Настраиваем IKEv2 VPN-сервер на роутерах Mikrotik с аутентификацией по сертификатам

Записки IT специалиста

Технический блог специалистов ООО"Интерфейс"

- Главная
- Настраиваем IKEv2 VPN-сервер на роутерах Mikrotik с аутентификацией по сертификатам

Сейчас, когда многие настраивают VPN для работы удаленных сотрудников, выбор протокола становится как никогда актуальным. С одной стороны стоят поддерживаемые современными ОС протоколы PPTP и L2TP, которые имеют ряд существенных недостатков и ограничений, с другой OpenVPN, который всем хорош, но требует установки стороннего ПО. При этом как-то забывают о быстром и безопасном IKEv2, основанном на IPsec новом протоколе, также поддерживаемом всеми современными ОС.

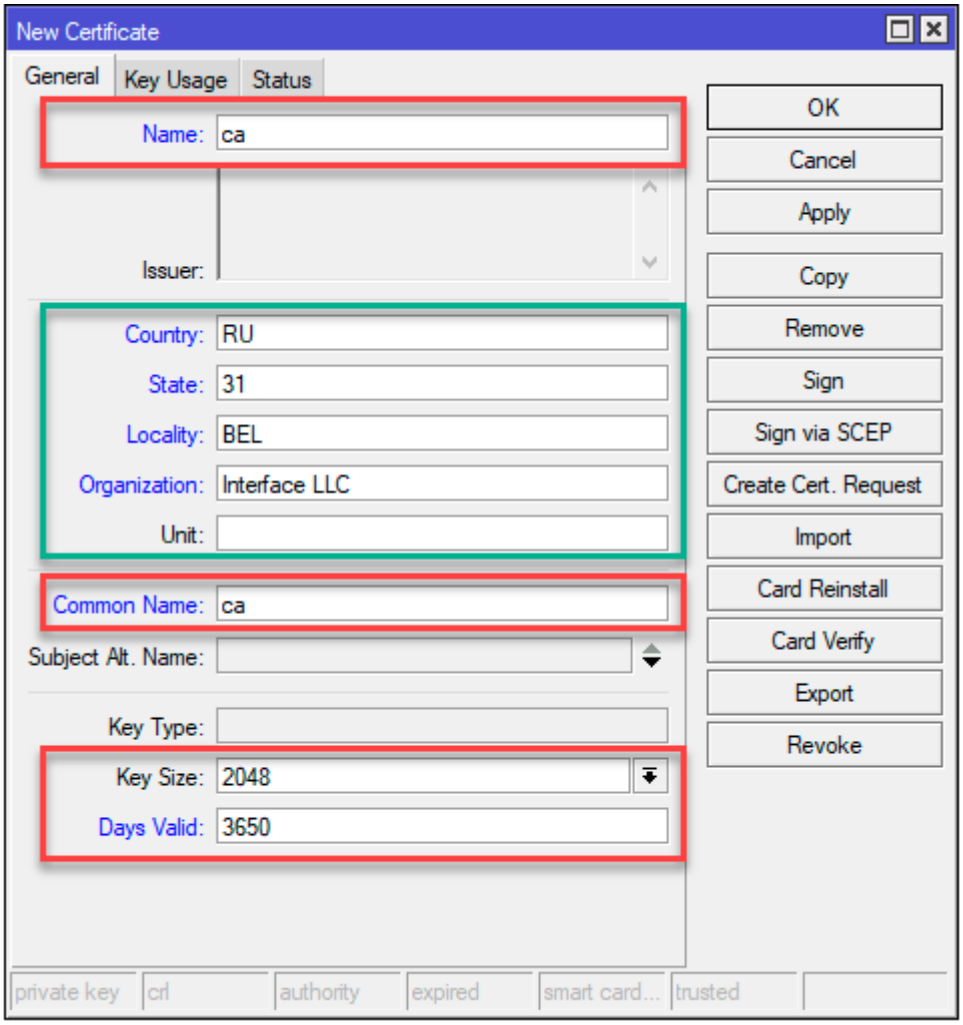
Почему именно IKEv2? Данный протокол входит в группу протоколов IPsec и обеспечивает высокий уровень безопасности, включая аутентификацию клиента с использованием сертификата, а также проверку подлинности сервера клиентом, что исключает атаки типа "человек посередине". При поддержке аппаратного ускорения IPsec со стороны оборудования показывает хорошую скорость соединения относительно других типов VPN в RouterOS и весьма прост в настройке с клиентской стороны, не требует добавления маршрутов.

К недостаткам можно отнести достаточную сложность настройки серверной части, которая требует выполнения определенных условий и наличия базового объема знаний о работе IPsec. В данной статье мы не будем углубляться в теорию, сделав упор на практическую сторону вопроса, ограничившись краткими пояснениями необходимости тех или иных настроек.

Создание центра сертификации и выпуск сертификатов

Когда мы говорим об использовании сертификатов для аутентификации, то подразумеваем наличие инфраструктуры открытых ключей (PKI), образующей область доверия, за счет чего появляется возможность проверки подлинности любого субъекта инфраструктуры без привлечения третьих служб и списков пользователей. В основе PKI лежит центр сертификации - CA, выпускающий сертификаты и дающий возможность убедиться в их подлинности при помощи корневого публичного сертификата.

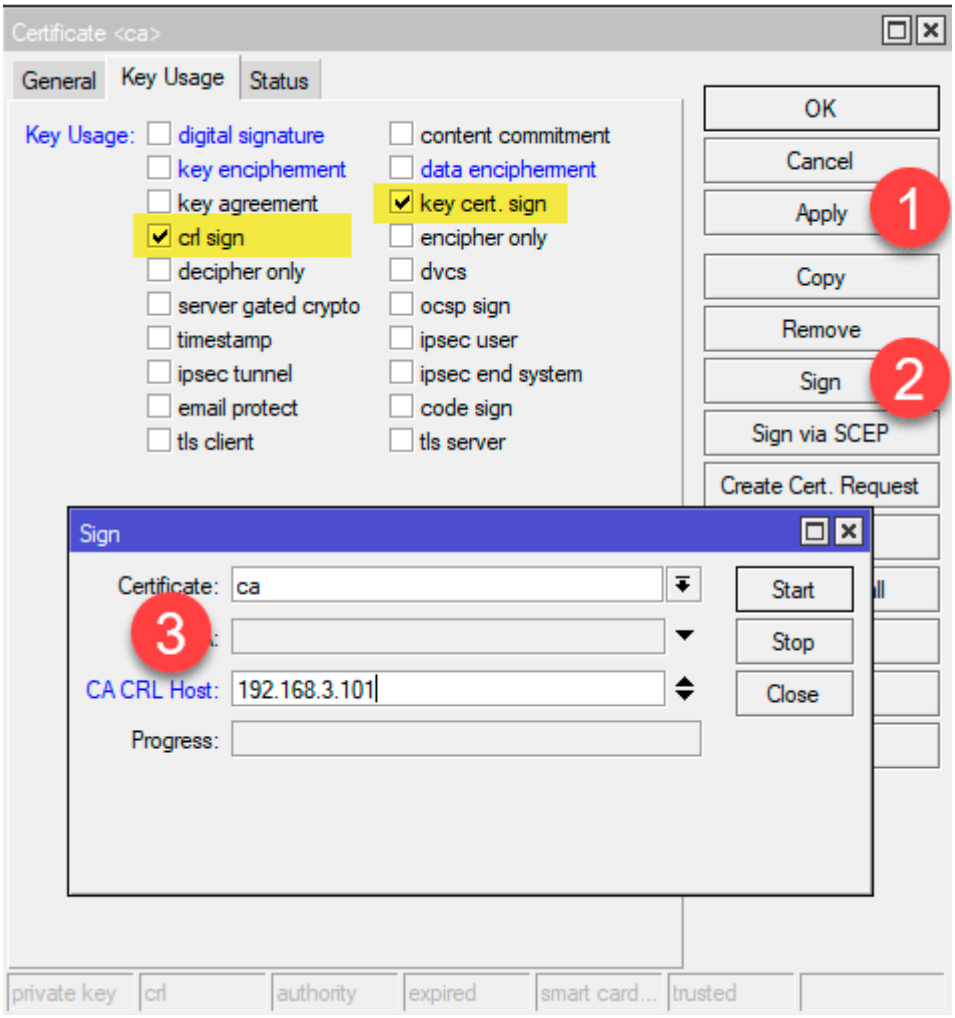
В нашем случае центр сертификации будет создан средствами RouterOS прямо на маршрутизаторе. Для этого перейдем в **System - Certificate** и выпустим корневой сертификат нашего CA.



Красным указаны обязательные к заполнению поля. **Name** - видимое имя сертификата и **Common Name** - имя субъекта, которому выдан сертификат, в нашем случае это **ca**. **Key Size** - размер ключа, ключи размером менее 2048 байт не считаются безопасными, **Days Valid** - время действия сертификата, в нашем случае 10 лет.

Выделенный зеленым блок не является обязательным, но мы советуем его заполнять, дабы в дальнейшем не пришлось угадывать, что это за сертификат и кому и кем он выдан.

