


# Основы iptables для начинающих. Часть 4. Таблица nat - типовые сценарии использования

 [interface31.ru/tech\\_it/2021/08/osnovy-iptables-dlya-nachinayushhih-chast-4-tablica-nat-tipovye-scenarii-ispolzovaniya.html](https://interface31.ru/tech_it/2021/08/osnovy-iptables-dlya-nachinayushhih-chast-4-tablica-nat-tipovye-scenarii-ispolzovaniya.html)

## Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Основы iptables для начинающих. Часть 4. Таблица nat - типовые сценарии использования

В [прошлом материале](#) нашего цикла мы рассмотрели таблицу NAT брандмауэра iptables и основные приемы работы с ней. Данная таблица широко используется для выхода устройств локальной сети в интернет через единственный шлюз или для проброса портов. Достигается это за счет изменения сетевых адресов пакетов и требует определенных теоретических знаний, которые мы привели в предыдущей статье. Теперь же рассмотрим типовые сценарии использования, а также проблемы и тонкости, с которыми вы можете столкнуться.



### Онлайн-курс по устройству компьютерных сетей

На углубленном курсе "[Архитектура современных компьютерных сетей](#)" вы с нуля научитесь работать с Wireshark и «под микроскопом» изучите работу сетевых протоколов. На протяжении курса надо будет выполнить более пятидесяти лабораторных работ в Wireshark.

### Выход в интернет

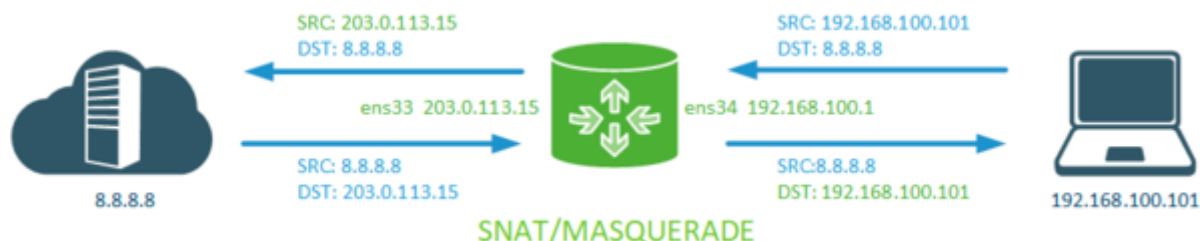
Для выхода устройств, не обладающих собственным выделенным IP-адресом в сеть интернет, используется пограничное устройство, способное маршрутизировать пакеты, т.е. передавать от одного интерфейса к другому. Для его обозначения используют термины: роутер, маршрутизатор, шлюз. Все они обозначают одно и то же устройство. Для того, чтобы включить маршрутизацию в Linux необходимо в файл /etc/sysctl.conf добавить строку:

```
net.ipv4.ip_forward = 1
```

И перечитать настройки командой:

```
sysctl -p
```

Теперь нам нужно отправить все пакеты из локальной сети с назначением в сеть интернет с внешнего интерфейса маршрутизатора таким образом, чтобы они вернулись назад. В этом нам поможет SNAT, который заменит внутренние адреса источников пакетов, на внешний адрес роутера. Ниже показана условная схема, на которой отображены адреса и интерфейсы, все преобразования адресов производимые SNAT выделены зеленым цветом.



Наиболее простым способом будет использование частного случая **SNAT - MASQUERADE**, он позволяет работать с динамическими внешними адресами и для каждого пакета заново определяет исходящий адрес. В случае остановки интерфейса все установленные соединения разрываются, так как при его включении может быть получен новый IP-адрес. При настройке маскардинга хорошим тоном будет указать диапазон адресов сети источника, к которому должно применяться преобразование адресов. В противном случае через ваш шлюз смогут выходить в сеть интернет совсем не те, кому бы вы хотели предоставить доступ.

