Основы iptables для начинающих. Часть 3. Таблица nat

Записки IT специалиста

Технический блог специалистов ООО"Интерфейс"

- Главная
- Основы iptables для начинающих. Часть 3. Таблица nat

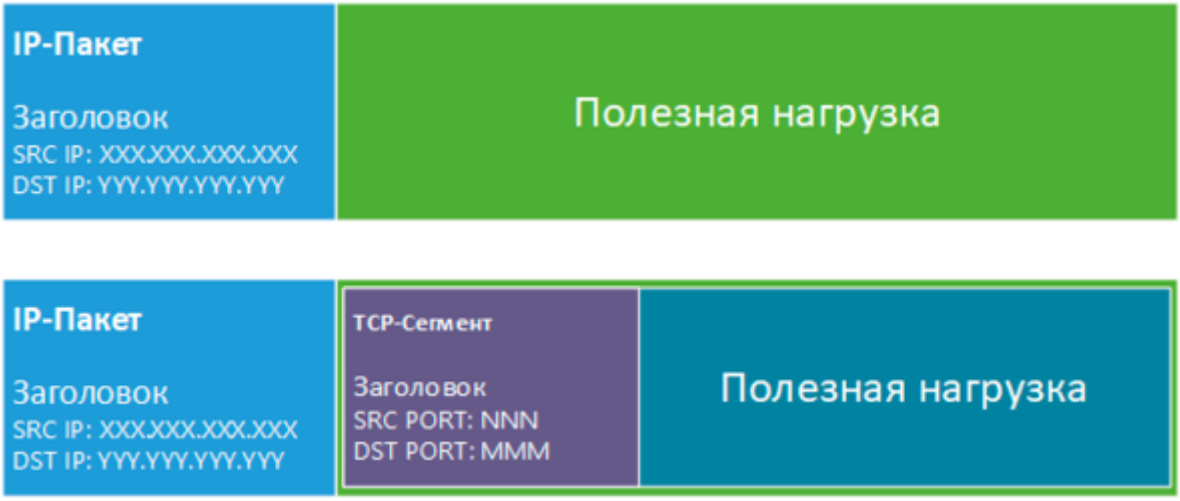
В качестве продолжения нашего цикла о брандмауэре **iptables** мы рассмотрим таблицу **nat**, в которой происходит преобразование сетевых адресов. С этой таблицей, также, как и с **filter** вы будете часто встречаться, выполняя такие привычные и повседневные задачи, как выход в интернет, проброс портов или перенаправление трафика. При этом действие механизма NAT более сложное и требует более глубокого понимания происходящих процессов, поэтому советуем уделить внимание в первую очередь теоретической части.

NAT (*Network Address Translation, преобразование сетевых адресов*) - это механизм работающий в IP-сетях и позволяющий преобразовывать IP-адреса транзитных пакетов. **Протокол IP** - маршрутизируемый протокол сетевого уровня (L3), предназначенный для передачи данных между сетями. Единицей передачи данных на уровне IP-протокола является **пакет**.

Сразу с этого места сделаем небольшое отступление, потому как с понятием "пакет" существует изрядная путаница, которая может приводить начинающих в некоторое замешательство. Начнем с того, что понятие **пакет** применимо только к **протоколу IP** сетевого уровня (L3). Пакет состоит из заголовка и полезной нагрузки. Заголовок, вместе с иной служебной информацией, содержит два важных поля: **адрес отправителя** (SRC IP) и **адрес получателя** (DST IP). Полезной нагрузкой IP-пакета является один из протоколов **транспортного уровня** (L4): TCP, UDP, GRE и т.д.

Единицей передачи данных в TCP является сегмент, в UDP - дейтаграмма, также оба этих протокола используют **порты**, порт - это логическая сущность, определяемая целым неотрицательным числом, которая позволяет создавать и поддерживать на одном узле несколько независимых соединений. Часть портов

зарезервирована за типовыми службами, так каждый знает, что TCP 80 - это HTTP, а UDP 53 - DNS. Т.е. номер порта является дополнительным идентификатором, позволяющим отделять одни данные от других и направлять их нужным службам. TCP-сегмент или UDP-дейтаграмма также содержат заголовок и полезную нагрузку, заголовок также содержит два важных поля: **порт источник** (SRC PORT) и **порт назначения** (DST PORT).