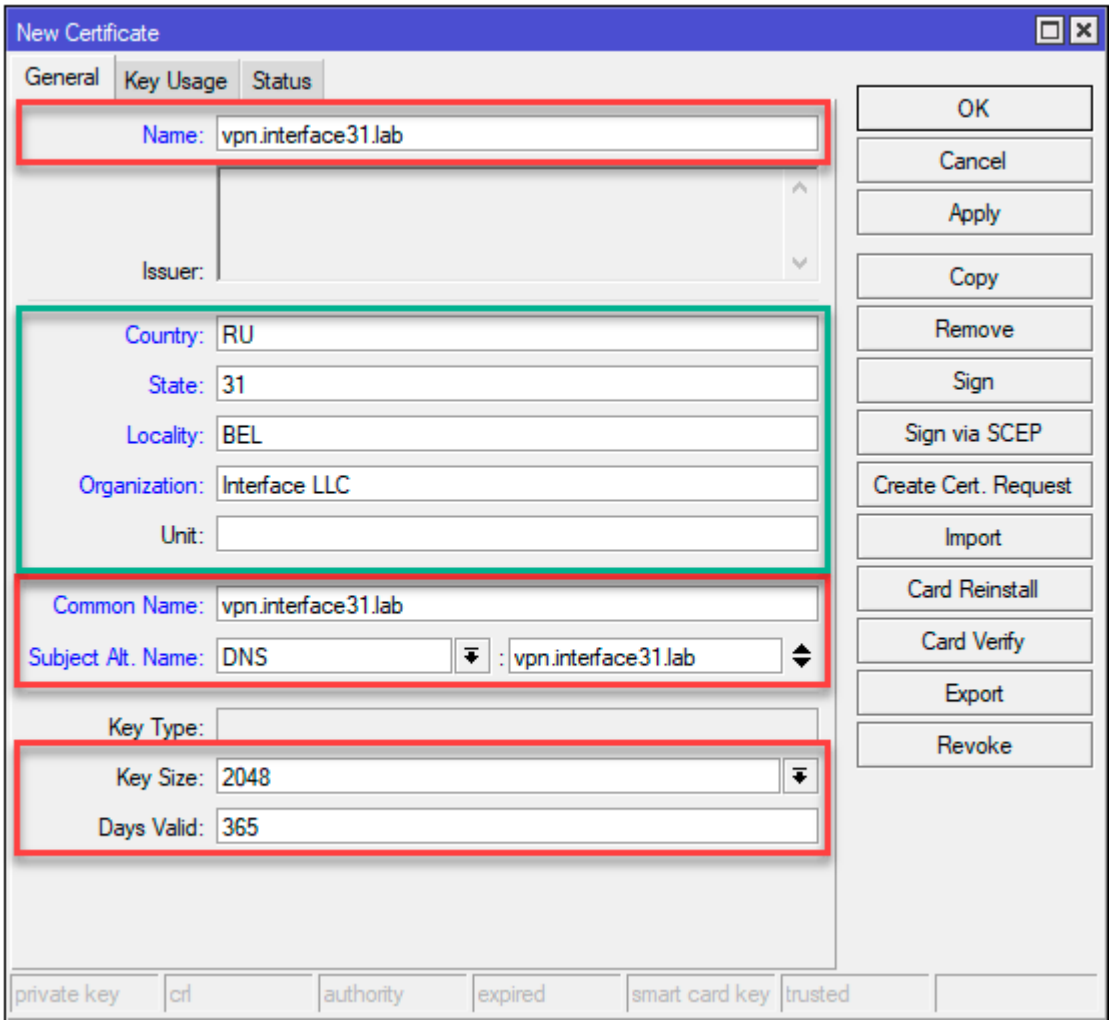
Затем перейдем на закладку **Key Usage** и оставим только **crl sign** и **key cert. sign**, затем нажмем **Apply**, чтобы применить изменения, после чего подпишем сертификат. нажав кнопку **Sign**, в открывшемся окне укажем **CA CRL Host**, в качестве которого следует использовать один из IP-адресов роутера.



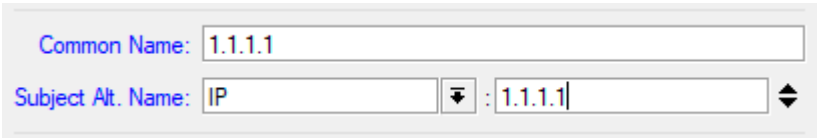
В терминале эти же действия можно выполнить командой:

```
/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=192.168.103.1
```

Следующим шагом выпустим сертификат сервера. Обратите внимание, что сервер **обязательно** должен иметь **выделенный IP адрес** и, желательно, доменное имя. Последнее условие не является обязательным, но предпочтительно, так как позволит отвязаться от использования адреса и в случае изменения IP вам не придется перевыпускать сертификаты и менять настройки клиентских подключений.

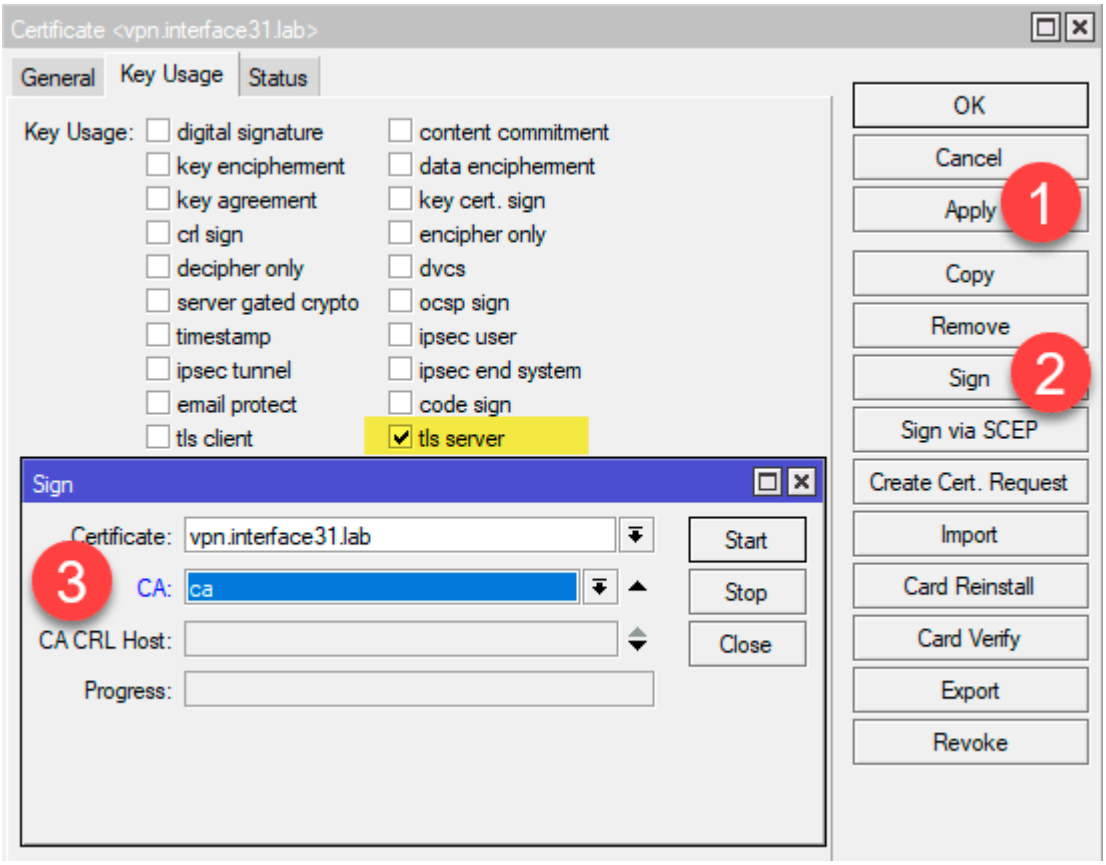


Заполнение полей в целом повторяет предыдущий пример, за исключением **Common Name** и **Subject Alt. Name**. Здесь мы указываем IP-адрес или FQDN по которому клиенты будут подключаться к серверу. Если вы используете IP-адрес, то тип записи в поле **Subject Alt. Name** нужно сменить на **IP**.



Обратите внимание, если вы выпустили сертификат с указанием FQDN, а подключить клиента попытаетесь по IP-адресу, либо наоборот, то такое соединение окажется **невозможным**.

