Полноценный почтовый сервер с iRedMail на Ubuntu или Debian

dmosk.ru/instruktions.php

Дмитрий Моск

Обновлено: 07.06.2024 Опубликовано: 13.05.2017
Тематические термины: iRedMail, Ubuntu, Postfix, Dovecot, Roundcube, POP3, IMAP, PTR, SPF, DKIM

В нашей инструкции будет выполнено развертывание полноценного почтового сервера на базе сборки iRedMail. После ее установки мы получим систему со следующими возможностями:

- Хранение данных в СУБД.
- Защита от СПАМа и вирусов.
- Подключение к ящикам по POP3 и IMAP (Dovecot).
- Возможность работать с почтой удаленно в браузере.
- Поддержка виртуальных доменов и почтовых ящиков.
- Управление почтовыми ящиками с помощью веб-интерфейса.
- Шифрование при передаче сообщений (STARTTLS, TLS, SSL).
- Защита сообщений от попадания в СПАМ (корректная настройкай DNS).

Подготовка сервера

Задаем правильное имя сервера:

hostnamectl set-hostname mail.dmosk.ru

** имя сервера должно быть в формате FQDN, в противном случае мы получим ошибку* ***<< ERROR >> Please configure a fully qualified domain name (FQDN) in /etc/hosts before we go further.***

Заданное имя сервера должно разрешаться в IP-адрес через DNS. Если на момент установки это невозможно, создадим запись в файле hosts:

vi /etc/hosts

127.0.0.1 mail.dmosk.ru mail

** очень важно, чтобы имя FQDN было первым.*

Останавливаем веб-сервер apache (в данном примере будет использоваться nginx):

systemctl stop apache2

systemctl disable apache2

** если не остановить apache и попытаться установить nginx, мы получим ошибку* ***Errors were encountered while processing: nginx-full****.*

Если в нашей системе настроен брандмауэр, мы должны открыть следующие порты:

iptables -I INPUT -p tcp --match multiport --dports 80,443 -j ACCEPT

iptables -I INPUT -p tcp --match multiport --dports 25,465,587 -j ACCEPT

iptables -I INPUT -p tcp --match multiport --dports 110,143,993,995 -j ACCEPT

* где мы откроем следующие порты:

- **25** — стандартный SMTP (без шифрования или через STARTTLS);
- **80** — HTTP для порталов iRedAdmin и Roundcube;
- **110** — стандартный POP3 (без шифрования или через STARTTLS);
- **143** — стандартный IMAP (без шифрования или через STARTTLS);
- **443** — защищенный HTTPS для порталов iRedAdmin и Roundcube;
- **465** — защищенный SMTP через SSL/TLS;
- **587** — защищенный SMTP через STARTTLS;
- **993** — защищенный IMAP через SSL/TLS;
- **995** — защищенный POP3 через SSL/TLS.

Для сохранения правил установим утилиту iptables-persistent:

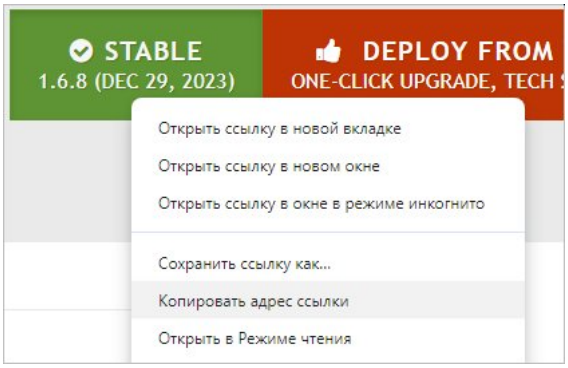
apt install iptables-persistent

И запустим ее:

netfilter-persistent save

Установка iRedMail

Заходим на страницу iredmail.org/download.html и копируем ссылку на скачивание последней версии почтового сервера:



Теперь используем ссылку для загрузки дистрибутива на сервере:

wget -O iredmail.tar.gz <скопированная ссылка>

Например:

wget -O iredmail.tar.gz https://github.com/iredmail/iRedMail/archive/refs/tags/1.6.8.tar.gz

И распаковываем скачанный архив:

tar -zxf iredmail.tar.gz

Переходим в каталог с распакованным установщиком:

cd iRedMail-*/

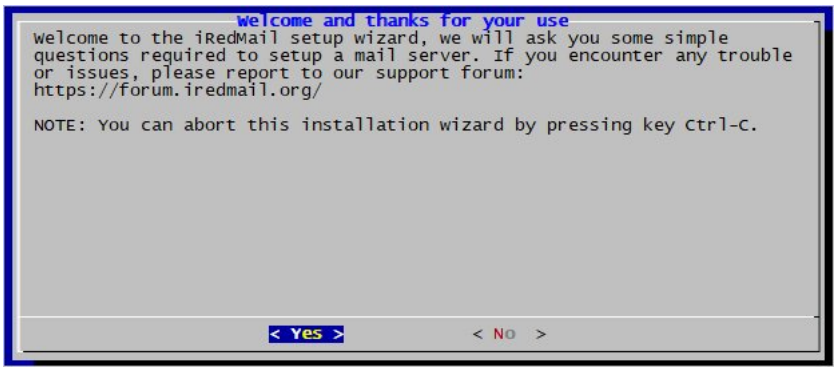
И запускаем скрипт установки:

bash iRedMail.sh

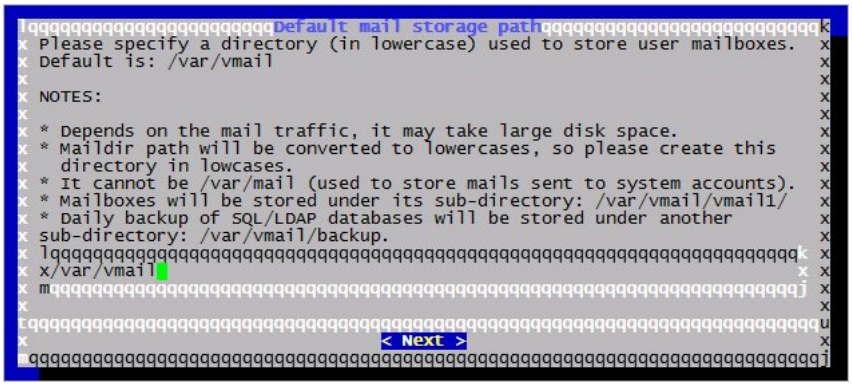
> Если мы хотим установить устаревшую версию iRedMail, мы получим ошибку:
>
> Your iRedMail version (x.x.x) is out of date, please
> ...
>
> Решение описано внизу.

На все запросы отвечаем **Enter**.

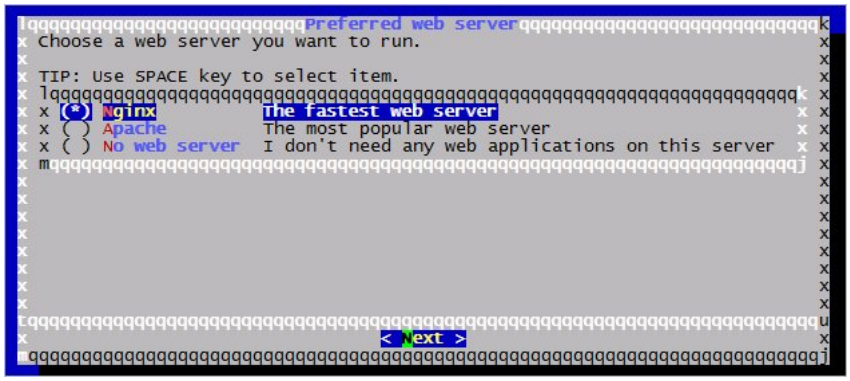
Запустится мастер настроек. В первом окне с приветствием ответьте **Yes**:



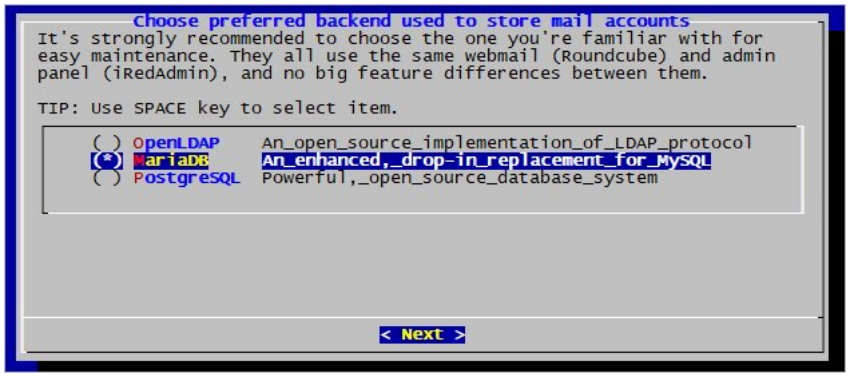
В окне **Default mail storage path** оставляем **/var/vmail** и задаем свой путь для хранения сообщений:



В следующем окне **Preferred web server** желательно оставить **Nginx**:

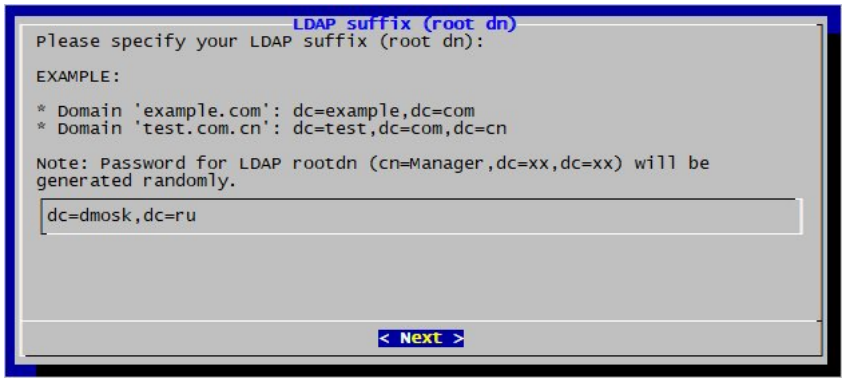


В окне **Choose preferred backend used to store mail accounts** выбираем **Mariadb** или OpenLDAP:

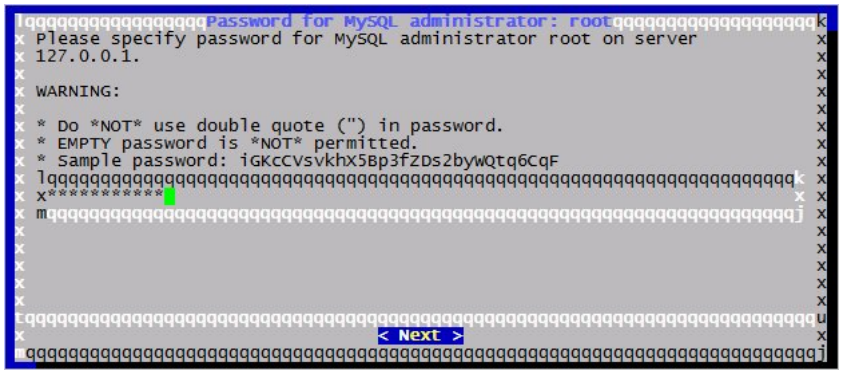


** в нашем примере мы выбираем MariaDB.*

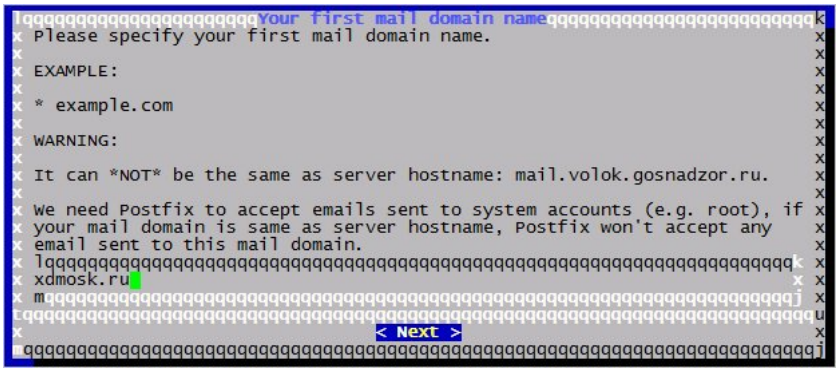
> Если мы выберем на предыдущем шаге OpenLDAP, система потребует ввести домен:



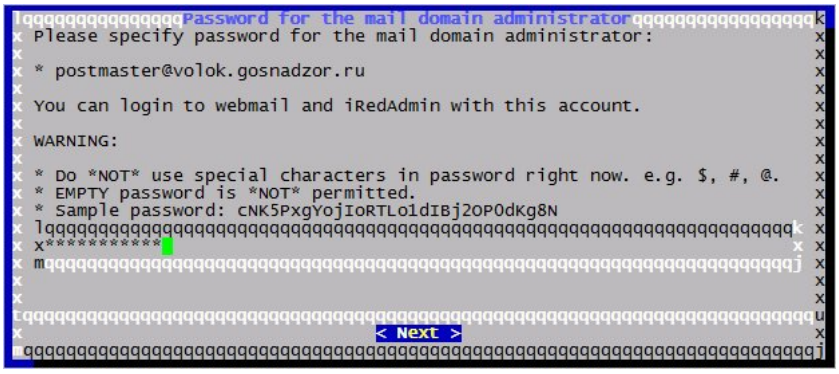
И задаем пароль для пользователя СУБД:



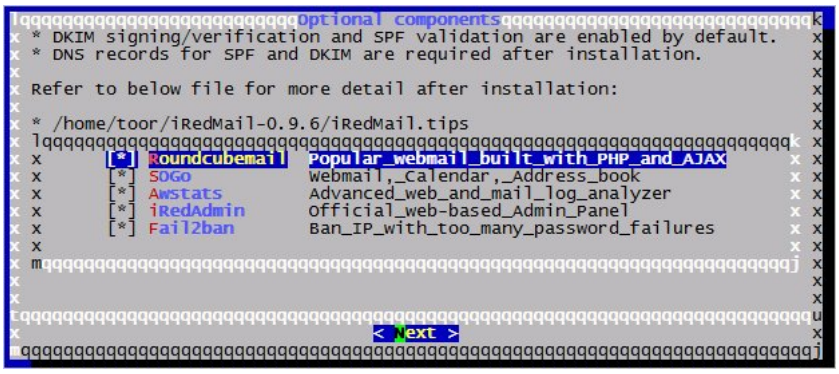
На следующем шаге вводим наш первый почтовый домен:



Теперь вводим пароль для управления почтовыми ящиками:



В окне **Optional components** выбираем все доступные компоненты:



В самом конце вводим **Y**, чтобы подтвердить введенные настройки.