Поэтому добавим следующее правило:

```
iptables -t nat -A POSTROUTING -o ens33 -s 192.168.100.0/24 -j MASQUERADE
```

Оно достаточно просто и разбиралось нами в предыдущей статье, поэтому не будем на нем останавливаться подробно. Оно предписывает изменить адрес источника на внешний адрес интерфейса **ens33** всем пакетам из сети **192.168.100.0/24** выходящим через этот интерфейс.

Если вы являетесь обладателем выделенного IP-адреса, то можете снизить нагрузку на устройство используя **SNAT**:

```
iptables -t nat -A POSTROUTING -o ens33 -s 192.168.100.0/24 -j SNAT --to-source 203.0.113.15
```

Обязательными критериями в обоих случаях являются интерфейс выхода и диапазон адресов сети источника.

В иных сценариях нужно предоставить доступ к сети интернет VPN-пользователям, которые подключаются непосредственно к маршрутизатору. Проще всего с OpenVPN, в этом случае существует единая сеть /24 для всех клиентов и достаточно просто указать ее в качестве сети-источника пакетов. Если сетей несколько, то можно их перечислить через запятую:

```
iptables -t nat -A POSTROUTING -o ens33 -s 10.8.8.0/24, 10.9.9.0/24 -j MASQUERADE
```

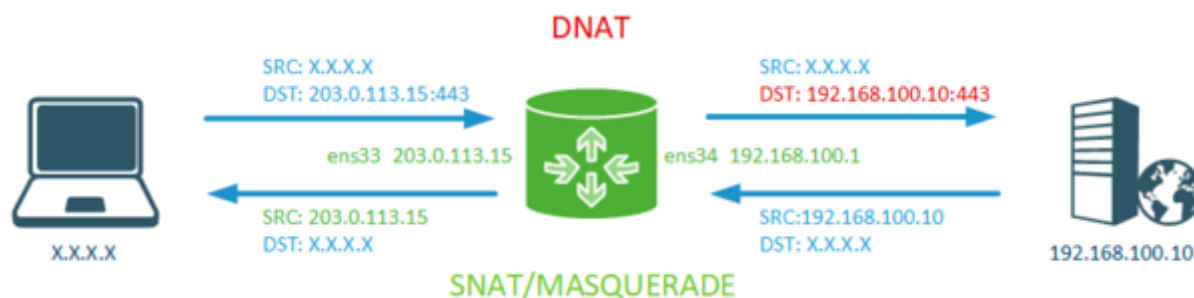
При наличии выделенного внешнего адреса **MASQUERADE** также рекомендуется заменить на **SNAT**.

Если у вас несколько исходящих интерфейсов, то настроить **SNAT/MASQUERADE** нужно для каждого из них.

## Проброс портов

Еще одна часто встречающаяся в повседневной практике задача. Проброс портов предназначен для предоставления внешним пользователям доступа к некоторым внутренним службам используя единственный внешний IP-адрес. Как следует из названия задачи для этой цели мы будем использовать порты. Напомним вам, что протокол IP не использует понятия порта, он оперирует только адресами источника и назначения. Порт - понятие транспортного уровня (L4) для протоколов его использующих, например, TCP или UDP. Заголовки сегмента или дейтаграммы не содержат адресов, но зато содержат порты источника и назначения. Иные протоколы, скажем, GRE не используют порты, поэтому они испытывают проблемы с прохождением NAT.

Ниже показана условная схема, преобразования **DNAT** на ней показаны красным, **SNAT** - зеленым.



Итак, некий узел, совершенно не важно какой, хочет получить доступ к веб-серверу, находящемуся в нашей внутренней сети. Но сервер находится за роутером и не имеет выделенного IP-адреса, поэтому все подключения от внешних клиентов мы будем принимать на внешний адрес роутера и выполнять на нем преобразование адресов назначения - **DNAT**. В самом простом случае внешний и внутренний порты сервиса совпадают, т.е. мы принимаем соединения на 443 порт и отправляем его на такой же порт внутреннего сервера.

