Настройка iptables в Linux

Что такое iptables

iptables — это утилита командной строки, используемая для управления встроенным брандмауэром netfilter, доступным в ядре Linux, начиная с версии 2.4. Брандмауэр — это приложение, на котором происходит фильтрация сетевого трафика на основе заданных администратором правил. Обеспечить безопасность сервера или инфраструктуры, означает обеспечить отказоустойчивость и стабильность работы ваших серверов и приложений, что крайне чувствительно для бизнеса или персональных проектов.

В глобальной сети огромное количество угроз — боты, периодически прощупывают стандартные точки входа в системы, хулиганы, любопытные, взломщики — люди, целенаправленно пытающиеся получить несанкционированный доступ к информационным системам. Задача iptables — исключить либо хотя бы минимизировать негативное воздействие со стороны разного рода правонарушителей.

Установка iptables

В образах CentOS 8 64-bit и CentOS 8 Stream 64-bit *iptables* отсутствует, поскольку разработчики отказались от него в пользу более нового пакета — *nftables*. Его поддержка на уровне ядра доступна с версии 3.13. Если существует необходимость использовать именно *iptables*, требуется выполнить следующий порядок действий:

```
yum install iptables-services
```

Включение сервиса в автозагрузку:

```
systemctl enable iptables
```

Запуск сервиса:

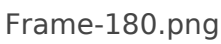
```
systemctl start iptables
```

Межсетевой экран готов к использованию.

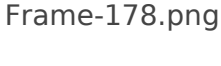
Порядок прохождения таблиц и цепочек

Любой поступивший пакет на сервер с iptables проходит через ядро, а именно — межсетевой экран netfilter. Каждый из них классифицируется в зависимости от его назначения, попадает в соответствующую ему таблицу и проходит по цепочкам, содержащим правила, установленные администратором.

На основе этих правил выполняется действие: принять пакет, отбросить, удалить или передать следующему узлу сети. Иллюстрация, представленная ниже, наглядно показывает путь прохождения пакета по системе:



Данный рисунок не отражает истинную архитектуру брандмауэра, а показывает только логику работы. Существует много распространенных заблуждений по поводу уровней вложенности таблиц, цепочек, и правил. Самым верхним уровнем представления являются таблицы, которые содержат набор свойственных им цепочек. Цепочки содержат списки правил. Схематично «матрешка» выглядит следующим образом:



Синтаксис iptables

Сетевой экран iptables очень гибок в настройке и имеет огромное количество разнообразных ключей и опций. Общий вид управляющей команды:

```
iptables таблица команда цепочка критерии действие
```

Рассмотрим каждый элемент в отдельности.

Пакет

Под пакетом понимают структурированный блок данных, содержащий в себе как пользовательскую информацию, называемую ещё полезной нагрузкой, так и служебную информацию, например об адресе отправителя, получателя, времени жизни пакета и многое

другое.

Цепочки

Существует 5 видов цепочек:

- **PREROUTING** — предназначена для первичной обработки входящих пакетов, адресованных как непосредственно серверу, так и другим узлам сети. Сюда попадает абсолютно весь входящий трафик для дальнейшего анализа.
- **INPUT** — для входящих пакетов, отправленных непосредственно этому серверу.
- **FORWARD** — для проходящих пакетов, не адресованных этому компьютеру, предназначены для передачи следующему узлу, в случае, если сервер выполняет роль маршрутизатора.
- **OUTPUT** — для пакетов, отправленных с этого сервера.
- **POSTROUTING** — здесь оказываются пакеты, предназначенные для передачи на другие узлы сети.

Также есть возможность создавать и удалять собственные цепочки, в большинстве случаев, в этом нет необходимости. Названия цепочек пишут заглавными буквами.

Таблицы

В netfilter существуют 5 типов таблиц, каждая из них имеет свое назначение.

Таблица raw

Содержит цепочки **PREROUTING** и **OUTPUT**, здесь производятся манипуляции с пакетами до задействования механизма определения состояний.