Начнется установка почтового сервера. В зависимости от производительности, процесс может занять от 10 до 20 минут. В конце система предложит активировать брандмауэр — соглашаемся вводом **Y**.

> Если мы получим ошибку:
>
> The following packages have unmet dependencies:
> mariadb-server : Depends: mariadb-server-xxx...
>
> То значит, что в репозитории системы нет нужной версии MariaDB. Для подключения переходим по ссылке https://mariadb.org/download/?t=repo-config и вводим данные своей системы в нужную версию системы и СУБД. Мы получим набор команд для настройки репозитория.
>
> Например, для Ubuntu 20.04 и mariadb версии 10.3 вводим команды:
>
> apt install apt-transport-https curl
>
> curl -o /etc/apt/trusted.gpg.d/mariadb_release_signing_key.asc 'https://mariadb.org/mariadb_release_signing_key.asc'
>
> echo 'deb https://mirrors.xtom.ee/mariadb/repo/10.3/ubuntu focal main' >>/etc/apt/sources.list
>
> Обновим кэш репозиториев:
>
> apt update
>
> И снова запускаем установку iRedMail:
>
> bash iRedMail.sh
>
> На вопрос о продолжении использовать ранее введенные настройки отвечаем утвердительно — Y.

После завершения, установщик даст подсказку, что необходимо перезапустить сервер для начала работы всех компонентов. Выполним перезагрузку:

shutdown -r now

Настройка iRedMail

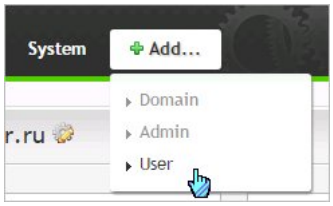
Создание пользователя

Открываем браузер и в адресной строке вводим **https://IP-адрес_сервера/iredadmin/**

Откроется страница входа в панель управления. Вводим логин **postmaster@dmosk.ru** и пароль (пароль и домен **dmosk.ru** — данные, которые мы вводили при установке iRedMail).

Создадим первого пользователя. Для этого переходим по **Add** - **User**:

Заполняем поля и создаем пользователя:



Отключение Graylisting

Graylisting — мощное оружие против СПАМа, но с существенным минусом — все входящие сообщения, отправленные с определенного домена, в первый раз, будут приходить с задержкой. На момент тестирования это создает массу неудобств.

Для отключения серого списка, добавляем права на редактирование следующему файлу:

chmod u+w /opt/iredapd/settings.py

После открываем его:

vi /opt/iredapd/settings.py

Находим перечисление плагинов:

plugins = ["reject_null_sender", "reject_sender_login_mismatch", "greylisting", "throttle", "amavisd_wblist", "sql_alias_access_policy"]

И вырезаем greylisting.

Возвращаем права:

chmod u-w /opt/iredapd/settings.py

Перезагружаем iredapd:

systemctl restart iredapd

Проверяем работу сервера

Для проверки сервера можно выполнить тестовую отправку и получения писем.

Отправка

Открываем браузер и в адресной строке вводим **https://IP-адрес_сервера/mail/**

Откроется панель для работы с почтой — вводим логин и пароль от созданного пользователя (логин должен быть с доменом, в нашем примере, **test1@dmosk.ru**)

Нажимаем **Написать сообщение** и отправляем тестовое сообщение на один из своих адресов:

Получение

Для возможности получать письма, необходимо прописать в DNS для нашего домена запись типа MX.

Пример такой записи:

MX 10 mail.dmosk.ru

** где MX — тип; 10 — приоритет (таких записей может быть несколько); mail.dmosk.ru — имя нашего почтового сервера (на данное имя необходима также запись типа A).*

После создания такой записи необходимо подождать от 1 до 8 часов, так как настройки DNS могут применяться не сразу.

Установка сертификата

Вместе с iRedMail создается самоподписный сертификат, которому по умолчанию, не доверяют другие системы. Если мы хотим, чтобы пользователи не видели предупреждений об использовании потенциально не безопасного сертификата, можно установить последний, выданный акредитованным центром сертификации. Мы же рассмотрим, как получить для iRedMail бесплатный сертификат от Let's Encrypt (подробнее в статье получение сертификата от Let's Encrypt).