Для этого добавим следующее правило:

```
iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 443 -j DNAT --to-destination 192.168.100.10
```

Согласно нему все пакеты, приходящие на внешний интерфейс и имеющие порт назначения в TCP-сегментах 443, будут перенаправлены внутреннему узлу 192.168.100.10. Так как мы не указали порт назначения, то он не будет изменен в

заголовках, что и не требовалось. Адресом источника пакета по-прежнему остается некий внешний адрес.

Получив такой пакет веб-сервер формирует ответ, с назначением на адрес источника, так как он не принадлежит локальной сети, то такой пакет будет направлен шлюзу, на котором произойдет его обработка в уже существующем правиле **SNAT/MASQUERADE**, который заменит внутренний адрес источника внешним адресом роутера. Таким образом удаленный узел будет считать, что работает непосредственно с веб-сервером, не получая никакой дополнительной информации о реальной конфигурации сети с противоположной стороны.

Если внешний и внутренний порты отличаются, то правило будет иметь несколько иной вид:

```
iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 443 -j DNAT --to-destination 192.168.100.10:4443
```

Обратите внимание, что в качестве критерия **--dport** мы указываем порт, используемый на внешнем интерфейсе, а в параметре **--to-destination** реальный порт назначения.

Оба указанных правила будут принимать соединения с любого адреса на внешнем интерфейсе **ens33**, но, если там у нас настроено несколько адресов, но принимать подключения мы хотим только с одного из них? Нужно изменить критерий в условии правила:

```
iptables -t nat -A PREROUTING -d 203.0.113.15 -p tcp --dport 443 -j DNAT --to-destination 192.168.100.10
```

Теперь, вместо пакетов, приходящих на интерфейс **ens33**, мы принимаем только те, которые предназначены адресу **203.0.113.15**.

Также напомним, что TCP и UDP-порты, не смотря на одинаковые номера являются различными и могут без проблем работать одновременно. Допустим у нас в сети есть два OpenVPN-сервера, один с транспортом UDP для большинства клиентов, а второй с TCP, например, на базе Mikrotik. Нам нужно пробросить их оба на внешний интерфейс. Нет ничего сложного:

```
iptables -t nat -A PREROUTING -i ens33 -p udp --dport 1194 -j DNAT --to-destination 192.168.100.14
iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 1194 -j DNAT --to-destination 192.168.100.15
```

Один адрес, один порт, но разные протоколы позволяют по-разному обрабатывать трафик.

В некоторых случаях бывает нужно пробросить несколько внешних портов на один внутренний. Скажем некоторое время ваш RDP-сервер был сброшен на порт 3390, так как порт 3389 занимал другой сервер. Затем его убрали, но перенастроить

большое количество внешних клиентов быстро невозможно, да и не нужно:

```
iptables -t nat -A PREROUTING -i ens33 -p tcp -m multiport --dports 3389, 3390 -j DNAT --to-destination 192.168.100.11:3389
```

Здесь мы подключили внешний модуль **multiport**, позволяющий указывать в критерии несколько портов через запятую, но в параметре **--to-destination** обязательно нужно указать порт назначения, иначе они будут проброшены как есть, без изменения.

Кстати, этим можно воспользоваться для веб-сервера, пробросив сразу порты HTTP и HTTPS:

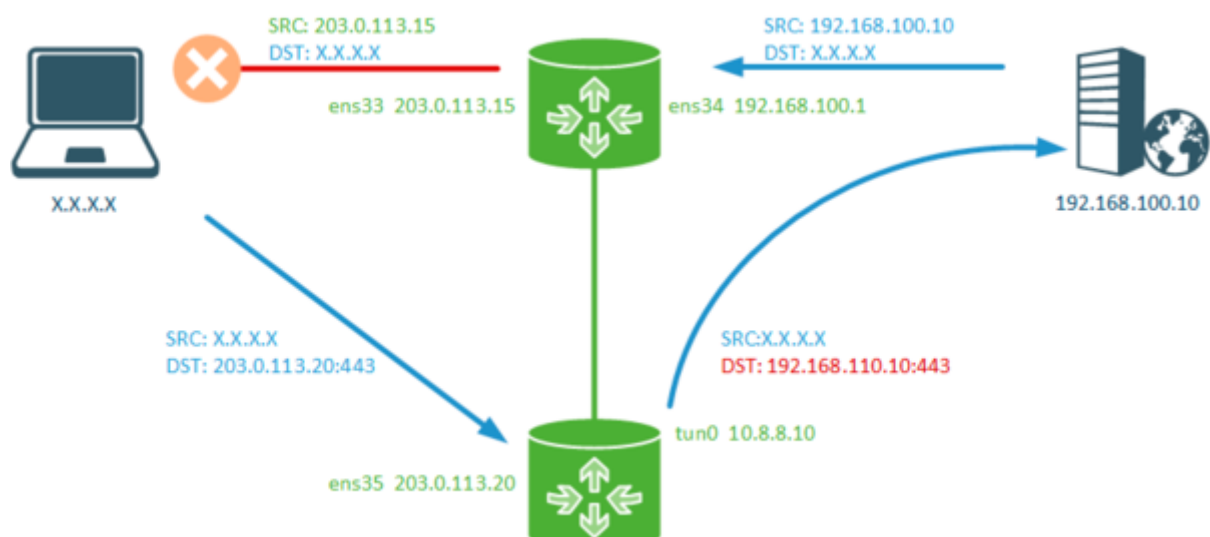
```
iptables -t nat -A PREROUTING -i ens33 -p tcp -m multiport --dports 80, 443 -j DNAT --to-destination 192.168.100.10
```

Так как мы не указали порт назначения, то порт 80 будет проброшен на этот же самый порт узла назначения, а 443 на 443.

## Проброс портов с узла, не являющегося основным шлюзом сети

Это более редкий и более сложный сценарий, но мы рассмотрим его, максимально приблизив к реальным задачам. Допустим у нас есть филиал, в котором есть собственный маршрутизатор и был собственный внутренний веб-сервер, на работу с которым были настроены внешние клиенты. Затем этот веб-сервер упразднили, перенесли все на сервер в центральном офисе. Чтобы не перенастраивать клиентов решили просто изменить проброс портов с узла внутренней сети филиала на узел центрального офиса. Казалось бы, в чем проблема? Между площадками стабильный VPN-канал, настроена маршрутизация, бегают пинги... Но внезапно ничего не работает! Почему?

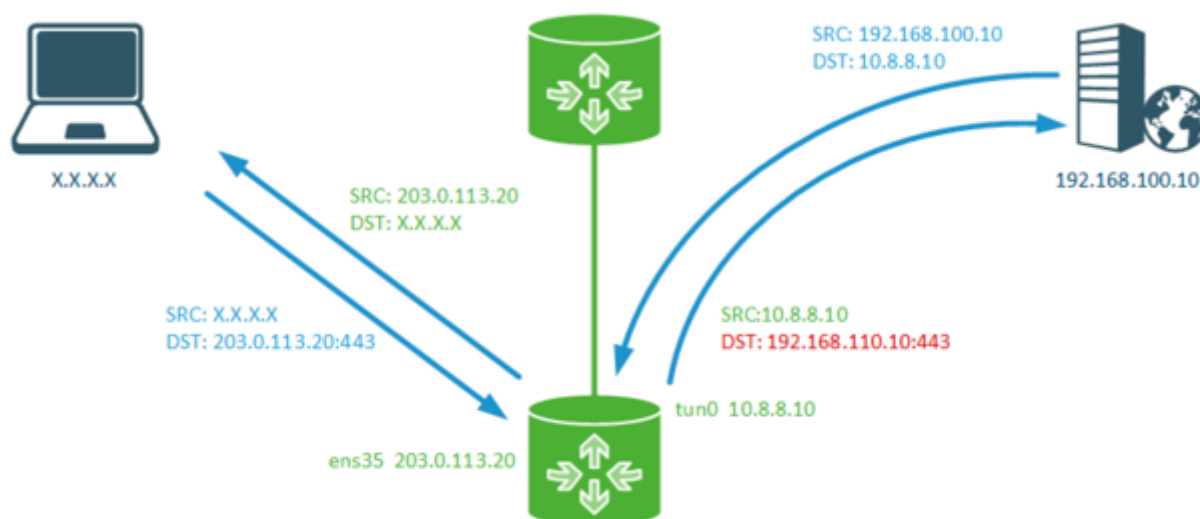
Давайте разберем следующую схему, как и ранее **DNAT**-преобразования показаны красным, **SNAT** - зеленым.



