

OpenVPN параметры

Приведенные в статье команды без -- (двумя знаками дефиса) перед командой должны быть использованы в конфигурационном файле, команды с -- в начале используются только из командной строки. Подробности использования команд и их параметры - `man openvpn`.

remote < *host* > - определяет удаленный конец туннеля. Могут использоваться записи IP и DNS.

local < *host* > - определяет локальный ip или имя хоста, на котором будет работать OpenVPN. Актуально, если на локальной машине несколько адресов.

dev < *device* > - определяет какой использовать тип устройства tun или tap. Например:

`dev tun`

или

`dev tap`

При одном из таких указаний будет использован свободный интерфейс tun или tap. Так же можно явно указывать номер виртуального интерфейса, например `tun0`.

port < *port number* > - указывает на каком порту будет работать OpenVPN (локально и удаленно).

proto < *proto* > - какой протокол будет использоваться. Возможные значения: `udp`, `tcp`, `tcp-client`, `tcp-server`. С первыми двумя все ясно, а на последних двух остановимся чуть подробнее:

`tcp-client` - сам пытается установить соединение

`tcp-server` - только ждет подключений

Примечательно, что с использованием протокола `udp` VPN будет работать чуть быстрее, чем `tcp`. Но в плане стабильности работы лучше выбирать `tcp` (как показывает практика, VPN-соединение более устойчиво)

remote-random - если указана данная опция и в `random` перечисленно несколько удаленных хостов, то OpenVPN в случайном порядке будет к ним подключаться. Применяется для балансировки нагрузки.

float - позволяет удаленному хосту изменять IP во время работы туннеля. Соединение при этом не разрывается.

ipchange < *cmd* > - выполняет скрипт или команду указанную в < *cmd* >, если IP сменился.

Пример:

`ipchange script-ip.sh`

connect-retry < *seconds* > - пробует переподключиться через указанное время в секундах, если соединение было разорвано.

connect-retry-max < *n* > - максимальное количество повторов если соединение было разорвано

resolv-retry < *seconds* > - если OpenVPN не удалось узнать имя удаленного хоста по DNS, то через указанное количество секунд попытаться переподключиться.

lport < port > - указывает на локальный порт для использования OpenVPN

rport < port > - аналогично для удаленного порта. Пример:
rport 8000 - OpenVPN будет пытаться подключиться к удаленному порту 8000

nobind - использовать динамический порт для подключения (только для клиента)

shaper < bytes > - указывает скорость передачи данных в байтах для исходящего трафика (только для клиента)

tun-mtu < mtu size > - устанавливает максимальный размер MTU. По умолчанию tun-mtu равен 1500. Использование:
tun-mtu 1200

dev-node < interface name > - устанавливает имя виртуального интерфейса. Например:
dev-node openvpn1

ifconfig - устанавливает локальный IP и маску подсети для туннельного интерфейса.
Например:
ifconfig 10.3.0.1 255.255.255.0

server < network > < mask > - автоматически присваивает адреса всем клиентам (DHCP) в указанном диапазоне с маской сети. Данная опция заменяет ifconfig и может работать только с TLS-клиентами в режиме TUN, соответственно использование сертификатов обязательно.
Например:
server 10.3.0.0 255.255.255.0
Подключившиеся клиенты получают адреса в диапазоне между 10.3.0.1 и 10.3.0.254.

server-bridge < gateway > < mask > < pool > - сервер в режиме моста для TAP устройств.
Пример:
server bridge 10.3.0.1 255.255.255.0 10.3.0.128 10.3.0.254
Клиентам будут выданы адреса в диапазоне 10.3.0.128 - 10.3.0.254, в качестве шлюза будет указан 10.3.0.1.

mode server - переключает OpenVPN в режим сервера (начиная с 2-й версии)

mode p2p - данная опция идет по умолчанию.

Опции в режиме сервера

push < options > - передача клиенту конфигурационных параметров. Пример:
push "route 192.168.0.0 255.255.255.0"

Аналогично с помощью push клиенту могут передаваться следующие параметры:

route
route-gateway
route-delay
redirect-gateway
inactive
ping, ping-exit, ping-restart
persist-key, persist-tun
comp-lzo
dhcp-option
ip-win32

Последние две опции применимы только для Window-клиентов. Например передадим Windows-клиенту адрес DNS-сервера 11.11.11.11:
push "dhcp-option DNS 11.11.11.11"

comp-lzo - параметр сжатия трафика, идущего через виртуальный туннель. Может принимать значения yes, no, adaptive. Последнее используется по умолчанию.

Например:

comp-lzo yes - принудительно включить сжатие

comp-lzo no - принудительно отключить сжатие

comp-lzo adaptive - адаптивный режим.

Команды и параметры при работе с сертификатами x509 и параметрами шифрования

cipher < alg > - указываем алгоритм шифрования. Например:

cipher AES-256-CBC

Рекомендуется использование шифров в режиме CBC (Cipher Block Chaining).

keysize < n > - размер ключа в битах. Например:

keysize 128

auth < alg > - алгоритм хэширования. Пример:

auth SHA1

df < file > - файл с ключем Диффи-Хелмана

ca < file > - файл сертификата для СА

cert < file > - сертификат локальной машины

key < file > - локальный ключ машины

tls-server - явно указывает, что данный хост является tls-server

tls-client - соответственно tls-client

pkcs12 < file > - указываем файл (PKCS12), который содержит в себе сертификат, ключ и СА в одном файле. Пример:

pkcs12 /file

cr1-verify < *file* > - список отозванных сертификатов, т.е. blacklist.

no-replay - отключает защиту OpenVPN от атаки повторного воспроизведения (replay attack). Крайне не рекомендуется отключать!

no-iv - отключает использование вектора инициализации шифра (IV). Крайне не рекомендуется отключать!

Последние две опции очень сильно снижают безопасность OpenVPN, крайне не рекомендуется их использование.

secret < *file* > - включает режим шифрования и аутентификации на статических ключах. В качестве параметра использует заранее сгенерированный, командой --genkey, файл.

Например:

```
secret key.txt
```

Все доступные алгоритмы шифрования можно просмотреть выполнив из командной строки:

```
openvpn --show-ciphers
```

Алгоритмы хэширования:

```
openvpn --show-digests
```

Показать все доступные TLS-шифры (TLS используется только для шифрования канала управления)

```
openvpn --show-tls
```

Показать все доступные крипто-устройства в системе (если такие имеются):

```
openvpn --show-engines
```

Для улучшения безопасности рекомендовано запускать все сервисы с минимальными правами. Следующими двумя командами мы укажем с правами какого пользователя и группы будет работать openvpn:

```
user nobody
```

```
group nogroup
```

Где, соответственно, nobody и nogroup имена пользователя и группы.

Команды для управления маршрутизацией

Обозначение: VPN-хост - удаленная сторона (удаленный хост)

route < *network* > - устанавливает указанную маршрутизацию на VPN-хосте, после успешного запуска туннеля. Пример:

```
route 10.0.10.0 255.255.255.252
```

route-gateway < *IP* > - устанавливает шлюз на VPN-хосте. Пример:

```
route-gateway 192.168.0.22
```

После успешного запуска виртуального туннеля клиенту будет задан шлюз 192.168.0.22

route-delay < *seconds* > - указывает подождать n-секунд перед установкой маршрутов.

Пример:

```
route-delay 5
```

Т.е. через 5 секунд после установки туннеля будут заданы маршруты.

route-up < *cmd* > - выполнить скрипт или программу < *cmd* > после установки маршрутов.

Пример:

```
route-up /script.sh
```

redirect-gateway - установить шлюзом по умолчанию удаленный сервер. Т.е. когда удаленный пользователь подключается к нашему серверу, то ему будет задан шлюз по умолчанию на наш сервер.

Команды для управления туннелем

ping < *seconds* > - указывает отсылать ping на удаленный конец туннеля после указанных n-секунд, если по туннелю не передавался никакой трафик. Пример:

```
ping 10
```

ping-restart < *seconds* > - если за указанное время не было получено ни одного пакета с удаленной стороны, то перезапускать туннель. Пример:

```
ping-restart 60
```

- если в течении 60 секунд не было получено ни одного пакета, то туннель будет перезапущен.

ping-timer-rem - позволяет перезапускать туннель, только когда указан удаленный адрес.

persist-tun - данная опция оставляет без изменения устройства tun/tap при перезапуске OpenVPN.

persist-key - указывает не перечитывать файлы ключей при перезапуске туннеля.

resolv-retry < *seconds* > - устанавливает время в секундах для запроса об удаленном имени хоста. Актуально только если используется DNS-имя удаленного хоста. Пример:

```
resolv-retry 86400
```

inactive < *seconds* > - после n-секунд неактивности устройство TUN/TAP автоматически отключается. Пример:

```
inactive 120
```

ping-exit < *seconds* > - если за указанные n-секунд не было получено ни одного пакета, то отключать OpenVPN. Пример:

```
ping-exit 120
```

keepalive < *seconds* > < *seconds* > - является совмещением сразу двух команд - ping и ping-restart. Использует сразу два параметра в секундах, перечисленных через пробел. Пример:

```
keepalive 10 180
```

Означает следующее: каждые 10 секунд посылать ping на удаленный хост, и, если за 180 секунд не было получено ни одного пакета - то перезапускать туннель.

