The Book of Postfix

State-of-the-Art Message Transport

Ralf Hildebrandt, Patrick Koetter

NO STARCH
PRESS

HIGH TECH

Postfix

Подробное руководство

Современный транспорт
для сообщений

Ральф Гильдебрандт,
Патрик Кеттер

Санкт-Петербург — Москва
2008

Серия «High tech»

Ральф Гильдебрандт, Патрик Кеттер

Postfix. Подробное руководство

Перевод П. Шера

Главный редактор	*А. Галунов*
Зав. редакцией	*Н. Макарова*
Научный редактор	*Ф. Торчинский*
Редактор	*Е. Бочкарева*
Художник	*В. Гренда*
Корректор	*С. Минин*
Верстка	*Д. Орлова*

Гильдебрандт Р., Кеттер П.

Postfix. Подробное руководство. – Пер. с англ. – СПб.: Символ-Плюс, 2008. – 512 с., ил.

ISBN-10: 5-93286-109-6

ISBN-13: 978-5-93286-109-7

Практический подход книги «Postfix. Подробное руководство» будет полезен как специалистам, так и новичкам, предоставив им возможность управлять этим современным открытым почтовым сервером. Независимо от того, используете ли вы Postfix для потребностей маленькой компании, в качестве сервера-ретранслятора или же как корпоративный почтовый сервер, вы научитесь максимально полно использовать возможности этого мощного средства передачи корреспонденции и его ценные средства защиты.

Авторы, весьма уважаемые специалисты по Postfix, собрали в одной книге множество технической информации, документации и ответов на часто задаваемые вопросы. Рассказывается о наиболее распространенных способах применения Postfix и о редко используемых функциях, приводится масса практических примеров, показывающих пути решения таких повседневных задач, как защита пользователей от спама и вирусов, управление несколькими доменами и обеспечение роумингового доступа. Вы узнаете, как осуществлять интеграцию с OpenLDAP, MySQL или PostgreSQL, как ограничивать перемещение корреспонденции на основе черных списков, аутентифицировать пользователей, применять шифрование TLS и автоматизировать каждодневные операции. Руководство необходимо всем, кто заинтересован в использовании и понимании Postfix, начиная от домашнего пользователя и заканчивая администратором крупных почтовых систем.

ISBN–10: 5–93286–109–6

ISBN–13: 978–5–93286–109–7

ISBN 1–59327–001–1 (англ)

Издательство «Символ-Плюс». 199034, Санкт-Петербург, 16 линия, 7,
тел. (812) 324-5353, www.symbol.ru. Лицензия ЛП N 000054 от 25.12.98.
Подписано в печать 28.03.2008. Формат 70x100 1/16 . Печать офсетная.
Объем 32 печ. л. Тираж 2000 экз. Заказ N
Отпечатано с готовых диапозитивов в ГУП «Типография «Наука»
199034, Санкт-Петербург, 9 линия, 12.

Тем, кто любит
хорошее программное обеспечение

Оглавление

Отзывы на книгу «The Book of Postfix»

Многие технические книги мало чем отличаются от пересказа документации по продукту, в то время как Кеттер и Гильдебрандт проникают в самые сокровенные глубины Postfix. Дав читателям понимание основ, они принимаются за более сложные возможности Postfix. Прочитав книгу, я подумал, что если бы подобные руководства были написаны и для других почтовых программ, технология стала бы более понятной.

– Том Томас (Tom Thomas), автор книги «Network Security First-Step» (CISCO PRESS)

Postfix получает все большее распространение, в нем возникают новые дополнительные возможности, а вместе с этим растет потребность в исчерпывающем руководстве, к которому администраторы могли бы обращаться при развертывании и сопровождении своих Postfix-систем. Патрик Кеттер и Ральф Гильдебрандт – специалисты, посвятившие себя Postfix с самых первых его дней, и их книга как раз отвечает сложившейся потребности.

– Лутц Джанике (Lutz Janicke), создатель патча TLS для Postfix

Лично меня книга Ральфа и Патрика больше всего поразила тем, как они смогли сделать сложные понятия простыми для понимания. Очевидно, что авторы знают свой предмет вдоль и поперек и предлагают его в удобной для восприятия форме. Они ничего не упустили.

– Тобиас Отикер (Tobias Oetiker), автор программ Round Robin Database Tool (RRDTOOL) и Multi Router Traffic Grapher (MRTG)

В этой книге множество практических примеров и понятных объяснений – кажется, будто рядом с вами сидит специалист по Postfix.

– Дэвид Швайкерт (David Schweikert),
автор POSTGREY (Postfix greylisting policy server)

Рекомендую эту книгу всем пользователям Postfix, а особенно тем, кто планирует использовать его для проверки на вирусы AMaViS.

– Рейнер Линк (Rainer Link),
создатель OPENANTIVIRUS.ORG

Книга абсолютно необходима любому, кто заинтересован в использовании и понимании Postfix, начиная от домашнего пользователя и заканчивая администратором самых крупных почтовых систем.

– Доктор Ливиу Дайа (Liviu Daia),
старший исследователь Института
математики Румынской академии

Об авторах

Ральф Гильдебрандт и Патрик Кеттер – активные и хорошо известные в сообществе Postfix фигуры. Гильдебрандт является исполнительным директором немецкой компании T-Systems, поставщика решений в области информационных технологий и связи (ICT). Кеттер занимается информационной архитектурой в собственной компании, которая предоставляет услуги консалтинга и проектирования корпоративных телекоммуникационных систем для клиентов из Европы и Африки. Оба делают доклады о Postfix на бизнес-конференциях и встречах специалистов, а также регулярно участвуют в работе нескольких общедоступных почтовых рассылок.

Благодарности

Нам необходимо поблагодарить за эту книгу огромное количество людей, и далее каждый из нас представит свой список.

Ральф Гильдебрандт

Когда я писал эту книгу, я заметил, что очень мало знаю о том, что находится у Postfix «под капотом». Я знал, как он ведет себя, но не знал точно почему. (По крайней мере, не в каждом отдельном компоненте и не в экзотических случаях.) В некоторых случаях я чего-то не знал, в каких-то областях мои знания (или их отсутствие) оказывались ошибочными. Для получения подробной информации мне пришлось прочесть эту чертову инструкцию и задать множество вопросов в полезных рассылках postfix-users. Эта книга не сможет заменить более чем пятилетний опыт работы с Postfix, но поможет узнать его лучше.

Известно, что в 1994 году, когда я начинал заниматься UNIX, Интернет был гораздо более безопасным местом, чем сейчас. Не было никакого спама! Я познакомился с Postfix лишь потому, что у меня сломался Sendmail. Недолго попользовавшись qmail, я открыл для себя Postfix и остался верен ему. Я никогда не оглядывался назад.

Когда Билл обратился ко мне с предложением написать книгу о Postfix, я сначала сомневался. Мне был необходим соавтор, так как объем предполагаемой работы был слишком велик для одного человека. В то время Патрик занимался тем, что проклинал SASL в рассылке. Он поклялся, что если ему удастся увидеть SASL работающим, он напишет о том, как этого добиться. У Патрика получилось, и он написал руководство. Я прочитал его, оно мне понравилось, и я пригласил Патрика в соавторы.

Как оказалось, объем работы был слишком велик и для двоих, так что к нам в качестве технического редактора присоединился Брайан Уорд (Brian Ward), чей опыт оказался очень ценным в недостаточно изведанных нами областях.

Если бы не помощь Витсе Венема (Wietse Venema), Виктора Духовны (Vi(c|k)tor Duchovni), Лутца Джанике (Lutz Janicke), Андреаса Винкельмана (Andreas Winkelmann) и Питера Берингера (Peter Bieringer), эта книга никогда не стала бы такой, какая она есть, поэтому они по-

лучат по экземпляру в подарок. Не то чтобы книга была им необходима, но она, несомненно, станет замечательным подарком. Огромная благодарность и вся моя любовь моей жене Констанце, которая выдерживала мои постоянные отговорки «Но мне еще нужно написать главу!», благодаря чему я смог завершить книгу и не дал ей превратиться в химеру. Да, когда будете читать замечания Патрика, пожалуйста, имейте в виду, что я лишь слегка ненормален.

Патрик Кеттер

Пройдут годы, прежде чем Интернет предоставит нам все необходимые услуги. Как и в случае с любой другой новой средой, первым побуждением поставщиков услуг является стимуляция роста, особенно в части увеличения количества контента и услуг. Качество обслуживания и его функциональные возможности обычно остаются на втором плане, по крайней мере, до тех пор, пока обслуживание не начнет окупаться. Пока же он беззащитен перед теми, кто предпочитает злоупотребление и разрушение движению и развитию.

Так обстояли дела с электронной почтой, когда появился Postfix, предлагающий новый уровень качества обслуживания.

Когда мне потребовался собственный SMTP-сервер, я был возмущен тем, что работа с Sendmail, похоже, требует наличия какого-нибудь диплома, особенно если надо разобраться с макросами. Тогда я решил поискать другое программное обеспечение. Поиск не был долгим – я влюбился в Postfix.

