

Настройка OpenVPN-сервера на роутерах Mikrotik в RouterOS 7

 interface31.ru/tech_it/2023/02/nastroyka-openvpn-servera-na-routerah-mikrotik-v-routeros-7.html

Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Настройка OpenVPN-сервера на роутерах Mikrotik в RouterOS 7

OpenVPN, при всех ограничениях и недостатках реализации этой технологии в RouterOS пользуется высокой популярностью у владельцев роутеров Mikrotik. И хотя, на наш взгляд, это не самое оптимальное решение для данного оборудования, игнорировать его популярность мы не можем, поэтому решили рассмотреть настройку OpenVPN сервера в среде новой версии RouterOS 7, где были сделаны ряд доработок и изменений. В любом случае OpenVPN - только инструмент его выбор и применение полностью зависит от специалиста, который должен знать и понимать все сильные и слабые стороны.



Онлайн-курс по MikroTik

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на [углубленном курсе по администрированию MikroTik](#). Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.

Данная статья предназначена для RouterOS 7.x, для настройки OpenVPN сервера в RouterOS 6.x [перейдите к статье](#).

В RouterOS 7 был серьезно доработан OpenVPN, теперь он наконец-то поддерживает транспорт UDP, но остался ряд принципиальных ограничений:

- Нет LZO сжатия
- Нет TLS аутентификации
- Нельзя передавать опции на клиент
- Требуется аутентификация по логину/паролю

В целом, реализация OpenVPN в RouterOS остается на достаточно примитивном уровне, и мы еще раз советуем трезво взвесить все за и против перед внедрением данной технологии на базе Mikrotik.

Подготовка роутера

OpenVPN, как и любой другой продукт, использующий TLS-шифрование, чувствителен к правильному времени, поэтому сразу настроим синхронизацию часов роутера с серверами точного времени. Откроем System - NTP Client и укажем сервера для синхронизации, мы используем сервера www.ntppool.org.

В терминале это же действие можно выполнить так:

```
/system ntp client
set enabled=yes
```

```
/system ntp client servers
add address=0.ru.pool.ntp.org
add address=1.ru.pool.ntp.org
```

Не забудьте убедиться, что синхронизация времени прошла успешно.

Создание ключей и сертификатов

В данном случае у нас есть два пути: создать все необходимые ключи и сертификаты в стороннем CA, либо использовать возможности RouterOS, в этом разделе мы рассмотрим последний вариант. Но учтите, что CA на RouterOS имеет свои особенности, которые мы рассмотрели в статье:

[Особенности эксплуатации CA на роутерах Mikrotik: резервное копирование, экспорт и импорт сертификатов](#)

Всю дальнейшую работу будем выполнять в **System - Certificate** и начнем с создания собственного центра сертификации, сгенерировав корневую ключевую пару.

New Certificate

General | Key Usage | Status

Name: CA

Issuer:

Country: RU

State: 31

Locality: BEL

Organization: Interface LLC

Unit:

Common Name: CA

Subject Alt. Name:

Key Type: RSA

Key Size: 2048

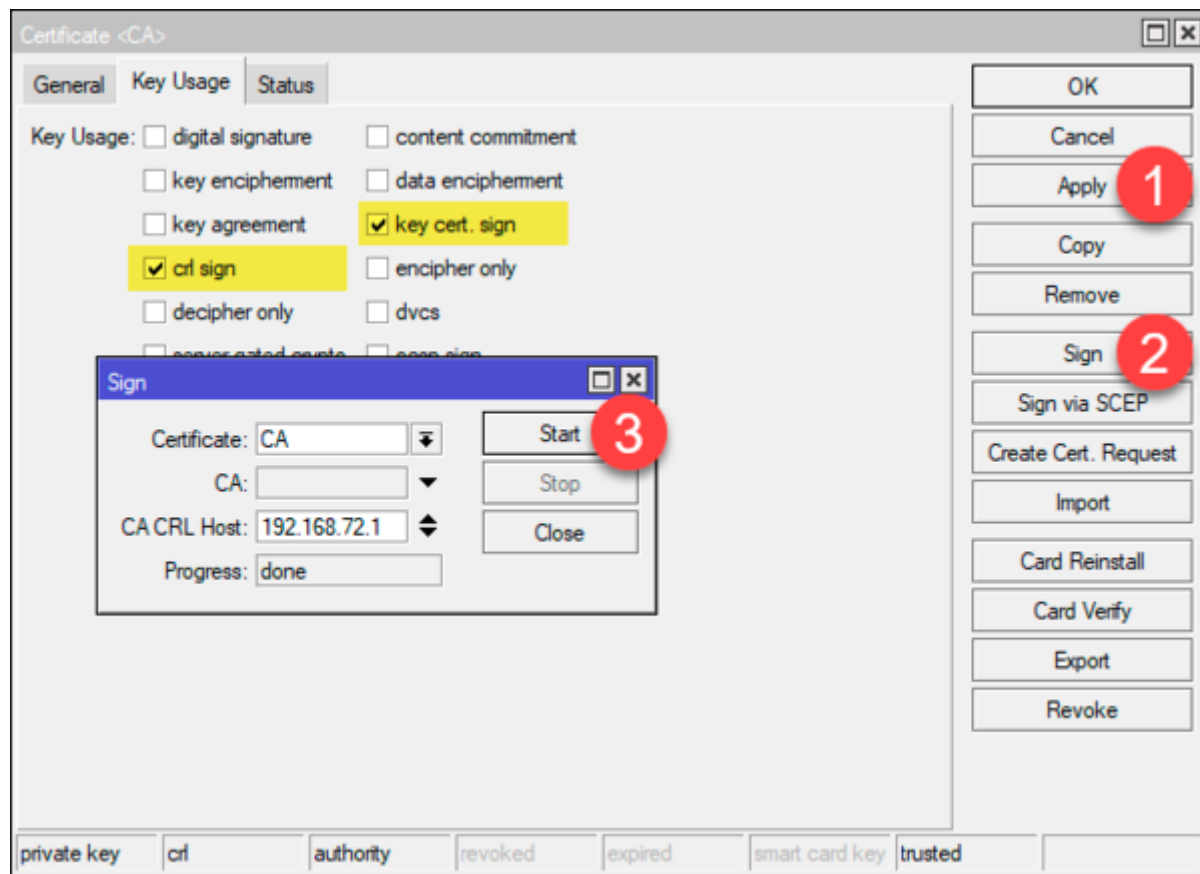
Days Valid: 3650

private key | crl | authority | revoked | expired | smart card... | trusted

OK
Cancel
Apply
Copy
Remove
Sign
Sign via SCEP
Create Cert. Request
Import
Card Reinstall
Card Verify
Export
Revoke

Обязательные поля отмечены нами красным, это **Name** и **Common Name** - **CA** и срок действия - **Days Valid** - **3650** или 10 лет, для локального центра сертификации это нормально. Выделенные зеленым поля содержат информацию о владельце сертификата и не обязательны, но их заполнение является правилом хорошего тона и помогает быстро понять, что это за сертификаты и кем они выпущены.

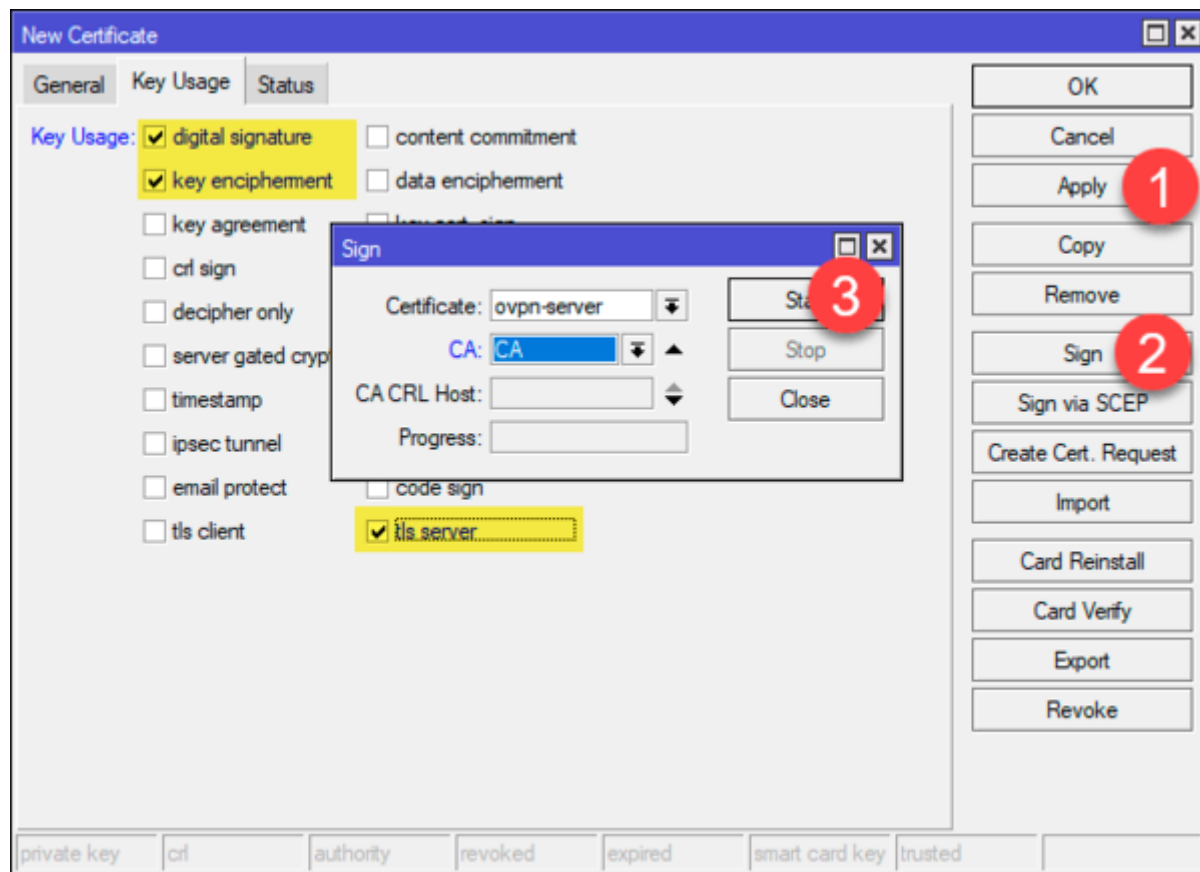
Перейдем на закладку **Key Usage** и укажем **crl sign** и **key cert. sign**, остальные флаги следует снять, затем нажмем кнопку **Apply**, применив изменения и подпишем сертификат нажав **Sign**. В открывшемся окне следует указать адрес, на котором будет опубликован список отозванных сертификатов - CRL, если это будет единственный OpenVPN сервер, то можно указать 127.0.0.1, иначе лучше указать один из реальных адресов роутера, в нашем случае внутренний. После чего нажимаем **Start** и ожидаем окончания процесса подписи.



