

Проброс портов и Hairpin NAT в роутерах Mikrotik

 interface31.ru/tech_it/2019/07/probros-portov-i-hairpin-nat-v-routerah-mikrotik.html

Реклама•16+

Записки IT специалиста

Технический блог специалистов ООО"Интерфейс"

- [Главная](#)
- Проброс портов и Hairpin NAT в роутерах Mikrotik

Проброс портов - на первый взгляд тривиальная повседневная задача, которая не должна вызывать никаких затруднений. Действительно, в бытовых роутерах это так, но если речь идет об оборудовании Mikrotik, то сложности могут не заставить себя ждать. А все потому, что RouterOS дает в руки администратора богатые сетевые возможности, требуя в ответ понимания работы сетей хотя бы на базовом уровне. Конечно, можно настроить все по готовой инструкции, но гораздо лучше понимать каждое свое действие, и именно для тех, кто хочет разобраться предназначена эта статья.



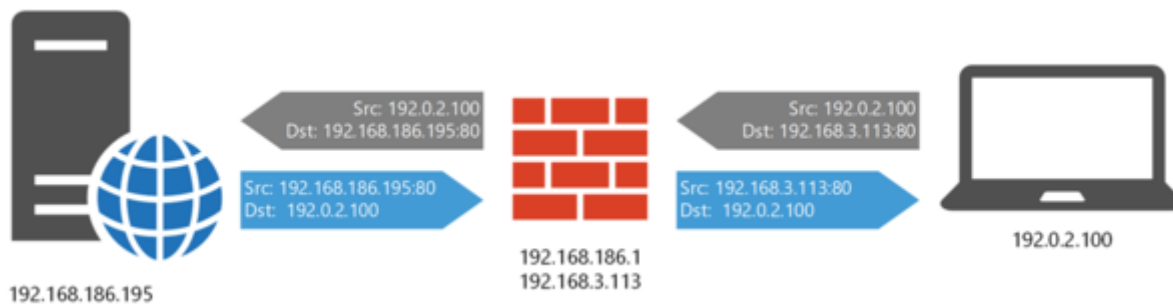
Онлайн-курс по MikroTik

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.

Проброс портов

Проброс, он же форвардинг, портов - что это такое? Это технология, которая позволяет обращаться к узлам, находящимся за маршрутизатором путем перенаправления трафика для определенных портов с внешнего адреса маршрутизатора на внутренний адрес узла в локальной сети. Это становится возможным благодаря технологии NAT.

Давайте рассмотрим небольшую схему, которая поможет понять принцип действия проброса портов.



Допустим, некий удаленный ПК хочет обратиться к нашему веб-серверу, который находится за маршрутизатором в локальной сети. Но обратиться он может только по внешнему адресу маршрутизатора (в нашем примере 192.168.3.113), на который мы пробросили порт 80 веб-сервера. Поэтому он формирует пакет, в котором адресом назначения выступает маршрутизатор и отправляет его получателю, в качестве адреса источника он указывает собственный адрес.

Маршрутизатор, получив такой пакет выполняет замену адреса назначения с собственного, на адрес веб-сервера, также, при необходимости, он может изменить и порт назначения. После чего пакет отправляется в локальную сеть и достигает веб-сервера. Данные о преобразовании заносятся в специальную таблицу трансляции NAT.

Нужно ли менять адрес отправителя? Нет, если маршрутизатор выступает основным шлюзом сети, а в подавляющем большинстве случаев это так. Веб-сервер обработает запрос, а так как он ничего не знает о сети получателя, то обратный пакет будет направлен **шлюзу по умолчанию**, т.е. назад нашему маршрутизатору.

Маршрутизатор на основании данных таблицы трансляции выполнит обратное преобразование, заменив на этот раз адрес источника с внутреннего адреса веб-сервера, на свой внешний адрес и отправит пакет запросившему его узлу.

На первый взгляд все понятно, поэтому перейдем к практике. В RouterOS в качестве сетевого фильтра используется iptables, поэтому далее мы будем оперировать его терминами. В таблице NAT используются две цепочки: PREROUTING - в которую попадают все пришедшие на маршрутизатор пакеты и POSTROUTING - через нее проходят все прошедшие через устройство пакеты. В PREROUTING нам доступно действие dst-nat (DNAT), которое позволяет изменить адрес назначения пакета, в POSTROUTING будут доступны src-nat (SNAT) и masquerade, которые позволяют изменить адрес источника пакета.

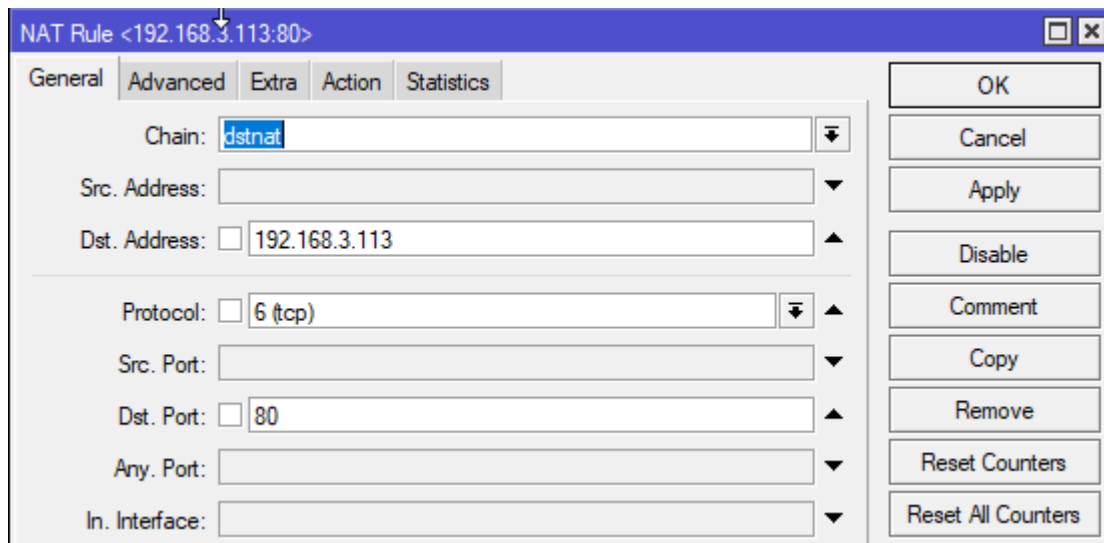
Мы не даром заостряем на этом особое внимание, так как непонимание процесса прохождения пакета по цепочкам сетевого фильтра способно привести к значительным затруднениям, когда вроде-бы все правила составлены верно, но ничего не работает.

Так для пакета от клиента к веб-серверу и обратно порядок прохождения цепочек будет следующим:

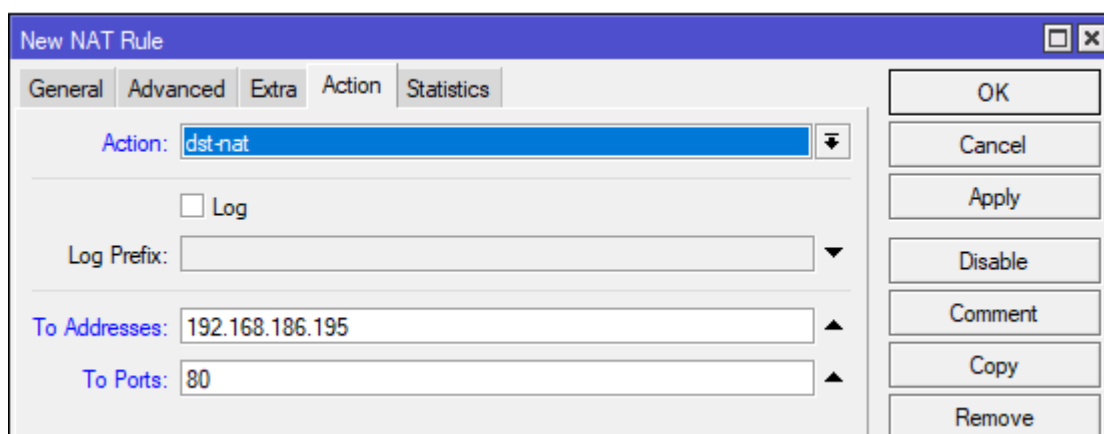
PREROUTING -> FORWARD -> POSTROUTING

Обратите внимание, что пакет не попадает в цепочки INPUT и OUTPUT, которые используются для собственных пакетов маршрутизатора.

Как правильно выполнить настройку? Перейдем в **IP - Firewall - NAT** и добавим следующее правило: **Chain - dstnat** (читай PREROUTING), **Dst. Address - 192.168.1.113** - внешний IP-адрес нашего роутера, **Protocol - tcp** - используемый протокол, если сервис может использовать несколько протоколов, скажем TCP и UDP - потребуется создать отдельное правило для каждого протокола, **Dst. Port - 80** - порт, на котором маршрутизатор будет принимать внешние соединения.



На закладке **Action** укажите: **Action - dst-nat** - выполняем замену адреса получателя, **To Addresses - 192.168.186.195** - внутренний адрес веб-сервера, **To Ports - 80** - порт, на котором веб-сервер принимает соединения. Если внешний и внутренний порт совпадают, то последний можно не указывать.

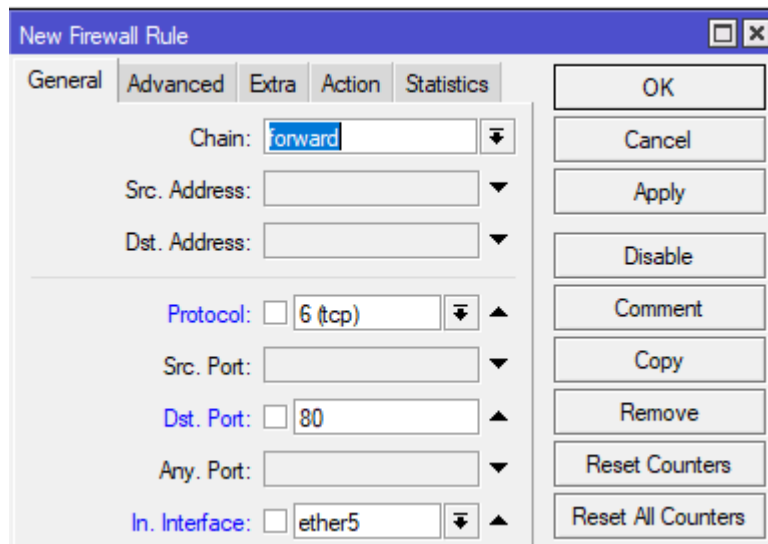


Либо выполните в терминале:

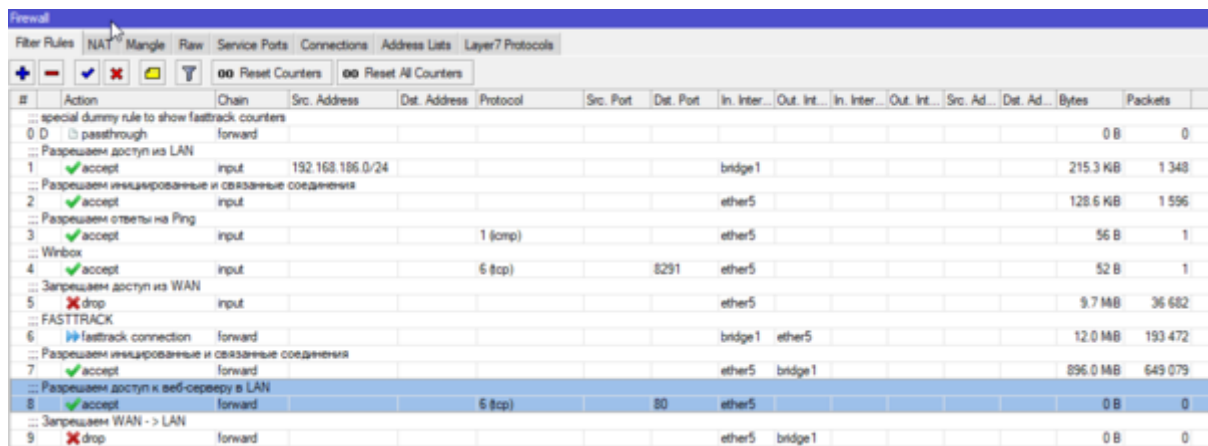
```
/ip firewall nat
add action=dst-nat chain=dstnat dst-address=192.168.3.113 dst-port=80 protocol=tcp
to-addresses=192.168.186.195 to-ports=80
```

Но это еще не все, после PREROUTING пакет попадет в цепочку FORWARD, в которой, если вы настраивали роутер по нашей инструкции, запрещены любые внешние пакеты, кроме инициированных из локальной сети соединений.

Создадим новое правило. **IP - Firewall - Filter Rules**, где добавим: **Chain - forward** - цепочка FORWARD, **Protocol - tcp**, **Dst. Port - 80** - протокол и порт, **In. Interface - ether5** - внешний интерфейс вашего роутера. Так как по умолчанию в новых правилах стоит действие **accept**, на закладку **Action** можно не переходить.



Располагаться данное правило должно выше правила, запрещающего транзитный трафик из внешней сети во внутреннюю.



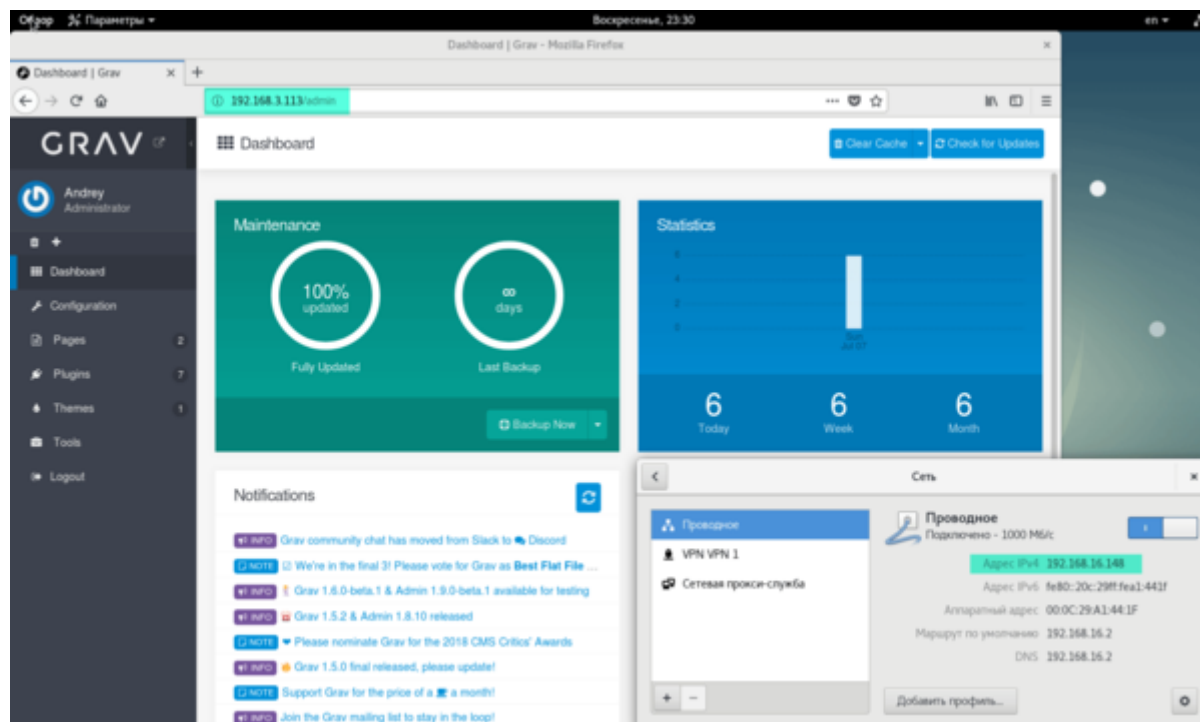
#	Action	Chain	Src. Address	Dst. Address	Protocol	Src. Port	Dst. Port	In. Inter.	Out. Inter.	In. Inter.	Out. Inter.	Src. Ad.	Dst. Ad.	Bytes	Packets
0	special dummy rule to show fasttrack counters	forward												0 B	0
1	Разрешаем доступ из LAN	input	192.168.186.0/24					bridge1						215.3 KB	1 348
2	Разрешаем инициированные и связанные соединения	input						ether5						128.6 KB	1 596
3	Разрешаем ответы на Ping	input			1 (icmp)			ether5						56 B	1
4	Winbox	input			6 (tcp)		8291	ether5						52 B	1
5	Запрещаем доступ из WAN	input						ether5						9.7 MB	36 682
6	FASTTRACK	forward						bridge1	ether5					12.0 MB	193 472
7	Разрешаем инициированные и связанные соединения	forward						ether5	bridge1					896.0 MB	649 079
8	Разрешаем доступ к веб-серверу в LAN	forward			6 (tcp)		80	ether5						0 B	0
9	Запрещаем WAN -> LAN	forward						ether5	bridge1					0 B	0

Обратите внимание, что если вы пробрасываете порт с изменением номера, скажем внутренний 3389 (RDP) на внешний 3390, то в правиле брандмауэра вы всегда должны указывать **внутренний порт**, т.е. 3389, так как пакет уже прошел цепочку PREROUTING и данные о получателе в нем уже изменены на адрес и порт внутреннего сервера.

Из терминала это можно сделать командами:

```
/ip firewall filter
add action=accept chain=forward dst-port=80 in-interface=ether5 protocol=tcp
```

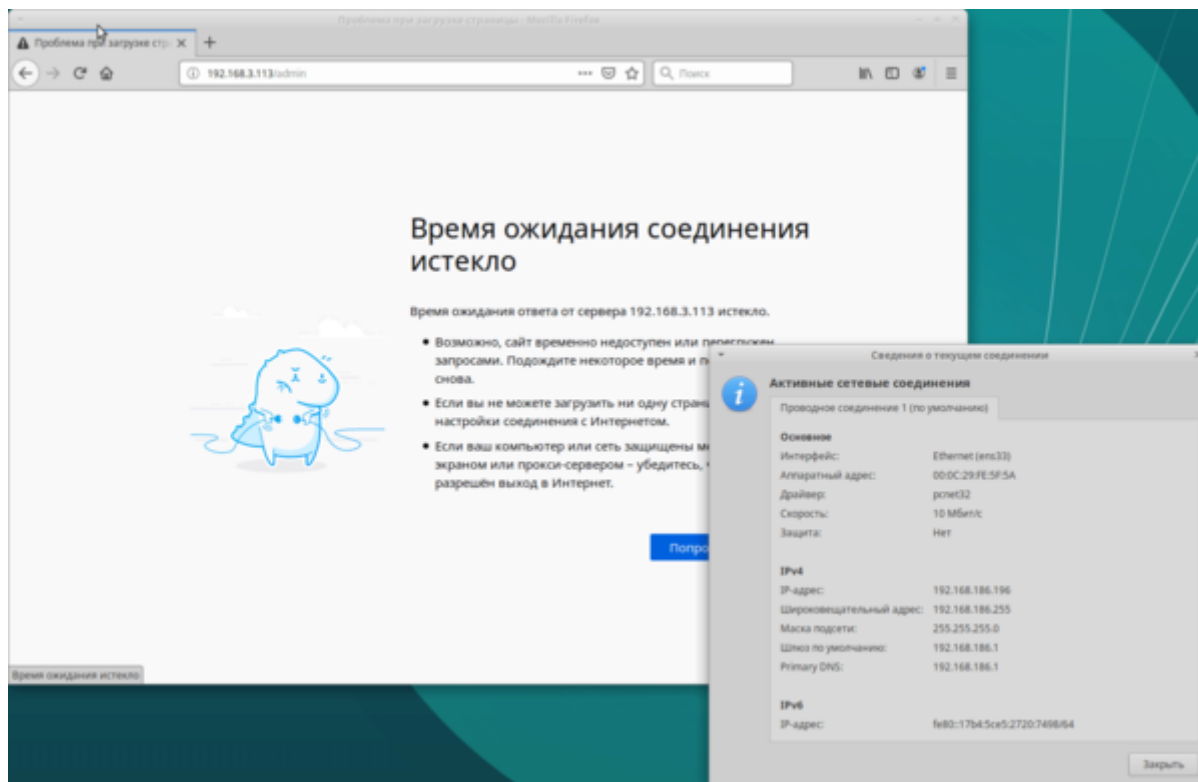
Осталось только проверить работу наших правил, попробуем получить доступ к веб-серверу со внешнего узла:



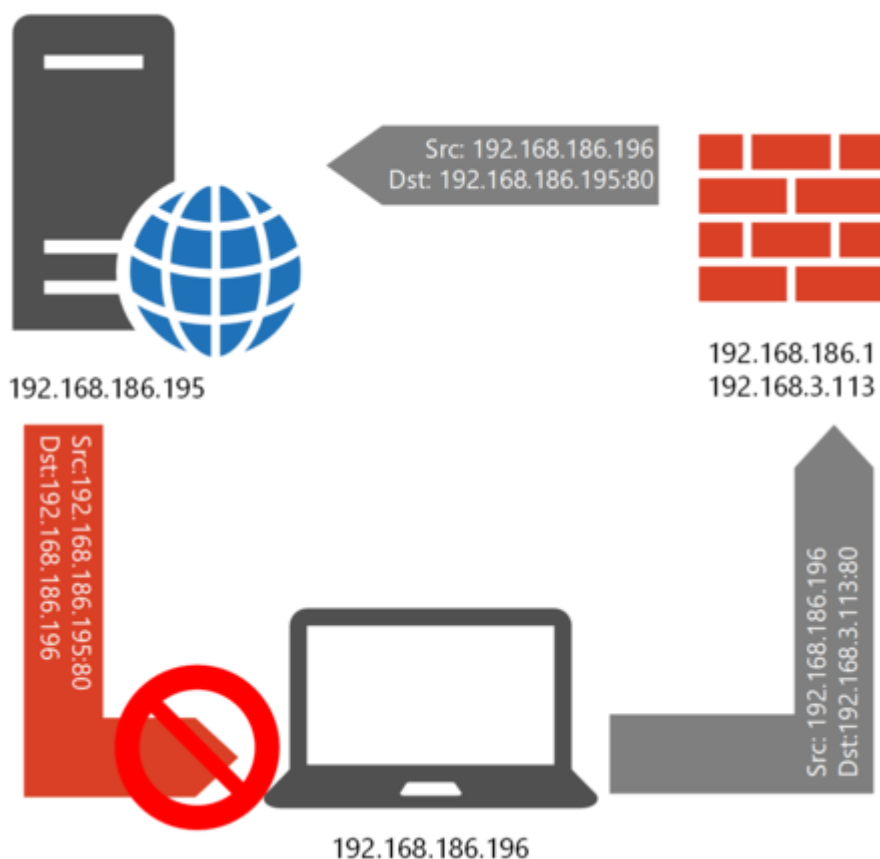
Как видим, все прекрасно работает.

Hairpin NAT

Все это хорошо, но ровно до тех пор, пока мы не попытаемся получить доступ по внешнему адресу из внутренней сети. Вообще, таких ситуаций следует избегать, предпочтительно использовать для доступа доменные имена и двойной горизонт DNS, когда для внешних пользователей одно и то же имя разрешается во внешний адрес, а для внутренних - во внутренний. Но это возможно не всегда. Попробовав же обратиться изнутри по внешнему адресу, мы получим ошибку соединения:



Почему так происходит? Давайте рассмотрим еще одну схему:

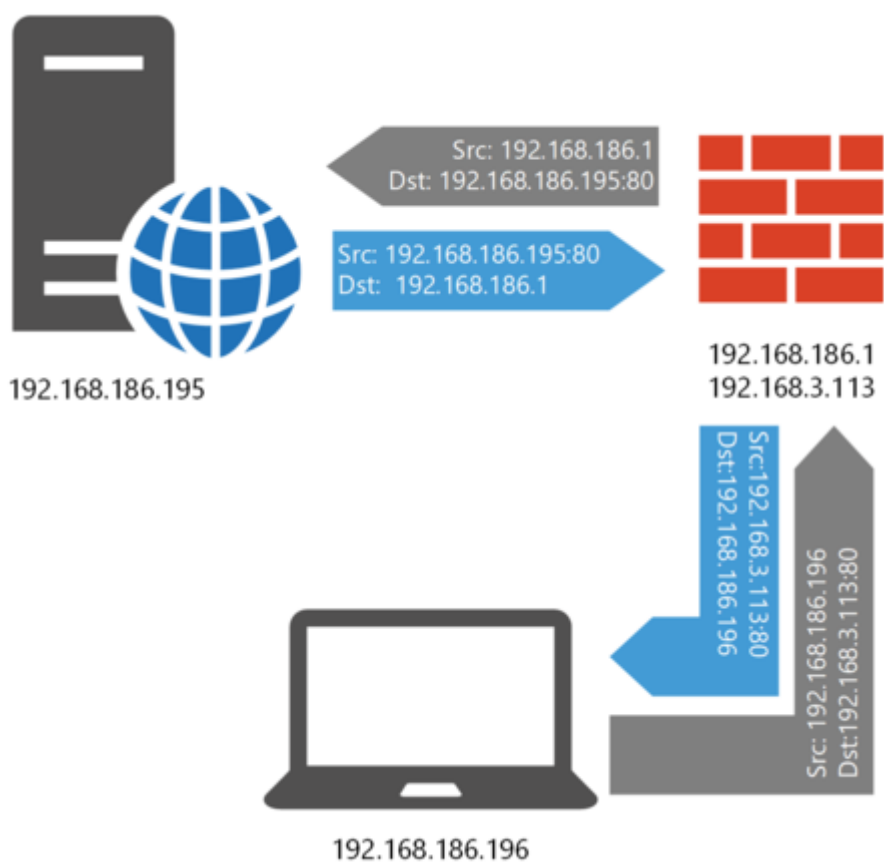


Первая ее часть нам уже знакома, узел отправляет запрос на внешний адрес маршрутизатора, он заменяет адрес назначения адресом внутреннего сервера и отправляет пакет к нему, адрес отправителя остается неизменным. А вот дальше начинается самое интересное, веб-сервер видит, что отправитель находится в нем в

одной сети и отправляет ответ ему напрямую. Но отправитель направлял запрос на внешний адрес сервера и ждет ответа именно от него, поэтому ответный пакет, отправителем которого указан внутренний адрес веб-сервера будет отброшен и связь установить не удастся.

Что же делать? Очевидно, что нужно каким-то образом заставить веб-сервер отвечать не напрямую клиенту, а пересылать пакет обратно маршрутизатору. Здесь нам на помощь придет действие src-nat (SNAT), которое позволяет изменить адрес отправителя, находящееся в цепочке POSTROUTING. В терминах Mikrotik данная настройка носит наименование **Hairpin NAT**.

Чтобы понять, как это работает мы подготовили следующую схему:



Теперь наш маршрутизатор не только изменяет адрес назначения, но и адрес источника, указывая в его качестве собственный внутренний IP. Обработав такой пакет веб-сервер отправит его обратно к маршрутизатору, который выполнит обратные преобразования (согласно таблице трансляции) и оправит его получателю. А так как отвечать будет именно тот узел, к которому был отправлен запрос, то в данном случае все будет работать.

Для настройки на Mikotik перейдем в **IP - Firewall - NAT** и добавим: **Chain - src-nat** (POSTROUTING), **Src. Address - 192.168.186.0/24** - диапазон вашей локальной сети, **Dst. Address - 192.168.186.195** - внутренний адрес веб-сервера, **Protocol - tcp**, **Dst. Port - 80** - протокол и порт.

Обратите внимание, что в качестве адреса и порта назначения мы указываем **внутренние адрес и порт**, так как пакет уже прошел цепочку PREROUTING, где данные получателя были изменены. К сожалению, не все это понимают, во многих инструкциях в сети в данном правиле фигурирует внешний адрес, стоит ли говорить, что такое правило работать не будет.

Затем переходим на закладку **Action** и указываем: **Action - src-nat (SNAT)**, **To Addresses - 192.168.186.1** - указываем внутренний адрес нашего маршрутизатора.

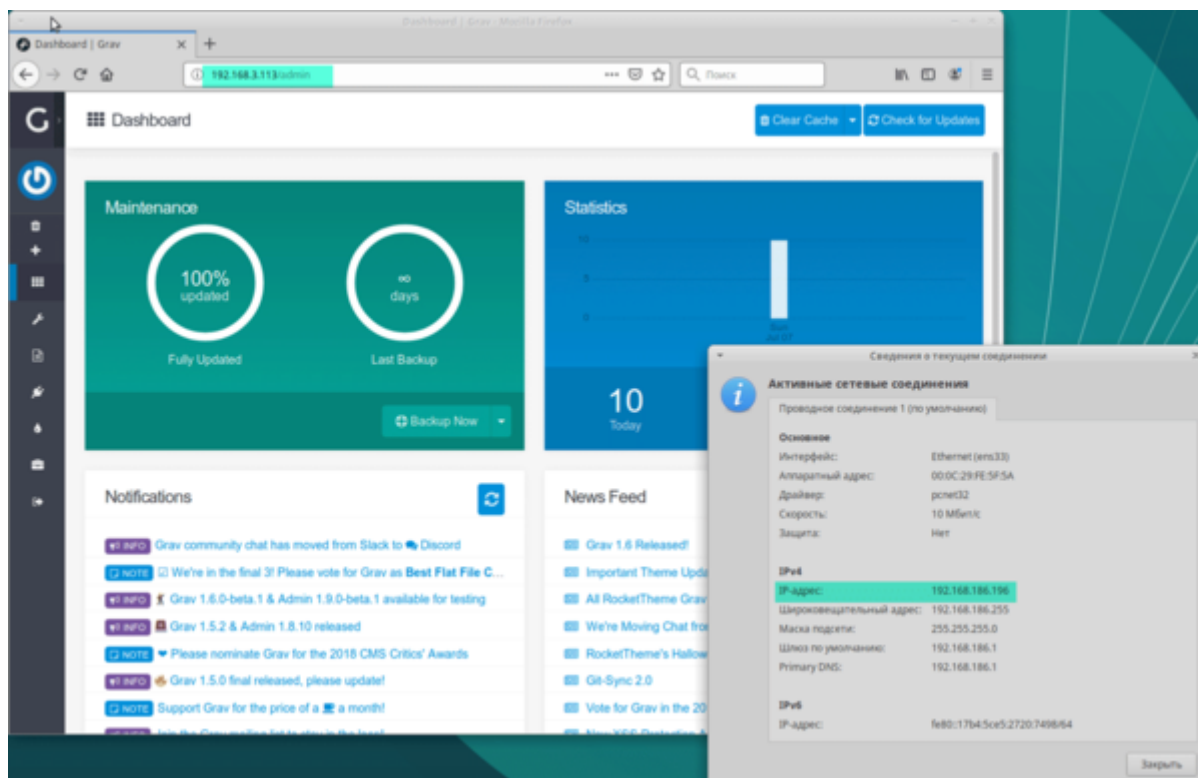
Через терминал данное правило можно создать следующим образом:

```
/ip firewall nat
add action=src-nat chain=srcnat dst-address=192.168.186.195 dst-port=80
protocol=tcp src-address=192.168.186.0/24 to-addresses=192.168.186.1
```

Во многих иных статьях для данного правила используется действие masquerade, давайте разберемся в чем разница. В первом приближении оба действия делают одно и то же - изменяют адрес источника пакета, но **src-nat** работает только со статическими адресами, зато позволяет указать в качестве источника любой адрес, **masquerade** работает с динамическими адресами, но в качестве адреса источника может использовать только адрес того интерфейса, с которого уходит пакет. За счет того, что masquerade при каждом запросе определяет адрес интерфейса - это действие требует больше вычислительных ресурсов, нежели src-nat.

В нашем случае все адреса статические, поэтому использование src-nat здесь будет более уместно.

Правило создано, проверим его работу:



Если вы нигде не допустили ошибок - все будет работать, как и предполагалось.

Подводя итог, можем сказать, что в пробросе портов нет ничего сложного, но только в том случае, если вы представляете в каком порядке и как обрабатываются пакеты, в противном случае даже такая элементарная задача способна превратиться в длительные мучения на ровном месте. Поэтому мы надеемся, что данный материал пригодится вам не только в практическом плане, но и позволит повысить собственный уровень знаний.

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.