Откроем на редактирование файл:

vi /etc/nginx/sites-enabled/00-default-ssl.conf

... и добавим в секцию server (отмечено желтым):

```
server {
    ...

    root /var/www/html;
    index index.php index.html;

    location ~ /.well-known {
        root /usr/share/nginx/html;
        allow all;
    }

    include ...

    ...
}
```

Перечитаем конфиг nginx

systemctl reload nginx

Устанавливаем утилиту для получения сертификата:

apt install certbot

Для нашего удобства, создаем переменную с нашим хостом, для которого мы будем получать сертификат:

DOMAIN=$(hostname)

** данная команда в качестве значения задаст имя нашего сервера.*

И получаем сертификат командой:

certbot certonly --webroot --agree-tos --email postmaster@dmosk.ru --webroot-path /usr/share/nginx/html/ -d $DOMAIN

** подробнее параметры описаны в статье получение сертификата от Let's Encrypt. Обратите внимание, что в данном примере мы получим сертификат для узла **mail.dmosk.ru**.*

При успешном завершении команды, мы получим сообщение на подобие:

```
Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/mail.dmosk.ru/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/mail.dmosk.ru/privkey.pem
```

Удаляем старые сертификаты:

rm -f /etc/ssl/private/iRedMail.key

rm -f /etc/ssl/certs/iRedMail.crt

И создаем симлинки на полученные:

ln -s /etc/letsencrypt/live/$DOMAIN/fullchain.pem /etc/ssl/certs/iRedMail.crt

ln -s /etc/letsencrypt/live/$DOMAIN/privkey.pem /etc/ssl/private/iRedMail.key

** **cert.pem** и **iRedMail.crt** — открытые ключи (public); **privkey.pem** и **iRedMail.key** — закрытые (private); **$DOMAIN** — переменная, которая содержит наш домен.*

Перезапускаем службы nginx, postfix и dovecot:

systemctl reload nginx postfix dovecot

Для автоматического продления сертификата создаем в cron задачу:

crontab -e

Добавим:

0 0 * * 1,4 /usr/bin/certbot renew --noninteractive

После автоматического продления, добавим настройку для перезапуска сервисов:

vi /etc/letsencrypt/cli.ini

deploy-hook = systemctl reload nginx postfix dovecot

Настройка Clamav

Если наш сервер находится за российскими IP-адресами, то clam не будет обновляться, так как настроенные по умолчанию заркала заблокированы. Выполняем следующие настройки.

Открываем файл:

vi /etc/clamav/freshclam.conf

Комментируем строки, которые начинаются на ##DatabaseMirror

##DatabaseMirror ...
##DatabaseMirror ...

Добавляем строки:

PrivateMirror https://clamav-mirror.ru/
PrivateMirror http://mirror.truenetwork.ru/clamav/
ScriptedUpdates no

Останавливаем службу clamav-freshclam:

systemctl stop clamav-freshclam

Удаляем старую информацию об обновлениях:

rm -f /var/lib/clamav/freshclam.dat

Запускаем обновление:

freshclam

Ждем обновления, после запускаем службу clamav-freshclam и перезапускаем clamav-daemon, amavis:

systemctl start clamav-freshclam

systemctl restart clamav-daemon amavis

Защищаем сообщения от попадания в СПАМ

Чтобы другие почтовые системы не принимали наши письма за СПАМ, выполняем следующие рекомендации:

A-запись в DNS

Для FQDN-имени почтового сервера должна быть создана A-запись в DNS. Пример записи:

mail.dmosk.ru A 90.156.242.197

Создаем PTR-запись для внешнего IP-адреса

Она должна вести на имя сервера (в данном примере, mail.dmosk.ru). Чтобы создать такую запись, нужно написать обращение Интернет-провайдеру или хостеру виртуальной машины. Пример записи:

171.23.222.83.in-addr.arpa name = mail.dmosk.ru

** данная запись соответствует IP-адресу **83.222.23.171**.*

Добавляем SPF-запись для домена

Эта запись создается в DNS для домена, от которого идет отправка сообщений. Пример:

dmosk.ru text = "v=spf1 a mx -all"

Прописываем DKIM в DNS

Для начала, смотрим ключ, который был сформирован во время установки iRedMail:

amavisd-new showkeys

Пример ответа:

```
dkim._domainkey.dmosk.ru.     3600 TXT (
  "v=DKIM1; p="
  "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDHNu0ZlYkq8pKsp131jnoZ+Ief"
  "zcSP1WxGzGQXssg3yiRGBlqsRGBnnKgitrsPYTZbzqqL+/rW0ptGNhAqfTWHvMia"
  "+f4RSMLJPMREFtakVEZvTIK5iZvxuCZpVhvM6ldadTLAxbcupX38yMfJV73EwCHK"
  "d2mdqfW+emSW/paUwQIDAQAB")
```

Копируем DKIM и создаем в DNS запись TXT. Пример:

```
dmosk.ru     text = "v=DKIM1;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDHNu0ZlYkq8pKsp131jnoZ+IefzcSP1WxGzGQXssg3yiRGBlqsRGBnnKgitrsPYTZb
```

Создать другую подпись DKIM

Для удобства, создаем переменную с именем домена:

```
DOMAIN=dmosk2.ru
```

** где **dmosk2.ru** — новый домен, для которого мы сгенерируем подпись dkim.*

Генерируем новый ключ:

```
amavisd-new genrsa /var/lib/dkim/$DOMAIN.pem 1024
```