Эти же действия в терминале:

```
/certificate  
add name=CA country="RU" state="31" locality="BEL" organization="Interface LLC"  
common-name="CA" key-size=2048 days-valid=3650 key-usage=crl-sign,key-cert-sign  
sign ca ca-crl-host=192.168.72.1
```

Затем выпустим ключевую пару сервера. Закладка **General** нового сертификата заполняется аналогично, только в полях **Name** и **Common Name** указываем **ovpn-server** (или любое наименование на свое усмотрение). На вкладке **Key Usage** указываем **digital-signature**, **key-encipherment** и **tls-server**. Затем подписываем сертификат, для этого в поле **CA** выбираем созданный ранее сертификат **CA**.



В терминале:

```
/certificate
add name=ovpn-server country="RU" state="31" locality="BEL"
organization="Interface LLC" common-name="ovpn-server" key-size=2048 days-
valid=3650 key-usage=digital-signature,key-encipherment,tls-server
sign ovpn-server ca="CA"
```

После этого нужно отдельно выпустить сертификат для каждого клиента, в полях **Name** и **Common Name** укажите наименование клиента, лучше выбирать осмысленные имена, чтобы было легко понять кому принадлежит данный сертификат. Также следует подумать над **сроком действия сертификата**, если клиентом будет роутер в удаленном офисе, то можно также выпустить сертификат на 10 лет, а вот если клиентом будет ноутбук сотрудника на испытательном сроке, то лучше выдать его на срок испытательного срока. Выпустить новый сертификат не представляет проблемы, в то время как не отзыванный вовремя сертификат может привести к несанкционированному доступу и утечке данных.

Для клиентского сертификата на закладке **Key Usage** выберите только **tls client** и точно также подпишите сертификат.

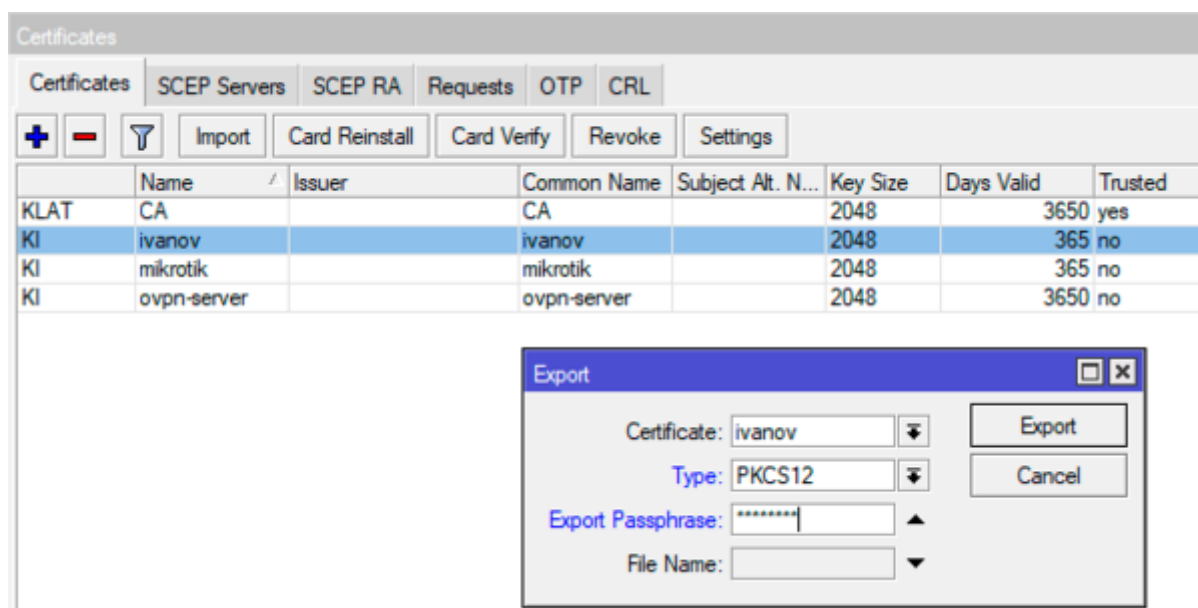
В терминале:

```
/certificate
add name=mikrotik country="RU" state="31" locality="BEL" organization="Interface
LLC" common-name="mikrotik" key-size=2048 days-valid=365 key-usage=tls-client
sign mikrotik ca="CA"
```

Обратите внимание, что мы выпустили сертификат на 1 год - 365 дней.

Если все сделано правильно, то корневой сертификат будет иметь флаги **KLAT**, а остальные - **KI**. Для передачи клиентам мы должны экспортировать: сертификат CA, сертификат клиента, закрытый ключ клиента. Для этого лучше всего использовать формат PKCS12, позволяющий разместить все сертификаты и ключи в одном контейнере.

Чтобы экспортировать сертификат щелкните на нем правой кнопкой мыши и выберите **Export**, в открывшемся окне укажите формат **Type - PKCS12** и **парольную фразу** для экспорта (минимум 8 символов) в поле **Export Passphrase**. Без указания пароля закрытые ключи выгружены не будут, и вы не сможете использовать такой сертификат для клиента.



В терминале экспортировать сертификат можно командой:

```
/certificate  
export-certificate ivanov type=pkcs12 export-passphrase=12345678
```

В данном случае мы выгрузили сертификат для клиента **ivanov** с паролем **12345678**.