Следует четко запомнить: IP-заголовок содержит только адрес отправителя и адрес получателя, понятие порт в протоколе IP отсутствует. Порт появляется в заголовках транспортного протокола L4 (TCP, UDP), но информация об адресах источника и назначения там отсутствует. Но как быть, если где-то говорят о пакете на порт TCP 80? Это выражение следует понимать, как IP-пакет с полезной нагрузкой в виде TCP-сегмента с портом назначения 80. Также следует помнить, что у каждого транспортного протокола свой набор портов и одни и те же порты могут быть использованы одновременно с разными протоколами.

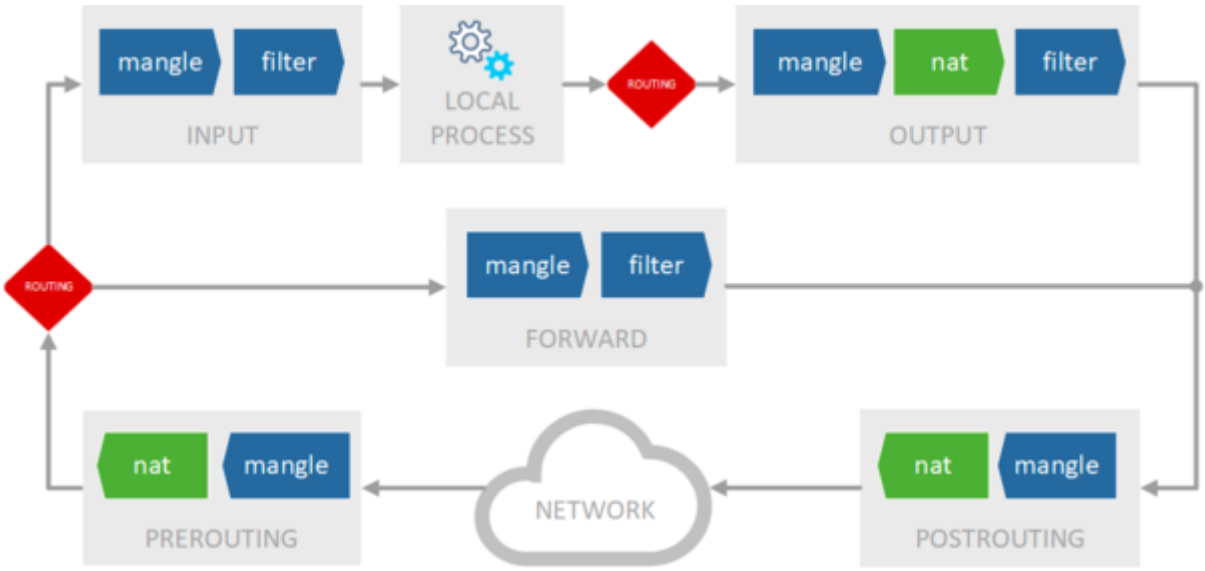
Теперь перейдем к самому понятию NAT, эта технология возникла тогда, когда потребовалось предоставить доступ к внешней сети узлам сети внутренней, не имеющим выделенного IP-адреса. По мере роста сетей при ограниченном количестве "белых" IP-адресов это становилось все большей проблемой и привело к совершенствованию механизма NAT.

Существует несколько видов NAT:

- **Симметричный** - когда все порты внутреннего адреса транслируются на порты внешнего адреса, при этом устройство становится полностью доступным из внешней сети.
- **Динамический** - когда порт внутреннего адреса случайным образом транслируется на порт одного из внешних адресов, причем для каждого нового соединения может быть использован отличающийся адрес
- **Перегруженный** - когда порты нескольких внутренних адресов транслируются на случайные порты единственного внешнего адреса.