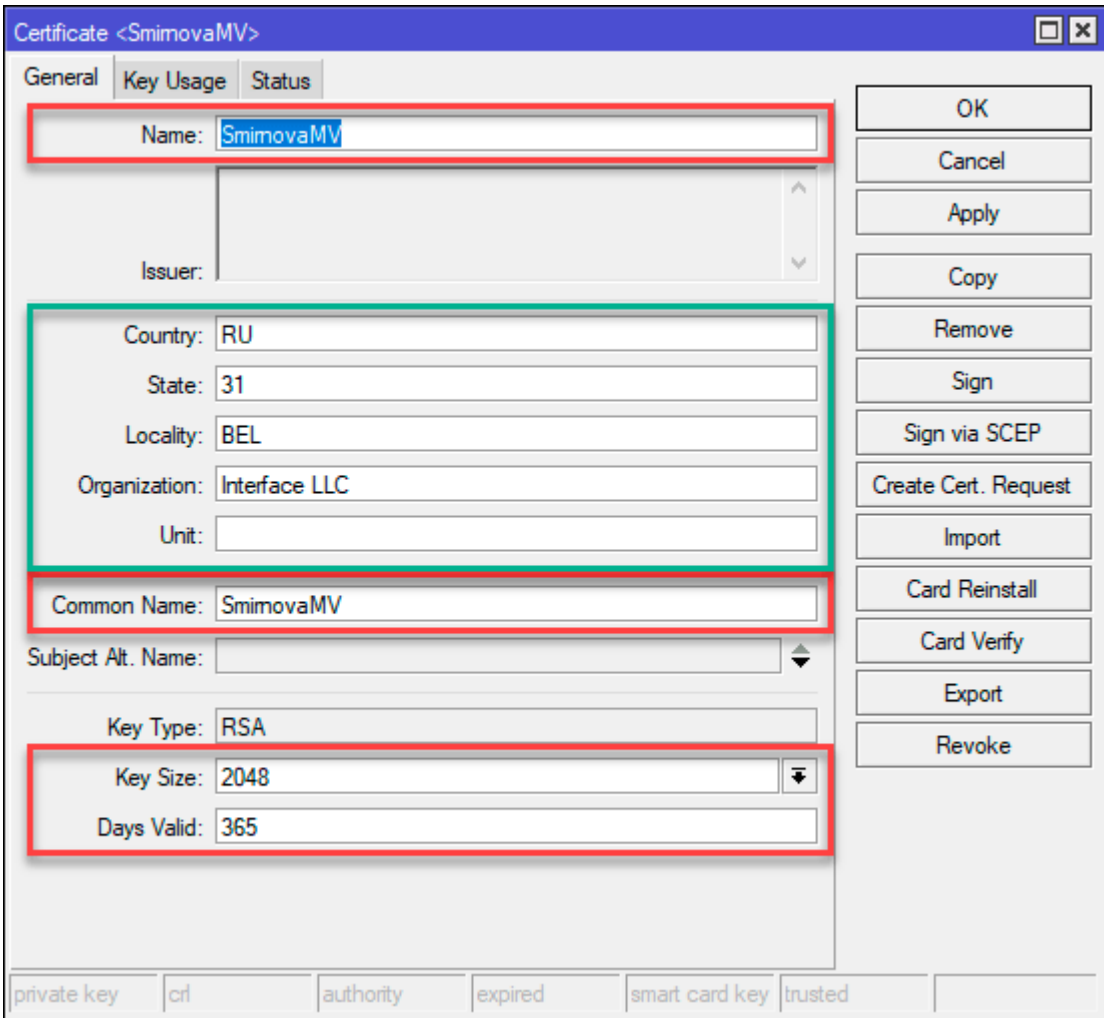
На закладке **Key Usage** укажем единственное значение **tls server** и подпишем наш сертификат закрытым ключом центра сертификации CA.



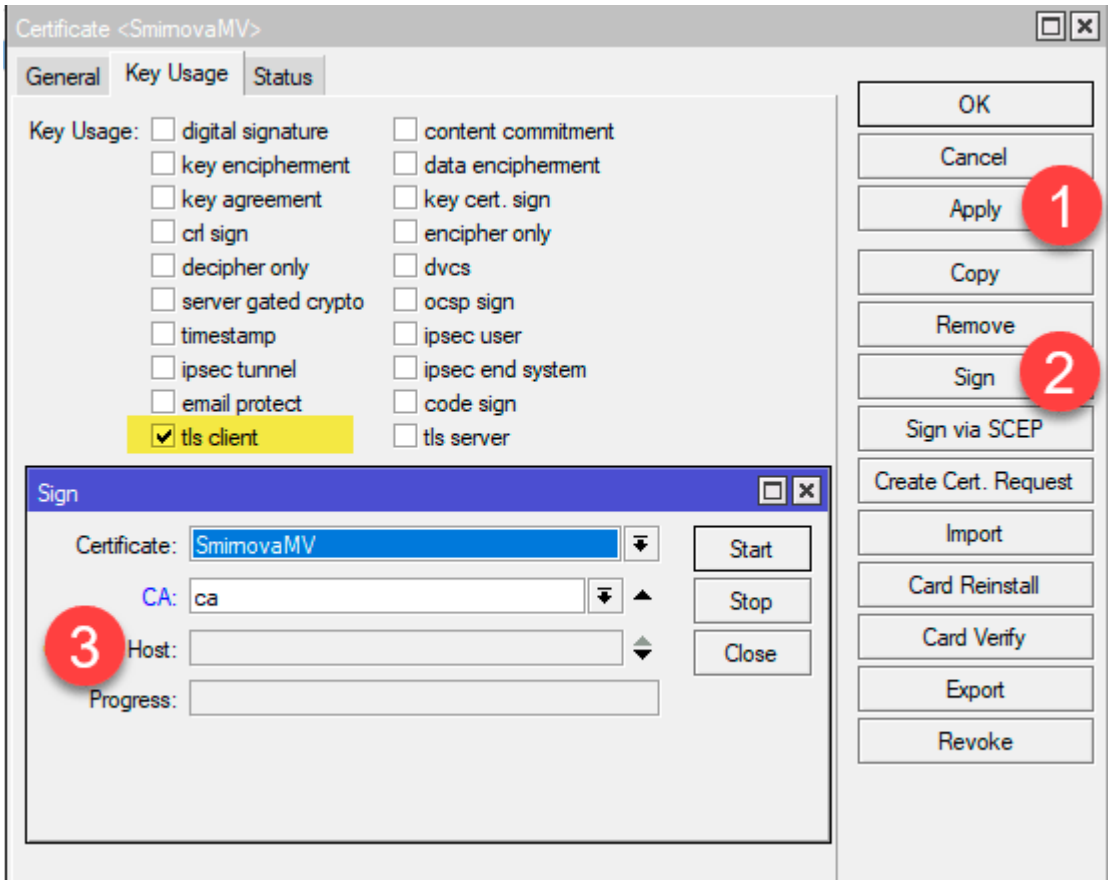
Эти же действия в терминале:

```
/certificate
add name=vpn.interface31.lab country="RU" state="31" locality="BEL"
organization="Interface LLC" common-name="vpn.interface31.lab" subject-alt-
name=DNS:"vpn.interface31.lab" key-size=2048 days-valid=3650 key-usage=tls-server
sign vpn.interface31.lab ca="ca"
```

Теперь можно выпускать клиентские сертификаты, это можно сделать как сразу, так и потом. Никаких особых требований здесь нет, в качестве имени указывайте максимально понятное значение, скажем, ФИО сотрудника или наименование офиса. Потому как понять кому принадлежит сертификат с **CN IvanovIA** не составит особого труда, в отличие от какого-нибудь безликого **client3**. Также обратите внимание на опцию **Days Valid**, не следует выдавать клиентские сертификаты на большой срок.



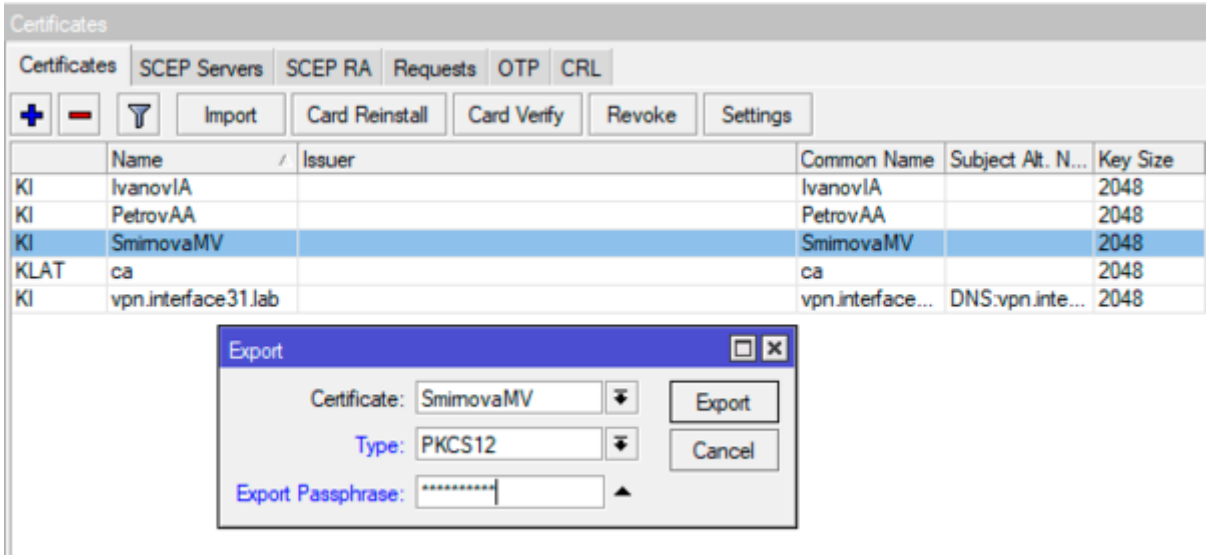
В **Key Usage** также указываем единственное назначение сертификата - **tls client** и подписываем его закрытым ключом CA.