Некий удаленный узел отправляет пакет веб-серверу **203.0.113.20**, указанный узел получает такой пакет и меняет в нем адрес назначения на реальный адрес веб-севера в сети центрального офиса. Адрес источника при этом не меняется. Веб-сервер, получив такой пакет формирует ответ и так как источник находится за пределами внутренней сети и к нему нет отдельного маршрута, то он будет отправлен основному шлюзу **своей сети**. Т.е. маршрутизатору центрального офиса, а не маршрутизатору филиала! Тот выполнит подмену адреса источника на свой внешний адрес и отправит пакет по назначению, но получатель его отбросит, т.к. он пытался установить связь с узлом **203.0.113.20**, а отвечает ему почему-то **203.0.113.15**, такой пакет получит статус **INVALID** и будет заблокирован брандмауэром.

Как быть? Очевидно, что нам нужно принять меры для того, чтобы ответный пакет был отправлен маршрутизатору филиала, для этого мы снова используем **SNAT**, заменив адрес источника пакета внутренним адресом маршрутизатора филиала, так как он подключен через VPN, то этим адресом следует указать его адрес в VPN-сети.

Отлично, пакет вернется в филиал, но, чтобы успешно отправить его по назначению, нам нужно выполнить еще один SNAT, теперь уже для покидающего маршрутизатор пакета, заменив адрес источника - сервера в сети центрального офиса, адресом внешнего интерфейса. Обратите внимание, что адрес назначения ответного пакета будет изменен автоматически, согласно записи в таблице трансляции, которую сделал первый SNAT.



Какие правила нам для этого потребуются? Прежде всего выполним **DNAT** для пришедшего на внешний интерфейс пакета:

```
iptables -t nat -A PREROUTING -i ens35 -p tcp --dport 443 -j DNAT --to-destination 192.168.100.10
```

На выходе из брандмауэра изменим адрес источника при помощи **SNAT**, указав внутренний адрес маршрутизатора филиала:

```
iptables -t nat -A POSTROUTING -d 192.168.100.10 -p tcp --dport 443 -j SNAT --to-source 10.8.8.10
```

Внимательный читатель заметит, что под это правило попадут не только внешние пакеты, но и пакеты локальной сети филиала, направленные веб-серверу. Но ничего страшного в этом случае не произойдет, работа сети не будет нарушена, но между веб-сервером и филиалом появится еще один NAT. На что это может повлиять? Как минимум на то, что вместо реальных адресов клиентов в логи будет попадать внутренний адрес маршрутизатора. Чтобы изменить такое поведение усложним правило:

```
iptables -t nat -A POSTROUTING ! -s 192.168.200.0/24 -d 192.168.100.10 -p tcp --dport 443 -j SNAT --to-source 10.8.8.10
```

В него мы добавили новый критерий: адрес источника не должен принадлежать диапазону сети филиала, т.е. **192.168.200.0/24**.

