OpenVPN Настройка сервера

Зачем 100500 статейка на одну и ту же тему? Просто для себя!

Задача:

- Сервер OpenVPN на Ubuntu (настройки пойдут на любом другом дистрибутиве, вся разница только в установке пакетов)
- Разумные (но не параноидальные) настройки безопасности, больше информации можно найти тут: https://blog.securityevaluators.com/hardening-openvpn-in-2020-1672c3c4135a
- Скорость работы через соединение должна быть приемлемой. Чем сильнее затянуты гайки в плане безопасности - тем неудобнее работать

Что будет в итоге:

- Авторизация по сертификатам
- Согласование ключей на основе эллиптических кривых ECDH (Elliptic Curve Diffie-Hellmann), а не на привычных DH. Что это и чем лучше можно посмотреть тут: https://habr.com/ru/post/335906/
- TSL-криптование соединения
- Списки отзыва сертификатов
- Пуш клиентам необходимых настроек

Что нужно:

- Сервер с операционной системой Linux (в моем случае это Ubuntu)
- Белый IP - в примере это XXX.XXX.XXX.XXX
- Крайне желательна DNS-запись для данного IP (но можно обойтись) - в примере это vpn.mysite.com

Установка OpenVPN

Подробно процесс установки описан тут: OpenVPN Установка в Ubuntu

Так же для создания сертификатов нам понадобится пакет easy-rsa:

```
sudo apt install easy-rsa
```

Примечание: нижеследующее описание для easy-rsa 2.x, в версии 3.x команды другие

Создание директории центра сертификации

OpenVPN использует сертификаты для шифрования трафика между сервером и клиентами. Для выпуска доверенных сертификатов нам потребуется создать наш собственный центр сертификации.

Для начала скопируем шаблонную директорию easy-rsa в нашу домашнюю директорию с

помощью команды make-cadir и перейдем в нее:

```
make-cadir ~/openvpn-ca
cd ~/openvpn-ca
```

Настройка переменных центра сертификации

Настройка переменных нужна для ускорения и упрощения создания сертификатов.

Необходимо отредактировать файл vars и установить переменные как указано ниже:

```
...
export KEY_SIZE=2048 # Длина ключа. Чем больше, тем безопаснее, но и меньше производительность
...
export CA_EXPIRE=3650 # Срок жизни сертификата центра авторизации. 3650 - 10 лет
...
export KEY_EXPIRE=3650 # Срок жизни сертификатов пользователей
export KEY_COUNTRY="RU"  # Наименование страны
export KEY_PROVINCE="YourRegion"  # Название региона
export KEY_CITY="YourCity"  # Название города
export KEY_ORG="YourCompany"  # Название компании
export KEY_EMAIL="your@email.com"  # Адрес почты
export KEY_OU="YourOU"  # Наименование подразделения
export KEY_CN="CommonName" # Это имя сертификата и он меняется при создании или его нужно будет вводить руками
...
# X509 Subject Field
export KEY_NAME="vpn.mysite.com"  # Имя ключа, необязательно. Можно указать что угодно, но лучше или DNS-имя или белый IP
...
```

Как работать с сертификатами

При любых операциях с сертификатами желательно начинать работу с применения переменных из файла vars:

```
cd ~/openvpn-ca
source vars
```

Создание сертификата центра сертификации (CA)

Убедитесь, что вы находитесь в директории центра сертификации и применим настройки из файла vars:

```
cd ~/openvpn-ca
source vars
```

Вы должны увидеть следующий вывод:

```
NOTE: If you run ./clean-all, I will be doing a rm -rf on
/home/fireball/openvpn-ca/keys
```

Убедимся, что мы работаем в “чистой среде” выполнив следующую команду:

```
./clean-all
```

Теперь мы можем создать наш корневой центр сертификации командой:

```
./build-ca
```

Эта команда запустит процесс создания ключа и сертификата корневого центра сертификации. Поскольку мы задали все переменные в файле vars, все необходимые значения будут введены автоматически. Нажимайте ENTER для подтверждения выбора. По окончании процесса у нас появится сертификат центр сертификации (CA), который мы сможем использовать для создания всех остальных необходимых нам сертификатов.