Далее мы будем говорить именно о **перегруженном NAT** (*NAPT, NAT Overload, PAT, маскарадинг*), потому что в большинстве реализаций используется именно он, также следует отметить, что **PAT** (*Port Address Translation, Трансляция порт-адрес*) требует обязательного наличия открытого порта, что может вызвать проблемы с прохождением NAT для транспортных протоколов не использующих порты, например, GRE.

Вернемся к **iptables**, функции NAT реализованы в одноименной таблице **nat** (все таблицы обозначаются строчными буквами), которая содержит цепочки **PREROUTING, POSTROUTING** и **OUTPUT**. В первые две цепочки попадает весь трафик узла, в последнюю только собственный исходящий.



Обратите внимание, что все преобразования NAT выполняются либо до, либо после принятия решения о маршрутизации и фильтрации трафика брандмауэром, этот момент следует учитывать при построении правил фильтрации.

В таблице **nat** доступно два основных действия:

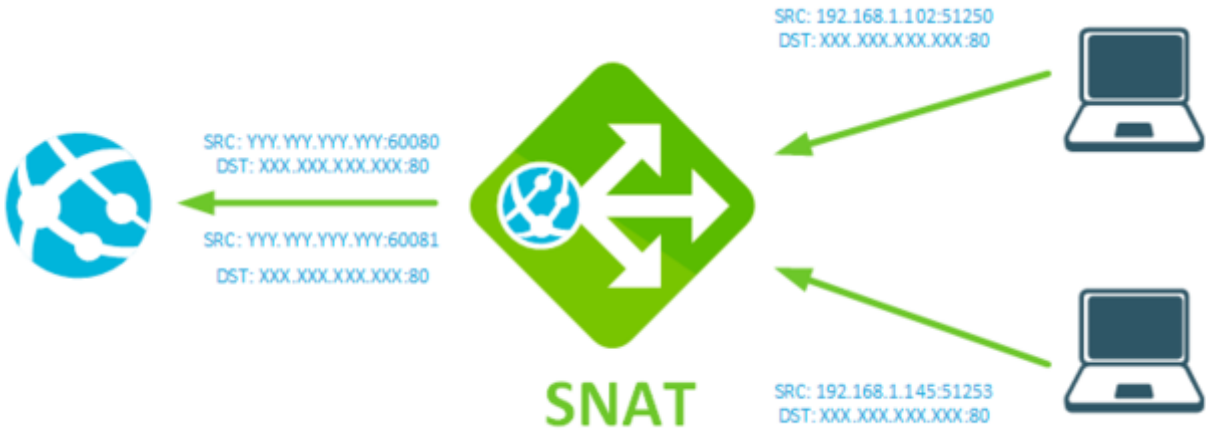
- **SNAT** - *Source Network Address Translation* - изменение адреса и порта источника пакета, доступен в цепочке POSTROUTING
- **DNAT** - *Destination Network Address Translation* - изменение адреса и порта назначения пакета, доступен в цепочках PREROUTING и OUTPUT

Таким образом изменить адрес и порт назначения мы можем только перед входом пакета в брандмауэр и до принятия решения о маршрутизации, а изменить адрес и порт источника можем только при выходе пакета из брандмауэра, после принятия решения о маршрутизации. Это еще одно важное правило, которое следует твердо запомнить: DNAT - на входе, SNAT - на выходе.

Более подробное знакомство мы начнем со **SNAT**, как мы уже говорили выше данное действие изменяет адрес и порт источника пакета, одно из основных практический применений - обеспечения доступа локальной сети в интернет. Давайте посмотрим, в чем состоит проблема и как SNAT помогает ее решить. Для

использования в локальных сетях используются диапазоны частных или т.н. "серых" IP-адресов, их особенностью является то, что они не маршрутизируются в интернете, т.е. если маршрутизатору попадает такой пакет и для него нет отдельного маршрута - он будет отброшен. При этом внешний адрес у данной сети один и его как-то следует использовать для обращения к внешнему миру всех локальных ПК.

