openvpn

OpenVPN — свободная реализация технологии виртуальной частной сети (VPN) с открытым исходным кодом для создания зашифрованных каналов типа точка-точка или сервер-клиенты между компьютерами. Она позволяет устанавливать соединения между компьютерами, находящимися за NAT и сетевым экраном, без необходимости изменения их настроек.

```
# yum -y install epel-release
# yum -y install openvpn
```

Доустановим утилиты:

```
# yum install wget unzip zip
```

устанавливаем утилиту Easy-RSA:

```
# cd /etc/openvpn/keys
# wget https://github.com/OpenVPN/easy-rsa/archive/master.zip
# unzip master.zip
# cd /etc/openvpn/keys/easy-rsa-master/easyrsa3
```

Создаем структуру публичных PKI ключей:

```
# mv vars.example vars
# ./easyrsa init-pki
```

Создайте удостоверяющий центр CA:

```
# ./easyrsa build-ca
```

Не забудьте указанный пароль. Его нужно будет вводить каждый раз при создании нового сертификата openvpn.

Мы получили 2 ключа:

/etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/private/**ca.key**
/etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/**ca.crt**

Первый ключ секретный, его нужно оставить на сервере и никому не отдавать. Второй — открытый, его мы будем вместе с

пользовательскими сертификатами передавать клиентам.

Создаем запрос сертификата для сервера без пароля с помощью опции **nopass**, иначе придется вводить пароль с консоли при каждом запуске сервера:

```
# ./easyrsa gen-req server nopass
```

Подписываем запрос на получение сертификата у нашего CA:

```
# ./easyrsa sign-req server server
```

В процессе работы скрипта вводим пароль от CA, который указывали раньше и отвечаем на вопрос yes. Мы получили подписанный удостоверяющим центром сертификат для сервера – /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/issued/**server.crt**

Нам еще пригодится ключ Диффи-Хелмана, генерируем его:

```
# ./easyrsa gen-dh
```

По завершению работы скрипта получаем файл dh сертификата – /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/**dh.pem**.

Копируем в папку /etc/openvpn все необходимые для работы openvpn сервера ключи:

```
# cp pki/ca.crt /etc/openvpn/ca.crt
# cp pki/dh.pem /etc/openvpn/dh.pem
# cp pki/issued/server.crt /etc/openvpn/server.crt
# cp pki/private/server.key /etc/openvpn/server.key
```

Создадим ключ для клиента openvpn (nopass – без пароля, но лучше с паролем):

```
# ./easyrsa gen-req user1 nopass
# ./easyrsa sign-req client user1
```

Процедура аналогична созданию сертификата для сервера. Так же вводим пароль (pass-фразу сервера), отвечаем yes. В результате получаем подписанный сертификат клиента:

/etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/issued/**user1.crt**
/etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/private/user1.**key**

Создание статического ключа HMAC

Для создания ключа HMAC используйте команду openvpn с опциями –genkey и –secret:

```
# cd /etc/openvpn
# openvpn --genkey --secret ta.key
```

Клиенту, которым у нас является шлюз филиала нужно будет передать следующий набор файлов – **user1.crt, user1.key, ca.crt, ta.key**.

Теперь приступаем к настройке. Создаем файл конфигурации openvpn:

```
# nano /etc/openvpn/server.conf
port 1194 # я предпочитаю использовать нестандартные порты для работы
proto udp # протокол может быт и tcp, если есть необходимость в этом
dev tun

ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key
dh /etc/openvpn/dh.pem

# Проверка, не отозван ли сертификат клиента
crl-verify /etc/openvpn/crl.pem

# Включаем TLS
tls-auth /etc/openvpn/ta.key 0
tls-server
tls-timeout 120
auth SHA512
cipher AES-256-CBC
#auth MD5
```

```
#cipher BF-CBC

server 10.8.0.0 255.255.255.0 # подсеть для туннеля, может быть любой
route 10.8.0.0 255.255.255.252 # указываем подсеть, к которой будем обращаться через vpn rab
push "route 192.168.113.0 255.255.255.0" # передаем маршрут клиентам

# Для доступа клиентов через удаленный шлюз в Internet
# Если не нужен - закоментировать
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 10.8.0.1"

ifconfig-pool-persist ipp.txt # файл с записями соответствий clinet - ip
client-to-client # позволяет клиентам openvpn подключаться друг к другу
client-config-dir /etc/openvpn/ccd # директория с индивидуальными настройками клиентов

keepalive 10 120

# сжатие трафика
comp-lzo
persist-key
persist-tun

max-clients 100
user nobody
group nobody

status /var/log/openvpn/openvpn-status.log
log /var/log/openvpn/openvpn.log
verb 4
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
```