Команды для терминала:

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

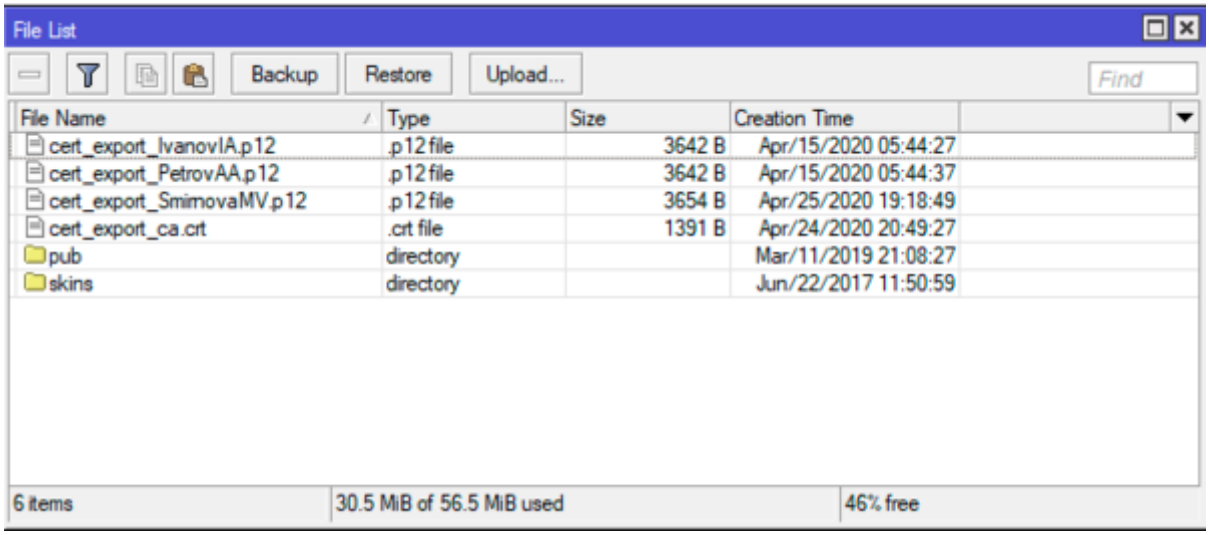
Для использования на клиентских устройствах сертификаты следует экспортировать, наиболее удобно использовать для этого формат PKCS12, который в одном файле содержит закрытый ключ клиента, его сертификат и корневой сертификат CA. Для этого выберите сертификат в списке и в меню правой кнопки мыши укажите действие **Export**. В поле **Type** укажите **PKCS12**, а в **Export Passphrase** следует указать **пароль** (не менее 8 символов), в противном случае закрытый ключ выгружен не будет.



Это же можно сделать командой:

```
/certificate
export-certificate SmirnovaMV type=pkcs12 export-passphrase=0123456789
```

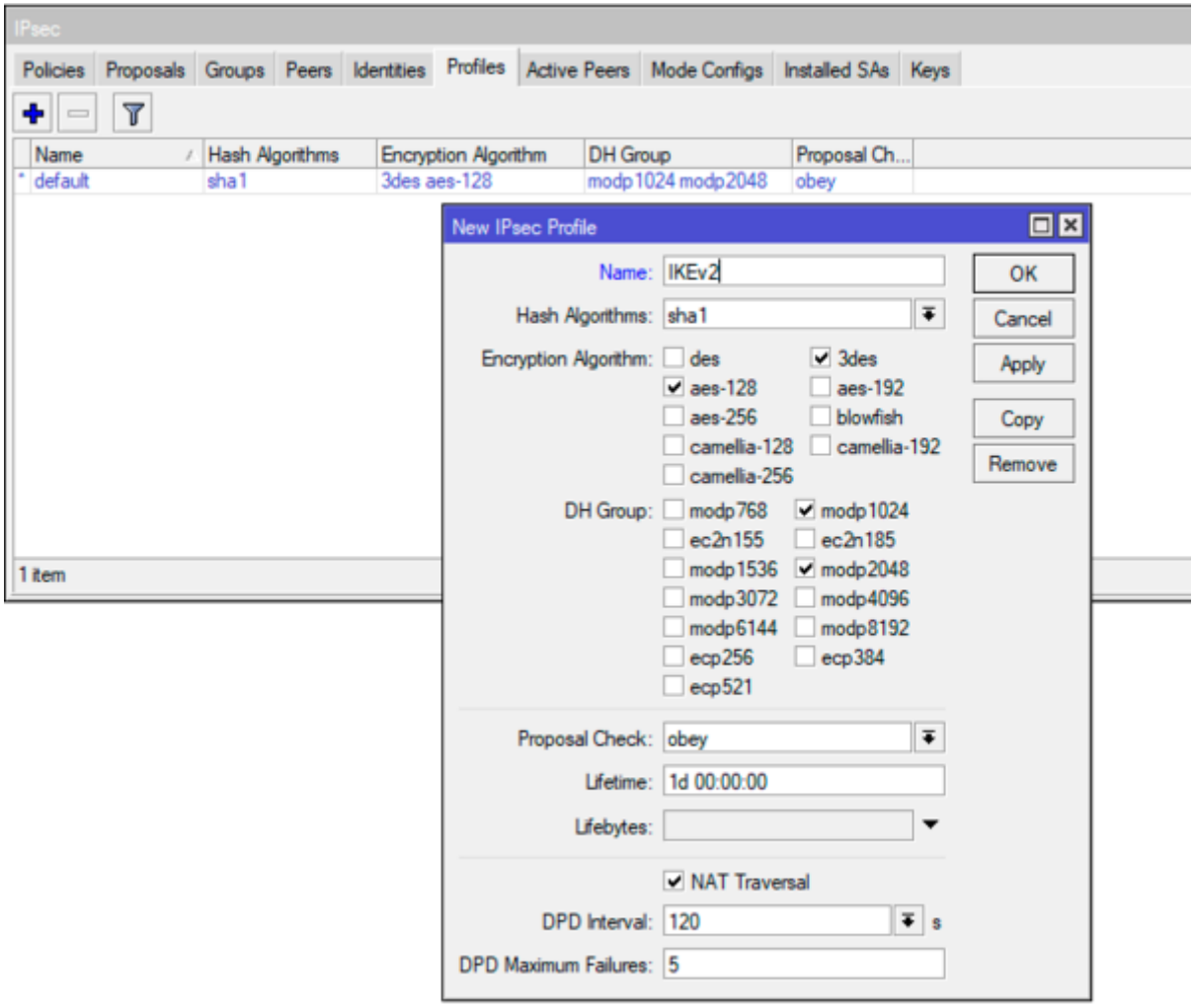
Скачать экспортированные сертификаты можно из раздела **Files**.



Настройка IKEv2 VPN-сервера

Здесь мы вступаем в достаточно сложную область настройки IPsec, объем статьи не позволяет подробно останавливаться на назначении каждой настройки, поэтому если вы не уверены в своих действиях, то мы не рекомендуем отклоняться от указанных ниже настроек.

Перейдем в **IP - IPsec - Profiles** и создадим новый профиль, который задает параметры для установления соединения. Все параметры оставляем **по умолчанию**, кроме наименования, которому следует дать осмысленное имя.



Либо выполните команду в терминале:

```
/ip ipsec profile
add name=IKEv2
```

Затем перейдем на закладку **Proposals** - предложения, который содержит параметры криптографии предлагаемые для соглассования подключающимся клиентам. Создадим новое предложение, которое сформировано с учетом используемых современными ОС алгоритмов и изменение его состава может либо ослабить безопасность, либо сделать подключение некоторых клиентов невозможным.

Параметры по умолчанию нам не подойдут, поэтому в блоке **Encr. Algorithms** убираем **3des** и добавляем **aes-128-cbc, aes-192-cbc, aes-256-cbc**.

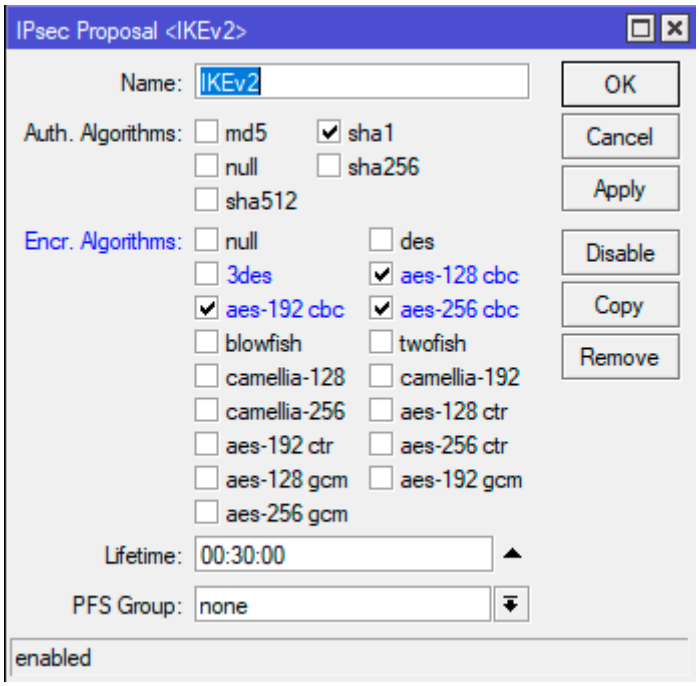
В терминале достаточно простой команды:

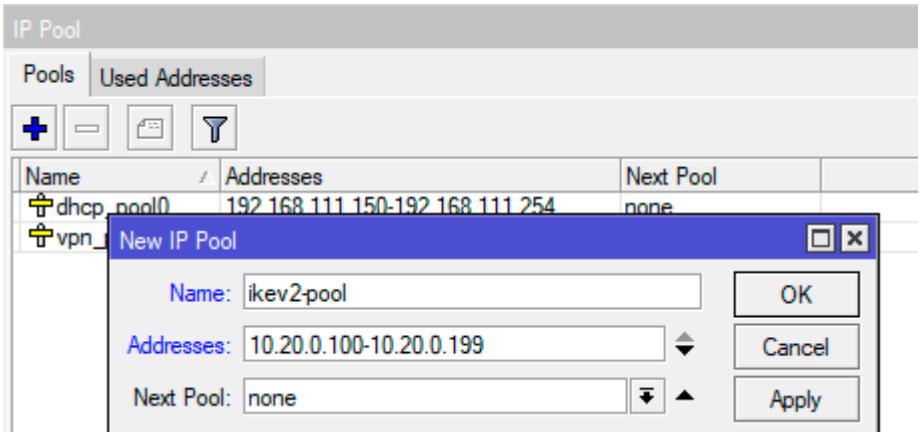
```
/ip ipsec proposal
add name=IKEv2 pfs-group=none
```

Здесь мы сталкиваемся с одной особенностью: создаваемые через терминал и Winbox предложения содержат различный набор параметров. То, что создается в терминале полностью соответствует приведенным выше на скриншоте

требованиям.