Таблица mangle

Предназначена для модификации заголовков сетевых пакетов, таких параметров как **ToS** (Type of Service), **TTL** (Time To Live), **MARK**. Содержит все существующие пять цепочек.

Таблица nat

Используется для трансляции сетевых адресов, т.е. подмены адреса получателя/отправителя, применяется, если сервер используется в качестве маршрутизатора. Содержит цепочки **PREROUTING**, **OUTPUT**, **POSTROUTING**.

Таблица filter

Основная таблица, служит для фильтрации пакетов, именно здесь происходит принятие решений о разрешении или запрете дальнейшего движения пакета в системе. Используется по умолчанию, если явно не указано имя другой таблицы. Содержит цепочки **INPUT**, **FORWARD** и **OUTPUT**.

Таблица security

Используется для взаимодействия с внешними системами безопасности, в частности с SELinux и AppArmor. Содержит цепочки **INPUT**, **OUTPUT** и **FORWARD**.

Имена таблиц принято писать строчными буквами.

Действия

Правилами задается поведение для iptables, каким образом поступить с тем или иным пакетом при попадании под заданные критерии. Решения, которые принимает брандмауэр, называют действиями, самые распространенные из них:

- **ACCEPT** — разрешить дальнейшее прохождение пакета по системе;
- **DROP** — выбросить пакет без уведомления отправителя;
- **REJECT** — отказать в прохождении пакета с уведомлением отправителя, такой способ может привести к дополнительным затратам ресурсов процессора, поэтому, в большинстве случаев рекомендуется использовать DROP;
- **LOG** — зафиксировать информацию о пакете в файле системного журнала;
- **MARK** — позволяет помечать определенные пакеты, например для маршрутизации, данная метка перестает существовать, как только пакет покинет брандмауэр;
- **CONNMARK** — то же самое, что и MARK, только для соединений;
- **QUEUE** — отправляет пакет в очередь приложению для дальнейшего взаимодействия;
- **RETURN** — прекращение движения пакета по текущей цепочке и возврат в предыдущую цепочку. Если текущая цепочка единственная — к пакету будет применено действие по умолчанию;
- **REDIRECT** — перенаправляет пакет на указанный порт, в пределах этого же узла, применяется для реализации «прозрачного» прокси;
- **DNAT** — подменяет адрес получателя в заголовке IP-пакета, основное применение — предоставление доступа к сервисам снаружи, находящимся внутри сети;
- **SNAT** — служит для преобразования сетевых адресов, применимо, когда за сервером находятся машины, которым необходимо предоставить доступ в Интернет, при этом от провайдера имеется статический IP-адрес;
- **MASQUERADE** — то же, что и SNAT, но главное отличие в том, что может использоваться, когда провайдер предоставляет динамический адрес, создаёт дополнительную нагрузку на систему по сравнению с SNAT;
- **TOS** — позволяет управлять битами в одноименном поле заголовка IP-пакета;
- **ULOG** — более продвинутый вариант записи информации, может писать как в обычный текстовый файл, так и в базу данных;
- **TTL** — используется для изменения значения поля одноименного заголовка IP-пакета, устанавливает время жизни пакета.

Команды

Для iptables команда — это инструкция к действию, при помощи них можно добавлять, удалять и сбрасывать все правила, задавать действия по умолчанию и многое другое. Команды могут подаваться как в сокращенном, так и в полном виде, более подробно в таблице:

d9fc616ce5d293bcde5d6.png

b20ad97ff7f5eeeeb0153.png

Критерии

Чтобы к пакету применить какое-либо действие, он должен попасть под определенные критерии. Одно правило может содержать несколько критериев. Они, как и команды, имеют полную и сокращенную форму. Некоторые из них поддерживают логическую **НЕ**, если перед ними поставить знак **!** — критерий будет инвертирован. Список в таблице ниже:

69a83e856dac60533d7dd.png

69797e7d64163f45d474f.png

Состояние соединений

Система отслеживания состояния соединений conntrack — важная часть сетевого стека linux, встроенная в ядро. Используется для сопоставления пакетов с конкретными соединениями. Под анализ попадают все пакеты, кроме помеченных **NOTRACK**, в таблице **raw**. Все пакеты классифицируются на:

- **NEW** — открывается новое соединение, пришел только первый пакет;
- **ESTABLISHED** — соединение установлено, пришел уже не первый пакет в рамках этого сеанса. При правильной настройке iptables — такие пакеты проходят по системе без фильтрации, поскольку она уже была выполнена для первого пакета соединения;
- **RELATED** — открывается новое соединение, связанное с другим сеансом, имеющим статус ESTABLISHED;
- **INVALID** — помечаются пакеты, которые не связаны ни с одним из существующих соединений, и не могут создать новое, их невозможно идентифицировать. В целях безопасности рекомендуется остановить движение таких пакетов по системе, используя действие DROP.

Основные команды iptables

Как посмотреть список правил iptables

```
iptables --line-numbers -L -v -n
```

Ключ *—line-numbers* нумерует строки, *-L* выводит список правил всех цепочек, *-v* отвечает за детализацию вывода, *-n* выводит IP-адреса и номера портов в числовом формате.

Как удалить правило в iptables:

В первую очередь необходимо определить номер правила, которое требуется удалить, выводим список действующих правил командой:

```
iptables --line-numbers -L -v -n
```

Предположим, требуется удалить правило 4 в цепочке **INPUT**:

```
iptables -D INPUT 4
```

Как сохранить правила iptables

Утилита iptables, как и маршрутизаторы Cisco, не сохраняет правила, если это явно не указать и после перезагрузки возвращается в предыдущее состояние. Установим пакет:

```
apt install iptables-persistent
```

В процессе установки на оба вопроса ответить Yes. Сохранить текущие правила:

```
service netfilter-persistent save
```

Система при следующей загрузке использует последние сохраненные правила.

Как восстановить правила

В процессе настройки брандмауэра, по разным причинам, возникает необходимость вернуться к заведомо рабочим, испытанным правилам. Сервис *netfilter-persistent* сохраняет их в файле */etc/iptables/rules.v4*, если не успели сохранить активные правила, значит в файле предыдущая версия, восстанавливаем:

```
iptables-restore < /etc/iptables/rules.v4
```

Примеры настройки iptables

В данном разделе рассмотрим задачи, с которыми придется столкнуться, работая с iptables.

Как заблокировать IP-адрес в iptables

Допустим, необходимо заблокировать компьютер с IP-адресом 172.10.10.1, тогда правило будет выглядеть следующим образом:

```
iptables -A INPUT -s 172.10.10.1 -j DROP
```

Как разрешить IP-адрес в iptables

Необходимо разрешить весь трафик к серверу для клиента с IP-адресом 192.168.111.1:

```
iptables -A INPUT -s 192.168.111.1 -j ACCEPT
```

Как открыть порт в iptables

Предположим, что политика по умолчанию — блокировать все, что явно не разрешено. Откроем порты веб-сервера для обеспечения работы HTTP протокола — порт 80, и поддержки HTTPS протокола совместно с SSL — порт 443. Также для доступа к серверу по SSH откроем порт 22. Эту задачу можно решить как минимум двумя способами: создать однострочное правило, либо прописать правила по каждому из портов, рассмотрим оба. В одну строку:

```
iptables -A INPUT -p tcp -m multiport --dports 22,80,443 -j ACCEPT
```

При использовании расширения multiport, всегда необходимо использовать критерий *-p tcp* или *-p udp*, таким образом, одной строкой можно указать до 15 разных портов через запятую. Важно не путать критерии *—-dport* и *—-dports*. Первый из них используется для указания одного порта, второй сразу для нескольких, аналогично с *—sport* и *—sports*.

Вариант многострочной записи — для каждого порта свое правило:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

Revision #1
Created 2023-10-24 18:12:31 UTC by odiljonov
Updated 2023-10-24 18:37:20 UTC by odiljonov