

# Настройка WireGuard VPN на роутерах Mikrotik

 [interface31.ru/tech\\_it/2022/04/nastroyka-wireguard-vpn-na-routerah-mikrotik.html](https://interface31.ru/tech_it/2022/04/nastroyka-wireguard-vpn-na-routerah-mikrotik.html)

## Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Настройка WireGuard VPN на роутерах Mikrotik

Рассматривая настройку WireGuard в наших предыдущих материалах, мы намеренно не касались Mikrotik, запланировав для этого отдельную статью. И для этого есть свои причины. RouterOS, под управлением которой работают данные устройства, имеет свои особенности и подходы к настройке, малоинтересные другим читателям. А для пользователей Mikrotik будет лучше, если все нужное будет в одной статье. При этом мы подразумеваем, что администратор, работающий с ROS, имеет более высокий уровень подготовки и владеет основами сетей, поэтому не будем пояснять простые настройки, а сосредоточимся именно на WireGuard.



### 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.

Если вас интересует тема WireGuard в более широком ключе, безотносительно оборудования Mikrotik, то рекомендуем прочесть наши материалы:

- [Организация каналов между офисами через WireGuard VPN на платформе Linux](#)
- [Настройка WireGuard VPN для доступа в интернет](#)

Следующий очень важный момент: Wireguard доступен только в версии **RouterOS 7**, которая не смотря на статус **stable** все еще имеет достаточно "детских болезней" и прочих проблем, поэтому не стоит переходить на новую версию без вдумчивого анализа всех плюсов и минусов, а также предварительного тестирования.

В нашем случае будет использоваться **CHR с RouterOS 7.2.1** запущенная в виртуальной машине нашей тестовой лаборатории.

Как мы уже говорили, WireGuard - это простой туннель без сохранения состояния, к нему не применимы понятия **клиент** и **сервер**, каждый узел WireGuard способен подключаться к другим узлам и сам принимать соединения. Более правильно

называть узлы сети - **пиры** (*peer*) - **инициатором** и **респондером**. Первый инициирует соединение, второй его принимает. Хотя даже в среде профессионалов к узлам WireGuard продолжают применяться термины клиент и сервер, первый подключается, второй принимает подключения. Большой беды в этом нет, но вы должны понимать, что любой узел WireGuard способен выполнять обе роли одновременно.

## Mikrotik как респондер (сервер)

В RouterOS 7 появился новый пункт меню - **WireGuard**, переходим в него на одноименную закладку и создаем новый интерфейс. Заполняем поля **Name** и **Listen Port**, их назначения понятны, советуем использовать осмысленные названия интерфейсов, чтобы вы могли понимать, для чего тот или иной предназначен. Ключи будут созданы автоматически.

Interface <wireguard-sts>

General Status Traffic

Name: wireguard-sts

Type: WireGuard

MTU: 1420

Actual MTU: 1420

Listen Port: 34567

Private Key: .....

Public Key: HCtHGocpFySKCr2hQLfHTYu3DmQMYubhAob/tlh5pUM=

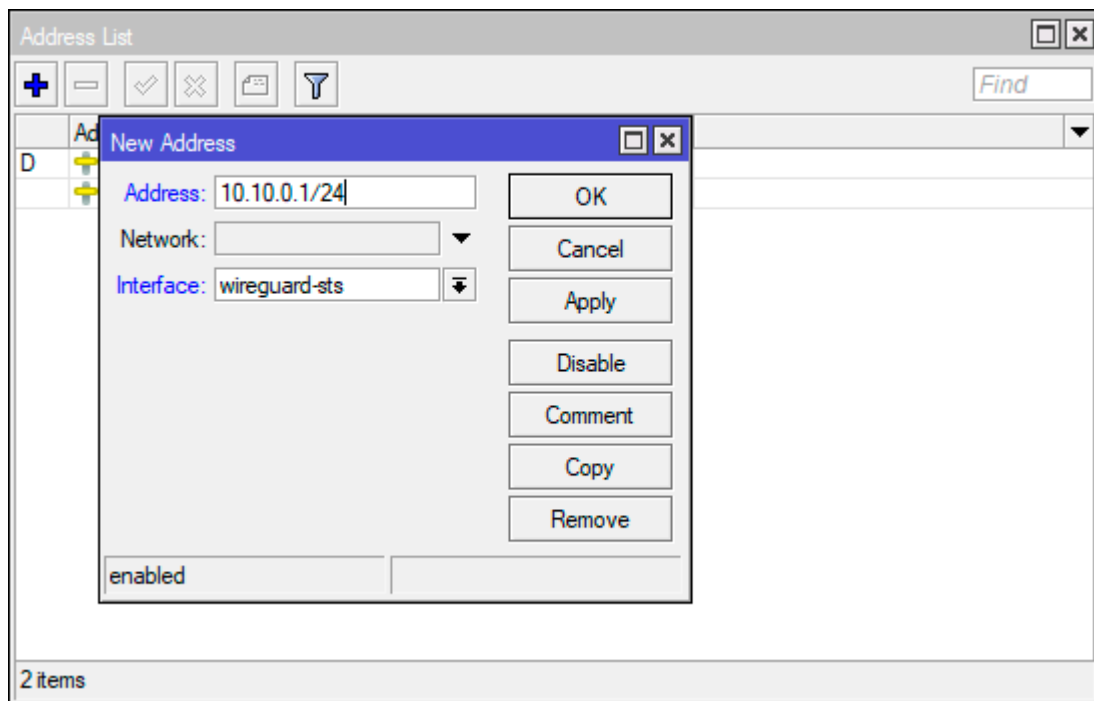
OK Cancel Apply Disable Comment Copy Remove Torch