Создание сертификатов сервера

Создаем сертификат, в примере я выбрал наименование ServerVPN

```
./build-key-server ServerVPN
```

Согласитесь со всеми значениями по умолчанию (кроме Common Name, оно должно быть ServerVPN), нажимая ENTER. Не задавайте challenge password. В конце процесса два раза введите «Y» для подписи и подтверждения создания сертификата.

Ключ протокола Диффи-Хеллмана, используемые при обмене ключами **НЕ ДЕЛАЮ**, т.к. будет использоваться согласование ключей на основе эллиптических кривых ECDH (Elliptic Curve Diffie-Hellmann).

Но если вдруг понадобится, вот команда:

```
./build-dh
```

Создаем подпись HMAC для проверки целостности TLS:

```
openvpn --genkey --secret keys/ta.key
```

В результате в папке keys должно быть несколько файлов, из которых 4 нужные для дальнейшей работы:

```
ca.crt - сертификат центра авторизации
```

```
ServerVPN.crt - сертификат сервера
ServerVPN.key - приватный ключ сервера
ta.key - ключ для проверки TLS
```

> **Внимание**
>
> **Важно:** все файлы с расширением key являются приватными и должны быть доступны только их владельцам

Создаем файл отзыва сертификатов

Файл отзыва нужен в ситуации когда какой-то сертификат «утек» и требуется запретить подключение с ним.

Для создания файла необходимо создать сертификат и тут же отозвать его.

Создаем, отвечая на все вопросы и в конце два раза «Y»:

```
./build-key empty-crl
```

Отзываем:

```
./revoke-full empty-crl
```

В папке keys появится файл crl.pem - нужно сделать на него символическую ссылку или каждый раз после отзыва сертификатов копировать в папку откуда его увидит OpenVPN.

Создаем сертификаты пользователей

Создаем сертификат пользователя с именем client1, отвечая на все вопросы и в конце два раза «Y»:

```
./build-key client1
```

Итак, все готово и вот какие файлы для подключения будут нам нужны:

- ca.crt
- client1.crt
- client1.key
- ta.key

Конфигурационный файл сервера OpenVPN

Копируем наши сертификаты в папку:

```
cd ~/openvpn-ca/keys
sudo cp ca.crt ServerVPN.crt ServerVPN.key ta.key /etc/openvpn
```

