и затем назначаете эти карты различным параметрам фильтров, которые обращаются к разделам. Постфикс использует карты со

встроенным анализатором MIME, чтобы исследовать содержимое сообщения. Анализатор работает подобно команде *egrep*; он может только распознавать слова обычного текста на одной строке одновременно. Вот, как процесс работает:

1. Анализатор следует через сообщение строка за строкой
2. Анализатор решает, какому разделу сообщения принадлежит текущая строка
3. Если проверка существует для раздела, Постфикс использует назначенную карту, для поиска соответствия сообщения и образцов.
4. Если соответствие образцу найдено , Постфикс выполняет, действие, и выполнение дальнейших проверок будет прекращено. Поэтому, первое соответствие "выигрывает".

Как Вы, возможно, отметили к настоящему времени, проверки загружают CPU, и Вы можете также видеть, что порядок образцов поиска в карте может стать критическим, потому что более раннее совпадение заставляет процесс проверки использовать меньше времени центрального процессора.

## Применение фильтров, к различным частям сообщения

Постфикс использует отдельные параметры конфигурации для каждой части сообщения, которую он знает. Параметры проверки - следующие (обратите внимание, что не все они задействованы в main.cf файле по умолчанию),

**header_checks**

Они обращаются к заголовку сообщения то есть ко всему от первой строки сообщения до первой чистой строки. Они также включают заголовки которые занимают несколько физических строк.

**body_checks**

Они обращаются к телу сообщения; анализатор полагает, что телом является все содержимое между заголовками.

*ПРЕДОСТЕРЕЖЕНИЕ, частое использование body_checks может очень сильно нагрузить центральный процессор, заметно замедляя вашу машину, потому что проверка тела просматривает каждую строку тела, и сравнивает ее с каждым регулярным выражением, которое Вы определяете в карте body_check$, Чтобы предотвратить чрезмерную перегрузку центрального процессора, Постфикс по умолчанию проверяет только первые 51 200 байтов тела сообщения. Вы можете увеличить этот предел с помощью параметра body_checks_ size_limit. Вы можете также можете разделить повышенную нагрузку, делегируя контроль содержимого различным приложениям на отдельной машине, используя возможность content_filter ,описанную в Главе 11.*

**mime_header_checks**

Они обращаются к MIME заголовкам в выше расположенных заголовках сообщения, к основным MIME заголовкам и MIME заголовкам во вложенном сообщении (RFC 822 заголовкам сообщения (см. Рис 6-3)).

**nested_header_checks**

Они обращаются к заголовкам сообщений вложенной электронной почты кроме MIME заголовков. Эти проверки работают только с заголовках вложенных rfc822 сообщений, кроме заголовков MIME, внесенные в mime_header_checks описанные выше.

Что особенного в этих параметрах?

Постфикс 2.x обращается с телом сообщения как с n частями тела, каждая часть отмечена MIME заголовком. Эта MIME обработка задействована по умолчанию, но Вы можете отменить ее использующий disable_mime_input_processing = yes в вашем main.cf файле.

MIME анализатор принимает решение для каждой строки, которую он читает: строка принадлежит заголовку или части тела сообщения? Постфикс применяет проверки, основываясь на этом решении. Если часть сообщения имеет почтовые заголовки (то есть если это - приложенное сообщение типа message/rfc822), то к заголовкам применяется nested_header_checks.

Все в пределах сегмента после этих вложенных заголовков проверяется с помощью body_checks, до предела, определенного параметром *body_checks_size_limit* . Например, если Вы имеете сообщение с пятью частями MIME по 100 КБ (или приложениями), Постфикс проверяет первые body_checks_size-limit байт каждой части. Постустановите использует mime_header_checks, чтобы оценить каждый MIME заголовок (начало каждой новой доли). Если есть почтовые заголовки после любого MIME заголовка, они оцениваются с помощью nested_header_checks в каждой части.

Рисунок 9-1 показывает какие проверки применяются к каждой строке сообщения.


Рис 9.1 Постфикс решает для каждой строки какую проверку применять

## Когда Постфикс, применяет Проверки ?

Клиент передает сообщение после успешного завершения начального SMTP сеанса. Таким образом, Постфикс производит *_checks после того, как smtpd_*_restrictions были обработаны. Рисунок 9-2 показывает, когда Постфикс демон cleanup производит проверки.

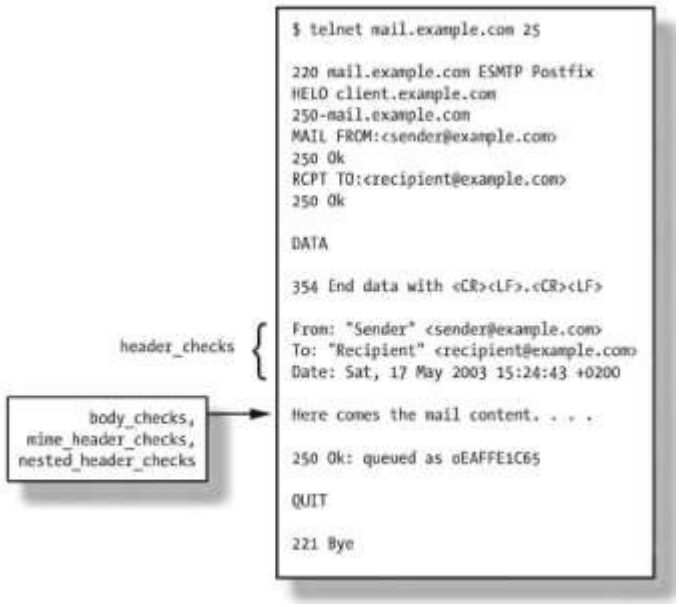



Рис 9-2 Проверки применяются после ограничений и только к содержимому сообщения

## Какие действия могут вызывать проверки?

Вы можете определить только одно действие в образце поиска. Постфикс в настоящее время поддерживает эти действия:

**REJECT (Отклонить) [дополнительный текст ...]**

Отклонение сообщения. Дополнительный текст будет послан назад клиенту, пытающемуся доставить сообщение. Постфикс также будет делать запись текста в лог файле.

**IGNORE (Игнорировать)**

Удаляет линию в сообщении, соответствующую образцу поиска.

**WARN (Предупредить) [optional text...]**

Заставляет Постфикс, написать уведомление в лог файле. Если есть дополнительный текст, Постфикс запишет и его. Постфикс доставит сообщение без каких либо изменений.

**HOLD (Удержать) [optional text...]**

Перемещает сообщение в очередь Hold, пока администратор не посмотрит его и не решит, что с ним делать. Постфикс занесет в лог соответствующую строку заголовка/тела с дополнительным текстом.

**DISCARD [optional text...]**

Сообщит почтовому клиенту, что сообщение успешно доставили, но тихо удаляет сообщение вместо того, чтобы передать его конечному предназначению. Если есть дополнительный текст, Постфикс поместит его вместе с текстом, которому с соответствует образец, в лог фаил.

**FILTER transport: next hop**

Посылает сообщение фильтру (службе, определенной в master.cf которая передает сообщение  другой системе обработки, типа вирусного сканера). Вы можете больше узнать о использовании внешних фильтров в Главе 11.

**REDIRECT user@domain**

Перенаправляет сообщение на адрес, указанный вместо оригинального получателя (ей), и это отвергает любое действие FILTER.

# Использование встроенных фильтров содержимого

Постфикс может исследовать содержимое сообщения с помощью таблиц образцов и действий, как описано в Главе 9. Эта глава покажет Вам, как создовать эти образцы и действия. Имейте в виду, что проверки используются для простого довольного фильтрования. Для решения более сложных задач, обратитесь к Главе 11.

Проверки ищут знаки в сообщении и могут также изменить сообщение. Имена параметров конфигурации, которые задействуют проверки заканчиваются на check, и используете ли вы header_checks, body_cheks, mime_header_checks, или nested_header_checks, все они следуют одной и той же схеме:

1. Постфикс, исследует сообщение строка за строкой сопоставляя с картой образцов, состоящей из регулярных выражений (regexps) или Perl регулярных выражений (PCREs).
2. Если строка соответствует регулярному выражению, Постфикс, выполняет действие, определенное для выражения *и* исследует следующую входящую строку.

*ПРЕДОСТЕРЕЖЕНИЕ, примеры этой главы сильно используют синтаксис продолжения линии, которые Постфикс предлогает для улучшения удобочитаемости на бумаге. А именно, строка начинающаяся с пробела продолжает предыдущую строку.*

## Проверка поддержки Постфиксом проверок

Постфикс поддерживает фильтрование заголовков или тела сообщения по умолчанию, но так как при этом может использоваться regexp и/или PCRE карты, Вы должны узнать, поддерживает ли Постфикс оба типа или только regexps. Для проверки типов карт, которые поддерживает ваш Постфикс используйте postconf-m, чтобы вывести все карты, которые поддерживает ваша система. Постфикс, в следующем примере, поддерживает и regexp и PCRE, среди нескольких других карт:

**$ postconf -m**

btree

cidr

environ

hash

nis

*pcre*

proxy

*regexp*

static

tcp

unix

Все системы должны поддержать regexp таблицы по умолчанию. Если ваша система имеет проблемы с производительностью, когда использует карты regexp-стиля (или еще хуже, если ваша система использует «детские» инструменты regexp), Вы можете установить PCRE библиотеки и заголовки и пересобрать, Постфикс с поддержкой PCRE.

## Сборка Постфикса с поддержкой карт PCRE

Для сборки Постфикса с поддержкой PCRE, Вы нуждаетесь в PCRE библиотеках и файлах заголовков. Вы можете получить их в пакете для вашей системе, или Вы можете загрузить исходные коды PCRE с http://www.pcre.org и собрать его вручную. Сконфигурируйте Постфикс с поддержкой PCRE, добавляя -DHAS_PCRE, и -I флаг предпроцессора для PCRE включенной непосредственно в CCARGS, и указав пути к PCRE библиотеке флагом AUXLIBS. Например, скажем, что, pcre.h находится в /usr/local/include, а pcre.a, находится в /usr/local/lib:

**$ CCARGS="-DHAS_PCRE -I/usr/local/include" **

**AUXLIBS="-t/usr/local/lib -lpcre" **

**make makefiles**

**$ make**

Примечание для соляриса нужно также: -R/usr/local/lib

## Безопасное применение фильтрования заголовков и тела сообщений

Регулярные выражения могут быть весьма сложными, и Вы могли бы закончить образцом, который не работает, который соответствует большему содержанию чем Вы хотите, или понимаете.
Чтобы помочь Вам отладить образцы, Постфикс предлагает действие **WARN (Предупредить)** . Если Вы используете это действие в правой части вашего образца, Постфикс, доставит сообщение, при совпадении выражения, и также поместит запись в лог файл. После того, как Вы уверены, что ваши образцы работают, Вы можете благополучно изменить действие **WARN (Предупредить)** на желаемое.

Безопасная процедура добавления фильтров следующая:

1. Добавление, Вашего образца к карте с **WARN** в качестве действия
2. Создание файла, который содержит выражение, которое попадающего под действие вашего фильтра.
3. Проверка того, что образец в карте соответствует тестовому образцу.
4. Задействование проверки в файле конфигурации Постфикса main.conf
5. Проверка с помощью реального письма

## Добавление регулярного выражения и установка действия WARN

Первым делом нужно добавить образец, который Вы хотите проверить к карте, и определять действие WARN которое будет выполнятся, когда содержание сообщения соответствует тестовому образцу. Пример, который мы будем

использовать в тестах этого раздела – образец фильтра заголовка «тема» , но Вы может использовать процедуру, описанную здесь для других заголовков и других *_checks параметров.

Добавьте образец фильтра в файл /etc/postfix/header_checks. Этот файл отвечает за карты для header_checks . Вот - пример:

**/^Subject: FWD: Look at pack from Microsoft/**

**WARN Unhelpful virus warning**

## Создание тестового образца

Все, что Вы должны сделать для того, чтобы создать тестовый образец - поместить соответствующее сообщение в файле типа /tmp/test_pattern. Напишите следующее для этого примера:

From: dingdong@example.com

Subject: FWD: Look at pack from Microsoft

blah blah

## Регулярное выражение соответствует тестовому образцу?

Проверьте ваш образец фильтра, и применив карту проверок к тестовому образцу, выполнив команду postmap, Например выполните эту команду:

**$ postmap - q - regexp:/etc/postfix/header_checks < /tmp/test_pattern**

Если все работает, команда должна печатать строку соответствующую тестовому образцу, подобно этому:

Subject: FWD: Look at pack from Microsoft WARN Unhelpful virus warning

Если не работает команда postmap ничего не выведет.

## Настройка проверки в файле конфигурации

Если все выглядит хорошо, Вы можете редактировать ваш main.cf файл, чтобы использовать содержимое файла header_checks, который Вы только создали и проверили:

header_checks = regexp:/etc/postfix/header_checks

Перезапустите Постфикс и пошлите письмо подобное тестовому.

## Проверка с помощью реальной почты

Чтобы проверить фильтр с помощью реальной почты, пошлите ваш ранний тестовый образец. Следующая команда сделает это:

$ /usr/sbin/sendmail recipient@example.com < /tmp/test_pattern

Теперь, исследуйте ваш лог файл , чтобы проверить, что Постфикс, зарегистрировал предупреждение для тестового образца. Вторая строка в следующей выдержке из лог файла - сообщение предупреждения:

Mar 30 17:17:52 mail postfix/pickup[2455]: 53CAB633B3: uid=7945

from=<sender@example.com >

Mar 30 17:17:52 mail postfix/cleanup[246l]: 53CAB633B3: warning: header Subject: FWD;

Look at pack from Microsoft from local; from=<sender@example.com

to=<recipient@example.com>: Unhelpful virus warning

Mar 30 17:17:52 mail postfix/cleanup[246l]: 53CAB633B3:

message- id=<2OO40330151752.53CAB633B3@fnail. example. com>

Mar 30 17:17:52 mail post-fix/qmgr[2456]; 53CAB633B3: from=<sendergexample.com>

size=346,

nrcpt=1(queue active)

После того, как Вы уверены, что ваш фильтр работftn, Вы можете благополучно изменить, действие WARN на действия, которые фактически выполняются, типа REJECT или DISCARD.

## Проверка Заголовков

Постфикс может исполнить разнообразие действий с header_checks, типа отклонения или удержания сообщений, удаление заголовков, или отказа переадресации, или фильтрования сообщений. Этот раздел обсуждает, использование этих действий.

### Отклонение Сообщений

Постфикс может отклонить сообщения, используя действие REJECT. Вы можете использовать это действие, чтобы блокировать сообщения, соответствующие образцу, типа тех, который содержат специфические части в заголовке ТЕМА.

Отклонение не позволяет сообщениям поступать в вашу систему, и поэтому держит их на растоянии, от требовательных к ресурсам антивируса, антиспама , или ( что возможно хуже) ваших пользователей. Мы рассмотрим несколько примеров. Этот образец отклоняет бесполезные предупреждения антивируса, произведенные ScanMail (который всегда предупреждает отправителя, даже если вирус фальсифицирует адрес отправителя):

/^Subject: ScanMail Message: To Sender, sensitive content -found and action/

REJECT Unhelpful virus warning

Если Вы хотите блокировать сообщения с неправильным заголовком: Undisclosed-recipients, Вы можете использовать следующий образец. Это соответствует ситуации когда в заголовке To встречается <undisclosed recipients> (со или без скобок, с или без "s" в конце). (Правильный заголовок Undisclosed recipients будет To: undisclosed-

recipients :;.)

/^To:.*<?Undisclosed Recipients?>?$/

REJECT Wrong undisclosed recipients header

Этот образец лучше всего описан его комментарием и сопровождающим сообщением:

#
# спам содержащий Тема: что нибудь 565876
#

/^Subject:.*[[:space:].]{5,}\(?#?[[:digit:]]{2,}\)?i./

REJECT More than 5 whitespaces and a number follow the Subject:

Мы никогда не видели To: ..<> в заголовках настоящих писем

/^To:.*<>/

REJECT To: <> in headers

Наконец, некоторые строки тем которые используются только для спама. Вы должны понять идею этих четырех образцов. (Использование различных номеров для каждого предупреждающего сообщения облегчает отладку ложных срабатываний.)

#

# Certain Subject lines are indicative of fraud spam.

#

| | |
|---|---|
| /^Subject:.*is NOT being SEEN/ | REJECT fraud spam #1 |
| /^Subject:.*URGENT BUSINESS RELATIONSHIP/ | REJECT fraud Spam #2 |
| /^Subject: .*Confidential Proposal/ | REJECT fraud spam #3 |
| /^Subject: SEX-FLATRATE/ | REJECT fraud spam #4 |

**Задержка доставки**

Постфикс может задержать доставку сообщений с помощью действия HOLD. Вы можете использовать это, чтобы поместить подозрительные сообщения "в ожидание" дальнейшего просмотра. Чтобы просмотреть сообщения, используйте команду postcat, для разрешения доставки сообщения , используйте postsuper - H. Если Вы хотите удалить сообщение из очереди Постфикса , используйте postsuper-d.
Вот - образец, который соответствует любому сообщению, содержащему заголовок Тема: начинающийся Subject: [list name]. Одно из его применений: удержание почты для всех пользователей от внутренних списков рассылки до конца рабочего дня - когда система не используется полностью:

/^Subject: \[listname\]/

HOLD

Вот - образец, который удерживает сообщения, в заголовках MIME которых используется одинокий перевод каретки . Большинство этих сообщений - вирусы и спам, с немногими исключениями, являющимися сообщениями от неправильных Windows инсталляций SquirrelMail:

/\r/

HOLD Lone CR in headers indicates virus or spam!

**Удаление заголовков**

Если Вы хотели бы удалить строки из заголовков, используйте действие IGNORE. Вы можете использовать это, чтобы скрыть информацию, записанную в заголовках, типа вида MUA который вы используете , или сократить заголовки, в которые ваши внутренние почтовые серверы, брандмауэры, или вирусные сканеры могли бы добавить информацию. Вот - образец, который удаляет заголовки, которые добавляет программа которой Постфикс, делегировал сообщения , чтобы сделать что- либо с ними посредством директивы content_filter (например, amavisd-new):

/^Received: from localhost/

IGNORE

Вот - другой, который удаляет заголовок Sender,некоторые версии Outlook ведут себя странно при ответе на сообщение, которое содержит этот заголовок:

/^Sender:/

IGNORE

**Отказ от Сообщений**

Постфикс может по тихому отказаться от сообщений, используя действие DISCARD. Например, Вы хотите что бы сообщения с некоторой строкой в поле Тема:, были удален без любого уведомления. Вот - глупый пример:

/^Subject:.*deadbeef/

DISCARD No dead meat!

Когда Постфикс бракует сообщение, он регистрирует действие как обычно. Например, Вы могли бы видеть в вашем почтовом логе:

Apr 9 23:14:28 mail postfix/cleanup[11580]: BB92B15C009: discard: header Subject: deadbeef from client.example.com[10.0.0.1]; from=<sender@example.com> to=<recipient@example.com> proto=ESMTP helo=<client.example.com>: No dead meat!

**Переадресация Сообщений**

Постфикс может перенаправить сообщения другому получателю, используя

действие REDIRECT, если заголовок и тело сообщения соответствуют образцу. Вот - пример, который переадресовывает неправильные сообщения (хотя мы не рекомендуем это):

```
/^subject:.*deadbeef/
REDIRECT bigbrotheriswatchingyou@example.com
```

В лог файле, переадресованное сообщение выглядит так:

```
Apr  9 23:20:38 mail postfix/smtpd[ll873]: 9305215C009:  client=client.example.com[10.0.0.1]
Apr  9 23:20:38 mail postfix/cleanup[ll865]:  9305215C009:  redirect:  header Subject: deadbeef
from client.example.com[10.0.0.l]j  from=<sender@example.com> to=<recipient@example.com>
proto=ESMTP helo=<client.example.com>:  bigbrotheriswatchingyou@example.com
 Apr  9 23:20:38 mail postfix/cleanup[ll865]:  9305215C009:
message-id=<20040409212038.GK3406@example.com>
Apr  9 23:20:38 mail postfix/qmgr[11857]:  9305215C009:  from=<sender@example.com>,
size=1111, nrcpt=1 (queue active)
Apr  9 23:21:08 mail postfix/smtp[n874]:  9305215C009:
to=<bigbrotheriswatchingyou@example.com>,
orig_to=<recipient@example.com>,  relay=none,  delay=30  status=deferred (connect to
example.com [192.0.34.166]:  Connection timed out)
```

**Фильтрация сообщений**

Постфикс может перенаправлять сообщения к фильтрам содержимого, (см. Главу 11) используя действие FILTER. Например, Вы можете переадресовать некоторые классы почты различным видам транспорта, основываясь на их заголовках.

Это действие отвергает значение content_filter назначеное в вашем main.cf файле и требует, чтобы Вы настроили различные сервесы *cleanup* один перед фильтром, и одним после фильтра. Header_checks и body_checks должены быть выключены во втором сервере *cleanup*, или Вы создадите петлю! (См. Главу 12 больше информации об этой проблеме (ищите опции no_header_ body_checks и receiue_override_options). Первый из следующих образцов не посылает сообщение фильтру, а второй делает это.

```
/^To:.*@example\.org/      FILTER nofilter:dummy
/^To:.*@example\.com/      FILTER virusfilter:dummy
```

*ПРИМЕЧАНИЕ Имеет в виду, что это - только пример. Вы не должны использовать его на рабочем сервере! Одно сообщение, предназначенное получателям в обеих доменам соответствовало бы первому регулярному выражению, и не будет, таким образом никогда фильтроваться (первое совпадение выигрывает); второе действие никогда бы не использовалось.*

Фильтрованные сообщения производят эти виды сообщений в лог файле почты:

Apr 9 23:34:12 mail post-fix/cleanup[i2543]: 2B97315C00D: -filter: header To: nofilter@example.com from client.example.com[10.0.0.1]; from=<sender@example.com> to=<nofilter@example.com> proto=ESMTP helo=<client.example.com>: nofilter:dummy

Apr 9 23:38:00 mail postfix/cleanup[12543]: 2299815C00E: filter: header To: virusfilter@example.com from client.example.com[10.0.0.1]; from=<sender@example.com> to=<virusfilter@example.com> proto=ESMTP helo=<client.example.com>: virusfilter:dummy

**Проверка MIME Заголовков**

MIME заголовки относятся к файлам, вложенным в сообщение. По умолчанию, карта header_checks используется для образцов просмотра MIME заголовка, если Вы не определяете отдельную карту и не укажите, Постфиксу, использовать ее параметром mime_header_c hecks.

*ПРИМЕЧАНИЕ имеет смысл определять отдельные карты, когда Вы хотите чтобы ваша карта mime_header_cbecks была маленькой насколько возможно, используя образцы MIME заголовков только если, Постфикс определит, что есть вложение в сообщении.*

Сначала Вы должны создать файл карты, чтобы хранить ваши образцы MIME заголовков. Скажем, Вы выбирали /etc/postfix/mime_header_checks, и он содержит следующие проверки:

```
# Files blocked by their suffix
/name=\"(.*)\.(386|bat|bin|chm|cmd|com|do|exe|hta|jse|lnk|msi|ole)\"$/
REJECT Unwanted type of attachment $l.$2
/name=\"(.*)\.(pif|reg|rm|scr|shb|shm|shs|sys|vbe|vbs|vxd|xl|xsl)\"$/
REJECT Unwanted type of attachment $1.$2
```

В этом примере, Постфикс, ищет MIME заголовки, которые содержат имя = " сопровождаемый произвольным числом знаков, с точкой (.). Большое приложение в круглых скобках, содержит несколько запрещенных расширений, отделенных вертикальной чертой (|). Регулярное выражение заканчивается знаком (\")> который также должен быть в конце строки ($).

Действие в правой части использует дополнительный текст позади REJECT . В этом примере, ссылки на два подмножества $1 (для первого подмножества- (.*) ) и $2 (для расширения файла).

Так, если кто - то пошлет вложение, названное image.pif, тогда строка MIME заголовка в сообщении будет походить на это:

```
filename="image.pif"
```

И сгенерированное сообщение об ошибке будет иметь вид:

Unwanted type of attachment image.pif

Потому, что $1 имеет значение image, а $2 имеет значение pif.

Теперь добавьте параметр mime_header_checks указывающий на карту образцов в Ваш main.cf фаил.

mime_header_checks = pcre:/etc/postfix/mime_header_checks

После перезагрузки Постфикса параметр mime_header_checks станет эффективным.

**Проверка заголовков во вложенных сообщениях**

Постфикс может применить отдельные действия к заголовкам, которые находятся в сообщениях, присоединенный к основному сообщению. По умолчанию, любой header_checks параметр будет заботиться о них, но если Вы хотите создать отдельную карту (чтобы экономить ресурсы процессора или создавать исключения), Вы можете использовать параметр nested_header_checks , чтобы определить отдельную карту. Подобно другим видам проверок, Вы должны создать отдельный файл карты, типа /etc/postfix/nested_header_checks, содержащий образцы для проверки. Вот - образец, который регистрирует ID сообщения во вложенном заголовке:

/^Message-Id;/ WARN Nested Message-Id;

Теперь добавте параметр nested_header_checks в файл main.cf.

После перезагрузки Постфикса, Вы можете найти записи в лог файле подобные этому:

Apr 14 13:17:55 mail postfix/cleanup[32397]: 59C3115C02A: warning; header Message-ID: <DIDL27HLlt4H87CA@example.com> from mgate22.so-net.ne.jp[210.139.254.169]; from=<> to=<recipient@example.com> proto=ESMTP helo=<mgate22.so-net.ne.jp>: Nested Message-Id:

*ОБРАТИТЕ ВНИМАНИЕ пример в этой секции не является действительно полезным, потому что ни мы, ни кто в списке рассылки не могли придумать реальный-сценарий. Если Вы нуждаетесь в nested_header_checks, Вы будете вероятно знать зачем вам это надо.*

**Проверка тела сообщения**

Просмотр частей тела сообщения полезен, когда Вы должны обнаружить образец в части тела письма, чтобы выполнить действие. Подобно другим проверкам, Вы исследуете содержание на данный образец, используя параметр body_checks в комбинации с картой, которая содержит образцы и соответствующие действия.

*ПРЕДОСТЕРЕЖЕНИЕ проверка тела сообщения относится ко всем сообщениям, и приходящим и исходящим, и всем отправителям и получателям. Поэтому, они также применяются к почте, посланной, abuse и postmaster.*
*Если Вы осуществляете проверку на спам, и сообщения людей, жалуются на*

*spam, который был возможно послан из ваших сетей, не могут быть доставлены на abuse и postmaster, если их жалобы содержат оригинал спама, который они получили. Вы не можете исключить проверку для некоторых пользователей в текущей версии Постфикса*

Начните, как обычно, с файла карты, типа /etc/postfix/body_checks. Вот - некоторые образцы и действия:

```
# Skip over base 64 encoded blocks. This saves lots of CPU cycles.
Expressions by Liviu Daia, amended by Victor Duchovni.
^~[[:alnum:]+/]{60,}\s*$~
```

Предыдущий образец соответствует, base64-en кодированным блокаем. Обратите внимание, что используется тильда (~) вместо обычного слэша(/), для разграничения регулярного выражения, это нужный, чтобы использовать слеш в пределах регулярного выражения.
Вот - некоторые образцы, которые содержат известные и уникальные образцы spam сообщений. Постфикс отклонит их:

```
# SPAM
/(AS SEEN ON NATIONAL TV|READ THIS E-MAIL TO THE END)/
REJECT Span №l
/We are shanghai longsun electrical alloy/
REJECT Chinese spammer from hell
/Do you want EVERYONE to know your business/
REJECT Spam #2
/(Zainab|San?ni) Abacha/
REJECT Nigeria spam
/MILITARY HEAD OF STATE IN NIGERIA/
REJECT Nigeria fraud spam
/antivirus\.5xx\,net/
REJECT Virus hoax (0190-dialer)
/MOSE CHUKWU/
REJECT Business fraud spam #1
/Ahmed Kabbah/
REJECT Business fraud spam #2
/Godwin Igbunu/
REJECT Business fraud spam #3
/I PRESUME THIS EMAIL WILL NOT BE A SURPRISE TO YOU/
REJECT Business fraud spam #4
```

/http:\/\/www\.al-opportunity4u\.com\/euro2/

REJECT Business fraud spam #5

/http:\/\/66.151.240.30\//

REJECT Spam of the worst kind

/http :\/\/members.tripod.com.br\/lev3irkd/

REJECT Spam of the worst kind II

Сообщения, содержащие следующие образцы будут отклонены; отправитель конверта получит сообщение, указывающее на базу данных Hoax.