** некоторые системы не работают с ключами более чем 1024 бит.*

Задаем права на созданный файл:

```
chown amavis:amavis /var/lib/dkim/$DOMAIN.pem
```

```
chmod 0400 /var/lib/dkim/$DOMAIN.pem
```

Открываем конфигурационный файл amavisd

```
vi /etc/amavis/conf.d/50-user
```

Находим строчку:

```
dkim_key('dmosk.ru', "dkim", "/var/lib/dkim/dmosk.ru.pem");
```

** где **dmosk.ru —** наш первый домен, с которым мы развернули сервер.*

И добавляем радом с ней новую. Получится так:

```
dkim_key('dmosk.ru', "dkim", "/var/lib/dkim/dmosk.ru.pem");
dkim_key('dmosk2.ru', "dkim", "/var/lib/dkim/dmosk2.ru.pem");
```

Теперь находим строчку:

```
@dkim_signature_options_bysender_maps = ( {
  ...
  "dmosk.ru" => { d => "dmosk.ru", a => 'rsa-sha256', ttl => 10*24*3600 },
```

И также после нее добавляем новую. Должно получиться:

```
@dkim_signature_options_bysender_maps = ( {
  ...
  "dmosk.ru" => { d => "dmosk.ru", a => 'rsa-sha256', ttl => 10*24*3600 },
  "dmosk2.ru" => { d => "dmosk2.ru", a => 'rsa-sha256', ttl => 10*24*3600 },
```

Перезапускаем amavisd:

```
amavisd-new restart
```

Посмотреть новый ключ можно командой:

```
amavisd-new showkeys $DOMAIN
```

Политика DMARC

Данная политика определяет, что делать с письмом, которое не проходит проверку. Подробнее о DMARC.

Для создания данной политики необходимо в DNS добавить TXT запись, примерно, такого содержания:

_dmarc.dmosk.ru. 3600 IN TXT "v=DMARC1; p=quarantine; sp=none; pct=100; fo=0; rua=mailto:postmaster@dmosk.ru"

** данная запись означает, что все письма, которые не прошли проверку, необходимо отправить в карантин, а отчет написать на ящик **postmaster@dmosk.ru**.*

Ящик abuse

По аналогии с тем, как мы создавали тестовую учетную запись, необходимо создать ящик abuse@... На данный ящик могут приходить жалобы на СПАМ. Стоит время от времени просматривать его (или настроить переадресацию), и реагировать на жалобы.

Управление белыми и черными списками

Переходим в каталог с утилитами iredmail:

cd /opt/iredapd/tools/

Просмотреть содержимое белого и черного списков:

python3 wblist_admin.py --list --whitelist

python3 wblist_admin.py --list --blacklist

Добавить в списки:

python3 wblist_admin.py --add --whitelist 111.112.113.114 info@domain.ru @dmosk.ru @.dmosk.ru

python3 wblist_admin.py --add --blacklist 111.112.113.115 @baddomain.com

** Первая команда добавит в белый список адрес **111.112.113.114**, email **info@domain.ru** и домен **dmosk.ru** со всеми поддоменами. Вторая команда добавит в черный список адрес **111.112.113.115** и домен **baddomain.com.***

Удалить из списка:

python3 wblist_admin.py --delete --whitelist 111.112.113.114 info@domain.ru @dmosk.ru @.dmosk.ru

python3 wblist_admin.py --delete --blacklist 111.112.113.115 @baddomain.com

** как видим, процесс удаления аналогичен — просто меняем **--add** на **--delete**.*

Отключение антивируса и антиспама

Отключить защиту для почты может понадобиться при различных обстоятельствах, например:

1. Для диагностики проблем отправки сообщений.
2. Экономии ресурсов (антивирус может слишком много потреблять ресурсов).
3. При отсутствии необходимости.

И так, для отключения amavis (clam + spamassassin) открываем файл:

vi /etc/amavis/conf.d/50-user

Приводим к следующему виду настройку:

```
@bypass_virus_checks_maps = (1);
@bypass_spam_checks_maps  = (1);
```

** в данном примере мы сняли комментарий с данных строк (если они были закомментированы) и задаем для них значение 1. Опция **bypass_virus_checks_maps** отвечает за включение проверки писем на вирусы; **bypass_spam_checks_maps** — на СПАМ.*

Перезапускаем службу amavis:

systemctl restart amavis

После данной настройки письма будут отправляться без проверок. Однако, сервис антивируса будет, по-прежнему, работать.

Останавливаем и отключаем сервис clamd:

systemctl disable clamav-daemon

systemctl stop clamav-daemon

Разрешить соединение без STARTTLS

После установки iRedMail, система будет требовать от клиента безопасного соединения по TLS. При необходимости, можно отключить данную возможность.