Что делает SNAT? Он заменяет адрес источника пакета внешним адресом, также, при необходимости, меняет и порт источника, что позволяет различать запросы, сделанные с одного и того же порта разных ПК, запись о выполненной трансляции заносится в специальную **таблицу трансляций** (она же таблица NAT). Получив ответ роутер находит запись в таблице трансляций и на ее основании изменяет адрес и порт назначения ответного пакета, после чего он прозрачно доставляется адресату в локальной сети.



В общем виде запись в iptables для SNAT будет выглядеть следующим образом:

```
iptables -t nat -A POSTROUTING -o ens33 -j SNAT --to-source 198.51.100.1
```

Первое, на что следует обратить внимание - это ключ **-t**, который указывает используемую таблицу, если данный ключ не указан, то по умолчанию используется таблица **filter**. Запись добавляется в цепочку POSTROUTING и предписывает выполнить действие SNAT для всех пакетов, попадающих под критерий **-o ens33**, т.е. использующих в качестве исходящего внешний интерфейс **ens33**. Для SNAT обязательно указывается параметр **--to-source**, который используется для указания адреса, присваиваемому пакету, теперь именно этот адрес будет указываться в качестве исходящего.

Приведенное выше правило будет изменять адрес источника для всех исходящих пакетов без разбору, что не всегда корректно и безопасно, поэтому правилом хорошего тона считается явно задавать диапазон адресов для преобразования:

```
iptables -t nat -A POSTROUTING -o ens33 -s 192.168.100.0/24 -j SNAT --to-source
198.51.100.1
```

Также мы можем выполнять трансляцию адресов не для всего трафика, а выборочно, указывая необходимые порты и протоколы, скажем только для SSH:

```
iptables -t nat -A POSTROUTING -o ens33 -s 192.168.100.0/24 -p tcp --dport22 -j
SNAT --to-source 198.51.100.1
```

Частным случаем SNAT является действие **MASQUERADE** (маскарадинг), его основным отличием является то, маскарадинг самостоятельно получает IP-адрес от заданного сетевого интерфейса и не требует его явного указания. Это удобно если на внешнем интерфейсе используется динамический IP-адрес. Вторая особенность MASQUERADE в том, что при остановке интерфейса таблица трансляций полностью очищается и все текущие соединения разрываются. Причина такого поведения в том, что при следующем запуске интерфейса имеется возможность получить новый IP-адрес и все текущие записи сразу окажутся неверны.

Можно ли использовать MASQUERADE со статическим внешним адресом? Можно, но следует учитывать, что данное действие дает более высокую нагрузку на систему, так как определяет адрес внешнего интерфейса для каждого пакета, и в данном случае предпочтительно использовать SNAT.

Маскарадинг, также, как и SNAT, можно использовать только в цепочке POSTROUTING, и типичная запись будет выглядеть так:

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

Оба этих действия также позволяют явно указывать исходящий порт или диапазон портов для трансляции, для SNAT это делается в параметре --to-source через двоеточие, а в MASQUERADE используется специальная опция **--to-ports**:

```
iptables -t nat -A POSTROUTING -o ens33 -s 192.168.100.0/24 -p tcp --dport22 -j
SNAT --to-source 198.51.100.1:2222
iptables -t nat -A POSTROUTING -o ens33 -s 192.168.100.0/24 -p tcp --dport22 -j
MASQUERADE --to-ports 2222
```

Теперь вернемся к порядку прохождения пакетов по таблицам и цепочкам. Как несложно заметить - **цепочка POSTROUTING таблицы nat** является завершающим этапом обработки трафика, после которого он покидает брандмауэр. Это говорит о том, что в правилах фильтрации следует использовать **только исходные** адреса и порты источников, до трансляции. Новые адреса в брандмауэр не попадают.

