



# Инструкция для клиента

 [1] Устанавливаем client >

```
sudo apt install client_0.1-1_all.deb
```

 [2] Меняем владельца и ставим права >


На директорию (что бы владелец имел все права на папку а все остальные нет)

```
sudo chown -R $USER:$USER /opt/easy-rsa
```

```
sudo chmod -R 700 /opt/easy-rsa
```

в итоге в созданной директории у вас должно получится примерно такое

```
toorr2p@debian-srv:~/easy-rsa$ lsa
total 8
drwx-----  2 toorr2p toorr2p 4096 Dec  2 20:11 .
drwx----- 16 toorr2p toorr2p 4096 Dec  2 20:11 ..
lrwxrwxrwx   1 toorr2p toorr2p   27 Dec  2 20:11 easyrsa -> /usr/share/easy-rsa/easyrsa
lrwxrwxrwx   1 toorr2p toorr2p   39 Dec  2 20:11 openssl-easyrsa.cnf -> /usr/share/easy-rsa/openssl-easyrsa.cnf
lrwxrwxrwx   1 toorr2p toorr2p   32 Dec  2 20:11 vars.example -> /usr/share/easy-rsa/vars.example
lrwxrwxrwx   1 toorr2p toorr2p   30 Dec  2 20:11 x509-types -> /usr/share/easy-rsa/x509-types
```

 [3] Создаем запрос на подпись >

Переходим в директорию

```
cd /opt/easy-rsa
```

После чего запускаем команду создания запроса на подпись  
будут созданы приватный ключ (.key) и запрос на подпись (.req)

```
./easyrsa gen-req client-1 nopass
```

Где:

- gen-req - ключ указывающий что нужно создать запрос на подпись
- client-1 - условно-произвольное имя клиента (например. ваш логин в системе)
- nopass - ключ указывающий что не нужно использовать пароль для ключа

```
client5@client5:/opt/easy-rsa$ ./easyrsa gen-req client-1 nopass
```

Note: using Easy-RSA configuration from: ./vars

```
Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020
ec:/opt/easy-rsa/pki/ecparams/secp384r1.pem
/opt/easy-rsa/pki/safessl-easyrsa.cnf
/opt/easy-rsa/pki/private/client-1.key.1pwFJvq953
/opt/easy-rsa/pki/reqs/client-1.req.BlnntWxqJk
-nodes
Generating an EC private key
writing new private key to '/opt/easy-rsa/pki/private/client-1.key.1pwFJvq953'
-----
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.
-----
Common Name (eg: your user, host, or server name) [client-1]:

Keypair and certificate request completed. Your files are:
req: /opt/easy-rsa/pki/reqs/client-1.req
key: /opt/easy-rsa/pki/private/client-1.key
```

- `/opt/easy-rsa/pki/reqs/client-1.req` - это запрос на подпись сертификата **именно этот файл нужно передать в PKI!**
- `/opt/easy-rsa/pki/private/client-1.key` - приватный ключ **ВЖНО: передавать приватный ключ никому нельзя!**

#### 📄 [4] Создаем пару приватный\публичный ключ GPG >

Следующая команда сгенерирует пару приватный\публичный ключ

```
gpg --gen-key
```

Программа запросит данные, Имя, Email и длину ключа (от 1024 до 4096, использовать ключи размера меньше 2048 бит небезопасно), срок действия ключа, passphrase и др (укажите реальное Имя и свой email адрес, с которого вы будите отправлять файлы системному администратору PKI). После чего, ключи сохранятся во внутреннюю память системы (файлы созданы не будут!).

При запросе Passphrase укажите пароль посложнее (без литературных фраз)!

```
Please enter the passphrase to
protect your new key
```

```
Passphrase: _____
```

```
<OK>
```

```
<Cancel>
```

в итоге будут создана пара приватный-публичный ключи

```
public and secret key created and signed.

pub      rsa3072 2023-12-18 [SC] [expires: 2025-12-17]
        63206BF64DF59ED2E839D18F0D29877BCA9EF39E
uid           Client1 <client1@gmail.com>
sub      rsa3072 2023-12-18 [E] [expires: 2025-12-17]
```

где:

- Client1 - то что было указано в запросе "Name" (укажите реальное ФИО латиницей!)
  - [client1@gmail.com](mailto:client1@gmail.com) - должен быть казан ваш реальный email
- Эти данные будут служить для идентификации, что полученный публичный ключ, действительно пренадлежит вам!

## 📄 [5] Передаем запрос на подпись и публичный ключ >

Для выпуска сертификата, передаем созданный файл запроса на подпись, любым доступным способом (например по email), системному администратору сервера с PKI инфраструктурой

```
/opt/easy-rsa/pki/reqs/client-1.req
```

Также передаем свой публичный ключ GPG, им будет зашифрован архив с выпущенным сертификатом и другими файлами (ta.key, ca.crt), записать ключ в файл можно следующей командой

```
gpg --export -a "toorrp4@gmail.com" > ~/gpgpub.key
```

Лучше их упаковать в архив, и передать все одним файлом, выполните следующую команду, для копирования файлов во временную директорию, и затем упаковку их в архив (имена ваших файлов могут отличаться!)

```
cp /opt/easy-rsa/pki/reqs/client-1.req ~/gpgpub.key /tmp && cd /tmp && tar -zcvf client-1_req.tar.gz client-1.req gpgpub.key
```

Отправляем файл архива client-1\_req.tar.gz на email админа pki [toorrp4@gmail.com](mailto:toorrp4@gmail.com)

```
client5@client5:/tmp$ cp /opt/easy-rsa/pki/reqs/client-1.req ~/gpgpub.key /tmp &
& cd /tmp && tar -zcvf client-1_req.tar.gz client-1.req gpgpub.key
client-1.req
gpgpub.key
client5@client5:/tmp$ ls
total 52
drwxrwxrwt 10 root    root    4096 Dec 18 21:12 .
drwxr-xr-x 18 root    root    4096 Dec 18 20:32 ..
-rw-----  1 client5 client5  440 Dec 18 21:12 client-1.req
-rw-rw-r--  1 client5 client5 2333 Dec 18 21:12 client-1_req.tar.gz
```

## 📄 [7] Получить файлы от админа сервера PKI >

Например по почте, или по предоставленному доступу по SSH

вы должны получить файлы (важно что бы канал передачи был зашифрован, или сами файлы были зашифрованы!)

Дешифруем файл архива полученный от админа

```
gpg client-1_sig.tar.gz.asc
```

в итоге из client-1\_sig.tar.gz.asc появится обычный архив client-1\_sig.tar.gz, содержимое которого можно просмотреть и котоый можно распаковать

```

client5@client5:~$ cd /tmp
client5@client5:/tmp$ gpg client-1_sig.tar.gz.asc
gpg: WARNING: no command supplied. Trying to guess what you mean ...
gpg: encrypted with 3072-bit RSA key, ID D51FC8EB90E9BB6A, created 2023-12-1
"Client1 <client1@gmail.com>"
gpg: Signature made Mon 18 Dec 2023 09:59:00 PM UTC
gpg: using RSA key 66D8B394C1DECAD854CE768D63C94CBE240B4D3C
gpg: Can't check signature: No public key
client5@client5:/tmp$ ls
total 72
drwxrwxrwt 10 root root 4096 Dec 18 22:10 .
drwxr-xr-x 18 root root 4096 Dec 18 20:32 ..
-rw----- 1 client5 client5 440 Dec 18 21:12 client-1.req
-rw-rw-r-- 1 client5 client5 2333 Dec 18 21:12 client-1_req.tar.gz
-rw-rw-r-- 1 client5 client5 4869 Dec 18 22:10 client-1_sig.tar.gz
-rw-rw-r-- 1 client5 client5 7926 Dec 18 22:03 client-1_sig.tar.gz.asc
drwxrwxrwt 2 root root 4096 Dec 18 20:32 .font-unix
-rw-rw-r-- 1 client5 client5 2448 Dec 18 21:12 gpgpub.key
drwxrwxrwt 2 root root 4096 Dec 18 20:32 .ICE-unix
drwx----- 3 root root 4096 Dec 18 20:32 systemd-private-36375ab9e0fa
drwx----- 3 root root 4096 Dec 18 20:32 systemd-private-36375ab9e0fa
drwx----- 3 root root 4096 Dec 18 20:32 systemd-private-36375ab9e0fa
drwxrwxrwt 2 root root 4096 Dec 18 20:32 .Test-unix
-rw-rw-r-- 1 client5 client5 2333 Dec 18 21:25 toorr2p@51.250.85.189
drwxrwxrwt 2 root root 4096 Dec 18 20:32 .X11-unix
drwxrwxrwt 2 root root 4096 Dec 18 20:32 .XIM-unix
client5@client5:/tmp$ tar -tf client-1_sig.tar.gz
./
./gpgpub.key
./ta.key
./ca.crt
./client-1.crt
./client-1.req

```

Распаковываем дешифрованный архив

```
tar -zxvf client-1_sig.tar.gz -C /opt/easy-rsa/pki/private
```

После распаковки должен быть набор из таких файлов

- **client-1.crt** - ваш сертификат
- **ta.key** - открытый публичный ключ сервера OpenVPN
- **ca.crt** - корневой сертификат сервера PKI (удостоверяющего центра)

```

client5@client5:/tmp$ tar -zxvf client-1_sig.tar.gz -C /opt/easy-rsa/pki/private
./
./gpgpub.key
./ta.key
./ca.crt
./client-1.crt
./client-1.req

```

⚡ **Обязательно проверьте это командой**

Если не будет хватать хотя бы одного файла, конфиг создан не будет, а скрипт выведет сообщение об ошибке!

```
ls -la /opt/easy-rsa/pki/private
```

## 📄 [7] Создать конфигурационный файл .ovpn >

Запустить файл генерации конфигурационного файла

```
/usr/bin/client_gen_ovpn.sh client-1
```

Где:



- client-1 - это логин пользователя (например. ваш логин в системе), или по имени файла сертификата, до точки (т.е если имя файла сертификата полученного от админа PKI было client-1.crt, то указывать нужно "client-1" команда будет выглядеть так \$ /usr/bin/client\_gen\_ovpn.sh client-1)  
Скрипт сгенерирует конфигурационный файл с расширением .ovpn и сохранит его в директории

```
/home/\<LOGIN\>/ovpn
```

Можете посмотреть его так:

```
cat ~/ovpn
```

## [7] Подключится к серверу >

Финальная часть, это подключение к серверу, оно выполняется из консоли

```
sudo openvpn --config ~/ovpn/client-1.ovpn --daemon
```

ВАЖНО: без sudo будет ошибка, которая указывает на то что запрещено создавать устройства (сетевой интерфейс tun0)

```
ERROR: Cannot ioctl TUNSETIFF tun: Operation not permitted (errno=1)
```

с sudo такой проблемы не возникнет, и увидеть новый сетевой интерфейс можно командой

```
ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 4c:cc:6a:bc:a3:99 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global noprefixroute enp0s31f6
        valid_lft forever preferred_lft forever
    inet6 fe80::4ecc:6aff:febc:a399/64 scope link
        valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.0.2/24 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::5c02:6da1:d6c0:9368/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
```

Проверить ваш новый IP можно на сервисе <https://ifconfig.me/>

или введя команду

```
curl -i ifconfig.me
```

Отключится от VPN сервера можно командой

```
pid=$(ps -aux | grep openv | grep -v grep | awk '{print $2}'); sudo kill -9 $pid
```