Использование сертификатов выданных сторонним CA

Если вы не хотите использовать роутер в качестве удостоверяющего центра, то можете сгенерировать сертификаты самостоятельно, например, это можно сделать при помощи Easy-RSA на отдельной Linux-машине:

Создание ключей и сертификатов для OpenVPN при помощи Easy-RSA 3

В этом случае вам нужно будет импортировать следующие сертификаты и ключи:

- Корневой сертификат удостоверяющего центра (обычно ca.crt)
- Сертификат сервера

- Закрытый ключ сервера

Сертификаты и ключи требуется предварительно загрузить в роутер в разделе **Files**. Затем их можно импортировать в **System - Certificate** кнопкой **Import**. Если все сделано правильно, то корневой сертификат будет иметь ключ **T (LT)**, а сертификат сервера - **KT**.

Certificates							
<div> <div>Certificates</div> <div>SCEP Servers</div> <div>SCEP RA</div> <div>Requests</div> <div>OTP</div> <div>CRL</div> </div> <div> <div>+</div> <div>-</div> <div>Filter</div> <div>Import</div> <div>Card Reinstall</div> <div>Card Verify</div> <div>Revoke</div> <div>Settings</div> </div>							
	Name	Issuer	Common Name	Subject Alt. Name	Key Size	Days Valid	Trusted
T	ca.crt_0	CN=Easy-RSA CA	Easy-RSA CA		2048	3650	yes
KT	ovpn-ubnt.crt_0	CN=Easy-RSA CA	ovpn-ubnt	DNS:ovpn-ubnt	2048	1080	yes

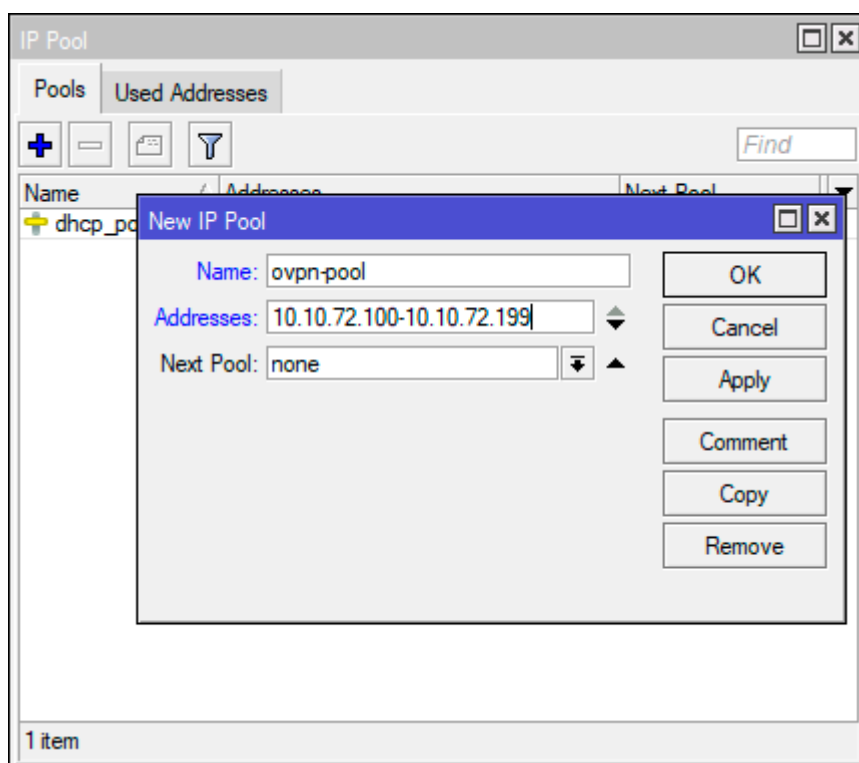
В консоли это можно сделать командой:

```
/certificate
import file-name=ca.crt
```

Где в опции **file-name** укажите имя импортируемого сертификата или ключа.

Настройка OpenVPN сервера

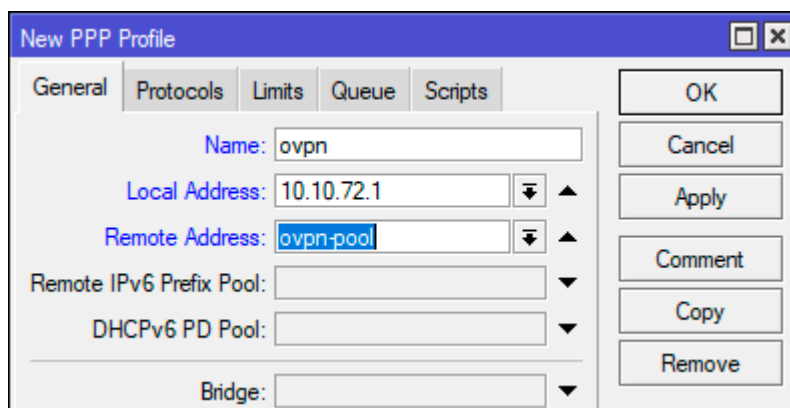
Начнем с создания пула адресов для выдачи клиентам, для этого перейдем в **IP - Pool** и создадим новый пул адресов **ovpn-pool**, в нашем случае будет использоваться диапазон **10.10.72.100 - 10.10.72.199**.