Второе основное действие в таблице **nat** - это **DNAT** - или изменения адреса и порта назначения пакета. Это действие применимо в цепочках **PREROUTING** и **OUTPUT**, т.е. на самом входе в брандмауэр, раньше таблицы **nat** пакеты попадают только в **mangle**. На практике DNAT применяется в основном для проброса портов, типовой пример:

```
iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 3389 -j DNAT --to-
destination 192.168.100.1
```

Данное правило изменит адрес назначения всех пакетов пришедших на внешний интерфейс **ens33** по протоколу **tcp** с портом назначения TCP-сегмента **3389** на адрес внутреннего сервера **192.168.100.1**. Изменение номера порта при этом не производится. Если же вы используете нестандартный порт, то правило будет выглядеть несколько иначе:

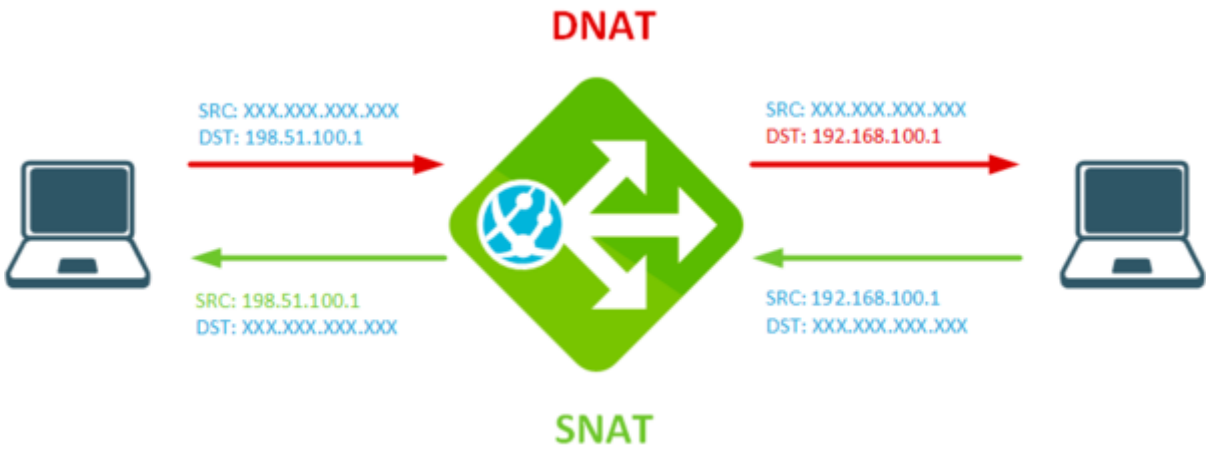
```
iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 3390 -j DNAT --to-
destination 192.168.100.1:3389
```

При выполнении проброса портов обязательным критерием должен быть указан входящий интерфейс (как в правиле выше) или адрес этого интерфейса, в противном случае преобразованию будут подвергаться все пакеты, в том числе и проходящие из внутренней сети наружу. При использовании адресов вместо интерфейсов правило будет иметь следующий вид:

```
iptables -t nat -A PREROUTING -d 198.51.100.1 -p tcp --dport 3389 -j DNAT --to-
destination 192.168.100.1
```

А теперь давайте попробуем разобраться как оно вообще работает. Некий узел **XXX.XXX.XXX.XXX** отправляет пакет на внешний адрес нашего роутера **198.51.100.1**, адресом источника будет являться адрес узла, адресом назначения - внешний адрес роутера. Получив такой пакет брандмауэр находит нужное правило и изменяет адрес назначения пакета на внутренний адрес целевого узла, т.е. на **192.168.100.1** и отправляет его в локальную сеть.

Узел назначения видит, что ему пришел пакет от удаленного узла **XXX.XXX.XXX.XXX** и он формирует ответный пакет с этим адресом в качестве назначения, а в качестве источника подставляет собственный адрес. Так как удаленный узел не принадлежит локальной сети данный пакет будет направлен основному шлюзу, т.е. назад нашему роутеру. А так как на шлюзе в любом случае включен либо **SNAT**, либо **MASQUERADE**, то локальный адрес источника будет заменен адресом внешнего интерфейса и такой пакет будет отправлен назад удаленному узлу.



Наблюдательный читатель сделает здесь два вывода. Первый - DNAT не использует таблицу трансляций, второй - уже имеющийся на роутере **SNAT** или **MASQUERADE** автоматически выполняет обратное преобразование. Таким образом удаленный узел общается только с внешним адресом роутера и не имеет ни малейшего представления о сети за ним, а локальный узел продолжает считать, что работает с удаленным узлом напрямую, все необходимые трансляции NAT делает прозрачно и на лету.