Отключение для SMTP

Открываем конфигурационный файл postfix:

vi /etc/postfix/main.cf

Задаем следующие настройки:

```
...
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
#smtpd_tls_auth_only = yes
...
```

** где **smtpd_sasl_auth_enable** разрешает или запрещает аутентификацию; **smtpd_sasl_security_options** — дополнительные опции для аутентификации; **smtpd_tls_auth_only** — разрешает соединение SMTP только по TLS. В данном примере мы разрешаем аутентификацию, запрещаем анонимные соединения и комментируем опцию, которая требует только безопасного соединения.*

Перезапускаем postfix:

systemctl restart postfix

Отключение для IMAP/POP3

Открываем конфигурационный файл dovecot:

vi /etc/dovecot/dovecot.conf

Задаем следующие настройки:

```
ssl = yes
disable_plaintext_auth = no
```

** где **disable_plaintext_auth** запрещает аутентификацию без защиты; **ssl** задает опцию защиты (в данном примере, разрешить, но не требовать).*

Перезапускаем dovecot:

systemctl restart dovecot

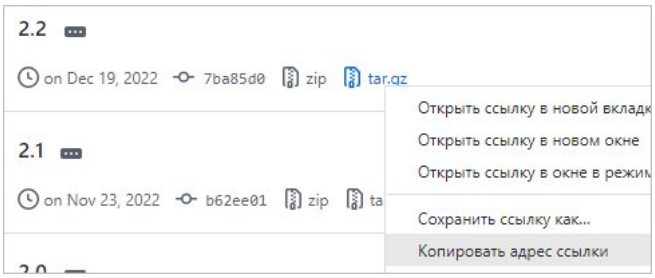
Обновление iRedAdmin

При выходе новой версии интерфейса iRedAdmin мы можем получить соответствующее уведомление в консоли управления. Процедура обновления не сложная.

Переходим в каталог для хранения исходников:

cd /usr/local/src/

Для начала зайдем на страницу https://github.com/iredmail/iRedAdmin/tags и скопируем ссылку на последнюю версию пакета:



Используя скопированную ссылку, скачиваем архив:

wget https://github.com/iredmail/iRedAdmin/archive/refs/tags/2.2.tar.gz

Распаковываем его:

tar -zxf 2.2.tar.gz

** имя файла зависит от версии iRedAdmin.*

Переходим в скачанный каталог:

cd iRedAdmin-*/tools/

Запускаем скрипт для обновления:

bash upgrade_iredadmin.sh

Сброс пароля

В бесплатной версии iredmail сброс паролей выполняется напрямую в СУБД или каталоге LDAP. Подробнее процедура описана на официальном сайте. Мы же рассмотрим пример работы с базой MySQL.

Создаем хэш для пароля:

doveadm pw -s 'ssha512'

Система попросит нас ввести дважды пароль — вводим тот, что хотим использовать. В итоге, мы получим хэш, например:

{SSHA512}Y9brnGEeFeLU3OAdzzv4RASAf4i53yugvrtJtrI7slkcIFUyqzO9uQ+FVxCJUm0Y5+ov/B6w4tfalBJnLZq5Tie10no=

Подключаемся к базе:

mysql -uroot -p

Выбираем базу:

> USE vmail

Смотрим пользователей:

> select username, isadmin, isglobaladmin from mailbox;

Сбрасываем пароль для пользователя с правами администратора, например:

> UPDATE mailbox SET
password='{SSHA512}Y9brnGEeFeLU3OAdzzv4RASAf4i53yugvrtJtrI7slkcIFUyqzO9uQ+FVxCJUm0Y5+ov/B6w4tfalBJnLZq5Tie10no='
WHERE username='postmaster@dmosk.ru';

** в нашем примере, для **postmaster@dmosk.ru**.*

Дополнительные настройки

Рассмотрим дополнительные настройки, которые могут нам пригодиться.

Настройка лимита на объем вложения

По умолчанию, допустимый размер отправляемого вложения, отправленного через iRedMail может быть размером не больше 15 Мб. Для увеличения этого порога вводим команду:

postconf -e "message_size_limit = 52428800"

** в данном примере выставлен лимит в 50 мб.*

Квота по умолчанию

При создании новых пользователей веб-интерфейс подставляет значение для квоты. Если значение по умолчанию для данной опции нужно изменить, настройку придется сделать в базе данных.

Подключаемся к sql-оболочке (в нашем примере это MySQL):

mysql -uroot -p

Используем базу vmail:

> use vmail

Смотрим настройки для нашего домена:

> SELECT domain, settings FROM domain WHERE domain='dmosk.ru';

** на сервере может быть настроено много доменов. У каждого могут быть свои настройки. Мы смотрим настройки для домена **dmosk.ru**.*

В моем случае ответ был такой:

```
+------------+-------------------------+
| domain     | settings                |
+------------+-------------------------+
| dmosk.ru   | default_user_quota:1024; |
+------------+-------------------------+
```

Это значит, что для домена **dmosk.ru** есть только одна настройка **default_user_quota**, которая задает значение квоты 1024 МБ.

