Exim + Dovecot (без MySQL)

https://unix.uz/articles/tuning/2646-ustanovka-exim-i-dovecot-bez-mysql.html

Среди системных администраторов и инженеров распространено мнение, что чем система проще, тем она надёжнее. Конечно, не стоит воспринимать это утверждение буквально. Но, как говорится, умному – достаточно. Чем меньше компонентов в системе, чем проще взаимосвязи между компонентами, тем меньше вероятность того, что она сломается и тем выше вероятность того, что её будет просто починить.

Именно поэтому большинство системных администраторов Unix любят использовать текстовые файлы, которые легко редактируются при помощи любого текстового редактора (файлы XML к таковым обычно не относятся). Здесь я опишу настройку почтовой системы на основе Dovecot и Exim без использования базы данных. Вместо этого все данные будут помещаться в текстовых файлах.

1. Dovecot

Начнём с настройки Dovecot, поскольку для настройки Exim необходимо иметь уже настроенный Dovecot.

1.1. Установка Dovecot

Установим пакеты Dovecot, содержащие поддержку серверов POP3 и IMAP, а также пакет, позволяющий использовать данные из MySQL:

```
# apt-get install dovecot-core dovecot-imapd dovecot-pop3d
```

1.2. Подготовка системы

Создадим *группу* и *пользователя* **vmail**, от имени которого будет работать **Dovecot** и дадим этому пользователю доступ к каталогу, в котором будет храниться почта пользователей почтовой системы. На серверах для размещения почтовых ящиков, как и

другой часто меняющейся информации, обычно используется раздел **/var**, который заранее делается достаточно большим. На настраиваемой мной системе больше всего свободного места на разделе **/home**, поэтому я размещу почтовые ящики пользователей на нём:

```
# groupadd -g 120 -r vmail
# useradd -g 120 -r -u 120 vmail
# mkdir /home/vmail
# chown vmail:vmail /home/vmail
# chmod u=rwx,g=rx,o= /home/vmail
```

1.3. Базовая настройка Dovecot

Я буду использовать *защищённые* версии протоколов **IMAP** и **POP3**, поэтому настрою в файле **/etc/dovecot/conf.d/10-auth.conf** механизмы **PLAIN** и **LOGIN**, чтобы хранить пароли в базе данных в хэшированном виде:

```
disable_plaintext_auth = no
auth_default_realm = domain.tld
auth_mechanisms = plain login
!include auth-passwdfile.conf.ext
```

Настроим использование учётных данных из файла, подобного **/etc/passwd**, прописав в файле **/etc/dovecot/conf.d/auth-passwdfile.conf.ext** следующие секции:

```
passdb {
 driver = passwd-file
 args = scheme=CRYPT username_format=%u /etc/dovecot/passwd
}

userdb {
 driver = passwd-file
 args = username_format=%u /etc/dovecot/passwd

# Поля по умолчанию, которые могут быть заменены значениями из файла passwd
     default_fields     =     uid=vmail     gid=vmail
userdb_home=/home/vmail/%Ld/%Ln
```

```
userdb_location=maildir:/home/vmail/%Ld/%Ln
userdb_quota_rule=*:storage=1G
# Поля, значения которых заменяют значения из файла passwd
#override_fields = home=/home/vmail/%Ld/%Ln
}
```

Создадим в каталоге **/etc/dovecot** файл **passwd** и проставим права доступа:

```
# cd /etc/dovecot
# touch passwd
# chown root:dovecot passwd
# chmod u=rw,g=r,o= passwd
```

В файле **/etc/dovecot/passwd** могут быть следующие поля:

```
user:{plain}password:uid:gid:gecos:home:shell:extra_fields
```

Назначение полей:

user - почтовый ящик (в данном случае - вместе с доменом),
password - пароль (можно явным образом указывать алгоритм хэширования пароля),
uid - системный идентификатор владельца файлов почты,
gid - системный идентификатор группы владельца файлов почты,
gecos - справочная информация о почтовом ящике (игнорируется),
home - путь к каталогу почты,
shell - интерпретатор (игнорируется),
extra_fields - дополнительные настройки (квота, например).
Любое из полей может быть не определено в файле, если в настройках
Dovecot указаны значения этих полей по умолчанию. При указании дополнительных полей,
используемых в секции **userdb** необходимо перед именем поля указывать префикс «**userdb_**», как в примере выше, в настройках default_fields. Имеется возможность зафиксировать часть настроек почтового ящика при помощи настройки override_fields, так что значения из файла будут игнорироваться.