Для выдачи VPN-клиентам нам потребуется отдельный диапазон адресов, перейдем в **IP - Pool** и создадим новый пул, в нашем случае будет использован диапазон адресов **10.20.0.100 - 10.20.0.199**:



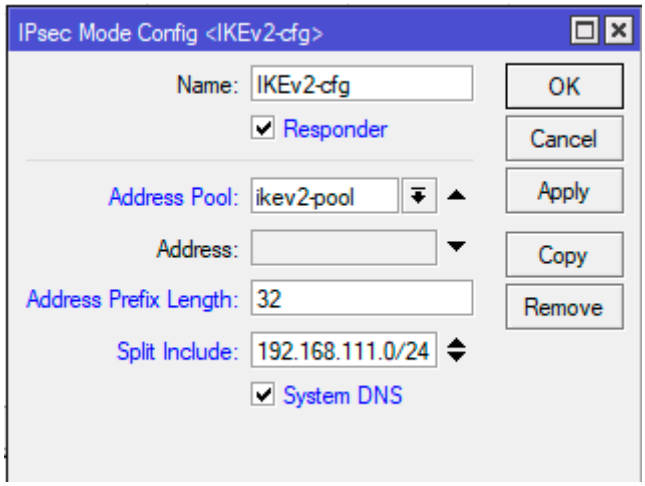


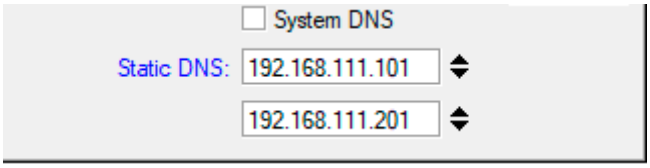
Снова вернемся к настройкам IPsec и создадим конфигурацию, передаваемую клиенту для настройки его сетевых параметров, для этого перейдем на в **IP - IPsec - Mode Configs**. При создании новой конфигурации установим флаг **Responder**, в поле **Address Pool** укажем **имя созданного нами пула**, в поле **Address Prefix Lenght** укажем префикс адреса - **32**, поле **Split Include** указываем подсети, запросы к которым следует направлять в туннель, здесь следует указать одну или несколько внутренних сетей, доступ к которым должны получать удаленные клиенты. В нашем случае это сеть условного офиса - 192.168.111.0/24. Наконец флаг **System DNS** предписывает клиенту использовать DNS сервера указанные в **IP - DNS** роутера. Если передавать DNS-сервера не требуется, то данный флаг следует **снять**.

Это же действие в терминале:

```
/ip ipsec mode-config
add address-pool=ikev2-pool address-prefix-length=32 name=IKEv2-cfg split-
include=192.168.111.0/24
```

Если же вам нужно, чтобы клиенты использовали внутренние сервера имен, например, в Active Directory, то флаг **System DNS** также следует **снять** и указать адреса требуемых DNS-серверов.



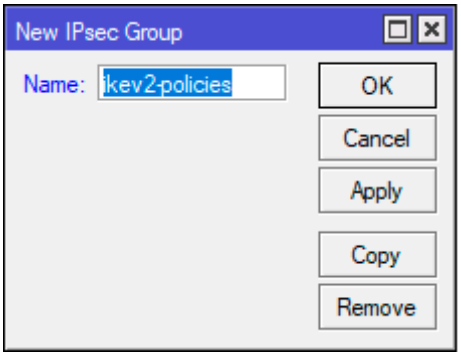


Команда для терминала будет выглядеть так:

```
/ip ipsec mode-config
add address-pool=ikev2-pool address-
prefix-length=32 name=IKEv2-cfg
split-include=192.168.111.0/24
static-
dns=192.168.111.101,192.168.111.201
system-dns=no
```

На закладке **Groups** создадим новую группу, никаких настроек здесь нет, просто укажите уникальное имя:

```
/ip ipsec policy group
add name=ikev2-policies
```



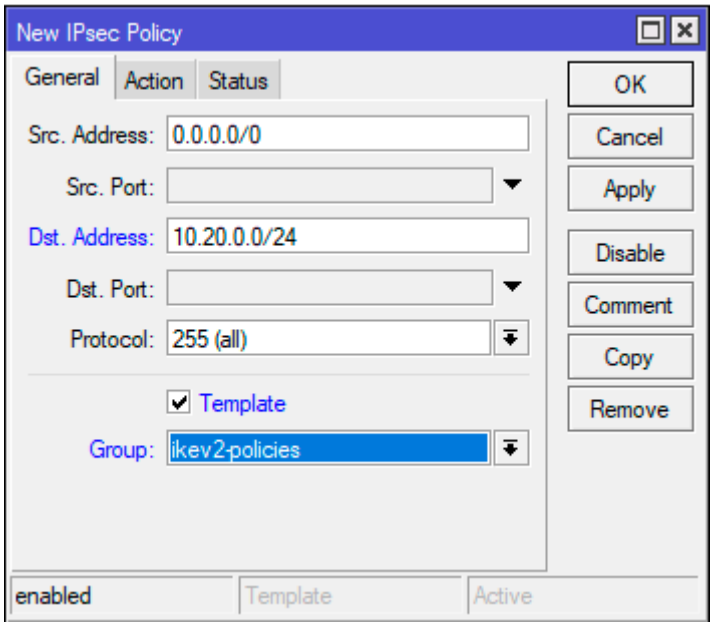
Затем на закладке **Policices** создадим шаблон политики, которая будет указывать какой именно трафик будет подвергаться обработке IPsec и отправляться в туннель. В поле **Src. Address** оставляем **0.0.0.0/0**, в поле **Dst. Address** указываем выделенный для VPN-сети диапазон: **10.20.0.0/24**, устанавливаем флаг **Template** и указываем созданную нами ранее группу в поле **Group**.

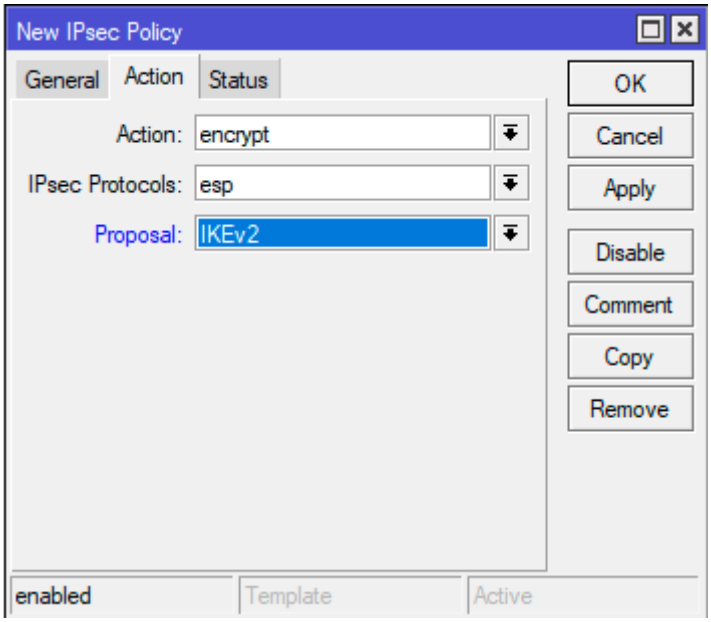
На закладке **Action** в поле **Proposal** укажите созданный нами ранее набор предложений.