Эти же действия в терминале:

```
/ip pool
add name=ovpn-pool ranges=10.10.72.100-10.10.72.199
```

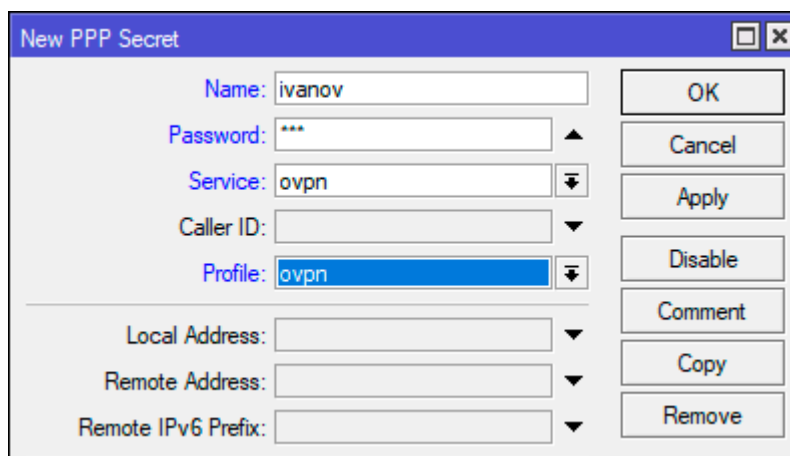
Затем перейдем в **PPP - Profiles** и создадим новый профиль. Укажем его имя **Name - ovpn**, локальный и удаленный адреса: **Local Address - 10.10.72.1**, **Remote Address - ovpn_pool**. На всякий случай напомним, что локальный адрес должен принадлежать той-же /24 сети, что и диапазон пула адресов.



В терминале:

```
/ppp profile  
add local-address=10.10.72.1 name=ovpn remote-address=ovpn-pool
```

Теперь в **PPP - Secrets** создадим учетные записи пользователей, это одна из особенностей реализации OpenVPN в RouterOS - обязательная аутентификация по логину-паролю. Мы советуем давать учетной записи такое же самое имя, как и сертификату, что позволит избежать путаницы. К паролю особых требований нет, все равно основная аутентификация производится по сертификатам. Ниже в поле **Service** указываем **ovpn**, что ограничит ее работу только с OpenVPN-сервером и в поле **Profile** выбираем созданный на предыдущем шаге профиль, в нашем случае тоже **ovpn**.



Если вы предпочитаете терминал:

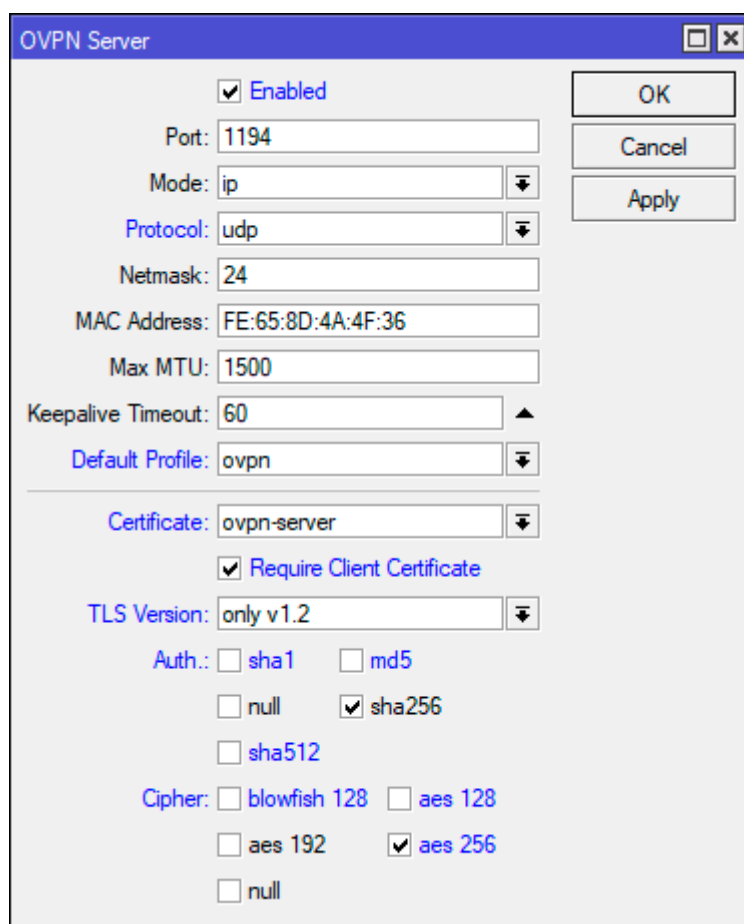
```
/ppp secret  
add name=ivanov password=123 profile=ovpn service=ovpn
```

Аналогичным способом создаем учетные записи для остальных пользователей.

После чего настроим сам OpenVPN сервер, для этого перейдем в **PPP - Interface** и нажмем на кнопку **OVPN Server**, в открывшемся окне включим службу установив флаг **Enabled**, **Protocol - udp**, **Default Profile - ovpn**, в поле **Certificate** укажем созданный нами **сертификат сервера**. Для дополнительной безопасности включим **Require Client Certificate**, в этом случае сервер будет проверять сертификат клиента на принадлежность к цепочке сертификатов CA.,

Ниже расположены настройки шифрования. Первым делом отключим устаревшие и небезопасные версии протокола TLS выбрав **TLS Version - only v1.2**, а вот в выборе шифров однозначных рекомендаций дать нельзя, опирайтесь на собственные предпочтения с оглядкой на мощность используемого устройства. Варианты и **md5** использовать по соображениям безопасности не следует. И помните, чем сильнее шифр, тем выше нагрузка на процессор устройства.