persist-local-ip < *IP* > - оставлять неизменными локальный IP адрес и номер порт, если туннель был перезапущен.

persist-remote-ip < *IP* > - оставлять неизменными удаленный IP адрес и номер порт, если туннель был перезапущен.
persist-remote-ip 192.168.50.1

Методы аутентификации

auth-user-pass-verify < *script* > < *method* > - указывается только на серверной стороне.
< *script* > - путь к скрипту, который будет производить авторизацию. Скрипт должен возвращать 0 если авторизация успешна, и соответственно, 1 если авторизация не успешна.
< *method* > - метод авторизации, может быть двух типов: *via-env* и *via-file*

auth-user-pass < *file* > - указывается на клиентской стороне. Параметр

username
password

client-cert-not-required - отключает авторизацию по сертификатам.

Работа с прокси

OpenVPN без проблем может работать через http и socks прокси.

http-proxy < *server port [auth]* > - указываем адрес и порт прокси-сервера.
http-proxy 192.168.0.12 8080

Если требуется авторизация на прокси-сервере:

http-proxy < *server port authfile* > - где *authfile* - файл содержащий две строки (имя пользователя и пароль) или *stdin* (будет запрошено имя пользователя и пароль).

Так же после *authfile* требуется указать метод авторизации. Можно оставить *auto* для автоматического выбора метода авторизации или указать явно через *auth-method*.

auth-method может быть трех видов "none", "basic" или "ntlm".

Используется в OpenVPN начиная с версии 2.1.

http-proxy-retry - переподключаться, если соединение было разорвано.

http-proxy-timeout < *seconds* > - считать соединение с прокси-сервером разорванным после *n*-секунд неактивности. Например:

http-proxy-timeout 5

socks-proxy < *server port* > - указываем сокс-прокси сервер. Пример:
socks-proxy 192.168.0.12 8080

socks-proxy-retry - переподключаться, если соединение было разорвано.

auto-proxy - автоматически определять прокси-сервер. Требуется версия OpenVPN 2.1 и выше.

Скриптинг

up < *command* > - выполнить команду после запуска устройства TUN/TAP. Пример:
up script-up.sh

up-delay < *seconds* > - подождать n-секунд перед запуском команды указанной в up. Пример:
up-delay 5

down < *command* > - выполнить команду когда интерфейс TUN/TAP выключится. Пример:
down script-down.sh

down-pre - выполнить команду, указанную в down перед выключением интерфейса TUN/TAP

up-restart < *command* > - выполнить команду после каждого реконнекта

route-up < *command* > - выполнить команду после установки сетевых маршрутов. Пример:
route-up script.sh

learn-address < *command* > - выполнить указанную команду, если ip удаленной стороны изменился.

ipchange < *command* > - выполнить команду, если ip сервера изменился.

client-connect < *command* > - выполнить команду, когда клиент подключился.

client-disconnect < *command* > - выполнить команду, когда клиент отключился.

Команды отладки и поиска неисправностей

verb < *verbosity level* > - устанавливает уровень информативности отладочных сообщений. Может принимать параметр от 0 до 11. По умолчанию verb равен 1.

При уровне verb 5 и выше в логе будут встречаться подобные записи: RwtW. R (read), W (write) - соответственно чтение и запись. Большая буква обозначает, что пакет был считан (R) или записан (W) на виртуальном устройстве TUN/TAP, а маленькие - считан (r) и записан (w) в туннеле.

mute < *number of messages* > - если значение установлено в 10, то в лог будет записываться только по 10 сообщений из одной категории.

Логирование

log < file > - указываем лог-файл. Если данный параметр не указан, то весь вывод openssl будет производиться в stdout.

log-append < file > - дописывать сообщения в лог-файл, а не перезаписывать.

status < file > - указывает путь к статус-файлу, в котором содержится информация о текущих соединениях и информация о интерфейсах TUN/TAP.

В. Чем отличаются виртуальные устройства tun и tap?

О. TUN - туннель, соединение по которому указывается по типу: локальный IP < --- > удаленный IP. Например, при явном указании ifconfig:

```
--ifconfig 10.3.0.2 10.3.0.1
```

в этом примере 10.3.0.2 - локальный IP, 10.3.0.1 - удаленный IP

TAP - эмулирует виртуальную ethernet карточку, для которой требуется указывать локальный IP и маску подсети. Например:

```
--ifconfig 10.3.0.2 255.255.255.0
```

В. Для чего нужны файлы serial и index.txt при генерации ключей с easy-rsa?

О. Эти два файла используются в качестве временной базы данных, используемой при генерации ключей. Должны находиться в том каталоге, где и ключи.