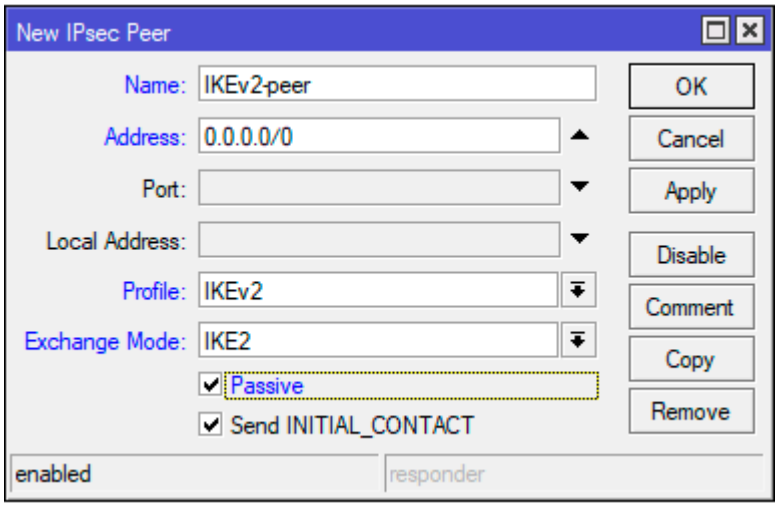
Эти же действия в терминале:

```
/ip ipsec policy
add dst-address=10.20.0.0/24 group=ikev2-policies proposal=IKEv2 src-
address=0.0.0.0/0 template=yes
```

После чего перейдем в **IP - IPsec - Peers** создадим новый пир для приема подключений. Сразу установим флаг **Passive**, в поле **Address** указываем **0.0.0.0/0** (разрешаем подключаться из любого места), в поле **Profile** указываем созданный нами профиль, а в поле **Exchange Mode** укажем протокол обмена ключами - **IKE2**.







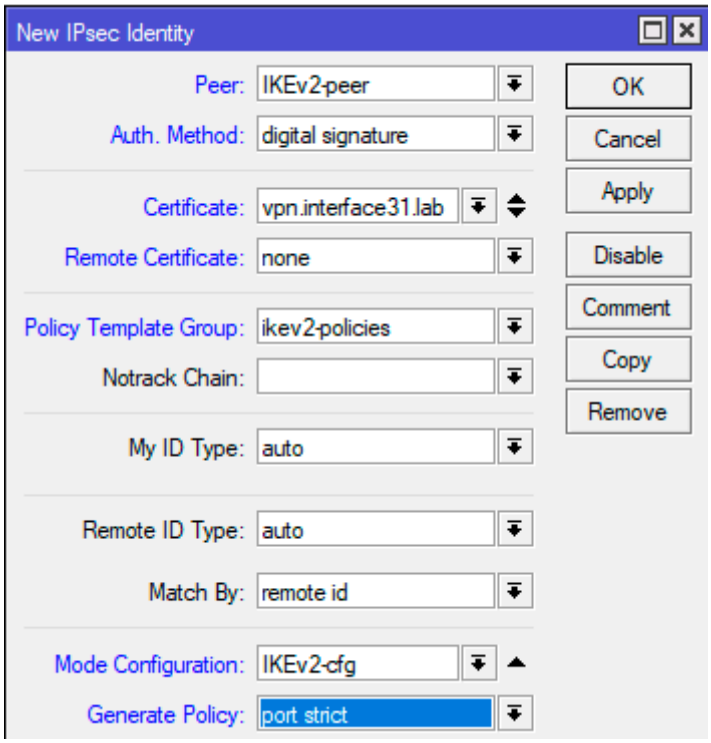
В терминале для получения аналогичного результата выполните:

```
/ip ipsec peer
add exchange-mode=ike2 name=IKEv2-peer passive=yes profile=IKEv2
```

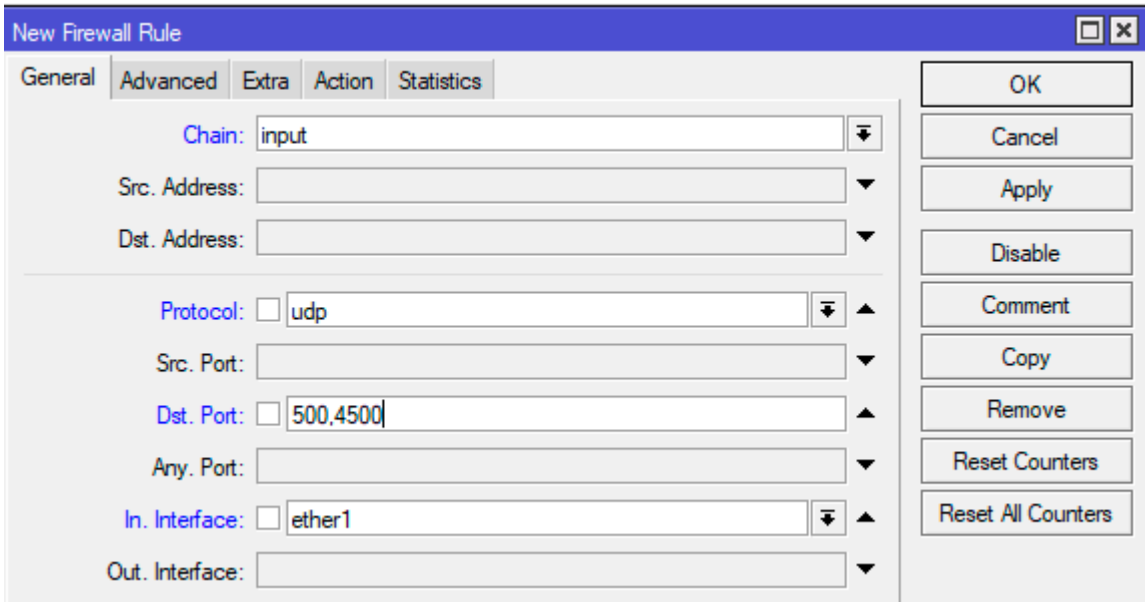
На закладке **Identities** создадим новую настройку идентификации подключающихся клиентов. Здесь много настраиваемых полей и нужно быть предельно внимательными, чтобы ничего не упустить и не перепутать. В поле **Peer** - указываем созданный нами пир, **Auth. Method** - способ аутентификации - **digital signature**, **Certificate** - сертификат сервера. **Policy Template Group** - группа шаблонов политик - выбираем созданную нами группу, **Mode Configuration** - указываем созданную нами конфигурацию для клиентов, **Generate Policy - port strict**.

Команда для терминала:

```
/ip ipsec identity
add auth-method=digital-signature
certificate=vpn.interface31.lab
generate-policy=port-strict mode-
config=IKEv2-cfg peer=IKEv2-peer
policy-template-group=ikev2-
policies
```



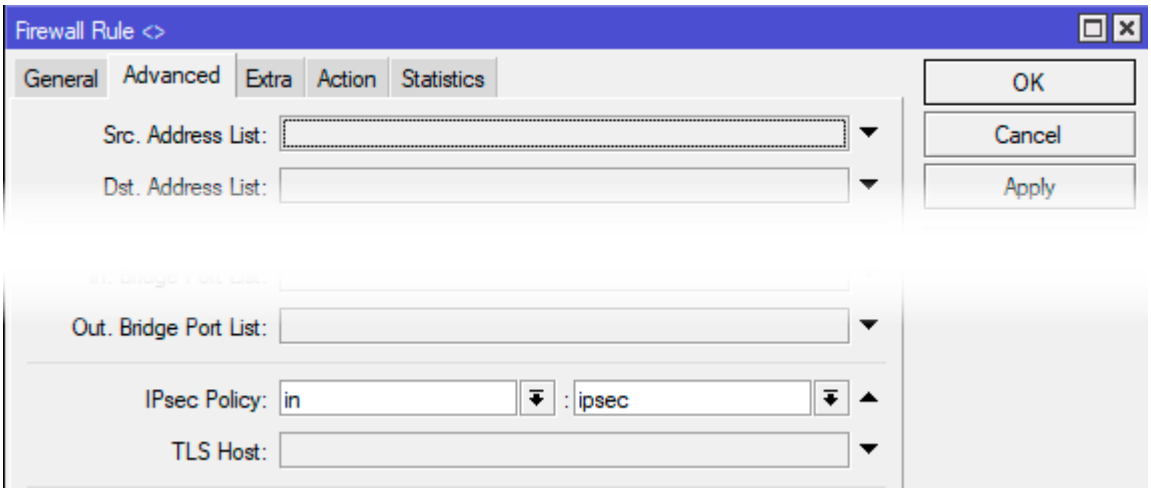
На этом настройка сервера завершена, осталось лишь добавить правила брандмауэра, разрешающие работу с ним. Для того, чтобы клиенты могли подключаться к серверу перейдем в **IP - Firewall - Filter Rules** и добавим правило: **Chain - input, Protocol - udp, Dst. Port - 500, 4500, In. Interface** - ваш внешний интерфейс (в нашем случае это **ether1**). Действие не указываем, так как по умолчанию применяется **accept**.



Для добавления правила в терминале:

```
/ip firewall filter
add action=accept chain=input dst-port=500,4500 in-interface=ether1 protocol=udp
```

Но это еще не все, чтобы VPN-клиенты могли получить доступ к внутренней сети, следует добавить еще одно правило. На закладке **General** укажите **Chain - forward** и **Interface** - внешний интерфейс, затем на **Advanced**: **IPsec Policy - in:ipsec**.