Подробнее о формате файла и других настройках можно прочитать на официальной wiki-странице Dovecot: Passwd-file

Изменим форматирование отметок времени, вписав в файл

/etc/dovecot/conf.d/10-logging.conf следующую настройку:

```
log_timestamp = "%Y-%m-%d %H:%M:%S "
```

На время отладки также можно включить другие опции из этого файла:

```
auth_verbose = yes
auth_verbose_passwords = yes
auth_debug = yes
mail_debug = yes
```

В файле /etc/dovecot/conf.d/10-mail.conf настроим путь к почтовым ящикам и пользователя, от имени которого dovecot будет работать с ящиками:

```
mail_location = maildir:/home/vmail/%Ld/%Ln
mail_uid = vmail
mail_gid = vmail
first_valid_uid = 120
last_valid_uid = 120
first_valid_gid = 120
last_valid_gid = 120
```

Сейчас настроим сервис, при помощи которого *Exim* будет *проверять учётные данные почтовых клиентов*. Для этого отредактируем файл **/etc/dovecot/conf.d/10-master.conf** и впишем в него настройки сервиса:

```
service auth {
 unix_listener auth-client {
 mode = 0660
 user = Debian-exim
 #group =
 }
}
```

Зададим в файле **/etc/dovecot/conf.d/15-lda.conf** адрес, с которого Dovecot будет отправлять сообщения об ошибках:

```
postmaster_address = postmaster@domain.tld
```

Осталось отредактировать файл **/etc/dovecot/dovecot.conf**,

указав в нём адрес, на котором сервер будет ожидать подключений:

```
!include_try /usr/share/dovecot/protocols.d/*.protocol
listen = *
!include conf.d/*.conf
!include_try local.conf
```

Начальная настройка сервера окончена. Осталось перезапустить Dovecot, чтобы настройки вступили в силу:

```
# /etc/init.d/dovecot restart
```

1.4. Настройка плагина acl

Плагин acl позволяет пользователям предоставлять друг другу доступ к папкам в своих почтовых ящиках. Это может быть полезно для корпоративных пользователей. Например, для директора и его заместителя. Или для директора и его секретаря. Или для сотрудников из одного отдела, которые подменяют друг друга на время обеда или отпуска. Эта возможность, естественно, доступна только при использовании протокола IMAP.

В файле **/etc/dovecot/conf.d/10-mail.conf** включаем использование плагина:

```
mail_plugins = acl
```

В файле **/etc/dovecot/conf.d/20-imap.conf** включаем использование плагина в IMAP-сервере:

```
protocol imap {
 mail_plugins = $mail_plugins imap_acl
}
```

В файле **/etc/dovecot/conf.d/10-mail.conf** прописываем следующие настройки:

```
namespace inbox {
 type = private
 separator = /
```

```
 prefix =
 inbox = yes
}

namespace {
 type = shared
 separator = /
 prefix = shared/%%u/
 location = maildir:%%h:INDEX=%h/shared/%%u
 subscriptions = yes
 list = children
}
```

Эти настройки описывают два пространства имён: в первом хранится личная почта пользователя, а во втором будут отображаться каталоги других пользователей, к которым этот пользователь имеет доступ.

Поясню смысл настроек **location** для пространства имён общих каталогов:

maildir:%%h - означает место расположения чужого почтового ящика в формате Maildir,
%%h - полный путь к Maildir-каталогу чужого ящика,
INDEX=%h/shared/%%u - задаёт каталог, в который как бы монтируются каталоги чужой почты, к которым её владелец дал нам доступ,
%h - путь к Maildir-каталогу нашего ящика,
%%u - имя другого пользователя в виде box@domain.tld.

В файл **/etc/dovecot/conf.d/90-acl.conf** прописываем настройки плагина:

```
plugin {
 acl = vfile
 acl_shared_dict = file:/home/vmail/%Ld/shared-mailboxes.db
}
```

Значение **vfile** предписывает создавать внутри почтового ящика файл dovecot-acl, в котором и будут прописываться права доступа к нему со стороны других пользователей.

Значение **acl_shared_dict** указывает путь к файлу словаря, который позволит пользователям узнавать, к каким каталогам в чужих почтовых ящиках у них имеется доступ. В данном случае для каждого домена будет создан отдельный файл словаря, расположенный в каталоге домена, на одном уровне с ящиками.