Postfix показал мне, что сложное программное обеспечение может быть настроено с помощью простого и понятного структурированного синтаксиса. Если вы знакомы с SMTP, то вы уже знаете большую часть важных элементов настройки Postfix. Когда Ральф предложил мне писать книгу вместе с ним, я по-настоящему не знал SMTP. Работа над книгой заставила меня изучить намного больше, чем я предполагал, и помогла опровергнуть ряд заблуждений.

Я очень горжусь тем, что эта книга предоставила мне возможность поделиться своими знаниями о сегодняшних компьютерах и электронной почте. Надеюсь, она укажет вам путь к творческому использованию Postfix. А лучшей основой для творчества являются знания.

Эта книга не появилась бы на свет, если бы не знания, любознательность и поддержка Витсе Венема (Wietse Venema), Виктора Духовны (Vi(c|k)tor Duchovni), Ливиу Дайа (Liviu Daia), Лутца Джанике (Lutz Janicke), Флориана Кирштейна (Florian Kirstein), Уолтера Стейнсдорфера (Walter Steinsdorfer), Роланда Роллингера (Roland Rollinger), Тома Томаса (Tom Thomas), Алексея Мельникова (Alexey Melnikov), Андреаса Винкельманна (Andreas Winkelmann), Эрика «cybertime hostmaster», а также подписчиков рассылки Postfix, чьи вопросы и проблемы показали нам, чего не хватает, когда уже казалось, что все сказано.

Что самое важное, я должен поблагодарить Ральфа, чьи знания о Postfix может превзойти лишь его же мастерство использования компьютеров. Здесь он чувствует себя как рыба в воде. Это Ральф выбрал меня своим спутником в приключении под названием «Руководство по Postfix», и я очень признателен этому ненормальному парню, который стал моим близким другом, пока мы писали книгу.

Эта книга стала большим испытанием не только для меня, но и для моей жены Биргит; ее вера в меня служила мне поддержкой на протяжении этих бесконечных строк. Когда вас приглашают сделать что-то, к чему у вас лежит душа, – это большая честь. А если, когда вы наконец сделаете это, рядом с вами есть такой человек, как Биргит, – это просто дар богов.

Предисловие

Использовать слова для описания волшебства –
это все равно, что пытаться разрезать ростбиф отверткой.

– Том Роббинс

Эта книга представляет собой пошаговое руководство по Postfix. Вы начинаете читать ее, будучи новичком, а перевернув последнюю страницу, становитесь (надеемся) специалистом. Каждая глава относится к одному из трех видов: учебное пособие, теория и практика. Учебное пособие – это «букварь», который поможет вам понять суть проблемы, прежде чем пытаться реализовать ее решение в Postfix. Теоретические главы расскажут о том, как Postfix поступает в такой ситуации. Практические главы покажут, как перейти от теории к работающей системе.

Книга состоит из четырех частей, которые делят обучение работе с Postfix на следующие этапы:

Основы

Часть I представляет основы Postfix. Вы научитесь конфигурировать Postfix для сервера с коммутируемым соединением для одного домена. Кроме того, вы кратко ознакомитесь с анатомией Postfix и узнаете, какие инструменты он предлагает.

Контроль содержимого

Postfix обеспечивает широкие возможности управления потоком сообщений в вашей системе. Часть II начинается с рассказа о том, как работает протокол SMTP и каков формат сообщений электронной почты. Далее вы узнаете, как Postfix управляет различными аспектами обработки сообщений.

Сложные конфигурации

Postfix часто взаимодействует с другими приложениями сторонних компаний, такими как SQL-серверы, Cyrus SASL, OpenSSL OpenLDAP. В части III будет описано, как это делается.

Настройка Postfix

Конфигурируемое программное обеспечение всегда оставляет возможность для настройки. Часть IV поможет найти узкие места ва-

шей почтовой системы и даст советы, как повысить ее производительность.

Дополнительные ресурсы

В дополнение к данной книге и документации, поставляемой вместе с Postfix, существуют еще два ресурса, к которым можно в случае необходимости обращаться за информацией или помощью.

Документация по Postfix, практические советы и часто задаваемые вопросы

На сайте Postfix есть страница (*http://www.postfix.org/docs.html*), содержащая документацию по Postfix, практические советы (how-to) и ответы на часто задаваемые вопросы (FAQ), предложенные Postfix-сообществом.

Списки рассылки

Витсе Венема (Wietse Venema) ведет несколько списков рассылки, которые обслуживают Postfix-сообщество. На странице сайта Postfix *http://www.postfix.org/lists.html* вы найдете информацию о подписке на следующие рассылки:

`postfix-announce@postfix.org`

Список рассылки для объявлений о выходе новых редакций и версий Postfix.

`postfix-users@postfix.org`

Обмен мнениями по поводу работы с почтовой системой Postfix. Список не модерируется и требует регистрации.

`postfix-users-digest@postfix.org`

Ежедневная рассылка статей, опубликованных в списке рассылки *postfix-users*.

`postfix-devel@postfix.org`

Малопосещаемый список для людей, заинтересованных в разработке для Postfix.

Сообщество Postfix обсуждает идеи, проблемы, ошибки, патчи и многие другие вопросы в списке рассылки *postfix-users@postfix.org*. Если у вас возникнет какая-то проблема или вам захочется получить информацию о чем-то, связанном с Postfix, велики шансы, что вы найдете то, что ищете, в архивах списка рассылки. Несколько людей и компаний поддерживают архивы *postfix-users@postfix.org*, к которым можно получить доступ через броузер. Подробный перечень архивов приведен на странице списков рассылки сайта Postfix.

Обозначения в книге

Курсив

Используется для выделения названий списков рассылки, URL и адресов электронной почты.

`Моноширинный шрифт`

Используется для выделения имен доменов, демонов, файлов, каталогов, команд, имен параметров и переменных, переменных окружения, параметров командной строки.

`Моноширинный курсив`

Используется для выделения параметров и заполнителей, которые должны быть заменены соответствующими значениями в вашей системе, а также в комментариях примеров команд и кода.

`Моноширинный полужирный`

Используется для выделения команд и параметров, вводимых в окне оболочки.

`Моноширинный полужирный курсив`

Используется для выделения отдельных строк, упоминающихся в обсуждении.

Примечание

Символ `$` обозначает обычное приглашение на ввод в командной строке, символ `#` – это приглашение на ввод в оболочке для привилегированного пользователя.

Домены и имена, используемые в книге

Книга рассказывает о почтовых услугах, поэтому мы будем очень много говорить о доставке и передаче сообщений, и для примеров нам понадобятся имена доменов, отправителей и получателей. Обычно мы будем использовать следующие имена.

Локальный домен

На протяжении всей книги мы будем считать нашим собственным домен `example.com`. Почтовый сервер предположительно будет принимать (или, по крайней мере, рассматривать) сообщения для локальных пользователей `anyuser@example.com` и `anyuser@mail.example.com`. Если вы будете использовать примеры для создания собственного сервера Postfix, необходимо будет заменить `example.com` именем вашего домена.

Примечание

Конечно же, на самом деле `example.com`, `example.org` и `example.net` нам не принадлежат. IANA (Internet Assigned Numbers Authority – уполномоченная организация по распределению нумерации в сети Интернет) зарезервировала их для использования в документации.

Наш провайдер

В качестве имени домена нашего интернет-провайдера будет фигурировать `example-isp.com`.

Сценарии

Вспомогательные сценарии и другую полезную информацию, например список опечаток, вы можете найти по адресу *http://www.postfix-book.com*.

Комментарии

Если вы обнаружите в книге ошибку или захотите отправить какой-то комментарий, используйте адрес *comments@postfix-book.com*.

8

Использование ограничений на передачу сообщений

Спам – это война. Правила RFC не действуют.

– Витсе Венема

Ограничения управляют потоком сообщений, принимая решения на основе информации, передаваемой клиентом в процессе SMTP-диалога. Количество случаев, в которых можно применять ограничения, огромно, поэтому перечислять все ограничения и все их возможные параметры в этой главе мы не будем, а опишем ситуации, которые часто встречаются в списках рассылки Postfix и в повседневной работе. Для каждой ситуации будут подробно рассмотрены ограничения и их параметры, с тем чтобы вы поняли, как их реализовывать и почему они реализованы именно так.

Создание и тестирование ограничений

Прежде чем приступать к изменению ограничений по умолчанию, вы должны знать, что именно вы пытаетесь ограничить. Если вы просто включаете или выключаете булево ограничение, в этом нет ничего сложного, но если речь идет об отказе в приеме сообщений хостам, скрывающим свое происхождение, задача несколько усложняется.

В списке рассылки Postfix популярно следующее изречение: «Журнал твой – друг твой». Может быть, вам нелегко представить журнал в роли друга, но он на самом деле очень полезен при сборе информации для ограничения потока сообщений. Дело в том, что почтовый журнал хранит большую часть информации, необходимой для создания эффективных ограничений. Давайте посмотрим на записи журнала для входящего сообщения:

```
Apr 14 21:14:48 mail postfix/smtpd[31840]: 4F2A643F30:
  client=unknown[172.16.0.1] ❶
Apr 14 21:14:48 mail postfix/cleanup[31842]: 4F2A643F30:
  message-id=<002101c42254$792c2530$010010ac@stateofmind.de> ❷
Apr 14 21:14:48 mail postfix/nqmgr[31836]: 4F2A643F30:
  from=<test@example.com>, ❸
  size=666, nrcpt=1 ❹ (queue active)
Apr 14 21:14:48 mail postfix/smtpd[31840]: disconnect from unknown[172.16.0.1]
Apr 14 21:14:48 mail postfix/smtp[31844]: 4F2A643F30:
                                          to=<p@state-of-mind.de>, ❺
  relay=mail.state-of-mind.de[212.14.92.89], ❻
  delay=0, status=sent (250 Ok: queued as 97E70E1C65) ❼
```

Сообщение состоит из следующих частей:

❶ Клиент (IP-адрес и имя хоста), который доставил сообщение.