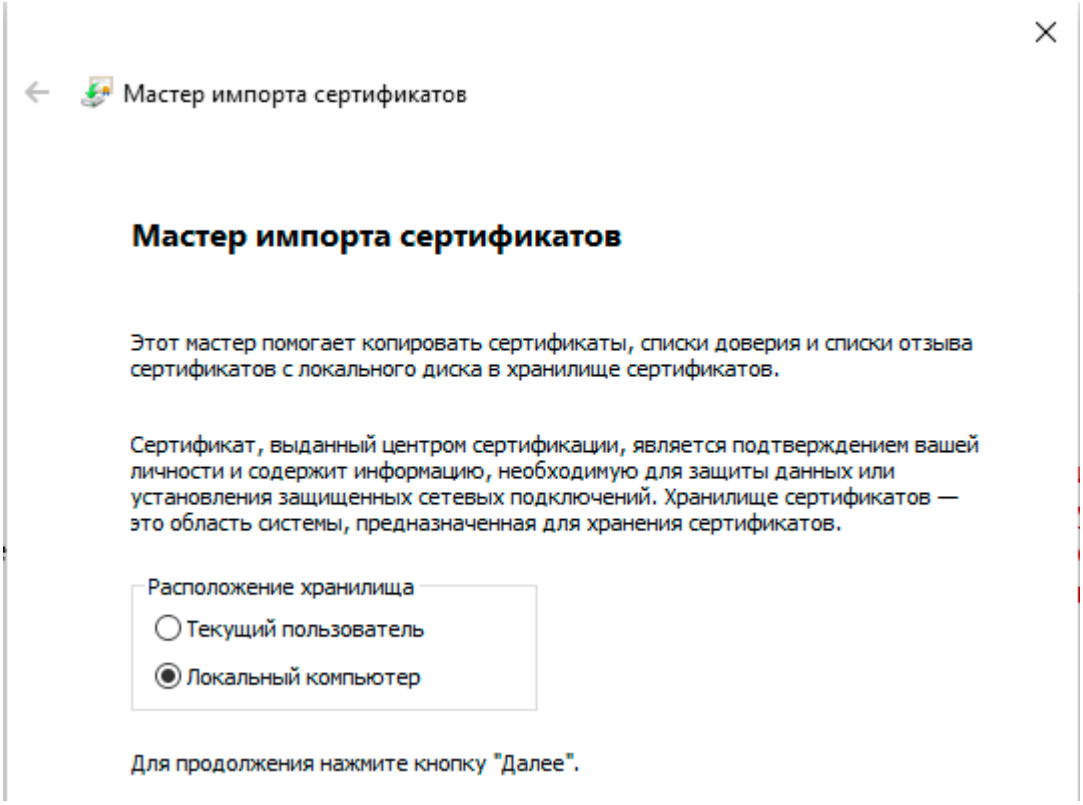



```
/ip firewall filter
add action=accept chain=forward in-interface=ether1 ipsec-policy=in,ipsec
```

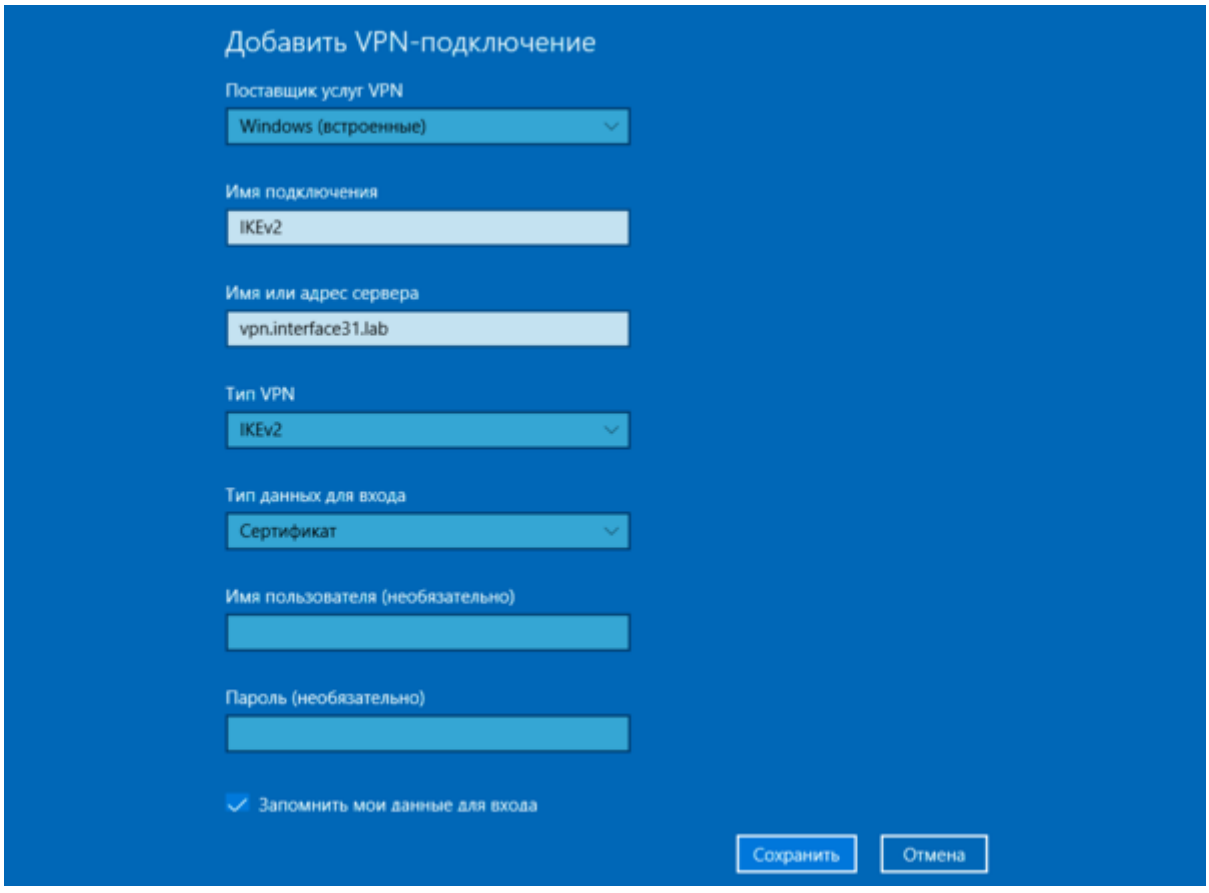
Оба правила следует расположить выше, чем запрещающие в каждой из цепочек.

Настройка подключения клиента в Windows

Прежде всего импортируем сертификат, для этого можно просто выполнить двойной клик на файле сертификата, в открывшемся **Мастере импорта** в качестве **Расположения хранилища** укажите **Локальный компьютер**, остальные параметры принимаются по умолчанию.

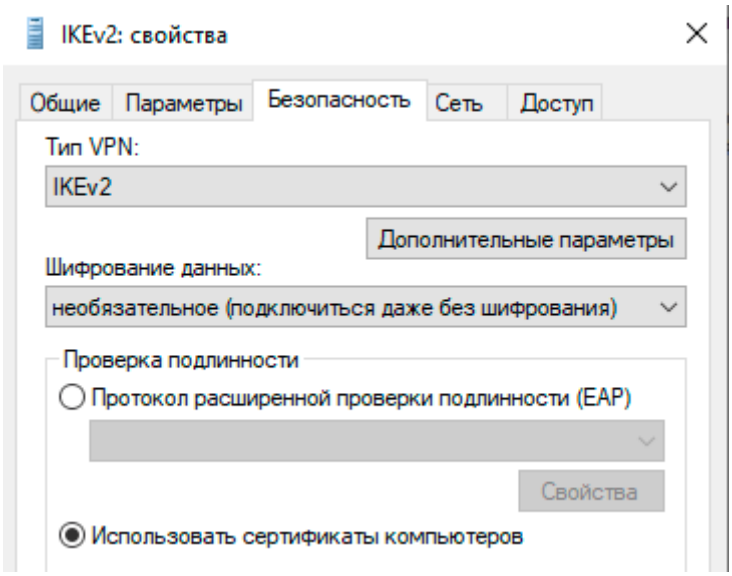


Затем создадим новое подключение штатными инструментами. А качестве **Типа VPN** укажем **IKEv2**, а в качестве **Типа данных для входа** - **Сертификат**. Также обратите внимание, что в строка **Имя или адрес сервера** должно совпадать с **Common Name** сертификата сервера, в противном случае подключение установить не удастся.



После чего откроем свойства созданного подключения и перейдем на закладку **Безопасность**, где установим переключатель **Проверка подлинности** в положение **Использовать сертификаты компьютеров**.

Теперь можно подключаться, если все сделано правильно - подключение будет успешно. Проверим таблицу маршрутов:




```
IPv4 таблица маршрута
===========================================================================
Активные маршруты:
Сетевой адрес           Маска сети      Адрес шлюза       Интерфейс  Метрика
          0.0.0.0          0.0.0.0    192.168.186.1  192.168.186.199     25
         10.0.0.0        255.0.0.0         On-link       10.20.0.100     26
     10.20.0.100  255.255.255.255         On-link       10.20.0.100    281
  10.255.255.255  255.255.255.255         On-link       10.20.0.100    281
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
    192.168.3.101  255.255.255.255    192.168.186.1  192.168.186.199     26
    192.168.111.0    255.255.255.0         On-link       10.20.0.100     26
  192.168.111.255  255.255.255.255         On-link       10.20.0.100    281
    192.168.186.0    255.255.255.0         On-link   192.168.186.199    281
  192.168.186.199  255.255.255.255         On-link   192.168.186.199    281
  192.168.186.255  255.255.255.255         On-link   192.168.186.199    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link   192.168.186.199    281
        224.0.0.0        240.0.0.0         On-link       10.20.0.100    281
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link   192.168.186.199    281
  255.255.255.255  255.255.255.255         On-link       10.20.0.100    281
===========================================================================
Постоянные маршруты:
  Отсутствует
```

Как видим, маршрут к нашей внутренней сети 192.168.111.0/24 был добавлен автоматически и никаких ручных настроек клиента не требуется.