# Hoaxes

/jdbgmgrN.exe/

REJECT Virus hoax!

/ready to dictate a war/

REJECT Hoax: http://www.tu-berlin.de/www/software/hoax/unicwash.shtml /inquiries@un\.org/

REJECT Hoax: http://www.tu-berlin.de/www/software/hoax/unicwash.shtml

/UNO is ready to receive signatures/

REJECT Hoax: http://www.tu-berlin.de/www/software/hoax/unicwash.shtml

/Third World War/

REJECT Hoax: http://www.tu-berlin.de/www/software/hoax/unicwash.shtml

Иногда Вы можете захотеть использовать блокировки в качестве немедленной меры, чтобы отклонить злонамеренные сообщения, если ваш сканер почтовых вирусов еще не знает вирус . Не забудьте удалить образец, как только ваш сканер узнает вирус:

## Virus

# Win32.Netsky.V

/The processing of this message can take a few minutes\.\.\./

REJECT Win32.Netsky.V

/Converting message. Please wait\.\.\./

REJECT Win32.Netsky.V

/Please wait while loading failed message\.\.\./

REJECT Win32.Netsky.V

/Please wait while converting the message\.\.\./

REJECT Win32.Netsky.V

После того, как Вы создали файл карты, добавьте body_checks параметр в ваш main.cf фаил:

body_checks = regexp:/etc/postfix/body_checks

Как и прежде перезагрузите Постфикс, чтобы активировать проверку тела сообщений.

# КАК РАБОТАЮТ ВНЕШНИЕ ФИЛЬТРЫ СОДЕРЖИМОГО

Будьте либеральны, с тем что Вы принимаете,
и консервативны в том что Вы посылаете.
– Ян Постал, пионер Интернета (194 3-1998)
–

Встроенные фильтры, описанные в предыдущих главах предназначены, для решения простых задач; более сложное фильтрование должно быть делегировано внешнему программному обеспечению.
Эта глава выделяет процесс делегирования. Вы увидите, как сформировать архитектуру Постфикс демонов, чтобы послать сообщения внешним механизмам фильтрования и как позволить им повторно поступать в Постфикс систему для конечной доставки, после того как они были успешно фильтрованы.
Внешние фильтры содержимого вступают там, где встроенные фильтры заголовоков и фильтры тела сообщения не справляются; мало того, что они позволяют внешним приложениям осматривать и отклонять сообщения, но они также позволяют приложениям изменять содержание сообщения. Вот - некоторые типовые задачи для фильтров:

- Добавление примечаний
- Сканирование на вирусы и черви
- Обнаружение спама
- Архивация почты

Постустановите имеет два фильтрующих механизма, названные content_filter и smtpd_proxy_filter, которые похожи по духу, но отличаются по их способностям и способу, которым они обрабатывают содержание. Таблица 11-1 содержит список различия между двумя типами фильтров.

Таблица 11-1 Различия фильтров

| Название фильтра | Транспорты | Поведение отклонение |
|---|---|---|
| content_filter | smtp, lmtp, pipe | отклонение после помешения в очередь |
| smtpd_proxy_filter | smtp | отклонение до помещения в очередь |

Эта глава подробно объясняет эти различия и поможет Вам решить, какой фильтр, лучше всего соответствует вашей ситуации.

**Когда лучший момент для фильтрования содержимого ?**

RFC стандарты Интернета говорят, что сервер почты должен решить, принимает ли он или отклоняет сообщение не позже чем на стадии DATA SMTP диалога, К сожалению, это оставляет немного времени для почтового сервера, чтобы просмотреть содержание сообщения, потому что почтовые клиенты осуществляют относительно короткий перерыв, чтобы обезопасить себя при работой со сбоями, при отправке сообщения серверу. Например, SMTP перерыв клиента Постфикса определен параметром smtp_data_done_timeout , который является очень терпимым и по умолчанию составляет 600 с

Если почтовый сервер заканчивает просмотр содержания прежде, чем клиент сталкивается с перерывом, все работает прекрасно, потому что сервер может уведомить клиента о принятом решении . Однако, если сервер слишком медленен, клиент разрывает связь и пробует передать сообщение позже, и возможно , что следующая попытка будет столь же неудачна. Фильтр содержимого Постфикса избегает, прерывания, обрабатывая просмотр содержимого по-другому:

1. Почтовый клиент посылает содержание сообщение в течение стадии DATA.
2. Постфикс сервер принимает и помещает в очереди сообщение. Клиент предполагает, что передача была успешна.
3. Менеджер очереди осматривает почту и намечает доставку согласно content_filter входу.
4. Постфикс передает сообщение внешнему приложению.
5. Внешнее приложение берет ответственность за доставку сообщения. Внешнее приложение может сделать любое следующее действие с сообщением:
   - Принять сообщение и возвратить его, Постфиксу для доставки.
   - Принять сообщение и передать его другому приложению или серверу.
   - Задержать или возвратить сообщение.

Второй фильтр (smtpd_proxy_filter) обращается с почтой по-другому:

1. Почтовый клиент посылает содержание сообщение в течение стадии DATA.
2. Постфикс smtpd демон демона делегирует SMTP команды и содержание сообщения внешнему приложению.
3. Внешнее приложение посылает SMTP ответы назад Постфикс демону smtpd, и smtpd затем передает их почтовому клиенту.

Этот фильтр может иметь проблемы с перерывами и не может работать параллельно с другими почтовыми клиентами. Это происходит, потому что smtpdproxy _filter не имеет никакого механизма очереди, чтобы наметить фильтрование содержимого. Без механизма очереди, внешнее приложение должно начать работу немедленно с каждым сообщением, которые получает Постфикс. В результате, Вы можете иметь серьезное замедление, если внешнее приложение не сможет обрабатывать сообщения с такой же скоростью , с какой они поступают. Это, вероятно, будет проблемой при сканировании на спам или вирусы,что часто отнимает много времени на распаковку и расшифровку вложения.

Оба подхода имеют недостатки:

- content_filter генерирует дополнительный трпфик, потому что Постфикс первоначально принимает сообщения перед обработкой. Он может отклонить сообщение позже, если фильтруующее приложени решает отклонить сообщение.
- smtpdproxy_filter отклоняет нежелательное содержание вначале, но он может работать не достаточно быстро.

**Фильтры и переписывание адреса**

При переписывании адресов в заголовках почты, Вы должны думать о том, где применить фильтры. В частности Вы должны решить, делать ли, Постфиксу, замену адреса (например, из virtual_alias_maps) до или после фильтрования. Если Вы хотите переписывать адреса перед фильтрованием, Вы рискуете использовать внутренние адреса для отвергнутых сообщений и предупреждений. Например, предупреждение, вызванное сообщением к moe_helden@example.com могло бы

быть послоно с адресом типа mh123@mailbox.example.com.
Поэтому, с нашей точки зрения, Вы должны сделать так, чтобы Постфикс, переписывал адреса (посредством virtual_alias_maps, или canonical_maps) *после* возвращения почты назад в Постфикс очередь для конечной доставки'. Это позволяет внешним приложениям (типа сканера вирусов) видеть первоначальных получателей и генерировать соответствующие предупреждения прежде, чем Постфикс, перепишет адреса.
Существует два способа убрать замену адреса (virtual_alias, canonical_maps , masquerading, и так далее) перед фильтрованием. Один путь состоит в том, чтобы установить следующую опцию в main.cf:

receive_override_options = no_address_mappings

Вы можете также выключить переписывание адреса в master.cf только для демона, который принимает почту из сети (им является обычно smtpd):

smtp inet n - n - - smtpd

-o content_filter=foo:[127.0.0.1]:54321

**-o receive_override_options=no_address_mappings**

## content_filter: Сначала организация очереди, фильтрование позже

Чтобы настроить почтовый сервер с content_filter механизмом, Вы обычно нуждаетесь в двух копиях smtpd (см. рисунок 11-1), первый smtpd демон принимает не фильтрованные сообщения и использует content_filter, чтобы делегировать сообщения к внешнему фильтрующему приложению. Второй smtpd демон слушает порты внешнего приложения, чтобы сообщения могли повторно войти в Постфикс систему очереди для дальнейшей обработки.

*ПРЕДОСТЕРЕЖЕНИЕ не конфигурируйте вторую копию, для запуска content_filter, это работа первого демона. Это создало бы бесконечную петлю, где Постфикс, послал бы сообщение приложению фильтрования, и сообщение возвратится в Постфикс очередь в том же самом месте как и прежде*

Вот, как это работает:

1. Smtpd, настроенный с content_filter передаст сообщение менеджеру очереди.
2. Менеджер очереди передает сообщение или smtp, lmtp, или pipe, чтобы доставить его внешнему фильтрующему.

*ОБРАТИТЕ ВНИМАНИЕ, что рисунок 11-1 показывает один из возможных трех сценариев, где qmgr передает почту smtp.*

3. Внешнее фильтрующее приложение берет под свой контроль сообщение и обрабатывает его.
4. Если фильтрующая программа повторно вводит сообщение назад в, Постфикс систему, она соединяется с Постфикс smtpd демоном, настроенным без использования content_filter.

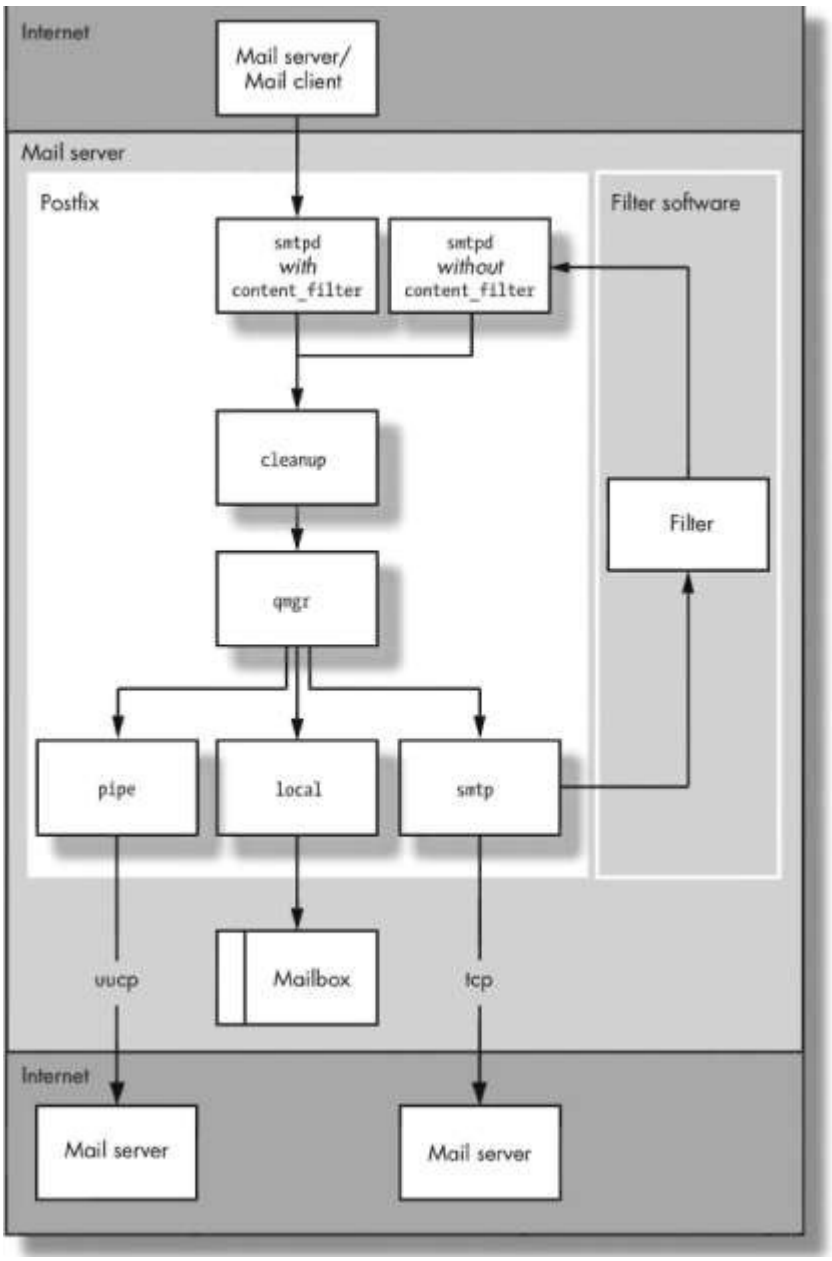



Рис 1-11 процесс доставки при использовании content_filter

5. Второй smtpd демон вручит сообщению менеджеру очереди,
6. Постфикс, доставляет сообщение локально или передаст его другому почтовому серверу.

В дополнение к сообщению, Постфикс, может послать дополнительную информацию внешнему фильтрующему приложению, чтобы помочь тому. Точная информация зависит от демона (lmtp, smtp, или pipe), которые Постфикс, вызывает, чтобы послать сообщение приложению.

**Демоны делегации фильтру**

При использовании content_filter, Вы имеете три основных демона в вашем распоряжении чтобы делегировать почту к внешнему фильтрующему преложению. Демоны отличаются по тому, что они могут сделать и передать. Вот - краткий обзор:

**pipe**

Демон pipe посылает сообщения сценариям и другим выполняемым программам. Они могут вызвать почти любое воображаемое действие, от архивирования сообщений до выполнению других видов автоматизированной работы над содержанием сообщения, типа обнаружения вирусов.

Широкий диапазон задач, которые эти программы исполняют, заставляет передавать несколько аргументов и флагов фильтрующей программе наряду с сообщением. Вы можете прочитать об этих аргументах в pipe(8) ман странице .

**smtp**

Вы можете использовать Постфикс smtp демон, чтобы передать сообщение к фильтрующему приложению с помощью SMTP или ESMTP (например, другому MTA). Информация, которую Вы можете послать наряду с сообщением, ограничена в соответствии с протоколом; smtp(8) ман расскажет подробнее деталях.

**lmtp**

Постфикс lmtp демон также доступен, для посылки сообщения программам фильтра через LMTP протокол. Как и с SMTP, LMTP протокол ограничивает количество дополнительной информации, которую Вы можете передать наряду с сообщением, Вы можете прочитать об этом в lmtp (8) мане.

*ОБРАТИТЕ ВНИМАНИЕ В отличие от SMTP клиента Постфикса, который в настоящее время не осуществляет Уведомление Статуса Доставки (DSN), чтобы производить отдельные уведомления, LMTP протокол позволяет серверу, послать - сообщения о статусе сообщения по получателям, (то есть, сообщения о том было ли сообщение отклонено или принято для каждого получателя). Это позволяет избежать путаницы с уведомления о статуса для многократных получателей когда, сообщение доходит для некоторых получателей, но не для других.*

## Основы конфигурирования content_filter

Чтобы посылать сообщения внешней фильтрующей программе, используя content_filter, Вы должны изменить поведение всех демонов отвечающих за входящую почту. В частности Вы должны сделать следующее:

1. Определить транспорт для content_filter в вашем main.cf файле.
2. Сконфигурировать транспорт в вашем master.cf файле.
3. Сформировать дополнительный путь возврата в master.cf, если Вы хотите посылать сообщение назад в Постфикс очередь после фильтрования.

## Определение транспорта

Укажите Постфиксу , что он должен передать сообщения внешнему приложению, используя content_filter в вашем main.cf файле. Вы должны сказать, Постфиксу, передавать все сообщения (в дальнейшем созданной), Постфикс службе, которая ожидает, для передачи сообщения к фильтрующему приложению. Например, следующая строка указывает, Постфиксу, послать сообщения транспорту , названному foo, через порт 54321 на localhost (127.0.0.1). Помните, что квадратные

скобки указывают, Постфиксу не выполнять запрос MX записи для 127.0.0.1:

content_filter = foo:[l27.0.0.l]:54321

**Настройка транспорта**

Затем Вы должны сформировать транспортную службу, которую Вы только создали в main.cf. Файл конфигурации транспортной службы - master.cf, потому что - демон master, должен знать обо всех доступных службах. Для новой транспортной службы, Вы должны дать демону master следующую информацию:

1. Имя службы.
2. Имя Постфикс программы демона, которая выполнит транспортировку.
3. Опции и другая информация, необходимая программе для выполнения работы

Вот пример построения транспорта foo:

```
# service type  private unpriv chroot wakeup maxproc command
#                (yes)   (yes)  (yes)  (never)  (100)
#
...
foo             unix    -      -      n        -       2       smtp
-o smtp_data_done_timeout=1200s
-o disable_dns_lookups=yes
...
```

Строка, которая начинается с foo – собственно, конфигурация транспорта, содержащая восемь колонок. Первая колонка должна соответствовать имени транспорта, который Вы определили в main.cf. Колонка comand содержит команду, которая посылает сообщение фильтрующей команде (здесь, это – Постфикс SMTP клиент). Следующие две строки - опции команды.

*ПРЕДОСТЕРИЖЕНИЕ Синтаксис опций команды: При внесении в список дополнительных параметров команды, добавьте пробел к началу каждой новой строки, которая содержит параметры, потому что строка, которая начинается с , продолжает логическую строку. Однако, Вы должны урезать пробелы между параметрами и значениями (например между smtp_data_done_time_out и 1200); иначе демон master не будет признавать ваши дополнения.*

Пока, все хорошо - Вы можете передавать сообщения внешним приложениям Однако, если приложение должно возвратить сообщение Постфиксу, Вы должны сформировать путь ввода.

**Настройка дополнительного пути ввода**

Путь возврата просто локальный Постфикс метод ввода (типа SMTP, LMTP, или локального ввода через sendmail), который не использует content_filter. Обычно это другая копия smtpd демона запущенная со специальными опциями, чтобы отвергнуть глобальный набор параметров в main.cf файле. Например, если бы Вы хотели, чтобы дополнительный smtpd демон пути ввода слушал на 10025 порту, Вы

могли бы поместить следующее в ваш master.cf file:

```
#  service type            private unpriv chroot wakeup maxproc command
#                           (yes)    (yes)   (yes)   (never)    (100)
# ==========================================================================
...
127.0.0.1:10025    inet n          -          n          -          -          smtpd
        -o content_filter=
        -o receive_override_options=no_unknown_recipient_checks
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks=127.0.0.0/8
...
```

Заметьте, что транспортный тип - inet на сей раз, для транспортаИнтернета ( в предыдущем примере тип транспорта был сокет Unix).

Вы можете, сначала, определить сервисы Интернета, как host:port, не определяя транспорт в main.cf файле (host может быть именем хоста или IP адресом, определенным в /etc/hosts, тогда как порт может быть числом или именем сервиса, определенным в файле /etc/services).
Вы можете опустить хост:, но это делает службу доступной на всех сетевых интерфейсах определенных в inet_interfaces. Чтобы минимизировать риск создания открытого релея с вашим путем ввода, Вы должны ограничить слушающие интерфейсы сети только теми, в которых нуждаетесь, в этом случае Вы нуждаетесь только localhost/127.0.0.1.
Вот дополнительные командные опции:

- Явно пустая опция content_filter отменяет транспорт фильтра в main.cf файле, чтобы Вы не получили бесконечную петлю фильтрования.
- Установка receive_override_options отменяет проверку получателей, для local_recipient_maps, и relay_recipient_maps -потому-что эти проверки были уже выполнены smtpd демоном, который принял почту из Интернета, и нет никакой необходимости выполнять их во второй раз.
- Два последних параметра работают вместе, сначала позволяя почту только от параметра mynetworks, и затем явно устанавливая mynetworks параметр равным 127.0.0.0/8. Это - дополнительная гарантия против внешних хостов, пытающихся получить доступ к вашему пути ввода как к открытому релею.

## smtpd_proxy_filter: сначала фильтрование, затем организация очереди

Чтобы использовать smtpd_proxy_filter механизм, Вы должны изменить существующий smtpd демон (before-filter smtpd) к для организации прокси соединения от почтовых клиентов с внешней фильтрующей программой (см. Рис 11-2).

*ОБРАТИТЕ ВНИМАНИЕ, что Постфикс smtpd демон защищает внешнее приложение, от потенциально опасных команд типа конвейерной обработки, длинных аргументов, и странных знаков, которые могут поступить от smtp связи.*

В зависимости от фильтрующей программы и целей в которых вы ее используете, Вам, вероятно, также придется создать второй smtpd  демон (после - фильтровый smtpd), который принимает сообщения, посланных назад внешним фильтрующим приложением.
Вот, как это работает:

1. Дофильтровый smtpd демон соединяется с внешним приложением.
2. Smtpd демон передает  SMTP команды и данные  внешнему приложению.
3. Внешние фильтрующие приложение держит связь открытой, поскольку оно обрабатывает содержание сообщения.
4. Если фильтрующее приложение принимает сообщение, оно может ввести его через  после -фильтровый smtpd демон или послать его другому приложению
5. После принятия решения о принятии или отклонении сообщения, внешнее фильтрующее приложение посылает SMTP ответы (типа 250 OK, или 554 REJECT)  до фильтровому smtpd демону.
6. До фильтровый smtpd демон передает эти ответы почтовому клиенту.

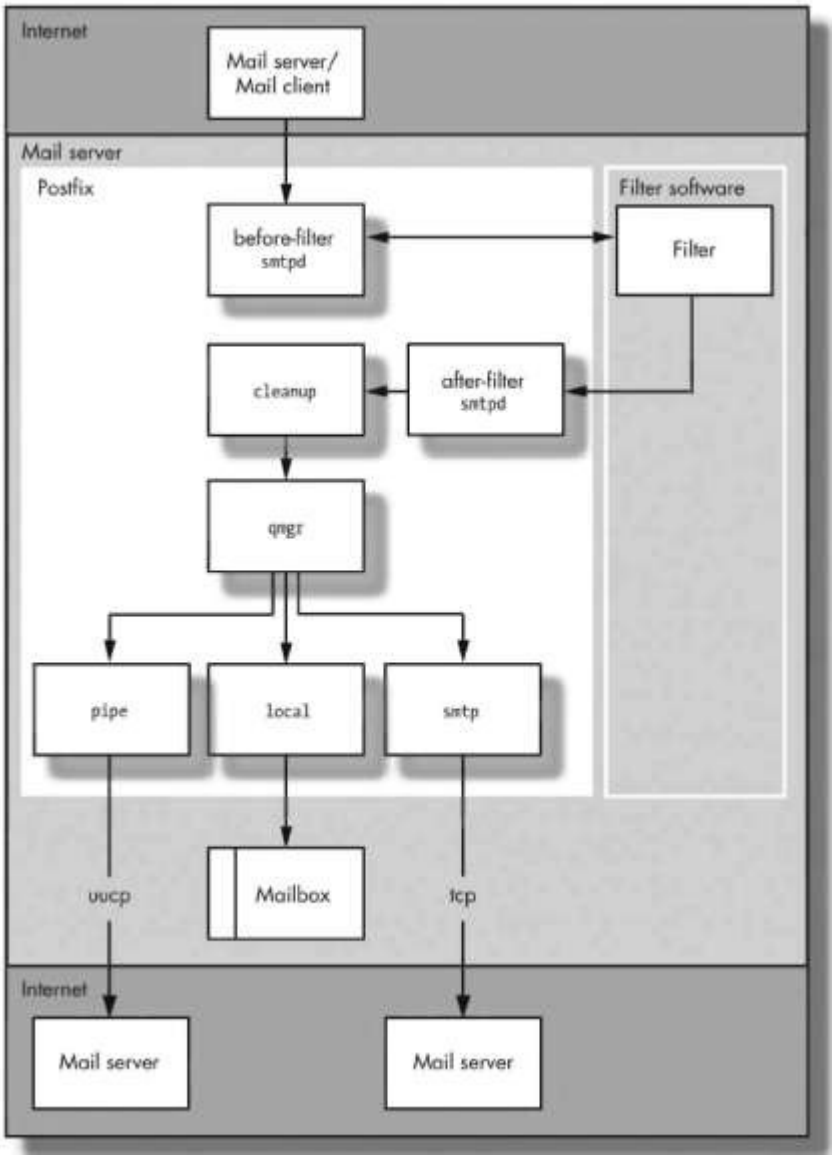



Рис 11-2 процесс доставки при прохождении через SMTP proxy

## Рассмотрение прокси фильтров

При работе с smtpd_proxy-filter, держите следующие в памяти следующие пункты:

### ESMTP связь

При посылке сообщения в фильтр, Постфикс, использует ESMTP, но о не использует конвейерную обработку команд . Постфикс smtpd генерирует свои собственные команды EHLO, XFORWARD (чтобы регистрировать IP адрес удаленного клиента вместо localhost [127.0.0.1]), DATA, и QUIT. Иначе, Постфикс только передает неизмененные копии команд MAIL FROM и RCPT TO , которые до фильтровый smtpd демон получил от удаленного почтового клиента.

### Требования внешних приложений

Внешние программы (которые должны понимать SMTP) должны получать те же команды MAIL FROM и RCPT TO что и Постфикс smtp демон.

Повторный ввод содержимого

Фильтрущее приложение , как ожидается, передаст неизмененные SMTP, команды полученные от до фильтрового smtpd демона после фильтровому Постфикс smtpd демону (который обычно слушает на нестандартном порту для ввода содержимого по пути, который не работает с тем же самым фильтром; это подобно случаю с content_filter механизмом, обсужденным ранее в этой главе),

### Отклонение содержания

Если фильтр отклоняет содержание, он должен послать отрицательный SMTP ответ (5xx, код) назад до фильтровому Постфикс smtpd демону и затем прерывает, связь с после фильтровым Постфикс smtpd демоном, не заканчивая SMTP связь с после фильтровым , Постфикс smtpd демоном. Иначе, после фильтровый smtpd демон может случайно доставить сообщение.

## Основы конфигурирования smtpd_proxy_filter

Чтобы посылать сообщения внешнему фильтру, используя smtpd_proxy_filter , Вы должны изменить поведение smtpd демона. Следующие два шага необходимы:

1. Изменить существующий smtpd. Мы будем этого демона как до фильтровый smtpd демон.
2. Сформировать дополнительную копию smtpd, чтобы повторно ввести почту назад в Постфикс очередь; это – после фильтровый smtpd.

Изменение Существующего smtpd (До фильтровый smtpd)

Для осуществления smtpd прокси связи к фильтрующему приложению, добавьте параметр smtpd_proxy_filter в конец smtpd службы в вашем master.cf файлт. Вы должны обеспечить IP адрес или FQDN и порт прокси.