Теперь осталось выполнить обратный **SNAT** для ответных пакетов:

```
iptables -t nat -A POSTROUTING -o ens35 -s 192.168.100.10 -j SNAT --to-source 203.0.113.20
```

Данное правило для всех пакетов, покидающих роутер с внешнего интерфейса ens35 и источником с адресом веб-сервера, выполняет замену адреса источника на внешний адреса маршрутизатора. Нужно ли здесь учитывать пакеты из локальной сети филиала? Нет, так как для них исходящим интерфейсом будет интерфейс, смотрящий в локальную сеть, а не наружу.

В случае динамического внешнего адреса замените **SNAT** на **MASQUERADE**:

```
iptables -t nat -A POSTROUTING -o ens35 -s 192.168.100.10 -j MASQUERADE
```

Ну и не забываем при построении всех остальных правил, что **DNAT** выполняется при входе в брандмауэр, до принятия всех решений о маршрутизации, а **SNAT** после, на выходе.

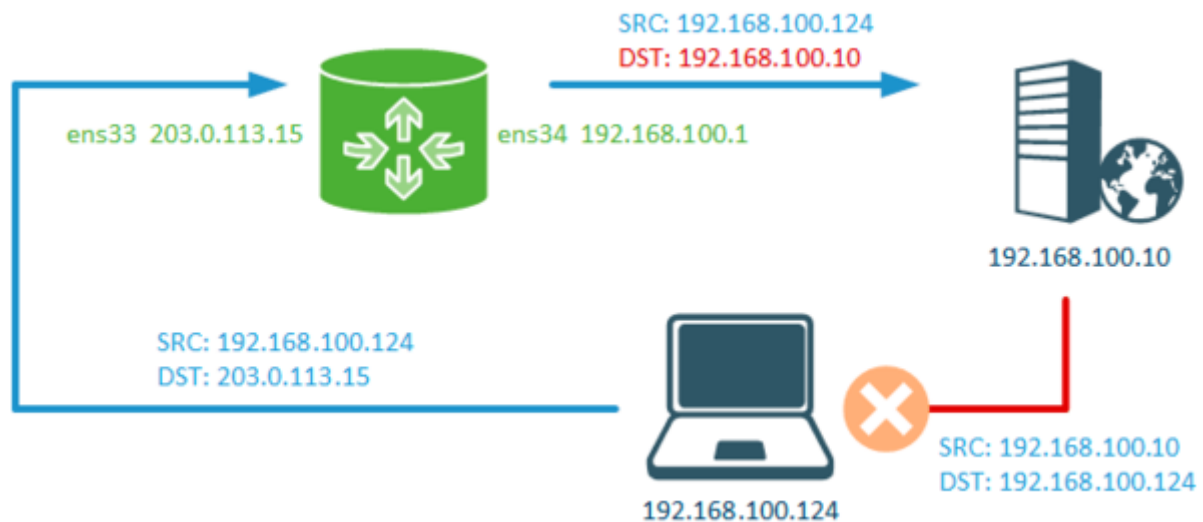
## NAT loopback или Hairpin NAT

---

Несмотря на интересное название, это очень распространенный случай. Мобильный клиент, для которого настроен проброс портов на внешнем интерфейсе приходит в офис и у него внезапно все перестает работать. Можно, конечно, настроить несколько вариантов подключения, для нахождения в офисе и для нахождения вне его, но зачем?

Прежде всего давайте разберемся, почему вдруг все сломалось, для этого внимательно рассмотрим следующую схему:





Мобильный клиент, с адресом принадлежащим офисной сети, обращается к серверу в этой же сети, но через внешний интерфейс маршрутизатора. Маршрутизатор привычно выполнит **DNAT** и отправит пакет серверу, который увидит, что адрес источник находится в одной сети с ним и пошлет обратный пакет напрямую, минуя маршрутизатор.

Но клиент устанавливал соединение с узлом **203.0.113.15**, а получает пакет от узла **192.168.100.10**, такой пакет получит статус **INVALID** и будет отброшен. Понятно, что ничего работать не будет.