enabled running slave

Если вы предпочитаете работать в терминале, то выполните команду:

```
/interface wireguard
add listen-port=34567 mtu=1420 name=wireguard-sts
```

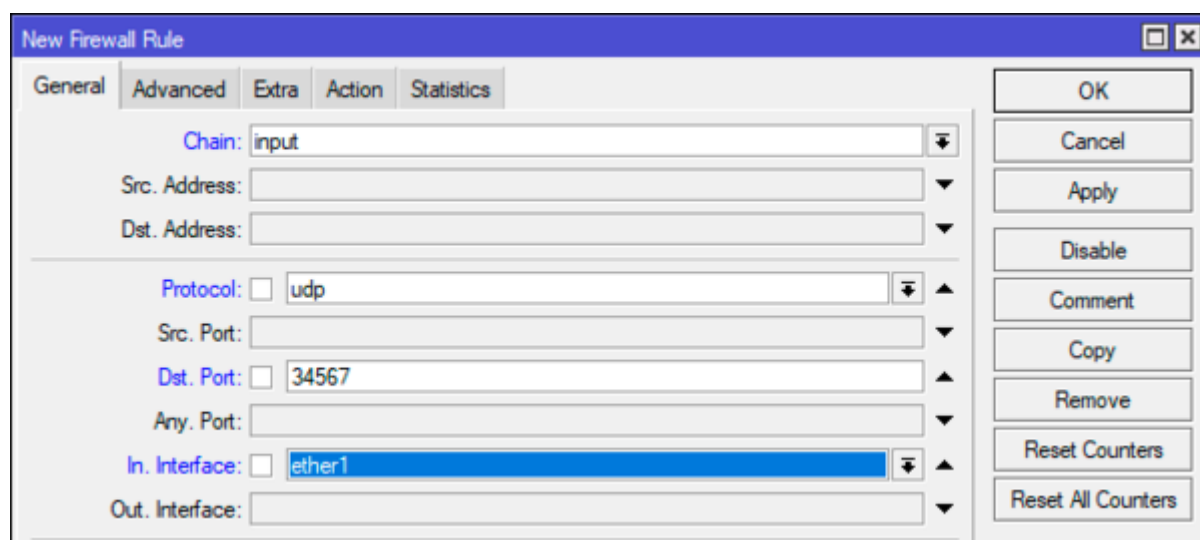
Затем присвоим созданному интерфейсу IP-адрес, для чего перейдем в **IP - Addresses** и просто добавим нужный адрес.



Или:

```
/ip address
add address=10.10.0.1/24 interface=wireguard-sts network=10.10.0.0
```

Также не забудьте разрешить входящие соединения на указанный при создании интерфейса порт, в нашем случае 34567. Это можно сделать в **IP - Firewall - Filter Rules** добавив правило: **Chain - input, Protocol - udp, Dst. Port - 34567, In. Interface - ether1** - в его качестве следует указать внешний интерфейс роутера. Действие можно не выбирать, так как по умолчанию - **accept**.

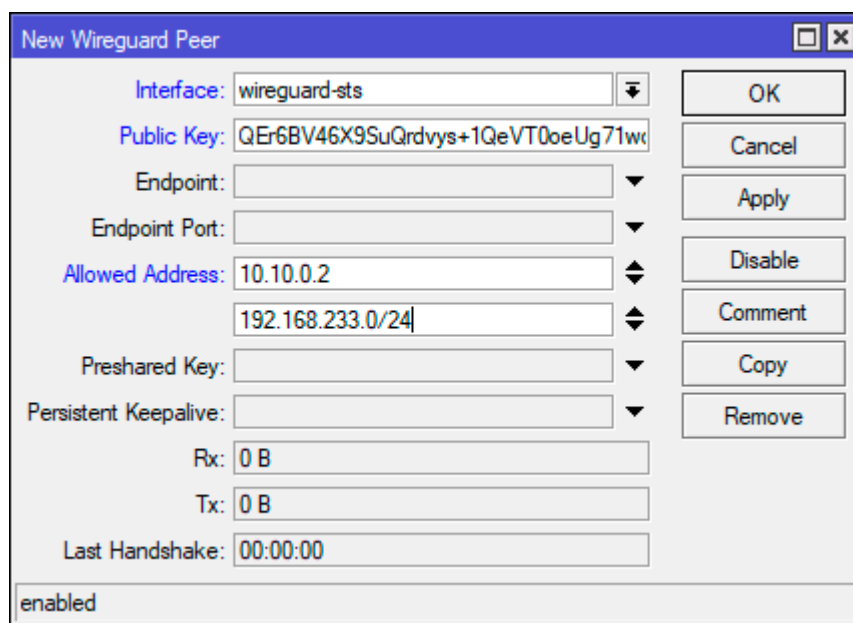


Это же можно сделать командой:

```
/ip firewall filter
add action=accept chain=input dst-port=34567 in-interface=ether1 protocol=udp
```

Данное правило следует расположить перед запрещающим принимать входящие подключения на внешний интерфейс.

Чтобы к нашему роутеру могла подключаться другие узлы нужно создать для каждого из них пир, для этого возвращаемся в **WireGuard - Peers** и создаем новую запись. Здесь нам потребуется **открытый ключ** пира, который следует внести в поле **Public Key** и указать разрешенные сети в **Allowed Address**. В нашем случае мы реализуем сценарий удаленного доступа или объединения сетей, поэтому укажем там внутренний адрес в WireGuard сети, который мы выделили пиру и сеть за ним.