Создаем необходимые директории:

```
# mkdir /etc/openvpn/ccd && mkdir /var/log/openvpn
```

Создаем файл конфигурации клиента в папке, указанной в параметре client-config-dir :

```
# nano /etc/openvpn/ccd/user1
iroute 192.168.113.0 255.255.255.0
```

Здесь **user1** – имя сертификата пользователя.

Если вам нужно *объединить две разные локальные сети в одну условно общую, но с разной адресацией*, то вам нужен **tun**. То есть в нашем случае мы объединяем две сети 192.168.1.0/24 и 192.168.60.0/24 для взаимного совместного доступа.

Если же у вас стоит задача *объединить 2 удаленные сети в единое адресное пространство*, например сделать и в офисе и в филиале единую сеть *192.168.10.0/24*, то тогда бы мы использовали **tap** интерфейс и указывали бы на компьютерах в обоих сетях не пересекающиеся адреса из одной подсети. В таком состоянии openvpn работает в режиме *моста*.

Запускаем сервер:

```
# systemctl start openvpn@server
# systemctl enable openvpn@server
```

```
# netstat -tulnp | grep 1194
udp 0 0 0.0.0.0:1194 0.0.0.0:* 17719/openvpn
```

```
# ip a
 tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500
qdisc pfifo_fast state UNKNOWN group default qlen 100
 link/none
 inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
    valid_lft forever preferred_lft forever
```

Статический маршрут:

```
# netstat -rn
Kernel IP routing table
Destination    Gateway         Genmask         Flags MSS
Window irtt Iface
```

```
0.0.0.0         22.44.19.162    0.0.0.0         UG      0 0
0 enp4s0
10.8.0.0        10.8.0.2        255.255.255.252 UG      0 0
0 tun0
10.8.0.0        10.8.0.2        255.255.255.0   UG      0 0
0 tun0
10.8.0.2        0.0.0.0         255.255.255.255 UH      0 0
0 tun0
192.168.113.0   0.0.0.0         255.255.255.0   U       0 0
0 enp5s0
22.44.19.160    0.0.0.0         255.255.255.240 U       0 0
0 enp4s0
```

Траффик из подсети 10.8.0.0/24 будет маршрутизироваться в тоннель.

Клиентские ключи

Создадим ключ для клиента openvpn:

```
# cd /etc/openvpn/keys/easy-rsa-master/easyrsa3
# ./easyrsa gen-req user1 nopass
# ./easyrsa sign-req client user1
```

Опять можно отметить, что могут быть ситуации, когда лучше ключ клиента защищать паролем, хотя бы несложным. Тогда всякий раз при подключении к VPN необходимо будет ввести пароль. Это может быть удобным, если вы не хотите, например, чтобы ваш ребенок случайно подключился к вашей рабочей сети и натворил делов. Для этого просто не надо указывать “nopass” в конце команды выше.