❷ Заголовок Message-Id.

❸ Отправитель конверта (команда `MAIL FROM` в SMTP-диалоге).

❹ Количество получателей.

❺ Получатель (получатели) конверта (команда `RCPT TO` в SMTP-диалоге).

❻ Где побывало сообщение.

❼ Идентификатор очереди, который присвоил сообщению удаленный сервер Postfix.

Если вам необходимо ограничить передачу сообщения и нужны данные для того, чтобы разобраться, с чем вы имеете дело, журнал – это то место, где вы можете лучше узнать своего «противника».

Моделирование работы ограничений

С первой попытки редко удается найти удачный набор ограничений. Обычно, чтобы получить желаемый результат, вы проходите через последовательность проб и ошибок. Для проверки ваших ограничений вам необходимы сообщения, к которым их можно было бы применить, при этом вполне вероятно, что в вашем распоряжении нет тестовой машины и вы создаете свои ограничения на рабочем сервере. К сожалению, это порождает риск получения ложноположительных результатов и потери важных сообщений.

Для решения проблемы Postfix предлагает для тестирования ограничений параметр `warn_if_reject`, который аналогичен действию `WARN` в проверках. Если поставить этот параметр перед ограничением, которое вы хотите проверить, то Postfix будет записывать в журнал результаты действия ограничения, но сообщения отвергать не будет. Вот как следует использовать данный параметр для проверки ограничения `reject_unknown_sender_domain`:

```
smtpd_recipient_restrictions =
   permit_mynetworks
```

```
    reject_unauth_destination
    warn_if_reject reject_unknown_sender_domain
    permit
```

Как только параметр установлен, в журнале сообщений появляются записи о мнимом отклонении сообщений:

```
Jun 25 16:10:52 mail postfix/smtpd[32511]: 8075015C02F: reject_warning: RCPT
   from sccrmhc11.comcast.net[204.127.202.55]: 550 <DickinsL@newfaces.gr>:
   Sender address rejected: Domain not found; from=<DickinsL@newfaces.gr>
   to=<example@charite.de> proto=ESMTP helo=<sccrmhc11.attbi.com>
```

После того как вы убедитесь в том, что ограничения работают, можно удалить параметр `warn_if_reject` из ограничения. Последующие журнальные записи сообщат вам об успешном отклонении сообщений:

```
Jun 25 16:11:23 mail postfix/smtpd[32511]: 8075015C02F: reject: RCPT from
   sccrmhc11.comcast.net[204.127.202.55]: 550 <DickinsL@newfaces.gr>: Sender
   address rejected: Domain not found; from=<DickinsL@newfaces.gr>
   to=<recipient@example.com> proto=ESMTP helo=<sccrmhc11.attbi.com>
```

Немедленное введение ограничений в действие

Postfix состоит из множества различных демонов, которые при запуске загружают свои конфигурации. Некоторые демоны работают лишь небольшой период времени и завершают свою работу, чтобы не создавать излишнюю нагрузку. Однако есть демоны, которые *не* перезапускаются до тех пор, пока вы не укажете Postfix на необходимость их перезапуска.

Эти долго работающие демоны, `qmgr` и `nqmgr` (он назывался `nqmgr` только в старых версиях, в новых версиях Postfix по умолчанию используется новый диспетчер очередей, который здесь носит имя `qmgr`, при этом старый диспетчер очередей называют `oqmgr`), играют важную роль в ограничении потока электронной почты и *не* замечают изменений конфигурации до тех пор, пока вся система не будет перезапущена или вы не вмешаетесь вручную. Поэтому вам необходимо помнить о том, что каждый раз, когда вы изменяете файл `main.cf` или `master.cf`, необходимо выполнить команду `postfix reload`, чтобы диспетчер очередей перезагрузил конфигурацию.

Примечание

В принципе, со временем изменения будут замечены, т. к. демоны умирают и перезапускаются по достижении значения `max_use` (конечно, за исключением `qmgr`, который никогда не умирает). Изменения параметров `qmgr` требуют обязательного выполнения команды `postfix reload`. Если же разрешить принятие изменений «со временем», то может получиться так, что одни демоны будут использовать старую конфигурацию, а другие – новую, что вряд ли будет хорошо.

Ограничения по умолчанию

Postfix поставляется с надежным набором ограничений по умолчанию, которые не дают вашему компьютеру превратиться в открытый почтовый сервер.[1] Для того чтобы познакомиться с ограничениями по умолчанию, необходимо попросить `postconf` вывести эти значения для `smtpd_recipient_restrictions`:

```
# postconf -d smtpd_recipient_restrictions
smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination
```

Postfix оценивает ограничения в том порядке, в котором они приведены. В данном случае, если клиент хочет переслать сообщение, Postfix проверяет, исходит ли соединение от хоста, указанного в `mynetworks`. Если это так (ограничение `permit_mynetworks` оценивается как `OK`), то Postfix принимает сообщение для доставки.

Если клиент не относится к сети из `mynetworks`, Postfix оценивает `reject_unauth_destination`. Это ограничение отклоняет попытки пересылки, проверяя, относится ли получатель сообщения к доменам места назначения и доменам пересылки, указанным в ваших настройках. Если получатель не относится к этим доменам, то `reject_unauth_destination` возвращает `REJECT`, и Postfix сообщает клиенту о том, что пересылка невозможна.

Если место назначения сообщения входит в зону ответственности Postfix, ограничение `reject_unauth_destination` возвращает `DUNNO`, и Postfix переходит к оценке следующего ограничения. Если же в списке больше нет ограничений, то Postfix считает, что по умолчанию подразумевается `permit`, и принимает сообщение.

Эти два ограничения – та основа, которая защищает сервер от превращения в открытый почтовый сервер, но они не защищают пользователей от спама и не заставляют клиенты вести себя корректно. В оставшейся части главы будет показано, как сделать ограничения более строгими.

Требование соответствия RFC

Требование надлежащего поведения (в соответствии с RFC) от локальных и удаленных клиентов – это первый шаг к тому, чтобы управляемый вами корабль не дал течи. Это не только гарантирует, что ваш почтовый сервер передает другим почтовым серверам корректные со-

[1] Англоязычный термин «open relay» здесь и далее вернее переводить как «открытый ретранслятор», и поскольку в тексте встречаются оба термина, просто помните, что «открытый почтовый сервер» и «открытый ретранслятор» – это одно и то же. И одинаково неприемлемы в качестве рабочей (не тестовой) системы. – *Примеч. науч. ред.*

общения, но и заставляет удаленные клиенты вести себя правильно. Такое требование полезно для защиты от спамеров, которые всегда спешат, не следуют правилам и фальсифицируют идентификационную информацию.

В этом разделе будет показано, как накладывать ограничения на имя хоста, отправителя и получателя конверта, чтобы добиться соответствия RFC.

Примечание

Приведенные ограничения будут использованы в файле `main.cf` не в том порядке, в котором они поясняются здесь. Это сделано специально, и вы увидите, зачем, в разделе «Порядок обработки RFC-ограничений» ниже в этой главе. Пока же просто добавляйте ограничения в том порядке, в котором они появляются в листингах примеров.

Ограничения на имя хоста в команде HELO/EHLO

Хорошей отправной точкой будет требование Postfix к клиентам относительно их корректного приветствия, если они хотят, чтобы их сообщения были отправлены на ваш сервер или переданы через него. Существует целый ряд ограничений, которые могут быть наложены на `HELO/EHLO`-часть SMTP-диалога, начиная с простого требования отправки имени хоста до требования отправки достоверного имени хоста.