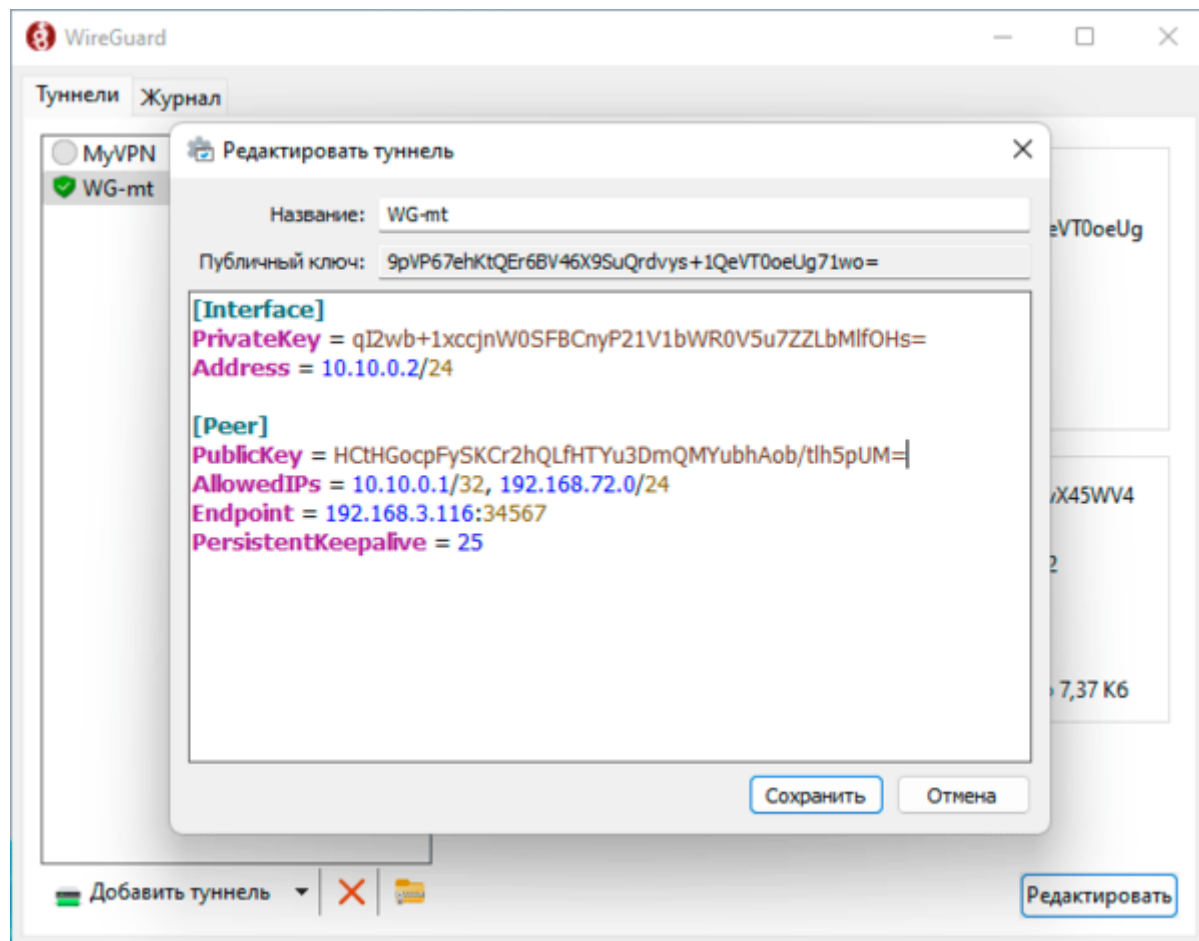


В терминале:

```
/interface wireguard peers
add allowed-address=10.10.0.2/32,192.168.233.0/24 interface=wireguard-sts public-
key="9pVP67ehKtQEr6BV46X9SuQrdvys+1QeVT0oeUg71wo="
```

Еще раз напомним, что вся адресация внутри WireGuard сети назначается администратором вручную и явно прописывается для каждого пира с обеих сторон. Это один из недостатков WireGuard, который следует из его простоты.

С другой стороны, у нас, допустим, будет Windows. Быстро настраиваем там новый туннель, в разделе **Interface** добавляем выделенный узлу адрес, а в разделе **Peer** указываем публичный ключ Mikrotik, его адрес и порт, а в разделе разрешенных адресов добавим адрес WireGuard интерфейса и сети за роутером.



Если вы нигде не ошиблись, то подключение будет установлено, и вы сможете получить доступ к сети за роутером. В случае нахождения пира за NAT не забывайте добавить опцию **PersistentKeepalive**.

```
PS C:\Users\Andrey> ping 10.10.0.1

Обмен пакетами с 10.10.0.1 по с 32 байтами данных:
Ответ от 10.10.0.1: число байт=32 время=1мс TTL=64
Ответ от 10.10.0.1: число байт=32 время<1мс TTL=64
Ответ от 10.10.0.1: число байт=32 время<1мс TTL=64

Статистика Ping для 10.10.0.1:
    Пакетов: отправлено = 3, получено = 3, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 1 мсек, Среднее = 0 мсек
Control-C
PS C:\Users\Andrey> ping 192.168.72.199

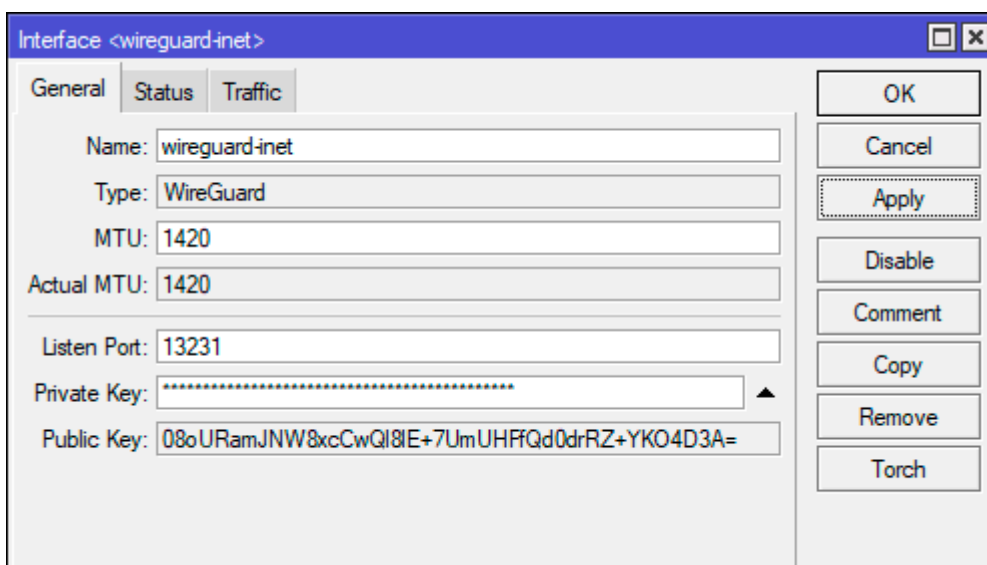
Обмен пакетами с 192.168.72.199 по с 32 байтами данных:
Ответ от 192.168.72.199: число байт=32 время=1мс TTL=127
Ответ от 192.168.72.199: число байт=32 время<1мс TTL=127
Ответ от 192.168.72.199: число байт=32 время<1мс TTL=127
Ответ от 192.168.72.199: число байт=32 время<1мс TTL=127
```

Как видим, ничего сложного нет, но при большом количестве пиров прибавляется ручной работы: вы должны сами распределить адреса и прописать настройки с обеих сторон. Никаких средств автоматизации для этого не предусмотрено.

## Mikrotik как инициатор (клиент)

В данном разделе мы рассмотрим иной сценарий - использование WireGuard для доступа в интернет, но принципиальной разницы нет, если вы соединяете сети, то можно точно также настроить роутер и все будет работать. Просто мы дополнительно рассмотрим некоторые вопросы касающиеся маршрутизации.

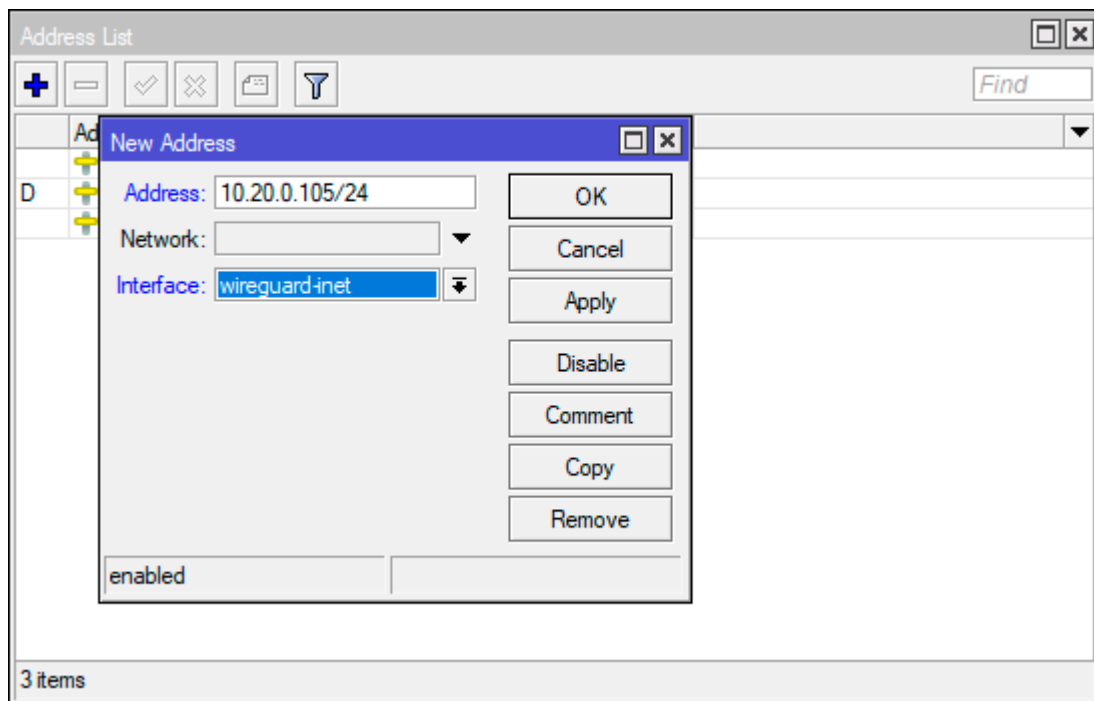
Снова переходим в раздел WireGuard и создаем новый интерфейс. В данном случае достаточно только указать имя, порт нас не интересует, так как мы не собираемся принимать соединения, но его все-таки требуется указать, выбираем любой.



Или вводим команду:

```
/interface wireguard
add listen-port=13231 mtu=1420 name=wireguard-inet
```

Затем назначаем интерфейсу адрес, если все узлы ваши - то назначаете сами, если подключаетесь к чужому респондеру, то вводите адрес, выданный его администратором. Это действие производится в **IP - Addresses**.



Это же действие в терминале:

```
/ip address  
add address=10.20.0.105/24 interface=wireguard-inet network=10.20.0.0
```

Чтобы наш роутер смог куда-то подключиться мы снова должны создать пир. В WireGuard пир - это просто вторая сторона туннеля и не важно мы подключаемся к ней, или она к нам. В любом случае у нас должен быть интерфейс - наша сторона, и пир - противоположная сторона.

Переходим в **WireGuard - Peers** и создаем новый пир, настроек тут будет побольше, указываем: **Interface** - созданный нами интерфейс, **Public Key** - публичный ключ респондера, получаем с той стороны, **Endpoint** и **Endpoint Port** - адрес респондера и его порт, **Allowed Address** - 0.0.0.0/0 - т.е. разрешаем любой трафик в туннеле. Если вы находитесь за NAT, то обязательно добавьте опцию **Persistent Keepalive**, рекомендуемое значение - 25 секунд.

В терминале тоже достаточно длинная команда:

```
/interface wireguard peers
add allowed-address=0.0.0.0/0 endpoint-address=x.x.x.x endpoint-port=34567
interface=wireguard-inet persistent-keepalive=25s public-
key="kKxQ4wF+kUrpsTGwjMvISwX45WV4nixG76/+sKlzeQA="
```

Также не забудьте включить маскардинг для созданного интерфейса, переходим в **IP - Firewall - NAT** и создаем новое правило, на закладке **General: Chain** - srcnat, **Src. Address** диапазон локальной сети, например, 192.168.72.0/24, **Out. Interface** - интерфейс WireGuard, в нашем случае wireguard-inet, на закладке **Action** выбираем действие **masquerade**.

Либо в терминале:

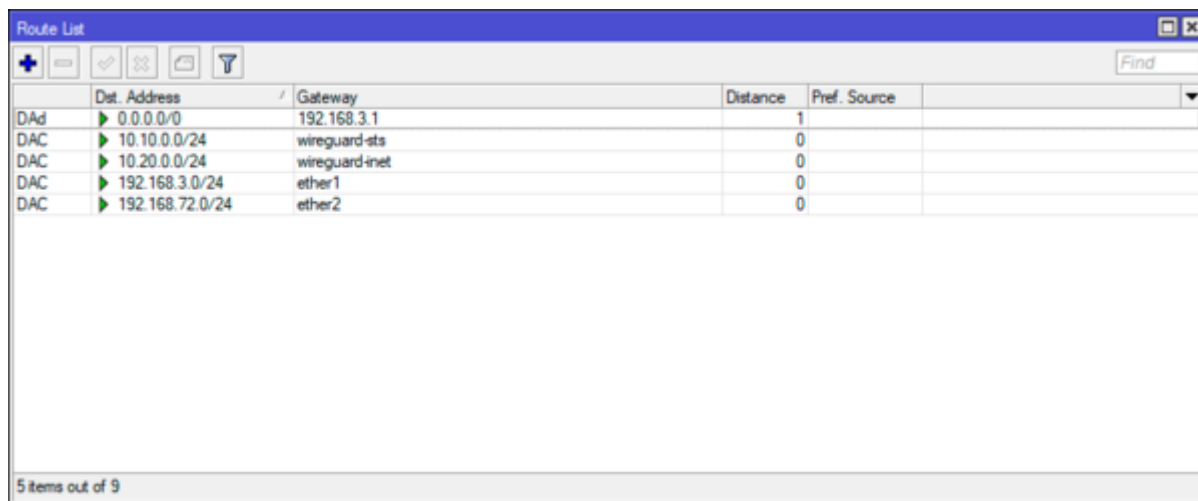
```
/ip firewall nat
add action=masquerade chain=srcnat out-interface=wireguard-inet src-
address=192.168.72.0/24
```

Затем, с другой стороны, также потребуется создать пир для нашего роутера, вам понадобится публичный ключ и назначенный адрес.

```
[Peer]
Publickey = 08oURamJNW8xcCwQl8lE+7UmUHFfQd0drRZ+YK04D3A=
AllowedIPs = 10.20.0.105/32
```

Перезапускаем службу и соединение будет установлено. Но трафик как шел через основного провайдера - так и идет. Почему так? Заглянем в таблицу маршрутизации, которая находится в IP - Routes, как видим нулевой маршрут как был через основного провайдера - так и остался. В отличие от официальных пакетов WireGuard, которые управляют маршрутами на хосте, в Mikrotik все отдано в руки администратора.

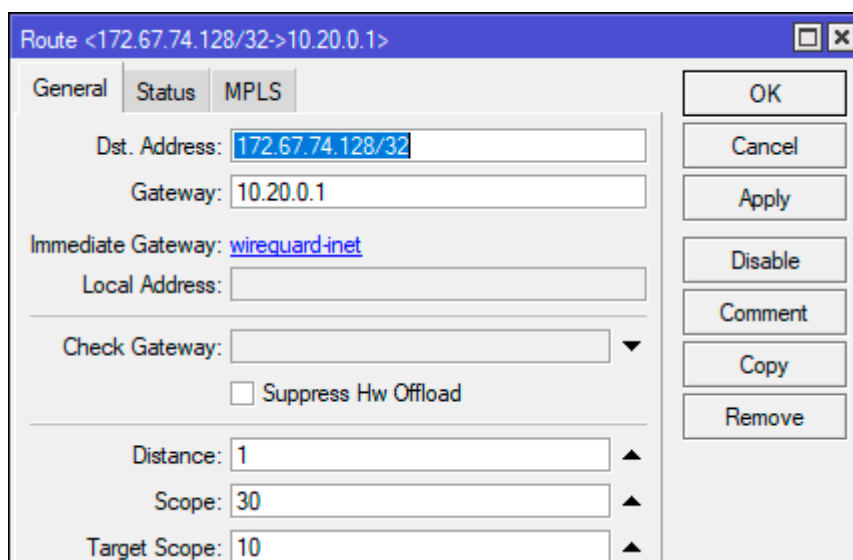




	Dst. Address	Gateway	Distance	Pref. Source
DAd	0.0.0.0/0	192.168.3.1	1	
DAC	10.10.0.0/24	wireguard-ets	0	
DAC	10.20.0.0/24	wireguard-inet	0	
DAC	192.168.3.0/24	ether1	0	
DAC	192.168.72.0/24	ether2	0	

5 items out of 9

А дальше все зависит от того, что именно мы хотим получить. Если нам нужен доступ через туннель к отдельным узлам, то просто достаточно создать для них отдельные маршруты. Создаем новое правило, в котором указываем нужный адрес и шлюз, в качестве которого будет выступать противоположный конец WireGuard туннеля.



Route <172.67.74.128/32->10.20.0.1>

General Status MPLS

Dst. Address: 172.67.74.128/32

Gateway: 10.20.0.1

Immediate Gateway: wireguard-inet

Local Address:

Check Gateway: ☐ Suppress Hw Offload

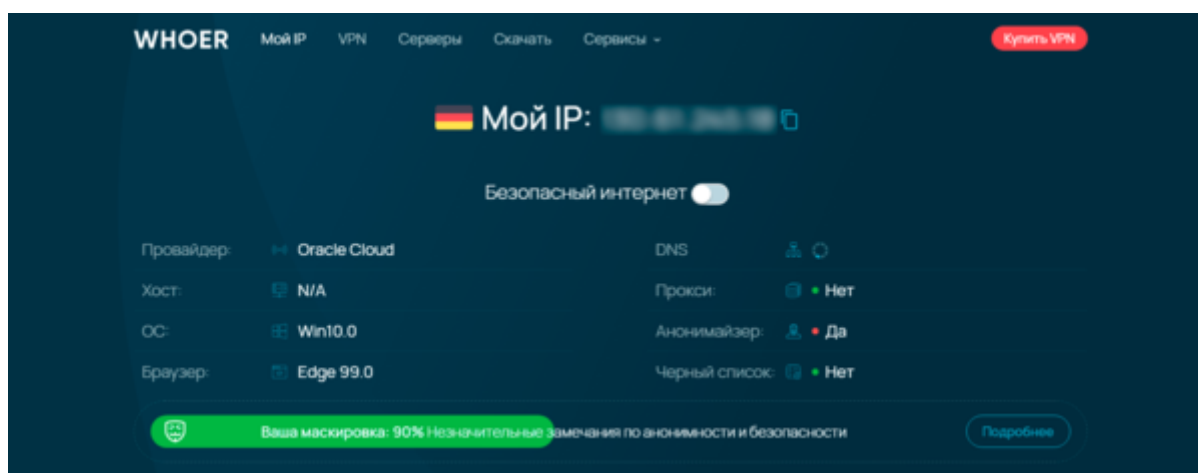
Distance: 1

Scope: 30

Target Scope: 10

OK Cancel Apply Disable Comment Copy Remove

Теперь снова проверяем (мы добавили маршрут к сервису проверки IP) - все хорошо, мы обращаемся к данному узлу через VPN-сервер.



WHOER Мой IP VPN Серверы Скачать Сервисы - Купить VPN

Мой IP: 192.168.3.1

Безопасный интернет ☐

Провайдер: Oracle Cloud DNS: ☐

Хост: N/A Прокси: Нет

ОС: Win10.0 Анонимайзер: Да

Браузер: Edge 99.0 Черный список: Нет

Ваша маскировка: 90% Незначительные замечания по анонимности и безопасности

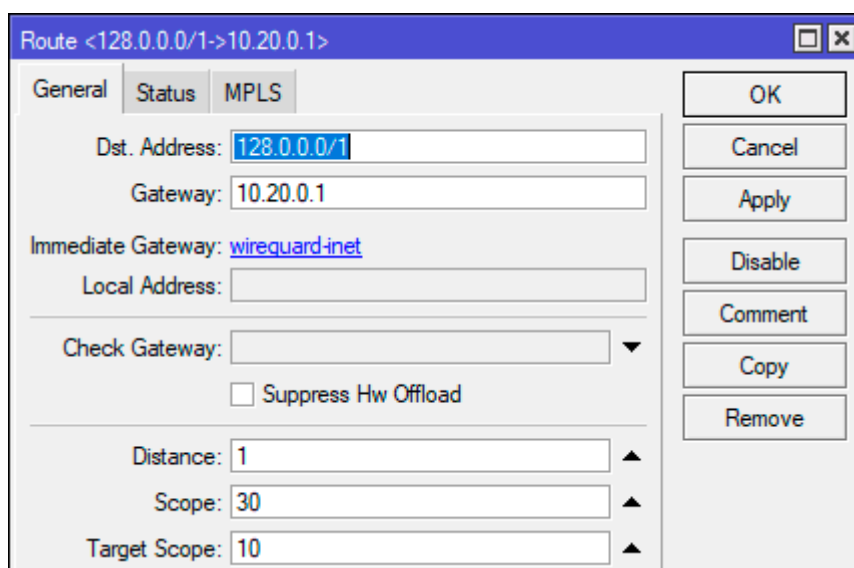
Подробнее

Добавить маршрут из командной строки можно следующим образом:

```
/ip route
add disabled=yes distance=1 dst-address=172.67.74.128/32 gateway=10.20.0.1 pref-
src=0.0.0.0 routing-table=main scope=30 suppress-hw-offload=no target-scope=10
```

Если же мы хотим направить весь интернет трафик в туннель, то нам нужно изменить нулевой маршрут, казалось бы, все просто, но не будем спешить. Обратим внимание на флаги текущего маршрута: **DAd** - динамический, активный, получен по DHCP, можно, конечно отключить получение маршрутов в DHCP-клиенте, но мы пойдем другим путем.

Вспомним, что если к одной цели ведут несколько маршрутов, то будет выбран тот, у которого самая узкая маска. Поэтому вместо одного нулевого маршрута добавим два, к сетям **0.0.0.0/1** и **128.0.0.0/1**.



В терминале выполните две команды:

```
/ip route
add disabled=no distance=1 dst-address=0.0.0.0/1 gateway=10.20.0.1 pref-src=""
routing-table=main scope=30 suppress-hw-offload=no target-scope=10
add disabled=no distance=1 dst-address=128.0.0.0/1 gateway=10.20.0.1 pref-src=""
routing-table=main scope=30 suppress-hw-offload=no target-scope=10
```

Вроде бы все сделано правильно, но интернет вообще перестал работать. Что случилось? Мы только что завернули в туннель **весь** исходящий трафик, в том числе и к нашему VPN-серверу, естественно, что соединение будет невозможно.

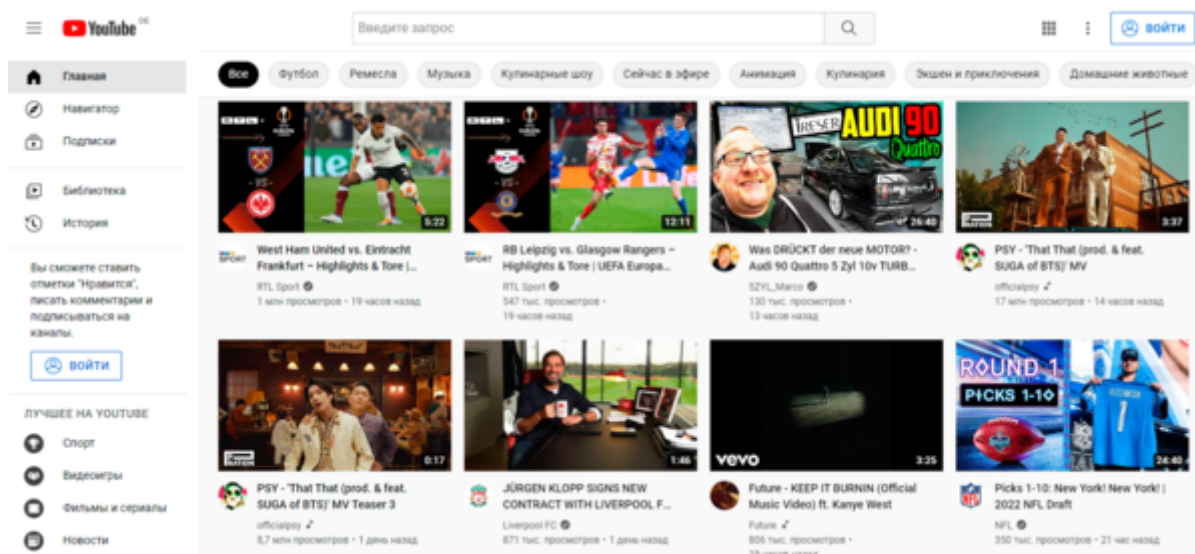
Поэтому добавим еще один маршрут к респондеру через основного провайдера.

Или:

```
/ip route
add disabled=no dst-address=x.x.x.x/32 gateway=192.168.3.1 routing-table=main
suppress-hw-offload=no
```

Где **192.168.3.1** - шлюз основного провайдера.

После чего все снова заработает. При этом уже только по одному внешнему виду сайтов несложно понять, что мы работаем через VPN с точкой выхода в Германии.



Более подробные настройки для того или иного конкретного сценария выходят за рамки данной статьи и, как таковые, уже не относятся к настройкам WireGuard. Сам же WireGuard в RouterOS 7 есть и работает, при этом достаточно несложен в настройке, в чем мы только что убедились.

## 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.

---