В итоге мы получим два файла:

Публичный сертификат клиента:

```
/etc/openvpn/keys/easy-rsa-
master/easyrsa3/pki/issued/user1.crt
```

Приватный ключ клиента:

```
/etc/openvpn/keys/easy-rsa-
master/easyrsa3/pki/private/user1.key
```

Клиенту вместе с конфигом (см ниже) нужно будет передать копии следующих файлов:

user1.crt;
user1.key;
ca.crt;
ta.key;

которые все используются в клиентском конфиге. **Никакие иные файлы, кроме тех, которые указаны в конфиге клиента, передавать клиенту не надо!**

Дальше создаем файл конфигурации для этого клиента:

```
# cd /etc/openvpn/ccd
# nano user1
push "route 192.168.113.0 255.255.255.0"
#push "route 192.168.50.0 255.255.255.0"
```

Этими параметрами мы передаем клиенту маршруты к обоим сетям офисов. Если нужно подключать клиента только к какой-то одной сети, то оставляйте одну сеть, вторую удаляйте.

IPTABLES

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT ! -i lo -d 127.0.0.0/8 -j REJECT

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# OPENVPN
iptables -A INPUT -i tun+ -j ACCEPT
iptables -A FORWARD -i enp5s0 -o tun+ -j ACCEPT
iptables -A FORWARD -i enp5s0 -o enp4s0 -j ACCEPT

iptables -A FORWARD -i enp5s0 -o enp4s0 -j ACCEPT
.....
# NAT
iptables -t nat -F POSTROUTING
```

```
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o enp4s0 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 192.168.113.0/24 -o enp4s0 -j MASQUERADE
....
# OPENVPN
iptables -A INPUT -i enp4s0 -p udp --dport 1194 -j ACCEPT
....
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

Настройка Windows клиента

Теперь нужно скачать openvpn client под нашу *версию windows*, запускать **установщик** нужно обязательно с **правами администратора**.

```
cat client.ovpn
```

```
dev tun
proto udp
remote 194.44.219.161
port 1194
client
resolv-retry infinite
ca ca.crt
cert user1.crt
key user1.key
remote-cert-eku "TLS Web Server Authentication"
remote-cert-tls server
tls-client
tls-auth ta.key 1
auth SHA512
cipher AES-256-CBC
comp-lzo
persist-key
persist-tun
mute-replay-warnings
verb 3
```

Сохраняем конфигурацию под именем client.ovpn в папку

C:\Program Files\OpenVPN\config, туда же копируем файлы сертификатов и запускаем OpenVPN GUI от имени **администратора**!

Правой мышкой в трее – подключиться. Теперь при подключении нам доступны ресурсы сети 192.168.113.0/24 и интернет. При заходе на страничку, например, 2ip.ru – сайт покажет IP рабочего сервера.

Создание списка отзывов сертификатов

Если сотрудник уволился, необходимо заблокировать его доступ в сеть VPN компании. Специально для этой цели в OpenVPN предусмотрен список отзыва сертификатов CRL. Создайте его такой командой:

```
# cd /etc/openvpn/keys/easy-rsa-master/easyrsa3
# ./easyrsa gen-crl
```

У вас будет запрошен пароль доступа к приватному ключу ca.key удостоверяющего центра. Список отзыва сертификатов будет создан в файле /home/ca/easy-rsa-master/easyrsa3/pki/crl.pem.

Если нужно заблокировать выданный ранее сертификат, воспользуйтесь следующей командой:

```
# ./easyrsa revoke user1
```

Здесь мы отозвали сертификат для клиента user1. Далее нужно скопировать новый файл CRL на сервер OpenVPN и перезапустить демон OpenVPN.