Частным случаем **DNAT** является **REDIRECT**, это действие позволяет перенаправлять пакеты на другой порт текущего узла. Это удобно при прозрачном проксировании, либо перенаправлении трафика. Например, перехватим все DNS-запросы и направим их локальному DNS-серверу, развернутому на этом же узле:

```
iptables -t nat -A PREROUTING -i ens34 -p udp --dport 53 -j REDIRECT --to-ports 53
```

А следующее правило выполнит прозрачное проксирование HTTP-трафика, не предназначенного узлу на развернутый на нем прокси-сервер:

```
iptables -t nat -A PREROUTING -i ens34 ! -d 192.168.100.2 -p tcp --dport 80 -j
REDIRECT --to-ports 3128
```

Где **192.168.100.2** - внутренний адрес нашего роутера, остальные критерии мы сознательно не будем комментировать, если вы внимательно изучили предыдущие статьи цикла, то должны без труда прочитать условие.

Мы же перейдем к не менее интересной теме последующего движения пакетов. Давайте еще раз внимательно изучим схему выше. Действие DNAT в цепочке PREROUTING выполняется до решения о маршрутизации, а, следовательно, пакеты могут претерпевать "чудесные превращения". Непонимание этого момента приводит ко многим грубым ошибкам в настройке брандмауэра и неработоспособным конфигурациям.

Допустим у нас с внешнего адреса 198.51.100.1 и порта 3390 сделан проброс на внутренний адрес 192.168.100.1 и порт 3389, но нормально закрытый брандмауэр блокирует подключения. Что нам нужно сделать? Правильно, открыть порт. Если долго не думать, то вырисовывается такой набор правил:

```
iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 3390 -j DNAT --to-
destination 192.168.100.1:3389
iptables  -A INPUT -i ens33 -p tcp --dport 3390 -j ACCEPT
```

На первый взгляд все правильно: мы "открыли" порт 3390 для внешних подключений, а затем перенаправили соединения с него на внутренний узел. Однако это грубая ошибка. Почему? Давайте разбираться.

И снова схема движения пакетов! Так, что мы видим? А видим мы то, что пакет первым делом попадает в цепочку **PREROUTING** таблицы **nat** и именно здесь у него будут изменены адрес и порт назначения. После чего пакет из локального,

предназначенного данному узлу, превратится в транзитный и вместо цепочки **INPUT** таблицы **filter** пойдет в цепочку **FORWARD** и приведенное выше правило никогда не сработает.

А как будет правильно?

```
iptables -t nat -A PREROUTING -i ens33 -p tcp --dport 3390 -j DNAT --to-
destination 192.168.100.1:3389
iptables -A FORWARD -i ens33 -d 192.168.100.1 -p tcp --dport 3389 -j ACCEPT
```

Обратите внимание, что вместе с локальным адресом в качестве критерия мы используем именно локальный порт - 3389, а не проброшенный 3390, так как DNAT изменит как адрес назначения, так и порт.

Аналогично, действие **REDIRECT** превращает транзитные пакеты в локальные и при возникновении такой необходимости фильтровать мы их должны уже не в цепочке **FORWARD**, а в **INPUT**.

Коротко отдельно коснемся цепочки **OUTPUT**, вопреки распространенному мнению, это вход в брандмауэр, а не выход. Сюда поступают исходящие пакеты от локальных служб узла и последовательно проходят таблицы **mangle - nat -filter**. Поэтому если вы фильтруете исходящий трафик после преобразования сетевых адресов, то следует учитывать уже новые адреса и порты назначения. Единственное, что не меняется - это статус пакета, он остается локальным и не при каких обстоятельствах не попадет в цепочку **FORWARD**, даже если после всех преобразований сетевых адресов он попадает под критерии транзитного.

В данной статье мы рассмотрели только базовые возможности таблицы **nat** в самых простых конфигурациях, но именно прочное усвоение базовых принципов позволяет успешно изучать более сложные схемы, которых мы обязательно коснемся в наших будущих публикациях.