Создаем файл /etc/openvpn/server.conf:

```
port 1194 # Порт 1194
proto udp # Протокол UDP
dev tun # Тип интерфейса: TUN - L3, TAP - L2
ca /etc/openvpn/ca.crt # Сертификат ЦА (CA)
cert /etc/openvpn/ServerVPN.crt # Сертификат сервера
key /etc/openvpn/ServerVPN.key  # Ключ сервера
crl-verify /etc/openvpn/crl.pem # Список отозванных сертификатов
dh none # Не используем Diffie-Hellmann
ecdh-curve secp384r1 # Используем эллиптические кривые
topology subnet # Топология
server 10.8.0.0 255.255.255.0 # Адрес сервера и /24 подсеть
ifconfig-pool-persist ipp.txt # перед тем как выдать клиенту свободный адрес из пула сервер должен свериться с файлом ipp.txt в котором прописывается привязка имени пользователя к ip-адресу
client-config-dir /etc/openvpn/ccd # Директория для индивидуальных настроек пользователей (IP, маршруты и т.д.)
push "dhcp-option DNS 10.8.0.1" # Принудительная отправка DNS-сервера клиентам
client-to-client # Разрешить клиеентам подключаться друг к другу
float # позволять пользователям менять IP
keepalive 10 60 # Проверка таймаутов каждый 10 сек и переподключение через 60 секунд неактивности
persist-key # не перечитывать файлы ключей при перезапуске туннеля
persist-tun # оставлять без изменения устройства tun/tap при перезапуске OpenVPN
explicit-exit-notify 1 # Уведомлять клиентов о необходимости переподключения
tls-crypt /etc/openvpn/ta.key # Ключ TLS
tls-version-min 1.2 # Минимальная версия TLS
remote-cert-tls client # Требовать сертификат клиента
verify-client-cert require # Запретить подключения без сертификата
cipher AES-256-GCM # Протокол шифрования. AES-128-GCM менее безопасно, зато быстрее на 40% - решать вам
auth SHA256 # Протокол аутентификации
opt-verify #не разрешать подключение клиентов с отличающимися настройками
verify-client-cert require #не разрешать подключение без сертификатов
compress lz4-v2 # Режим сжатия
push "compress lz4-v2" # Уведомить клиента об используемом режиме сжатия
user nobody # Запуск процесса под непривелегированным пользователем
group nogroup # Запуск процесса под непривелегированным пользователем
status openvpn-status.log # путь к статус-файлу, в котором содержится информация о текущих соединениях и информация о интерфейсах TUN/TAP
verb 3 # уровень логирования
mute 20 # кол-во записей в каждой из категорий
```

Файлы индивидуальных настроек для определенных клиентов

Если требуется каким-то клиентам выдавать индивидуальные настройки, должно быть сделано:

- Должна существовать папка ccd (можно назвать как угодно), например /etc/openvpn/ccd
- Включена настройка client-config-dir /etc/openvpn/ccd в файле конфигурации
- В папке создан файл для конкретного клиента с его именем

Содержимое файла client1 для примера:

```
ifconfig-push 10.8.0.101 255.255.255.0
push "route 10.10.10.0 255.255.255.0"
```

Файл настроек клиента (*.ovpn)

Пример файла клиента:

```
client # Режим работы Клиент
proto udp # Протокол UDP
dev tun # Тип интерфейса: TUN - L3, TAP - L2
nobind # использовать динамический порт для подключени
remote vpn.mysite.com 1194 # Сервер к которому подключаемся и порт
resolv-retry infinite # Бесконечное кол-во попыток подключения
keepalive 10 60 # Проверка таймаутов каждый 10 сек и переподключение через 60 секунд неактивности
persist-key # не перечитывать файлы ключей при перезапуске туннеля
persist-tun # оставлять без изменения устройства tun/tap при перезапуске OpenVPN
cipher AES-256-GCM # Протокол шифрования. AES-128-GCM менее безопасно, зато быстрее на 40% - решать вам
auth SHA256 # Протокол аутентификации
tls-version-min 1.2 # Минимальная версия TLS
tls-client # Режим клиента TLS
remote-cert-tls server # Проверять TLS от сервера
verify-x509-name vpn.mysite.com name # Проверять кем выдан сертификат
comp-lzo # Уровень сжатия
float # позволять менять IP
verb 3 # Уровель логирования
auth-nocache # Не кэшировать аутентификационные данные
<ca>
=== Сертификат центра авторизации (ca.crt) ===
</ca>
<cert>
=== Сертификат пользователя (client1.crt) ===
</cert>
<key>
=== Приватный ключ пользователя (client1.key) ===
```

```
</key>
<tls-crypt>
=== Ключ HMAC для TLS (ta.key) ===
</tls-crypt>
```

Настройка firewall (упрощенный вариант)

Для начала необходимо разрешить прохождение трафика между интерфейсами в файле /etc/sysctl.conf:

```
net.ipv4.ip_forward=1
```

и применить изменения:

```
sysctl -p
```

Разрешаем клиентам ходить куда угодно (eth0 - основной шлюз):

```
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j SNAT
```