Обновление списка отзывов сетрификатов

По умолчанию, продолжительность жизни списка отзывов сертификатов crl.pem равна 180 дням, за это отвечает переменная:

```
# cat /etc/openvpn/keys/easy-rsa-master/easyrsa3/vars

#set_var EASYRSA_CRL_DAYS 180
```

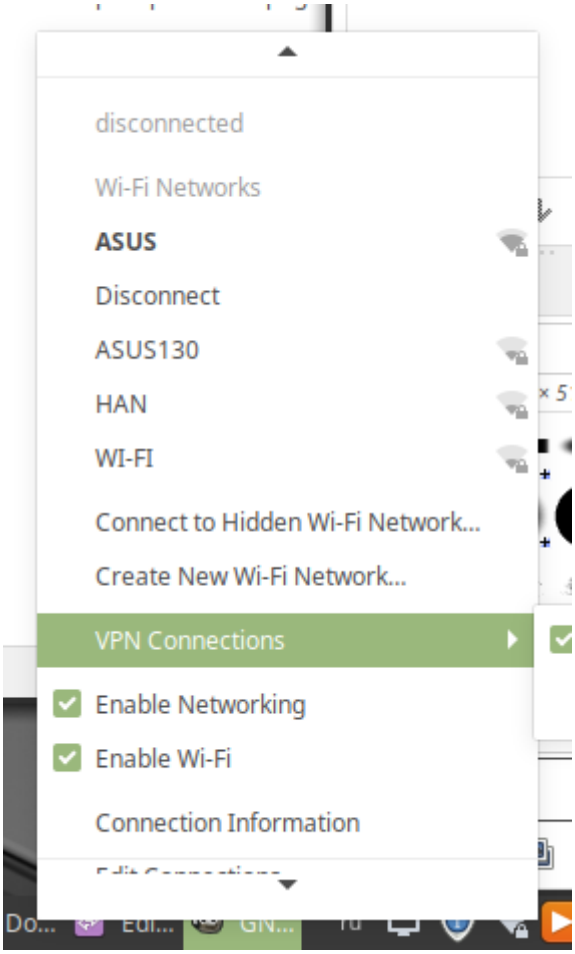
Поэтому, по истечению срока указанного в vars, нужно будет

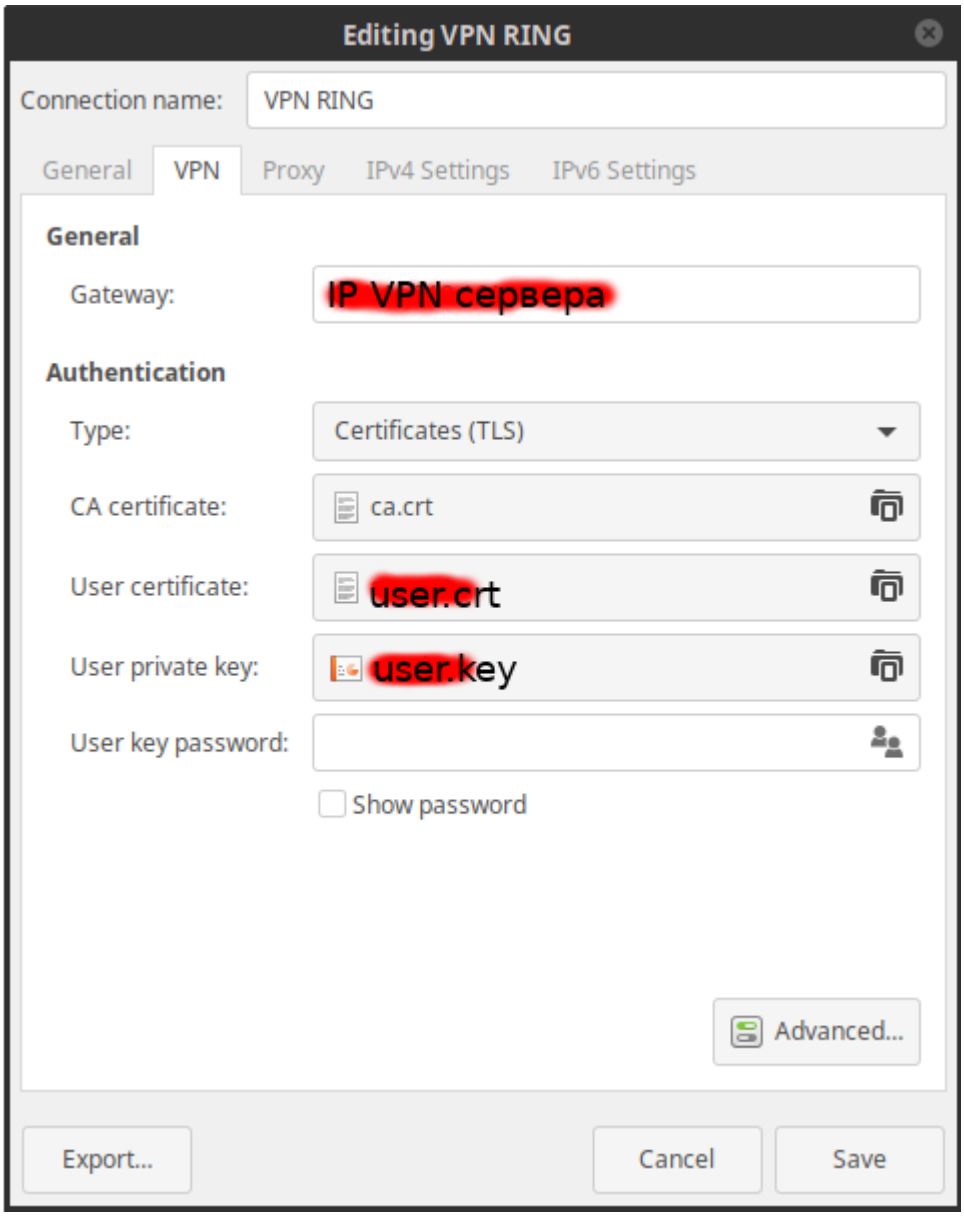
обновить список (понадобится парольная фраза для доступа к приватному ключу ca.key удостоверяющего центра):

