Некоторое время назад я рассказывал, как настроить почтовый сервер на базе готовой сборки iredmail. Я подробно разобрал все актуальные вопросы почтового сервера на linux на базе postfix. Сейчас хочу настроить похожий функционал, но с нуля, за основу взяв postfix + dovecot. Я расскажу про установку и настройку postfix на centos 7, причем только тех модулей и дополнений, которые сам считаю нужными и полезными на почтовом сервере.

Содержание:

Данная статья является частью единого цикла статьей про сервер Centos.

> Вышла обновленная версия этой статьи для системы Centos 8 — Настройка postfix + dovecot + mysql база + postfixadmin + roundcube + dkim

Введение

Я буду настраивать почтовый сервер на ОС linux, а точнее на **CentOS 7**. За основу будет взят **postfix**, который присутствует в этой системе из коробки. Инструкция получится универсальной, можно использовать и для других дистрибутивов. Все основные конфиги легко переносятся на разные системы, требуя минимальной правки, в основном путей.

Я напишу статью на самом что ни на есть реальном примере, без какой-либо правки доменов, ip и прочего, чтобы не ошибиться и показать максимально возможный реальный пример. У меня есть технический домен zeroxzed.ru. Я буду использовать его в своей работе. Почтовый сервер будет иметь имя mail.zeroxzed.ru. Всю теорию по подготовке dns к установке и настройке почтового сервера я рассказывал в предыдущей статье о почтовом сервере. Не хочу здесь повторяться. Уточню только список действий, которые вам нужно проделать с ДНС:

1. Создаем A запись в DNS — mail.zeroxzed.ru.
2. Добавляем или редактируем MX запись, указывая в качестве почтового сервера mail.zeroxzed.ru.
3. Просим провайдера прописать PTR для внешнего ip адреса, который будет использовать почтовый сервер. В качестве ptr записи просим установить имя нашего сервера — mail.zeroxzed.ru.

Я предпочитаю в качестве dns хостинга использовать сервера яндекса, даже если не прикрепляю его почту к домену. На картинке показан минимально необходимый набор записей, кроме PTR. Этими записями управляете не вы, а провайдер, который вам выдает ip. Пока с днс все. Позже мы вернемся к этому вопросу, когда будем добавлять dkim и spf записи. Но обо всем по порядку.

Подготовим систему centos к установке и настройке почтового сервера postfix. Если у вас еще нет готовой системы, то рекомендую воспользоваться моими статьями по установке и настройке centos. Отдельно потратьте время на настройку iptables. Я не буду касаться этого вопроса в данной статье, чтобы не раздувать ее второстепенными вещами. Удобнее, когда все по отдельности рассказано и описано с должной глубиной. Сваливать все в одну кучу не хочется.

По вступлению вроде все, основное рассказал. Приступим к настройке нашего почтового сервера.

> Сразу хочу сделать предупреждение. Настройка почтового сервера достаточно трудоемкий процесс, требует определенных навыков, знаний и понимания принципов работы используемых средств. Я не ставлю для себя цель расписать максимально подробно так, чтобы было понятно даже неподготовленному администратору linux. Вы должны быть так или иначе подготовлены, либо запаситесь терпением и разбирайтесь внимательно сами в нюансах. Эта статья на полный копипаст не подходит, что-то остается за кадром для самостоятельного выполнения. Иначе нельзя, получится очень большой и громоздкий материал.

Установка postfixadmin

Начнем с установки и настройки панели управления почтовым сервером postfix — **postfixadmin**. Без него начинать что-то делать неудобно, так как управлять пользователями, ящиками, алиасами будет нечем. По своей сути postfixadmin — набор php скриптов для управления записями в mysql базе данных, которую использует сервер postfix во время своей работы. Соответственно, для работы postfixadmin нам нужен web сервер. Подробно о настройке web сервера на centos читайте отдельно. Сейчас же мы быстро установим все необходимое. Привожу только команды, без комментариев. Все подробности по приведенной выше ссылке.

```
# yum install httpd php phpmyadmin mariadb mariadb-server php-imap
```

Этих пакетов со всеми зависимостями будет достаточно для установки всех необходимых компонентов веб сервера. Я специально ставлю phpmyadmin, с ним удобно работать с базой. В нашем случае все пользователи будут храниться в mysql, иногда может понадобиться туда заглянуть. Подробнее с установкой и настройкой phpmyadmin можете ознакомиться отдельно.

Запускаем httpd и mariadb и добавляем их в автозагрузку.

```
# systemctl start httpd
# systemctl enable httpd
# systemctl start mariadb
# systemctl enable mariadb
```

Задаем пароль root для mysql.

```
# /usr/bin/mysql_secure_installation
```

Проверяем работу web сервера. Заходим по ip адресу сервера — *http://188.35.19.125/*, а также проверяем работу phpmyadmin — *http://188.35.19.125/phpmyadmin/*. Его нужно настроить, об этом рассказано в статье, которую я привел чуть выше. По-умолчанию в phpmyadmin доступ закрыт. Если все сделали правильно, то увидите примерно следующее.

Сразу создадим тут пользователя postfix и одноименную базу данных. Запомните учетные данные, они нам далее понадобятся.

Веб сервер готов, продолжаем настройку почтового сервера. Скачиваем последнюю версию postfixadmin.

```
# cd /usr/src
# wget https://downloads.sourceforge.net/project/postfixadmin/postfixadmin/postfixadmin-3.0.2/postfixadmin-3.0.2.tar.gz
```

> Скорее всего во время вашей установки версия postfixadmin изменится и ссылка может быть неактуальной. Но даже если она будет актуальна, возможно выйдет более новая версия. Проверьте ее по ссылке https://sourceforge.net/projects/postfixadmin/ и скачайте самую свежую версию.

Распаковываем архив и копируем в директорию веб сервера.

```
# tar -xvzf postfixadmin-*
# mv /usr/src/postfixadmin-3.0.2 /var/www/html/postfixadmin
```

Назначаем владельцем пользователя веб сервера:

```
# chown -R apache. /var/www/html/postfixadmin/
```

Дальше редактируем конфигурационный файл postfixadmin.

```
# mcedit /var/www/html/postfixadmin/config.inc.php
```

Приводим параметры к следующему виду:

```
$CONF['configured'] = true;
$CONF['default_language'] = 'ru';
$CONF['database_type'] = 'mysqli';
$CONF['database_host'] = 'localhost';
$CONF['database_user'] = 'postfix';
$CONF['database_password'] = '12345678';
$CONF['database_name'] = 'postfix';
$CONF['admin_email'] = 'root@zeroxzed.ru';
$CONF['encrypt'] = 'md5crypt';
$CONF['default_aliases'] = array (
 'abuse' => 'root',
 'hostmaster' => 'root',
 'postmaster' => 'root',
 'webmaster' => 'root'
);
$CONF['domain_path'] = 'YES';
$CONF['domain_in_mailbox'] = 'YES';
```

Обращаю внимание на выделенный параметр. Он указывает на то, в каком виде хранить пароли пользователей в базе данных. Конечно, хранить обычным текстом без шифрования это дурной тон и может быть опасно. Я указал хранение в шифрованном виде. Но если мы говорим о небольшой компании без публичного доступа к серверу, можно использовать нешифрованные пароли. Для этого указываем значение параметра **cleartext**. Я сам так часто делаю просто из соображений удобства. Объясню, в чем удобство.

К примеру, у пользователя несколько устройств подключены к почте и он забыл свой пароль. Админ при создании почему-то тоже его никуда не записал, или забыл, или потерял. Вам придется сбросить пароль и перенастроить все устройства. Если же у вас пароль хранится в открытом виде, вы просто смотрите в базу и говорите пользователю пароль. Пароли в открытом виде удобно просто выгрузить дампом из базы, если понадобится кому-то все учетки

передать. Но тут как посмотреть :) С одной стороны плюс, с другой минус — кто-то очень просто может спереть все ваши пароли. В общем, тут от ситуации зависит, решайте сами, как вам удобнее хранить пароли.

У меня распространены ситуации, когда я удаленно администрирую сервера, а на месте эникеи работают с пользователями. Чаще всего это не очень аккуратные и ответственные люди, иначе они бы работали с серверами :) Они часто забывают записать пароль, путают что-то и т.д. В итоге, когда один человек увольняется и приходит другой, оказывается, что найти пароли на некоторые ящики просто невозможно. Тут очень выручает возможность посмотреть пароль в базе. Я новому админу либо пароль говорю, либо весь дамп сразу отдаю, пусть работает.

Если вы сами работаете с сервером и все аккуратно ведете, записываете, например, в keepass все пароли от почтовых ящиков, то смело шифруйте все пароли, так будет спокойнее.

Последние 2 параметра **domain_path** и **domain_in_mailbox** указывайте по своему усмотрению. В файле конфигурации в комментариях расписано, за что они отвечают и в чем отличие. Мне кажется, удобно хранить директории именно в таком виде, как я указал. Получится следующий путь до ящика, если у вас архив почты будет жить, к примеру, в директории */mnt/mail* — /mnt/mail/zeroxzed.ru/root@zeroxzed.ru.

С параметрами разобрались. Сохраняем конфиг. Идем по адресу *http://188.35.19.125/postfixadmin/setup.php* и начинаем установку postfixadmin. Первым делом идет проверка всех необходимых для установки и работы компонентов. Для продолжения установки у вас должна быть такая картинка.

Если чего-то не хватает, разбирайтесь по месту. Если делаете по моей инструкции, то все должно быть в порядке. Указывайте пароль установки и продолжайте. Вы должны получить строку с хэшем этого пароля.

Добавляем полученную строку в файл конфигурации postfixadmin.

```
# mcedit /var/www/html/postfixadmin/config.inc.php
```

```
$CONF['setup_password'] = '67e46bdcc7aeb431f7af9a6d02f43352:30672e5a9deacaf505d32807b967caf9fd0c32ef';
```

Используя этот пароль, можно создать учетную запись администратора панели управления. Делаем это, учитывая, что пароль должен содержать не менее

двух цифр. Если все сделали правильно, то увидите сообщение.

Переходим по ссылке и авторизуемся с помощью учетной записи администратора, которую только что сделали. Вы должны увидеть основную страницу интерфейса postfixadmin.

Теперь нам нужно добавить домен в панель управления. Идем в раздел Список доменов -> Новый домен и добавляем свой домен.

При создании домена были добавлены стандартные алиасы, получателя для которых мы указали еще в конфиге — ящик root@zeroxzed.ru. Создание таких алиасов требование стандартов, но по факту, кроме спама, вы скорее всего ничего не будете получать по этим адресам. Так что их создание оставляйте на свое усмотрения. Я обычно их не делаю, так как ящик для этих алиасов все равно не читаю.

Далее создадим почтовый ящик администратора — root@zeroxzed.ru. Для этого идем в раздел Обзор -> Создать ящик и заполняем поля.

Непосредственно ящик на диске создан не будет, так как у нас еще не настроена почтовая система, но запись в базе данных появится. Это можно проверить через phpmyadmin.

Как мы видим, пароль указан в зашифрованном виде. На этом установку и настройку postfixadmin завершаем. Интерфейс для управления почтовым сервером мы подготовили. Теперь можно заняться непосредственно настройкой postfix.

Настройка postfix

Сердце нашего почтового сервера на linux — postfix. В дистрибутиве centos он уже установлен, можно сразу переходить к настройке. Рисуем следующий конфиг.

```
# mcedit /etc/postfix/main.cf
```

```
soft_bounce = no
queue_directory = /var/spool/postfix
command_directory = /usr/sbin
daemon_directory = /usr/libexec/postfix
data_directory = /var/lib/postfix
mail_owner = postfix

myhostname = mail.zeroxzed.ru
mydomain = zeroxzed.ru
myorigin = $myhostname

inet_interfaces = all
inet_protocols = ipv4

mydestination = localhost.$mydomain, localhost
unknown_local_recipient_reject_code = 550
mynetworks = 127.0.0.0/8

alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

smtpd_banner = $myhostname ESMTP $mail_name

debug_peer_level = 2
```

```
# Строки с PATH и ddd должны быть с отступом в виде табуляции от начала строки
debugger_command =
         PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
         ddd $daemon_directory/$process_name $process_id & sleep 5

sendmail_path = /usr/sbin/sendmail.postfix
newaliases_path = /usr/bin/newaliases.postfix
mailq_path = /usr/bin/mailq.postfix
setgid_group = postdrop
html_directory = no
manpage_directory = /usr/share/man
sample_directory = /usr/share/doc/postfix-2.10.1/samples
readme_directory = /usr/share/doc/postfix-2.10.1/README_FILES

relay_domains = mysql:/etc/postfix/mysql/relay_domains.cf
virtual_alias_maps = mysql:/etc/postfix/mysql/virtual_alias_maps.cf,
 mysql:/etc/postfix/mysql/virtual_alias_domain_maps.cf
virtual_mailbox_domains = mysql:/etc/postfix/mysql/virtual_mailbox_domains.cf
virtual_mailbox_maps = mysql:/etc/postfix/mysql/virtual_mailbox_maps.cf

smtpd_discard_ehlo_keywords = etrn, silent-discard
smtpd_forbidden_commands = CONNECT GET POST
broken_sasl_auth_clients = yes
smtpd_delay_reject = yes
smtpd_helo_required = yes
smtp_always_send_ehlo = yes
disable_vrfy_command = yes

smtpd_helo_restrictions = permit_mynetworks,
 permit_sasl_authenticated,
 reject_non_fqdn_helo_hostname,
 reject_invalid_helo_hostname
```

```
smtpd_data_restrictions = permit_mynetworks,
 permit_sasl_authenticated,
 reject_unauth_pipelining,
 reject_multi_recipient_bounce,

smtpd_sender_restrictions = permit_mynetworks,
 permit_sasl_authenticated,
 reject_non_fqdn_sender,
 reject_unknown_sender_domain

smtpd_recipient_restrictions = reject_non_fqdn_recipient,
 reject_unknown_recipient_domain,
 reject_multi_recipient_bounce,
 permit_mynetworks,
 permit_sasl_authenticated,
 reject_unauth_destination,

smtp_tls_security_level = may
smtpd_tls_security_level = may
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtp_tls_session_cache_database = btree:$data_directory/smtp_tls_session_cache
smtpd_tls_key_file = /etc/postfix/certs/key.pem
smtpd_tls_cert_file = /etc/postfix/certs/cert.pem
tls_random_source = dev:/dev/urandom

# Ограничение максимального размера письма в байтах
message_size_limit = 20000000
smtpd_soft_error_limit = 10
smtpd_hard_error_limit = 15
smtpd_error_sleep_time = 20
anvil_rate_time_unit = 60s
```

```
smtpd_client_connection_count_limit = 20
smtpd_client_connection_rate_limit = 30
smtpd_client_message_rate_limit = 30
smtpd_client_event_limit_exceptions = 127.0.0.0/8
smtpd_client_connection_limit_exceptions = 127.0.0.0/8

maximal_queue_lifetime = 1d
bounce_queue_lifetime = 1d

smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/dovecot-auth

# Директория для хранения почты
virtual_mailbox_base = /mnt/mail
virtual_minimum_uid = 1000
virtual_uid_maps = static:1000
virtual_gid_maps = static:1000
virtual_transport = dovecot
dovecot_destination_recipient_limit = 1

sender_bcc_maps = hash:/etc/postfix/sender_bcc_maps
recipient_bcc_maps = hash:/etc/postfix/recipient_bcc_maps
```

Я выделил жирным имя домена и путь для директории с почтовыми ящиками. Не забудьте поменять эти параметры на свои. Сохраняем конфиг и продолжаем настройку. В таком виде сервер еще не готов. Нужно теперь создать все то, что описано в файле конфигурации. Создаем папку для файлов с конфигурацией подключения к mysql и сами файлы подключения.

```
# mkdir /etc/postfix/mysql && cd /etc/postfix/mysql
```

```
# mcedit relay_domains.cf
```

```
hosts = localhost
user = postfix
password = 12345678
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' and backupmx = '1'
```

```
# mcedit  virtual_alias_domain_maps.cf

hosts = localhost
user = postfix
password = 12345678
dbname = postfix
query = SELECT goto FROM alias,alias_domain WHERE alias_domain.alias_domain = '%d' and alias.address = CONCAT('%u', '@',
alias_domain.target_domain) AND alias.active = 1
```

```
# mcedit virtual_alias_maps.cf

hosts = localhost
user = postfix
password = 12345678
dbname = postfix
query = SELECT goto FROM alias WHERE address='%s' AND active = '1'
```

```
# mcedit virtual_mailbox_domains.cf
```

```
hosts = localhost
user = postfix
password = 12345678
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' AND backupmx = '0' AND active = '1'
```

```
# mcedit virtual_mailbox_maps.cf

hosts = localhost
user = postfix
password = 12345678
dbname = postfix
query = SELECT maildir FROM mailbox WHERE username='%s' AND active = '1'
```

Редактируем файл */etc/postfix/master.cf*. Нам надо добавить строки, касающиеся настройки Submission для того, чтобы почтовый сервер работал на 587 порту. Смартфоны очень часто при настройке используют этот порт по-умолчанию, где-то даже без возможности изменить эту настройку. Приводим секцию, отвечающую за эту работу к следующему виду.

```
submission inet n - n - - smtpd
 -o syslog_name=postfix/submission
 -o smtpd_tls_wrappermode=no
 -o smtpd_tls_security_level=encrypt
 -o smtpd_sasl_auth_enable=yes
 -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject
 -o smtpd_relay_restrictions=permit_mynetworks,permit_sasl_authenticated,defer_unauth_destination
 -o milter_macro_daemon_name=ORIGINATING
```

Обращаю внимание на пробел в начале строки, начиная со второй. Его надо обязательно оставить. Добавляем еще настройки для того, чтобы наш сервер поддерживал протокол SSL/TLS и слушал порт 465

```
smtps inet n - n - - smtpd
 -o syslog_name=postfix/smtps
 -o smtpd_tls_wrappermode=yes
 -o smtpd_sasl_auth_enable=yes
 -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject
 -o smtpd_relay_restrictions=permit_mynetworks,permit_sasl_authenticated,defer_unauth_destination
 -o milter_macro_daemon_name=ORIGINATING
```

В этот же файл добавляем еще одну настройку, которая будет указывать postfix, что доставкой почты у нас будет заниматься dovecot, который мы настроим следом. Добавляем в master.cf в самый конец.

```
dovecot unix - n n - - pipe
 flags=DRhu user=vmail:vmail argv=/usr/libexec/dovecot/deliver -f ${sender} -d ${recipient}
```

Сгенерируем самоподписанные ssl сертификаты для нашего почтового сервера. Позже отдельным пунктом я расскажу как использовать полноценные сертификаты. Они не всем нужны, поэтому показываю быструю настройку postfix на использование своих сертификатов, которые уже указаны в конфиге postfix.

Создаем директорию и сами сертификаты:

```
# mkdir /etc/postfix/certs
# openssl req -new -x509 -days 3650 -nodes -out /etc/postfix/certs/cert.pem -keyout /etc/postfix/certs/key.pem
```

Для генерации вам зададут несколько вопросов по поводу данных о сертификате. В принципе, можете там писать все, что угодно. Вот мои данные.

Создадим файлы для информации о ящиках, куда будет собираться вся входящая и исходящая почта.

```
# mcedit /etc/postfix/recipient_bcc_maps
```

```
@zeroxzed.ru all_in@zeroxzed.ru
```

```
# mcedit /etc/postfix/sender_bcc_maps
```

```
@zeroxzed.ru all_out@zeroxzed.ru
```

Создаем индексированные базы данных из этих файлов. Это нужно делать каждый раз, после изменения.

```
# postmap /etc/postfix/recipient_bcc_maps /etc/postfix/sender_bcc_maps
```

Теперь создайте два почтовых ящика all_in@zeroxzed.ru и all_out@zeroxzed.ru через postfixadmin.

Немного поясню по этим ящикам — для чего они нужны. Изначально я их делал, когда пользователи использовали протокол pop3 без сохранения писем на сервере. Это позволяло организовать бэкап всей переписки. Эти ящики очень быстро заполняются и занимают огромный объем, поэтому их обязательно надо чистить. Я просто скриптами регулярно собирал всю почту в архивы с именами в виде дат. Если нужно было какое-то письмо найти, то просто распаковывал нужный архив.

В случае с imap роль бэкапа отпадает, так как вся почта хранится на сервере. Но эти ящики все равно бывают полезны, когда пользователь, к примеру, удалил какое-то важное письмо и потом делает вид, что его и не было. Если это письмо пришло только сегодня и еще не успело улететь в бэкап, то кроме записи в логах об этом письме, вы не увидите само содержимое. А с такими ящиками все сразу будет понятно, и вопросы отпадут. Последнее применение — служба безопасности. Если у вас есть кто-то, кому положено читать всю переписку, то реализовать этот функционал можно таким простым способом.

Все основные настройки для postfix мы сделали. Некоторые из них завязаны на работу с dovecot, который мы еще не настроили. Поэтому больше postfix не трогаем, не перезапускаем. Идем настраивать dovecot — imap сервер нашей почтовой системы.

Настройка dovecot

Займемся настройкой dovecot — сервер доставки почты пользователю по протоколам pop3 и imap. Я не вижу причин использовать pop3. Он неудобен по сравнению с imap. Чаще всего pop3 отключаю вовсе. Но это уже на ваше усмотрение. Приведу пример с настройкой обоих протоколов. Помимо основного функционала по доставке почты, я настрою несколько полезных плагинов. Расскажу о них поподробнее:

- **Sieve** — выполняет фильтрацию почты по заданным правилам в момент локальной доставки на почтовом сервере. Удобство такого подхода в том, что вы один раз можете настроить правило сортировки, и оно будет работать во всех клиентах, которыми вы будете получать почту по imap. Правила создаются, хранятся и исполняются на самом сервере.
- **Acl** — позволяет пользователям расшаривать папки в своем почтовом ящике и предоставлять доступ к этим папкам другим пользователям. Не часто видел, чтобы этот функционал настраивали и использовали. Думаю, просто по незнанию. По мне так очень удобный и полезный функционал.

Часто вижу, что люди настраивают плагин **quota**, который позволяет ограничивать максимальный размер почтового ящика. Я лично в своей работе его не использую. Возможно, когда у тебя клиентов сотни и тысячи это имеет значение и надо обязательно настроить ограничение. Когда же ящиков меньше, нет смысла напрягать людей постоянной чисткой. Сейчас диски стоят не так дорого. Мне кажется, проще и дешевле увеличить место на сервере, нежели постоянно беспокоить пользователей необходимостью чистки ящика. Лучше ограничить максимальный размер письма, скажем 20-ю мегабайтами. Тогда

сильно забить ящик даже при большом желании быстро не получится. А почта все-таки важный инструмент в работе. Мне кажется, ее лучше хранить как можно дольше.

Есть еще один полезный плагин **expire**, который позволяет удалять устаревшие письма в определенных папках. Например, удалять все письма старше 30-ти дней в корзине и папке со спамом. Но реально пользоваться им не получается по простой причине. Разные почтовые клиенты создают различные папки для корзины и спама. Thunderbird создает папки с латинскими именами trash и spam, outlook с русскими, которые на почтовом сервере преобразуются в кодировку UTF7, мобильные клиенты тоже используют разные имена папок. В итоге нет единообразия, плагин полноценно не работает.

Я рассказал об этих плагинах для наводки. Сам их не настраиваю, но если вам захочется реализовать описанный функционал, можете сами разобраться и настроить.

Небольшую теорию я дал, теперь переходим к практике. Устанавливаем необходимые для dovecot пакеты.

```
# yum install dovecot dovecot-mysql dovecot-pigeonhole
```

Изначально конфиг dovecot разбит на отдельные сегменты и лежат в директории */etc/dovecot/conf.d*. Каждый файл — отдельный функционал. Мне не нравится прыгать по файлам, поэтому я храню все в едином общем файле конфигурации */etc/dovecot/dovecot.conf*. С ним мы и будем работать. Приводим его к следующему виду.

```
# mcedit /etc/dovecot/dovecot.conf
```

```
listen = * [::]

mail_plugins = mailbox_alias acl

protocols = imap pop3 sieve lmtp

mail_uid = 1000
mail_gid = 1000

first_valid_uid = 1000
last_valid_uid = 1000
```

```
log_path = /var/log/dovecot/main.log
info_log_path = /var/log/dovecot/info.log
debug_log_path = /var/log/dovecot/debug.log

ssl_protocols = !SSLv2 !SSLv3
ssl = required
verbose_ssl = no
ssl_cert = </etc/postfix/certs/cert.pem
ssl_key = </etc/postfix/certs/key.pem

ssl_cipher_list = ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-
GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-
SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-
AES256-SHA:DHE-RSA-AES256-SHA:ECDHE-RSA-DES-CBC3-SHA:ECDHE-ECDSA-DES-CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-
SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-
DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
ssl_dh_parameters_length = 2048
ssl_prefer_server_ciphers = yes

disable_plaintext_auth = yes

mail_location = maildir:/mnt/mail/%d/%u/

auth_default_realm = zeroxzed.ru

auth_mechanisms = PLAIN LOGIN

service auth {
 unix_listener /var/spool/postfix/private/dovecot-auth {
 user = postfix
 group = postfix
 mode = 0666
```

```
 }
unix_listener auth-master {
 user = vmail
 group = vmail
 mode = 0666
 }

unix_listener auth-userdb {
 user = vmail
 group = vmail
 mode = 0660
 }
}

service lmtp {
 unix_listener /var/spool/postfix/private/dovecot-lmtp {
 user = postfix
 group = postfix
 mode = 0600
 }

 inet_listener lmtp {
 address = 127.0.0.1
 port = 24
 }
}

userdb {
 args = /etc/dovecot/dovecot-mysql.conf
 driver = sql
 }

passdb {
```

```
 args = /etc/dovecot/dovecot-mysql.conf
 driver = sql
 }

auth_master_user_separator = *

plugin {
 auth_socket_path = /var/run/dovecot/auth-master

 acl = vfile
 acl_shared_dict = file:/mnt/mail/shared-folders/shared-mailboxes.db
 sieve = /mnt/mail/sieve/%u.sieve
 mailbox_alias_old = Sent
 mailbox_alias_new = Sent Messages
 mailbox_alias_old2 = Sent
 mailbox_alias_new2 = Sent Items
}

protocol lda {
 mail_plugins = $mail_plugins sieve
 auth_socket_path = /var/run/dovecot/auth-master
 deliver_log_format = mail from %f: msgid=%m %$
 log_path = /var/log/dovecot/lda-errors.log
 info_log_path = /var/log/dovecot/lda-deliver.log
 lda_mailbox_autocreate = yes
 lda_mailbox_autosubscribe = yes
 postmaster_address = root
}

protocol lmtp {
 info_log_path = /var/log/dovecot/lmtp.log
 mail_plugins = quota sieve
 postmaster_address = postmaster
```

```
 lmtp_save_to_detail_mailbox = yes
 recipient_delimiter = +
}

protocol imap {
 mail_plugins = $mail_plugins imap_acl
 imap_client_workarounds = tb-extra-mailbox-sep
 mail_max_userip_connections = 30
}

protocol pop3 {
 mail_plugins = $mail_plugins
 pop3_client_workarounds = outlook-no-nuls oe-ns-eoh
 pop3_uidl_format = %08Xu%08Xv
 mail_max_userip_connections = 30
}

service imap-login {
 service_count = 1
 process_limit = 500
 }

service pop3-login {
 service_count = 1
 }

service managesieve-login {
 inet_listener sieve {
 port = 4190
 }
}

namespace {
```

```
type = private
separator = /
prefix =
inbox = yes

mailbox Sent {
auto = subscribe
special_use = \Sent
}
mailbox "Sent Messages" {
auto = no
special_use = \Sent
}
mailbox "Sent Items" {
auto = no
special_use = \Sent
}
mailbox Drafts {
auto = subscribe
special_use = \Drafts
}
mailbox Trash {
auto = subscribe
special_use = \Trash
}
mailbox "Deleted Messages" {
auto = no
special_use = \Trash
}
mailbox Junk {
auto = subscribe
special_use = \Junk
}
```

```
 mailbox Spam {
 auto = no
 special_use = \Junk
 }
 mailbox "Junk E-mail" {
 auto = no
 special_use = \Junk
 }
 mailbox Archive {
 auto = no
 special_use = \Archive
 }
 mailbox Archives {
 auto = no
 special_use = \Archive
 }
}

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

Создаем группу и пользователя с указанными в конфиге uid 1000.

```
# groupadd  -g 1000 vmail
# useradd -d /mnt/mail/ -g 1000 -u 1000 vmail
# chown vmail. /mnt/mail
```

Создаем конфигурационные файлы для доступа к mysql базе.

```
# mcedit /etc/dovecot/dovecot-mysql.conf
```

```
driver = mysql
default_pass_scheme = CRYPT
connect = host=127.0.0.1 dbname=postfix user=postfix password=12345678
user_query = SELECT '/mnt/mail/%d/%u' as home, 'maildir:/mnt/mail/%d/%u' as mail, 1000 AS uid, 1000 AS gid,
concat('*:bytes=', quota) AS quota_rule FROM mailbox WHERE username = '%u' AND active = '1'
password_query = SELECT username as user, password, '/mnt/mail/%d/%u' as userdb_home, 'maildir:/mnt/mail/%d/%u' as
userdb_mail, 1000 as userdb_uid, 1000 as userdb_gid, concat('*:bytes=', quota) AS userdb_quota_rule FROM mailbox WHERE
username = '%u' AND active = '1'
```

Создадим директорию и файлы для логов.

```
# mkdir /var/log/dovecot
# cd /var/log/dovecot && touch main.log info.log debug.log lda-errors.log lda-deliver.log lmtp.log
# chown -R vmail:dovecot /var/log/dovecot
```

Создаем пару служебных папок для плагинов sieve и acl.

```
# mkdir /mnt/mail/sieve && mkdir /mnt/mail/shared-folders
# chown -R vmail. /mnt/mail
```

И небольшой штрих в завершении настройки.

```
# chown vmail. /var/run/dovecot/auth-master
```

Уже не помню, зачем это было нужно, запись осталась в черновиках. Знаю только, что какая-то ошибка всплывала без этого.

На этом основная настройка почтового сервера на базе postfix и dovecot завершена. Можно перезапускать службы и проверять работу системы.

```
# systemctl restart postfix
# systemctl start dovecot
# systemctl enable dovecot
```

Проверка работы почтового сервера

Самаый простой и быстрый способ проверить работу почтового сервера — отправить на него письмо. Я буду отправлять со своего почтового адреса zeroxzed@gmail.com на адрес root@zeroxzed.ru. Вот что должно быть в логе, если у вас все правильно настроено и почтовый сервер нормально работает.

```
# cat /var/log/maillog
```

```
Mar 10 21:56:27 mail postfix/smtpd[28075]: connect from mail-yw0-f172.google.com[209.85.161.172]
Mar 10 21:56:28 mail postfix/smtpd[28075]: Anonymous TLS connection established from mail-yw0-
f172.google.com[209.85.161.172]: TLSv1.2 with cipher ECDHE-RSA-AES128-GCM-SHA256 (128/128 bits)
Mar 10 21:56:28 mail postfix/smtpd[28075]: D4263420BB7B: client=mail-yw0-f172.google.com[209.85.161.172]
Mar 10 21:56:29 mail postfix/cleanup[28086]: D4263420BB7B: message-id=<CAHWPLcOeqf6uNHRg34+wuppDUGPDLY=fp8s-
E=o9fmxYMS48cQ@mail.gmail.com>
Mar 10 21:56:29 mail postfix/qmgr[28042]: D4263420BB7B: from=<zeroxzed@gmail.com>, size=2533, nrcpt=2 (queue active)
Mar 10 21:56:29 mail postfix/pipe[28089]: D4263420BB7B: to=<all_in@zeroxzed.ru>, relay=dovecot, delay=0.39,
delays=0.33/0.02/0/0.05, dsn=2.0.0, status=sent (delivered via dovecot service)
Mar 10 21:56:29 mail postfix/pipe[28090]: D4263420BB7B: to=<root@zeroxzed.ru>, relay=dovecot, delay=0.4,
delays=0.33/0.03/0/0.04, dsn=2.0.0, status=sent (delivered via dovecot service)
Mar 10 21:56:29 mail postfix/qmgr[28042]: D4263420BB7B: removed
Mar 10 21:56:29 mail postfix/smtpd[28075]: disconnect from mail-yw0-f172.google.com[209.85.161.172]
```

Пояснять тут нечего, по логу все понятно. Письмо было доставлено в указанный ящик и в общий ящик для сбора всей входящей почты. В директории */mnt/mail* была создана директория с именем домена zeroxzed.ru, а в ней созданы 3 папки с именами ящиков:

- all_in@zeroxzed.ru
- all_out@zeroxzed.ru
- root@zeroxzed.ru

Директории с почтовыми ящиками создаются в момент получения первого письма в ящик. Непрочитанное письмо помещается в директорию /new в почтовом ящике. После прочтения переносится в /cur.

Попробуем теперь подключиться к почтовому ящику по imap, прочитать письмо и отправить ответ. Настроим любой почтовый клиент для проверки работы настроенного почтового сервера. Я для этих целей буду использовать Thunderbird. Из всех почтовых клиентов мне он нравится больше всего. В основном из-за его портированной версии. Указываем следующие настройки.

Так как мы используем самоподписанный сертификат ssl, почтовый клиент предупредит нас о том, что серверу нельзя доверять.

Нас это не пугает, добавляем сертификат в список доверенных и продолжаем работать. Позже получим и настроим нормальный сертификат.

Я подключился к почтовому ящику и увидел тестовые письма. Отвечу на одно из них и посмотрю в логе, как прошла отправка. У меня еще раз выскочило окно с предупреждением о небезопасном сертификате. Еще раз добавляю его в исключения. Это нормально, сертификат проверяется во время получения почты в dovecot, а во время отправки в postfix. Так что нужны 2 подтверждения. Отправляю письмо еще раз и смотрю лог.

```
# cat /var/log/maillog
```

```
Mar 10 22:10:12 mail postfix/smtpd[28764]: connect from broadband-75-37-235-139.moscow.gw.ru[75.37.235.139]
Mar 10 22:10:12 mail postfix/smtpd[28764]: Anonymous TLS connection established from
broadband-75-37-235-139.moscow.gw.ru[75.37.235.139]: TLSv1.2 with cipher ECDHE-RSA-AES128-GCM-SHA256 (128/128 bits)
Mar 10 22:10:12 mail postfix/smtpd[28764]: B24C2420BB70: client=broadband-75-37-235-139.moscow.gw.ru[75.37.235.139],
sasl_method=PLAIN, sasl_username=root@zeroxzed.ru
Mar 10 22:10:12 mail postfix/cleanup[28779]: B24C2420BB70: message-id=<aaac96c3-197e-c6bd-4dfe-85d09bce216a@zeroxzed.ru>
Mar 10 22:10:12 mail postfix/qmgr[28042]: B24C2420BB70: from=<root@zeroxzed.ru>, size=955, nrcpt=2 (queue active)
```

```
Mar 10 22:10:12 mail postfix/smtpd[28764]: disconnect from broadband-75-37-235-139.moscow.gw.ru[75.37.235.139]
Mar 10 22:10:12 mail postfix/pipe[28784]: B24C2420BB70: to=<all_out@zeroxzed.ru>, relay=dovecot, delay=0.14,
delays=0.07/0.01/0/0.06, dsn=2.0.0, status=sent (delivered via dovecot service)
Mar 10 22:10:13 mail postfix/smtp[28783]: B24C2420BB70: to=<zeroxzed@gmail.com>, relay=gmail-smtp-
in.l.google.com[64.233.163.26]:25, delay=0.62, delays=0.07/0.01/0.28/0.26, dsn=2.0.0, status=sent (250 2.0.0 OK 1489173013
13si2106703ljv.3 - gsmtp)
Mar 10 22:10:13 mail postfix/qmgr[28042]: B24C2420BB70: removed
```

Все в порядке. Видно подключение с моего ip, успешную sasl авторизацию, формирование письма на сервере, присваивание ему message-id, отправка копии письма в ящик для сбора исходящей почты и отправка оригинала в ящик получателя. Все этапы прошли без ошибок.

Расскажу, куда еще надо смотреть для отладки почтовой системы. Да и не только отладки, во время работы периодически придется разбираться, куда ушло то или иное письмо, кто и когда подключался к ящику. Разные ситуации бывают. В файле */var/log/dovecot/lda-deliver.log* содержится информация обо всех пришедших письмах — когда, от кого и в какой ящик было положено.

```
Mar 10 22:25:29 lda(all_in@zeroxzed.ru): Info: mail from zeroxzed@gmail.com:
msgid=<CAHWPLcNG=WMOoWW2Y_Lw4qn9+V4TOrbxZpwtA=O+CSEBaiwuBg@mail.gmail.com> saved mail to INBOX
Mar 10 22:25:29 lda(root@zeroxzed.ru): Info: mail from zeroxzed@gmail.com:
msgid=<CAHWPLcNG=WMOoWW2Y_Lw4qn9+V4TOrbxZpwtA=O+CSEBaiwuBg@mail.gmail.com> saved mail to INBOX
Mar 10 22:25:49 lda(all_out@zeroxzed.ru): Info: mail from root@zeroxzed.ru: msgid=<75358e4d-7c8e-24c2-
a21f-7ee0df2a4704@zeroxzed.ru> saved mail to INBOX
```

В */var/log/dovecot/info.log* информация о подключениях к почтовым ящикам — кто, когда, откуда и каким способом авторизовывался на сервере.

```
Mar 10 22:10:20 imap-login: Info: Login: user=<root@zeroxzed.ru>, method=PLAIN, rip=75.37.235.139, lip=188.35.19.125,
mpid=28790, TLS, session=<3tDeHGVKpQBNJeCL>
Mar 10 22:19:39 imap-login: Info: Login: user=<root@zeroxzed.ru>, method=PLAIN, rip=75.37.235.139, lip=188.35.19.125,
mpid=29248, TLS, session=<7VY8PmVKbwBNJeCL>
```

Остальное уже не так полезно. Сами посмотрите, что собирается в остальных лог файлах.

На текущий момент сервер полностью работоспособен. В таком виде им без проблем можно пользоваться. Но функционал полностью не раскрыт.

Использовать плагины sieve и acl через удаленные почтовые клиенты неудобно. Проще всего их настроить через web почту roundcube. Установим эту web панель на наш почтовый сервер.

Установка web интерфейса roundcube

Установим и настроим самый популярный web интерфейс для postfix — roundcube. Скачиваем исходники.

```
# cd /usr/src
# wget https://github.com/roundcube/roundcubemail/releases/download/1.2.9/roundcubemail-1.2.3-complete.tar.gz
```

> Не забудьте проверить в момент установки, какая версия является самой свежей на текущий момент. Нет необходимости устанавливать устаревшую версию. Рекомендую ставить самую последнюю на момент настройки.

```
# yum install php-pear php-mcrypt php-intl php-ldap php-pear-Net-SMTP php-pear-Net-IDNA2 php-pear-Mail-Mime php-pear-Net-
Sieve
# tar -xzvf roundcubemail-*
# mv roundcubemail-1.2.9 /var/www/html/webmail
# chown -R apache. /var/www/html/webmail
```

Переходим в браузер по следующей ссылке для установки roundcube — *http://188.35.19.125/webmail/installer/*. Вы увидите несколько незначительных замечаний. На них можно не обращать внимание, если установщик позволяет нажать кнопку NEXT. Единственное, рекомендую установить правильный часовой пояс в */etc/php.ini* и перезапустить после этого httpd.

На следующем этапе нам надо указать настройки подключения к mysql базе. Предварительно ее следует создать через phpmyadmin. Я создал пользователя roundcube и такую же базу с полными правами пользователя на нее. Эти параметры указал в настройках.

Так же на этой странице нужно будет указать несколько параметров:

- smtp_server — пусто (ничего не пишем)
- language — ru_RU
- Выбираем плагины — managesieve, userinfo, acl. Остальные на свое усмотрение.

Жмем CREATE CONFIG. Должны увидеть сообщение:

```
The config file was saved successfully into RCMAIL_CONFIG_DIR directory of your Roundcube installation.
```

Жмем CONTINUE. Открывается страница с проверкой настроек. Тут проверять неудобно, можно этого не делать. Зайдем в почтовый ящик и там все проверим. Если что, конфиг потом все равно можно вручную отредактировать. Папку */var/www/html/webmail/installer* удаляем. Заходим в почтовый ящик через roundcube — *http://188.35.19.125/webmail/* Набирать нужно полное имя ящика и пароль. Если все сделали правильно, должны попасть в свой почтовый ящик.

Ну вот и все. Можно пользоваться web интерфейсом для почтового сервера. У меня есть статья по настройке мобильной версии roundcube. Рекомендую ее настроить, если есть необходимость. Тема качественная и добротная. Пользоваться удобно.

Дальше рассмотрим настройку и использование плагинов acl и sieve с помощью roundcube.

Настройка фильтра почты sieve

Sieve очень удобная штука, но вот хорошего интерфейса для управления через почтовый клиент я не знаю. Существует плагин для thunderbird, который так и называется sieve. Но лично мне он не понравился вообще, так как предлагает писать правила определенным кодом. Для этого надо знать синтаксис, тратить время. Можете сами на него посмотреть — https://github.com/thsmi/sieve.

К счастью, есть удобный способ писать правила фильтрации для sieve через roundcube. Там это реализовано отдельным плагином **managesieve**, который мы активировали во время установки. Для создания правила фильтрации, зайдите в свой почтовый ящик через roundcube. Переходите в раздел Настройки -> Фильтры и создавайте новое правило.

После этого письма будут обрабатываться правилом сразу после поступления в почтовый сервер, в независимости от вашего подключения к ящику. В папке */mnt/mail/sieve* появилась запись с настроенным правилом. Можете познакомиться с синтаксисом написания правил. Он не сложный.

Настройка автоответчика

В roundcube есть замечательная возможность настроить автоответчик в почтовом ящике. Это актуально, к примеру, если вы уходите в отпуск. Вы можете сами настроить автоответчик, который будет отправлять письмо с указанной вами информацией всем, от кого будут приходить письма в ваш ящик. Возможность эта реализована на базе того же плагина managesieve. По-умолчанию она отключена. Активировать ее нужно вручную.

Для того, чтобы модуль автоответчика заработал, отредактируйте конфигурационный файл плагина. Для этого открываем его в моем случае по следующему адресу:

```
# mcedit /var/www/html/webmail/plugins/managesieve/config.inc.php
```

Изменяем там параметр:

```
$config['managesieve_vacation'] = 1;
```

После этого достаточно просто обновить веб интерфейс roundcube, и появятся новые настройки по адресу Настройки -> Отпуск

На вкладке Дополнительные настройки есть возможность настроить различные полезные действия, в том числе пересылку входящей почты вашему заместителю.

Общие папки по imap

Рассмотрим настройку необычного и полезного функционала в виде общих папок. С их помощью один пользователь почтового ящика может предоставить другому пользователю доступ к папке внутри своего почтового ящика. Где и как использовать этот функционал, каждый может придумать сам, в зависимости от своих потребностей. Мне кажется это удобным в том случае, когда создан какой-то общий ящик, на который только поступает информация и нет необходимости писать ответ от его имени. То есть по сути работает как обычный почтовый алиас. Но в случае с алиасом и несколькими почтовыми

ящиками, письмо падает в каждый ящик. Если таких писем и получателей много, то идет большое дублирование одного и того же письма в рамках почтового сервера. Если сделать ящик и расшарить на нем папку, подключить ее всем пользователям, то дублирования почты не будет. Каждый сможет прочитать письмо, без необходимости его доставки в каждый конкретный ящик.

Настроим общую папку imap. Хотя настраивать нам, по сути, нечего. Мы уже все настроили ранее. Добавили соответствующие настройки в dovecot и активировали плагин **acl** в roundcube. Теперь нужно просто сделать папку и открыть ее для другого пользователя. Для этого идем в раздел Настройки -> Папки. Создаем там любую папку. В моем случае это папка с названием Общая.

Добавляем необходимый доступ либо всем ящикам в домене, либо какому-то конкретному. Так же можно указать, какого рода это будет доступ:

- чтение
- запись
- удаление

Заходим в ящик, которому добавили общий доступ и проверяем.

Все в порядке, общая папка imap настроена и подключена. В папке */mnt/mail/shared-folders* появился файл с настроенным выше правилом.

На этом настройка пользовательского функционала закончена. В принципе, почтовый сервер полностью готов к работе. Но мы сделаем еще несколько полезных настроек на стороне сервера.

Настройка dkim и spf

Напишу своими словами как я понимаю работу **dkim**. С помощью dkim вся исходящая почта сервера подписывается электронной цифровой подписью, связанной с именем домена. Открытый ключ шифрования с помощью DNS публикуется в txt записи. Таким образом, удаленный сервер, при получении письма от вас, сравнивает цифровую подпись с опубликованным в dns открытым ключом вашего домена. Если все в порядке, то считает, что ваше письмо в

самом деле пришло от вас, а не от мошенников. То есть с помощью этой технологии можно однозначно идентифицировать отправителя.

Некоторые считают, что эта технология помогает бороться со спамом. Не знаю, насколько это верно. Спамеру не составит большого труда настроить на своем сервере dkim и отправлять спам, но подписанный электронной цифровой подписью. Теоретически, dkim помогает защититься от подделки адреса отправителя, когда письмо якобы от вас шлет совсем другой сервер. Но с этим можно бороться и другими способами. В общем, я до конца не понимаю, зачем это надо. Прошу поделиться в комментариях тем, к кого есть бОльший опыт. Я только недавно стал шагать в ногу со временем и настраивать dkim. Раньше всегда без него обходился.

> Плюс настройки dkim я вижу только в том, что автоматические фильтры определения спама будут добавлять больше траста письмам с корректной dkim подписью. У вас будет больше шанса не попасть в спам при прочих равных условиях. В принципе, ради этого можно немного заморочиться.

Для настройки dkim устанавливаем соответствующий пакет:

```
# yum install opendkim
```

Создаем директорию для хранения ключей:

```
# mkdir -p /etc/postfix/dkim && cd /etc/postfix/dkim
```

Генерируем ключи для домена:

```
# opendkim-genkey -D /etc/postfix/dkim/ -d zeroxzed.ru -s mail
```

zeroxzed.ru	имя почтового домена
mail	непосредственно имя сервера

На выходе получаете пару файлов — закрытый (приватный) и открытый ключ. Закрытый останется на сервере, открытый будет опубликован в dns. Переименуем их сразу, чтобы не путаться, если у вас будет несколько доменов. Ключи нужно будет делать для каждого домена.

```
# mv mail.private mail.zeroxzed.ru.private
# mv mail.txt mail.zeroxzed.ru.txt
```

Создаем файл с таблицей ключей, в которой будут описаны все домены. В данном случае только один.

```
# mcedit keytable
```

```
mail._domainkey.zeroxzed.ru zeroxzed.ru:mail:/etc/postfix/dkim/mail.zeroxzed.ru.private
```

Тут же создаем еще один файл, в котором будет описано, каким ключом подписывать письма каждого домена. У нас один домен, поэтому только одна запись.

```
# mcedit signingtable
```

```
*@zeroxzed.ru mail._domainkey.zeroxzed.ru
```

Выставляем права доступа на все файлы:

```
# chown root:opendkim *
# chmod u=rw,g=r,o= *
```

Рисуем конфиг службы.

```
# mcedit /etc/opendkim.conf
```

```
AutoRestart Yes
AutoRestartRate 10/1h
PidFile /var/run/opendkim/opendkim.pid
Mode sv
```

```
Syslog yes
SyslogSuccess yes
LogWhy yes
UserID opendkim:opendkim
Socket inet:8891@localhost
Umask 022
Canonicalization relaxed/relaxed
Selector default
MinimumKeyBits 1024
KeyFile /etc/postfix/dkim/mail.zeroxzed.ru.private
KeyTable /etc/postfix/dkim/keytable
SigningTable refile:/etc/postfix/dkim/signingtable
```

Добавляем в конфигурационный файл postfix в самый конец следующие параметры:

```
# mcedit /etc/postfix/main.cf
```

```
smtpd_milters = inet:127.0.0.1:8891
non_smtpd_milters = $smtpd_milters
milter_default_action = accept
milter_protocol = 2
```

Перезапускаем postfix и dkim, последний добавляем в автозагрузку.

```
# systemctl restart postfix
# systemctl restart opendkim.service
# systemctl enable opendkim.service
```

Теперь нам надо добавить открытый ключ в dns. Идем в консоль управления dns и добавляем новую txt запись. Ее содержание берем из файла */etc/postfix/dkim/mail.zeroxzed.ru.txt*

```
cat /etc/postfix/dkim/mail.zeroxzed.ru.txt
```

```
mail._domainkey IN TXT ( "v=DKIM1; k=rsa; "
"p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQClZX2xWRDISlVLF4b4pUiinY5N9WN7VXEHeyPw8smHTamXh35wJoh+j0+MIQDWG/KtdCcETeawTuypXbvtb
neXniYR0iiv6kt754T2WXBjz7O/uHL+vK58LhJsm4TGyhUN6ZBit+w22jG92zdeybSZeU/g7hQdkaAAi0I+0nIkUwIDAQAB" ) ; ----- DKIM key mail for
zeroxzed.ru
```

Убираем кавычки, лишние проблемы и вставляем. Должно получиться вот так:

Проверяю работу. Отправляю письмо на gmail и смотрю лог почтового сервера:

```
# cat /var/log/maillog
```

```
Mar 17 17:40:26 mail postfix/smtpd[22352]: connect from localhost[127.0.0.1]
Mar 17 17:40:26 mail postfix/smtpd[22352]: BB1794195584: client=localhost[127.0.0.1]
Mar 17 17:40:26 mail postfix/cleanup[22364]: BB1794195584: message-id=<baf63dcec016594d49f2d80f815e5d26@zeroxzed.ru>
Mar 17 17:40:26 mail opendkim[21744]: BB1794195584: DKIM-Signature field added (s=mail, d=zeroxzed.ru)
Mar 17 17:40:26 mail postfix/qmgr[21990]: BB1794195584: from=<root@zeroxzed.ru>, size=593, nrcpt=2 (queue active)
Mar 17 17:40:26 mail postfix/pipe[22369]: BB1794195584: to=<all_out@zeroxzed.ru>, relay=dovecot, delay=0.14,
delays=0.11/0.02/0/0.02, dsn=2.0.0, status=sent (delivered via dovecot service)
Mar 17 17:40:26 mail postfix/smtpd[22352]: disconnect from localhost[127.0.0.1]
Mar 17 17:40:27 mail postfix/smtp[22370]: BB1794195584: to=<zeroxzed@gmail.com>, relay=gmail-smtp-
in.l.google.com[64.233.163.26]:25, delay=0.84, delays=0.11/0.02/0.31/0.4, dsn=2.0.0, status=sent (250 2.0.0 OK 1489761627
185si4568435lfa.398 - gsmtp)
```

Все в порядке, электронная цифровая подпись установлена. Проверим, как гугл отреагировал на нашу подпись:

Тоже все в порядке. Подпись выполнена корректно, проверку прошла. Дополнительно, проверить корректность dkim записи в dns можно онлайн сервисом — http://dkimcore.org/c/keycheck.

Настроим еще одно средство для повышения доверия к нашей почте со стороны других серверов — **spf**. Расскажу опять своими словами для чего это нужно. Spf запись добавляется в виде txt записи в dns вашего домена. С помощью этой записи вы указываете, какие ip адреса имеют право отправлять почту от вашего имени. Если кто-то из спамеров будет использовать ваше имя домена при рассылке спама, он не пройдет проверку по spf и скорее всего будет идентифицирован как спам.

Можно указать конкретные ip адреса в записи, а можно сказать, чтобы ip адреса проверялись по спискам A и MX записей. У нас простой случай и только 1 сервер с одним ip, поэтому укажем конкретно этот ip адрес. Идем в панель управления dns и добавляем новую txt запись.

```
zeroxzed.ru. TXT v=spf1 ip4:188.35.19.125 ~all
```

Больше ничего делать не надо. Снова отправляем письмо на gmail и смотрим логи.

Обращаю внимание на прошлый скрин, когда мы проверяли dkim и еще не настроили spf, и этот. Видно, что запись работает, гугл определил наш ip, как доверенный для отправки писем с этого домена.

Дополнительный функционал почтового сервера postfix

Я рассмотрел и настроил наиболее актуальный с моей точки зрения функционал почтового сервера. В статье я основывался исключительно на своем опыте работы с почтой в малых и средних организациях, поэтому не претендую на истину в последней инстанции. Рекомендую осмысленно подходить к настройке своего сервера и решать, что нужно именно вам. Будет хорошо, если кто-то укажет на мои ошибки или подскажет какие-то более удобные и эффективные приемы для решения затронутых задач.

В данном виде почтовый сервер представляет собой готовое и законченное решение, но есть еще несколько вещей, которые ему бы не помешали. Я их сейчас перечислю, а затем постараюсь постепенно писать статьи на указанные темы и ставить на них ссылки в этой теме. Вот список того, что по моему

мнению нужно еще настроить на почтовом сервере:

1. Защиту от подбора паролей с помощью fail2ban.
2. Мониторинг почтового сервера postfix с помощью zabbix.
3. Сбор статистики с помощью pflogsumm или чего-то подобного.
4. Просмотр и анализ логов с помощью webmin.
5. Использование бесплатных сертификатов let’s encrypt.
6. Регулярную очистку служебных почтовых ящиков.
7. Бэкап всей почтовой базы.

Расскажу еще почему я не настраиваю некоторые популярные программы, которые использую на почтовых серверах:

1. Clamav — известный антивирус. Считаю, что сейчас он не актуален, так как вирусов, от которых он способен защитить, я уже давно не видел. Сейчас вирусная эпидемия шифровальщиков. От них он не защищает.
2. Spamassasin — популярный бесплатный антиспам фильтр. Скажу честно, работал с ним очень мало и могу быть не объективен. Насколько я видел его настройку и работу — он требует к себе некоторого внимания, калибровки, особенно на начальном этапе. Мне обычно не хочется этим заниматься.
3. Graylist — эффективное средство борьбы со спамом. Я уже подробно его рассматривал, когда писал про iredmail, так что не буду повторяться. Скажу лишь, что режет спам очень эффективно и бесплатно, но есть существенные неудобства, которые по моему мнению не перекрывают плюсы. Поэтому я не использую.

В качестве антиспама я предпочитаю коммерческое решение — Kaspersky Anti-Spam. Я знаю этот продукт уже лет 8. Он действительно отлично фильтрует спам. Ложных срабатываний вообще не припоминаю, 95% спама фильтрует, может больше. С ним вопрос спама отпадает вообще. Стоит он недорого, можно купить лицензию на меньшее количество ящиков, чем реально используется в системе. Этот вопрос никак не отслеживается и на качество работы не влияет. Но нужно понимать, что это уже нарушение лицензионного соглашения. Но можно всякие хитрости придумать, чтобы и фильтровать и не нарушать.

Борьба со спамом средствами postfix

Сначала хотел сразу все настройки postfix разместить в соответствующем разделе в едином конфиге, но потом передумал и решил все же вынести этот вопрос на отдельное рассмотрение. Возможно, не каждому захочется сразу в эту тему углубляться. Все, что рассказано выше, позволит настроить полноценный почтовый сервер, который будет успешно принимать почту и доставлять ее пользователям. Но в таком виде он будет принимать слишком много спама, но зато не будет проблем с тем, что от кого-то что-то не придет. Как ни крути, но все средства борьбы со спамом так или иначе несут

накладные расходы в виде ложных срабатываний с той или иной вероятностью. Если вы решите не заморачиваться и купить Kaspersky Anti-Spam, можете этот раздел не читать. Он сам реализует все те проверки, что мы будем делать. Если же хотите своими силами бороться со спамом средствами postfix, то давайте дальше разбираться.

Я буду использовать штатные возможности postfix, позволяющие отсеять спам по тем или иным параметрам еще до получения письма. Это очень эффективный способ с точки зрения производительности. Благодаря этому, правильно настроенный на отсев спама postfix часто ставят перед exchange, чтобы снизить на него нагрузку. Сразу дам ссылки на официальную документацию с описанием параметров, которые я буду использовать:

1. smtpd_helo_restrictions
2. smtpd_sender_restrictions
3. smtpd_recipient_restrictions
4. smtpd_data_restrictions
5. smtpd_client_restrictions

Есть еще парочка — smtpd_etrn_restrictions и smtpd_end_of_data_restrictions, но я ими не пользуюсь.

> Обращаю внимание на то, что нужно очень аккуратно работать с настройками, о которых пойдет речь. Нужно четко понимать как, зачем и что вы делаете. Неверные настройки могут нарушить нормальное хождение почты. Нужно уметь анализировать лог файл почтового сервера и понимать, что там происходит.

Долго думал, как лучше всего представить информацию по этому разделу. В итоге решил просто написать часть конфига, которая отвечает за restrictions с комментариями. Более подробную информацию по каждой настройке вы можете найти в официальной документации postfix, ссылки я привел выше, либо в гугле. Данные настройки это моя многолетняя калькуляция различных параметров, собранных из черновиков и рабочих серверов. Не везде все было настроено именно в таком виде, так как ситуации бывают разные. Сейчас я постарался все собрать в одном месте и аккуратно описать. Те проверки в борьбе со спамом, что вам не нужны, просто закомментируйте. В конце я еще пройдусь по некоторым из них и поделюсь своим опытом.

```
#Описание списков исключений
smtpd_restriction_classes = white_client_ip,
                            black_client_ip,
                            block_dsl,
```

```
                          white_client,
                          white_helo,
                          black_client,
                          mx_access

# IP адреса, которые нужно пропускать всегда
white_client_ip         = check_client_access hash:/etc/postfix/lists/white_client_ip

# IP адреса, которые нужно блокировать всегда
black_client_ip         = check_client_access hash:/etc/postfix/lists/black_client_ip

# E-mail, которые нужно пропускать всегда
white_client            = check_sender_access hash:/etc/postfix/lists/white_client

# E-mail, которые нужно блокировать всегда
black_client            = check_sender_access hash:/etc/postfix/lists/black_client

# Неправильные значения HELO, которые мы тем не менее принимаем
white_helo = check_sender_access hash:/etc/postfix/lists/white_helo
# Правила для блокировки различных динамических ip.
block_dsl               = check_client_access regexp:/etc/postfix/lists/block_dsl

# Список приватных сетей, которые не могут быть использованы в качестве IP для MX записей
mx_access               = check_sender_mx_access cidr:/etc/postfix/lists/mx_access

# Проверки на основе данных, переданных в HELO/EHLO hostname
smtpd_helo_restrictions =       permit_mynetworks,
                permit_sasl_authenticated,
                white_client_ip,
                white_helo,
                black_client_ip,
                block_dsl,
                # Отказываем серверам, у которых в HELO несуществующий или не FQDN адрес
```

```
                reject_invalid_helo_hostname,
                reject_non_fqdn_helo_hostname,
                # Запрещаем приём писем от серверов, представляющихся адресом, для которого не существует A или MX записи.
                reject_unknown_helo_hostname

# Проверки клиентского компьютера или другого почтового сервера, который соединяется с сервером postfix для отправки письма
smtpd_client_restrictions =     permit_mynetworks,
                permit_sasl_authenticated,
                # Отвергает запрос, когда клиент отправляет команды SMTP раньше времени, еще не зная, поддерживает ли
Postfix конвейерную обработку команд ESMTP
                reject_unauth_pipelining,
                # Блокируем клиентов с адресами from, домены которых не имеют A/MX записей
                reject_unknown_address,
                reject_unknown_client_hostname

# Проверки исходящей или пересылаемой через нас почты на основе данных MAIL FROM
smtpd_sender_restrictions =     permit_mynetworks,
                permit_sasl_authenticated,
                white_client,
                black_client,
                # Запрет отправки писем, когда адрес MAIL FROM не совпадает с логином пользователя
                reject_authenticated_sender_login_mismatch,
                # Отклоняем письма от несуществующих доменов
                reject_unknown_sender_domain,
                # Отклоняем письма от доменов в не FQDN формате
                reject_non_fqdn_sender,
                # Отклонение писем с несуществующим адресом отправителя
                reject_unlisted_sender,
                reject_unauth_destination,
                # Отклонять сообщения от отправителей, ящики которых не существуют, использовать аккуратно
                #reject_unverified_sender,
                mx_access
```

```
# Правила приема почты нашим сервером на основе данных RCPT TO
smtpd_recipient_restrictions =  permit_mynetworks,
                permit_sasl_authenticated,
                # Отклоняет всю почту, что адресована не для наших доменов
                reject_unauth_destination,
                # Отклонение писем с несуществующим адресом получателя
                reject_unlisted_recipient,
                # Отклоняет сообщения на несуществующие домены
                reject_unknown_recipient_domain,
                # Отклоняет сообщения если получатель не в формате FQDN
                reject_non_fqdn_recipient,
                # Отклоняем прием от отправителя с пустым адресом письма, предназначенным нескольким получателям.
                reject_multi_recipient_bounce
```

У меня во всех ограничениях первыми правилами стоят разрешения для mynetworks и авторизовавшихся пользователей. Важно понимать, что это значит и для чего сделано. Ограничения читаются последовательно в порядке их перечисления. Таким образом, мы своих пользователей пускаем мимо ограничений, а для всех остальных выполняются проверки.

Теперь важные комментарии по указанным параметрам. Если бы все почтовые сервера всех системных администраторов были настроены по правилам, то эти комментарии не были бы нужны. Пройдемся по некоторым ограничениям, которые нужно включать осторожно:

- reject_invalid_helo_hostname и reject_unknown_helo_hostname — под эти правила иногда попадают почтовые серверы клиентов, которые не очень хорошо настроены. У них бывают неправильные адреса, кривые записи dns, отсутствие обратных зон и т.д. Их не много, но попадаются. Это не страшно, если вы регулярно следите за сервером. Отправитель получит сразу сообщение о том, что его письмо не дошло до вас. Если он как-то сообщит вам о проблеме, вы легко добавите его в белый список и все будет нормально. Если вам не хочется следить за сервером, лучше не указывайте эти ограничения. Но спам они отсекают не плохо. Сюда попадают все завирусованные компьютеры и сервера без нормальных настроек dns (а их чаще всего и нет).
- reject_unverified_sender — специально его закомментировал. Я тестировал этот параметр. В принципе, работает нормально, но есть, как обычно, нюансы. Поясню, что делает этот параметр. Когда вам кто-то шлет письмо, ваш сервер обращается к серверу отправителю и спрашивает его стандатрной командой, есть ли на сервере такой отправитель. Если удаленный сервер отвечает, что есть, то никаких проблем — письмо принимается. Если удаленный сервер не отвечает или говорит, что такого адресата нет — письмо отклоняем. Очевидно, что такие проверки создают дополнительную постоянную нагрузку. Это нужно учитывать. К тому же, у вас почтовый лог постоянно будет забит этими проверками, особенно, если вам приходит много спама. На каждое спамовое письмо будет идти проверка, а сервера отправителя скорее всего либо нет, либо он

неправильный, либо не отвечает и т.д. Все это будет постоянно проверяться и фиксироваться. В общем, я не использую.

На время отладки ограничений, рекомендую пользоваться параметром:

```
soft_bounce = yes
```

Когда он включен, все ответы сервера с кодами ошибок 5XX, заменяются на 4XX. То есть постоянная ошибка, которая сразу отклоняет письмо, заменяется на временную, которая предлагает повторить отправку позже. Таким образом, вы увидите работу всех ограничений, но письма не будут отклонены навсегда. Сервер отправителя через некоторое время снова придет к вам с новой попыткой доставки почты. Письмо безвозвратно не отклоняется. Вы можете проанализировать работу фильтра и решить, ставить его на постоянную работу или с ним что-то не так.

Создадим теперь файлы с белыми и черными списками.

```
cd /etc/postfix/lists && touch white_client_ip black_client_ip white_client black_client white_helo block_dsl mx_access
```

Ниже пример содержания этих файлов. Вы можете менять по своему усмотрению.

```
# cat white_client_ip
195.28.34.162 OK
141.197.4.160 OK
```

```
# cat black_client_ip
205.201.130.163 REJECT You IP are blacklisted!
198.2.129.162 REJECT You IP are blacklisted!
```

```
# cat white_client
# Принимать всю почту с домена яндекс
yandex.ru OK
# Разрешить конкретный ящик
spammer@mail.ru OK
```

```
# cat black_client
# Блокировать всю почту с домена mail.ru
mail.ru REJECT You domain are blacklisted!
# Блокировать конкретный ящик
spam@rambler.ru REJECT You e-mail are blacklisted!
```

```
# cat white_helo
# Могут попадаться вот такие адреса, которые не пройдут наши проверки
ka-s-ex01.itk.local     OK
exchange.elcom.local    OK
```

```
# cat block_dsl
/^dsl.*\..*\..*/i                               553 AUTO_DSL spam
/dsl.*\..*\..*/i                                553 AUTO_DSL1 spam
/[ax]dsl.*\..*\..*/i                            553 AUTO_XDSL spam
/client.*\..*\..*/i                             553 AUTO_CLIENT spam
/cable.*\..*\..*/i                              553 AUTO_CABLE spam
/pool.*\..*\..*/i                               553 AUTO_POOL spam
/dial.*\..*\..*/i                               553 AUTO_DIAL spam
/ppp.*\..*\..*/i                                553 AUTO_PPP spam
/dslam.*\..*\..*/i                              553 AUTO_DSLAM spam
/node.*\..*\..*/i                               553 AUTO_NODE spam
/([0-9]*-){3}[0-9]*(\..*){2,}/i                 553 SPAM_ip-add-rr-ess_networks
/([0-9]*\.){4}(.*\.){3,}.*/i                    553 SPAM_ip-add-rr-ess_networks
/.*\.pppool\..*/i                               553 SPAM_POOL
/[0-9]*-[0-9]*-[0-9]*-[0-9]*-tami\.tami\.pl/i   553 SPAM_POOL
/pool-[0-9]*-[0-9]*-[0-9]*-[0-9]*\..*/i         553 SPAM_POOL
/.*-[0-9]*-[0-9]*-[0-9]*-[0-9]*\.gtel.net.mx/i  553 SPAM_POOL
/dhcp.*\..*\..*/i                               553 SPAM_DHCP
```

```
# cat mx_access
```

```
127.0.0.1       DUNNO
127.0.0.2       550 Domains not registered properly
0.0.0.0/8       REJECT Domain MX in broadcast network
10.0.0.0/8      REJECT Domain MX in RFC 1918 private network
127.0.0.0/8     REJECT Domain MX in loopback network
169.254.0.0/16 REJECT Domain MX in link local network
172.16.0.0/12  REJECT Domain MX in RFC 1918 private network
192.0.2.0/24   REJECT Domain MX in TEST-NET network
192.168.0.0/16 REJECT Domain MX in RFC 1918 private network
224.0.0.0/4    REJECT Domain MX in class D multicast network
240.0.0.0/5    REJECT Domain MX in class E reserved network
248.0.0.0/5    REJECT Domain MX in reserved network
```

По сути файлы белых и черных списков не отличаются друг от друга. Можно использовать только один файл и в нем в каждой отдельной строке указывать либо запрет, либо разрешение. Я разделил просто для удобства восприятия и редактирования. Возможно вам будет удобнее с одним файлом.

После редактирования файлов обязательно выполняем команду на перестроение базы данных. Я перестрою сразу все файлы:

```
cd /etc/postfix/lists && postmap white_client_ip black_client_ip white_client black_client white_helo block_dsl mx_access
```

Еще упомяну о таком популярном явлении в спамерских письмах, как подделка адреса отправителя. Причем не просто подделка на абы кого, а именно на ваше имя домена. Пользователь получает спам письмо и в почтовом клиенте видит, что оно отправлено с вашего домена. Только по заголовкам можно определить реального отправителя. Такой подход позволяет обходить некоторые антиспам системы, которые не фильтруют письма внутреннего домена. Бороться с подменой адреса отправителя в нашем случае очень просто. Для этого в black_client добавим следующую запись:

```
zeroxzed.ru 554 Stop spam from my name
```

Отправка с нашего домена осуществляется только с SASL авторизацией, а она во всех ограничениях стоит первой, поэтому письма реальных пользователей без проблем будут уходить. А всех тех, кто только притворяется нашим доменом, мы будет отфильтровывать этим правилом.

Приведу в завершении описания методов борьбы со спамом простой пример. Добавим в black_client почтовый адрес и отправим с него письмо.

```
# cat black_client
zeroxzed@gmail.com REJECT Your e-mail was banned!
```

```
# postmap black_client
```

Отправляем сообщение и проверяем почтовый лог.

```
# cat /var/log/maillog
```

```
Mar 20 02:21:34 mail postfix/smtpd[10816]: connect from mail-yw0-f177.google.com[209.85.161.177]
Mar 20 02:21:35 mail postfix/smtpd[10816]: Anonymous TLS connection established from mail-yw0-
f177.google.com[209.85.161.177]: TLSv1.2 with cipher ECDHE-RSA-AES128-GCM-SHA256 (128/128 bits)
Mar 20 02:21:35 mail postfix/smtpd[10816]: NOQUEUE: reject: RCPT from mail-yw0-f177.google.com[209.85.161.177]: 554 5.7.1
<zeroxzed@gmail.com>: Sender address rejected: Your e-mail was banned!; from=<zeroxzed@gmail.com> to=<root@zeroxzed.ru>
proto=ESMTP helo=
Mar 20 02:21:35 mail postfix/smtpd[10816]: disconnect from mail-yw0-f177.google.com[209.85.161.177]
```

Вот и результат. На этом по борьбе со спамом все.

Заключение

Проверить настроенный почтовый сервер можно с помощью онлайн сервиса https://www.mail-tester.com. Не факт, что получите максимальный бал, но все недочеты будут указаны. Критичное нужно исправить (например, если обратная зона неправильная), некритичное можно пропустить (если dkim, к примеру, не настраивали).

Кажется все написал, что знал по поводу почтового сервера на linux в небольших и средних организациях. У некоторых может возникнуть вопрос, а зачем свой почтовый сервер? Почему бы не воспользоваться средствами корпоративной почты, которую представляют популярные почтовые сервисы бесплатно? Я планирую написать по этому поводу отдельную заметку (в итоге написал — выбор почтового сервера). У меня тоже есть определенный опыт на этот счет. И если некоторое время назад я считал, что свои почтовые серверы в небольших организациях уже не актуальны, то сейчас я так не думаю, поэтому и появилась эта статья.

Буду рад замечаниям по делу и советам в комментариях. Напоминаю, что данная статья является частью единого цикла статьей про сервер Centos.