Настройка подключения клиента в Linux

Точно также начнем с сертификата, но в данном случае нам потребуется немного больше действий. Будем считать, что сертификат находится в корневой директории пользователя, для которого мы настраиваем подключение. Все последующие команды также следует выполнять от его имени.

Перейдем в домашнюю директорию и создадим скрытую папку для хранения ключей и сертификатов:

```
cd ~
mkdir .ikev2
```

Теперь нам нужно экспортировать из PKCS12 файла корневой сертификат CA, а также ключ и сертификат пользователя. Начнем с корневого сертификата:

```
openssl pkcs12 -in cert_export_SmirnovaMV.p12 -out .ikev2/IKEv2_CA.crt -nodes -
nokeys -cacerts
```

Затем экспортируем сертификат клиента:

```
openssl pkcs12 -in cert_export_SmirnovaMV.p12 -out .ikev2/SmirnovaMV.crt -nodes -
nokeys
```

И его закрытый ключ. При экспорте закрытого ключа нас попросят установить для него пароль, минимальная длинна пароля 8 символов. Пропустить этот шаг нельзя.

```
openssl pkcs12 -in cert_export_SmirnovaMV.p12 -out .ikev2/SmirnovaMV.pass.key -
nocerts
```

На каждом из этих этапов нам нужно будет вводить парольную фразу, указанную при экспорте сертификата пользователя на роутере.

И наконец уберем пароль с закрытого ключа пользователя:

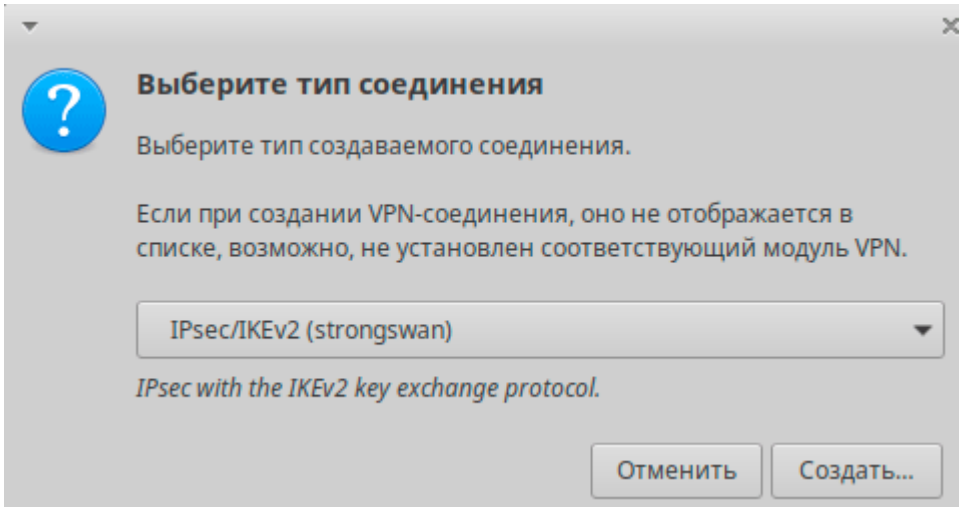
```
openssl rsa -in .ikev2/SmirnovaMV.pass.key -out .ikev2/SmirnovaMV.key
```

Во время этого действия вы должны будете ввести пароль, который указали при создании ключа.

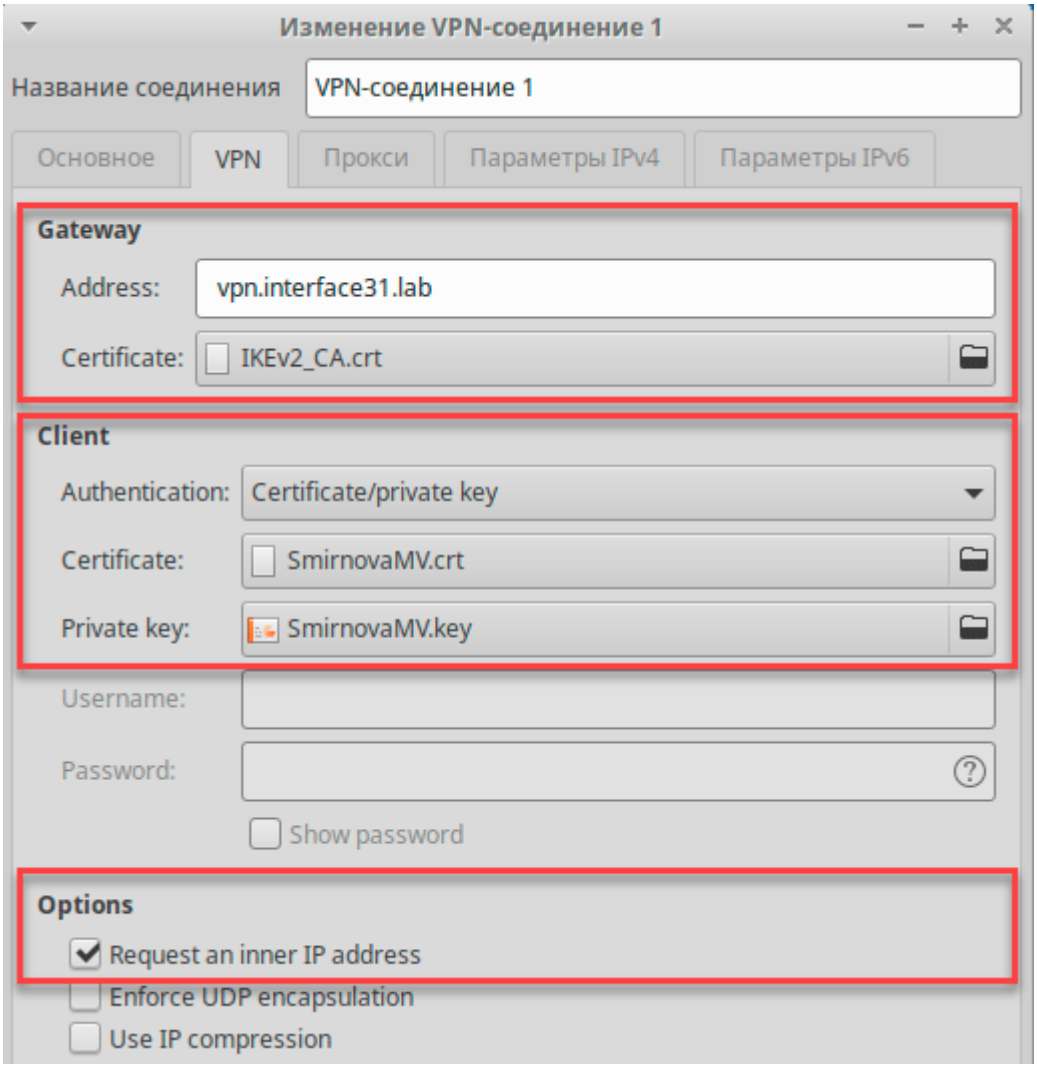
Для того, чтобы иметь возможность создавать VPN-подключения в графическом интерфейсе установим необходимый плагин для Network Manager:

```
sudo apt install network-manager-strongswan
```

После чего вам станут доступны настройки VPN IKEv2 соединения.



Настройки соединения достаточно просты. В секции **Gateway** указываем **адрес сервера** и путь к **корневому сертификату CA**. В секции **Client** устанавливаем **Authentication: Certificate/private key** и указываем пути к **сертификату и закрытому ключу** клиента. И в секции **Option** обязательно устанавливаем флаг **Request an inner IP address**. На этом настройка соединения окончена, можно подключаться.



Если мы после подключения проверим таблицу маршрутизации, то не обнаружим маршрута к офисной сети, но при этом она будет доступна:

```
root@andrey-xfce:/var/log# ip r
default via 192.168.16.2 dev ens33 proto dhcp metric 100
10.20.0.102 dev ens33 proto kernel scope link src 10.20.0.102 metric 50
10.20.0.102 dev ens33 proto kernel scope link src 10.20.0.102 metric 100
169.254.0.0/16 dev ens33 scope link metric 1000
192.168.3.101 via 192.168.16.2 dev ens33 proto static metric 100
192.168.16.0/24 dev ens33 proto kernel scope link src 192.168.16.132 metric 100
192.168.16.2 dev ens33 proto static scope link metric 100
root@andrey-xfce:/var/log# ping 192.168.111.150
PING 192.168.111.150 (192.168.111.150) 56(84) bytes of data.
64 bytes from 192.168.111.150: icmp_seq=1 ttl=127 time=0.973 ms
64 bytes from 192.168.111.150: icmp_seq=2 ttl=127 time=1.13 ms
64 bytes from 192.168.111.150: icmp_seq=3 ttl=127 time=1.18 ms
64 bytes from 192.168.111.150: icmp_seq=4 ttl=127 time=0.990 ms
```

Но никакой ошибки здесь нет. Просто Linux в данной ситуации поступает более правильно, вместо маршрута в системе создается соответствующая политика IPsec, которая направляет трафик к внутренней сети в туннель согласно тому, что мы указали в конфигурации клиента (**Mode Configs**) на роутере.

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.