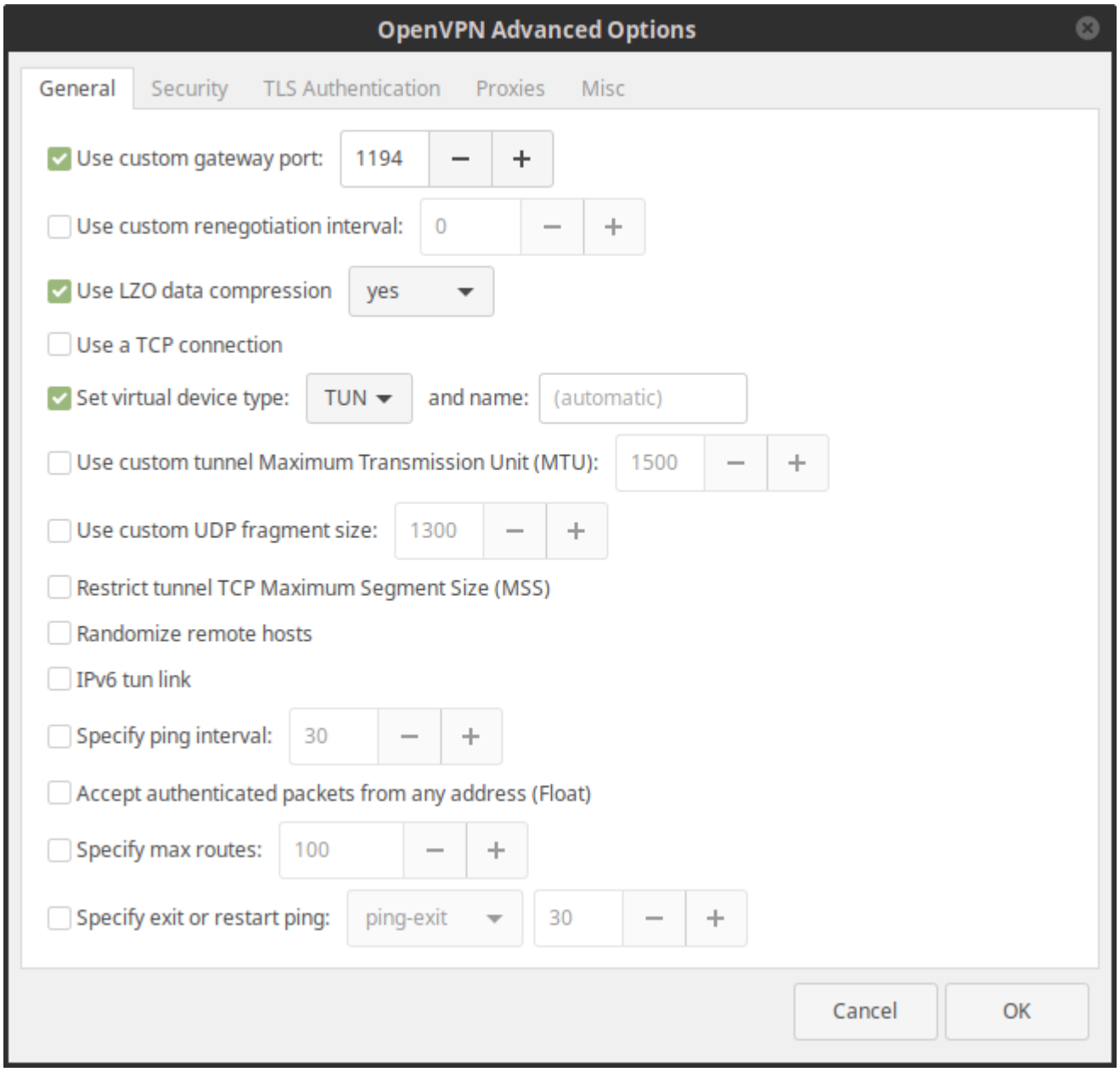
```
# cd /etc/openvpn/keys/easy-rsa-master/easyrsa3/
# ./easyrsa gen-crl
# cp ./pki/crl.pem /etc/openvpn/crl.pem
# systemctl restart openvpn@server
```