Требование указания имени хоста

Параметр `smtpd_helo_required` требует, чтобы все клиенты, открывающие SMTP-соединение, выполняли команду `HELO` или `EHLO`. Такого обязательного приветствия требует как RFC 821 (*ftp://ftp.rfc-editor.org/in-notes/rfc821.txt*), так и RFC 2821 (*ftp://ftp.rfc-editor.org/in-notes/rfc2821.txt*), но Postfix по умолчанию устанавливает этот параметр в значение `no`. Для того чтобы включить данное требование, добавьте в файл `main.cf` такую строку:

```
smtpd_helo_required = yes
```

После перезагрузки конфигурации Postfix будет отклонять сообщения любого клиента, который не представится должным образом. Для того чтобы проверить это, подключитесь к вашему серверу и попробуйте инициировать передачу сообщения без команды `HELO`. Вот как отреагирует Postfix, требующий указания имени хоста:

```
$ 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 указания полностью определенного доменного имени хоста

Команда HELO/EHLO – это, конечно, хорошо, но клиенты также должны передать вместе с приветствием полное имя хоста (например, HELO client.example.com). Более того, RFC требуют, чтобы указывалось полностью определенное доменное имя хоста (FQDN).

Примечание

FQDN не обязательно присутствует в записях DNS.

Postfix будет отвергать сообщения от любого клиента, не предоставившего полностью определенное доменное имя хоста, если вы установите параметр reject_non_fqdn_hostname внутри ограничения smtpd_recipient_restrictions.

Предупреждение

Будьте аккуратны с этим ограничением. Некоторые почтовые клиенты, такие как Microsoft Outlook, по умолчанию используют только локальную часть имени (например, client), если только вы не настроите операционную систему так, чтобы она передавала своим приложениям полностью определенное доменное имя.

Когда вы добавите параметр reject_non_fqdn_hostname в список smtpd_recipient_restrictions, он будет иметь такой вид в файле main.cf:

```
smtpd_recipient_restrictions =
  permit_mynetworks
  reject_unauth_destination
  reject_non_fqdn_hostname
  permit
```

Проверьте ограничение, подключившись к своему почтовому серверу и указав простое имя хоста, например:

```
$ telnet mail.example.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
```

Требования к символам, составляющим имя хоста

RFC определяет, что не только имена хостов, отправляемые с командой HELO/EHLO, должны быть полностью определенными доменными именами, но и используемые для составления таких имен символы

должны подчиняться требованиям к построению имени хоста. Корректное доменное имя должно включать в себя как минимум следующие элементы:

- Домен верхнего уровня, такой как `com`
- Имя домена, например `example`
- Точку (.), разделяющую домен верхнего уровня и доменное имя

Любое другое имя хоста, скорее всего, не будет корректно разрешено, что затруднит (или даже сделает невозможным) взаимодействие между сервером и клиентом. Используя параметр `reject_invalid_hostname` в списке `smtpd_recipient_restrictions`, вы можете указать Postfix, что не следует общаться с такими клиентами. Вот пример, показывающий, куда можно вставить этот параметр:

```
smtpd_recipient_restrictions =
  permit_mynetworks
  reject_unauth_destination
  reject_non_fqdn_hostname
  reject_invalid_hostname
  permit
```

Как и прежде, проверяем ограничение, подключаясь к своему почтовому серверу от удаленного хоста и предоставляя недействительное имя хоста. В данном примере клиент представляется как «.».

```
$ 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
```

Ограничения на отправителя конверта

Доменная часть имени отправителя конверта должна содержать полностью определенное доменное имя (FQDN), и конверт должен принадлежать к существующему домену. Такие отправители конверта, как `sender` и `sender@example`, не указывают полностью определенное доменное имя. Примером полного имени отправителя конверта может быть `sender@example.com`. Неполные адреса могут вызвать путаницу, т. к. адрес отправителя выглядит так, как будто сообщение создано на данном сервере. Возможны два варианта нежелательного развития событий:

- Агент передачи сообщений, который должен возвратить сообщение с неполным именем отправителя конверта, будет возвращать его локальным пользователям. Возврат не дойдет до исходного отправителя.

- Postfix может попытаться «исправить» неправильный адрес, что только усложнит дело. Так как Postfix знает, что отправитель конверта должен иметь полностью определенное доменное имя, он запустит демон `trivial-rewrite` для канонизации адреса путем добавления `$myorigin` для отправителя `sender` (в результате получится `sender@$myorigin`) и `$mydomain` — для `sender@example` (получится `sender@example.$mydomain`). Следовательно, отправитель конверта для сообщений, полученных от удаленного сервера, будет абсолютно неверным.

Для того чтобы избежать подобных ситуаций, добавьте параметр `reject_non_fqdn_sender` в список `smtpd_recipient_restrictions`, например:

```
smtpd_recipient_restrictions =
  reject_non_fqdn_sender
  permit_mynetworks
  reject_unauth_destination
  reject_non_fqdn_hostname
  reject_invalid_hostname
  permit
```

Проверьте ограничение, подключившись с удаленной машины к вашему почтовому серверу и указав некорректное имя отправителя конверта. Покажем, как ограничения заставят Postfix отклонить сообщения 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
```

Сообщение из несуществующих доменов

Ответственный почтовый сервер не принимает сообщения от получателей, домены которых не существуют, т. к. в случае невозможности доставки сообщения он не сможет сообщить об этом отправителю. В других конфигурациях возникает двойной возврат, как только агент передачи сообщений пытается уведомить отправителя, и в конце концов сообщение с несуществующим доменом отправителя окажется в почтовом ящике администратора почтовой системы.

Примечание

Почтовым серверам приходится иметь дело с несуществующими доменами, так как пользователи иногда неправильно набирают свои почтовые адреса при настройке своих почтовых клиентов; кроме того, несуществующие домены используют спамеры, чтобы скрыть фактический источник своих сообщений.

Для защиты получателей и администраторов почтовой системы от двойного возврата и неправильно составленных сообщений добавьте параметр `reject_unknown_sender_domain` в список `smtpd_recipient_restrictions`, например:

```
smtpd_recipient_restrictions =
  reject_unknown_sender_domain
  permit_mynetworks
  reject_unauth_destination
  reject_non_fqdn_hostname
  reject_invalid_hostname
  permit
```

Следующий пример показывает, как можно проверить ограничение (ищем код ошибки 450, который Postfix отправляет в качестве ответа команде `MAIL FROM`):

```
$ 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
```

Ограничения на получателя конверта

В качестве последнего действия по приведению входящих соединений в соответствие RFC вы можете отвергать сообщения, в которых для получателя конверта указан несуществующий домен или пользователь.

Почтовый сервер не должен принимать никакие сообщения для несуществующего домена, т. к. их невозможно доставить. Если почтовый сервер примет сообщение, а затем вернет его, то пользователь может подумать, что у сервера какие-то проблемы, ведь изначально сообщение было принято.

Если настроить почтовый сервер так, чтобы сообщения в несуществующие домены отклонялись, то ваши проблемы станут проблемами того клиента или пользователя, которые отправили сообщение. Для введения такого ограничения используйте параметр `reject_unknown_recipient_domain` в списке `smtpd_recipient_restrictions`, например:

```
smtpd_recipient_restrictions =
  reject_unknown_recipient_domain
  permit_mynetworks
  reject_unauth_destination
  reject_non_fqdn_hostname
  reject_invalid_hostname
  permit
```

Как обычно, вы можете протестировать ограничение, отправив письмо в несуществующий домен получателя при ручном подключении к серверу. Покажем на примере, как Postfix отвергает сообщение из-за того, что домен `invalid.domain` не существует:

```
$ 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
```

Сообщения неизвестным получателям

Вы можете настроить Postfix так, чтобы сообщения для неизвестного пользователя из вашего домена доставлялись администратору почтовой системы. На первый взгляд такая идея кажется весьма удачной, т. к. администратор может изучить сообщение и при наличии возможности доставить его вручную.

Но несмотря на то, что в теории подобная конфигурация могла бы обеспечить идеальное обслуживание клиентов, использование адресата по умолчанию может привести к DoS-атакам (Denial-of-service – отказ в обслуживании) на ваш сервер, как только он станет целью для спамера или червя, использующего *атаку по словарю*. Такая атака заключается в попытке отправить сообщение существующим получателям за счет рассылки сообщений по адресам, составленным из всех возможных сочетаний букв. Например, атакующий может начать с адреса `aa@yourdomain.com`, затем попробовать `ab@yourdomain.com` и так пройти через все двухбуквенные комбинации вплоть до `zz@yourdomain.com`.

Дело не только в сложности отделения законных сообщений от сообщений, созданных в процессе подобной атаки, но и в том, что сервер подвергается риску в связи с исчерпанием пропускной способности канала, производительности процессора, памяти и дискового пространства, и в результате сдается и останавливает обслуживание запросов на передачу сообщений. Например, вирус Sobig.F привел к перегрузке многих почтовых серверов в августе 2003 года.

Помните, что для Postfix приоритетом является надежность обслуживания. Надежность подразумевает согласованность, и поэтому сервер отвергает почту, адресованную неизвестным пользователям, по умолчанию без какого-либо ручного вмешательства. Это замечательно для автономного сервера Postfix, но полезно и для сервера Postfix, работающего на интеллектуальном хосте, который защищает остальные почтовые серверы.

Postfix определяет корректность адресов получателей, сверяясь с картами. Существуют два параметра конфигурации, которые указывают

Postfix, где следует искать такую информацию: local_recipient_maps и relay_recipient_maps. В обоих параметрах указывается одна или несколько карт, содержащих действительных получателей. Параметр local_recipient_maps определяет действительных локальных получателей, как показано в примере, где получатели определены в файле паролей UNIX и картах псевдонимов:

```
# postconf -d local_recipient_maps
local_recipient_maps = proxy:unix:passwd.byname $alias_maps
```

В то же время параметр relay_recipient_maps определяет получателей, для которых Postfix пересылает сообщения к конечному месту назначения (серверу почтовых ящиков):

```
# postconf -d relay_recipient_maps
relay_recipient_maps = hash:/etc/postfix/relay_recipients
```

При использовании relay_recipient_maps позаботьтесь о том, чтобы Postfix знал всех действительных получателей в тех системах, куда он осуществляет пересылку. Если место назначения – это сервер Microsoft Exchange, обратитесь к главе 13 за сведениями о том, как можно извлечь карту пользователей.