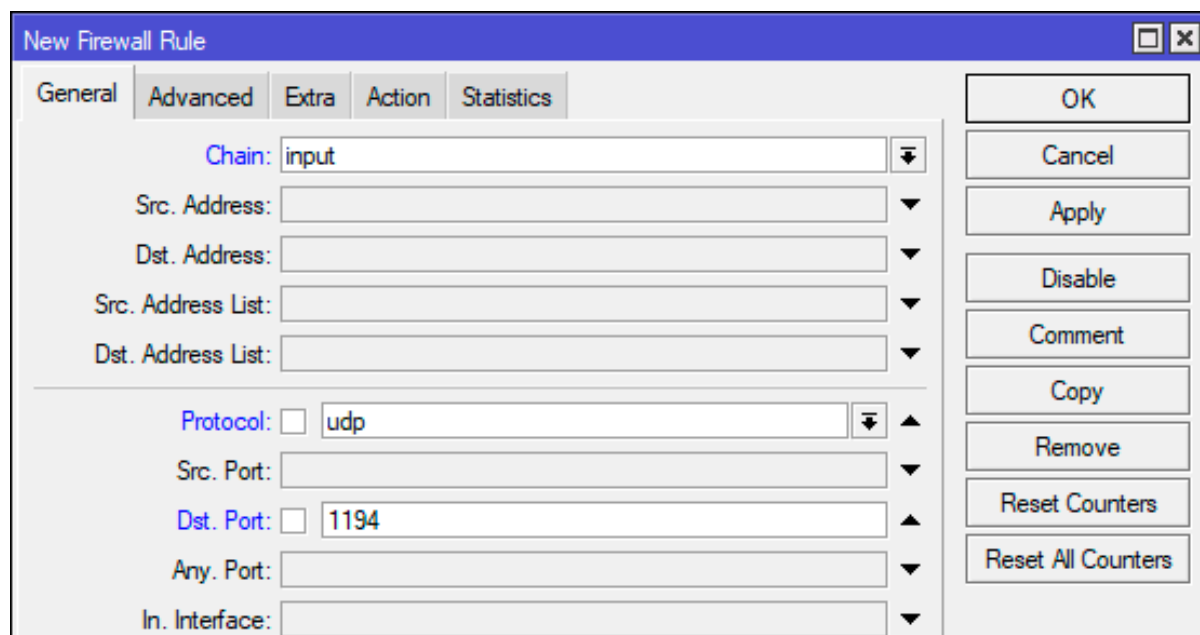
Параметры, показанные на рисунке ниже выбраны нами сугубо в ознакомительных целях.



Все вышеописанные действия в терминале:

```
/interface ovpn-server server
set auth=sha256 certificate=ovpn-server cipher=aes256 \
default-profile=ovpn enabled=yes protocol=udp require-client-certificate=yes \
tls-version=only-1.2
```

На этом настройка OpenVPN сервера закончена, осталось разрешить входящие подключения к нему. Откроем **IP - Firewall** и добавим правило: **Chain - input**, **Protocol - udp**, **Dst. Port - 1194**. Действие можно не указывать, так как по умолчанию применяется **accept**.



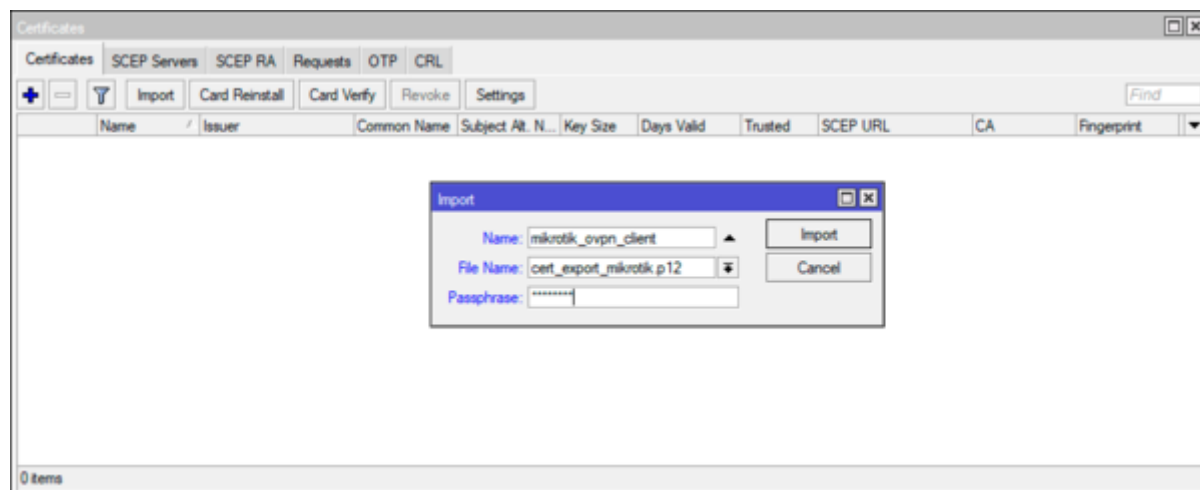
Чтобы создать правило в терминале - выполните команду:

```
/ip firewall filter  
add action=accept chain=input dst-port=1194 protocol=udp
```

Данное правило должно располагаться выше запрещающего в цепочке INPUT.