Настроек может быть несколько, копируем содержимое поля settings и меняем в нем необходимые значения (у нашем случае мы должны заменить 1024 для default_user_quota). Получается такой запрос:

```
> UPDATE domain SET settings='default_user_quota:2048' WHERE domain='dmosk.ru';
```

** в данном примере мы указываем размер квоты для создаваемого пользователя в 2 Гб.*

Язык по умолчанию

При создании новых пользователей веб-интерфейс подставляет значение для используемого языка. Данное значение по умолчанию можно изменить в базе данных.

Подключаемся к sql-оболочке (в нашем примере это MySQL):

```
mysql -uroot -p
```

Используем базу vmail:

```
> use vmail
```

Смотрим настройки для нашего домена:

```
> SELECT domain, settings FROM domain WHERE domain='dmosk.ru';
```

** на сервере может быть настроено много доменов. У каждого могут быть свои настройки. Мы смотрим настройки для домена **dmosk.ru**.*

В моем случае ответ был такой:

```
+------------+-------------------------+
| domain     | settings                |
+------------+-------------------------+
| dmosk.ru   | default_user_quota:2048; |
+------------+-------------------------+
```

** на предыдущем шаге мы меняли настройку для квот.*

Параметры должны перечисляться через точку с запятой. Для настройки языка по умолчанию нужна опция default_language. Итого, запрос будет таким:

```
> UPDATE domain SET settings='default_user_quota:0;default_language:ru_RU;' WHERE domain = 'dmosk.ru';
```

** в нашем примере для домена будет указано 2 настройки — квота и язык по умолчанию.*

Использование нестандартных портов

После установки iRedMail мы можем обнаружить, что нестандартные порты для входящих запросов могут быть заблокированы, при этом:

- Нужный порт разрешен с помощью iptables.
- tcpdump видит, что запрос приходит.

Дело в том, что для фильтрации сетевых пакетов используется подсистема nftables. Для настройки открываем файл:

```
vi /etc/nftables.conf
```

И в секцию table inet filter добавим, например:

```
...
# ssh
tcp dport 2222 accept
...
```

** в данном примере мы хотим использовать SSH на порту **2222**.*

Применяем настройку:

nft flush ruleset

Аналоги iRedMail

Если есть причины, по которым iRedMail не подходит, можно рассмотреть другие варианты:

1. iRedMail Pro. Платная версия настроенного в данной инструкции программного обеспечения. Обладает расширенными настройками и возможностями.

2. Ручная сборка компонентов для почтового сервера. Пример данной настройки описан в статье Почтовый сервер Postfix на CentOS 7 с виртуальными доменами, системой управления, веб-доступом и многим другим.

3. Microsoft Exchange Server (на Windows). Платный сервер, работает в среде Active Directory.

4. Zimbra (на Linux). Готовая сборка. Только платная версия. Пример установки и настройки в инструкции Установка и настройка Zimbra на Linux.

5. Carbonio (на Linux). Готовая сборка. Есть платная и бесплатная версии. Пример установки и настройки в инструкции Установка и настройка Carbonio CE на Linux Ubuntu.

Возможные проблемы

Рассмотрим ошибки, с которыми приходилось столкнуться мне.

1. Ошибка обновления clamav с помощью freshclam

Антивирус clamav не обновляется. В логе мы можем увидеть сообщение о блокировке подключения к базе обновлений.

Причина: блокировка доступа к антивирусным базам с IP-адресов в России.

Решение: меняем источник баз. Открываем файл:

vi /etc/clamav/freshclam.conf

Комментируем строки, которые начинаются на DatabaseMirror

##DatabaseMirror ...
##DatabaseMirror ...

Добавляем строки:

PrivateMirror https://clamav-mirror.ru/
PrivateMirror http://mirror.truenetwork.ru/clamav/
ScriptedUpdates no

Останавливаем службу clamav-freshclam:

systemctl stop clamav-freshclam

Удаляем старую информацию об обновлениях:

rm -f /var/lib/clamav/freshclam.dat

Запускаем обновление:

freshclam

Ждем обновления, после запускаем службу clamav-freshclam и перезапускаем clamav-daemon с amavis:

systemctl start clamav-freshclam

systemctl restart clamav-daemon amavis

2. Your iRedMail version (x.x.x) is out of date

При попытке установить старую версию iRedMail, мы можем увидеть ошибку:

<< ERROR >> Your iRedMail version (x.x.x) is out of date, please
<< ERROR >> download the latest version and try again:
<< ERROR >> http://www.iredmail.org/download.html

Причина: разработчик iRedMail решил технически ограничивать возможность установки старой версии.

Решение: для установки устаревшей версии нужно использовать системную переменную. Запуск установщика будет выполняться командой:

CHECK_NEW_IREDMAIL=NO bash ./iRedMail.sh

Читайте также

Другая информация по iRedMail:

1. Установка и настройка iRedMail на CentOS.
2. Резервное копирование и восстановление iRedMail.
3. Интеграция iRedMail с Microsoft Active Directory.
4. Обновление почтового сервера iRedMail.