Как избежать этой ситуации? Нужно сделать так, чтобы обратный пакет не отправлялся напрямую клиенту, а был возвращен маршрутизатору, для этого снова используем **SNAT**. Обратите внимание, что к этому времени пакет уже пройдет цепочку PREROUTING и в критериях нам следует оперировать внутренним адресом и портом назначения.

Правило **DNAT** при этом не меняется:

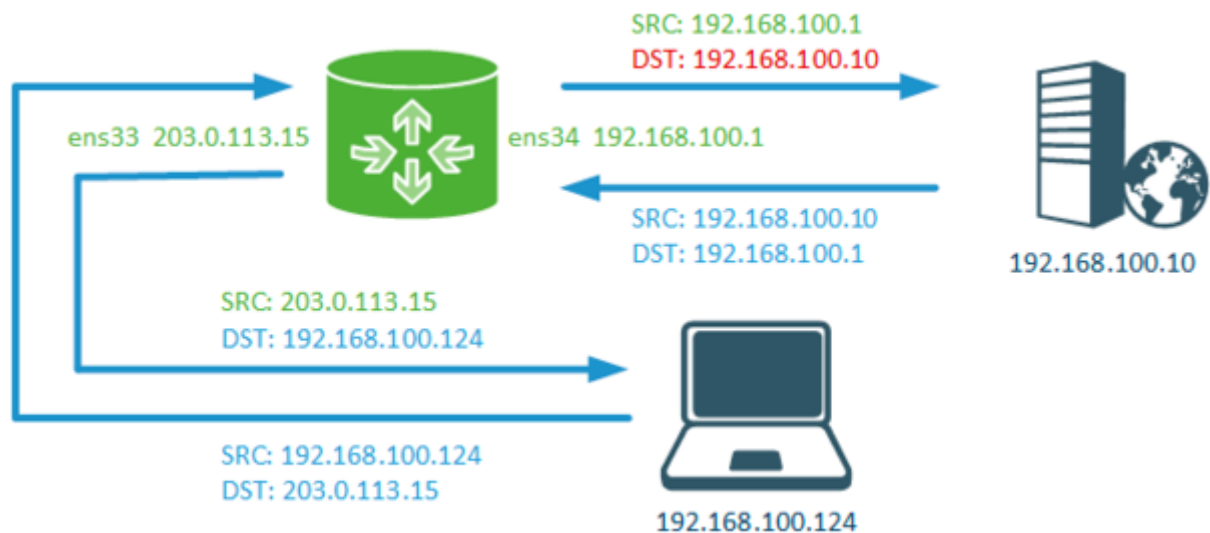
```
iptables -t nat -A PREROUTING -i ens35 -p tcp --dport 443 -j DNAT --to-destination 192.168.100.10
```

А вот в цепочку POSTROUTING мы добавляем следующее правило:

```
iptables -t nat -A POSTROUTING -s 192.168.100.0/24 -d 192.168.100.10 -p tcp --dport 443 -j SNAT --to-source 192.168.100.1
```

Что оно делает? Все пакеты, проходящие через маршрутизатор и имеющие в качестве адреса источника диапазон локальной сети и предназначенные серверу 192.168.100.10 в этой же сети, будут подвергнуты преобразованию адреса источника SNAT, который будет заменен на внутренний адрес маршрутизатора.





Для обратного пакета будет выполнена автоматическая замена на основании таблицы трансляции NAT, и он будет возвращен клиенту с внешнего адреса маршрутизатора, несмотря на то что клиент находится во внутренней сети.

### Онлайн-курс по устройству компьютерных сетей

На углубленном курсе "Архитектура современных компьютерных сетей" вы с нуля научитесь работать с Wireshark и «под микроскопом» изучите работу сетевых протоколов. На протяжении курса надо будет выполнить более пятидесяти лабораторных работ в Wireshark.



### Дополнительные материалы:

Помогла статья? Поддержи автора и новые статьи будут выходить чаще: