Как настроить сервер OpenVPN в Ubuntu 16.04

Posted December 20, 2016 404.9k UBUNTU VPN UBUNTU 16.04

By Justin Ellingwood
Become an author

Введение

Хотите иметь безопасный и защищённый доступ в Интернет с вашего смартфона или ноутбука при подключении к незащищённой сети через WiFi отеля или кафе? Виртуальная частная сеть (Virtual Private Network, VPN) позволяет использовать незащищённые сети таким образом, как если бы вы работали в частной сети. Весь ваш трафик в этом случае проходит через VPN-сервер.

В комбинации с использованием HTTPS-соединения описываемые далее настройки позволят вам обезопасить свою приватную информацию, например, логины и пароли, а также ваши покупки. Более того, вы сможете обходить региональные ограничения и цензуру, а также скрывать своё местонахождение и незашифрованный HTTP-трафик от незащищённой сети.

OpenVPN представляет собой мощное и гибко настраиваемое программное обеспечение с открытым исходным кодом для работы с Secure Socket Layer (SSL) VPN. В этой статье мы установим и настроим OpenVPN сервер, а также научимся осуществлять к нему доступ из Windows, Mac OS, iOS и Android. Для этого мы выполним несколько простых шагов.

Необходимые условия

Прежде всего вам необходимо иметь сервер с Ubuntu 16.04.

Перед тем, как начать следовать шагам, описанным в этой статье, вам необходимо настроить отдельный, не-рутовый (non-root) профиль пользователя с привилегиями `sudo`

на вашем сервере. Вы можете сделать это следуя инструкциям, описанным в статье о первичной настройке сервера на Ubuntu 16.04. В этой же статье описан процесс настройки файрвола; далее мы будем считать, что файрвол настроен на вашем сервере.

Когда вы будете готовы начать, зайдите на ваш сервер под созданным вами `sudo` - пользователем и следуйте шагам, описанным ниже.

Шаг 1. Установка OpenVPN

Сначала установим OpenVPN на наш сервер. OpenVPN доступен в стандартных репозиториях Ubuntu, мы можем использовать `apt` для его установки. Также мы установим пакет `easy-rsa`, который позволит нам настроить наш собственный внутренний центр сертификации (certificate authority, CA) для использования с нашей VPN.

Обновим список пакетов сервера и установим необходимые пакеты следующими командами:

```
$ sudo apt-get update
$ sudo apt-get install openvpn easy-rsa
```

Необходимое программное обеспечение установлено и готово к настройке.

Шаг 2. Создание директории центра сертификации

OpenVPN это виртуальная частная сеть, использующая TLS/SSL. Это означает, что OpenVPN использует сертификаты для шифрования трафика между сервером и клиентами. Для выпуска доверенных сертификатов (trusted certificates) нам потребуется создать наш собственный центр сертификации.

Для начала скопируем шаблонную директорию `easy-rsa` в нашу домашнюю директорию с помощью команды `make-cadir`:

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

Далее зайдём в эту директорию для начала настройки центра сертификации:

```
$ cd ~/openvpn-ca
```

Шаг 3. Настройка переменных центра сертификации

Для настройки переменных нашего центра сертификации нам необходимо отредактировать файл `vars`. Откройте этот файл в вашем текстовом редакторе:

```
$ nano vars
```

Внутри файла вы найдёте переменные, которые можно отредактировать, и которые задают параметры сертификатов при их создании. Нам нужно изменить всего несколько переменных.

Перейдите ближе к концу файла и найдите настройки полей, используемые по умолчанию при создании сертификатов. Они должны выглядеть примерно так:

~/openvpn-ca/vars

```
. . .

export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL="me@myhost.mydomain"
export KEY_OU="MyOrganizationalUnit"

. . .
```

Замените значения, выделенные красным, на что-нибудь другое, не оставляйте их не заполненными:

~/openvpn-ca/vars

```
. . .

export KEY_COUNTRY="US"
export KEY_PROVINCE="NY"
export KEY_CITY="New York City"
export KEY_ORG="DigitalOcean"
```

```
export KEY_EMAIL="admin@example.com"
export KEY_OU="Community"

. . .
```

Пока мы в этом файле, отредактируем значение `KEY_NAME` чуть ниже, которое заполняет поле субъекта сертификатов. Для простоты зададим ему название `server`:

~/openvpn-ca/vars

```
export KEY_NAME="server"
```

Сохраните и закройте файл.

Шаг 4. Создание центра сертификации

Теперь мы можем использовать заданные нами переменные и утилиты `easy-rsa` для создания центра сертификации.

Убедитесь, что вы находитесь в директории центра сертификации и используйте команду `source` к файлу `vars`:

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

Вы должны увидеть следующий вывод:

Вывод

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

Вывод

```
Generating a 2048 bit RSA private key
.....................................................................................+
.............................+++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:
State or Province Name (full name) [NY]:
Locality Name (eg, city) [New York City]:
Organization Name (eg, company) [DigitalOcean]:
Organizational Unit Name (eg, section) [Community]:
Common Name (eg, your name or your server's hostname) [DigitalOcean CA]:
Name [server]:
Email Address [admin@email.com]:
```

Теперь у нас есть центр сертификации, который мы сможем использовать для создания всех остальных необходимых нам файлов.

Шаг 5. Создание сертификата, ключа и файлов шифрования для сервера

Далее создадим сертификат, пару ключей и некоторые дополнительные файлы, используемые для осуществления шифрования, для нашего сервера.

Начнём с создания сертификата OpenVPN и ключей для сервера. Это можно сделать следующей командой:

> Внимание: Если ранее вы выбрали имя, отличное от `server`, вам придётся немного изменить некоторые инструкции. Например, при копировании созданных файлов в

директорию `/etc/openvpn` вам придётся заменить имена на заданные вами. Вам также придётся изменить файл `/etc/openvpn/server.conf` для того, чтобы он указывал на корректные `.crt` и `.key` файлы.

```
$ ./build-key-server server
```

Вывод опять будет содержать значения по умолчанию, переданные этой команде (`server`), а также значения из файла `vars` .

Согласитесь со всеми значениями по умолчанию, нажимая **ENTER**. *Не задавайте* challenge password. В конце процесса два раза введите **y** для подписи и подтверждения создания сертификата:

Вывод

```
. . .

Certificate is to be certified until May  1 17:51:16 2026 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Далее создадим оставшиеся файлы. Мы можем сгенерировать сильные ключи протокола Диффи-Хеллмана, используемые при обмене ключами, командой:

```
$ ./build-dh
```

Для завершения этой команды может потребоваться несколько минут.

Далее мы можем сгенерировать подпись HMAC для усиления способности сервера проверять целостность TSL:

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

Шаг 6. Создание сертификата и пары ключей для клиента

Далее мы можем сгенерировать сертификат и пару ключей для клиента. Вообще это можно сделать и на клиентской машине и затем подписать полученный ключ центром сертификации сервера, но в этой статье для простоты мы сгенерируем подписанный ключ на сервере.

В этой статье мы создадим ключ и сертификат только для одного клиента. Если у вас несколько клиентов, вы можете повторять этот процесс сколько угодно раз. Просто каждый раз передавайте уникальное значение скрипту.

Поскольку мы можем вернуться к этому шагу позже, мы повторим команду `source` для файла `vars`. Мы будем использовать параметр `client1` для создания первого сертификата и ключа.

Для создания файлов без пароля для облегчения автоматических соединений используйте команду `build-key`:

```
$ cd ~/openvpn-ca
$ source vars
$ ./build-key client1
```

Для создания файлов, защищённых паролем, используйте команду `build-key-pass`:

```
$ cd ~/openvpn-ca
$ source vars
$ ./build-key-pass client1
```

В ходе процесса создания файлов все значения по умолчанию будут введены, вы можете нажимать **ENTER**. Не задавайте challenge password и введите **y** на запросы о подписи и подтверждении создания сертификата.

Шаг 7. Настройка сервиса OpenVPN

Далее настроим сервис OpenVPN с использованием созданных ранее файлов.

Копирование файлов в директорию OpenVPN

Нам необходимо скопировать нужные нам файлы в директорию `/etc/openvpn`.

Сначала скопируем созданные нами файлы. Они находятся в директории `~/openvpn-ca/keys`, в которой они и были созданы. Нам необходимо скопировать сертификат и ключ центра сертификации, сертификат и ключ сервера, подпись HMAC и файл Diffie-Hellman:

```
$ cd ~/openvpn-ca/keys
$ sudo cp ca.crt ca.key server.crt server.key ta.key dh2048.pem /etc/openvpn
```

Далее нам необходимо скопировать и распаковать файл-пример конфигурации OpenVPN в конфигурационную директорию, мы будем использовать этот файл в качестве базы для наших настроек:

```
/share/doc/openvpn/examples/sample-config-files/server.conf.gz | sudo tee /etc/openvpn/server
```

Настройка конфигурации OpenVPN

Теперь, когда наши файлы находятся на своём месте, займёмся настройкой конфигурационного файла сервера:

```
$ sudo nano /etc/openvpn/server.conf
```

Базовая настройка

Сначала найдём секцию HMAC поиском директивы `tls-auth`. Удалите ";" для того, чтобы раскомментировать строку с `tls-auth`. Далее добавьте параметр `key-direction` и установите его значение в "0":

/etc/openvpn/server.conf

```
tls-auth ta.key 0 # This file is secret
key-direction 0
```

Далее найдём секцию шифрования, нас интересуют закомментированные строки `cipher`. Шифр `AES-128-CBC` обеспечивает хороший уровень шифрования и широко поддерживается другими программными продуктами. Удалите ";" для раскомментирования строки `AES-128-CBC`:

/etc/openvpn/server.conf

```
cipher AES-128-CBC
```

Под этой строкой добавьте строку `auth` и выберите алгоритм HMAC. Хорошим выбором будет `SHA256`:

/etc/openvpn/server.conf

```
auth SHA256
```

Наконец, найдите настройки `user` и `group` и удалите ";" для раскомментирования этих строк:

/etc/openvpn/server.conf

```
user nobody
group nogroup
```

(Опционально) Проталкивание изменений DNS для перенаправления всего трафика через VPN

Сделанные нами настройки создают VPN соединение между двумя машинами, но они не заставляют эти машины использовать VPN соединение. Если вы хотите использовать VPN соединение для всего своего трафика, вам необходимо протолкнуть (push) настройки DNS на клиентские машины.

Для этого вам необходимо раскомментировать несколько директив. Найдите секцию `redirect-gateway` и удалите ";" из начала строки для расскоментирования `redirect-gateway`:

/etc/openvpn/server.conf

```
push "redirect-gateway def1 bypass-dhcp"
```

Чуть ниже находится секция `dhcp-option`. Удалите ";" для обеих строк:

/etc/openvpn/server.conf

```
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
```

Это позволит клиентам сконфигурировать свои настройки DNS для использования VPN соединения в качестве основного.

(Опционально) Настройка порта и протокола

По умолчанию OpenVPN использует порт 1194 и протокол UDP для соединения с клиентами. Если вам необходимо изменить порт из-за каких-либо ограничений для ваших клиентов, вы можете сделать это изменив настройку `port`. Если вы не хотите веб-контент на вашем OpenVPN сервере, вы можете использовать порт 443, поскольку этот порт обычно разрешён для использования в большинстве файрволов.

/etc/openvpn/server.conf

```
# Optional!
port 443
```

Используемый протокол может иметь ограничения по номеру порта. В этом случае измените `proto` с UDP на TCP:

/etc/openvpn/server.conf

```
# Optional!
proto tcp
```

Если у вас нет явной необходимости использовать другой порт, лучше оставить обе эти настройки со значениями по умолчанию.

(Опционально) Использование кастомного имени сертификата и ключа

Если во время использования команды `./build-key-server` чуть выше вы указали параметр, отличный от `server`, измените настройки `cert` и `key`, чтобы они указывали на правильные файлы `.crt` и `.key`. Если вы использовали `server`, эти настройки должны выглядеть таким образом:

/etc/openvpn/server.conf

```
cert server.crt
key server.key
```

Шаг 8. Настройка сетевой конфигурации сервера

Далее нам необходимо настроить сетевую конфигурацию сервера, чтобы OpenVPN мог корректно перенаправлять трафик.

Настройка перенаправления IP

Сначала разрешим серверу перенаправлять трафик. Это ключевая функциональность нашего VPN сервера.

Настроим это в файле `/etc/sysctl.conf`:

```
$ sudo nano /etc/sysctl.conf
```

Найдите строку настройки `net.ipv4.ip_forward`. Удалите “#” из начала строки, чтобы раскомментировать её:

/etc/sysctl.conf

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

Сохраните и закройте файл.

Для применения настроек к текущей сессии наберите команду:

```
$ sudo sysctl -p
```

Настройка правил UFW для сокрытия соединений клиентов

Если вы следовали статье о настройке Ubuntu 16.04, упомянутой в начале этой статьи, у вас должен быть установлен и настроен файрвол UFW. Вне зависимости от того, используете ли вы файрвол для блокировки нежелательного трафика (что вам стоит делать практически всегда), в этой статье нам потребуется файрвол для манипулирования с входящим на сервер трафиком. Мы должны изменить файл настроек для сокрытия соединений (masquerading).

Перед тем, как изменить этот файл, мы должны найти публичный интерфейс сети (public

Публичный интерфейс должен следовать за словом “dev”. Например, в нашем случае этот интерфейс называется `wlp11s0`:

Вывод

```
default via 203.0.113.1 dev wlp11s0  proto static  metric 600
```

Зная название интерфейса откроем файл `/etc/ufw/before.rules` и добавим туда соответствующие настройки:

```
$ sudo nano /etc/ufw/before.rules
```

Это файл содержит настройки UFW, которое применяются перед применением правил UFW. Добавьте в начало файла выделенные красным строки. Это настроит правила, применяемые по умолчанию, к цепочке `POSTROUTING` в таблице `nat` и будет скрывать весь трафик от VPN:

Внимание: не забудьте заменить `eth0` в строке `-A POSTROUTING` на имя интерфейса, найденное нами ранее.

/etc/ufw/before.rules

```
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#

# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0
```

```
# Don't delete these required lines, otherwise there will be errors
*filter
. . .
```

Сохраните и закройте файл.

Теперь мы должны сообщить UFW, что ему по умолчанию необходимо разрешать перенаправленные пакеты. Для этого откройте файл `/etc/default/ufw`:

```
$ sudo nano /etc/default/ufw
```

Найдите в файле директиву `DEFAULT_FORWARD_POLICY`. Мы изменим значение с `DROP` на `ACCEPT`:

/etc/default/ufw

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Сохраните и закройте файл.

Открытие порта OpenVPN и применение изменений

Далее настроим сам файрвол для разрешения трафика в OpenVPN.

Если вы не меняли порт и протокол в файле `/etc/openvpn/server.conf`, вам необходимо разрешить трафик UDP для порта 1194. Если вы изменили эти настройки, введите указанные вами значения.

Даже мы добавим порт SSH на случай, если вы не сделали этого ранее.

```
$ sudo ufw allow 1194/udp
$ sudo ufw allow OpenSSH
```

Теперь деактивируем и активируем UFW для применения внесённых изменений:

```
$ sudo ufw disable
```

Теперь наш сервер сконфигурирован для обработки трафика OpenVPN.

Шаг 9. Включение сервиса OpenVPN

Мы готовы включит сервис OpenVPN на нашем сервере. Мы можем сделать это с помощью systemd.

Нам необходимо запустить сервер OpenVPN указав имя нашего файла конфигурации в качестве переменной после имени файла systemd. Файл конфигурации для нашего сервера называется `/etc/openvpn/server.conf`, поэтому мы добавим `@server` в конец имени файла при его вызове:

```
$ sudo systemctl start openvpn@server
```

Убедимся, что сервис успешно запущен командой:

```
$ sudo systemctl status openvpn@server
```

Если всё получилось, вывод должен выглядеть примерно следующим образом:

Вывод

```
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2016-05-03 15:30:05 EDT; 47s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
  Process: 5852 ExecStart=/usr/sbin/openvpn --daemon ovpn-%i --status /run/openvpn/%i.statu
 Main PID: 5856 (openvpn)
    Tasks: 1 (limit: 512)
   CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
           └─5856 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.statu

May 03 15:30:05 openvpn2 ovpn-server[5856]: /sbin/ip addr add dev tun0 local 10.8.0.1 peer
May 03 15:30:05 openvpn2 ovpn-server[5856]: /sbin/ip route add 10.8.0.0/24 via 10.8.0.2
May 03 15:30:05 openvpn2 ovpn-server[5856]: GID set to nogroup
May 03 15:30:05 openvpn2 ovpn-server[5856]: UID set to nobody
```

```
May 03 15:30:05 openvpn2 ovpn-server[5856]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0
May 03 15:30:05 openvpn2 ovpn-server[5856]: IFCONFIG POOL LIST
May 03 15:30:05 openvpn2 ovpn-server[5856]: Initialization Sequence Completed
```

Вы также можете проверить доступность интерфейса OpenVPN `tun0` следующей командой:

```
$ ip addr show tun0
```

Вы должны увидеть конфигурацию интерфейса:

Вывод

```
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN grc
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
       valid_lft forever preferred_lft forever
```

Если всё в порядке, настроем сервис на автоматическое включение при загрузке сервера:

```
$ sudo systemctl enable openvpn@server
```

Шаг 10. Создание инфраструктуры настройки клиентов

Далее настроим систему для простого создания файлов конфигурации для клиентов.

Создание структуры директорий конфигурации клиентов

В домашней директории создайте структуру директорий для хранения файлов:

```
$ mkdir -p ~/client-configs/files
```

Поскольку наши файлы конфигурации будут содержать клиентские ключи, мы должны настроить права доступа для созданных директорий:

Создание базовой конфигурации

Далее скопируем конфигурацию-пример в нашу директорию для использования в качестве нашей базовой конфигурации:

```
$ cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-configs/base
```

Откройте этот файл в вашем текстовом редакторе:

```
$ nano ~/client-configs/base.conf
```

Сделаем несколько изменений в этом файле.

Сначала найдите директиву `remote`. Эта директива сообщает клиенту адрес нашего сервера OpenVPN. Это должен быть публичный IP адрес вашего сервера OpenVPN. Если вы изменили порт, который слушает сервер OpenVPN, измените порт по умолчанию `1194` на ваше значение:

~/client-configs/base.conf

```
. . .
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote server_IP_address 1194
. . .
```

Убедитесь, что протокол совпадает с настройками сервера:

~/client-configs/base.conf

```
proto udp
```

Далее раскомментируйте директивы `user` и `group` удаляя ";":

~/client-configs/base.conf

Найдите директивы `ca`, `cert` и `key`. Закомментируйте эти директивы, так как мы будем добавлять сертификаты и ключи в самом файле:

~/client-configs/base.conf

```
# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
#ca ca.crt
#cert client.crt
#key client.key
```

Добавьте настройки `cipher` и `auth` согласно заданным в файле `/etc/openvpn/server.conf`:

~/client-configs/base.conf

```
cipher AES-128-CBC
auth SHA256
```

Далее добавьте директиву `key-direction` в любое место в файле. Она **должна** иметь значение “1” для корректной работы сервера:

~/client-configs/base.conf

```
key-direction 1
```

Наконец, добавьте несколько **закомментированных** строк. Мы ходим добавить эти строки в каждый файл конфигурации, но они будут включены только для клиентов на Linux, которые используют файл `/etc/openvpn/update-resolv-conf`. Этот скрипт использует утилиту `resolvconf` для обновление информации DNS на клиентах Linux.

~/client-configs/base.conf

```
# down /etc/openvpn/update-resolv-conf
```

Если ваш клиент работает на Linux и использует файл `/etc/openvpn/update-resolv-conf`, вы должны раскомментировать эти строки в сгенерированном клиентском файле конфигурации OpenVPN.

Сохраните и закройте файл.

Создание скрипта генерации файлов конфигурации

Теперь создадим простой скрипт для генерации файлов конфигурации с релевантными сертификатами, ключами и файлами шифрования. Он будет помещать сгенерированные файла конфигурации в директорию `~/client-configs/files`.

Создайте и откройте файл `make_config.sh` внутри директории `~/client-configs`:

```
$ nano ~/client-configs/make_config.sh
```

Вставьте следующие текст в этот файл:

~/client-configs/make_config.sh

```
#!/bin/bash

# First argument: Client identifier

KEY_DIR=~/openvpn-ca/keys
OUTPUT_DIR=~/client-configs/files
BASE_CONFIG=~/client-configs/base.conf

cat ${BASE_CONFIG} \
    <(echo -e '<ca>') \
    ${KEY_DIR}/ca.crt \
    <(echo -e '</ca>\n<cert>') \
    ${KEY_DIR}/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/${1}.key \
    <(echo -e '</key>\n<tls-auth>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-auth>') \
```

Сохраните и закройте файл.

Сделайте его исполняемым файлом командой:

```
$ chmod 700 ~/client-configs/make_config.sh
```

Шаг 11. Генерация конфигураций клиентов

Теперь мы можем легко сгенерировать файлы конфигурации клиентов.

Если вы следовали всем шагам этой статьи, вы создали сертификат `client1.crt` и ключ клиента `client1.key` командой `./build-key client1` на шаге 6. Вы можете сгенерировать конфигурацию для этих файлов перейдя в директорию `~/client-configs` и используя только что созданный нами скрипт:

```
$ cd ~/client-configs
$ ./make_config.sh client1
```

Если всё прошло успешно, мы должны получить файл `client1.ovpn` в директории `~/client-configs/files`:

```
$ ls ~/client-configs/files
```

Вывод

```
client1.ovpn
```

Доставка конфигураций клиентам

Теперь мы должны переместить файл конфигурации на клиентское устройство. Например, на компьютер или смартфон.

Способ доставки файла зависит от операционной системы вашего устройства и программного обеспечения, которое вы захотите использовать для перемещения файла. Мы рекомендуем передавать файл по защищённому соединению, например, с

Ниже мы приводим пример передачи файла client1.ovpn с использованием SFTP. Следующую команду можно использовать на вашем локальном компьютере под управлением Mac OS или Linux. Она перемещает файл `.ovpn` в вашу домашнюю директорию:

```
$ sftp sammy@openvpn_server_ip:client-configs/files/client1.ovpn ~/
```

Ниже представлено несколько ссылок на инструменты и статьи о безопасном переносе файлов с сервера на локальный компьютер:

- WinSCP
- Как использовать SFTP для безопасной передачи файлов с удалённого сервера
- Как использовать Filezilla для передачи и управления файлами на удалённом сервере

Шаг 12. Установка файлов конфигураций клиентов

Теперь мы поговорим о том, как устанавливать клиентские профили VPN на Windows, Mac OS, iOS и Android. Процесс установки уникален для каждой платформы, поэтому пропускайте платформы, которые вы не планируете использовать.

Название соединения OpenVPN зависит от того, как вы называли свой `.ovpn` файл. В нашем примере это будет `client1.ovpn`.

Windows

Установка

Вы можете загрузить клиент для работы с OpenVPN для Windows со страницы загрузок OpenVPN. Выберите необходимую вам версию установщика.

Внимание: установка OpenVPN требует администраторской учётной записи.

После установки OpenVPN скопируйте ваш `.ovpn` файл в эту директорию:

```
C:\Program Files\OpenVPN\config
```

Клиент OpenVPN требует запуска с правами администратора даже для аккаунтов администратора. Для запуска сделайте щелчок правой кнопкой мыши на клиенте и выберите **Run as administrator** каждый раз при запуске клиента. Это также означает, что обычные пользователи должны будут вводить пароль администратора для использования OpenVPN.

Для того, чтобы приложение OpenVPN всегда запускалось с правами администратора, сделайте щелчок правой кнопкой мыши на иконке клиента и перейдите в раздел **Properties**. В нижней части вкладки **Compatibility** нажмите на кнопку **Change settings for all users**. В открывшемся окне выберите **Run this program as an administrator**.

Соединение

Каждый раз при запуске клиента OpenVPN Windows будет спрашивать, хотите ли вы разрешить программе внести изменения в настройки вашего компьютера. Нажмите **Да**. Запуск клиента OpenVPN просто помещает приложение в системный трей, при этом само соединение не устанавливается автоматически.

Для установки соединения сделайте щелчок правой кнопкой мыши на иконке OpenVPN в системном трее. В открывшемся контекстном меню выберите **client1** (это наш профиль `client1.ovpn`) и нажмите **Connect**.

Откроется окно статуса, которое будет отображать лог соединения. При завершении соединения вы увидите соответствующее сообщение.

Закрыть VPN соединение можно точно так же: сделайте щелчок правой кнопкой мыши на иконке OpenVPN в системном трее, выберите профиль клиента и нажмите **Disconnect**.

Mac OS

Установка

Tunnelblick - это бесплатный OpenVPN клиент для Mac OS с открытым исходным кодом. Вы можете загрузить его со страницы загрузок Tunnelblick. Сделайте двойной щелчок на загруженном `.dmg` файле и следуйте инструкциям в процессе установки.

В конце процесса установки Tunnelblick спросит, есть ли у вас конфигурационные файлы. Проще всего ответить **No** и завершить установку Tunnelblick. Откройте Finder и сделайте двойной щелчок на `client1.ovpn`. Tunnelblick установит клиентский профиль. Для этого

Запустите Tunnelblick двойным щелчком из папки **Applications**. После запуска в панели меню в правой верхней части экрана появится иконка Tunnelblick. Для установки соединения нажмите на иконку, а затем **Connect**. Далее выберите соединение **client1**.

Linux

Установка

В зависимости от используемой вами версии Linux, вы можете использовать самые разные программы для установки соединения. Возможно, это умеет делать даже ваш менеджер окон.

Наиболее универсальным способом установки соединения, тем не менее, является программное обеспечение OpenVPN.

В Ubuntu или Debian вы можете установить его точно так же, как и на сервере:

```
$ sudo apt-get update
$ sudo apt-get install openvpn
```

В CentOS вы можете активировать EPEL репозитории и затем ввести следующие команды:

```
$ sudo yum install epel-release
$ sudo yum install openvpn
```

Настройка

Сначала проверьте, содержит ли ваш дистрибутив скрипт `/etc/openvpn/update-resolv-conf`:

```
$ ls /etc/openvpn
```

Вывод

```
update-resolve-conf
```

Далее отредактируйте полученный с сервера файл конфигурации клиента OpenVPN:

Если вам удалось найти файл `update-resolv-conf`, раскомментируйте следующие строки файла:

client1.ovpn

```
script-security 2
up /etc/openvpn/update-resolv-conf
down /etc/openvpn/update-resolv-conf
```

Если вы используете CentOS, измените `group` с `nogroup` на `nobody`:

client1.ovpn

```
group nobody
```

Сохраните и закройте файл.

Теперь вы можете соединиться с VPN используя команду `openvpn` следующим образом:

```
$ sudo openvpn --config client1.ovpn
```

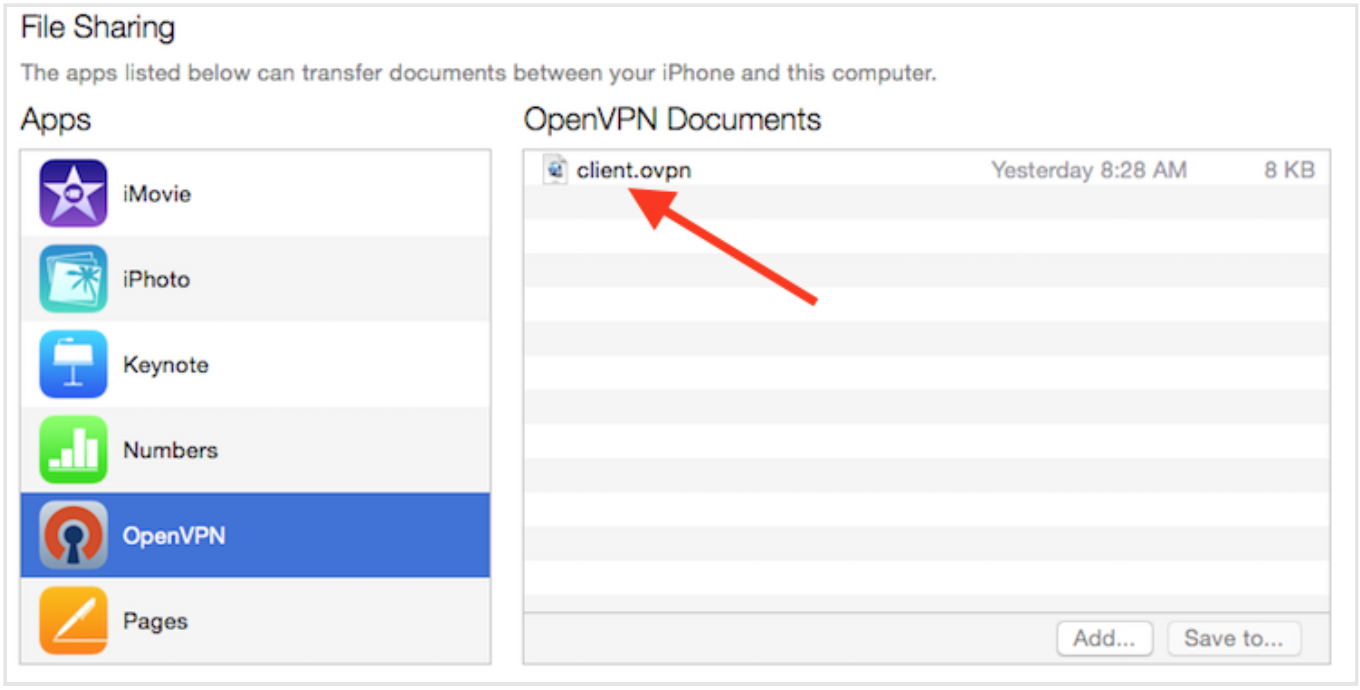
В результате вы подключитесь к серверу.

iOS

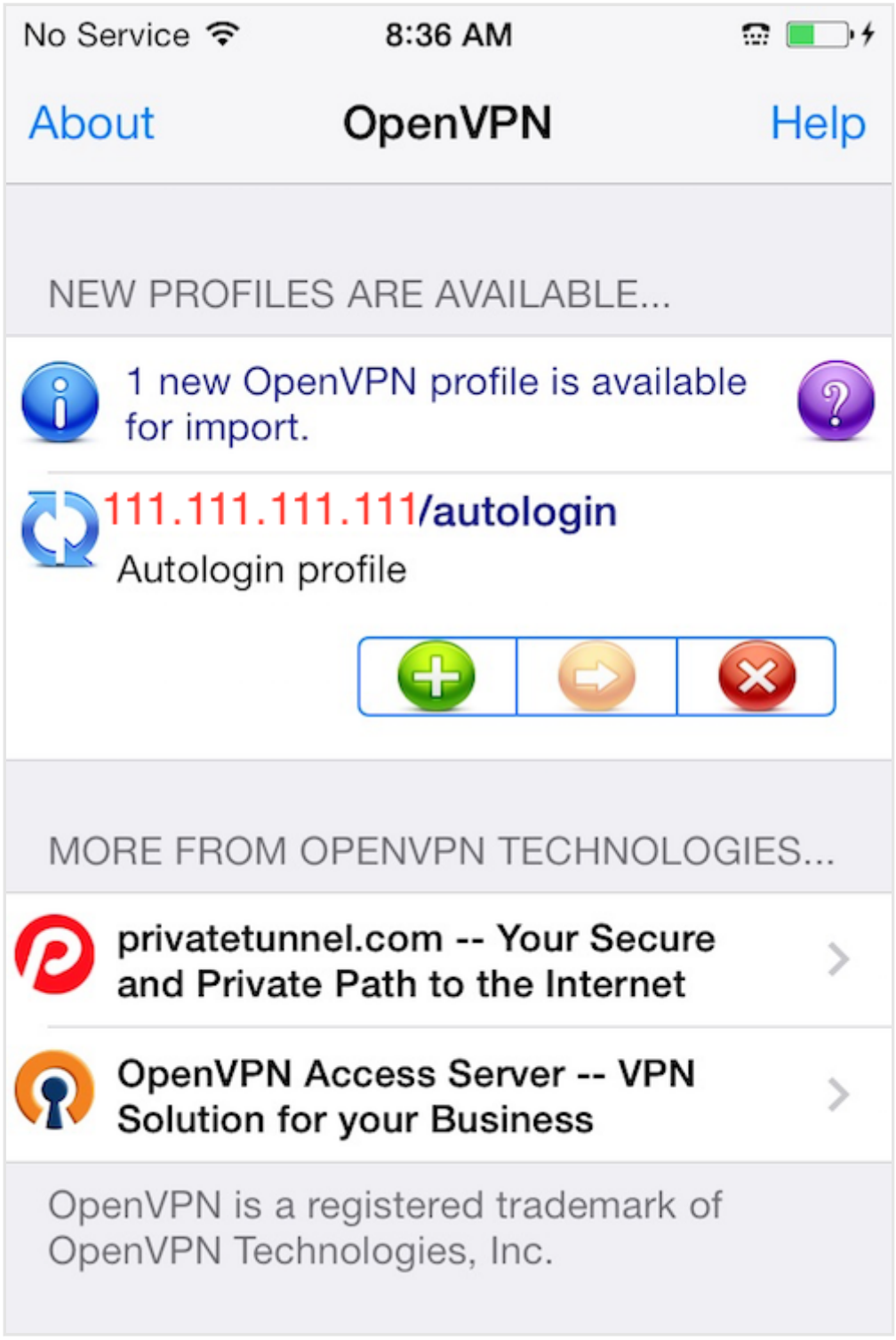
Установка

Найдите в iTunes App Store официальный клиент OpenVPN Connect и установите его. Для переноса файла конфигурации клиента на ваше устройство, подключите устройство к компьютеру.

Запустите iTunes на компьютере и выберите **iPhone > apps**. Найдите секцию **File Sharing** и нажмите на приложение OpenVPN. Перенесите ваш файл `.ovpn` в правую часть окна **OpenVPN Documents**.



Далее запустите приложение OpenVPN на iPhone. Вы получите уведомление, что новый профиль готов к импорту. Нажмите на зелёный плюсик для импорта профиля.



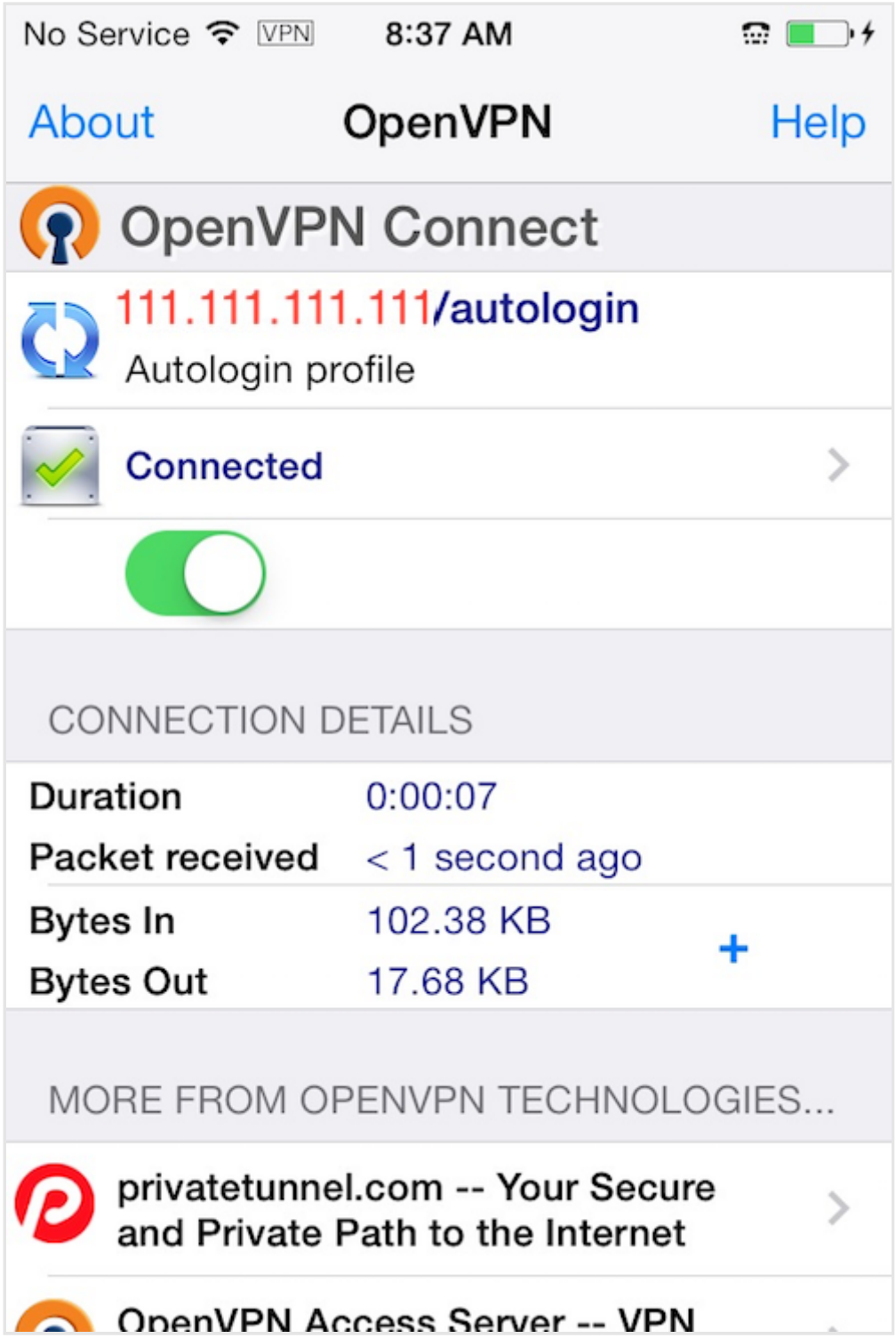
Соединение

OpenVPN готов к использованию с новым профилем. Для установки соединения потяните слайдер **Connect** в позицию **On**. Для остановки соединения переместите этот же слайдер в положение **Off**.

Внимание

инения с

но только в



Android

Установка

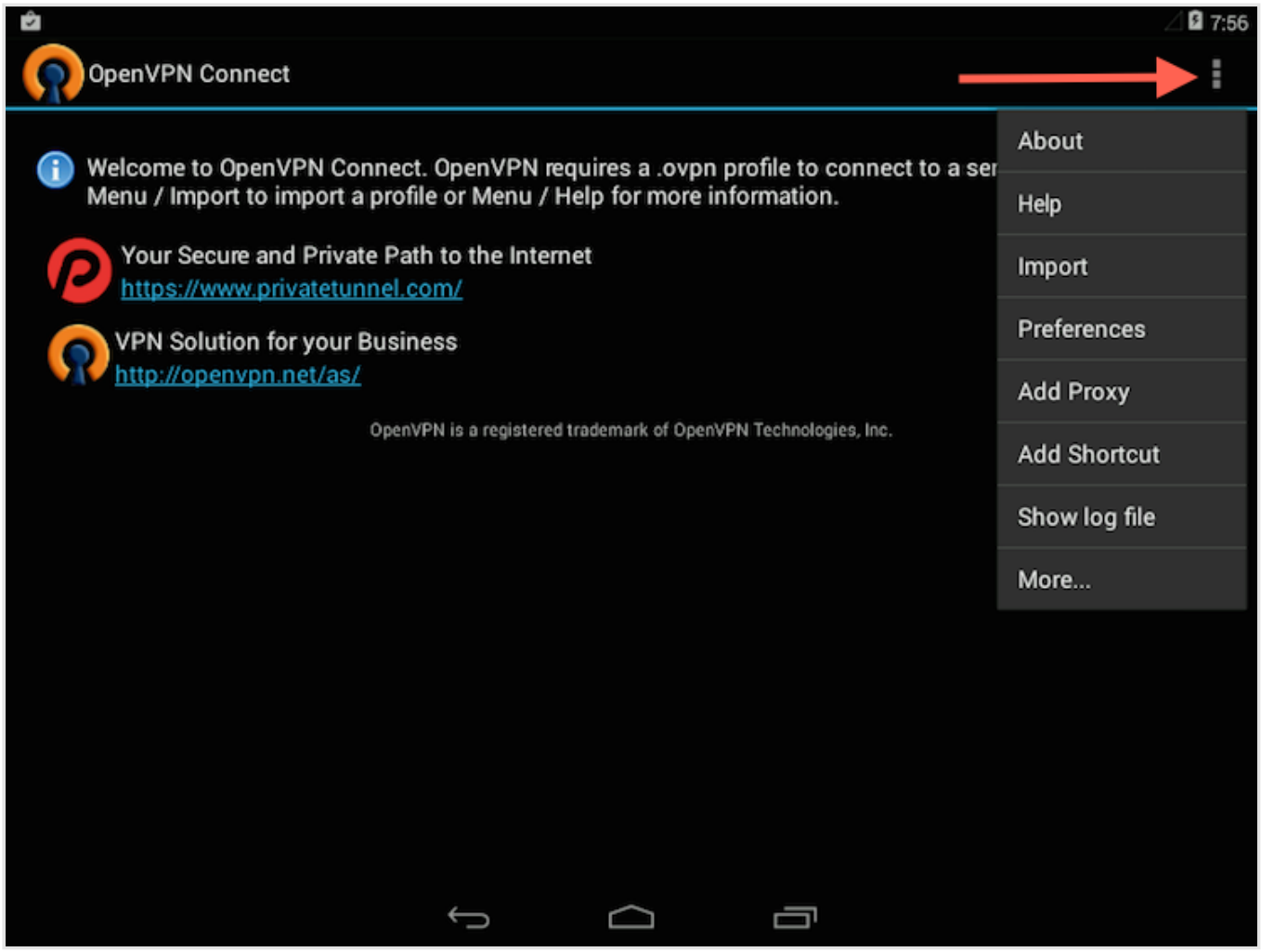
Откройте Google Play Store. Найдите и установите официальное приложение OpenVPN Android OpenVPN Connect.

Передать профиль с компьютера на телефон можно подключив Android устройство к

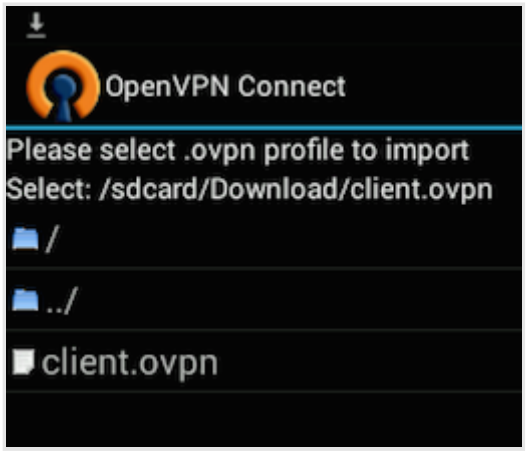
профиля с

ройство.

Запустите приложение OpenVPN и нажмите на меню для импорта профиля.



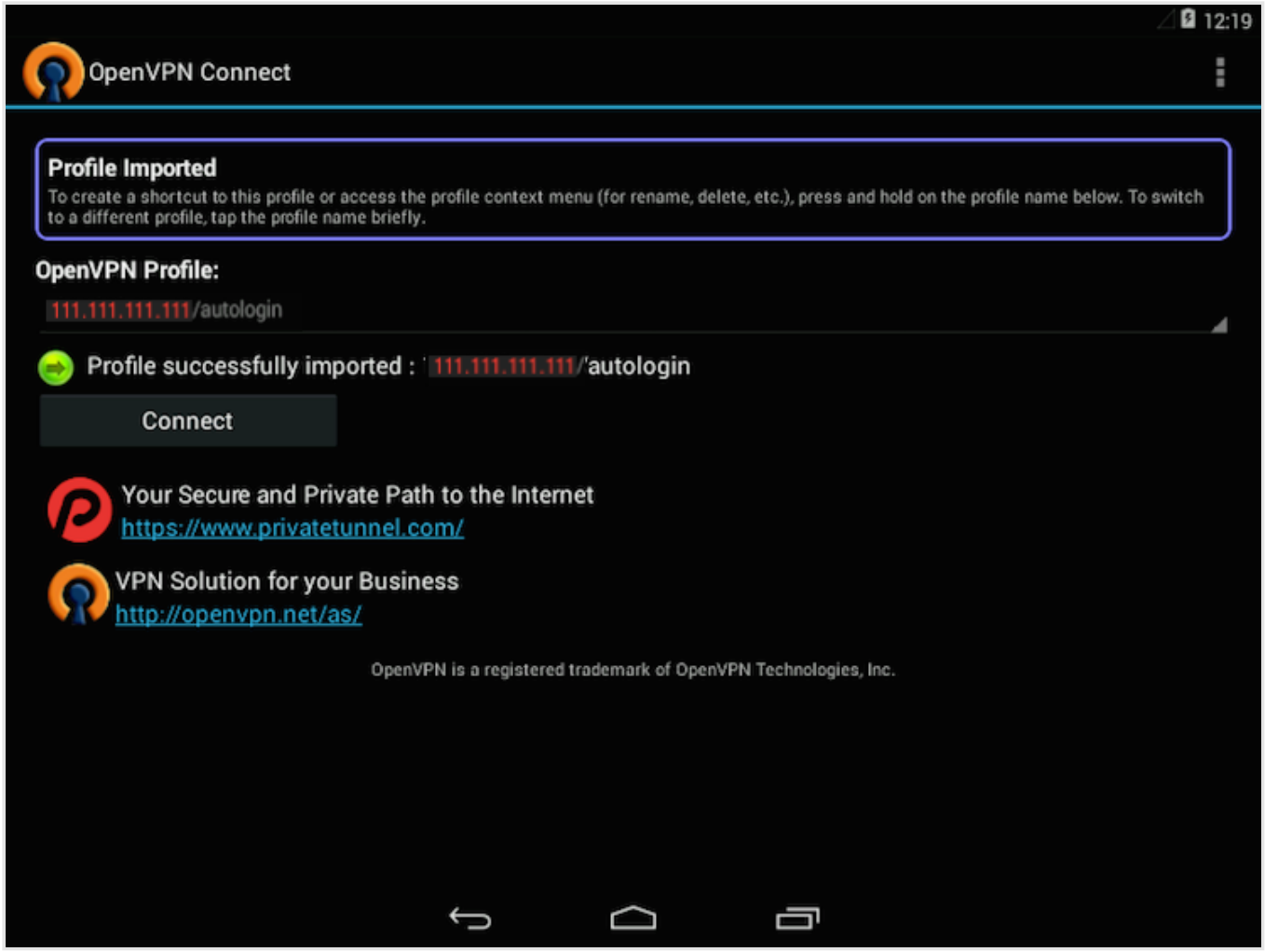
Далее найдите файл в файловой системе (в нашем примере это `/sdcard/Download/`) и выберите найденный файл. Приложение сообщит, что профиль был импортирован.



Соединение

Для установки соединения нажмите кнопку **Connect**. Вам будет задан вопрос, доверяете

новки



Шаг 13. Тестирование VPN соединения

После того, как всё установлено и настроено, убедимся, что всё работает правильно. Без установки соединения с VPN откройте браузер и зайдите на DNSLeakTest.

Этот сайт вернёт IP адрес, назначенный вам вашим Интернет-провайдером. Для того, чтобы проверить, какие DNS сервера используются, нажмите на **Extended Test**.

Теперь установите соединение, используя ваш VPN клиент и обновите страницу в браузере. Выдаваемый вам IP адрес должен быть совершенно другим. Теперь для всех в Интернете вы используете этот новый IP адрес. Нажмите **Extended Test** ещё раз, чтобы проверить ваши настройки DNS и убедитесь, что теперь вы используете DNS сервера вашего VPN.

Время от времени, вам может понадобиться отозвать клиентский сертификат для предотвращения доступа к серверу VPN&

Для этого зайдите в вашу директорию центра сертификации и введите команды:

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

Далее используйте команду `revoke-full` с именем клиента, сертификат которого вы хотите отозвать:

```
$ ./revoke-full client3
```

Вывод результатов работы этой команды будет оканчиваться ошибкой 23. Это нормально. В результате работы будет создан файл `crl.pem` в директории `keys` с необходимой для отзыва сертификата информацией.

Переместите этот файл в директорию `/etc/openvpn`:

```
$ sudo cp ~/openvpn-ca/keys/crl.pem /etc/openvpn
```

Далее откройте файл конфигурации сервера OpenVPN:

```
$ sudo nano /etc/openvpn/server.conf
```

Добавьте в конец файла строку `crl-verify`. Сервер OpenVPN будет проверять список отозванных сертификатов каждый раз, когда кто-то устанавливает соединение с сервером.

/etc/openvpn/server.conf

```
crl-verify crl.pem
```

Сохраните и закройте файл.

Перезапустите OpenVPN для завершения процесса отзыва сертификата:

Теперь клиент не сможет устанавливать соединение с сервером OpenVPN используя старый сертификат.

Для отзыва дополнительных сертификатов выполните следующие шаги:

1. Сгенерируйте новый список отозванных сертификатов используя команду `source vars` в директории `~/openvpn-ca` и выполняя команду `revoke-full` с именем клиента.
2. Скопируйте новый список отозванных сертификатов в директорию `/etc/openvpn` перезаписав тем самым старый список.
3. Перезапустите сервис OpenVPN.

Эта процедура может быть использована для отзыва любых созданных вами ранее сертификатов.

Заключение

Поздравляем! Теперь вы можете безопасно выходить в Интернет, весь ваш трафик защищён от прослушки цензоров и злоумышленников.

Для конфигурации дополнительных клиентов повторите шаги **6** и **11-13** для каждого нового устройства. Для отзыва доступа того или иного клиента используйте шаг **14**.

By Justin Ellingwood

Translation: maxmikheev

Report an issue

Related

TUTORIAL

Настройка аутентификации по паролю для Apache в Ubuntu 18.04

Вам как веб-администратору может быть полезна ...

TUTORIAL

Настройка аутентификации по паролю для Apache в Ubuntu 18.04 [Краткое руководство]

В этом обучающем руководстве мы ...

TUTORIAL

Оптимизация запросов MySQL с помощью кеширования ProxySQL в Ubuntu 16.04

Автор выбрал фонд Free Software Foundation для получения ...

TUTORIAL

Установка Tinc и настройка базового VPN в Ubuntu 18.04

Tinc — демон виртуальной частной сети (VPN) с открытым исходным кодом, имеющий много ...

Still looking for an answer?

79 Comments

Leave a comment...

Sign In to Comment

ipeacocks December 27, 2016

0

Спасибо. Статья такая - что особо и думать не нужно.

Но хотелось бы почитать о технических деталях, генерации CA/ключей и их ролях и т.п.

Если udp менять на tcp при подключению к openvpn-серверу насколько проседает скорость?

Reply Report

> **general1308** April 22, 2017
>
> 0
>
> потесть а потом отпишись нам)
>
> Reply Report
>
> > **ipeacocks** October 2, 2018
> >
> > 0
> >
> > В общем суть в том, что tcp over tcp использовать не нужно, только в случае, если udp заблокирован.
> >
> > http://sites.inka.de/bigred/devel/tcp-tcp.html
> >
> > Reply Report

egortarget January 19, 2017

доступ к

general1308 April 22, 2017

0

добавить только порт и включить выключить фаирвол. Т.е. разбиритесь с сттатее базованя настройка сервера(там и фаирвол). +не рекомендовал бы делавть опционально с днс пункт. После него пропадает интернет через опенвпн на клиенте.

Reply Report

latsi January 16, 2019

0

ufw allow ssh
или если изменили порт то
ufw allow номер порта

Reply Report

Gazovik72 January 24, 2017

3

Ошибка на
Шаг 9. Включение сервиса OpenVPN
Мы готовы включит сервис OpenVPN на нашем сервере. Мы можем сделать это с помощью systemd.

sudo systemctl start openvpn@server

в ответ пишет
Job for openvpn@server.service failed because a timeout was exceeded. See “systemctl status openvpn@server.service” and “journalctl -xe” for details.

Reply Report

aramms January 29, 2017

0

Spasbo za xaroshuyu statyu,

Tolko umenya odna problema voznikla s podklyucheniem RDP po VPN, podklyuchayus por RDP pishu ligin i parol i vse , zaveslo ves traffik. pomogite reshat problemu, Sposibo

Reply Report

me4b30871faaf52ab5f675372f March 19, 2017

0

Когда мы пишем правила для UFW, статья говорит добавить такую строчку:

```
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
```

Но по умолчанию OpenVPN выделяет для себя сетку 10.8.0.0/24. Поэтому для безопасности стоит заменить ту строку на

Reply Report

general1308 April 22, 2017

0

а как сделать чтоб ипишник на клиенте менялся на ипишник сервака?

Reply Report

denis0k October 24, 2017

0

проделать все обязательные пункты по инструкции и

(Опционально) Проталкивание изменений DNS для перенаправления всего трафика через VPN

. пока не сделал, не менялся адрес на клиенте :-)

Reply Report

Impulse April 7, 2017

0

START OPENVPN RULES

NAT table rules

*nat

:POSTROUTING ACCEPT [0:0]

Allow traffic from OpenVPN client to eth0

-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE

COMMIT

END OPENVPN RULES

Советую *'nat' поменять на* 'NAT', потому что может команда не отрабатываться и будет ругаться на эту строку.

Reply Report

general1308 April 22, 2017

0

К опенвпну конектится но ипишник который который родной так и остался родным. Вывод не работает. Если вам надо ипишник сервера.

А если делаешь опционально пункт с днс еще и дотсуп пропадает к интернету через опенвпн.

С фаирволом я б рекомендовал поокуратней и разобрался по стате базованя настройка которая упоминаеться в самом верху.

Reply Report

Спасибо, все получилось. Правда были некоторые проблемы с запуском и еще некоторые пункты, но разрешил все. Сервер Ubuntu 16.04. Хорошая и подробная статья!

Reply Report

vladiksonic May 17, 2017

0

Можно дополнить вот этим:

1) `block-outside-dns`

в конфиг-файл OpenVPN-клиента что бы прирезать утечку DNS

2) `echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all`

что бы сервак не отзывался на пинги - усложнит обнаружение VPN

3) `net.ipv4.icmp_echo_ignore_all = 1`

положить внутрь /etc/sysctl.conf с той же целью, что пункт 2, для того, что бы между ребутами жило

4) `-I FORWARD -i tun0 -o tun0 -j DROP`

Это в /etc/ufw/before.rules, если мы не хотим, что бы клиенты могли общаться друг с другом.

Reply Report

> **vladiksonic** May 18, 2017
>
> 0
>
> Итак, 2 + 3 слетает при перезапуске ufw.
>
> Вместо 2) и 3) я вписал внутрь /etc/ufw/before.rules:
>
> ```
> -A ufw-before-input -p icmp --icmp-type echo-request -j DROP
> ```
>
> Вместо аналогичной строки:
>
> ```
> -A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT
> ```
>
> и после этого настройка сохраняется.
>
> Reply Report

Equus May 24, 2017

0

Подключается, но нет интернета и даже не пингуется сервер. Как исправить?

Reply Report

> **Equus** May 31, 2017
>
> 0
>
> nano /etc/rc.local
>
> ```
> добавляем строки до exit 0
> iptables -A FORWARD -s 10.8.0.0/24 -j ACCEPT
> iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
> iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
> ```
>
> Reply Report

START OPENVPN RULES
NAT table rules

*nat
:POSTROUTING ACCEPT [0:0]

Allow traffic from OpenVPN client to eth0

-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT

END OPENVPN RULES

Я лично буковку -s не поставил перед 10.8.0.0/8 и правило у меня тупо не выполнялось и была такая же проблема как у тебя. Исправил и все стало норм.

Reply Report

beletskiyv August 29, 2017

0

Ну а за подсказку спасибо :) Именно изучая что делают написанные тобой строки, понял в чем ошибка у меня))

Reply Report

yachmenevsv October 1, 2019

0

Все сделал по инструкции и твою запись пробовал добавлять, но интернет так и не появился(Есть еще варианты куда копать?

Reply Report

kuzentio June 17, 2017

0

Прошел по туториалу, все работает, но только с файлом конфигурации client1.ovpn, при генерации сертификатов, ключей и файлов конфигураций для других клиентов (client2.ovpn, client3.ovpn, ...), не может подключиться к серверу.

```
write UDPv4: Can't assign requested address (code=49)
```

Reply Report

freelancr June 21, 2017

0

Огромнейшее-преогромнейшее человеческое спасибо.

Reply Report

г, там поле

Reply Report

swhtlover June 28, 2017

0 Добрый день. Все сделал как гласит статья, но в Android клиенте постоянно пишится Looking up DNS name, что я сделал не так?

Reply Report

vope June 29, 2017

0 Если вы все сделали как написано, но ip все равно тот же - в файл **client1.ovpn** нужно дописать `redirect-gateway def1`

Статья отличная, очень помогла.

Reply Report

swhtlover June 29, 2017

0 Настроил сервер. Все работает отлично, покрайней мере на Андроиде, если сижу через десктоп браузер (Chrome), то большинство сайтов не открывается. Подскажите в чем проблема?

Reply Report

Clu July 28, 2017

1 Статья замечательная, автору большое спасибо! Хоть и удалось настроить сервер не с первой попытки, но в целом все очень понятно.
Пользуюсь OpenVPN чуть больше полугода, очень доволен результатом и стабильностью работы.
Недавно у меня возник один вопрос, а так как я новичок в этом деле, то решил задать его здесь, вдруг кто поможет.
Как настроить лимит трафика для каждого клиента по отдельности?
К примеру для client1 надо 10 Гб, для client2 5 Гб и т.д. С ежемесячным обновлением счетчика трафика для каждого из клиентов по отдельности или для удобства, определенного списка клиентов. Заранее спасибо!

Reply Report

megashurik August 12, 2017

0 Отличная статья, спасибо автору. Столкнулся только с проблемой, что два клиента, подключенные к серверу - не видят друг друга. Как можно исправить?
Мне не нужно, чтобы клиенты ходили в интернет через сервер. Мне нужно только чтобы они друг друга все видели. Чтобы у клиентов, подключенных по VPN образовалась своя локалка.

Reply Report

```
topology subnet
client-to-client
```

Reply Report

megashurik August 12, 2017

0

Йееееее! Спасибо огромное!!!

Reply Report

visionserg March 31, 2018

0

а если мне нужно чтоб они и между собой общались, и наружу выходили? первое убирать?

Reply Report

johnsonafbc1adcd05c4e192d2 August 29, 2017

0

подскажите, не могу добавить новых клиентов...

root@zur02:~# cd ~/openvpn-ca

root@zur02:~/openvpn-ca# source vars

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

root@zur02:~/openvpn-ca# ./build-key client2

pkitool: Need a readable ca.crt and ca.key in /root/openvpn-ca/keys

Try pkitool –initca to build a root certificate/key.

Reply Report

pivden August 31, 2017

0

выйдите из root-а

Reply Report

pivden August 31, 2017

0

Need a readable ca.crt and ca.key in /root/openvpn-ca/keys

Reply Report

pivden August 31, 2017

0

<^>WARNING: INSECURE cipher with block size less than 128 bit (64 bit). This allows attacks like SWEET32. Mitigate by using a –cipher with a larger block size (e.g. AES-256-CBC).

<^>

не страшно конечно, можно и AES-128-CBC оставить.

Reply Report

Load More Comments

BECOME A CONTRIBUTOR

You get paid; we donate to tech nonprofits.

GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.