Заодно опишем в файле **/etc/dovecot/conf.d/15-mailboxes.conf** назначение различных каталогов внутри пространства имён, в котором хранится личная почта пользователя:

```
namespace inbox {
 mailbox Drafts {
 special_use = Drafts
 }
 mailbox Junk {
 special_use = Junk
 }
 mailbox Trash {
 special_use = Trash
 }
 mailbox Sent {
 special_use = Sent
 }
}
```

Назначение каталогов:

Drafts - каталог черновиков,
Junk - каталог для спама,
Trash - каталог для удалённых писем,
Sent - каталог для отправленных писем.

Современные почтовые программы смогут прямо по протоколу IMAP узнать назначение каждого из специальных каталогов, вне зависимости от их названия.
Это бывает полезно, если каталог имеет нестандартное название или название
на языке пользователя ящика, например “Входящие” или “Спам”.

Чтобы настройки плагина acl вступили в силу, нужно перезапустить Dovecot:

```
# /etc/init.d/dovecot restart
```

1.5. Настройка плагина quota

Плагин quota позволяет назначить для почтового ящика ограничения на объём хранящихся в нём писем или даже на их общее количество. На мой взгляд, ограничение на общее количество писем имеет довольно мало смысла. Единственная польза, которая мне приходит на ум – это возможность защититься от исчерпания inode’ов в файловой системе, если кто-то намеренно решит отправить огромное количество мелких писем в ящики пользователей, с целью нарушить работу почтовой системы.

Мы настроим плагин так, чтобы он использовал значения квот, указанные в интерфейсе Postfixadmin. Эти квоты ограничивают только максимальный объём писем в ящике.

Включим использование плагина в файле **/etc/dovecot/conf.d/10-mail.conf**:

```
mail_plugins = acl quota
```

Жирным шрифтом показан добавленный текст, а курсивом – текст, добавленный нами при включении плагина acl. Если вы не включали плагин acl, то вписывать этот текст не нужно.

В файл **/etc/dovecot/conf.d/15-lda.conf** впишем, что в случае превышения квоты Dovecot должен сообщать о временной ошибке, но не отклонять письмо окончательно. Почтовый сервер отправителя (или наш MTA) будет периодически предпринимать повторные попытки в надежде на то, что адресат почистит свой ящик от ненужных писем.

```
quota_full_tempfail = yes
```

В файл **/etc/dovecot/conf.d/20-imap.conf** добавим поддержку квот в IMAP-сервере:

```
protocol imap {
 mail_plugins = $mail_plugins imap_acl imap_quota
```

```
 }
```

Этот плагин позволит почтовым клиентам, работающим по протоколу IMAP, узнавать квоту почтового ящика и её текущее использование.

Укажем в файле **/etc/dovecot/conf.d/90-quota.conf**, что значения квот берутся из словаря и зададим пустое правило по умолчанию:

```
plugin {
 quota = dict:user::file:%h/dovecot-quota
 quota_rule = *:
 }
```

Осталось перезапустить Dovecot, чтобы настроенный плагин начал работать:

```
# /etc/init.d/dovecot restart
```

В каталоге каждого почтового ящика будет создаваться файл dovecot-data, внутри которого будет вестись учёт текущего количества сообщений в ящике и их объёма. Чтобы принудительно пересчитать квоты всех почтовых ящиков, можно воспользоваться следующей командой:

```
# doveadm quota recalc -A
```

1.6. Настройка плагина expire

Этот плагин не поддерживает работу со словарями, хранящимися не в базах данных, поэтому здесь его настройка не описывается.

Для периодической очистки почтовых ящиков от устаревших сообщений можно добавить в планировщик задач выполнение, например, такой команды:

```
doveadm expunge -A mailbox Spam savedbefore 2w
```

Эта команда найдёт в почтовых ящиках пользователей каталоги Spam и удалит из них те сообщения, которые были сохранены в ящик более двух недель назад. Подобным образом можно очищать и другие каталоги, например – Trash.

1.7. Настройка SSL

Настройка SSL будет подробнее рассмотрена в одной из следующих заметок. Там будет описана настройка отдельных сертификатов SSL для разных доменов, а также будет освещён вопрос подготовки самих сертификатов – самоподписанных или подписанных центром сертификации.