Настройка OpenVPN клиента на роутере Mikrotik

Прежде всего разместим на устройстве файл(ы) с сертификатами. Если у вас сертификат PKCS12, то достаточно просто импортировать его в **System - Certificate**, в поле **Name** желательно указать понятное имя, чтобы впоследствии было меньше путаницы:



В терминале:

```

/certificate
import file-name=cert_export_mikrotik.p12 passphrase=12345678
name="mikrotik_ovpn_client"

```

Если у вас сертификаты от стороннего CA, то последовательно импортируйте:

- Корневой сертификат удостоверяющего центра (обычно ca.crt)
- Сертификат клиента
- Закрытый ключ клиента

В результате у вас должно появиться два сертификата: сертификат CA с флагами **LT** или **T** и сертификат клиента с флагами **KT**.

Certificates						
<div> <div>Certificates</div> <div>SCEP Servers SCEP RA Requests OTP CRL</div> <div> <div>+</div> <div>-</div> <div>Filter</div> <div>Import</div> <div>Card Reinstall</div> <div>Card Verify</div> <div>Revoke</div> <div>Settings</div> </div> </div>						
	Name	Issuer	Common Name	Subject Alt. N...	Key Size	Days Valid
KT	mikrotik_ovpn_client	C=RU,ST=31,L=BEL,...	mikrotik		2048	365
LT	mikrotik_ovpn_client_1	C=RU,ST=31,L=BEL,...	CA		2048	3650

Затем перейдем в **PPP - Interface** и создадим новый интерфейс типа **OVPN Client**. В поле **Connect To** указываем адрес OpenVPN сервера, **Port - 1194**, **Mode - ip**, **Protocol - udp**. Ниже указываем учетные данные, созданные для этого пользователя на сервере. Параметры шифрования указываем аналогично тому, что вы выбрали на сервере. В поле **Certificate** выберите сертификат клиента, также можно установить флаг **Verify Server Certificate** для большей безопасности.

New Interface

General

Dial Out

Status

Traffic

Connect To: 192.168.233.145

Port: 1194

Mode: ip

Protocol: udp

User: mikrotik

Password: ***

Profile: default

Certificate: mikrotik_ovpn_client

☒ Verify Server Certificate

TLS Version: only v1.2

Auth.: sha256

Cipher: aes 256

Use Peer DNS: yes

☐ Add Default Route

☒ Disconnect Notify

OK

Cancel

Apply

Disable

Comment

Copy

Remove

Torch

enabled

running

slave

Hw. Crypto

Status:

Эти же действия в терминале:

```
/interface ovpn-client
add auth=sha256 certificate=mikrotik_ovpn_client cipher=aes256-cbc \
connect-to=192.168.233.145 name=ovpn-out1 protocol=udp tls-version=\
only-1.2 user=mikrotik verify-server-certificate=yes
```

Если все сделано правильно, то соединение будет установлено сразу после его создания.

Для того, чтобы клиенты этой сети имели доступ в сеть за сервером, нужно настроить маршрутизацию. Для этого переходим в **IP - Routes** и добавляем новый маршрут. В поле **Dst. Address** указываем сеть за сервером, в нашем случае это **192.168.72.0/24**, в поле **Gateway** укажем внутренний адрес OpenVPN сервера - **10.10.72.1**, интерфейс выхода будет подобран роутером автоматически.

Route <192.168.72.0/24->10.10.72.1>

General Status MPLS

Dst. Address: 192.168.72.0/24

Gateway: 10.10.72.1

Immediate Gateway: 10.10.72.1%ovpn-out1

Local Address:

Check Gateway:

☐ Suppress Hw Offload

Distance: 1

Scope: 30

Target Scope: 10

VRF Interface:

Routing Table: main

Pref. Source:

☐ Blackhole

OK Cancel Apply Disable Comment Copy Remove

enabled active static Hw Offload... ECMP

В терминале добавить маршрут можно следующим образом:

```
/ip route
add disabled=no dst-address=192.168.72.0/24 gateway=10.10.72.1 routing-table=main
suppress-hw-offload=no
```

Если задачи доступа в данную сеть из сети за сервером не стоит, то на этом можно закончить. Иначе возвращаемся на роутер, выполняющий роль сервера и продолжаем настройку маршрутизации там.

По умолчанию для подключившегося клиента создается динамический интерфейс, настроить маршрутизацию для него нельзя, поэтому создадим статический интерфейс и привяжем к нему подключения от данного клиента. Перейдем в **Interfaces** и создадим новый интерфейс типа **OVPN Server Binding**. В настройках укажем желаемое имя, **Name** - **mikrotik-f2**, в поле **User** - укажем пользователя, подключение которого будет привязано к этому интерфейсу - **mikrotik**.

The screenshot shows the 'New Interface' window with the following details:

- General Tab:**
 - Name:** mikrotik-f2
 - Type:** OVPN Server Binding
 - Actual MTU:** (empty)
 - User:** mikrotik
- Buttons (Right):** OK, Cancel, Apply, Disable, Comment, Copy, Remove, Torch, Reset Traffic Counters.
- Bottom Bar:** enabled, running, slave, passthrough, Hw. Crypto, Status: (dropdown menu).

В терминале используйте:

```
/interface ovpn-server  
add name=mikrotik-f2 user=mikrotik
```

Теперь можно добавить маршрут к сети за клиентом, в качестве шлюза укажите присвоенный клиенту адрес.

В терминале:

```
/ip route
add disabled=no dst-address=192.168.232.0/24 gateway=10.10.72.194 routing-
table=main suppress-hw-offload=no
```

Теперь можем проверить связь между сетями, если все сделано без ошибок, то сети должны видеть друг друга.

Настройка стандартного клиента OpenVPN на ПК

Рассмотрим настройки на примере клиента OpenVPN для Windows. Будем считать, что клиент установлен в **C:\OpenVPN**, а для хранения ключей используется директория **C:\OpenVPN\keys**.

Прежде всего разместим файл сертификатов в формате PKCS12 в директории для хранения ключей, а также создадим файл с учетными данными

C:\OpenVPN\auth.cfg и разместим в нем в разных строках логин и пароль:

```
ivanov
123
```

Где **ivanov** - имя пользователя, **123** - пароль которые мы задали для этой учетной записи на сервере.

Теперь создадим файл **C:\OpenVPN\keypass.cfg** в котором разместим парольную фразу для сертификата:

```
12345678
```

За основу конфигурационного файла мы примем стандартный шаблон **client.ovpn**, который расположен в **C:\OpenVPN\sample-config**. Его следует скопировать **C:\OpenVPN\config**, ниже будут приведены только ключевые опции, а также те, которые мы изменяем или добавляем.

Укажем режим работы - клиент, тип туннеля и протокол:

```
client
dev tun
proto udp
```

Адрес и порт сервера:

```
remote 192.168.233.145 1194
```

Убеждаемся в наличии опций:

```
persist-key
persist-tun
```

Затем заменяем весь блок с ключами и сертификатами:

```
ca ca.crt
cert client.crt
key client.key
```

Строками:

```
pkcs12 C:\\OpenVPN\\keys\\cert_export_ivanov.p12
auth-user-pass C:\\OpenVPN\\auth.cfg
askpass C:\\OpenVPN\\keypass.cfg
```

Проверяем опцию:

```
remote-cert-tls server
```

И добавляем маршрут в сеть за сервером:

```
route 192.168.72.0 255.255.255.0 10.10.72.1
```

Затем закомментируем:

```
#tls-auth ta.key 1
```

Укажем используемые шифры:

```
auth SHA256
cipher AES-256-CBC
```

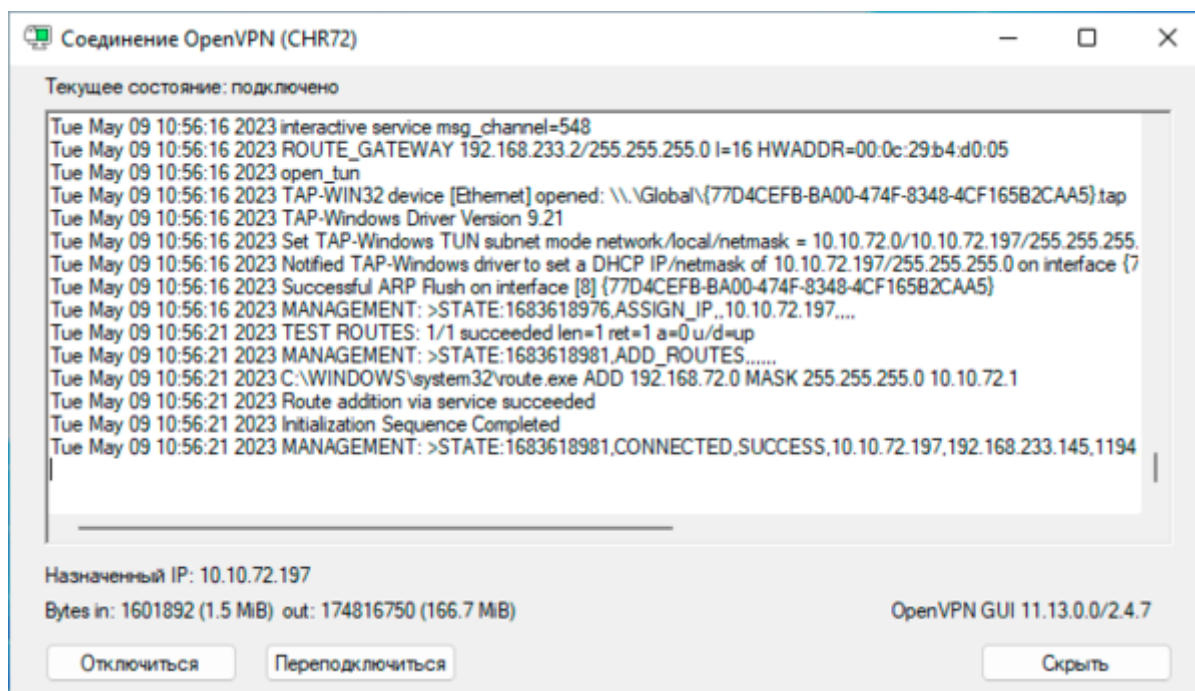
Отключаем шифрование:

```
#comp-lzo
```

Данный конфигурационный файл можно использовать и в Linux системах, для этого нужно раскомментировать следующие опции:

```
user nobody  
group nogroup
```

Проверяем подключение, если все сделано правильно - оно будет успешным.



Как видим, настроить OpenVPN сервер на Mikrotik под управлением RouterOS 7 достаточно несложно, но при этом надо учитывать все плюсы и минусы данной реализации и трезво подходить к выбору.

Онлайн-курс по MikroTik

Научиться настраивать MikroTik с нуля или систематизировать уже имеющиеся знания можно на [углубленном курсе по администрированию MikroTik](#). Автор курса, сертифицированный тренер MikroTik Дмитрий Скоромнов, лично проверяет лабораторные работы и контролирует прогресс каждого своего студента. В три раза больше информации, чем в вендорской программе MTCNA, более 20 часов практики и доступ навсегда.