Вот - пример который использующий порт 10024 на localhost:

```
#==========================================================================
# service     type     private unpriv chroot wakeup maxproc command
#                        (yes)     (yes)   (yes)   (never)   (100)
# ==========================================================================
smtp          inet       n          -          n          -          20          smtpd
-o smtpd_proxy_filter=localhost:10024
```

**Настройка дополнительной smtpd копии повторного ввода (После фильтровый smtpd демон)**

Чтобы создать другую копию smtpd, который принимает фильтрованные сообщения на localhost, Вы должен добавить другую строку к вашему master.cf файлу. Она будет подобна smtpd по умолчанию, но будет слушать на другом порту и не должен иметь опции smtpd_proxy_filter как до фильтровый smtpd. Вот - пример для после фильтрового smtpd демона, который слушает 10025 порту:

```
#==========================================================================
# service          type      private unpriv chroot wakeup maxproc command
#                             (yes)  (yes)  (yes)     (never)   (100)
# ==========================================================================
127.0.0.1:10025  inet       n       -          n          -          -          smtpd
-o smtpd_authorized_xforward_hosts=127.0.0.0/8
-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 receive_override_options=no_unknown_recipient_checks
-o content filter=
```

# ИСПОЛЬЗОВАНИЕ ВНЕШНИХ ФИЛЬТРОВ СОДЕРЖИМОГО

Каждый инструмент имеет свою цель. Вообразите, что Вы пробуете придумать молоток, который может также использоваться для полировки. Наиболее вероятно, Вы получите  плохой молотки плохой полировщик. Это – причина, по которой Постфикс, не выполняет фильтрование спама, архивирование почты, или очистки почты. Вместо этого он дает Вам возможность подключит лучший внешний фильтр, который доступен. В предыдущих главах, Вы видели, что Постфикс предлагает немного различные подходы для фильтрования почты, которые отличаются по тому, когда они обрабатывают входящие сообщения. Эта глава посвящена практическому применению этих подходов.
В частности Вы увидеть, как добавить комментарии при передачи сообщения сценарию и как выполнить поиск вирусов используя или content_filter или smtpd_proxy_filter, чтобы отослать их  amavisd-new.

### Добавление комментариев к Сообщениям с помощью сценариев

Среди бесчисленных вещей, которые Вы можете делать с помощью сценариев content_filter - добавлять комментарии ко всем исходящим сообщениям. Следующий пример использует маленькую программу alterMIME,  которая используется, чтобы изменить MIME кодируемую почту, в сценарии, для добавления комментариев к каждому сообщению  от внутренних клиентов. Рисунок 12-1 показывает Вам, как alterMIME интегрирован в процесс транспортировки сообщения.

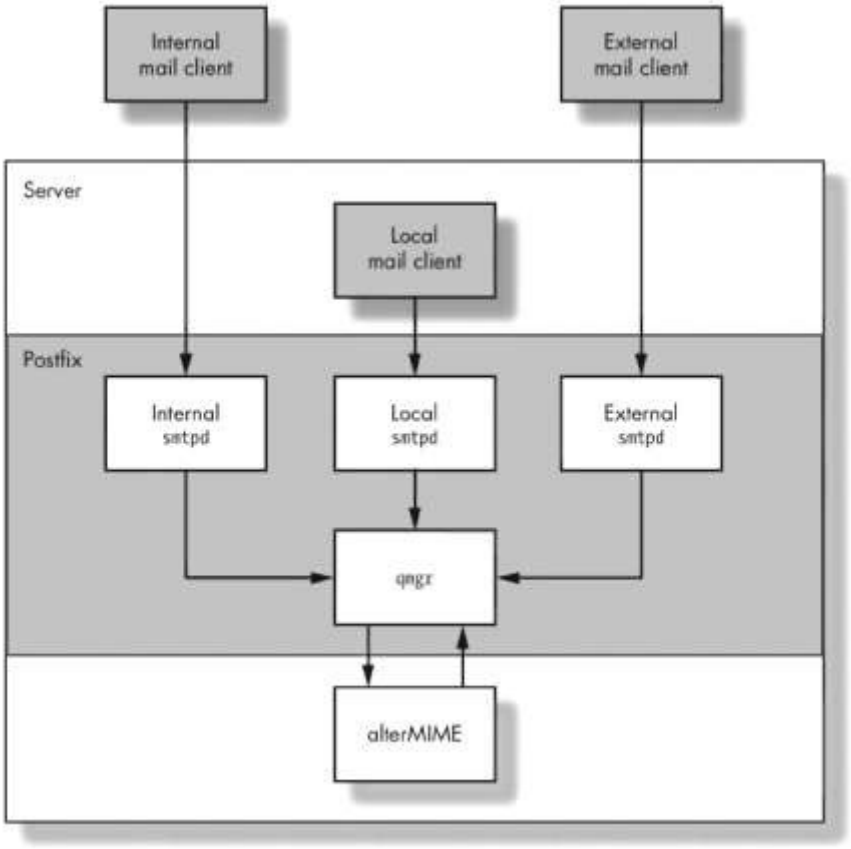



Рисунок 12-1: интеграция AlterMIME в Постфикс

Чтобы добавлять комментарии к исходящим  сообщениям, не касаясь входящих и

локальных сообщений, Вы должны отделить транспорты для каждого направления. Скажем, Ваш почтовый сервер имеет отдельные сетевые интерфейсы для ваших внутренних и внешних сетей. Это означает, что Вы должны создать три отдельных копии smtpd и связать их с localhost, внутренним, и внешним сетевыми интерфейсами. Следующий пример показывает Вам, как будет происходить транспортировка сообщения из вашей внутренней сети до удаленного предназначения , если Вы создали отдельные копии smtpd для различных сетевых интерфейсов.

1. Когда сообщение покидает вашу сеть, почтовый клиент соединяется с демоном smtpd, слушающим на внутреннем итерфейсе.
2. Этот внутренний stmtpd принимает сообщение и посылает это менеджеру очереди qmgr.
3. qmgr посылает сообщение службе content_filter.
4. Служба content_filter использует демон pipe, чтобы передать сообщение сценарию.
5. Сценарий добавляет комментарий.
6. Сценарий повторно вводит сообщение демону stmtpd, слушающему на локальном сетевом интерфейсе.
7. Локальный smtpd посылает повторно введенное сообщение qmgr.
8. qmgr посылает сообщение smtpd, который передает его в Интернет.

Прежде, чем Вы сформируете транспорт, Вы должны создать сценарий, который вызывает alterMIME из, Постфикса.

**Установка alterMIME и создание сценария фильтра**

Сценарий будет запускать alterMIME (<http://www.pldaniels.com/altermime>), чтобы изменить исходящие сообщение. Если Вы не имеете alterMIME (и Вы не имеете бинарного пакета для вашей операционной системы), загрузите его, распакуйте, добавьте в дерево исходных кодов Вашей системы, и выполните команды: make и make install. Эти команды должны скомпилировать и поместить исполняемые файлы alterMIME в /usr/local/bin/altermime.

Создание окружающей среды alterMIME

Вы должны запускать alterMIME от непривилегированного пользователя системы. Например, если Вы хотели бы использовать имя пользователя filter на вашей машине, Вы могли бы создать пользователя с помощью этих команд:

```
# groupadd filter
# useradd -d /var/spool/aItermine -C filter altermime
```

Создание папок сценариев

Не очень хорошая идея, загромождать директорию etc/postfix различными сценариями. Создайте отдельную поддиректорию как суперпользователь, чтобы хранить ваши сценарии, и сделайте поддиректорию доступной только для фильтра и пользователя root.
Например, следующая последовательность команд создает директорию с правильными разрешениями и владельцами:

```
# mkdir /etc/postfix/filter
# chown root /etc/postfix/filter
# chgrp filter /etc/postfix/filter
# chmod 770 /etc/postfix/filter
```

### Создание сценария

Следующий сценарий, названный /etc/postfix/filter/adddisclaimer.sh, вызывает alterMIME к входящим сообщениям от, Постфикса (посланные демоном pipe). Программа alterMIME добавляет коментарии к сообщению и повторно вводит его назад в очередь Постфикс. AlterMIME требует, локацию для временных файлов, и не может работать в stdn.

```
#!/bin/sh
# System dependent settings ALTERMIME=/usr/local/bin/altermime
ALTERMIME_DIR=/var/spool/altermime
SENDMAIL=/usr/sbin/sendmail
# Exit codes of commands invoked by Postfix are expected
# to follow the conventions defined in <sysexits.h>.
TEMPFAIL=75
UNAVAILABLE=69
# Change in to alterMIME's working directory and
# notify Postfix if 'cd' fails.
cd $ALTERMIME_DIR || { echo $ALTERMIME_DIR does not exist; exit $TEMPFAIL; }
# Clean up when done or when aborting.
trap "rm -f in.$$" 0 1 2 3 15
# Write mail to a temporary file
# Notify Postfix if this fails
cat >in.$$ || { echo Cannot write to $ALTERMIME_DIR; exit $TEMPFAIL; }
# Call alterMIME, hand over the message and
# tell alterMIME what to do with it
$ALTERMIME --input=in.$$ \
                --disclaimer=/etc/postfix/disclaimer.txt \
                --disclaimer-html=/etc/postfix/disclaimer.txt \ -
                --xheader="X-Copyrighted-Material: Please visit http: \
                www.example.com/message_disclaimer.html" || \
```

```
        { echo Message content rejected; exit $UNAVAILABLE; }
# Call sendmail to reinject the message into Postfix
 $SENDMAIL "$@" <in.$$
# Use sendmail's EXIT STATUS to tell Postfix
# how things went.
exit $?
```

После создания сценария, дайте, права на запись только пользовотелю root, но дайте разрешение на исполнение пользователю filter:

```
# chown root add_disclaimer.sh
# chgrp filter adddisclaimer.sh
# chmod 750 add_disclaimer.sh
```

Конечно, теперь Вы должны создать комментарий, на который ссылается скрипт.

**Создание комментария**

Если Вы уже имеете комментарий вставьте текст в файл /etc/postfix/filter/ disclaimer.txt.
Если Вы все еще ищете правильные комментарии, Вы можете посетить emaildisclaimers.com, сайт просвещенный комментариям и законам связанным с электронной почтой. Этот пример только использует следующий фиктивный текст из ( <http: // www.lipsum.com>):

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nam commodo lobortis magna. Ouisque neque. Etiam aliquam. Nulla tempor vestibulum.

После размещения текта, разрешите только группе filter читать ваш комментарий:

```
# chgrp filter disclaimer.txt
# chmod 640 disclaimer.txt
```

Это сценарий фильтра. Теперь Вы должны сконфигурировать, Постфикс, чтобы использовать сценарий.

**Конфигурирование Постфикса для сценария комментариев**

Конфигурирование Постфикса, для вызова сценария состоит из двух шагов:

1. Определить content_filter параметр для нужного smtpd демона в файле master.cf
2. Определить транспорта в файле master.cf.

**Определение параметра content_filter**

Как объяснено в Главе 11, сейчас вы должны были бы добавить content_filter параметр в main.cf и назначили бы имя транспорта. Однако, это определило бы content_filter глобально, и фильтр относился бы ко всем процессам отвечающим за входящую почту . Вы не хотите, чтобы это случилось в этом специфическом примере, так как Вы  хотите чтобы фильтр применялся только к сообщениям поступающим от внутреннего сетевого интерфейса.

Для связывания фильтра с сообщениями, поступающим только от внутренней сети, Вы должны добавить параметр content_filter только к единственному smtpd демону в файле master.cf. Адрес внутреннего интерфейса в следующем примере master.cf - 172.16.0.1:

```
127.0.0.1 : smtp        inet    n    -    n    -    -    smtpd    1
172.16.0.1 : smtp       inet    n    -    n    -    -    smtpd    2
-o content_filter=disclaimer:
192.0.34.166 : smtp     inet    n    -    n    -    -    smtpd    3
```

**1** Это – локальный демон smtpd.
**2** Это - демон smtpd, который слушает на внутреннем сетевом интерфейсе. ©
**3** Это - демон smtpd, который слушает внешнюю сеть.

Заметьте, что название транспорта фильтра -  disclaimer; это - не имя сценария. Вы определите этот транспорт в следующем разделе.

**Определение транспорта**

Теперь Вы должны определить транспорт disclaimer в master.cf файле. Создайте копию. транспорта pipe, который вызывает сценарий add_disclaimer.sh. Вот, как Вы сделали бы это в случае с сценарием, показанным ранее в этой главе:

```
disclaimer  unix  -       n       n       -       -       pipe
flags=Rq user=filter argv=/etc/'postfix/filter/add_disclaimer.sh -f ${sender} -- ${recipient}
```

Это определение запускает демон pipe в качестве пользователя фильтра, вызывая add_disclaimer.sh при получении сообщения. Данное определение  также передает адреса отправителя конверта и получателя конверта сценарию. Флаг R создает заголовок Return-path сообщения с адресом отправителя конверта, и q флаг указывает пробелы и другие специальные знаки в командной строке $sender и $recipient аргументов.

ОБРАТИТЕ ВНИМАНИЕ,  pipe(8) справочная страница руководства содержит полный список флагов и аргументов.

**Испытание фильтра**

Чтобы проверить фильтр, Вы будете должны выполнить следующие шаги, которые

обсуждаются в следующих разделах:

1. Послать почту удаленному пользователю через внутренний сетевой интерфейс.
2. Проверить лог фаил на наличие действия фильтра.
3. Проверить посланное сообщение на наличие комментария.

### Отправка почты удаленному пользователю

Для формирования сообщение для Постфикса, для посылки фильтру, используйте telnet, чтобы соединиться с внутренним сетевым интерфейсом (где демон smtpd должен использовать фильтр). Вот пример сессии:

**$ telnet 172.16.0.1 25**

Trying 172.16.0.1...

Connected to 172.16.0.1.

Escape character is '^]'.

220 mail.example.com ESMTP Postfix

**HELO client.example.com**

250 mail.example.com

**MAIL FROM: <sender@example.com>**

250 Ok

**RCPT TO: < recipient@remote-example.com>**

250 Ok

**DATA**

354 End data with <CR<LF>.<CR>.<LF>

**FROM: Sender <sender@example.com>**

**TO: Recipient <recipient^remote-example.com>**

**Subject: Testing disclaimer**

**This is a test. There should be text at the bottom of this message**

**added by a disclaimer script.**

**.**

250 Ok: queued as 3C4D043F2F

**QUIT**

221 Bye

### Проверка лог файла почты.

Лог файл должен содержать свидетельство действия фильтра, как в этом примере:

Mar 12 01:59:53 mail postfix/smtpd[30206]: connect from client.example.com[172.16.0.2]

Mar 12 02:00:21 mail postfix/smtpd[30206]: 3C4D043F2F: client=client.example.com[l72.16.0.2]

Mar 12 02:01:53 mail postfix/cleanup[3O209]: 3C4D043F2F:

message-id=<20040312010021.3C4D043F2F@mail.example.com>

Mar 12 02:01:53 mail postfix/nqmgr[30193]: 3C4D043F2F: from=<sender@example.com>,

size=444, nrcpt=l (queue active)

Mar 12 02:01:53 mail postfix/pipe[30213]: 3C4D043F2F; to=<recipient@remote-example.com>,

relay=disclaimer, delay=92, status=sent (mail.example.com) **1**

Mar 12 02:01:53 mail postfix/pickup[30192]: 8421143F2F: uid=100

from=<sender@example.com> **2**

Mar 12 02:01:53 mail postfix/cleanup[30209]: 8421143F2F:

message-id=<20040312010021,3C4D043F2F@mail.example.com>

Mar 12 02:01:53 mail postfix/nqmgr[30193]: 8421143F2F: from=<sender@example.com>,

size=977,nrcpt=1 (queue active)

Mar 12 02:01:55 mail postfix/smtpd[30206]: disconnect from client.example.com[172.16.0.2]

Mar 12 02:02:03 mail postfix/smtp[30220]: 8421143F2F: to=<recipient@remote-example.com>,

relay=mail.remote-example.com[212.14.92.89], delay=10, status=sent (250 Ok: queued as

56851E1C65) **3**

**1** Демон pipe использует транспорт disclaimer, чтобы передать сообщение сценарию.
**2** Сценарий повторно вводит сообщение с оригинальным отправителем конверта.
**3** smtp демон успешно доставляет сообщение получателю конверта.

Проверка сообщения на наличие комментария.

В качестве заключительного (и несколько очевидного) теста, отыщите сообщение и посмотрите, содержит ли оно X-заголовок и комментарий, который alterMIME должен добавить к исходящим сообщения. Вы можете видеть оба в следующем примере:

Return-Path: <sender@example.com>

X-0riginal-To: recipient@remote-example.com

Delivered-To: recipient@remote-example.com

Received: from mail.example.com (mail.example.com [192.0.34.166])

by mail.remote-example.com (Postfix) with ESMTP id 56851E1C65

for <recipient@remote-example.com>; Fri, 12 Mar 2004 02:01:25 +0100 (CET)

Received: by mail.example.com (Postfix, from userid 100)

id 8421143F2F; Fri, 12 Mar 2004 02:01:53 +0100 (CET)

Received: from client.example.com (client.example.com [l72.16.0.2])by

mail.example.com+(Postfix) with SMTP id 3C4D043F2Ffor

<recipient@remote-example.com>; Fri, 12 Mar 2004+02:00:21 +0100 (CET)

From: Sender <sender$example.com>

To: Recipient <recipient@remote-example.com>

Subject: Testing disclaimer

Message-Id: <20040312010021.3C4D043F2F@mail.example.com>

Date: Fri, 12 Mar 2004 02:00:21 +0100 (CET)

X-Copyrighted-Material: Please visit http://www.example.com/

message_disclaimer.html

This is a test. There should be text at the bottom of this message added by a disclaimer script.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nam commodo lobortis magna. Quisque neque. Etiam aliquam. Nulla tempor vestibulum.

## Поиск  вирусов с помощью content_filter и amavisd-new.

Этот раздел описывает расширенное использование content_filter, описанного в Главе 11 - как связать популярную программу, amavisd-new с Постфикс, amavisd связывает MTA и один или более вирусных сканеров или антиспам программы, типа SpamAssassin. Она активно развивается и рекомендуется многими почтовыми администраторами в Постфикс списке рассылки.

*ОБРАТИТЕ ВНИМАНИЕ, Чтобы получить функциональные возможности сканера вирусов, Вы должны иметь по крайней мере один поддерживаемый сканер вирусов , установленный в дополнение к amavisd-new; проверьте документацию для обзора поддерживаемых продуктов.*

Рисунок 12-2 иллюстрирует, как Постфикс и amavisd-new взаимодействует с другими приложениями, типа антиспам фильтров и сканеров вирусов. Вот – порядок прохождения сообщения:

1. Почтовый клиент посылает сообщение Постфиксу.
2. smtpd принимает сообщение.
3. smtpd посылает сообщение qmgr.
4. qmgr посылает сообщение amavisd-new.
5. amavisd-new посылает сообщение другим приложениям (вирусные сканеры в этом примере).
6. amavisd-новый повторно вводит сообщение через локальный smtpd.
7. Локальный smtpd посылает сообщение qmgr.
8. qmgr или отклоняет или доставляет сообщение.

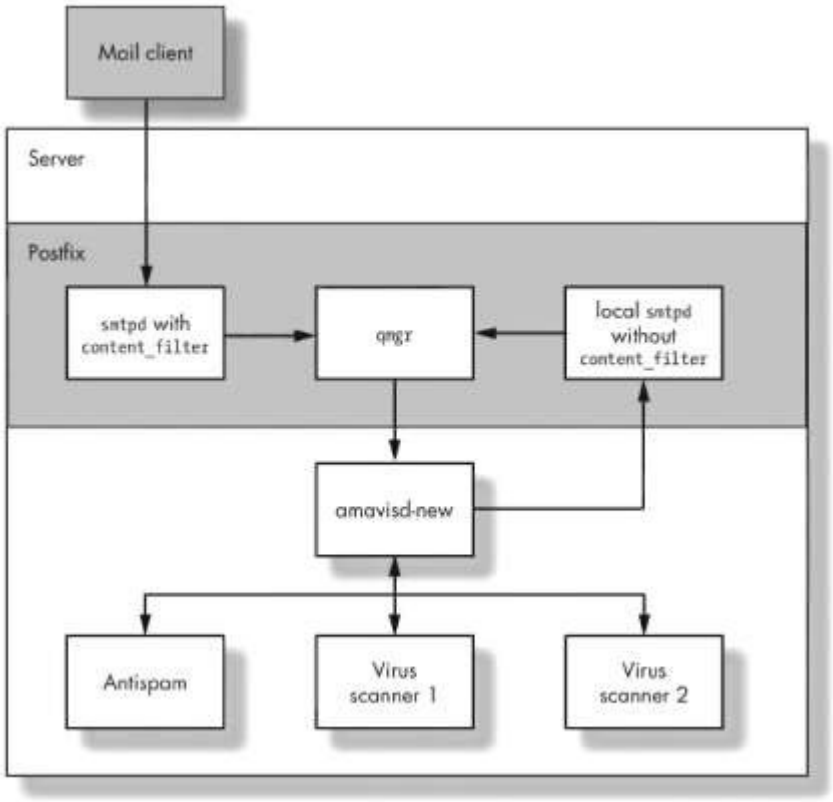



Рис 12-1 взаимодействие Постфикса с amavisd-new с использованием content_filter

**Установка amavisd-new.**

Чтобы установить amavisd-new, загрузите его с одного из зеркал, упомянутых на вебсайте amavisd-new (<http:// www.ijs.si/software/amavisd>). После распаковки архива, следуйте указаниям в файле INSTALL, чтобы установить amavisd-new для Постфикса.
Вы должны также читать README.postfix (<http://www.ijs.si/software/>amauisd/README.postfix) файл для инструкций и замечаний специфичных для Постфикса.

**Совет:** должны скомпилировать только демон amavisd-new. Вспомогательные приложения , типа amavis(.c) и amavisd-milter(.c), не нужны для использования с Постфиксом.

**Установка Модулей Perl для amavisd-new от CPAN**

amavisd-new требует множество Perl модулей, чтобы работать правильно, документ INSTALL в каталоге исходных кодов amavisd-new содержит полный список этих модулей. Устанавливая модули, Вы обычно имеете выбор между выбором пакета, идущего с вашей системой или непосредственно загрузки модулей от CPAN (Comprehensive Perl Archive Network, <http://www.cpan.org>).

*ОБРАТИТЕ ВНИМАНИЕ, что CPAN - вообще лучший источник для новых модулей, но Вы можете захотеть выбрать пакеты вашей операционной системы.*

Для установки модулей, типа Compress:Zlib от CPAN, Вы должны выполнить Perl с модулем CPAN следующим образом:

**# perl -MCPAN -e shell;**

cpan shell -- CPAN exploration and modules installation (vl.76)

ReadLine support enabled

cpan> **install Compress::Zlib**

Running install for module Compress::Zlib

Running make for P/PM/PMOS/Compress-Zlib-1.33.tar.gz

Fetching with LWP;

ftp://ftp-stud.fht-esslingen.de/pub/Mirrors/CPAN/authors/id/P/PM/PMOS/ Compress-21ib-l.33.tar.gz

CPAN:  Digest::MD5 loaded ok

Fetching with LWP:

ftp://ftp-stud.fht-esslingen.de/pub/Mirrors/CPAN/authors/id/P/PM/PMOS/ CHECKSUMS

Checksum for /root/.cpan/sources/authors/id/P/PM/PMOS/Compress-Zlib-1.33.tar.gz ok

Scanning cache /root/.cpan/build for sizes Compress-Zlib-1.33/

... lots of building output  ...

All tests successful, 1 test skipped. Files=6, Tests-287,    2 wallclock sees  ( 0.73 cusr +    0.11 csys -    0.84 CPU)

/usr/bin/make test -- OK

Running make install

Installing /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Compress/Zlib/ Zlib.so

Files found in blib/arch: installing files in blib/lib into architecture

dependent library tree

Installing /usr/lib/perls/site_perl/s.6.0/i386-linux/Compress/Zlib.pm

Installing /usr/man/man3/Compress::Zlib.3pm

Writing /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Compress/Zlib/.packlist

Appending installation info to /usr/lib/perl5/site_perl/5.6.0/i386-linux/ perllocal.pod

/usr/bin/make install    -- OK

После получения модулей и установки amavisd-new, Вы должны проверить ее.

**Проверка  amavisd-new.**

Для проверки amavisd-new в изоляции, перед попыткой получить взаимодействие с Постфиксом, выполните следующие шаги:

1. Запустите amavisd-new в режиме отладки для проверки, что установка прошла должным образом.
2. Выполните сетевой тест, чтобы посмотреть, слушает ли amavisd-new на

сетевом порту.

**Запуск amavisd-new в режиме отладки**

Запуск amavisd-new в режиме отладки сразу дает Вам ответы на следующие вопросы

- Запускается ли он? Все необходимые Perl модули должны быть установлены для запуска amavisd-new. Если модуль будет отсутствовать, то Вы получите сообщение об ошибке, которое указывает на недостающиймодуль.
- Можете ли Вы запустить его как непривилегированного пользователя? amavisd-new требует новой группы (vscan по умолчанию) и учетной записи пользователя в этой группе (также vscan по умолчанию).
- Находит ли он дополнительные модули Perl, которые осуществляют дополнительные функциональные возможности, типа SpamAssassin, LDAP, и SQL?
- Использует ли он правильную установку Perl? Если Вы имеете больше чем одну версию установленного Perl, Вы можете не иметь всех модулей, установленных для специфической версии Perl, который Вы пробуете использовать.
- Находит ли вспомогательные программы, типа вирусных сканеров?
- Какой файл конфигурации он использует? Обычно, это -/etc/amavisd.conf, но Вы можете изменить эту установку , если Вы знаете точно, что Вы делаете.
- Может ли он связаться с портом, определенным в файле конфигурации?