Если у вас имеются готовый подписанный сертификат, можно включить поддержку SSL в файле /etc/dovecot/conf.d/10-ssl.conf и указать в нём пути к файлам сертификата:
ssl = yes
ssl_cert = </etc/ssl/mail_public.pem
ssl_key = </etc/ssl/mail_private.pem

После настройки сертификатов нужно перезапустить Dovecot, чтобы изменения вступили в силу:
/etc/init.d/dovecot restart
1.8. Настройка плагина sieve

Sieve – это скрипты фильтрации почты, которые выполняются агентом локальной доставки (LDA) в момент получения письма от почтового сервера (MTA). Скрипты позволяют раскладывать письма в разные папки, ориентируясь на их содержимое – тему письма, получателей, отправителей и т.п. Можно удалить письмо, переслать его на другой ящик или отправить уведомление отправителю, причём использовать можно любое поле заголовка или содержимое тела письма.

Главное преимущество Sieve заключается в том, что пользователю не нужно настраивать правила фильтрации в каждом из используемых им почтовых клиентов – правила едины для всех почтовых клиентов сразу. Кроме того, фильтрация происходит вообще без участия клиента. Клиент, подключившись к почтовому ящику, имеет возможность работать уже с отсортированной почтой. Кроме того, отправка уведомлений о получении или пересылка писем на другой ящик вообще может происходить без участия почтового клиента.

Конечно, в наши времена больших почтовых сервисов типа Gmail или Яндекс-почты, этим никого не удивишь. Но тут плюс заключается в том, что перед нами не стоит дилемма “удобство” – “безопасность”. Мы можем хранить почту у себя, не делясь ею с посторонними компаниями, имея над ней полный контроль, и в то же время можем пользоваться удобствами, характерными для больших почтовых сервисов.

Установим пакет с плагином Sieve:
```
# apt-get install dovecot-sieve
```

Включим использование плагина в файле /etc/dovecot/conf.d/15-lda.conf:
```
protocol lda {
mail_plugins = $mail_plugins sieve
}
```

Укажем настройки плагина в файле /etc/dovecot/conf.d/90-sieve.conf:
```
plugin {
sieve = /home/vmail/%Ld/%n/active.sieve # Расположение активного скрипта
sieve_dir = /home/vmail/%Ld/%n/sieve # Каталог для скриптов
sieve_max_script_size = 1M # Максимальный размер одного скрипта
sieve_quota_max_scripts = 50 # Максимальное количество скриптов
sieve_quota_max_storage = 1M # Максимальный общий объём скриптов
}
```

Каждый пользователь может обладать собственным набором Sieve-скриптов, из которых в любой момент времени активным может быть только один. Каталог для скриптов указывается в настройке sieve_dir, а в настройке sieve указывается имя символической ссылки, которая будет указывать на активный скрипт.

После настройки плагина нужно перезапустить Dovecot, чтобы

изменения вступили в силу:
/etc/init.d/dovecot restart

Подробнее о скриптах Sieve можно почитать на Википедии, в статье Sieve.

1.9. Настройка сервиса managesieve

Плагин sieve не был бы столь полезным, если бы Sieve-скриптами нельзя было бы управлять прямо из почтового клиента. Именно эту функцию и реализует сервис ManageSieve. Он ожидает подключений клиентов на отдельном TCP-порту 4190. Для управления скриптами клиент использует учётные данные своего почтового ящика.

Для включения сервиса достаточно лишь установить дополнительный пакет:
apt-get install dovecot-managesieve

В следующих заметках фильтрация писем при помощи Sieve будет рассмотрена подробнее — я покажу, как им пользоваться в почтовых клиентах.

1.10. Результирующий файл конфигурации

Поскольку настроек очень много, проверить их можно при помощи следующей команды:

$ doveconf -n

Опция **n** предписывает показывать только те настройки, которые отличаются от настроек по умолчанию. У меня со всеми плагинами, настройка которых была тут описана, команда выдаёт следующий результат:

```
# 2.1.7: /etc/dovecot/dovecot.conf
# OS: Linux 3.2.0-4-amd64 x86_64 Debian 7.5 ext4
 auth_default_realm = domain.tld
 auth_mechanisms = plain login
 disable_plaintext_auth = no
 first_valid_gid = 120
```

```
first_valid_uid = 120
last_valid_gid = 120
last_valid_uid = 120
listen = *
log_timestamp = "%Y-%m-%d %H:%M:%S "
mail_gid = vmail
mail_home = /home/vmail/%Ld/%Ln
mail_location = maildir:/home/vmail/%Ld/%Ln
mail_plugins = quota acl
mail_uid = vmail
managesieve_notify_capability = mailto
managesieve_sieve_capability = fileinto reject envelope encoded-character vacation subaddress comparator-i;ascii-numeric relational regex imap4flags copy include variables body enotify environment mailbox date ihave
namespace {
list = children
location = maildir:%%h:INDEX=%h/shared/%%u
prefix = shared/%%u/
separator = /
subscriptions = yes
type = shared
}
namespace inbox {
inbox = yes
location =
mailbox Drafts {
special_use = Drafts
}
mailbox Junk {
special_use = Junk
}
mailbox Sent {
special_use = Sent
}
mailbox Trash {
special_use = Trash
}
prefix =
separator = /
type = private
```

```
 }
 passdb {
 args = scheme=CRYPT username_format=%u /etc/dovecot/passwd
 driver = passwd-file
 }
 plugin {
 acl = vfile
 acl_shared_dict = file:/home/vmail/%Ld/shared-mailboxes.db
 quota = dict:user::file:%h/dovecot-quota
 quota_rule = *:
 sieve = /home/vmail/%Ld/%Ln/active.sieve
 sieve_dir = /home/vmail/%Ld/%Ln/sieve
 sieve_max_script_size = 1M
 sieve_quota_max_scripts = 50
 sieve_quota_max_storage = 1M
 }
 postmaster_address = postmaster@domain.tld
 protocols = " imap sieve pop3"
 service auth {
 unix_listener auth-client {
 group = Debian-exim
 mode = 0660
 }
 }
 ssl_cert = </etc/ssl/mail.domain.tld.public.pem
 ssl_key = </etc/ssl/mail.domain.tld.private.pem
 userdb {
 args = username_format=%u /etc/dovecot/passwd
     default_fields     =     uid=vmail     gid=vmail
userdb_home=/home/vmail/%Ld/%Ln
userdb_location=maildir:/home/vmail/%Ld/%Ln
userdb_quota_rule=*:storage=1G
 driver = passwd-file
 }
 protocol lda {
 mail_plugins = quota acl sieve
 }
 protocol imap {
 mail_plugins = quota acl imap_quota imap_acl
 }
```

2. Настройка Exim

Настройка Exim, как уже было сказано, зависит от настроенного Dovecot. Exim использует сервис Dovecot для SMTP-аутентификации и список почтовых ящиков из файла /etc/dovecot/passwd.

2.1. Установка

Установим SMTP-сервер Exim:

```
# apt-get install exim4-daemon-heavy
```

Создаём файл конфигурации **/etc/exim4/exim4.conf** со следующим начальным содержимым:

```
# Имя нашей почтовой системы
 primary_hostname = mail.domain.tld

# Список доменов нашей почтовой системы
 [code]domainlist local_domains = /etc/exim4/local_domains

# Список доменов, для которых наша почтовая система является резервной
 [code]domainlist relay_domains = /etc/exim4/relay_domains

# Список узлов, почту от которых будем принимать без проверок
 hostlist relay_from_hosts =

# Правила для проверок
 acl_not_smtp = acl_check_not_smtp
 acl_smtp_rcpt = acl_check_rcpt
 acl_smtp_data = acl_check_data

# Сокет-файл антивируса ClamAV
 av_scanner = clamd:/var/run/clamav/clamd.ctl
 # Сокет-файл SpamAssassin
 # spamd_address =

# Отключаем IPv6, слушаем порты 25 и 587
 disable_ipv6
 daemon_smtp_ports = 25 : 587
```

```
# Дописываем домены отправителя и получателя, если они не указаны
 qualify_domain = domain.tld
 qualify_recipient = domain.tld

# Exim никогда не должен запускать процессы от имени пользователя root
 never_users = root

# Проверять прямую и обратную записи узла отправителя по DNS
 host_lookup = *

# Отключаем проверку пользователей узла отправителя по протоколу ident
 rfc1413_hosts = *
 rfc1413_query_timeout = 0s

# Только эти узлы могут не указывать домен отправителя или получателя
 sender_unqualified_hosts = +relay_from_hosts
 recipient_unqualified_hosts = +relay_from_hosts

# Лимит размера сообщения, 30 мегабайт
 message_size_limit = 30M

# Запрещаем использовать знак % для явной маршрутизации почты
 percent_hack_domains =

# Настройки обработки ошибок доставки, используются значения по умолчанию
 ignore_bounce_errors_after = 2d
 timeout_frozen_after = 7d

begin acl

# Проверки для локальных отправителей
 acl_check_not_smtp:
 accept

# Проверки на этапе RCPT
 acl_check_rcpt:
```

```
 accept hosts = :

# Отклоняем неправильные адреса почтовых ящиков
 deny message = Restricted characters in address
 domains = +local_domains
 local_parts = ^[.] : ^.*[@%!/|]

# Отклоняем неправильные адреса почтовых ящиков
 deny message = Restricted characters in address
 domains = !+local_domains
 local_parts = ^[./|] : ^.*[@%!] : ^.*/\.\./

# В локальные ящики postmaster и abuse принимает почту всегда
 accept local_parts = postmaster : abuse
 domains = +local_domains

# Проверяем существование домена отправителя
 require verify = sender

# Принимаем почту от доверенных узлов, попутно исправляя
заголовки письма
 accept hosts = +relay_from_hosts
 control = submission

# Принимаем почту от аутентифицированных узлов, попутно
исправляя заголовки письма
 accept authenticated = *
 control = submission/domain=

# Для не доверенных и не аутентифицированных требуется, чтобы
получатель был в домене,
# ящик которого находится у нас или для которого мы являемся
резервным почтовым
# сервером
 require message = Relay not permitted
 domains = +local_domains : +relay_domains

# Если домен правильный, то проверяем получателя
 require verify = recipient

accept
```

```
begin routers

# Поиск транспорта для удалённых получателей
 dnslookup:
 driver = dnslookup
 domains = ! +local_domains
 transport = remote_smtp
 ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8
 no_more

# Пересылки для локальных получателей из файла /etc/aliases
 system_aliases:
 driver = redirect
 allow_fail
 allow_defer
 domains = domain.tld
 data = ${lookup{$local_part}lsearch{/etc/aliases}}

# Пересылки для получателей в разных доменах
 aliases:
 driver = redirect
 allow_fail
 allow_defer
                            data                            =
${lookup{$local_part@$domain}lsearch{/etc/exim4/aliases}}

# Получение почты на локальный ящик
 mailbox:
 driver = accept
                        condition                           =
${lookup{$local_part@$domain}lsearch{/etc/dovecot/passwd}{yes}
{no}}
 user = dovecot
 transport = dovecot_virtual_delivery
 cannot_route_message = Unknown user

begin transports

# Транспорт для удалённых получателей
 remote_smtp:
 driver = smtp
```

```
# Транспорт для локальных получателей из Dovecot
 dovecot_virtual_delivery:
 driver = pipe
 command = /usr/lib/dovecot/dovecot-lda -d $local_part@$domain
-f $sender_address
 message_prefix =
 message_suffix =
 delivery_date_add
 envelope_to_add
 return_path_add
 log_output
 user = vmail
 temp_errors = 64 : 69 : 70: 71 : 72 : 73 : 74 : 75 : 78

begin retry

* * F,2h,15m; G,16h,1h,1.5; F,4d,6h

begin rewrite

begin authenticators

# Использование LOGIN-аутентификации из Dovecot
 dovecot_login:
 driver = dovecot
 public_name = LOGIN
 server_socket = /var/run/dovecot/auth-client
 server_set_id = $auth1

# Использование PLAIN-аутентификации из Dovecot
 dovecot_plain:
 driver = dovecot
 public_name = PLAIN
 server_socket = /var/run/dovecot/auth-client
 server_set_id = $auth1
```

Сразу поменяем права доступа к файлу конфигурации:

```
# chmod u=rw,g=r,o= /ect/exim4/exim4.conf
# chown root:Debian-exim /etc/exim4/exim4.conf
```

В этом случае можно дать остальным пользователям доступ на

чтение, т.к. никакой особо секретной информации в файле конфигурации нет. С другой стороны – нужды давать такой доступ тоже нет.

Чтобы Exim мог читать файл **/etc/dovecot/passwd**, включим пользователя **Debian-exim** в группу **dovecot**:

```
# usermod -aG dovecot Debian-exim
```

Осталось запустить Exim, чтобы он начал работать в минимальной конфигурации:

```
# /etc/init.d/exim4 start
```