Предупреждение

Использование luser_relay отменяет параметр local_recipient_maps, т. к. делает действительными всех локальных получателей. Аналогично запись с групповым именем catchall в списке virtual_alias_maps отменяет отклонение почты, адресованной несуществующим получателям, т. к. групповое имя делает действительными всех получателей. Например, следующая запись карты делает действительными всех получателей в домене example.com:

```
@example.com    catchall@localhost
```

Сообщения получателям с неполным именем

Неполностью указанный адрес, такой как recipient, содержит только локальную часть электронного адреса. С приемом таких сообщений для локальных пользователей на компьютере, получающем почту только для одного домена, проблем нет, но трудности появляются, если ваш почтовый сервер получает сообщения и для других доменов.

Есть указана одна лишь локальная часть, то остается слишком много простора для интерпретации почтового адреса.

Пусть, например, вы работаете интернет-провайдером для двух конкурирующих компаний, example.com и example.net. Если вы получите сообщение для получателя sales, куда оно будет отправлено? По какому адресу оно должно попасть – sales@example.com или sales@example.net, если один и тот же сервер обслуживает оба этих электронных адреса?

В силу вышесказанного вам следует отклонять сообщения с неполными адресами. Не принимайте на себя чужую ответственность. Подго-

товка сообщения для корректной доставки – это задача отправителя, и он должен определить получателя уникальным способом.

Примечание

Существует всего одно исключение: вы должны принимать сообщения для postmaster при неполностью указанном адресе. На адрес postmaster не накладываются вообще никакие ограничения, действующие для получателей (включая ограничения на отправителя, команду helo и клиент).

Postfix будет отвергать сообщения для получателей с неполным именем, если вы добавите параметр reject_non_fqdn_recipient в ваш список smtpd_recipient_restrictions, как в следующем 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
```

Проверим ограничение, подключившись к своему почтовому серверу с удаленного компьютера и отправив сообщение с неполным адресом получателя. Для проверки работы ограничения достаточно будет такого сеанса:

```
$ 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
```

Обеспечение соответствия RFC

Возможно, вы уже обратили внимание на то, что ограничения могут стать довольно сложными. И чем сложнее становятся ограничения, тем выше вероятность того, что среди них появится такое, которое приведет к неправильной работе вашей почтовой системы (или вообще сделает ее абсолютно бесполезной), отвергая сообщения, которые должны быть приняты при любых обстоятельствах. Последующие разделы покажут вам, как избежать непредумышленной блокировки некоторых или всех отправителей. Это важно, т. к. вы можете случайно исключить отправителей, которые могли бы сообщить о недостатках вашей конфигурации.

Пустое имя отправителя конверта

Во-первых, никогда не блокируйте пустого отправителя конверта (<>). Этот адрес принадлежит MAILER-DAEMON, почтовый сервер использует его при отправке возвратов и уведомлений о состоянии. Если заблокировать этот адрес, то удаленные серверы не смогут сообщить вашим пользователям о том, что с отправленными ими сообщениями возникли проблемы.

Предупреждение

Черные списки, такие как `dsn.rfc-ignorant.org`, приводят перечень почтовых серверов, которые категорически отказываются принимать почту от отправителей конвертов с пустым именем, так что использующие эти черные списки почтовые серверы не принимают почту от перечисленных там серверов (мы вернемся к этому вопросу в разделе «Отказ доменам отправителей из черных списков»).

Все, что вам нужно, – это рассматривать пустой адрес отправителя конверта как любой другой допустимый адрес и создать хорошие ограничения (противодействующие спаму) для защиты ваших получателей. Пусть ограничения выполняют свою работу, и если вы получите сообщение с пустым именем отправителя, примените его. В конце концов, любой адрес отправителя может оказаться подделкой...

Специальные учетные записи

На почтовом сервере имеются два адреса, для которых вы всегда должны принимать сообщения; они необходимы для того, чтобы работа почтового сервера соответствовала RFC:

`postmaster`

Всегда принимайте почту, адресованную администратору почтовой системы `postmaster`, – это центр обработки информации для вопросов, связанных с электронными сообщениями. Пользователи должны иметь возможность обратиться к администратору за помощью (см. RFC 2821 по адресу *http://www.rfc-editor.org/rfc/rfc2821.txt*).

`abuse`

Прием почты для адресата `abuse` гарантирует, что пользователи смогут уведомить вас о возможных почтовых злоупотреблениях, исходящих от вашего сервера (см. RFC 2142 по адресу *http://www.rfc-editor.org/rfc/rfc2142.txt*).

Дополнительно (но не обязательно) вы можете принимать сообщения для следующих адресатов, если поддерживаете соответствующие серверы (см. RFC 2142; *http://www.rfc-editor.org/rfc/rfc2142.txt*):

`webmaster`

Принимайте почту для `webmaster`, если у вас работает веб-сервер.

`hostmaster`

Принимайте почту для `hostmaster`, если у вас работает сервер имен.

Вы можете настроить прием сообщений для этих получателей, используя параметр `check_recipient_access` в сочетании с картой, такой как `/etc/postfix/roleaccount_exceptions`, где перечислены получатели, сообщения для которых должны быть приняты. Такая карта может выглядеть следующим образом (значение `OK` для каждого ключа карты сообщает Postfix о том, что следует принимать сообщения для данного получателя без учета ограничений для получателей):

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

После создания этого файла преобразуйте его в карту командой `postmap hash:/etc/postfix/roleaccount_exceptions`. Затем укажите карту в качестве значения параметра `check_recipient_access` в списке ограничений файла `main.cf`, например:

```
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/roleaccount_exceptions
   permit
```

После перезагрузки конфигурации вы можете спокойно переходить к созданию более сложных правил. Карта с исключениями запрашивается после того, как Postfix проводит проверки на неавторизованную пересылку, так что использование адреса `postmaster@` будет безопасным.

Порядок обработки RFC-ограничений

Возможно, вы обратили внимание на то, что параметры, добавляемые в список `smtpd_recipient_restrictions` в предыдущих разделах, указывались не в том порядке, как сами разделы. Это объясняется тем, что параметры ограничений могут влиять один на другой и мешать друг другу, если будут указаны не в нужном порядке. Давайте, например, посмотрим на такой список:

```
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/roleaccount_exceptions
reject_non_fqdn_hostname
reject_invalid_hostname
permit
```

Параметр `permit_mynetworks` обозначает важную границу между клиентами вашей внутренней сети и внешними клиентами. Параметры, заданные выше этого элемента (включая его самого), относятся как к внутренним, так и к внешним клиентам, в то время как параметры ниже `permit_mynetworks` применяются только к внешним клиентам.

Параметры, предшествующие `permit_mynetworks`, требуют базового соблюдения RFC от всех клиентов как внутри вашей сети, так и вне ее.

Параметр `reject_unauth_destination` не дает вашему серверу превратиться в открытый почтовый сервер. Лучше не указывать никакие параметры, разрешающие пересылку сообщений, пока не задан параметр `permit_mynetworks`. Далее как можно раньше следует указать `reject_unauth_destination`, чтобы быть уверенным в том, что неавторизованный хост никоим образом не сможет использовать ваш сервер как открытый почтовый сервер.

Проверка на SMTP-аутентификацию должна находиться между `reject_unauth_destination` и `permit_mynetworks`. Затем, прежде чем указывать еще какие-то параметры отказа от сообщений, используйте параметр `check_recipient_access` для разрешения безусловной доставки специальным почтовым ящикам вашей системы.

Наконец, после отражения возможных попыток возврата спама множественным получателям и отказа в приеме сообщениям с фальшивыми именами хостов получателей конверта вы можете принимать сообщения, используя параметр `permit`.

Меры борьбы со спамом

Спамерам необходимо замаскировать источник своих сообщений, если они не хотят судебного преследования.[1] Обычно они подделывают адрес отправителя конверта или пытаются усыпить бдительность принимающего сервера, сообщая ему, что их клиенту можно доверять – как

[1] По состоянию на март 2008 года законодательство РФ не предусматривает, к сожалению, никакой ответственности за рассылку спама; единственным ограничением для спамера является типовой договор клиента с провайдером, в котором клиент обычно обязуется не использовать предоставленный доступ в Интернет для осуществления массовых непрошеных рассылок. – *Примеч. науч. ред.*

будто он принадлежит к локальной сети. Ограничения могут проверить и отклонить такие сообщения. Более того, они могут запрашивать черные списки, в которых перечислены спамеры и другие адресаты, сообщения от которых вы не хотите принимать. В этом разделе будет показано, как провести такие ограничения в жизнь.

Предотвращение явных фальсификаций

Некоторые спам-программы пытаются скрыть источник сообщения, используя имя хоста вашего почтового сервера как свое собственное в приветствии HELO/EHLO. Postfix воспринимает ситуацию как парадоксальную, ведь единственный хост, который может использовать имя хоста сервера, – это сам сервер. Однако Postfix никогда не стал бы подключаться к своему демону smtpd для отправки почты самому себе, если только не была сделана ошибка конфигурации, приведшая к возникновению петли.

Добавление ограничений после строки permit_mynetworks сделает их применимыми только к внешним клиентам, но не к прокси-фильтрам или локальным клиентам с неполной реализацией SMTP.

Поэтому вы можете отказывать в SMTP-соединении любому клиенту, который приветствует ваш почтовый сервер с именем хоста этого сервера. Для этого сначала создайте файл карты с именем /etc/postfix/helo_checks, содержащий различные вариации имени вашего хоста. Приведем несколько примеров для имени хоста, IP-адреса хоста и IP-адреса в скобках, которые не должны использоваться внешними клиентами:

```
/^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
```

Документ RFC 2821 указывает, что сам по себе IP-адрес не является разрешенным аргументом для команды HELO. IP-адрес разрешен, если он задан в форме [ipv4address] (в квадратных скобках) или как IPv6-адрес, [ipv6:ipv6address], опять-таки в квадратных скобках. Чтобы соблюсти все формальности и отказать в обслуживании клиентам, которые отправляют IP-адрес без квадратных скобок, добавьте такую строку:

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

Для того чтобы привести карту в действие, укажите ее (и ее тип) как аргумент для параметра check_helo_access в списке smtpd_recipient_restrictions, например:

```
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/roleaccount_exceptions
reject_non_fqdn_hostname
reject_invalid_hostname
check_helo_access pcre:/etc/postfix/helo_checks
permit
```

Для проверки ограничения подключитесь к своему почтовому серверу и укажите собственное имя в приветствии HELO. Вы должны получить отказ, как показано в примере:

```
$ 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
```

Фиктивные записи сервера имен

Postfix может отвергать сообщения, если очевидно, что записи сервера имен для домена HELO, доменов отправителя и получателя отсутствуют или не позволяют обеспечить корректную передачу сообщения. Приведем ряд обстоятельств, которые могут показаться подозрительными в записях DNS:

Фальшивые сети

Некоторые почтовые серверы утверждают, что относятся к сетям, недостижимым для Postfix, в том числе неиспользуемым вами частным сетям (см. RFC 1918, *ftp://ftp.rfc-editor.org/in-notes/rfc1918.txt*), сети обратной связи (loopback), широковещательным сетям или сетям многоадресной рассылки.

Пристанища спамеров

Пристанища спамеров (spam havens) – это сети, про которые известно, что они принадлежат спамерам или предоставляют услуги спамерам. Можно отклонять все сообщения из таких доменов. Информацию о пристанищах спамеров и их деятельности вы можете найти в ROKSO (Register of Known Spam Operations – список известных спамеров, *http://www.spamhaus.org/rokso/index.lasso*).

«Безразличные» агенты передачи сообщений

«Безразличные» (wildcard) агенты передачи сообщений заявляют, что они ответственны за все домены, даже за несуществующие. Казалось бы, это не должно стать проблемой, ведь вы можете отказать в доступе в случае неизвестных доменов получателя и отправителя.

К сожалению, некоторые регистраторы доменов прониклись гениальной идеей перенаправлять неизвестные доменные имена в свой собственный домен. В результате неизвестные домены получают действительную A-запись, что делает бесполезными параметры ограничений `reject_unknown_sender_domain` и `reject_unknown_recipient_domain`.

Примечание

Первым доменным регистратором, перенаправившим неизвестные домены, стал VeriSign (*http://www.verisign.com*) в 2003 году. VeriSign злоупотребил своей властью над пространствами имен `.net` и `.com` и пересылал все несуществующие домены `.com` и `.net` на собственный сайт (`sitefinder.verisign.com`). Кроме того, VeriSign создал для неизвестных доменов собственную почтовую службу, что сделало невозможным отклонение сообщений из неизвестных доменов. Это явное приглашение для спамеров, и вы можете отклонять сообщения от «безразличных» агентов передачи сообщений, блокируя MX-хосты в таких доменах.

Во всех описанных случаях используются либо фальшивые записи сервера, либо поддержка спамеров. Для отклонения почты из таких доменов и сетей вы можете создать в файле `/etc/postfix/bogus_mx` карту, содержащую IP-адреса вместе с типом ответа, который вы хотите им дать (полный перечень таких ответов приведен в приложении C). Рассмотрим 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
224.0.0.0/4        550 Mail server in class D multicast network
192.168.0.0/16     550 No route to your RFC 1918 network
# spam havens
69.6.0.0/18        550 REJECT Listed on Register Of Known Spam Operations ❶
# wild-card MTA
64.94.110.11/32    550 REJECT VeriSign Domain wildcard ❷
```

❶ На момент написания книги эта сеть упоминалась в списке на сайте *spamhaus.org* (*http://www.spamhaus.org/sbl/sbl.lasso?query=SBL6636*) как известная своей спамерской деятельностью.

❷ На момент написания книги было известно, что этот хост действует как «безразличный» MTA.

Мы редактируем карту типа CIDR, которая относится к последовательному типу (см. главу 5), поэтому нам не нужно и мы не можем преобразовывать ее при помощи `postmap`. Postfix будет использовать файл «как есть». Просто добавьте параметр `check_sender_mx_access`, указав карту в качестве аргумента, в свой список `smtpd_recipient_restrictions`, например:

```
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/roleaccount_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 вступит в силу после перезагрузки параметров. Его действие найдет отражение в почтовом журнале:

```
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.com>
```

Вы можете проверить IP при помощи команды `host`:

```
# host -t mx joymail.com
# host -t a joymail.com
joymail.com has address 64.94.110.11
```

Примечание

Этот домен действительно существует в настоящее время; судя по всему, он был зарегистрирован в октябре 2003 года.

Возврат множеству получателей

В разделе «Пустое имя отправителя конверта» вы узнали о том, что не следует блокировать сообщения с пустым именем отправителем конверта. Это правило имеет одно исключение – необходимо блокировать сообщения с пустым именем отправителя конверта, отправленные множеству получателей. Дело в том, что в настоящее время нет оснований для легальной рассылки уведомлений о состоянии многочисленным получателям, так что любые такие сообщения, вероятно, являются запрещенными.

Для отказа в приеме сообщениям с пустым именем отправителя конверта, предназначенным нескольким получателям, добавьте параметр `reject_multi_recipient_bounce` в свой список `smtpd_recipient_restrictions`. Этот параметр может быть добавлен практически в любое место списка ограничений; в данном примере он поставлен в `smtpd_data_restrictions`:

```
smtpd_data_restrictions =
  reject_multi_recipient_bounce
```

В документации говорится, что параметр `reject_multi_recipient_bounce` может надежно использоваться только в `smtpd_data_restrictions`, когда известны все получатели.

Проверить это ограничение вы, как и раньше, можете посредством ручного подключения к почтовому серверу. Передача пустого имени отправителя конверта и нескольких получателей приведет к отказу, как показано в следующем примере:

```
$ 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:<>
250 Ok
RCPT TO: <recipient1@example.com>
RCPT TO: <recipient2@example.com>
550 : Recipient address rejected: Multi-recipient bounce
QUIT
221 Bye
```

Использование черных списков DNS

DNS-сервер черных списков – это сервер, который сообщает вам о ресурсах (IP-адресах, отправителях конвертов и доменах), которым, вероятно, не стоит доверять. Правильно выбранные черные списки могут быть чрезвычайно полезны для блокирования почты, отправленной клиентами на ваш сервер. Однако неверный выбор черного списка может заставить ваш сервер отвергать почту, которую вы считаете допустимой. Обязательно проверяйте принципы составления черного списка, прежде чем им воспользоваться. Любой сайт, ведущий черный список, должен представлять перечень условий, по которым ресурс вносится в черный список; кроме того, должна быть опубликована процедура удаления из списка ресурса, который более не должен там находиться.

Если вы ищете черный список, то можете начать с сайта *dmoz.org* (*http://dmoz.org/Computers/Internet/Abuse/Spam/Blacklists*).

Предупреждение

Основой всех черных списков является служба доменных имен, т. е. Postfix должен выполнять поиск в DNS. Некэшированный поиск в DNS может занять около секунды, и в случае тайм-аута скорость, с которой сервер может прини-

мать сообщения, значительно уменьшается. Поэтому проверки по черным спискам достаточно затратны с точки зрения времени отклика. Вы должны использовать их в своем списке ограничений только в качестве последнего средства.

Отказ клиентам из черных списков

Вы можете отвергать занесенные в черный список клиенты при помощи списков DNSBL (DNS-based Blackhole List). В Postfix есть параметр `reject_rbl_client`, который принимает в качестве аргумента полное имя хоста сервера черных списков. Приведем пример использования этого параметра:

```
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/roleaccount_exceptions
  reject_non_fqdn_hostname
  reject_invalid_hostname
  check_helo_access pcre:/etc/postfix/helo_checks
  reject_rbl_client relays.ordb.org
  permit
```

Новый параметр вступит в силу после перезагрузки параметров.

Примечание

Для того чтобы проверить, упомянут ли клиент в списке DNSBL, измените на обратный порядок четырех октетов IP-адреса клиента (т. е. замените `a.b.c.d` на `d.c.b.a`), добавьте в конец `rbl.domain` (например, `relays.ordb.org`) и ищите в списке полученное значение. Если хост занесен в черный список, то вы получите ответ, указывающий на исходный IP-адрес, как в следующем примере:

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

Многозначные результаты

Postfix может обработать дополнительную информацию, когда известно не только то, что хост занесен в черный список: возвращенный IP-адрес дает возможность определить, почему он туда занесен. Например, следующая конфигурация будет отклонять сообщения от любого хоста, который соответствует A-записи `127.0.0.2` в нашем воображаемом черном списке `domain.tld`:

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

Отказ отправителям из доменов черных списков

В дополнение к запрещению почты от определенных IP-адресов вы можете блокировать сообщения тех отправителей, домены которых занесены в черные списки. Такие списки называются RHSBL (Right-Hand-Side Blacklist – черный список правых частей)). Настройка Postfix для использования RHSBL требует выполнения тех же операций, что и для DNSBL. В качестве примера в этом разделе будет использоваться специальный список с сайта *dsn.rfc-ignorant.org*:

> **Программное заявление** *www.rfc-ignorant.org*:
>
> Мы ведем ряд списков (в настоящее время `dsn`, `abuse`, `postmaster`, `bogusmx` и `whois`), содержащих домены, администраторы которых решили не подчиняться требованиям RFC – стандартным правилам Сети.
>
> Важно отметить, что НИЧТО и НИКОГО не заставляет соответствовать RFC (Request for Comments – запрос на комментарии), однако возможность совместной работы, достигнутая в Сети, основывается на том, что у всех есть один и тот же свод правил и все ему следуют. Присутствие в списке означает только то, что для домена было решено не реализовывать условия, описанные в определенном RFC. Естественно, решение относительно того, взаимодействовать ли, например, с хостами доменов, не соответствующими RFC 2142 и имеющими работающий адрес `<abuse@domain>`, остается исключительно за вами.
>
> – dredd, *www.rfc-ignorant.org*

Существует множество агентов передачи сообщений, которые принимают почту не так, как этого требуют RFC (например, они могут отклонять сообщения с пустым именем отправителя конверта), по ряду ошибочных причин, включая такие:

- запрещение возвратов сообщений от анонимных отправителей;
- запрещение пустого имени отправителя (для борьбы со спамом).

Прокомментируем такое ошибочное поведение не соответствующих RFC почтовых серверов: кто угодно может подделать любой электронный адрес. Вы можете отправлять сообщения от имени `president@whitehouse.gov`, и они будут такими же анонимными, как и сообщения с пустым именем отправителя.

Спам может отправляться произвольными отправителями, но возвраты могут отправляться *только* с пустым именем отправителя конверта.

Любой почтовый сервер, блокирующий пустых отправителей конвертов, не дает своим пользователям возможности узнать о том, что их сообщения, возможно, были отклонены другим почтовым сервером: возвраты, отправляемые другим соответствующим RFC сервером, будут отклонены, т. к. для возврата используется пустое имя отправителя, как это и описано в RFC.

RFC 2821 явно определяет, что агент передачи сообщений *должен* принимать сообщения с пустым обратным путем (адресом отправителя конверта), т. к. использование пустого имени отправителя при отправке уведомления о возврате предотвращает бесконечные скитания недоставимого уведомления от одной системы к другой и обратно.

Postfix имеет параметр `reject_rhsbl_sender`, который отделяет локальную часть электронного адреса и использует доменную часть для обращения к черному списку (такому как `dsn.rfc-ignorant.org`). Если домен отправителя конверта внесен в черный список, то Postfix отклоняет входящее сообщение. Как и другие параметры черных списков, этот параметр должен быть помещен в конец списка ограничения `smtpd_recipient_restrictions`, например:

```
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_rbl_client relays.ordb.org
  check_recipient_access hash:/etc/postfix/roleaccount_exceptions
  reject_non_fqdn_hostname
  reject_invalid_hostname
  check_helo_access pcre:/etc/postfix/helo_checks
  reject_rhsbl_sender dsn.rfc-ignorant.org
  permit
```

После перезагрузки изменения вступят в силу, и вы сможете проверить ограничение, подключившись к вашему серверу и используя отправителя конверта из домена, входящего в список `.rfc-ignorant.org`, как в следующем примере (`sender@example.com` – это официальный адрес для проверок):

```
$ 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: <recipient@example.com>
554 Service unavailable; Sender address [sender@example.com] blocked \
    using dsn.rfc-ignorant.org; Not supporting null originator (DSN)
QUIT
221 Bye
```

Проверка вручную на вхождение в черный список

Проверка домена на вхождение в RHSBL аналогична процедуре проверки IP-адреса, с тем лишь отличием, что не нужно менять порядок элементов. Просто добавьте имя сервера черных списков после имени домена, который вы хотите проверить, и выполните поиск в DNS.

Здесь проверка была проведена для домена, которого не оказалось в черном списке:

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

Если же домен найден в черном списке, то результат будет таким:

```
$ host example.com.dsn.rfc-ignorant.org
example.com.dsn.rfc-ignorant.org has address 127.0.0.2
```

Исключения для доменов отправителя из черных списков

Если вы хотите отклонять сообщения от почтовых серверов, которые не следуют правилам, но при этом вам необходимо поддерживать связь с некоторым доменом, который попадает под ограничения (так что почта из него была бы отклонена), можно создать список исключений. Для реализации исключений используйте параметр `check_sender_access` и карту исключений.

Сначала создайте файл, например `/etc/postfix/rhsbl_sender_exceptions`, содержащий пользователей и домены, от которых вы хотите принимать сообщения. Например, следующий файл разрешает прием сообщений от всех пользователей `example.com` и одного пользователя `sender@example.org`:

```
example.com                OK
sender@example.org         OK
```

Используйте команду `postmap hash:/etc/postfix/rhsbl_sender_exceptions` для создания карты. Затем добавьте параметр `check_sender_access` сразу же после параметра `reject_rhsbl_sender`, например:

```
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_rbl_client relays.ordb.org
  check_recipient_access hash:/etc/postfix/roleaccount_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
```

Примечание

В примере использован параметр `check_sender_access`, всего же для исключений могут использоваться четыре параметра:

- `check_sender_access`
- `check_client_access`
- `check_helo_access`
- `check_recipient_access`

Вы уже встречались с параметром `check_helo_access` в разделе «Предотвращение явных фальсификаций». Сведения об оставшихся двух параметрах вы найдете в документации Postfix.

Проверка отправителя

Бриллиант в короне средств Postfix борьбы со спамом – это проверка адреса отправителя, в ходе которой проверяется, существует ли в домене отправителя адрес отправителя, и если такого отправителя не существует, то Postfix не принимает сообщение.

К сожалению, эта функциональность является весьма дорогостоящей, т. к. проверка занимает много времени и требует дополнительных системных ресурсов. Рассмотрим ее пошагово:

1. Клиент передает данные отправителя конверта.
2. Postfix формирует и ставит в очередь пробное сообщение отправителю конверта.
3. Postfix ищет MX- или A-запись домена отправителя конверта.
4. Postfix пытается подключиться к почтовому серверу отправителя. Если подключиться к удаленному серверу не удается, то `smtpd` откладывает решение о принятии сообщения, возвращая клиенту код временной ошибки 450. Тем временем Postfix продолжает пытаться проверить адрес.
5. Postfix инициирует сеанс SMTP с удаленным сервером.
6. Postfix передает данные отправителя конверта удаленному почтовому серверу в качестве сведений о получателе конверта.
7. В зависимости от ответа удаленного сервера Postfix делает одно из двух:
   - Если удаленный почтовый сервер принимает получателя (отправителя исходного конверта), то Postfix отключается, уничтожает пробное сообщение и принимает сообщение от исходного клиента.
   - Если удаленный почтовый сервер отклоняет получателя (отправителя исходного конверта), то Postfix отключается, уничтожает пробное сообщение и отклоняет сообщение от клиента.

При включенной проверке адресов отправителей сообщения обычно будут испытывать задержку до 9 секунд на проверку адреса, встретив-

шегося *впервые*. Однако затем Postfix кэширует статус адреса, так что последующие сообщения не будут задерживаться.

Если проверка длится более 9 секунд, то `smtpd` отвергает сообщение клиента (отправляющего компьютера) с кодом 450. Обычные почтовые клиенты повторят попытку через некоторое время, а захваченные (hijacked) прокси-серверы – нет, т. к. они занимаются только пересылкой команд SMTP, и человек, управляющий таким прокси-сервером, не захочет терять дополнительное время.

Настройка проверки адреса отправителя

Для включения проверки адреса отправителя добавьте параметр `reject_unverified_sender` в свой список ограничений `smtpd_recipient_restrictions`, например:

```
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_rbl_client relays.ordb.org
  check_recipient_access hash:/etc/postfix/roleaccount_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
```

Кроме `reject_unverified_sender` существуют и другие параметры, которые можно добавить в ограничения. Однако параметры обладают разумными значениями по умолчанию, и служат они скорее для регулировки проверки адреса отправителя, чем для ее настройки. Следующие подразделы описывают наиболее часто используемые изменения проверки адреса отправителя. Дополнительные параметры настройки вы сможете найти в файле `ADDRESS_VERIFICATION_README`, который присутствует в дистрибутиве Postfix.

Отправитель пробного конверта

Когда Postfix формирует пробное сообщение для проверки отправителя, он сам должен представиться удаленному серверу – указать своего отправителя конверта. Вы можете задать этот адрес в параметре `address_verify_sender`. Значение по умолчанию – `postmaster@$myorigin`.

При желании указать другого отправителя конверта для пробного сообщения добавьте параметр `address_verify_sender` в файл `main.cf`:

```
address_verify_sender = sender@example.com
```

Конечно, такой адрес отправителя должен существовать; не забывайте о том, что другие серверы также могут использовать по отношению к вам механизм проверки адреса отправителя.

Примечание

На адрес получателя, указанный в параметре `address_verify_sender`, не накладываются никакие ограничения.

Кэширование

По умолчанию Postfix хранит проверенные адреса отправителей в оперативной памяти. Когда вы перезагружаете параметры или перезапускаете Postfix, то теряете их, если только не будете использовать (что не обязательно) дополнительную базу данных для постоянного хранения адресов. Для того чтобы работать с базой данных, задайте путь к ней в параметре `address_verify_map` (убедившись, что в выбранной файловой системе достаточно *много* свободного места), например:

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

После перезагрузки Postfix создаст базу данных и будет добавлять туда результаты как положительных, так и отрицательных проверок. Если вы хотите отменить сбор отрицательных данных, задайте параметр `address_verify_negative_cache` в файле `main.cf`:

```
address_verify_negative_cache = no
```

Выборочная проверка адресов отправителей

По мере возрастания нагрузки на ваш почтовый сервер процедура проверки адресов получателей, вероятнее всего, приведет к нехватке ресурсов. В этот момент следует перейти к выборочной проверке адресов отправителей.

Выборочная проверка адресов отправителей работает на основе карты обычно используемых спамерами доменов отправителей конвертов. Если домен отправителя входящего сообщения присутствует в карте, то Postfix проверяет отправителя, иначе – не беспокоится. Вам нужно создать файл карты, например `/etc/postfix/common_spam_senderdomains`, и указать параметр `reject_unverified_sender` в качестве действия, которое следует предпринять в случае совпадения с доменом отправителя конверта. Приведем пример того, как может выглядеть такой файл:

```
hotmail.com reject_unverified_sender
web.de      reject_unverified_sender
msn.com     reject_unverified_sender
mail.ru     reject_unverified_sender
```

Страница руководства access(5) поясняет, что правая часть карты – это имя действующего ограничения или класса `smtpd_restriction_class`.

В данном примере при инициировании клиентом передачи сообщения Postfix делает одно из двух:

- Если домену отправителя соответствует запись в common_spam_senderdomains, то просмотр карты возвращает действие reject_unverified_sender, и Postfix проверяет отправителя конверта. В случае успешной проверки reject_unverified_sender возвращает DUNNO, и Postfix переходит к оценке следующего ограничения. Если же адрес недействителен, то Postfix отклоняет сообщение.
- Если домену отправителя не соответствует никакая запись в common_spam_senderdomains, то поиск в карте не дает результата, и выборочный оценщик возвращает DUNNO, а Postfix оценивает следующее ограничение, не проверяя адрес отправителя.

После создания карты преобразуйте ее в базу данных, используя команду postmap hash:/etc/postfix/common_spam_senderdomains. Наконец, замените существующий параметр reject_unverified_sender на параметр check_sender_access с картой в качестве аргумента. Приведем пример, в котором используется карта hash:/etc/postfix/common_spam_senderdomains:

```
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/roleaccount_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_senderdomains
  permit
```

Вы можете пойти дальше и ввести, помимо отправителя конверта, дополнительные критерии проверки, например содержимое сообщения. Создайте новую карту с именем common_spam_senderdomain_keywords – она будет содержать ключевые слова из имен доменов, которые будут запускать 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
```

Затем добавьте еще один параметр check_sender_access, указывающий на новую 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
  check_recipient_access hash:/etc/postfix/roleaccount_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_senderdomains
  check_sender_access regexp:/etc/postfix/common_spam_senderdomain_keywords
  permit
```

Порядок введения ограничений

Борьба со спамом весьма затратна с точки зрения системных ресурсов. Рассмотрим пример того, как следует упорядочивать параметры, предотвращающие прием спама:

```
smtpd_recipient_restrictions =
  reject_non_fqdn_recipient
  reject_non_fqdn_sender
  reject_unknown_sender_domain
  reject_unknown_recipient_domain
  permit_mynetworks ❶
  (permit_sasl_authenticated)
  (pop-before-smtp)
  reject_unauth_destination
  check_recipient_access hash:/etc/postfix/roleaccount_exceptions
  check_helo_access pcre:/etc/postfix/helo_checks ❷
  reject_non_fqdn_hostname
  reject_invalid_hostname
  check_sender_mx_access cidr:/etc/postfix/bogus_mx ❸
  check_sender_access hash:/etc/postfix/rhsbl_sender_exceptions ❹
  reject_rhsbl_sender dsn.rfc-ignorant.org ❺
  check_sender_access hash:/etc/postfix/common_spam_senderdomains ❻
  check_sender_access regexp:/etc/postfix/common_spam_senderdomain_keywords
  permit
```

Общая идея заключается в том, чтобы «дешевые» ограничения предшествовали «дорогостоящим»:

❶ Поместите все параметры борьбы со спамом после `permit_mynetworks`, с тем чтобы они относились только к внешним клиентам (т. е. клиентам, не перечисленным в `mynetworks`).

❷ Вы можете без промедлений отвергать любой клиент, который использует имя хоста вашего сервера. Уже не имеет значения, использует ли он полное имя хоста и действительно ли это имя.

❸ С этого параметра начинаются дорогостоящие ограничения. Для `check_sender_mx_access` требуется один или два поиска в DNS. Если у вас работает кэширующий сервер имен, вы можете разрешать запросы к DNS локально.

❹ Эта карта стоит перед параметрами черных списков, т. к. она содержит исключения для пользователей и доменов, которые в противном случае могли бы быть отвергнуты.

❺ Этот параметр требует запроса к удаленной системе (DNS-серверу для `dsn.rfc-ignorant.org`), которая может быть сильно загружена или временно не работать. Параметр дорогостоящий, поэтому он используется ближе к концу списка ограничений.

❻ Последними вводятся два самых дорогостоящих действия. Если они запущены, Postfix должен создать фиктивное сообщение, попытаться его доставить и записать результат. Очень затратно, поэтому используется в последнюю очередь.

Использование классов ограничений

В примере, рассматриваемом в данном разделе, ограничения на отправителей конверта накладываются с двух сторон. Во-первых, мы требуем, чтобы у сообщений, приходящих извне, адрес отправителя *не* относился к нашему домену, и во-вторых, чтобы сообщения от внутренних клиентов *имели* адрес отправителя, относящийся к нашему домену.

Идея заключается в том, чтобы Postfix сначала проверял, относится ли входящее клиентское соединение к вашей сети:

1. Если клиент находится в вашей сети, Postfix отправляет его в класс ограничений. Этот класс содержит проверку адреса отправителя конверта:
   - Если отправитель конверта соответствует шаблону вашего домена, проверка возвращает `OK`. Postfix прекращает оценку ограничений и позволяет клиенту продолжать.
   - Если отправитель конверта не соответствует шаблону вашего домена, то следующий параметр ограничений – это `reject`, так что Postfix отказывает клиенту в обслуживании.
2. Если клиент не относится к вашей сети, Postfix не использует класс ограничений. Вместо этого он переходит к следующему ограничению, в котором проверяется адрес получателя конверта:
   - Если клиент использует имя вашего домена в адресе отправителя конверта, то Postfix отказывает в обслуживании.

- Если клиент не использует имя вашего домена в адресе отправителя конверта, то тест пройден, и Postfix переходит к следующему ограничению.

Для реализации описанного алгоритма создадим файл карты, содержащий список IP-адресов и сетей внутри вашей сети. Назвать файл можно `/etc/postfix/internal_networks`; выглядеть он будет примерно так:

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

Затем создадим другой файл карты `/etc/postfix/our_domain_as_sender`, содержащий шаблон вашего домена и пустое имя отправителя конверта (помните, что ваш сервер должен без вопросов принимать сообщения такого отправителя); это будет список доменов отправителей конверта, которые могут использовать внутренние клиенты. Этот файл карты может быть таким:

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

Теперь создадим файл карты, содержащий домены, которые внешние клиенты не могут использовать для отправителя конверта. В нашем примере файл будет называться `/etc/postfix/not_our_domain_as_sender` и содержать всего одну строку:

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

После создания при помощи команды `postmap` карт на основе этих файлов задайте класс ограничений и необходимые параметры ограничений в файле `main.cf`:

```
smtpd_restriction_classes =
  has_our_domain_as_sender
has_our_domain_as_sender =
  check_sender_access hash:/etc/postfix/our_domain_as_sender
  reject
smtpd_recipient_restrictions =
  check_client_access hash:/etc/postfix/internal_networks
  check_sender_access hash:/etc/postfix/not_our_domain_as_sender
  reject_unauth_destination
  ...
  permit
```

Как обычно, для того чтобы изменения вступили в силу, необходимо перезагрузить конфигурацию Postfix.