Для вашей первой попытки, лучше запустить amavisd-new в интерактивном режиме, с помощью терминала. Чтобы сделать это , переключитесь как пользователь vscan и выполниться amavisd-new с опцией отладки. Этот пример сеанса показывает вывод, который Вы должны увидеть:

**# su - vscan**

**$ /usr/local/sbin/amavisd debug**

Jan 28 11:10:43 mail amavisd[29188]: starting, amavisd at mail amavisd-new-20030616-p6

Jan 28 11:10:43 mail amavisd[29188]: Perl version 5.006 **1**

Jan 28 11:10:43 mail amavisd[29l88]: Module Amavis::Conf 1.15

Jan 28 11:10:43 mail amavisd[29188]: Module Archive::Tar 1.08

Jan 28 11:10:43 mail amavisd[29188]: Module Archive::Zip 1.09

Jan 28 11:10:43 mail amavisd[29l88]: Module Compress::Zlib 1.33

Jan 28 11:10:43 mail amavisd[2gi88]: Module Convert::TNEF 0.17

Jan 28 11:10:43 mail amavisd[29i88]: Module Convert::UUlib 1.0

Jan 28 11:10:43 mail amavisd[29i88]: Module MIME::Entity 5.404

Jan 28 11:10:43 mail amavisd[29188]: Module MIME::Parser 5.406

Jan 28 11:10:43 mail artiavisd[29188]: Module MIME::Tools 5.411

Jan 28 11:10:43 mail amavisd[29188]: Module Mail::Header 1.60

Jan 28 11:10:43 mail amavisd[29188]: Module Mail:internet 1.60

Jan 28 11:10:43 mail amavisd[29l88]: Module Mail::SpamAssassin 2.63

Jan 28 11:10:43 mail amavisd[29i88]: Module Net::Cmd 2.24

Jan 28 11:10:43 mail amavisd[29l88]: Module Net::DN$ 0.40

Jan 28 11:10:43 mail amavisd[29188]: Module Net::SMTP 2.26

Jan 28 11:10:43 mail amavisd[29188]: Module Net::Server 0.86

Jan 28 11:10:43 mail amavisd[29188]: Module Time::HiRes 1.55

Jan 28 11:10:43 mail amavisd[29188]: Module Unix::Syslog 0.99

Jan 28 11:10:43 mail amavisd[29188]: Found myself: /usr/sbin/amavisd -c /etc/amavisd.conf

Jan 28 11:10:43 mail amavisd[29188]: Lookup::S0L code NOT loaded **2**

Jan 28 11:10:43 mail amavisd[29188]: Lookup::LDAP code NOT loaded

Jan 28 11:10:43 mail amavisd[29188]: AMCL-in protocol code loaded

Jan 28 11:10:43 mail amavisd[29l88]: SMTP-in protocol code loaded

Jan 28 11:10:43 mail amavisd[29i88]: ANTI-VIRUS code loaded

Jan 28 11:10:43 mail amavisd[29i88]: ANTI-SPAM code loaded **3**

Jan 28 11:10:43 mail amavisd[29188]: Net::Server: 2004/01/28-11:10:43 Amavis (type \
Net::Server::PreForkSimple) starting! pid(29188)

Jan 28 11:10:43 mail arnavisd [ 29188]: Net:: Server: Binding to UNIX socket file \
/var/amavis/amavisd.sock using SOCK_STREAM

Jan 28 11:10:43 mail amavisd[29188]: Net::Server: Binding to TCP port 10024 on host
127.0.0.1

Jan 28 11:10:43 mail amavisd[29188]: Net::Server: Setting gid to "54322 54322"

Jan 28 11:10:43 mail amavisd[29l88]: Net::Server: Setting uid to "7509"

Jan 28 11:10:43 mail amavisd[29188]: Net::Server: Setting up serialization via flock

Jan 28 11:10:43 mail amavisd[29188]: Found $file at /usr/bin/file

Jan 28 11:10:43 mail amavisd[29188]: Found $arc at /usr/bin/arc

Jan 28 11:10:43 mail amavisd[29188]: Found $gzip at /usr/bin/gzip

Jan 28 11:10:43 mail amavisd[29188]: Found $bzip2 at /usr/bin/bzip2

Jan 28 11:10:43 mail amavisd[29188]: Found $lzop at /usr/local/bin/lzop

Jan 28 11:10:43 mail amavisd[29l88]: Found $lha at /usr/bin/lha

Jan 28 11:10:43 mail amavisd[29i88]: Found $unarj at /usr/bin/unarj

Jan 28 11:10:43 mail amavisd[29i88]: Found $uncompress at /usr/bin/uncompress

Jan 28 11:10:43 mail amavisd[29188]: Found $unfreeze at /usr/local/bin/unfreeze

Jan 28 11:10:43 mail amavisd[29i88]: Found $unrar at /usr/bin/rar

Jan 28 11:10:43 mail amavisd[29i88]: Found $zoo at /usr/bin/zoo

Jan 28 11:10:43 mail amavisd[29l88]: Found $pio at /bin/cpio **4**

Jan 28 11:10:43 mail amavisd[29l88]: No primary av scanner: KasperskyLab Antiviral Toolkit \

Pro (AVP)

Jan 28 11:10:43 mail amavisd[29188]: No primary av scanner: KasperskyLab AVPDaemonClient

Jan 28 11:10:43 mail amavisd[29188]: No primary av scanner: H+BEDV AntiVir or \ CentralCommand Vexira Antivirus

Jan 28 11:10:43 mail amavisd[29l88]: No primary av scanner: Command Antivirus for Linux

Jan 28 11:10:43 mail amavisd[29188]: No primary av scanner: Symantec CarrierScan via \ Symantec CommandLineScanner

Jan 28 11:10:43 mail amavisd[29i88]: No primary av scanner: Symantec Antivirus Scan Engine

Jan 28 11:10:43 mail amavisd[29l88]: No primary av scanner: Dr.Web Antivirus for \ Linux/FreeBSD/Solaris

Jan 28 11:10:43 mail arnavisd[29188]: No primary av scanner: F-Secure Antivirus

Jan 28 11:10:43 mail amavisd[29188]: No primary av scanner: CAI InoculatelT

Jan 28 11:10:43 mail amavisd[29188]: No primary av scanner: MkSVir for Linux (beta)

Jan 28 11:10:43 mail amavisd[29188]: No primary av scanner: MkSVir daemon

Jan 28 11:10:43 mail amavisd[29l88]: No primary av scanner: ESET Software NOD32

Jan 28 11:10:43 mail arnavisd[29188]: No primary av scanner: ESET Software NOD32 - \ Client/Server Version

Jan 28 11:10:43 mail amavisd[29188]: No primary av scanner: Norman Virus Control v5 / Linux

Jan 28 11:10:43 mail amavisd[29188]: No primary av scanner: Panda Antivirus for Linux
Jan 28 11:10:43 mail amavisd[29188]: Found primary av scanner NAI McAfee Antivirus (uvscan) \
at /usr/local/bin/uvscan **5**
Jan 28 11:10:43 mail amavisd[29188]: No primary av scanner: VirusBuster
Jan 28 11:10:43 mail arnavisd[29188]: No primary av scanner: CyberSoft VFind
Jan 28 11:10:43 mail amavisd[29l88]: No primary av scanner: Ikarus Antivirus for Linux
Jan 28 11:10:43 mail amavisd[29i88]: No primary av scanner: BitDefender
Jan 28 11:10:43 mail amavisd[29i88]: No secondary av scanner: Clam Antivirus - clamscan
Jan 28 11:10:43 mail amavisd[29188]: No secondary av scanner: FRISK F-Prot Antivirus
Jan 28 11:10:43 mail amavisd[29188]: No secondary av scanner: Trend Micro FileScanner
Jan 28 11:10:43 mail amavisd[29188]: SparnControl: initializing Mail::SpamAssassin **6**

1 Эта строка показывает версию Perl.
2 Эта строка и следующие показывают, что никакой код SQL или LDAP не присутствует.
3 Эта строка указывает, что код антиспам филтров был загружен; что выполняется успешно , когда присутствуют SpamAssassin или dspam.
4 Эта линия и предыдущие линии указывают, что различные внешние распаковщики были найдены, позволяя amavisd-new, распаковывать вложения, сжатые с этими упаковщиками.
5 Эта строка показывает, что Антивирус McAfee присутствует.

6 В этом пункте, amavisd-new готов к работе.

**Тестирование сетевого соединения**

С автономным amavisd-new все ясно, теперь Вы должны посмотреть, принимает ли он подключения. Используйте telnet, чтобы проверить обе ESMTP и LMTP альтернативы.

Тестирование доступности ESMTP

Установите соединение telnet к локальному порту 10024 (значение по умолчанию для amavisd-new). Вы должны проверить, что оу слушает на порту и отвечает на команды ESMTP, amavisd-туц должен ответить на команду EHLO набором возможных команд, как в этом примере:

**# telnet localhost 10024**

220 [127.0.0.1] ESMTP amavisd-new service ready EHL0 mail.exampie.com

250-[l27.0.0.1]

250-PIPE LINING

250-SIZE

250-8BITMIME

250 ENHANCEDSTATUSCODES

QUIT

221 2.0.0 [127.0.0.1] (amavisd) closing transmission channel

Тестирование доступности LMTP

Затем Вы должны проверить, что amavisd-new слушает на локальном порту 10024 (значение amavisd-new по умолчанию для LMTP) и отвечает на команды LMTP. Вы должны быть в состоянии выполнить команду LHLO, как в этом сеансе:

**# telnet localhost 10024**

220 [127.0.0.1] ESMTP amavisd-new service ready

**LHLO mail.example.com**

250-[127.0.0.1]

250-PIPELINING

250-SIZE

250-8BITMIME

250 ENHANCEDSTATUSCODES

QUIT

221 2.0.0 [127.0.0.1] (amavisd) closing transmission channel

**Оптимизация производительности amavisd-new**

Если Вы получаете много почты, Вы возможно хотите достигнуть максимальной amavisd-new. Поскольку он сильно загружает файловую систему, чтобы подготовить сообщения к дальнейшему просмотру, производительность amavisd-new зависит от скорости дискового ввода - вывода. Вы можете значительно оптимизировать операцию чтения - записи, проносятся, перемещая эту подготовку в раздел RAM файловой системы. Процедура, описанная в следующих разделах использует временную фаиловую систему Linux (tmpfs).

**Действительно ли это безопасно?**

Вы можете быть уверены, что Вы не потеряете никакой электронной почты в течение процесса фильтрации из-за способа, с помощью которого Вы внедряете amavisd-new в Постфикс. Рассмотрим, что происходит во время фильтрации:

1. После получения нового сообщения, менеджер очереди Постфикс посылает запрос на доставку почты SMTP или LMTP клиенту ;клиента передает сообщение к amavisd-new.
2. amavisd-new начинает работу над сообщением (выполнение сканирования на спам, и так далее), но он не признает немедленно, что получило сообщение.
3. При ожидании amavisd-new, Постфикс сохраняет сообщение в очереди, в ожидании от amavisd-new сообщения о получении сообщения.
4. После окончания работы amavisd-new повторно вводит сообщение назад в Постфикс очередь.
5. Постфикс smtpd демон, который отвечает за повторный ввод сообщения, принимает сообщение от amavisd-new.
6. После получения подтверждения от повторного ввода smtpd демона, amavisd-new сообщает об успешной доставке Постфикс lmtp или smtp клиенту, который в свою очередь сообщает назад менеджеру очереди, что сообщение доставлено.

Как Вы можете видеть, amavisd-new только сообщает до фильтровому демону Постфикса , что он получил сообщение после того, как после фильтровый демон smtpd принимает обработанное сообщение. Таким образом, Вы никогда не потеряете почту в amavisd-new,

**Установка размеров tmpfs**

Чтобы вычислять правильный размер для tmpfs, рассмотрите это: Если Вы запускаете n amavisd-new rjgbq, и каждый принимает сообщения размером message_size_limit, Вы нуждаетесь в следующем количестве пространства:

n * (1 + maximum_expected_expansionfactor) * message_size_limit * 7/8

expansionfactor весьма не прост, но его значение равное 2 весьма хорошо подходит (принимается что *.zip или *.rar размер файлов увеличивает вдвое от оригинального размера архива).

Например, если бы Вы имеете пять amavisd-new демонов и максимальный размер сообщения состовляет 10 МБ, Вы получили бы следующий результат для размера tmpfs:

5 * (1 + 2) * 10MB * 7/3 = 131.25MB

*ПРИМЕЧАНИЕ Удостоверьтись, что Вы имеете достаточно физической памяти, чтобы содержать tmpfs; иначе ваша машина начнет кешировать память на диске, и Вы получите быстродействие , хуже чем с обычной файловой системой.*

Настройка оптимизации

Есть несколько шагов, необходимых для настройки использования amavisd-new tmpfs:

1. Найти $TEMPBASE amavisd-new параметр.
2. Создать tmpfs файловую систему.
3. Остановитесь amavisd-новый.
4. Смонтировать файловую систему tmpfs.
5. Запустить Amavisd-новое начало.
6. Удостоверьтесь, что amavisd-новый работает

Сначала, Вы должны узнать где определяется параметр $TEMPBASE amavisd-new . Это - точка монтирования для tmpfs, которую Вы создадите. По умолчанию $TEMPBASE - $MYHOME, которая является / var/amavis по умолчанию. Чтобы узнать наверняка, используйте grep в вашем файле конфигурации. Этот пример показывает, что используется $MYHOME:

```
# grep TEMPBASE /etc/amavisd.conf
STEMPBASE = JMYHOME;                    # (must be set if other config vars use is)
$ENV{TMPDIR} = $TEMPBASE;               # wise, but usually not necessary
"-f=$TEMPBASE {}",  [0,8],  [3,4,5,6],  qr/infected:  ([A\r\n]+)/  ],
# adjusting /var/amavis above to match your STEMPBASE.
#   directory $TEMPBASE specifies) in the 'Names=' section.
```

Выполните grep еще раз, чтобы узнать, каково значение $MYHOME. В следующем примере, Вы можете видеть, что определение $myhome pfкомментировано, таким образом используется значение по умолчанию:

```
#  grep MYHOME /etc/amavisd.conf
#  $MYH0ME serves as a quick default for some other configuration settings.
#  $MYH0ME is not used directly by the program.  No trailing slash!
#$MYH0M£ =  '/var/lib/amavis';      # (default is  Vvar/amavis')
$TEMPBASE = $MYHOME;                    # (must be set if other config vars use is)
#$TEMPBASE = "$MYHOME/tmp";          # prefer to keep home dir /var/amavis clean?
#$helpers_home = $MYH0ME;             # (defaults to $MYH0ME)
#$daemon_chroot_dir = $MYH0ME;    # (default is undef,  meaning:  do not chroot)
#$pid_file    = "$MYHOME/amavisd.pid";    # (default is "$MYHOME/amavisd.pid")
```

```
#$lock_file = "$MYHOME/amavisd.lock";  # (default is "$MYHOME/amavisd.lock")
#$forward_method = "bsmtp:$MYHOME/out-%i-%n.bsmtp";
$unix_socketname = "$MYHOME/amavisd.sock"; # amavis helper protocol socket
                                           # (usual setting is $MYH0ME/amavisd.sock)
$L0GFILE = "$MYHOME/amavis.log";    # (defaults to empty, no log)
"{}  -ss -i  '*'  -log=$MYHOME/vbuster.log'\   [0],  [1],
```

Теперь Вы должны создать запись tmpfs  в вашем файле /etc/fstab, используя размер файловой системы, вычисленный в предыдущем разделе ("Размер tmpfs"). Следующий пример использует размер 150 МБ и ограничивает доступ  определеным пользователем и группой. В данно случае user ID - 7509, и GID - 54322, которые соответствуют пользователю и группе vscan в файлах /etc/passwd и /etc/group; имейте в виду, что ваша система почти наверняка имеет другие числа, и Вы будете должны отыскать их:

```
/dev/shm  /var/amavis  tmpfs defaults,size=150m,mode=7O0,uid=7509,gid=54322 0 0
```

Прежде, чем Вы смонтируете /var/amavis, удостоверьтесь, чтобы остановиться amavisd-new следующей командой:

**# /etc/init.d/amavisd-new stop**

Затем, смонтируйте,/var/amavis (помните, что это файловая система tmpfs, которую Вы только определили в/etc/fstab):

**# mount /var/amavis**

Теперь заново запустите amavisd-new

**# /etc/init.d/amavisd-new start**

Проверьте, все ли хорошо, исследуя лог файл и исследуя вывод df-h. В следующем примере,/var/amavis - 100 МБ, и только 76 КБ используются в настоящее время:

**# df -h /var/amavis**

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/shm              100M   76k   99M   1%  /var/amavis
```

ОБРАТИТЕ ВНИМАНИЕ Иногда amavisd-new оставляет старые файлы в $TEMPBASE директории. Чтобы защитить $TEMPBASE от заполнения этими файлами, Вы можете ежедневно останавливать amavisd-new, удалять старые файлы, и перезапустить его. Следующий сценарий выполняет эту работу:

```
#!/bin/bash
/etc/init.d/amavisd stop
rm -Rf /var/amavis/amavis-200*
/etc/init.d/amavisd start
```

**Настройка Постфикса, для Использования amavisd-new**

В этом пункте, Постустановите, и amavisd-new должены работатьнезависимо друг от друга. Поэтому, Вы должны настроить Постфикс, для транспортировки сообщения amavisd-new и создать другую копию smtpd для повторного ввода сообщения. Следующие шаги (обсужденный в следующих разделах) интегрируют amavisd-new в, Постфикс:

1. Создание транспорта.
2. Настройка транспорта.
3. Настройка пути для повторного ввода.

*ОБРАТИТЕ ВНИМАНИЕ, поскольку почта нуждается в способе возврата назад систему Постфикс очереди, не будучи сканированной снова, Вы нуждаетесь в отдельном smtpd, который не использует content_filter. Это позволяет amavisd-new, повторно вводят почту в систему, не создавая петли. 25 порт уже занят, таким образом Вы можете заставить демон smtpd слушать на нестандартном порту. Этот пример использует 10025 порт на localhost.*
*amavisd-new также нуждается в порту, для приема сообщений от Постфикса. По умолчанию 10024 порт на localhost .*

**Создание Транспорта использующего conetnt_filter в main.cf**

Первый шаг в делегировании обработки содержимого внешней программе должен определить транспорт, который посылает сообщения программе фильтрования. Постфикс использует content_filter параметр в main.cf файле. Параметр имеет формат transportname:nexthop:port.

В данном примере мы принимаем что, amavisd-новый работает на той же самой машине, что и Постфикс. Таким образом Вы можете получить доступ к нему на 10024 порту localhost (127.0.0.1). Вы должны определить следующий параметр content_filter в main.cf файле, чтобы установить соединение Постфикса с amavisd-new:

contentfilter = amavisd-new:[127.0.0.l]:10024

*Запуск amavisd-new на другом хосте*

Если Вы чувствуете, что нагрузка - слишком велика для одной машины, Вы можете запустить amavisd-new на одной или более машинах nexthop часть параметра transportname:nexthop:port позволяет Вам легко определить другой хост для фильтра. Например на vscanners.example.com в следующем примере:

content_filter = amavisd-new:vscanners.example.com:10024

Имя vscanners.example.com могло быть любым из следующих:

- Одна машина (одна A запись)
- Многочисленные машины( одна коллективная A запись)
- Многочисленные машины(одна или более машин с различным приоритетом MX записей)

**Определение транспорта в master.cf**

Затем Вы должны определить демона, который соединится с amavisd-new и определить окружение  для демона. Помните, что демон может быть smtp, lmtp, или pipe. Вы видели пример использование демона pipe ранее в этой главе; пришло время рассмотреть другие два:

*Определение транспорта ESMTP*

Если Вы хотите использовать протокол ESMTP, чтобы посылать сообщения amavisd-new, добавьте следующие записи в ваш master.cf файл:

```
# service type      private unpriv chroot  wakeup  maxproc command
#                     (yes)   (yes)  (yes)   (never)  (100)
# ==========================================
 amavisd-new      unix        -             -            n
-o smtp_data_done_timeout=1200s
-o  disable_dns_lookup=yes
```

Есть несколько вещей, которые нужно отметить в предыдущих записях:

- специализированный amavisd-new транспорт – копия обычного транспорта smtp, Его имя должно соответствовать имени транспорта, которое Вы дали content_filter параметр, который Вы определили в main.cf.
- amavisd-new весьма требователен к ресурсам. Если у Вас не быстрая машина, Вы могли бы захотеть оставить максимальное число одновременных копий равным 2.
- smtp_data_done_timeout параметр является первым из двух дополнительных параметров окружения, которые изменяют поведение этого демона, amavisd-new может затратить существенное время, чтобы обработать входящие сообщение, и увеличение перерыва после того, как smtp посылает сообщение, защищает, Постфикс от отказа, прежде чем amavisd-new обработал сообщение.
- Поскольку Вы вероятно имеете дело только с локальными именами машин в этом пункте, параметр disable_dns_lookups устраняет ненужные запросы DNS для smtp клиента.

*ОБРАТИТЕ ВНИМАНИЕ, что Вы не обязательно нуждаетесь в специфических настройках транспорта SMTP, потому что по умолчанию smtp демон работает хорошо. Однако, по причинам быстродействия (и из-за относительно длинного amavisd-new ожидания), может быть имеет смысл настраивать параметры транспорта только для amavisd-new.*

**Определение транспорта LMTP**

Если Вы реш использовать протокол LMTP (вместо SMTP), чтобы передавать сообщения  amavisd-new, добавьте следующие записи в ваш файл master.cf:

```
#  service type             private   unpriv   chroot    wakeup    maxproc     command
#                             (yes)     (yes)     (yes)      (never)   (100)
```

```
amavisd-new     unix    -       -       n       -       2       lmtp
-o lmtp_data_done_timeout=1200
-o disable_dns_lookups=yes
```

**Настройка пути повторного ввода**

Наконец, Вы должны создать путь повторного ввода, который позволит amavisd-new возвращать сообщения назад в Постфиксную очередь. Важно что этот путь повторного ввода миновал amavisd-new транспорт. Иначе сообщение будет зациклено, куда Постфикс посылает сообщение amavisd-new, почта повторно вводится в Постфиксную очередь, и снова посылается назад amavisd-new.
Путь повторного ввода, который обходит любого предварительно, определенный параметр content_filter, похож на эту запись в вашем master.cf файле:

```
#.....................................................................
# service type      private unpriv chroot  wakeup  maxproc  command
#                   (yes)   (yes)  (yes)   (never) (100)
#
==========================================================================
===
127.0.0.1:10025 inet    n       -       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_rfc82i_enwelopes=yes
```

Из всех перечисленных опций, абсолютно необходимой , - пустой параметр content_fliter. Он отменяет значение параметр content_filter в main.cf файле. Остающиеся опции отменяют другие main.cf параметры, включая опции, выключаюшие ограничения, которые не имеют смысла для транспорта, слушающего только на локальном сетевом интерфейсе.

**Испытание работы Постфикса с amavisd-new фильтром**

Чтобы проверить, что Постфикс и amavisd-new работf.n хорошо вместе, Вы должны проверить, что Постфикс может послать почту amavisd-new, и что amavisd-new может

повторно вводить сообщения. Тест содержит следующие шаги:

1. Проверьте, что Постфикс слушает на порту повторного ввода
2. Пошлите сообщение Постфиксу, проверяя, что это посылает сообщение amavisd-new, и что сообщение возвращается в Постфикс очередь.
3. Удостоверитесь, что вирусный сканер обнаруживает тестовый образец.

Как только Вы изменили master.cf файл, перезагрузите Постфик, чтобы заставить Постфикс прочитать измененный файл и затем исследуйте лог файл на любые сообщения о проблемах. Затем, проверьте, слушает ли smtpd демон smtpd на , порт 10025 localhost, как в следующем сеансе:

**$ telnet 127.0.0.1 10025**

220 mail.example.com ESMTP Postfix

EHL0 127.0.0.1

250-mail.example.com

250-PIPELINING

250-SIZE 10240000

250-VRFY

250-ETRN

250-STARTTLS

250-AUTH LOGIN PLAIN DIGEST-MD5 CRAM-MD5

250-XVERP

250 8BITMIME

**QUIT**

221 Bye

### Отправка тестового Сообщения  Постфиксу

Постфикс должен быть в состоянии позволить незараженному сообщению проходить через систему. Пошлите сообщение из командной строки, и проследите его с помощью лог файлов Постфикса и amavisd-new. Например, Вы могли использовать следующую команду, чтобы отправить ваш main.cf файл по почте recipient@exainple.com:

**# sendmail -i sender@example.com recipient@example.com < /etc/postfix/main.cf**

Затем просмотрите лог файл.  В начале файла должен быть сеанс, который  похож следующий, где Постфикс назначает ID сообщению которое Вы можете использовать, чтобы проследить сообщение:

Jan 31 10:45:08 mail postfix/pickup[l0096]:  2788029AB29:  uid=0 from=<sender@example.com>

Jan 31 10:45:08 mail postfix/cleanup[l0652]:  2788029AB29: message-

id=<20040i3i094508.2788029AB29@mail.example.com>

Следующий набор сообщений должен показать, что Постфикс вручил сообщение localhost для того, чтобы обработать amavisd-new (к сожалению, Постфикс не регистрирует номер порта или имя транспорта):

Jan 31 10:45:08 mail postfix/qmgr[l0O97]: 2788029AB29: from=<sender@example.com>,
size=1271,
nrcpt=1 (queue active)
Jan 31 10:45:08 mail postfix/smtp[10660]: 2788029AB29: to=<recipient@example.com>,
relay=localhost[l27.0.0.1], delay=0, status=sent (250 2.6.0 Ok, id=25809-04, from MTA: 250
0k: queued as 377D829AB2A)

Теперь amavisd-new сканирует сообщение и заносит файл регистрации, что сообщение передано:

Jan 31 10:45:08 mail amavis[25809]: (25809-04) Passed, <sender@example.com> ->
<recipient@example.com>, Message-ID: <20040131094508.2788029AB29@mail.example.com>,
Hits: -

Затем, сообщение возвращается в Постфикс из amavisd-new для повторного ввода в очередь. Обратите внимание, что второй smtpd демон также регистрирует ID сообщения:

Jan 31 10:45:08 mail postfix/smtpd[10688]: connect from localhost[127.0.0.1]
Jan 31 10:45:08 mail postfix/smtpd[10688]: 377D829AB2A: client=localhost[l27.0.0.l]
Jan 31 10:45:08 mail postfix/cleanup[10652]: 377D829AB2A:
message-id=<20040131094508.2788029AB29@mail.example.com>
Jan 31 10:45:08 mail postfix/qmgr[10097]: 377D829AB2A: from=<sender@example.com>,
size=1723,
nrcpt=l (queue active)
Jan 31 10:45:08 mail postfix/smtpd[l0658]: disconnect from localhost[l27.0.0.l]

Наконец, Постфикс передает сообщение другому хосту для доставки (он мог бы также доставить его локально, если бы этот сервер, был заключительным адресатом):

Jan 31 10:45:08 mail postfix/smtp[l0655]: 377D829AB2A: to=<recipient@example.com>,
relay=relayhost[l0,0,0.l], delay=0, status=sent (250 OK id=1AmrgY-00073g-00)

**Проверка с помощью тестового вируса**

Ваш последний тест должен моделировать получение сообщения, зараженного вирусом. Вы можете сделать это, используя проверочный вирусный образец EICAR

(<http: // www.eicar.org>), послав его Постфиксу. Любые вирусные сканеры, в которых не отключен этот образец преднамеренно, должны быть в состоянии распознать его. Например, следующая команда должна послать вирус recipient@example.com:

**# sendmail -f sender@example.com recipient@example.com < eicar.com**

Сообщения в лог файле похожи, на прежние, до пункта, где amavisd-new просматривает сообщение:

Feb 6 15:48:54 mail postfix/pickup[3005l]: 13B9E29AB29: uid=0
from=<sender@example.com> Feb 6 15:48:54 mail postfix/cleanup[30741]: 13B9E29AB29:
message-id=<20040206144854.13B9E29AB29@mail.example.com>
Feb 6 15:48:54 mail post-fix/qmgr[l9295]: 13B9E29AB29: from=<sender@example.Com>,
size=347,
nrcpt=1 (queue active)
Feb 6 15:48:54 mail postfix/smtp[30744]: 13B9E29AB29: to=<recipient@example.com>,
relay=localhost[l27.0.0.1], delay=0, status=sent (250 2.5.0 Ok, id=10217-07, BOUNCE)
Feb 6 15:48:54 mail amavis[i02i7]: (10217-07) INFECTED (Eicar-Test-Signature),
<sender№example.com> -> <recipient@example.com>, quarantine virus-20040206-154854-
10217-07,
Message-ID: <20040206144854.13B9E29AB29@mall.example.com>, Hits: -

Видя, что сообщение содержит вирус, amavisd-new посылает предупреждение на адресс virusalert@example.com и отклоняет сообщение назад отправителю:

Feb 6 15:48:54 mail postfix/smtpd[30747]: connect from localhost[l27.0.0.l]
Feb 6 15:48:54 mail postfix/smtpd[30747]: 639A729AB2A: client=localhost[l27.0.0.l]
Feb 6 15:48:54 mail postfix/cleanup[30741]: 639A729AB2A: message-id=<VA10217-07@mail>
Feb 6 15:48:54 mail postfix/qmgr[19295]: 639A729AB2A: from=<>, size=1463, nrcpt=l
(queue active)
Feb 6 15:48:54 mail postfix/local[30749]: 639A729AB2A: to=<virusalert@example.com>,
relay=local, delay=0, status=sent (forwarded as 8484829AB2C)
Feb 6 15:48:54 mail postfix/smtpd[30747]: disconnect from localhost[l27.0.0.l]
Feb 6 15:48:54 mail postfix/smtpd[30747]: connect from localhost[l27.0.0.l]
Feb 6 15:48:54 mail postfix/smtpd[30747]: 7A2FD29AB2B: client=localhost[l27.0.0.l]
Feb 6 15:48:54 mail post-fix/cleanup[3074l]: 7A2FD29AB2B: message-id=<VS10217-07mail>
Feb 6 15:48:54 mail post-fix/qmgr[ 19295]: 7A2FD29AB2B: from=<>, size=2554,
nrcpt=i (queue active)
Feb 6 15:48:55 mail postfix/smtp[30744]: 7A2FD29AB2B: to=<sender@example.com>,

relay=relayhost[lO.O.O.l]j delay=l, status=sent (250 OK id=lAp7Ho-000141-00)

Отправка сообщения назад отправителю не особенно хорошая идея, потому отправитель почти всегда подделывается в текущих почтовых вирусах, но это, к сожалению, значение по умолчанию для amavisd-new.

**Сканирование вирусов с smtpd proxy_filter и amavisd-new**

Другой, более новый подход фильтрования содержимого в Постфикс состоит в том, чтобы просмотреть поступающие сообщения перед организацией их в очередь . Этот тип фильтра называют smtpd_proxy_filter. Вы можете использовать его с amavisd-new, как показано на рисунке 12-3.

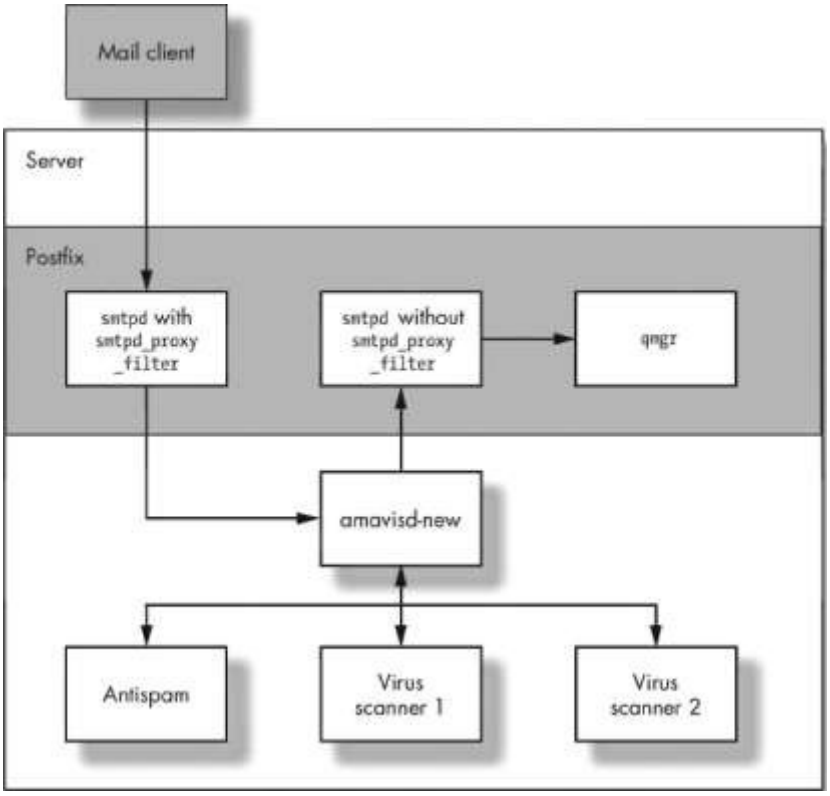



Рис 12-3 Интеграция amavisd-new и Постфик с использованием smtpd_proxy_filter.

Вот как обрабатывается сообщение, если Вы используете smtpd_proxy_filter:

1. Почтовый клиент посылает сообщение smtpd демону Постфикс.
2. smtpd (с smtpd_proxy_filter) передает сообщение amavisd-new. Обратите внимание, что это отличается от случая с content_filter, где менеджер очереди вызывает Постфикс lmtp или smtp клиента для передачи сообщения amavisd-new.
3. amavisd-туц посылает сообщение другим приложениям (в этом примере, двум вирусным сканерам).
4. Amavisd-new сообщает smtpd, принимает ли он или отклоняет сообщение. Если он принимает сообщение,он повторно вводит это назад второму smtpd демону, но если он отклоняет сообщение, то действует согласно вашей конфигурации.
5. Первый smtpd ждет ответ amavisd-new, принимая или отклоняя сообщение от клиента.

*ОТМЕТЬТЕ, что smtpd_proxy_filter - smtpd демон, в котором отменены две части:*

- *Первая часть обрабатывает поступающую почту фильтром.*
- *Другая часть организует очередь.*

Этот раздел объясняет, как конфигурировать amavisd-new с smtpd_proxy_filter, используя общие шаги, описанные в Главе 11. Вы должны выполнить следующие шаги, чтобы объединить amavisd-new с параметром smtpd_proxy_fiter:

1. Установить amavisd-new (как описано ранее в разделе "Установка amavisd-new".
2. Проверить amavisd-new ( как описано ранее в разделе "Проверка amavisd-new").
3. Сконфигурировать Постфикс, для использования amavisd-new.
4. Проверить конфигурацию,

**Настройка Постфикса, для использования amavisd-new с smtpd_proxy_filter**

Первым делом Вы должны определить транспорты, с помощью которых электронная почта должна попадать в программу фильтрации. Вы должны сделаеть эти три вещи:

1. Изменить существующий транспорт smtpd чтобы он стал proxy для amavisd-new.
2. Создать дополнительный демон smtpd, чтобы повторно ввести почту в Постфикс, в обход любого глобального smtpd_proxy_filter параметра.
3. Проверить конфигурацию как описано в предыдущем разделе

Изменение существующего smtpd до proxy

Чтобы smtpd демон выполнял функции proxy для amavisd-new, добавте параметр smtpd_proxy_filter для существующей smtp службы в master.cf файле. Например следующая запись заставляет smtpd послать сообщения, на 10024 порт localhost (помните, что это - порт по умолчанию для amavisd-new):