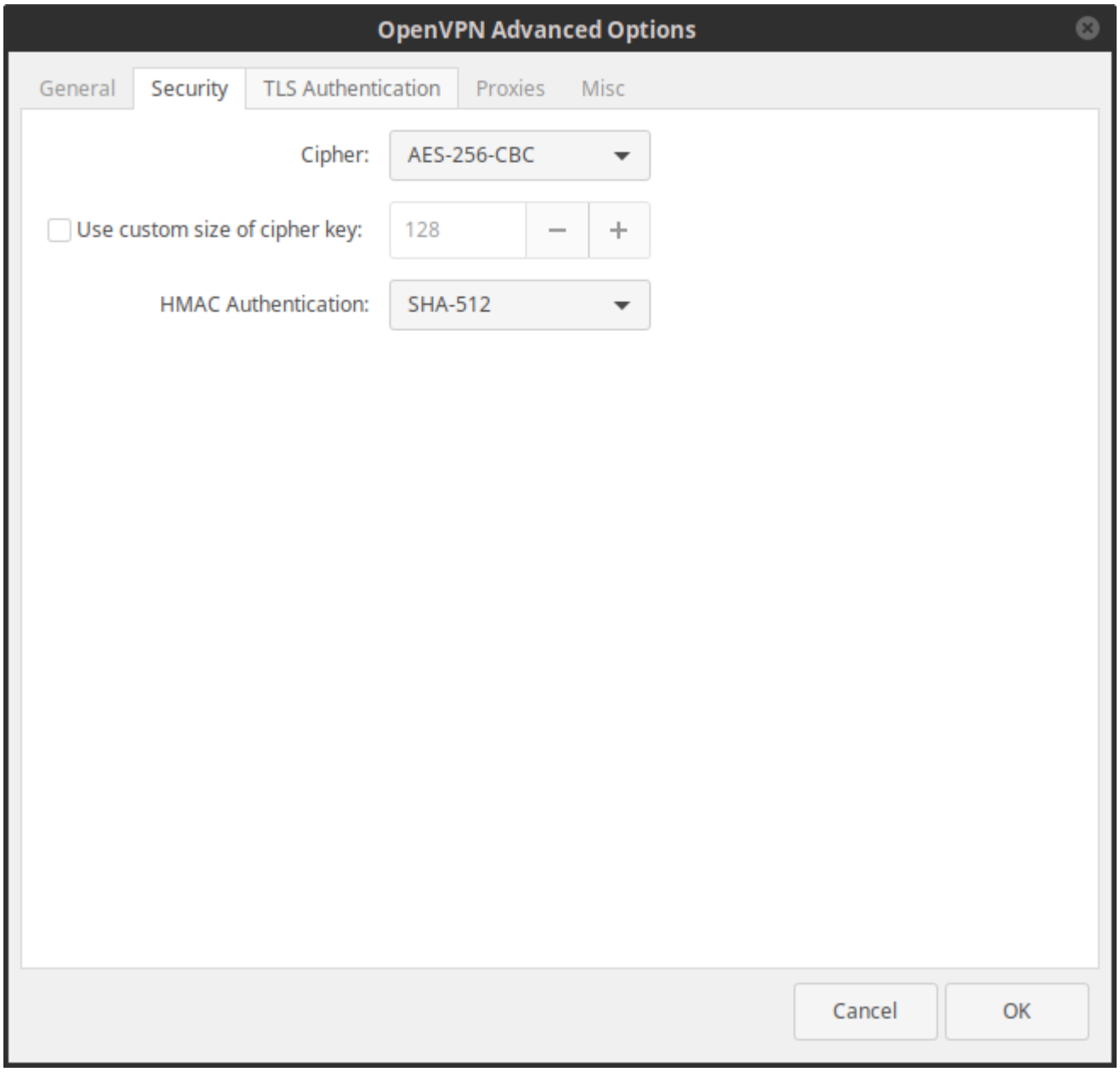
Клиент для Linux (Mint)

