Инструкция для админа

[1] Раскатываем на сервере PKI

Запускаем скрипт, которые пересоздаст .deb пакеты, выгрузит их на указанный сервер и запустит их установку

```
cd ~/WORK/DEB/PKI
```

запускаем скрипт

```
./push.sh
```

Чек лист тестирования после установки

[1] Проверяем статус сервера openvpn

```
sudo systemctl status openvpn-server@server.service
```

[2] Проверяем что сервер работает на 1194 порту

```
netstat -nlup
```

[3] Убеждаемся что сервер PKI доступен извне по UDP на 1194 порту

```
sudo nmap -sU -p U:1194 <SERVER_IP>
```

[2] Создаем GPG ключи для шифрования файлов

Чтобы шифровать данные с помощью GPG, создайте пару ключей.
Для этого введите команду:

```
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. После чего, ключи сохранятся во внутреннюю память системы (файлы

созданы не будут!).

```
toorr2p@pki-ovpn-001:~/clients/client-1$ gpg --gen-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: pki-ovpn-001
Email address: admin@pki-ovpn-001.com
You selected this USER-ID:
    "pki-ovpn-001 <admin@pki-ovpn-001.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 63C94CBE240B4D3C marked as ultimately trusted
gpg: directory '/home/toorr2p/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/toorr2p/.gnupg/openpgp-revocs.d/6
public and secret key created and signed.

pub   rsa3072 2023-12-18 [SC] [expires: 2025-12-17]
      66D8B394C1DECAD854CE768D63C94CBE240B4D3C
uid                      pki-ovpn-001 <admin@pki-ovpn-001.com>
sub   rsa3072 2023-12-18 [E] [expires: 2025-12-17]
```

Не забудьте создать сертификаты отзыва для ключей!

подробнее см. статью

[3] Выпуск сертификата клиента

Получаем файл запроса на подпись публичного ключа клиента (например client-1.req - запрос на выпуск сертификата), а также публичный GPG ключ клиента (например архивом по email)

Для удобства разделения файлов клиентов, создаем директорию (если ее нет), например по имени файла клиентского запроса, которое можно посмотреть в полученном архиве у файла с расширением .req ($ tar -tf client-1_req.tar.gz) в примере будет использоваться "client-1.req"

```
mkdir clients/client-1
```

Распаковываем архив в созданную клиентскую директорию

```
tar -zxvf /tmp/client-1_req.tar.gz -C clients/client-1
```

Переходим в директорию

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

Подписываем запрос (исп. тип запроса "client" и имя нашего клиента "client-1")

```
./easyrsa sign-req client client-1
```

Будет запрошена Passphrase к приватному ключу сервера PKI, вводим ее.
После чего запрос подпишется приватным ключом сервера и будет создан сертификат клиента. О чем будет сообщено

программой, и будет выведен путь к файлу сертификата

```
Certificate created at: /opt/easy-rsa/pki/issued/client-1.crt
```

```
toorr2p@pki-ovpn-001:~$ cd /opt/easy-rsa/
toorr2p@pki-ovpn-001:/opt/easy-rsa$ ./easyrsa sign-req client client-1

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

Using SSL: openssl OpenSSL 1.1.1f  31 Mar 2020


You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a client certificate for 1080 days:

subject=
    commonName                = client-1


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
Using configuration from /opt/easy-rsa/pki/safessl-easyrsa.cnf
Enter pass phrase for /opt/easy-rsa/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'client-1'
Certificate is to be certified until Dec  2 21:36:57 2026 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /opt/easy-rsa/pki/issued/client-1.crt
```

Этот сертификат нужно передать клиенту, с помощь него, он сможет сгенерировать скриптом client_gen_ovpn.sh, конфигурационный файл .ovpn для своего openvpn клиента и подключится к серверу PKI(VPN)

Также клиенту нужно передать сертификата клиента (client-1.crt), публичный ключ сервера OpenVPN (ta.key) и корневой сертификат удостоверяющего центра (ca.crt), копируем их в клиентскую директорию

```
cp /opt/easy-rsa/pki/issued/client-1.crt /opt/easy-rsa/pki/ta.key /opt/easy-rsa/pki/ca.crt ~/clients/client-1/
```

В итоге в клиентской директории вы должны получить такой минимальный набор файлов

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

Обязательно проверьте это командой

```
ls -la ~/clients/client-1/
```

```
toorr2p@pki-ovpn-001:~/clients/client-1$ lsa
total 28
drwxrwxr-x 2 toorr2p toorr2p 4096 Dec 18 21:39 .
drwxrwxr-x 3 toorr2p toorr2p 4096 Dec 18 21:32 ..
-rw------- 1 toorr2p toorr2p  749 Dec 18 21:39 ca.crt
-rw------- 1 toorr2p toorr2p 2791 Dec 18 21:37 client-1.crt
-rw------- 1 toorr2p toorr2p  440 Dec 18 21:12 client-1.req
-rw-rw-r-- 1 toorr2p toorr2p 2448 Dec 18 21:12 gpgpub.key
-rw------- 1 toorr2p toorr2p  636 Dec 18 21:39 ta.key
```

[4] Импорт публичного GPG ключа клиента

Находясь в клиентской директории, нужно импортировать публичный ключ клиента, из файла с расширением .key, командой

```
gpg --import gpgpub.key
```

Просмотреть и удостоверится о том что данный ключ пренадлежит клиенту можно командой (которая выведет отпечаток ключа)

```
gpg --fingerprint client1@gmail.com
```

пример

```
toorr2p@pki-ovpn-001:~/clients/client-1$ gpg --import gpgpub.key
gpg: directory '/home/toorr2p/.gnupg' created
gpg: keybox '/home/toorr2p/.gnupg/pubring.kbx' created
gpg: /home/toorr2p/.gnupg/trustdb.gpg: trustdb created
gpg: key 0D29877BCA9EF39E: public key "Client1 <client1@gmail.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
toorr2p@pki-ovpn-001:~/clients/client-1$ gpg --fingerprint client1@gmail.com
pub   rsa3072 2023-12-18 [SC] [expires: 2025-12-17]
      6320 6BF6 4DF5 9ED2 E839  D18F 0D29 877B CA9E F39E
uid           [ unknown] Client1 <client1@gmail.com>
sub   rsa3072 2023-12-18 [E] [expires: 2025-12-17]
```

как видно из команд email client1@gmail.com является указателем на ключ в системе после импорта

[5] Сборка архива и его шифрование GPG

После того как в клиентской директории содержится все необходимые файлы, нужно запаковать их в архив

```
tar -zcvf client-1_sig.tar.gz ./
```

```
toorr2p@pki-ovpn-001:~/clients/client-1$ tar -zcvf client-1_sig.tar.gz ./
./
./gpgpub.key
./ta.key
./ca.crt
./client-1.crt
./client-1.req
tar: .: file changed as we read it
toorr2p@pki-ovpn-001:~/clients/client-1$ lsa
total 36
drwxrwxr-x 2 toorr2p toorr2p 4096 Dec 18 21:52 .
drwxrwxr-x 3 toorr2p toorr2p 4096 Dec 18 21:32 ..
-rw------- 1 toorr2p toorr2p  749 Dec 18 21:39 ca.crt
-rw------- 1 toorr2p toorr2p 2791 Dec 18 21:37 client-1.crt
-rw------- 1 toorr2p toorr2p  440 Dec 18 21:12 client-1.req
-rw-rw-r-- 1 toorr2p toorr2p 4869 Dec 18 21:52 client-1_sig.tar.gz
-rw-rw-r-- 1 toorr2p toorr2p 2448 Dec 18 21:12 gpgpub.key
-rw------- 1 toorr2p toorr2p  636 Dec 18 21:39 ta.key
```

После чего этот архив нужно зашифровать используя команду

```
gpg --encrypt --sign --armor -r client1@gmail.com client-1_sig.tar.gz
```

Где:

- **-r** - параметр указывающий на получателя (зашифрованного сообщения, файла)
- client1@gmail.com - указатель на ключ клиента, который сможет прочесть-расшифровать файл или сообщение
- client-1_sig.tar.gz - файл который шифруем

Шифруем файл

```
toorr2p@pki-ovpn-001:~/clients/client-1$ lsa
total 36
drwxrwxr-x 2 toorr2p toorr2p 4096 Dec 18 21:52 .
drwxrwxr-x 3 toorr2p toorr2p 4096 Dec 18 21:32 ..
-rw------- 1 toorr2p toorr2p  749 Dec 18 21:39 ca.crt
-rw------- 1 toorr2p toorr2p 2791 Dec 18 21:37 client-1.crt
-rw------- 1 toorr2p toorr2p  440 Dec 18 21:12 client-1.req
-rw-rw-r-- 1 toorr2p toorr2p 4869 Dec 18 21:52 client-1_sig.tar.gz
-rw-rw-r-- 1 toorr2p toorr2p 2448 Dec 18 21:12 gpgpub.key
-rw------- 1 toorr2p toorr2p  636 Dec 18 21:39 ta.key
toorr2p@pki-ovpn-001:~/clients/client-1$ gpg --encrypt --sign --armor -r client1@gmail.com client-1_sig.tar.gz
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2025-12-17
gpg: D51FC8EB90E9BB6A: There is no assurance this key belongs to the named user

sub  rsa3072/D51FC8EB90E9BB6A 2023-12-18 Client1 <client1@gmail.com>
 Primary key fingerprint: 6320 6BF6 4DF5 9ED2 E839  D18F 0D29 877B CA9E F39E
      Subkey fingerprint: 2264 3637 9536 BD18 7982  6B76 D51F C8EB 90E9 BB6A

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
```

И на выходе получаем его зашифрованную копию client-1_sig.tar.gz.asc которую можно отправить клиенту

```
toorr2p@pki-ovpn-001:~/clients/client-1$ lsa
total 44
drwxrwxr-x 2 toorr2p toorr2p 4096 Dec 18 21:59 .
drwxrwxr-x 3 toorr2p toorr2p 4096 Dec 18 21:32 ..
-rw------- 1 toorr2p toorr2p  749 Dec 18 21:39 ca.crt
-rw------- 1 toorr2p toorr2p 2791 Dec 18 21:37 client-1.crt
-rw------- 1 toorr2p toorr2p  440 Dec 18 21:12 client-1.req
-rw-rw-r-- 1 toorr2p toorr2p 4869 Dec 18 21:52 client-1_sig.tar.gz
-rw-rw-r-- 1 toorr2p toorr2p 7926 Dec 18 21:59 client-1_sig.tar.gz.asc
-rw-rw-r-- 1 toorr2p toorr2p 2448 Dec 18 21:12 gpgpub.key
-rw------- 1 toorr2p toorr2p  636 Dec 18 21:39 ta.key
```

не шифрованный архив лучше сразу удалить

```
rm -f client-1_sig.tar.gz
```

Полученный файл client-1_sig.tar.gz.asc, клиент сможет расшифровать своим приватным ключом, и выполнить все остальные действия по генерации конфигурационного файла .ovpn с помощью клиентской инструкции.