```
#==========================================================
# service type  private  unpriv  chroot  wakeup  maxproc command
#                 (yes)   (yes)   (yes)    (never)    (100)
#==========================================================
.....
smtp      inet      n       -       n       -       20      smtpd
-o smtpd_proxy_filter=localhost:10024
-o smtpd_client_connection_count_limit=10
```

Обратите внимание, что параметр -o smtpd_client_connection_count_limit=10 препятствует одному клиенту SMTP израсходовать все 20 процессов сервера SMTP, определенных в столбце maxproc. Этот предел не нужен, если Вы получаете всю почту от релей хоста, которому доверяете.
Кроме того, в отличие от процесса, используемого ранее для механизма content_filter , Вы не определяете глобальный параметр в main.cf файле, таким образом вам не нужно будет явно отменять его в транспорте повторного ввода.

**Создание дополнительного демона smtpd для повторного ввода сообщения**

Чтобы позволить сообщениям повторно попадать в Постфикс очередь спомощью не прокси smtpd демона, Вы должны добавить специальный демон smtpd в вашем master.cf файле. Этот пример создает другой демон слушающий на 10025 порту localhost:

```
#==========================================================
#                  service type     private unpriv chroot  wakeup  maxproc  command
#                   (yes)  (yes)   (yes)   (never)     (100)
#==========================================================
127.0.0.1:10025   inet              n       -       n       -       -           smtpd
-o smtpd_authorized_xforward_hosts=127.0.0.0/8
-o smtpd_client_restrictions=
-o smtpd_helo_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipierrt_restrictions=permit_mynetworks,reject
-o mynetworks=127.0.0.0/8
-o receive_override_options=no_unknown_recipient_checks
```

Параметр -o smtpd_authorized_xforward_hosts=127.0.0.0/8 позволяет после фильтровому демону smtpd получать информацию о удаленном SMTP клиенте от до фильтрового smtpd демона. После фильтровый smtpd примет любые команды XFORWARD, посланные хостом, перечисленным в smtpd_authorized_xforward_hosts. Это очень полезно для отладки, потому что smtpd будет использовать оригинальный IP адрес клиента вместо localhost [127. 0.0.1],

Оставшиеся параметры облегчают нагрузку на после фильтровый smtpd, потому что, до фильтровый smtpd уже проделал эту работу.

# РАСШИРЕННЫЕ КОНФИГУРАЦИИ

В этой части книги, Вы будете видеть общие ситуации, когда Постфикс взаимодействует с другими сторонними приложениями, типа серверов SQL, Cyrus SASL, OpenSSL, и OpenLDAP. Вот - краткий обзор глав в этом разделе:

**Шлюзы электронной почты**

Почтовый шлюз передает сообщения от имени других почтовых серверов или клиентов. В большинстве случаев, почтовые релеи выставлены в Интернет, в то время как другие серверы находятся в безопасности за межсетевым экраном. В Главе 13, Вы будете видеть, как сделать "умный" хост из простого почтового Релея,

**Почтовый сервер для нескольких доменов**

Глава 14 описывает два пути, которыми Постфикс может обработать почту для нескольких доменов. Кроме того, Вы будете видеть, как настроить Постфикс, для выполнения запросов серверу SQL вместо того, чтобы просматривать статические карты.

**Понятие SMTP аутентификации**

аутентификация SMTP - система для подтверждения подлинности почтовых клиентов прежде, чем они передадут сообщения. Поскольку идентификация SMTP в Постфиксе выполняется через программное обеспечение Cyrus SASL, Глава 15 покажет Вам, как сконфигурировать Cyrus SASL прежде, чем Вы можете использовать его с Постфиксом.

**SMTP аунтификация**

Продолжая обсуждение SMTP аутентификации, Глава 16 показывает Вам, как настроить Постфикс для аутентификации на стороне сервера или клиента и обоих.

**Понятие безопасности транспортного уровня**

Безопасность транспортного уровня (TLS}) шифрует уровень связи между Постфиксом и другими хостами. Выполнение Постфиксом TLS требует наличия OpenSSL, таким образом Глава 17 показывает Вам не только как работает TLS, но также и как подготовить необходимые сертификаты.

**Использование безопасность транспортного уровня (TLS)**

Глава 18 показывает Вам, как установить Постфикс сервер, чтобы предложить кодирование другим хостам и как заставить Постфикс клиента использовать его, когда другие серверы предлагают TLS также Вы увидите, как работает передача на основе скртификата.

**Почтовый сервер компании**

Глава 19 объясняет, как сконфигурировать Постфикс, чтобы сделать запрос серверу LDAP. При этом, Вы будете делегировать локальную доставку MDA (агент доставки сообщения) и узнаете основы конфигурирования Courier IMAP  сервера . В конце, Вы будете иметь полную почтовую систему, которая получает данные о пользователях от сервера Open LDAP.

**Запуск Постфикса в chroot среде**

Глава 20 показывает Вам как,настроить Постфикс, для запуска chroot среде. Она объясняет, почему некоторые демоны не должны выполнятся в chroot окружении и дает Вам пример того, как запустить Постфикс в chroot окружении в связке с SASL.

# Шлюзы электронной почты

Шлюз электронной почты (также называемый "smarthost") - сервер, который находится между логически отдельными сетями. . Обычно, шлюз электронной почты обнаруживается как конечный адресат в записях ДНС для других почтовых серверов в Интернете. И те серверы понятия не имеют, что другие почтовые серверы находятся вне шлюза электронной почты. Эта глава показывает Вам, как установить шлюз электронной почты, и обсуждает характеристики «настоящего» smarthost`a.

Компании и провайдеры используют шлюзы электронной почты, чтобы управлять SMTP трафиком, идущим к и от их сети. Обычно сетевые настройки разрешает трафик на 25 порту только до достижения шлюза электронной почты, и вынуждает клиентов в пределах сети использовать эту машину для исходящей почты. Межсетевая защита выполняет работу по блокированию портов, и Постфикс выполняет роль почтового шлюза.

Рисунок 13-1 показывает сервер приложений который передает все сообщения шлюзу электронной почты и наоборот. Шлюз электронной почты защищает сервер программного обеспечения от внешних клиентов, нападений, и серверы снаружи не могут соединиться непосредственно с сервером приложений.

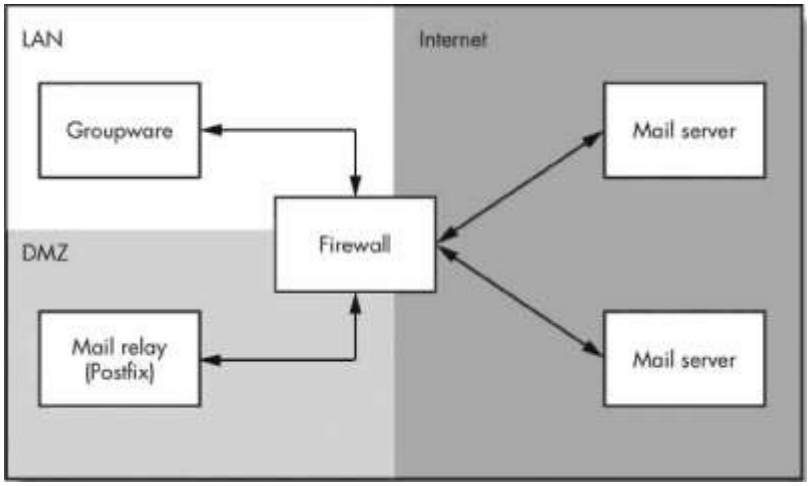



Рис 13-1 Почтовый шлюз для сервера приложений

*СОВЕТ, Вы можете расширить функциональные возможности шлюза электронной почты, добавив фозможности сканирования на вирусы или централизованного антиспам фильтра. При этом, Вы защищаете сервер приложений не только от злонамеренных соединений, но и от злонамеренного содержимого.*
*Если Вы обеспечиваете службу пересылки, почтовым клиентам эта глава может помочь Вам установить основной шлюз электронной почты. Вы можете расширить его возможности, добавив поддержку SMTP аутентификации (см. Главу 16), и TLS (см. Главу 18).*

## Основная настройка

Основная настройка шлюза электронной почты позволит Постфиксу работать на внешнем почтовом сервере и пересылать сообщения предназначенные для определенных доменов к другим (внутренним) почтовым серверам. Чтобы создать

такой шлюз электронной почты, Вы должны выполнить следующие шаги:

1. Позволить внутреннему серверу использовать шлюз как релей.
2. Настроить домены, для которых почта будет передаваться к внутренним серверам (relaydomains).
3. Установить хост,  для которому будет передана почта (relayhost).
4. Определить получателей почты для которых почта будет передана внутренним серверам

**Установка разрешения передачи на шлюзе**

Ваш первый шаг должен разрешить передавать почту шлюзе электронной почты вашему "скрытому" почтовому серверу. Добавьте адрес внутреннего почтового сервера к списку серверов в параметре mynetworks. Например, если адрес внутреннего сервера - 172.16.1.1, Вы могли бы поместить эту строку в main.cf файл шлюза электронной почты:

mynetworks = 127.0.0.0/8 172.16.1.1/32

Разрешение на пересылку ограничено  адресами localhost вашего шлюза электронной почты (127.0.0.1) и внутреннего почтового сервера (который является 172.16.1.1 в этом примере), так, что другие хосты внутри вашей сети не могут использовать данный шлюз как релей
.

**Установка релей домена  на шлюзе**

Следующий шаг должен указать Постфиксу принимать почту от внешней сети для хоста во внутренней сети. Постфикс использует relaydomains параметр, чтобы определить список доменов, для которых он пересылает почту, даже если он не конечный адресат для этих доменов. Например, если Вы хотите пересылать почту для example.com, используйте эти настройки:

relay_domains = example.com

**Установка внутреннего почтового хоста на шлюзе**

Теперь, когда шлюз знает, что он должен принять почту для определенного домена, Вы должны указать ему, куда передать поступающие сообщения для данного домена. Вы делаете это, создавая транспортную карту, которая является файлом, типа /etc/postfix/transport. Например, если Вы хотите перecskfnm сообщения для example.com хосту mail.office.example.com, файл мог бы выглядеть следующим образом

example.com     smtp:[mail.office.example.com]

В этой строке, smtp означает, что Постфикс должен использовать тип транспорта smtp, определенный в master.cf файле. Квадратные скобки важны, потому что они отключают поиски MX записи для mail.office.example.com . Без скобок, Постфикс искал бы MX запись для mail.office.example.com Поскольку эта запись вероятно принадлежит непосредственно данному хосту, он пробовал бы доставить почту самому себе, и входящие сообщения зациклятся. Теперь вы должны создать индексированный файл выполнить команду:

# postmap hash:/etc/postfix/transport

В конце, установите параметр transportmaps  в вашем main.cf файле следующим образом (и затем перезагрузите Постфикс):

transport_maps = hash:/etc/postfix/transport

***Определение получателей релея***

Что делает шлюз "умным"? Обычный шлюз принимает *любое* сообщение для *любого* получателя для домена пересылки, включая недопустимых получателей, которые не существуют на внутреннем почтовом сервере, который в конечном счете доставляет сообщения.

Учитывая явное количество спама и рекламы обращающийся в Интернете сегодня, и то что могут быть получатели, которым не позволено получать сообщения из вне (, типа общедоступных папок, внутренних списков рассылки, и возможно даже некоторых реальных людей), обычный шлюз может вызвать  проблемы, потому что он принимает все. В частности он передаст спам, и вирусы  несуществующим и неправомочным учетным записям.

Кроме того, принимая почту для несуществующего адреса, шлюз берет на себя ответственность за сообщение отправителю, если почту нельзядоставить. Это загрязняет почтовую очередь  сообщениями почтового демона, и может вызывать возврат почты людям в Интернете, которую они не посылали.

Когда Постфикс не принимает почту, то ответственность за сообщении о невозможности доставить почту лежит на клиенте. Когда этот клиент взломаная Windows, тогда никакая почта обратного рассеяния не будет послана вообще.

Умный хост знает, как отделить пшеницу от мякины, потому что он имеет список допустимых получателей на внутренних серверах. Используйте relay_recipient_maps параметр, чтобы определить и активизировать список допустимых пользователей. Например, если  ваше карты имеет имя - /etc/postfix/relayrecipients, Вы использовали бы эту запись в вашем main.cf файле:

relay_recipient_maps = hash:/etc/postfix/relay_recipients

*ПРЕДОСТЕРИЖЕНИЕ, Если вы определяете этот параметр, карта должна обеспечить список допустимых получателей . Иначе, ваш шлюз не будет иметь никакой надежности. Если Вы не можете обеспечить список, то отключите карту с установив ofrelay_recipient_maps =.*

Конечно, когда Вы говорите Постфиксу, где найти карту, Вы должны фактически обеспечить список. Если Вы определили вашу карту как в предыдущем примере, создайте текстовый файл, названный /etc/postfix/relay_recipients содержащий допустимых получателей. Например, следующий файл допускает пересылку сообщений для john@example.com  и linda@example.com:

john@example.com  OK

linda@example.com  OK

Если Вы хотите явно отрицать доступ пересылки для определенного получателя, используйте код ошибки и сообщение, типа 554 доставока, не разрешенна вместо OK.

Как с любой картой, Вы должны преобразовать карту которую Вы определили в relay-recipient_maps параметре в индексированный тип базы данных,. Например, выполните postmap hash:/etc/postfix/relay_recipients, чтобы сделать это.

*Примечание: Довольно просто создать список допустимых пользователей вручную, если Вы имеете только несколько пользователей на удаленном доступе мало изменяющихся время от времени. Более вероятно , тем не менее, что Вы будите имеете много пользователей, которые непрерывно изменяются, или Вы можете даже не знать список пользователей. См. hfpltk "Экспорт получателей из Active Directory " далее в этой главе, для обсуждения того, как автоматизировать процесс. В частности этот раздел описывает как экспортировать список получателей от сервера Microsoft Exchange 2003 .*

**Расширенная настройка шлюза**

Расширенный шлюз не только пересылает почту на другие серверы, но также и защищает локальные серверы от нападений и автоматизирует процесс обновления списка допустимых получателей в домене. Следующие разделы показывают несколько методик чтобы улучшить сервис, обеспечиваемый вашим шлюзом.

**Улучшение безопасности почтового шлюза**

Пока, ваш Postfix передает все сообщения обращенные example.com, внутреннему почтовому серверу, mail.office.example.com. Если это - единственная задача, которую ваш smarthost должен выполнять (то есть, если ваш smarihost не должен получать почту для локальных пользователей на smarthost), Вы должны отключить локальную доставку так, чтобы smarthost был не уязвим для злонамеренных сообщений, посланных локальным пользователям на smarthost.

1. Отмена локальной доставки. Первый шаг указать Постфиксу, он не конечный адресат, устанавливая параметр mydestination пустым:

```
mydestination =
```

2 . Отключите локальных получателей. Установите local_recipient_maps параметр пустым так, чтобы Постфикс был неспособен искать любого локального получателя:

```
local_recipient_maps =
```

3. Проброс требуемых локальных получателей. Когда Вы устанавливаете пустой local_recipientmaps параметр, сообщения всем локальным получателям заблокированы. Однако, Вы все еще должны сохранять почтовый шлюз RFC совместимым таким образом Вы должны установить адреса пересылки почты для postmaster и abuse, которые указывают на ваш внутренний почтовый сервер, Создайте карту, чтобы использовать ее как значение для virtual_alias_maps параметра (/etc/postfix/virtual, подходит прекрасно), и добавьте адреса для пересылки почты для этих двух получателей на внутренний почтовый сервер. Например, ваш файл карты мог бы выглядеть следующим образом:

```
postmaster          postmaster@example.com
abuse               abuse@example.com
```

Постройте индексированную карту из, файла с помощью postmap hash:/etc/postfix/virtual и сошлитесь в вашем main.cf на нее:

virtual_alias_maps = hash:/etc/postfix/virtual

Создайте сообщение об ошибке локальной доставки. Когда Вы отключаете локальную доставку, Вы должны также сообщить любому клиенту, пытающемуся послать сообщение smarthost`у , что Вы отключили доставку локальным получателям. Чтобы сделать это, определите специальный локальный транспорт с помощью параметра local_transport, который передает сообщение об ошибке. Например, следующая строка посылает все локальные сообщения демону ошибки, который обеспечит соответствующее сообщение об ошибке.

local_transport = error:local mail delivery is disabled

Переадресуйте ответы локальным службам. Если Вы следовали за шагами в предыдущих разделах, Постфикс теперь не будет принимать почту для локальных пользователей кроме postmaster и abuse. Однако, локальные службы, типа cron, использующие Постфикс, чтобы послать сообщения о состоянии администраторам и пользователям все еще отсылают почту, используя адреса отправителя, связанные с именем хоста машины. Это может вызвать конфликт, потому что Вы не можете ответить на эти сообщения.
Чтобы препятствовать пользователям посылать ответ этим приложения, измените значение параметра myorigin, который Постфикс добавляет как доменную часть адресов электронной почты. Установите myorigin равным домену, который фактически имеет почтовый сервер и который имеет почтовые ящики или псевдонимы для этих отправителей. Например, если внутренний почтовый сервер, который является окончательным адресатом для example.com , может обеспечить эту службу, Вы могли бы использовать эту установку:

myorigin = example.com

Отключение локального агента доставки. Наконец, Вы можете препятствовать демону master запускать локальный агент доставки - это фактически выключает локальный агент доставки, потому что нет никаких получателей на этой машине. Отредактируйте ваш файл master.cf, и закомментируйте строку, содержащую локальную службу, поместив # перед строкой, как показано здесь:

```
#==========================================================================
# service type  private unpriv  chroot  wakeup maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       n       -       -       smtpd
#local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       n       -       -       lmtp
anvil     unix  -       -       n       -       1       anvil
```

**Использование Постфикс с сервером Microsoft Exchange**

Сервер Microsoft Exchange, без сомнения, мощное серверное программное обеспечении для коллективной работы, но его безопасность и стабильность, под атаками не так хороши. Поэтому, много администраторов используют его функциональные возможности программного обеспечения для коллективной работы с Постфикс шлюзом. Этот раздел обсуждает, как Вы можете предоставить Постфикс почтовому шлюзу список допустимых получателей и как автоматизировать процедуру,
Самое легкое и самое обычное решение того, чтобы заставлять Постфикс и Microsoft Exchange взаимодействовать состоит в том, чтобы Постфикс релей хост выполнял запрос сервера Microsoft Exchange, используя LDAP (Легкий Протокол Доступа Каталогов). Реле хост делает запрос Microsoft Exchange каждый раз, когда поступает сообщение чтобы определить существует ли получатель.
Однако, этот подход содержит риск и ограничения. Альтернатива состоит в том чтобы иметь список получателей на сервере Microsoft Exchange, помещенный в список получателей на Постфикс сервере, что лучше в следующих отношениях:

**Безопасность**
Независимо от того, какой пакет выполняется у Вас на вашем внутреннем почтовом сервере, Вы хотите сохранить его в максимально возможной степени безопасности. Именно поэтому Вы помещаете его позади межсетевой защиты. Одно из основных правил безопасности состоит в том, чтобы разрешить только, что нужно разрешить и запретить все остальное.
Первый порыв многих администраторов систем состоит в том, чтобы Постфикс использовал LDAP запросы, чтобы опрашивать удаленный сервер Microsoft Exchange о допустимых получателях. Чтобы сделать это, Вы должны открыть порт 389 (TCP/UDP) на сервере Microsoft Exchange, разрешив подключения от Постфикса. Это относительно просто, но это открывает порт для вашей внутренней сети. Отключаемое соединение более безопасно, Microsoft Exchange , предоставляет Постфиксу список допустимых получателей только, когда список получателей изменяется. С администратором Microsoft Exchange, помещающего этот список в smarthost с помощью scp или rsync, Вы не должны открывать порт от демилитаризованной зоны до LAN.

**Быстродействие**

LDAP запросы более медленны по сравнению с индексированными картами, которые использует Постфикс. Если Вы предоставляете Постфиксу статический список допустимых}получателей, smarthost может обработать сообщения очень быстро.

**Стабильность**

smarthost существует, чтобы защитить внутренний почтовый сервер, и защита может сойти на нет, когда smarthost под атакой опрашивает внутренний сервер. Это может случиться, потому что спаммеры используют атаку по словарю, чтобы послать сообщения большому количеству получателей сразу, и это вынудило бы почтовый релей, посылать большое количество запросов на сервер используя LDAP, который он должен защитить, опрашивая о допустимых получателях. Это замедлило бы (если не отключило), Active Directory, и таким образом Exchange сервер, превратив нападение по словарю в DoS атаку. Если почтовый сервер должен загрузится, Вы

хотите, чтобы это произошло на внешнем сервере, а не на внутреннем почтовом сервере.
В этом разделе Вы пошлете список допустимых получателей с Сервера Exchange 2003 Постфикс почтовому релею, следуя этим шагам:

1. Экспортировать список всех допустимых получателей.
2. Скопировать список на почтовый релей.
3. Извлечь, допустимых получателей из списка.
4. Создать карту получателей на релее.
5. Индексировать карту получателей релея.
6. Автоматизировать этот процесс

### Экспорт списка всех допустимых получателей Microsoft Exchange из Active Directory

Microsoft использует признак *proxyAddresses* в ее Active Directory, чтобы хранить допустимые адреса получателей для Exchange . Простой способ экспортировать *proxyAddresses* из Microsoft Active Directory состоит в том, чтобы использовать csvde, инструмент командной строки, доступный из любого Exchange сервера - это не требует, чтобы Вы использовали самописный сценарий. Например, чтобы экспортировать значения для *proxyAddresses* в файл с именем C:\export\example_com_recipients.txt, Вы можете просто использовать эту команду в окне командной строки:

**C:\> csvde -m -n -g -f "C;\export\example_com_recipients,txt" **
**-r "(|(&(objectClass=user)(objectCategory=person)) **
**(objectClass=groupOfNames) (objectClass=msExchDynamicDistributionList))" **
**-1 proxyAddresses**

***Совет:*** *Существует тысячи способов организовать и структурировать Active Directory, таким образом может быть трудно найти - имя объекта, который Вы должны экспортировать из вашего Active Directory. Установка Exchange содержит опцию, чтобы установить несколько инструментальных средств поддержки, включая модуль ADSI Edit. Добавьте его к вашей MMC (оснастки управления Microsoft). С этим модулем запуск mmc.exe из командной строки дает Вам полный доступ к именам объектов в Active Directory.*

Вывод предыдущей команды содержит более подробную информацию чем нужно Постфиксу. Например, Вы могли бы получить это в выходном файле:

DN,proxyAddresses

"CN=Administrator,CN=Users,DC=example,DC=com",smtp;abuse@example.com;SMTP:\

Administrator@example.com; X400:c=DE\;a= \;p=Example Corporat\; o=Exchange\;s=Administrator\;

;smtp: postmaster@example.com

"CN=Gast,CN=Users,DC=example,DC=com", "CN=SUPP0RT_388945a0,CN=Users,DC=example,DC=com",

"CN=krbtgt,CN=Users,DC=example,DC=com", "CN=IUSR_MAIL,CN=Users,DC=example,DC=com")

,,CN=IWAM_MAIL,CN=Users,DC=example,DC=com",

"CN=WilmaPebble,OU=purchasing,DC=example,DC=com``,smtp:wilmapebble@example.com;

smtp:\

wilma@example.com;smtp:wilma.pebblegexample.com;SMTP:w.pebble@example.com;smtp:\

pebble@example.com;  X400:c=DE\;a= \;p=Example Corporat\;o=Exchange\;s=Pebble\;g=Wilma\;
"CN=BettyMcBricker,OU=purchasing,DC=example,DC=comi\smtp:mcbricker@example.com;smtp:\
bettymcbricker@example.com;smtp:betty@example.com;smtp:betty.mcbricker@example.com;\
SMTP:b.mcbrickergexample.com;X400:c=DE\;a= \;p=Example Corporat\;o=Exchange\;\
s=McBricker\;g=Betty\;
"CN=Fred Flintstone,OU=sales,DC=exampleJDC=com",smtp:fredflintstone@example.com;\
SMTP:fred.flintstone@example.com;smtp:f.-fliritstorie@example.com;smtp:fred@example.com;\
smtp:flintstone@example.com;X400:c=DE\;a= \;p=Example Corporat\;o=Exchange\;\
s=Flintstone\;g=Fred\;
"CN=Barney Rubble,OU=sales,DC=example,DC=com",SMTP:barney.rubblegexample.com;\
sntp:barneyrubble@exanple.con;  smtp:rubble@example.com;snt:p:barney@exanple.con;smtp:\
b.rubble@example.com;X400:t>DE\;a= \;p=Exanple Corporat\;o=Exchange\;s=Rubble\;g=Barney\; "CN=Bamn
Bafnm,OU=it,DC=example,DC=com",smtp:bamfnbafnmgexample.com;sfntp:\
bainn(dexample.com;smtp: bamm.b a mmljie xample.com; SMTP :b.bamn$example. com ;\
X400:c=DE\;a= \;p=Exanple Corporat\;o=Exchange\;s=Bamm\;g=Bamm\; ``CN=SystenMailbox{C5C3EAFC-
A32F-4925-85A5-3Co8709DE6i7},CN=Microsoft Exchange System\
Objects, DC=example,DC=com",,  SMTP:SystemMailbox{C5C3EAFC-A32F-4925-85A5-3CO8709DE617}\
(Bexanple.com;X400:c=DE\;a= \;p=Exanple Corporat\;o=Exchange\;s=SystemMailbox?\
C5C3EAFC-A32F-4925-85A5-3C\;
"CN=it-department,OU=it,DC=example,DC=com",SMTP:it-department@example.com;\
X400:c=DE\;a= \;p=Example Corporat\;o=Exchange\;s=it-department\; "CN=purchasing-
department,OU=pijrchasing,DC=example,DC=com",SMTP:purchasing-departnent@exanple.com;\
X400:c=DE\;a= \;p=Exanple Corporat\;o=Exchange\;s=purchasing-department:\; "CN=sales-
department,OU=sales,DC=example,DC=com",SMTP:sales-department@example.com;\
X400:c=DEN;a= \;p=Example Corporat\;o=Exchange\;s=sales-department\;

### Посылка списка получателей на почтовый релей

Есть много способов скопировать файл от вашего Microsoft Exchange  до вашего smarthost, но среди лучшего - безопасная копия (scp), шифрующая автоматизированная утилита, поддерживаемая и Windows и Unix.

1. Получите клиента безопасного копирования (scp) для Windows; например, Putty.
2. Создать пользователя копирования на smarthost.
3. Создать идентификационные ключи.
4. Добавить публичные ключи к идентификационным ключам.
5. Скопировать приватный ключ в Windows.
6. Преобразовать ключ SSH в формат ключа PuTTY.
7. Скопировать экспортный файл к smarthost

Получение клиента безопасной копии для Windows

Среди многих клиентов, которые позволяют, Вамиспользовать scp, чтобы скопировать файлы с хоста Windows - Putty, бесплатная Telnet и SSH клиент. Вы

можете загрузить его с http://www.chiark.greenend.org.uk/~sgtaham/putty .

Вы должны загрузить pscp.exe и puttygen.exe из этого пакета, чтобы выполнить операции, требуемые в этом примере. Скопируйте выполняемые файлы в системную папку Windows, например C:\Windows.

Создание Пользователя копирования на Smarthost

Чтобы выполнять передачу файла, создайте пользователя на вашем smarthost. Эта учетная запись будет служить только, чтобы получить экспортируемый список получателей. Например, Вы могли создать пользователя по имени e3k с помощью этой команды:

**# useradd e3k**

После создания пользователя, установите его пароль, используя команду passwd. Вы будете использовать пароль в течение процесса установки, но Вы можете отключить его, когда все выполняется гладко.

Создание идентификационных ключей

Следующий шаг должен создать ряд идентификационных ключей так, чтобы Вы не нуждались в пароле, чтобы передать файлы от сервера Windows до smarthost. Как root на smarthost, выполните **su - e3k**, чтобы перейти к пользователю e3k и выполнить ssh-keygen, чтобы создать ключи: Например, Вы можете выполнить следующую команду как e3k:

$ ssh-keygen -t rsa

Generating public/private rsa key part.

Enter file in which to save the key (/home/e3k/.ssh/id_rsa):

Created directory `home/e3k/.ssh'.

Enter passphrase (empty for no passphrase): **1**

Enter same passphrase again:

Your identification has been saved in /home/e3k/.ssh/id_rsa.

Your public key has been saved in /home/e3k/.ssh/id_rsa.pub.

The key fingerprint is:

17:7e:78:9e:39:0e:O4:b7:ee:6d:39:28:c6:21:e4:84 e3k@mail.example.com

**1** Не вводите passphrase, если Вы хотите, чтобы процесс копирования выполнился автоматический. Если Вы введете passphrase, то Вы будете должны использовать его, копируя экспортный файл на smarthost.
Предыдущая команда создает два файла: .ssh/id_rsa и .ssh/id_rsa.pub. Первый - приватный ключ; не позволяйте его никому (никакой хост не должен иметь его кроме вашей Windows машины).
Добавление публичного ключа к списку разрешеннных ключей.
Теперь, когда Вы имеете ключи, Вы должны сообщить вашему SSH серверу о публичном ключе, который Вы только что создали. Чтобы добавить публичный ключ

из файла id_rsa.pub к списку в $home/.ssh/ outorized_keys файлт для вашего пользователя копирования (e3k), выполните следующую команду:

**$ cd .ssh**
**$ cat id_rsa.pub >> authorized_keys**

После создания файла authorized_keys, удостоверьтесь, что он имеет правильные разрешения, или иначе файла сервер SSH не использовать файл для идентификации.

**$ chmod 644 authorized_keys**
**$ ls -1 authorized_keys**
**-rw-r--r-- 1 e3k e3k 230 May 13 10:38 authorizedkeys**

### Копирование приватного ключа в Windows

Затем Вы должны скопировать приватный ключ от smarthost на ваш Windows хост. Самый легкий способ сделать это состоит в том, чтобы использовать команду pscp.exe на машине Windows. Если бы IP адрес вашего smarthost был бы 172.16.1.1, то Вы выполнили бы эту команду в командной строке:

**> C:\export> pscp e3k@l72.16.1.l:/home/e3k/.ssh/id_rsa .**
**e3k@172.16.1.1's password:**
**id_rsa | 0 kB | 0.9 kB/s | ETA: 00:00:00 | 100%**

Используйте пароль, который Вы задали для пользователя в разделе "Создание пользователя копирования на Smarthost".

Преобразование ключа SSH в формату ключа PuTTY

PuTTY использует формат отличный чем большинство Unix пакетов SSH используют, чтобы хранить публичные и приватные ключи, таким образом Вы должны будете преобразовать приватный ключ в собственный формат PuTTY,
puttygen утилита может преобразовать ключи; выполните это из командной строки следующим образом:

**C:\export> puttygen id_rsa**

Эта команда запускает графический клиент, который загружает приватный ключ. Вы должны увидеть диалоговое окно как на рисунке 13-2.


Рис 13-2 Успешное импортирование ключа с помощью puttygen.

Нажмите OK, чтобы подтвердить сообщение. Появится диалоговое окно PuTTY Key Generator, показанное на рисунке 13-3 . Дайте преобразованному ключу имя

(например example_com.ppk), и щелкните на кнопке сохранения приватного ключа.


Рис 13-3 сохранение приватного ключа в PuTTYgen

Как показано на рисунке 13-4, генератор ключей предупредит Вас о пустом пароле в ключе. Щелкните Yes, чтобы сохранить ключ без пароля и сохраните приватный ключ. Теперь ваш Windows хост готов использовать scp, чтобы передавать файлы smarthost`у с использованием идентификационных ключей.

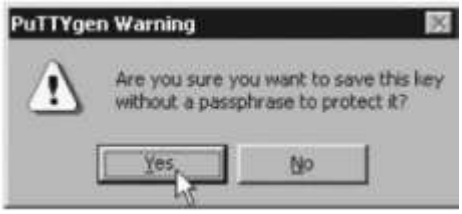



Рис 13-4 PuTTY предупреждает о пустом пароле в ключе

Копирование списка Получателей на Smarthost

Сохраните ваше окно командной строки открытым, чтобы скопировать файл от вашего Windows хоста на smarthost, используя  утилиту pscp.exe (scp, принадлежащая PuTTY). Когда Вы выполняете pscp, Вы должны пердоставить идентификационный приватный ключ, файл, который будет скопирован, и пользователя, который выполняет копирование. В нашем примере, приватный ключ находится в файле example_com.ppk, файл для копирования example_com_recipients.txt, и пользователь - e3k. Чтобы поместить файл в/home/e3k, Вы использовали бы эту команду:

**C:\export> pscp -i example_com.ppk example_com_recipients.txt e3k@172.16.1.1:/home/e3k/**
Authenticating with public key "rsa-key-20040512"

example_com_recients.txt | 2 kB | 2.4 kB/s | ETA: 00:00:00 | 100%

После успешного копирования файла на smarthost, Вы можете сохранить команду экспорта csvde описанную ранее в разделе"Экспорт допустимых получателей из Active Directory" и предыдущую команду pscp в bat файл по имени export_valid_recipients.bat. Тогда Вы можете выполнить его щелчком мыши всякий раз, когда Вы создаете, изменяете, или удаляете получателя. Файл выглядил бы примерно так:

```
csvde -m -n -g -f "C:\export\example_com_recipients.txt" \
 -r "(|(&(objectClass=user)(objectCategory=person)) \
(objectClass=groupOfNames) (objectClass=msExchDynamicDistributionlist))" \ -l proxyAddresses
pscp -i examplecom.ppk example_com_recients.txt e3k@172.16.1.1:/home/e3k/
```

После проверки, что этот bat файл работает, Вы можете отключить пароль пользователя копирования на smarthost используя команду, типа usermod-L e3k так, чтобы удаленный доступ к учетной записи e3k был возможен только с идентификационным ключом.

Построение карты получатилей

Теперь вы имеете файл экспорта из Active Directory на вашем smarthost`e, таким образом Вы можете извлечь получателей из файла с помощью сценария. Есть две вещи, которые необходимо помнить, делая это:

- Microsoft использует и SMTP и smtp, чтобы обозначить адреса получателя, таким образом сценарий должен охватывать оба варианта,
- Ваш сценарий должен быть в состоянии очистить несколько получателей, которые не должны получать почту из вне. Пример - почтовый ящик SystemMailbox, используемый Exchange для внутренних соединений.

Следующий сценарий, названный extract_valid_recipients, извлекает всех допустимых получателей и помещает их в файл, но он не включает получателей, перечисленных в файле черного списка.

```
#!/bin/sh
#  Extract all addresses that start with SMTP or smtp from
#  an Active Directory export,  but omit those that are listed in blacklist
cat $1  |  tr -d \»  |  tr ,  \\n  |  tr \;  \\n  |  awk -F\:   `/(SMTP|smtp):/ {printf(«%s \tOK\n,,,$2)}' | \
grep -v -f blacklist > $2
```

Файл черного списка выглядит следующим образом:

```
Administrator
SystemMailbox
```

Выполните команду extract_valid_recipients, чтобы выполнить сценарий, и он сгенерирует список допустимых получателей в файл relay_recipients.

**extract_valid_recipients /home/e3k/example_com_recipients.txt relay_recipients**

Вывод будет подобен этому:

abuse@example.com OK
postmaster@example.com OK
wilmapebble@example.com OK
wilma@example.com OK
wilma.pebble@example.com OK
w.pebble@example.com OK
pebblegexample.com OK
mcbricker@example.com OK
bettymcbrickergexample.com OK
betty@example,com OK
betty.mcbricker(3example.com OK
b.mcbricker@example.com OK
fredflintstonegexample.com OK
fred.flintstonegexample.com OK
f.flintstone@example.com OK
fred@example.com OK
flintstone@example.com OK
barney.rubblegexample.com OK
barneyrubble@example.com OK
rubblegexample.com OK
barneygexample.com OK
b.rubblegexample.com OK
bammbammgexample.com OK
bamm@example.com OK
bamm.bamm@example.com OK
b.bammgexample.com OK
it-departmentgexample.com OK
purchasing-departmentgexample.com OK
sales-department@example.com OK

Если этот вывода правильный, преобразуйте его используя postmap (например, командой подобной этой: postmap hash:relay_recipients), и переместите его в католог, на который указывает Ваш параметр relay_recipient_maps parameier (он обсуждался ранее в разделе "Определение получателей релея). Например, Вы можете использовать следующую команду:

# mv relay_recipients.db /etc/postfix/relay_recipients.db

***ПРЕДОСТЕРЕЖЕНИЕ не указывайте в relay_recipient_maps непосредственно на вашу недавно созданную карту relay_recipients (например, hash:/home/e3k/rela_recipients)! Постфикс остановил бы обслуживание, если бы преобразование карты прошло неудачно. Безопасный путь состоит в том, чтобы преобразовать карту сначала, и только если преобразование пройдет успешно переместить ее к местоположению, на которое указывает relay_recipient_maps.***

**Построение Карты sender_access**

В качестве бонуса, ваш файл экспорта из Active Directory может также дать Постфиксу список отправителей, которым разрешено послать почту во внешний мир. Чтобы сделать это, Вы можете написать скрипт как в «разделеПостроение карты получателей», но с одним отличием: Microsoft использует SMTP, чтобы обозначить допустимые адреса отправителей, таким образом сценарий извлечения адреса должен обработать только элементы с этой пометкой.

*ОТМЕТЬТЕ, Это полезно для того, чтобы препятствовать вирусам использовать список контактов Outlook Express, чтобы формировать. ложного отправителя и затем посылать самого себя из вашей сети.*

Вы можете назвать сценарий extract_valid_senders, и он должно выглядеть следующим образом

```
#!/bin/bash
# Extract all addresses that start with SMTP from an Active Directory
# export, but omit those that are listed in blacklist
cat $1 | tr -d \» | tr , \\n| tr \; \\n | awk -F\: '/SMTP:/ {printf(,%s\tOK\n",$2)}, | \
grep -v -f blacklist > $2
```

На сей раз, когда Вы выполняете следующую команду, Вы должны получить более короткий список чем прежде, потому что нет никаких псевдонимов:

**# ./extract_valid_senders /home/e3k/example_com_recipients.txt example_com_senders**

Вывод должен выглядеть как следующий основанный на более раннем примере:

```
w.pebblegexample.com OK
b.mcbricker@example.com OK
fred.flintstone@example.com OK
barney.rubble@example.com OK
b.bammgexample.com OK
it-department@example.com OK
purchasing-department@example.com OK
sales-department@example.com OK
```

В предыдущем примере, Вы переадресовывали вывод в файл, названному example_com_senders. Теперь создайте индексированную базу данных из этого файла с помощью команды postmap hash:example_com_senders Затем создайте парвило (см. Главу 8), которое проверяет отправителей конверта с этими ограничениями:

- Если почта исходит из внутреннего сервера, она должна содержать один из допустимых адресов отправителя конверта.
- Если почта не исходит от внутреннего сервера, ограничение не применяется,

Чтобы Постфикс, применил это условное ограничение, определите класс ограничений, который вызывает ограничение отправителя конверта, когда почта исходит от внутреннего почтового сервера. Например, ваш main.cf файл мог бы

содержать следующее:

```
smtpd_restriction_classes =
        must_be_valid_sender                    1
must_be_valid_sender =                          2
        check_sender_access hash:/etc/postfix/example_com_senders
        reject
smtpd_recipient_restrictions =
check_client_access hash:/etc/postfix/example_com_ip      3
reject_unauth_destination
```

**1** must_be_valid_sender - имя ограничения, которое содержит подмножество ограничений, которые применяются когда почты поступает от внутреннего почтового сервера; эта строка просто перечисляет классы ограничений.
**2** определение этого ограничение содержит процедуру для сообщений, исходящих от внутреннего сервера: сначала проверяется список допустимых отправителей конверта, и отклоняется любого другой отправитель конверта.
check_client_access вызывает выполнение must_be_valid_sender класса ограничений,

Наконец, Вы должны добавить IP адрес вашего внутреннего почтового сервера в файл the/etc/postfix/example_com_ip файлу, наряду с действием ограничения, которое будет предпринято. Например, следующая строка определяет, что, если сообщение исходит от 172.16.1.1, Постфикс должен применить must_be_valid_sender ограничение:

```
172.16.1.1      must be valid sender
```

После добавления этих опций в конфигурацию, Вы должны перезагрузить Постфикс, чтобы изменения вступили в силу.

**Автоматизация процесса построения карты**

Вы можете автоматизировать процесс построения карт на smarthost. Следующий пример использует Makefile, который Вы можете загрузить с сайта книги Постфикса http://www.postfix-book.com.

```
## Makefile to automate map build process

## configuration settings

# Location of the file we extract the data from

ADS_DUMP=/home/e3k/example_com_recipients.txt

# Location of the .proto files

PROTO_PATH=relay_recipients

PROTO_PATH2=valid_senders

# destination of successfully built maps

MAP_PATH=/etc/postfix/relay_recipients

MAP_PATH2=/etc/postfix/valid_senders

# type and suffix of the maps to build

DB_TYPE=hash
```

```
DB_SUFFIX=db
## Makefile options
#
# build all maps
all:            $(MAP_PATH).$(DB_SUFFIX) $(MAP_PATH2).$(DB_SUFFIX) blacklist
# extract valid recipients from $(ADS_DUMP) to $(PR0TO_PATH).proto
$(PROTO_PATH).proto:    $(ADS_DUMP)
                        ./extract_valid_recipients $(ADS_DUMP) $(PROTO_PATH).proto
# extract valid senders from $(ADS_DUMP) to $(PROTO_PATH2).proto
$(PR0T0_PATH2).proto:   $(ADS_DUMP)
                        ./extract_valid_senders $(ADS_DUMP) $(PR0TO_PATH2).proto
# build map of valid recipients from $(PROTO_PATH).proto
$(MAP_PATH).$(DB_SUFFIX): $(PR0TO_PATH).proto
                        /usr/sbin/postmap -w $(DB_TYPE):$(PROTO_PATH).proto && \
                    mv $(PROTO_PATH).proto.$(DB_SUFFIX) $(MAP_PATH).$(DB_SUFFIX)
# build map of valid senders from $(PR0T0_PATH2).proto
                        $(MAP_PATH2).$(DB_SUFFIX): $(PROTO_PATH2).proto
                    /usr/sbin/postmap -w $(DB_TYPE):$(PROTO_PATH2).proto && \
                mv $(PROTO_PATH2).proto.$(DB_SUFFIX) $(MAP_PATH2).$(DB_SUFFIX)
#  remove all proto maps
clean:
rm -f $(PR0T0_PATH).* $(PR0T0_PATH2).* *~
```

Как только Вы успешно выполнили,make, чтобы проверить, что преобразование работает, Вы можете создать в cron задание, чтобы выполнять его автоматически. Например, следующее задание в вашем crontab файле выполнилась бы каждые 15 минут:

```
0,15,30,45 * * * * cd /root/relay_recipients && /usr/bin/make
```

## Настройка взаимодействия Exchange и Постфикса

Этот раздел объясняет, как сконфигурировать Microsoft Exchange, чтобы передать всю почту через ваш Постфикс шлюз, и также как сконфигурировать Excange так, чтобы это случайно не «затопил» ваш Постфиксный сервер.
По умолчанию Exchange,  не передает исходящие сообщения к шлюзу. Чтобы инициализировать конфигурацию для релея, выполните следующие шаги:

1. Запустите системный менеджер Exchange в меню Программы.
2. Выбрите Сервер из древоподобного меню слева.
3. Выбрите ваш почтовый хост из меню
4. Выбрите Протоколы из меню Hosts.
5. Выбрить SMTP в меню Протоколы.
6. Щелкнете правой кнопкой мыши по меню SMTP, и выбрите Свойства изменю Виртуалный SMTP сервер значения по умолчанию.
7. Выберите вкладку доставка в окне свойств значения по умолчанию виртуального SMTP сервера.

Теперь Вы готовы закончить настройку. Шаги описаны в следующих разделах.
Установка Постфикс Сервера как Smarthost

Первое необходимое действие - выбора конфигурации Excange, чтобы послать все экспортные сообщения вашему Постфикс шлюзу. Выберите Расширенную доставку во вкладке Delivery, и введите , полностью определенное доменное имя вашего Постфикса smarthost'a ( например postfix.example.com ) показано на рисунке 13-5.

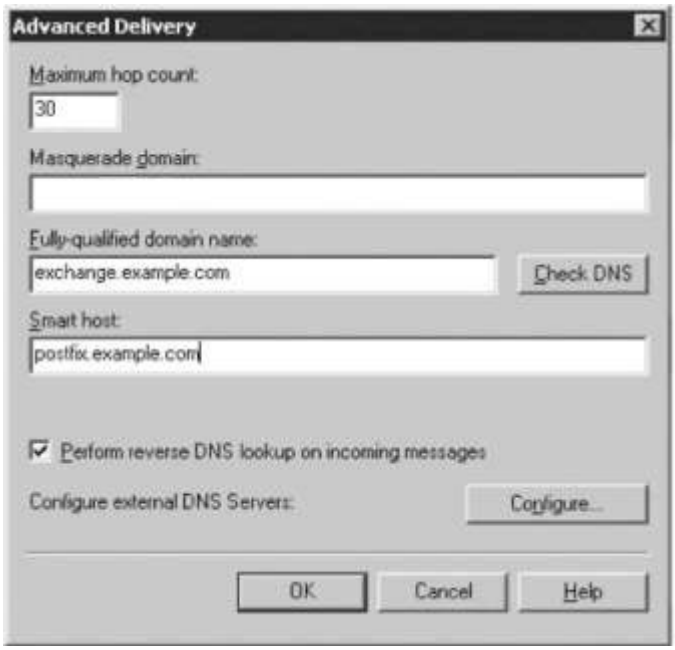



Рис 13-5 настройка местоположения шлюза в Exchange 2003

***ОТМЕТЬТЕ**, что Exchange не принимает IP адрес как значение для поля smarthost. Или добавьте smarthost к (внутренней) ДНС , к которому обращается Exachnge сервер, или установите tujo статически d файле hosts на Вашем Exachnge сервере.*

После установки полностью определенного доменного имени smarthost, Вы должны остановить и запустить виртуальный SMTP сервер по умолчанию вашего Exachnge сервера, чтобы изменения вступили в силу.

Ограничение исходящих связей.

Пока, Вы приложили много сил для защиты внутреннего почтового сервера от грубого поведения шлюза. Следующий шаг служит, чтобы защитить, Постфикс от того, чтобы быть перегруженным исходящими сообщениями сервера Exchange. Все, что Вы должны сделать, - уменьшить предел одновременных исходящих связей.

***ОТМЕТЬТЕ**, Что по умолчанию Exchange устанавливает предел - 1 000 одновременных связей. Если исходящая очередь почты на шлюзе получит так много сообщений (что, вероятно, может произойти в большой сети после перезагрузки службы SMTP Exachange), это может вызвать большую нагрузку шлюза в течении нескольких минут, и это может занять слишком много ресурсов, особенно если шлюз обслуживает не один внутренний почтовый сервер.*
*В этом случае лучше заставить Exachnge сервер, обрабатывать его исходящую почтовую очередь в более медленном темпе. Если Вы не имеете доступа к серверу, Вы можете также использовать механизмы Постфикс, описанные в Главе 21, чтобы ограничить число одновременных связей от хоста.*

Выберите исходящие связи из вкладки доставки окна Virtual SMTP Server По умолчанию, и установите предел 50 одновременных связей, как показано на рисунке 13-6. Это, хорошая установка при обычных обстоятельствах.

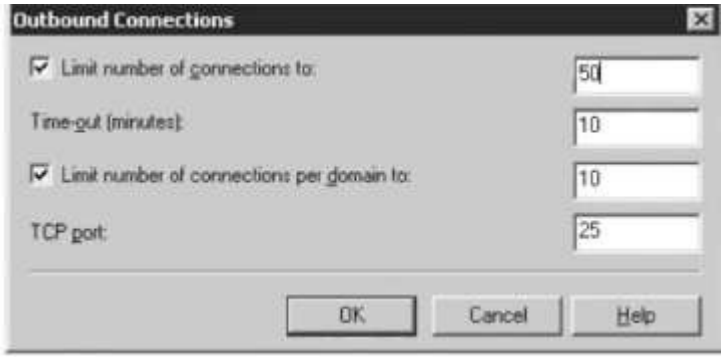



*Рис 13-6 Ограничение числа одновременных сообщений в Exchange*

Настройка NAT

Постфикс почтовый сервер позади NAT шлюза сталкивается с проблемами, потому что NAT шлюз изменяет IP пакеты (заменяет адрес назначения) прежде, чем передает их к Постфиксу. Это означает, что smtpd будет слушать только приватный адрес IP шлюза, в то время как шлюз принимает подключения к "официальному" IP адресу.
Это - только проблема, потому что почтовые RFC требуют, чтобы почтовый сервер принял почту, посланную postmaster@ [адрес], где адрес -IP адрес . Некоторые черные списки посылают информацию о прекращении котировки только по этому адресу.
Вы можете настроить Постфикс, чтобы принять почту к postmaster@[adress] с помощью использования параметра proxy_interfaces. Даже если Постфикс только будет слушать внутренний адрес IP, то это примет почту, к которой обращаются user@ [адрес]. Если бы ваш NAT шлюз имеет адрес 192.0.34.166, Вы использовали бы эту установку:

```
proxy_interfaces = 192.0.34.166
```

# ПОЧТОВЫЙ СЕРВЕР ДЛЯ НЕСКОЛЬКИХ ДОМЕНОВ

Постфикс может послать, получать, и хранить сообщения для больше чем одного домена при использовании любого из двух различных методов. Первый метод использует *virtual alias domains* (виртуальные псевдонимы доменов), которые просто разворачивает число{номер} доменов{областей}, для которых сервер является конечным{заключительным} адресатом. Второй метод использует *virtual mailbox domains* (виртуальные почтовые ящики доменов) ящика и более совершенен, потому что виртуальные почтовые ящики доменов не нуждаются в локальных учетных записях. Эта глава покажет Вам, как использовать оба этих подхода, предлагающих службу SMTP более чем одному домену.

### Виртуальные псевдонимы доменов

Обычно, Постфикс распознает себя как конечного адресата только для доменных имен определенных параметром mydestination[1] . Домены{области}, перечисленные в mydestination называют каноническими доменами, потому что они обычно перечисляют все имена локальной машины (и возможно ее родительское доменной имя).
В этой главе мы опишем множество методов, чтобы сделать Постфикс конечным адресатом для дополнительных доменов. Эти дополнительные домены называют виртуальными, потому что они не имеют никакого отношения к собственному имени машины.
Чтобы настроить основное обслуживание для виртуального псевдонима домена , Вы должны выполнить следующие шаги:

1. Установить имя виртуального псевдонима домена.
2. Создать карту адресов получателя.
3. Настроить Постфикс, чтобы он получал почту для виртуальных псевдонимов доменов.
4. Проверить новую конфигурацию.
5. Расширеное отображение.

Эти шаги описаны в следующих разделах.

### Установка имени виртуального псевдонима домена

Ваш первый шаг должен указать Постфиксу, что он является конечным адресатом для домена в дополнение к системному значению по умолчанию. Постфикс использует virtual_alias_domains параметр, чтобы определить карту виртуальных доменов. Чтобы использовать этот параметр, создайте файл карты, типа /etc/postfix/virtual_alias_domains, содержащий виртуальные домены в таком формате:

```
# virtual alias domains
postfix-book.con          20021125
```

В предыдущем примере, число справа - дата создания домена, но Вы можете установить здесь то, что хотите. Постфикс не использует правую часть, при просмотре карты для параметра virtual_alias_ domains, но Постфикс карты всегда требуют наличия и правой и левой частей.

---

1Постфикс также распознает себя, конечный адресатом для адреса в форме user@[ipaddress] где ipaddress один из списка IP адресов Постфикса.

После создания файла, преобразуйте его в индексированную карту с помощью этой команды:

# postmap hash;/etc/postfix/virtual_alias_domains

*ОТМЕТЬТЕ, Если Вы перечисляете домен, как виртуальный, не используйте ето как значение для вашего параметра mydestination, потому что неожиданные вещи могут случиться. Постфикс не знал бы, должен ли он доставить почту локально или отослать ее для виртуальной перезаписи. Именно поэтому Постфикс будет жаловаться громко на, эту конфигурацию в лог файле.*

Создание карты адресов получателя

Следующий шаг в настройке виртуального псевдонима домена должен создать файл /etc/postfix/virtual_alias_maps, чтобы преобразовать получателя виртуального домена псевдонима в локальный адрес получателя. Следующий пример включает единственных и многократных получателей.

# postfix-book.com

postmaster@postfix-book.com ralf@example.com

abuse@postfix-book.com abuse@example.com, patrick@example.com

ralf@postfix-book.com ralf@example.com

patrick@postfix-book.com patrick@example.com

Убедитесь, что Вы включили адресатов для postmaster и abuse , потому что RFC требуют чтобы все домены, имели получателей для этих адресов,

***ПРЕДОСТЕРЕЖЕНИЕ*** *Всегда полезно использовать полностью определенные доменные имена в ваших адресах получателей на правильной части вашего файла virtual_alias_maps файла. Иначе Вы оставляете слишком большое место для интерпретации. Если Вы определите только локальную часть (например, lalf), то Постфикс демон trivial-rewrite добавит доменую часть, определенную в параметре myorigin. Адрес user@$myorigin, возможно, не правилен, в зависимости от значений myorigin и адресата,*

Создайте индексированную карту из данного файла с помощью этой команды:

**# postmap hash:/etc/postfix/virtual_alias_maps**

Настройка Постфикса, для получения почты для виртуальных псевдонимов доменов

Теперь, когда Вы имеете обе карты, Вы должны настроить Постфикс, чтобы получать почту для вашего виртуального домена согласно записям в карте получателей. Параметры, которые Вы должны установить в файле main.cf, - virtual_alias_domains и virtual_alias_maps. Используя имена файла из предыдущих разделов, параметры должны выглядить следующим образом:

virtual_alias_domains = hash:/etc/postfix/virtual_alias_domains

virtual_alias_maps = hash:/etc/postfix/virtual_alias_maps

Перезагрузите Постфикс и проверьте виртуальные домены как описано в следующем разделе.

### Испытание параметров настройки виртуального псевдонима домена

Вы можете проверить ваши параметры настройки виртуального домена , посылая сообщение существующим и неизвестным получателям в обоих доменах.

### Посылка существующему адресату в виртуальном домене

Вот, как Вы могли бы послать сообщение правильному получателю (postmaster):

**$ echo test | /usr/sbin/sendmail postmaster@postfix-book.com**

Проверите, что сообщение дошло, просмотрев лог файл. Вы должны увидеть следующие сообщения:

Apr 19 11:20:50 mail postfix/pickup[l7850j: B8C4629AB38: uid=0 from=<root>

Apr 19 11:20:50 mail postfix/cleanup[l7863]: B8C4629AB38:

message-id=<20040419092050.B8C4629AB38@mail.example.com>

Apr 19 11:20:50 mail postfix/qmgr[l785l]: B8C4629AB38:

from=<root@mail.example.com>, size=282, nrcpt=l (queue active)

Apr 19 11:20:50 mail postfix/local[l7866]: B8C4629AB38; to=<ralf@example.com>,

orig_to=<postmaster@postfix-book.com>, relay=local, delay=0, status=sent

(mailbox)

Тестовое сообщение Сначала пошло в postmaster@postfix-book.com; из-за записи в virtual_alias_maps, почта postmaster@postfix-book.com пошла к ralf@example.com. И затем была локально доставлена пользователю ralf, потому что example.com - "реальный" домен.

### Посылка сообщения по недопустимому адресу в виртуальном домене

Вот как Вы могли бы послать сообщение недопустимому получателю (nouser):

**$ echo test | /usr/sbin/sendmail nouser@postfix-book.com**

**$ tail -f /var/log/mail.log**

Apr 19 11:21:23 mail postfix/pickup[l7850]: 9B61F29AB38: uid=0 from=<root>

Apr 19 11:21:23 mail postfix/cleanup[l7863]: 9B61F29AB38:

message-id=<20040419092123.9B61F29AB38@mail.example.com>

Apr 19 11:21:23 mail postfix/qmgr[17851]: 9B61F29AB38:

from=<root@mail.example.com>, size=282, nrcpt=1 (queue active)

Apr 19 11:21:23 mail postfix/error[17887]: 9B61F29AB38:

to=<nousergpostfix-book,com>, relay=none, delay=0, status=bounced

(user unknown in virtual alias table)

Этой письмо предназначалось nouser@postfix-book.com. Поскольку не было никакой записи в virtual_alias_maps, доставка почты для nouser@postfix-book.com обрывалось с сообщением об ошибке "пользователь неизвестен в виртуальной таблице псевдонимов."

**Расширенные Отображения**

Чем больше виртуальных доменов псевдонимов, Вы добавляете тем больше Вы должны будете добавить тех же самых записей в картах . Это тот случай, когда пригождаются общие записи, регулярные выражения и неявные отображения. Они описаны в следующих подразделах.

**Общие записи**

В небольшом количестве ситуаций, Вы можете хотеть, чтобы почта для неизвестного адресата в виртуальном домене псевдонима пошла на общий адрес. страница руководства virtual(5) перечисляет множество путей, которыми Вы можете сделать это с помощью записи в virtual_alias_maps. Вот, простейший из них:

@postfix-book.com catchall@example.com

В этом случае, если ваш Постфикс сервер не может найти соответствие для unknownuser@postfix-book.com в карте псевдонимов виртуального домена для postfix-book.com , Постфикс отображает этот адрес в catchall@example.com.

**Использование регулярных выражений**

Вы можете использовать регулярные выражения в virtual_alias_maps, чтобы переадресовать почту для ряда неизвестных пользователей в виртуальном домене в общею учетную запись. Кроме того, Вы можете поместить соответствие в левой части файла (LHS) а соответствующий адрес в правой части (RHS) - как показано в примере ниже. Это может быть удобным, если Вы передаете соответствующие адреса, программе, определенной в строке alias_maps .
Чтобы понять, как это работает, рассмотрим следующую строку virtual_alias_maps:

/^(.*)@postfix-book\.com$/ catchall+$1@example.com

1. Почта для unknownuser@postfix-book.com передается к catchall+unknownuser@example.com.
2. Постфикс доставляет сообщение локальному, существующему получателю catchall@example.com, но в течение доставки к программе, он устанавливает переменную окружения $EXTENSION равной unknownuser- как описано в странице руководства local(8) ( параметр recipient_delimiter устанавливает разделитель $EXTENSION ; по умолчанию, это - +.)
3. Если программа работает с почтой общего адреса, она может использовать переменную окружения $EXTENSION, чтобы найти получателя и создать информационное сообщение, для того чтобы послать его назад первоначальному отправителю. Вы можете найти пример такой программы, по имени fuzzy, на <http://www.stahl.bau.tu-bs.de/~hildeb/postfix>.

Есть большое количество других задач, для которых Вы можете использовать регулярные выражения в вашей карте virtual_alias_maps. Одно особенно полезное применение позволяет отобразить в карте адреса, которые соответствуют определенному образцу получателя. Скажем, то, что Вы имеете несколько, admin-name@example.com получателей, почта для которых должна идти в отдельный почтовый ящик. Вы могли пробовать использовать эту запись:

/^admin-.*@postfix-book\.com$/ mailbox@example.com

Это намного лучше чем определение каждого соответствия вручную, как здесь:

admin-firewall@postfix-book.com mailbox@example.com

admin-mail@postfix-book.com mailbox@exampie.com

admin-web@postfix-book.com mailbox@example.com

**Неявные отображения для нескольких доменов**

Время от времени, может быть необходимо создать общее отображение, которое относится к нескольким доменам. Например, Ваша цель может состоять в том, чтобы создать общего получателя postmaster, на который будут приходить письма , независимо от того сколько виртуальных доменов Вы хостите. Вы можете сделать это, добавив следующую строку в Вашу карту виртуальных псевдонимов:

postmaster postmaster@example.com

В этом случае, все сообщения, адресованные получателю с локальной частью postmaster, придут на адрес postmaster@example.com. Из-за предыдущей записи Постфикс примет почту для следующих адресов и сопоставит им postmaster @example.com:

- postmaster@$myorigin
- postmaster@[$inet_interfaces]
- postmaster@$mydestination

Заметьте, что не все они – виртуальные псевдонимы доменов, и что эти три домена не обязательно покрывают все Ваши виртуальные домены. Смотрите страницу справочного руководства *virtual(5)*, которая подробно описывает порядок поиска. Если Вы хотите, чтобы у всех своих виртуальных доменов был тот же самый адрес postmaster`a, напишите сценарий, чтобы добавить их в virtual_alias_maps.

*ПРЕДОСТЕРЕЖЕНИЕ Вы не можете использовать переменные конфигурации (такие как $myorigin) в virtual_alias_maps Постфикс не будет преобразовывать переменные. Наше примечание служит только иллюстрацией.*

**Домены виртуальных почтовых ящиков**

Домены виртуальных почтовых ящиков - домены для пользователей, у которых нет локальной учетной записи (то есть, для пользователей, которые не находятся в /etc/passwd). Первоначально введенная, как патч, который позволял использовать

раздельных демона агента доставки, данная возможность является теперь стандартным компонентом Постфикса.
Агент доставки virtual в Постфиксе основан на агенте доставки local. В отличие от агента local, агент доставки virtual , не может обратиться к локальной пользовательской информации Вашей системы (например /etc/passwd), чтобы найти имя получателя. Вместо этого агент доставки virtual полагается полностью на карты, которые не имеют никакого отношения к Вашей системе.
Есть две причины для того, чтобы препятствовать тому, чтобы агент доставки virtual знал что-нибудь о системных учетных записях:

Масшабируемость

На Linux, использование локальных учетных записей, определенных в /etc/passwd, ограничивает почтовые сервера примерно 65 536 получателями. Solaris и *BSD не имеют этого ограничения. Агент доставки virtual не имеет этих пределов.

Безопасность

Вероятность компрометирования системы гораздо ниже, если имена пользователя и пароли локальных учетных записей не требуются, для того чтобы просто послать и получить почту. Кроме того, агент доставки virtual не может использывать определенные пользователем командные оболочки и не имеет доступ к пользовательским файлам.
Поскольку виртуальный агент поставки ничего не знает о Вашей системе, он не может обработать файлы такие, как $HOME/.forward или использывать приложения, такие как procmail и vacation. Агент доставки, virtual был урезан до только доставки почты в почтовые ящики .

Проверка Постфикса на поддержку агента доставки virtual

Чтобы использовать домены виртуальных почтовых ящиков, демон master должен быть в состоянии запустить демона virtual. Проверьте установки своего файла master.cf на задействование запуска демона как в следующем примере:

```
# ==========================================================================
# service type private unpriv chroot wakeup maxproc command + args
#                 (yes)   (yes)   (yes)   (never)   (100)
# ==========================================================================
   smtp    inet      n       -       n       -       -                smtpd
   local   unix      -       n       n       -       -                 local
   virtual  unix     -       n       n       -       -                virtual
```

ПРИМЕЧАНИЕ убедитесь, что демон virtual не выполняется chmoted окружении (см. пятый столбец в предыдущем примере).

Основная конфигурация
Чтобы сконфигурировать основной домен виртуальных почтовых ящиков, Вы должны заставить агента доставки virtual хранить все сообщения, используя одни и те же UID и GID. Вы должны будете выполнить следующие шаги:

1. Определить имя домена виртуальных почтовых ящиков.

2. Установить владельцем файлов агента доставки virtual.
3. Установить основной каталог для почтовых ящиков домена.
4. Создать карту получателей.
5. Создать карту псевдонимов.

**Установка имени домена виртуальных почтовых ящиков**

Во-первых, Вы должны будете сообщить Постфиксу, что он - заключительный адресат для одного или более доменов виртуальных почтовых ящиков, перечислив список доменов в параметре virtual_mailbox_domains в Вашем main.cf файле. Например, если бы Вы хотели создать домен виртуальных почтовых ящиков для доменного имени example.com, то Вы использовали бы эти настройки:

virtual_mailbox_domains = example.com

**Установка владельца файлов**

Хотя домены виртуальных почтовых ящиков не требуют, чтобы у каждого почтового ящика был уникальный пользователь, Вы все еще нуждаетесь по крайней мере в одном идентификаторе пользователя (UID) и группы (GID), чтобы предоставить доступ агенту доставки virtual к почтовым ящикам. Чтобы сделать это, Вы должны определить владельца карт с помощью параметров virtual_uid_maps и virtual_gid_maps.

Установка Пользователя

Чтобы установить владельца почтовых ящиков, Вы должны создать локального пользователя для почтовых ящиков, если Вы еще не сделали это. Чтобы создать пользователя почтовых ящиков с именем vbox и UID=1000, выполните эту команду:

# useradd vuser -u 1000

***ПРЕДОСТЕРИЖЕНИЕ*** *По умолчанию, у владельца почтового ящика, возможно, не может иметь идентификатор пользователя, меньше чем 100. Это мера безопасности, определяемая параметром virtual_minimum_uid, который предотвращает перезапись файлов, принадлежавших системным учетным записям демоном virtual. Вы можете установить различную границу, устанавливая параметр virtual_minimum_uid в Вашем main.cf файле.*

После установки UID пользователя почтовых ящиков, Вы должны должны указать демону virtual использовать этот идентификатор пользователя, когда он пишет сообщения в файловую систему. Установите идентификатор пользователя с помощью параметра virtual_ uid_maps в Вашем main.cf файле, как в этом примере для идентификатора пользователя 1000:

virtual_uid_maps = static: 1000

ОТМЕТЬТЕ Используйте опцию static для идентификатора пользователя, чтобы заставить демона virtual использовать исключительно этот идентификатор пользователя. Вы можете также применить динамический идентификатор пользователя , как это коротко объяснено в разделе "Расширенная конфигурация".

Установка Группы

Вы нуждаетесь в локальной группе  дополнено к пользователю, которого Вы только что установили. Для базовой настройки , создайте GID с тем же самым номером как идентификатор пользователя в предыдущем разделе. Команда useradd, возможно, уже  автоматически сделала это (проверьте свой файл /etc/group), в противном случае используйте следующую команду:

# groupadd vuser -g 1000

Теперь установите параметр virtual_gid_maps  в Вашем main.cf  так же, как Вы сделали для пользователя виртуальных почтовых ящиков:

virtual_gid_maps = static:1000

**Установка основного каталога для виртуального  домена почтовых ящиков**

Агент доставки virtual должен знать, где найти почтовые ящики для получателей. Обычно, именно операционная система, предоставляет системные переменные и файлы конфигурации которые говорят приложению о значениях параметров настройки системы по умолчанию. Поскольку агент доставки virtual не работает с системными переменными, Вы должны явно указать, куда помещать сообщения, которые он должен доставить пользователям.
Установите virtual_mailbox_base параметр в Вашем main.cf  файле чтобы определить, где хранить поступающие сообщения. Например:

virtual_mailbox_base = /var/spool/virtual_mailboxes

*ОТМЕТЬТЕ, что полный путь для индивидуального виртуального почтового ящика состоит из значения virtual_mailbox_base и значения в запроса из таблицы (описанной в следующем разделе). Другими словами это - $viitual_mailbox_base/$mailboxname.*

После установки значения параметра virtual_mailbox_base,  хорошая идея создать каталог и сделать это доступным для пользователя, которого Вы определили в предыдущих разделах:

**# mkdir /var/spool/virtual_mailboxes**

**# chown vuser:vuser /var/spool/virtual_mailboxes**

**# chmod 700 /var/spool/virtual_mailboxes**

**Создание Карты Получателя**

Вы должны определить получателей домена виртуальных почтовых ящиков в карте. Например, Вы могли создать файл, с именем /etc/postfix/virtual_mailbox_recipients с полностью определенными получателями в левой части и именами почтовых ящиков в правой части. Вот пример того, как это могло бы выглядеть:

**wilma.pebble@example.com  wilmapebble**

**betty.mcbricker@exainple.com  bettymebricker**

**fred.-flintstone@example.com  fredflintstone**

**barney.rubble@example.com** **barneyrubble**

**bamm.bamm@example.com** **bammbamm/**

Демон virtual добавляет значение параметра virtual_mailbox_base к имени почтового ящика, чтобы сформировать полное имя пути к файлу почтового ящика. Формат значения по умолчанию для почтовых ящиков - формат mbox, но Вы можете определить формат Maildir, добавляяя слэш (/) к названию почтового ящика, как в предыдущей записи для bamm.bamm@example.com.
После успешного создания файла создайте его индексированную версию выполнив команду:

```
# postmap hash:/etc/postfix/virtual_mailbox_aliases
```

После этого укажите Постфиксу где он может найти таблицу получателей установив параметр virtual_mailbox_maps parameter в main.cf следующим образом:

```
virtual_mailbox_maps = hash:/etc/postfix/virtual_mailbox_recipients
```

Ограничения таблицы получателей

Из соображений безопасности есть несколько ограничений на карты получателя:

- Получатели домена виртуальных почтовых ящиков не могут использовать расширение адреса, такое как user+exterision@domain.tld.
- Демон virtual не может вызвать внешние программы, как демон local.
- Возможно использовать регулярные выражения, но Вы не можете использовать замену выражения (это означает, что Вы не можете поместить 1 $ в RHS).
- Вы не может выполнить запрос в таблице с помощью демона proxymap демоном.

Создание карты псевдонимов

У Вас могут быть псевдонимы для домена виртуальных почтовых ящиков, но Вы должны поместить их в отдельную карту, такую как /etc/postfix/virtual_mailbox_aliases. Формат требует указания полностью определенное имя в левой стороне и полностью определенного адресата в правой стороне, как в этом примере:

```
wilma@example.com          wilrna.pebble@exarnple.corn
pebble@example.com         wilma.pebble@example.com
postmast:er@example.com    bamm.bamm@example.com
abuse@example.com          bamm.bamm@example.com
```

Как и в случае других карт вы должны создать индексированный файл с помощью команды:

```
# postmap hash:/etc/postfix/virtual_mailbox_aliases
```

Наконец, скажите, Постфиксу, использовать карту псевдонимов, установив virtual_alias_maps параметры в Вашем main.cf файле, как здесь:

virtual_alias_maps = hash:/etc/postfix/virtual_mailbox_aliases

После перезагрузки Постфикса Ваш почтовый сервер примет сообщения для получателей в домене виртуальных почтовых ящиков.

**Расширенная конфигурация**

Если Вы должны предоставить почтовые услуги для нескольких доменов виртуальных почтовых ящиков, возможно хранение всех сообщений в иерархии единственного каталога вызовет проблему потому что может быть два пользователя с тем же самым именем. Кроме того, раздельное резервное копирование данных становится очень сложным. Чтобы решить эти проблемы, Вы можете сконфигурировать домены виртуальных почтовых ящиков, чтобы хранить сообщения для различных доменов в различных каталогах. У Вас также есть опция использования различных пользовательских (UID) и групповых идентификаторов (GID).
Чтобы установить этот вид расширенной конфигурации, Вы должны сделать следующее.

1. Определить имена доменов виртуальных почтовых ящиков.
2. Установить владельца файлов для агента доставки virtual.
3. Установите основной каталог для почтовых ящиков.
4. Создайте карту получателей.
5. Создайте карту псевдонимов.
6. Установить хранилище и разрешения на доступ.

Имена виртуальных доменов
Как описано в ранее в разделе "Установка имени домена виртуальных почтовых ящиков", Вы устанавливаете имена доменов виртуальных почтовых ящиков с помощью параметра virtual_mailbox_ domains. Вот пример с двумя доменами

virtual_mailbox_domains = example.com, post-fix-book.com

**Установка владельца файла**

Для расширенной конфигурации Вы должны создать набор идентификатора пользователя и группы для каждого виртуального домена получателя. Скажем, Вы хотите использовать имя пользователя и группы example для домена example.com и pfxbook для домена postfix-book.com <http://постфикс-book.com>. Вернитесь к разделу «Установка владельца файлов» за деталями, чтобы создать пользователей, Вы могли бы использовать эти команды:

```
# useradd example -u 1001
# useradd pfxbook -u 1002
# groupadd example -g 1001
# groupadd pfxbook -g 1002
```

Запомните эти значения GID и UID; Вы скоро снова будите их использовать, когда Вы создадите карты в разделе"Установка хранилища и разрешениях на доступ".

**Установка основного каталога для доменов виртуальных почтовых ящиков**

Вы должны установить параметр virtual_mailbox_base параметры, чтобы указать демону virtual , где он должен хранить сообщения, так же, как в ранее разделе "Установка основного каталога для домена виртуальных почтовых ящиков. Давайте использовать ту же самую установку как в том разделе:

virtual_mailbox_base = /var/spool/virtualmailboxes

Однако, различие между тем, что мы делаем здесь и базовая конфигурацией, которую мы устанавливаем ранее, - в том, что Вы должны изменить разрешения для каталога virtual_mailbox_base. Иначе демон virtual используя различные идентификаторы пользователя и группы для каждого домена, когда он сохраняет сообщение не получит разрешение на запись в подкаталоги:

# mkdir /var/spool/virtual_mailboxes

# chown vuser:vuser /var/spool/virtualmailboxes

# chmod 775 /var/spool/virtualmailboxes

Теперь Вы должны будете создать подкаталоги (для, example.com и postfix-book.com>), потому что virtual создаст только mbox или Maildir получателя, а не директорию предыдущего уровня его домена.

```
# mkdir  example.com
# chown   example example.com/
# chgrp   example example.com/
# chmod   700 example.com/
```

ПРЕДОСТЕРИЖЕНИЕ Постфикс 2.0, и ранее не будет создавать почтовые ящики Maildir-стиля в перезаписываемых директориях предыдущего уровня; Вы должны будете создать их заранее.

Если доставка будет терпеть неудачу из-за некоторой проблемы разрешений, то Вы увидите следующие сообщения в лог файле:

May 26 12:04:33 mail postfix/virtual[i4i96]: warning: maildir access problem for

UID/GID=1002/1002:

create /var/spool/mailboxes/postfix-book.com/patrick/tmp/1085565873.P14196.mail.example.com:

Permission denied

May 26 12:04:33 mail postfix/virtual[l4196]: warning: perhaps you need to create the maildirs in

advance

**Создание карт Получателей**

Теперь Вы должны создать карту для валидных получателей в Ваших доменах виртуальных почтовых ящиков. Процесс - то же самый как в ранее в разделе «Создание карты получателей», за исключением того, что Вы должны добавить

подкаталоги к названиям почтового ящика. Таким образом, сообщения для отдельных доменов входят в различные каталоги, так, чтобы Вы не должны были волноваться о конфликтах имен.
Например, Вы можете создать /etc/postfix/virtual_mailbox_recipients файл:

wilma.pebble@example.com example.com/wilmapebble/

betty.mcbrickergexample.com example.com/bettymcbricker/

fred.flintstonegexample.com example.com/fredflintstone/

barney.rubblegexample.com example.com/barneyrubble/

bamm.bamm@example.com example.com/bammbamm/

ralf@postfix-book.com postfix-book.com/ralf/

patrickgpostfix-book.com postfix-book.com/patrick/

Помните, что после создания карты, Вы должны построить индексированную версию с помощью этой команды:

**# postmap hash:/etc/postfix/virtual_mailbox_recepients**

Как и прежде, устанавите параметр virtual_mailbox_maps в Вашем main.cf файле.

virtual_mailbox_maps = hash:/etc/postfix/virtual_mailbox_recipients

Данная карта отвечает за всех получателях за исключением псевдонимов, которые Вы создаете таким же образом, как описано в ранее разделе «Создании карты псевдонимов».

**Установка хранилища и прав доступа**

Вы не можете определить отдельное хранилище и права доступа для других доменов виртуальных почтовых ящиков, как был описано в ранее разделе «Установка владельца файлов». Вместо этого Вы должны создать карты, которые сопоставляют почтовые ящики и идентификаторы пользователя и группы.
Сейчас Вам потребуются идентификаторы пользователей и группы, которые Вы создали в разделе «Установка владельца файлов». Вы должны назначить идентификатор пользователя для каждого почтового ящику в карте определяемой параметром virual_uid_maps. Например, Вы могли определить имя карты hash:/etc/postfix/virtual_uid_map с помощью этой строки в Вашем main.cf файле:

virtual_uid_maps = hash:/etc/postfix/virtual_uid_map

Теперь, поместите получателей в карту, вводя полный адрес получателя в левой части а идентификатор пользователя в правой стороне, как в этом примере:

wilma.pebble@example.com 1001

betty.mcbricker@example.com 1001

fred.flintstone@example.com 1001

barney.rubble@example.com 1001

bamm.bamm@example.com 1001

rah@postfix-book.com 1002

patrick@postfix-book.com 1002

*ПРИМЕЧАНИЕ не забывайте создавать индексированную версию карты при помощи команды postmap. Более короткая и четкая версия может быть:*

@example.com 1001

@postfix-book.com 1002

Отображение GID работает точно так же, как и отображение идентификатора пользователя. Вот карта, которую Вы можете использовать в качестве примера для /etc/postfix/virtual_gid_map файла:

wilma.pebble@example.com 1001

betty.mcbricker@exainple.com 1001

f red. flintstone@example.co 1001

barney.rubble@example.com 1001

bamm.bamm@example.com 1001

ralf@postfix-book.com 1002

patrick@postfix-book.com 1002

Определите этот файл в Вашем virtual_gid_maps параметре следующим образом:

virtual_gid_maps = hash:/etc/postfix/virtual_gid_map

*СОВЕТ, поскольку элементы для virtual_gid_maps в этом примере - точно такие же как для virtual_uid_maps, Вы можете пропустить создание файла карты GID а только сослаться на карту идентификации пользователя в Вашем main.cf файле следующим образом:*

virtual_gid_maps = virtual_uid_maps

После создания карт и обновления файла конфигурации, перезагрузите Постфикс, чтобы демон virtual доставлял сообщения в подкаталоги, директорий с именами доменов получателя.

**Генерация карт с помощью сценариев**

Как Вы, возможно, обратили внимание, виртуальные домены требуют по крайней мере три карты для поиска: получателей, почтовых ящиков, и прав доступа групп и владельцев. Вы можете намного облегчить свою жизнь при наличии сценария компоновки всех карт из отдельного исходного файла, например, /etc/postfix/virtual_build_map_source, который содержит всю запрашиваемую информацию. Например, скажем, то, что исходный файл содержит следующие

строки:

```
pebble@example.com                  example.com/wilmapebble/        1001    1001
betty.mcbricker@example.com         example.com/bettymcbricker/     1001    1001
fwilmared.flintstone@example.com    example.com/fredflintstone/     1001    1001
barney.rubble@example.com           example.com/barneyrubble/       1001    1001
bamm.bamm@example.com               example.com/bammbamm/           1001    1001
ralf@postfix-book.com               postfix-book.com/ralf/          1002    1002
patrick@postfix-book.com            postfix-book.com/patrick/       1002    1002
```

Следующий сценарий (давайте назовем его /etc/postfix/build_virtual_maps) читает данные из файла и создает три целевых карты:

```
#   !/bin/bash
#
# Build all virtual mailbox maps from one source
# section: paths
SOURCE=/etc/postfix/virtual_build_map_source
VMAP=/etc/postfix/virtual_rnailbox_recipients
VUID=/etc/postfix/virtual_uid_map
VGID=/etc/postfix/virtual_gid_map
 AWK=/usr/bin/awk
POSTMAP=/usr/sbin/postmap
# section: build
# build $virtual_mailbox_maps
$AWK '{printf("%s %s\n"$1,$2)}' $SOURCE > $VMAP
$POSTMAP hash:$VMAP
# build $virtual_uid_maps
$AWK '{printf("%s %s\n"$1,$3)}' $SOURCE > $VUID
$POSTMAP hash:$VUID
# build $virtual_gid_maps
$AWK   '{printf("%s %s\n",$1,$4)}'  $SOURCE > $VGID
$POSTMAP hash:$VGID
```

ОТМЕТЬТЕ, что Вы можете загрузить этот сценарий с сайта данной книги http://www.postfix-book.com. (Вы, возможно, должны изменить пути вначале, конечно).

После выполнения сценария у всех карт, кроме исходного файла и виртуальной

карты псевдонима, должна быть одинаковая дата и время:

```
-rw-r--r--    l root    root    532 May 26 12:12 virtual_build_map_source
-rw-r--r--    1 root    root    251 May 26 13:21 virtual_gid_map
-rw-r--r--    1 root    root    12288 May 26 13:21 virtual_gid_map.db
-iw-r--r--    l root    root    394 May 26 13:21 virtual_mailbox_recipients
-rw-r--r--    1 root    root    12288 May 26 13:21 virtual_mailbox_recipients.db
-rw-r--r--    1 root    root    251 May 26 13:21 virtual_uid_map
-rw-r--r--    1 root    root    12288 May 26 13:21 uirtual_uid_map.db
```

Виртуальные домены почтовых ящиков с использованием базы данных

Если Вы интересуетесь почтовым сервером масштаба предприятия или почтовым сервером интернет провайдера, Вы можете организовать доступ Постфикса к базе данных, чтобы получить информацию о виртуальных доменах почтового ящика. Эта возможность особенно удобна, потому что Вы можете делегировать администрирование пользовательских учетных записей другим людям, не давая им доступ с правами администратора к серверу. Если Вы предоставите мощный web интерфейс, Ваши заказчики могут управлять, собственными данными (добавление псевдонимов, извинение их SMTP, POP3, паролей, и так далее). Кроме того, изменения данных в базе немедленно обнаруживаются, таким образом Вы не должны перезагружать Постфикс каждый раз, когда Вы изменяете данные.
С другой стороны, индексированные карты быстрее, выполняют запросы , не используют так много системных ресурсов как SQL-запросы, потому что Вы не должны выполнять сервер базы данных. Кроме того, решение на основе базы данных может быть более сложным.
Если Вы сталкиваетесь с проблемами производительности при выполнении запросов в базе данных, Вы можете установить специализированный сервер базы данных, который может быть доступен нескольким Postfix серверам (и другим службам) в Вашей сети. В совокупности с механизмами балансировки нагрузки, например, циклического алгоритма и специальных аппаратных средств, Вы можете построить высокоэффективные почтовые службы с базой данных.
Этот раздел показывает Вам, как реализовать виртуальные домены почтовых ящиков под управлением базы данных, используя MySQL в качестве базы данных.
Вот то, что Вы должны сделать:

1. Проверка поддержки Постфиксом MySQL.
2. Сборка Постфикса, с поддержкой карт MySQL.
3. Настройка конфигурации базы данных
4. Тестирование виртуальные доменов почтовых ящиков под управлением базы данных.

*ОТМЕТЬТЕ Управляемые базой данных карты не ограничены применением в виртуальных доменах почтовых ящиков. Вы можете использовать их в многих других целях. Постфикс также поддерживает PostgreSQL, и LDAP (конфигурация PostgreSQL почти идентична MySQL; LDAP обсуждается в Главе 19).*

**Проверка поддержки Постфиксом карт MySQL**

Прежде, чем Вы сконфигурируете Постфикс, для запросов в MySQL, Вы должны вероятно проверить, что Ваша инсталляция фактически поддерживает этот тип карт. Используйте команду postconf -m команду, чтобы вывести поддерживаемые типы карт. Если у Вас есть поддержка MySQL, Вы должны видеть mysql в выводе, как в этом примере:

**# postconf -m**

btree

cidr

environ

hash

ldap

mysql

nis

pcre

proxy

regexp

sdbm

static

unix

Если у Вас отсутствует поддержка MySQL, Вы можете устанавливить пакет Постфикса из репозитария Вашей операционной системы, который поддерживает MySQL, или Вы можете собрать его вручную и затем установить новую версию (как описано в следующем разделе).

Сборка Постфикса с поддержкой карты MySQL

Чтобы скомпилировать Постфикс с поддержкой MySQL таблиц, сначала определите местонахождение заголовочных файлов и библиотек, которые необходимы Постфиксу. Чтобы найти каталог заголовочных файлов, используйте эту команду:

**# -find /usr -name 'mysql.h'**
/usr/include/mysql/mysql.h

Предыдущий вывод показывает, что заголовочные файлы на этой операционной системе находятся в /usr/include/mysql, для поиска клиентских библиотек MySQL, выполните эту команду:

**# find /usr -name 'libmysqlclient.*'**

/usr/lib/mysql/libmysqlclient.so.10

/usr/lib/mysql/libmysqlclient.so.10.0.0

```
/usr/lib/mysql/libmysqlclient.so
/usr/lib/mysql/libmysqlclient.a
```

Вывод показывает, что библиотеки находятся в /usr/lib/mysql.
Теперь, когда Вы знаете правильные пути, Вы можете установить переменные для конфигурации Постфикса и make файла. Для путей в этом примере Вы использовали бы следующую команду, чтобы сконфигурировать и собрать Постфикс:

```
$   make tidy
$   make make-files CCARGS='-DHAS_MYSQL -I/usr/include/mysql'
    AUXLIBS='-L/usr/lib/mysql -lmysqlclient -lz -lm'
$    make
```

После того, как сборка завершится, и Вы устанавите Постфикс, проверьте, что у Вас есть поддержка MySQL как описано в предыдущем разделе.

**Конфигурирование Базы данных**

Для создания базы данных MySQL, чтобы хранить информацию о Вашем виртуальном домене, соединиться с MySQL как root и создате базу данных. Следующая команда создает базу данных, с именем mail:

```
mysq> CREATE DATABASE "mail";
```

*СОВЕТ Вы можете загрузить полный набор SQL-выражений для этого раздела с сайта данной книги http://www.postfix-book.com. Вы должны изучить некоторые SQL-выражениях сначала. Простое Введение в SQL может быть найдено на http://sqlzoo.net>.*

**Создание таблицы доменов получателей**

Создайте таблицу с именем, например, virtual_mailbox_domains, чтобы хранить домены, для которых Постфикс сочтет себя заключительным предназначением. Вы можете использовать эту команду:

```
mysql>  CREATE TABLE "virtual_mailbox_domains"  (
mysql>      'Id'  int(10) unsigned NOT NULL auto_increment,
mysql>      'domain'   varchar(255)  default NULL,
mysq>     PRIMARY KEY    ('Id'),
mysql>      FULLTEXT KEY "domains'   ('domain')
mysql>   ) TYPE=MyISAM COMMENTS ='Postfix virtual aliases';
```

Если эта команда выполнена успешно, Вы можете добавить свои виртуальные домены в таблицу. Например, чтобы добавить example.com в таблицу, напечатайте эту команду SQL, чтобы вставить строку:

```
mysql> INSERT INTO virtual_mailbox_domains VALUES  (1,'example.com');
```

**Добавление пользователей**

Теперь пришло время создать таблицу, где каждая строка содержит получателя, имя почтового ящика, UID, и GID. Вы можете назвать ее virtual_users; структура nf,kbws очень похожа на столбцы /etc/postfix/virtual_build_map_sourcefile, что мы использовали ранее в разделе "Генерация карт при помощи сценариев" .

*ОТМЕТЬТЕ, что следующая таблица построена  на таблице MySQL SMTP AUTH, описанной в Главе 18. Она содержит пароли и другую информацию, таким образом Вы можете использовать его как внутренний источник и для аутентификации SMTP и для виртуальных доменов почтового ящика,*

Выполните следующую команду, чтобы создать таблицу virtual_users :

```
mysql> CREATE TABLE "virtual_users"  (
mysql>     'id'  int(11) unsigned NOT NULL auto_increment,
mysql>     'username'  varchar(255) NOT NULL default  '0',
mysql>     'userrealm'  varchar(255) NOT NULL default 'mail.example.com'  ,
mysql>     'userpassword' varchar(255) NOT NULL default  'lstP@ss',
mysql>     'auth' tinyint(1) default  '1',
mysql>     'active'  tinyint(l) default  '1',
mysql>     'email' varchar(255) NOT NULL default  ' ',
mysql>     'virtual_uid'  smallint(5) default  '1000',
mysql>     'virtual_gid'  smallint(5) default  '1000',
mysql>     'virtualmailbox' varchar(255) default NULL,
mysqb     PRIMARY KEY    ('id'),
mysql>     UNIQUE KEY 'id'   ('id'),
mysql>     FULLTEXT KEY 'recipient'   ('email')
mysql> ) TYPE=MyISAM COMMENT='SMTP AUTH and virtual users';
```

Поле active является опциональным; Вы можете использовать его, чтобы задейчтвовать или отключить почтовый ящик получателя (что может бы полезным, если клиент не заплатил, и Вы должны отключить сервис, но Вы не хотите потерять конфигурацию учетной записи).
После создания таблицы, Вы должны добавить данные для того, чтобы тестировать. Вот команда, которая добавляет строку в качестве примера:

```
mysql> INSERT INTO virtual_users VALUES (5,'bamm.bamm','mail.example.com','1stP@ss',1,1,
'bamm@example.com',1001,1001,'example.com/bammbamm/');
```

**Создание Таблицы для Виртуальных Псевдонимов**

Последняя таблица, которую Вы должны составить, таблица для виртуальных псевдонимов. Как с другими картами псевдонимов, строки таблицы содержат имя псевдонима и реальный адрес получателя. Составьте таблицу (с именем, например, virtual_aliases) следующим образом:

```
musql>  CREATE TABLE "virtual_aliases"   (
mysql>     'Id'  int(10) unsigned NOT NULL auto_increment,
mysql>     'alias' varchar(255) default NULL,
```

```
mysql>     'virtual_user_email'  text,
mysql>     PRIMARY KEY    ('Id'),
mysql>     FULLTEXT KEY 'aliases'   ('alias','virtual_user_email')
mysql>   ) TYPE=MyISAM COMMENT='Postfix virtual recipients';
```

Как и с другими таблицами, Вы должны заполнить ее некоторыми данными. Начните с псевдонимов, которые требуются RFC :

```
mysql> INSERT INTO virtual_aliases VALUES

(l,'postmaster@example.com','bamm.bamm@example.com');

mysql> INSERT INTO virtual_aliases VALUES

(2,'abuse@example.com','bamm.bamm@example.com');
```

### Создание пользователя MySQL для Постфикса

Ваша последняя задача в конфигурировании базы данных состоит в том, чтобы создать пользователя MySQL, для выполнения запросов в таблицы. Вы должны ограничить права доступа пользователя так, чтобы Постфикс не мог изменять данные. Следующая команда добавляет нового пользователя, с именем postfix, который может соединиться от localhost:

```
mysql> CONNECT mysql;
rnysql> INSERT INTO user VALUES
('localhost','postfix', ' ','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y'); mysql>
UPDATE mysql.user SET password=PASSWORD("Yanggt!") WHERE user='postfix'   AND
host='localhost';
mysql> FLUSH PRIVILEGES;
```

Вы должны ограничить учетную запись доступом только для чтения (SELECT). Постфикс не должен быть в состоянии изменять или создавать таблицы. Используйте команду GRANT, чтобы сделать это:

```
mysql> GRANT USAGE ON *.* TO 'postfix'@'localhost'  IDENTIFIED BY PASSWORD
'2fc8797l4f7d3e72';
mysql> GRANT SELECT ON mail.virtual_aliases TO ' postfix '@ 'localhost';
mysql> GRANT SELECT ON mail.virtual_users TO  'postfix'@'localhost';
mysql> GRANT SELECT ON mail.virtual_mailbox_domains TO  'postfix'@'localhost';
```

### Настройка Постфикса, для использования базы данных

Настраивая систему SQL-запроса для Постфикса, Вы должны установить следующие параметры в специальном файле. Постфикс вставляет эти параметры в прогрессию SQL-выражений, вершина которой операция SELECT:

**username**

Имя пользователя, который подключается к базе данных.

**password**

Пароль пользователя базы данных. Он должен быть в форме открытого текста.

**host**

Список одного или более имен хоста, полностью определенного доменного имени или IP адреса SQL сервера. Если Постфиксу не удастся установить связь с первым хостом в списке, то он попробует другой, хост в случайном порядке. Если ни один сервер не доступен, Постфикс прекратит работу, пока сервер не станет доступным . Когда Вы используете localhost как сервер, Постфикс автоматически использует Unix сокет вместо подключения TCP/UDP.

**dbname**

База данных, для соединения

**table**

Имя таблицы, которая содержит данные виртуального домена.

**select_field**

Поле, которое содержит результат запроса (например, адрес электронной почты пользователя).

**where_field**

Поле, для совпадения, при выполнении запроса в базе данных (например, дополнительный адрес электронной почты).

**Additional_conditions**

Дополнительные условия для запроса; например, Вы, возможно, хотите запрашивать только учетные записи пользователей, которые являются в настоящий момент активными. Этот параметр является опциональным.

Таблица 14-1 иллюстрирует, соответствие полей индексированных карт и параметров базы данных. Вы можете использовать его, создавая оператор SELECT, если Вы не уверены в полях таблицы, которые Вы должны определить.

Таблица 14-1 соответствие полей индексированных карт и параметров базы данных

| Тип Карты | Левая часть (LHS) | Правая часть | Условия |
|---|---|---|---|
| Индексированная карта | левая колонка | правая колонка | - |
| Таблица SQL | where_field | select_field | Additional_conditions |

**Защита SQL конфигурации Постфикса**

Постфикс в настоящее время поддерживает два способа настройки MySQL (и PostgreSQl). оператора SELECT.

Первый требует, чтобы Вы написали имя пользователя SQL сервера базы данных Постфикс и пароль в Вашем rrain.cf файле. Поскольку это не очень безопасно, этот способ не будет описан в этой книге (main.cf, обычно открыт для чтения, таким

образом каждый пользователь на Вашей системе был бы в состоянии получить эти параметры доступа). Если Вы настаиваете на том, чтобы использовать этот метод, см. MYSQLREADME для получения дополнительной информации, но знаете, что будущие версии Постфикса не будут поддерживать этот способ.
Второй метод предпочтителен, потому что он намного лучше защищен. Параметры оператора SELECT (включая имя пользователя и пароль) находятся в отдельных файлах за пределами main.cf. Кроме того, Вы поместите их в подкаталог, доступный только Постфиксу и пользователю root (Вы определите расположение файлов в main.cf).
Прежде чем создать файл и перейти к настройке, сначала создайте каталог, например /etc/postfix/sql, и установите соответствующие разрешения

**# mkdir /etc/postfix/ sql**

**# chown postfix /etc/postfix/sql**

**# chgrp root /etc/postfix/sql**

**# chmod 500 /etc/postfix/sql**

Теперь Вы готовы добавить файлы сюда , где они защищены от большинство любопытных глаз.

Построение запроса для доменов получателей

Первый файл, который Вы должны добавить, определяет параметры для запроса, который отыскивает домены, для которых Постфикс считает себя конечным адресатом. Добавьте следующую конфигурацию в файл, например /etc/postfix/sql/virtual_ domains.cf:

```
user = postfix
password = Yanggt!
dbname = mail
table = virtualmailboxdomains
selectfield = domain
where_field = domain
hosts = localhost
```

Как упомянуто ранее, эти параметры соответствуют значениям параметров оператора SE LECT SQL. Оператор для предыдущего файла был бы следующим, где *'domainpart'* доменная часть адреса получателя входящего сообщения электронной почты:

**mysql> SELECT domain FROM virtual_mailbox_domains WHERE domain = 'domainpait'**

Может показаться немного странным, искать домен из таблицы когда Вы уже знайте его. Однако предназначение этого запроса найти любые строки в базе данных, которые соответствуют доменной части. Если нет ни одного, запрос заканчивается неудачей и Постфикс знает, что он не является конечным адресатом для домена.

ОТМЕТЬТЕ, что Вы не нуждаетесь в введении любого из этих операторов SELECT (Постфикс создает их автоматически, при обращении к базе данных), но это полезно

при, решении проблем с базой данных из командной строки MySQl..

Теперь Вы должны указать Постфиксу (в main.cf) использовать MySQL и где найти параметры для virtual_itiailbox_domains. Строка очень напоминает строку при регулярных индексированных карт, но с зарезервированным словом, замененным на mysql:

virtual_mailbox_domains = mysql:/etc/postfix/sql/virtual_mailbox_domains.cf

**Указание параметров для запроса идентификатора пользователя и группы**

Затем Вы должны добавить параметры запроса идентификатора пользователя и группы (помните, что они определяют виртуального владельца файла почтового ящика). Добавьте следующие строки в файл, с именем /etc/postfix/sql/virtual_uid_maps.cf:

user = postfix
password = Yanggt!
dbname = mail
table = virtual_users
select_field = virtual_uid
where_field = email
hosts = localhost

Оператор SELECT, соответствующему этому файлу, подобен следующему (где *recipient* - адрес получателя входящего сообщения):

**mysql> SELECT virtualuid FROM virtualusers WHERE email = 'recipient'**

Используйте параметр virtual_uid_maps в main.cf, чтобы указать Постфиксу, где это может найти SQL запрос для поиска

virtual_uid_maps = mysql:/etc/postfix/sql/virtual_uid_maps.cf

Создайте SQL запрос для GID подобно UID. Используйте следующее имя файла, /etc/postfix/sql/virtual_gid_maps.cf, для создания SQL оператора SELECT:

user = postfix
password = Yanggt!
dbname = mail
table = virtual_users
select_field = virtual_gid
where_field = email
hosts = localhost

Соответствующий SQL оператор SELECT для этого файла выглядит так:

**mysql> SELECT uirtual_gid FROM virtual_users WHERE email = 'recipient'**

Затем укажите Постфиксу где искать GID. Установите virtual_gid_maps параметр в main.cf файле:

virtual_gid_maps = mysql:/etc/postfix/sql/virtual_gid_maps.cf

**Создание запроса для получателя**

Возможно самый важный запрос - тот, который отыскивает имя почтового ящика получателя когда дан адрес получателя. Добавьте следующие параметры запроса в файл /etc/postfix/sql/virtual_mailbox_recipients.cf:

```
user = postfix
password = Yanggt!
dbname = mail
table = virtual_users
select_field = virtualjnailbox
where_field = email
additional_conditions = and active = '1'
hosts = localhost
```

Обратите внимание на параметр additional_conditions указанный здесь . Параметры в этом файле соответствуют следующему оператору SELECT:

```
mysql> SELECT virtual_mailbox FROM virtual_users WHERE email = 'recipient' AND active = '1'
```

Параметр virtual_mailbox_maps укажет Постфиксу, где искать виртуальных получателей почтового ящика и их почтовые ящики. В main.cf, добавьте эту строку:

Создание запроса для псевдонимов

Наконец, пришло время определить параметры запроса для виртуальных псевдонимов. Поместите следующие строки в файл, /etc/postfix/sql/virtual_alias_maps.cf:

```
user = postfix
password = Yanggt!
dbname = mail
table = virtualaliases
select_field = virtualuseremail
where_field = alias
hosts = localhost
```

Следующий оператор SELECT, который соответствует предыдущему файлу:

**mysql> SELECT virtual_user_email FROM virtual_aliases WHERE alias = 'recipient'**

Чтобы построить его, скажите Постфиксу, где найти, файл конфигурации для запроса псевдонимов с помощью параметра virtual_alias_maps в main.cf:

```
virtual_alias_maps = mysql:/etc/postfix/sql/virtual_alias_maps.cf
```

Перезагрузите Постфикс, чтобы все изменения в main.cf в ступили в силу, и начните проверку .

### Тестирование виртуальных доменов почтовых ящиков

Решая проблему довольно трудно определить заключается ли она в Постфиксе или в MYSQL. Именно поэтому Вы должны протестировать MySQL и Постфикс отдельно. Если тесты MySQL успешно выполняются, то вы узнаете что проблема заключается в конфигурации Постфикса.

### Тестирование MySQL

Самая первая вещь, которую Вы должны проверить, - позволяют ли имя пользователя и пароль, который Вы поставляли в файлах конфигурации запросов, обращаться к MySQL и делать запросы.
Затем попытайтесь соединиться с базой данных, которая хранит данные Ваших виртуальных доменов почтовых ящиков. Оба теста показаны в следующем примере:

**# mysql -u postfix -p -h localhost -A**

В случае удачного соединения вы увидите следующее сообщение:

```
Welcome to the MySQL monitor.
 Commands end with ; or \g.
Your MySQL connection id is 144 to server version:  3.23.58
Type  'help;'  or  '\h'  for help.
Type  '\c' to clear the buffer.
mysql>
```

Используйте оператор CONNECT чтобы соединиться с почтовой базой данных Вы также должны получить подтверждение которое выглядит следующим образом:

```
mysql> CONNECT mail;
Connection id:       145
Current database:  mail
mysql>
```

Если Вы не можете подключиться к серверу, проверьте имя пользователя и пароль, который Вы определили для MySQL в ранее в разделе "Создание пользователя MySQL для Постфикса". С другой стороны, если у Вы не можете, соединиться с почтовой базой данных, проверьте оператор GRANT (в той же самой секции).

### Запрос Доменов Получателя

Теперь пришло время выполнить оператор SELECT, который проверит, что Ваш Постфикс пользователь MySQL может искать виртуальные домены почтовых ящиков в таблице virtual_mailbox_domains . Введите команду, такую как эта

**mysql> SELECT domain FROM virtual_mailbox_domains WHERE domain = example.com;**

Если она выполняется успешно, Вы должны увидеть одну строку вывода, которая содержит ввод запроса ( вернитесь к разделу «Построение  запроса доменов получателей», чтобы убедиться что это - правильное поведение):

```
+-------------------+
|  domain           |
+-------------------+
| example.com |
+-------------------+
l row in set (0.00 sec)
```

Если Вы не получаете соответствие, проверьте, существует ли домен в Вашей таблице. Простой способ проверить это состоит в том, чтобы вывести все строки из таблицы, опустив выражение WHERE:

**mysql> SELECT domain FROM virtual_mailbox_domains;**

### Запрос UID и GID виртуальных почтовых ящиков

Затем проверьте, что Ваш пользователь MySQL в состоянии восстановить известный адрес получателя. Попробуйте к восстанавить поле virtual_uid для известного адреса получателя в Вашей virtual_users таблице, как в этом успешном примере, который показывает отображение  bamm.bamm@example.com к UID виртуального почтового ящика 1001;

```
mysql> SELECT  virtual_uid FROM virtual_users WHERE email = 'bamm.bamm@example.com';

+-------------------+
| virtual_uid       |
+-------------------+
|        1001       |
+-------------------.+
1 row in set (0.00 sec)
```

Выполните тоже самое для GID:

```
mysql> SELECT virtual_gid FROM virtual_users WHERE email = 'bamm.bamm@example.com';

+-----------------+
|    virtual_gid  |
+-----------------+
|    1001         |
+-----------------+
1 row in set (0.00 sec)
```

### Запрос почтовых ящиков получателя

Протестируйте, может ли Постфикс пользователь MySQL найти почтовый ящик для данного получателя. Вспомните, что это поле virtual_mailbox в таблице virtual_users. Вот пример, который отображает bamm.bamm@example.com в почтовый ящик Maildir-стиля example.com/bammbamm/:

```
mysql> SELECT virtual_mailbox FROM virtual_users WHERE email =
'bamm.bamm@example.com';

+---------------------------------+
| virtual_mailbox                 |
+---------------------------------+
| example.com/bammbamm/  |
+---------------------------------+
1 row in set (0.00 sec)
```

### Запрос Псевдонимов

Ваша последняя проверка базы данных для псевдонимов. Для известного псевдонима, восстановите адрес получателя ( поле virtual_user_email в virtual_aliases таблице), Вот пример:

```
mysql> SELECT virtual_user_email FROM virtual_aliases WHERE alias =
'postmaster@example.com';

+-----------------------------------+
|   virtual_user_email              |
+-----------------------------------+
| bamm.bamm@example.com    |
+-----------------------------------+
1 row in set (0.00 sec)
```

### Тестирование Постфикса

Вы можете проверить выполнение Постфиксом запросов MySQL, не отправляя тестовых сообщений электронной почты. Команда postmap может выполнить любой вид запроса, включая в таблицах MySQL. Вот общий формат команды postmap, которая выполняет запрос MySQL:

**# postmap -q "value" mysql:path-to-parameter-file**

Например, вот, как указать Постфиксу сделать запрос MySQL для известного виртуального домена почтового ящика:

**# postmap -q "example.com" mysql:/etc/postfix/sql/virtual_mailbox_domains.cf**

Если он успешен, данные, соответствующие запросу, должны, отобразится в командной строке:

```
example.com
```

Если никакой результат не возвращается, проверьте, существует ли это имя виртуальное домена в Вашей таблице (см. ранее раздел «Запрос доменов

получателя). Если тот тест был успешен, проверьте имя пользователя и пароль в Вашем virtual_mailbox_domains.cf файле.

### Запрос UID и GID

Чтобы продолжить с тестирование, укажите Постфиксу делать выполнить MySQL запрос для UID и GID известного получателя виртуального почтового ящика:

**# postmap -q "bamm.bamm@example.com" mysql:/etc/postfix/sql/virtual_uid_maps.cf 1001**

**# postmap -q "bamm.bamm@example.com" mysql:/etc/postfix/sql/virtual_gid_maps,cf 1001**

### Запрос получателей

Затем, проверьте, что Постфикс может ваполнить запрос для известных получателей. Команда postmap, которая соответствует оператору SELECT из раздела «Запрос почтовых ящиков получателей», следующая:

# postmap -q "bamm.bamm@example.com" mysql:/etc/postfix/sql/virtual_mailbox_recipients.cf

example.com/bammbamm/

### Запрос Псевдонимов

Ваш последний тест - указать Постфиксу выполнить MySQL запрос для псевдонима пользовательского адреса электронной почты. Успешный запрос выглядит следующим образом

**# postmap -q "postmaster@example.com" mysql:/etc/post-fix/sql/virtual_alias_maps.cf**
**bamm.bamm@example.com**