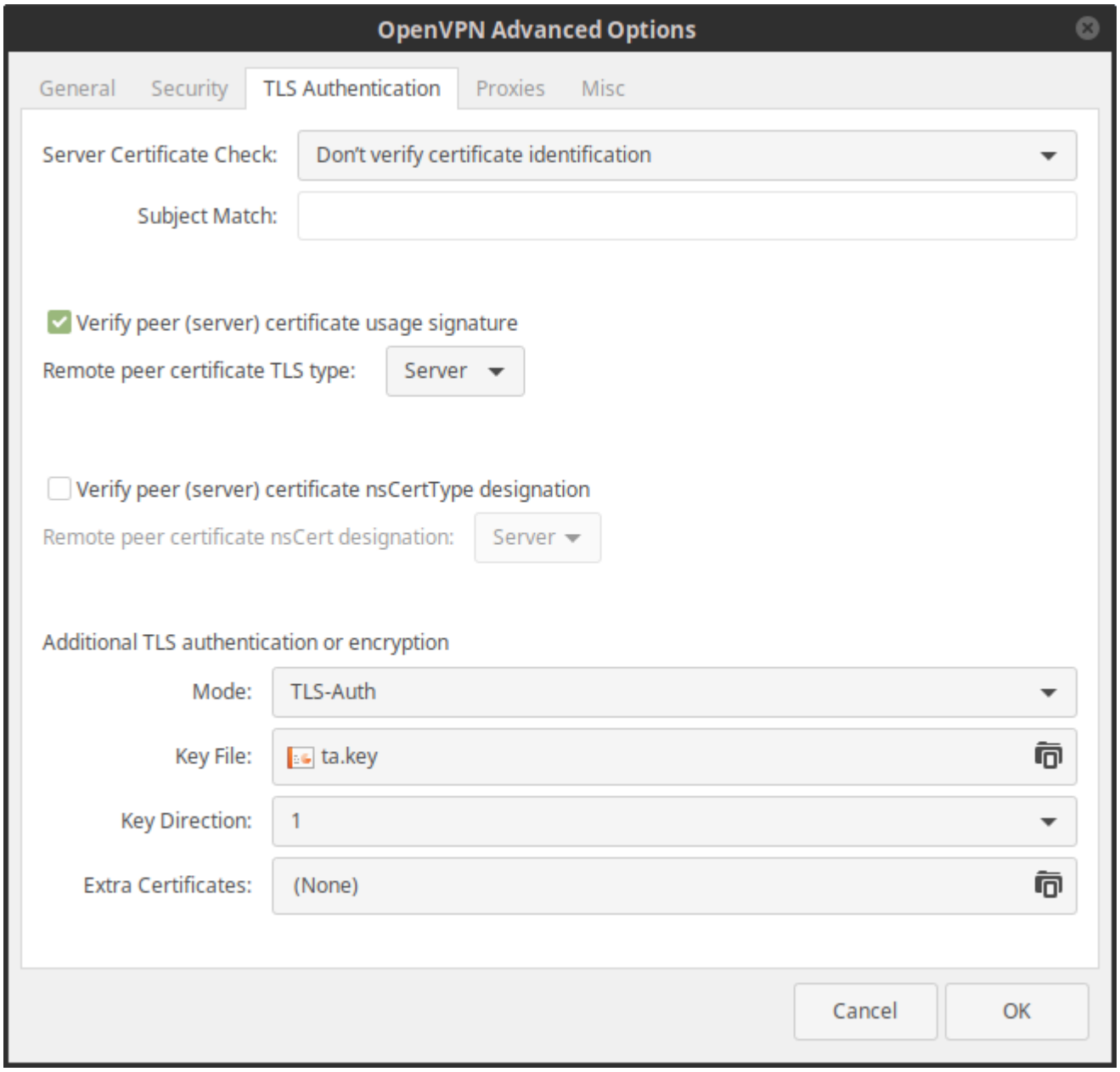
Выбираем VPN Connection и заполняем настройки:




Editing VPN RING
Connection name:
VPN RING
General
VPN
Proxy
IPv4 Settings
IPv6 Settings
General
Gateway:
IP VPN сервера
Authentication
Type:
Certificates (TLS)
CA certificate:
ca.crt
User certificate:
user.crt
User private key:
user.key
User key password:
Show password
Advanced...
Export...
Cancel
Save


OpenVPN Advanced Options
General
Security
TLS Authentication
Proxies
Misc
Use custom gateway port:
1194
Use custom renegotiation interval:
0
Use LZO data compression
yes
Use a TCP connection
Set virtual device type:
TUN
and name:
(automatic)
Use custom tunnel Maximum Transmission Unit (MTU):
1500
Use custom UDP fragment size:
1300
Restrict tunnel TCP Maximum Segment Size (MSS)
Randomize remote hosts
IPv6 tun link
Specify ping interval:
30
Accept authenticated packets from any address (Float)
Specify max routes:
100
Specify exit or restart ping:
ping-exit
30
Cancel
OK


OpenVPN Advanced Options
General
Security
TLS Authentication
Proxies
Misc
Cipher:
AES-256-CBC
Use custom size of cipher key:
128
HMAC Authentication:
SHA-512
Cancel
OK



Терминология

Файлы для сервера OpenVPN.

Файл	Описание
dh.pem	Файл Диффи-Хелмана для защиты трафика от расшифровки
ca.crt	Сертификат удостоверяющего центра CA
server.crt	Сертификат сервера OpenVPN
server.key	Приватный ключ сервера OpenVPN, **секретный**

crl.pem	Список отзыва сертификатов CRL
ta.key	Ключ HMAC для дополнительной защиты от DoS-атак и флуда

Настроить openvpn на CentOS 7
https://bozza.ru/art-269.html
https://1cloud.ru/help/linux/openvpnserver_debian7_ubuntu12
Руководство по установке и настройке OpenVPN