Лабораторная работа № 10

**Тема:** Iptables.

**Цель:** изучить варианты использования и способы настройки межсетевого экрана Linux с использованием утилиты iptables.

# Теоретические сведения

Iptables — утилита командной строки, которая является стандартным интерфейсом управления работой межсетевого экрана (брандмауэра) ядра Linux. Иногда под словом iptables имеется в виду и сам межсетевой экран.

Все сетевые пакеты, которые проходят через компьютер, отправляются компьютером или предназначены компьютеру, ядро Linux направляет через фильтры iptables. Там эти пакеты поддаются проверкам и затем для каждой проверки, если она пройдена выполняется указанное в ней действие. Например, пакет передается дальше ядру для отправки целевой программе, или отбрасывается.

Все пакеты делятся на три типа: входящие, исходящие и проходящие. Входящие — это те, которые были отправлены на этот компьютер, исходящие — отправленные из этого компьютера в сеть. А проходящие (или транзитные) — это пакеты, которые должны быть пересланы дальше, например, если компьютер выступает в качестве маршрутизатора.

Существует пять стандартных цепочек, встроенных в систему сетевого экрана:

* INPUT — обрабатывает входящие пакеты и подключения.
* FORWARD — применяется для проходящих соединений. Сюда попадают пакеты, которые отправлены на компьютер, но не предназначены ему.
* OUTPUT — используется для исходящих пакетов и соединений.

Кроме перечисленных выше, есть еще две дополнительные цепочки правил:

* PREROUTING — в эту цепочку пакет попадает перед обработкой iptables, система еще не знает куда он будет отправлен, в INPUT, OUTPUT или FORWARD;
* POSTROUTING — сюда попадают все проходящие пакеты, которые уже прошли цепочку FORWARD.

Названия цепочек всегда пишутся заглавными буквами.

Для каждого типа пакетов можно установить набор правил, которые по очереди будут проверяться на соответствие с пакетом и если пакет соответствует, то применять к нему указанное в правиле действие. Правила образуют цепочку, поэтому input, output и forward называют цепочками правил. Действий может быть несколько:

* ACCEPT — разрешить прохождение пакета дальше по цепочке правил;
* DROP — удалить пакет;
* REJECT — отклонить пакет, отправителю будет отправлено сообщение, что пакет был отклонен;
* LOG — сделать запись о пакете в лог файл;
* QUEUE — отправить пакет пользовательскому приложению.

Правила могут проверять любые соответствия, например, по ip, по порту получателя или отправителя, заголовкам пакета и многому другому. Если пакет не подходит ни одному из правил, то к нему применяется действие по умолчанию, обычно ACCEPT.

Цепочки организованны в 4 таблицы:

* raw — просматривается до передачи пакета системе определения состояний. Используется редко, например, для маркировки пакетов, которые не должны обрабатываться системой определения состояний. Для этого в правиле указывается действие NOTRACK. Содержит цепочки PREROUTING и OUTPUT.
* mangle — содержит правила модификации (обычно заголовка) IP‐пакетов. Среди прочего, поддерживает действия TTL (Time to live), TOS (Type of Service), и MARK (для изменения полей TTL и TOS, и для изменения маркеров пакета). Редко необходима и может быть опасна. Содержит все пять стандартных цепочек.
* nat — просматривает только пакеты, создающие новое соединение (согласно системе определения состояний). Поддерживает действия DNAT, SNAT, MASQUERADE, REDIRECT. Содержит цепочки PREROUTING, OUTPUT, и POSTROUTING.
* filter — основная таблица, используется по умолчанию если название таблицы не указано. Содержит цепочки INPUT, FORWARD, и OUTPUT.

Последовательность прохождения пакета по таблицам и цепочкам показана на рисунке 1.

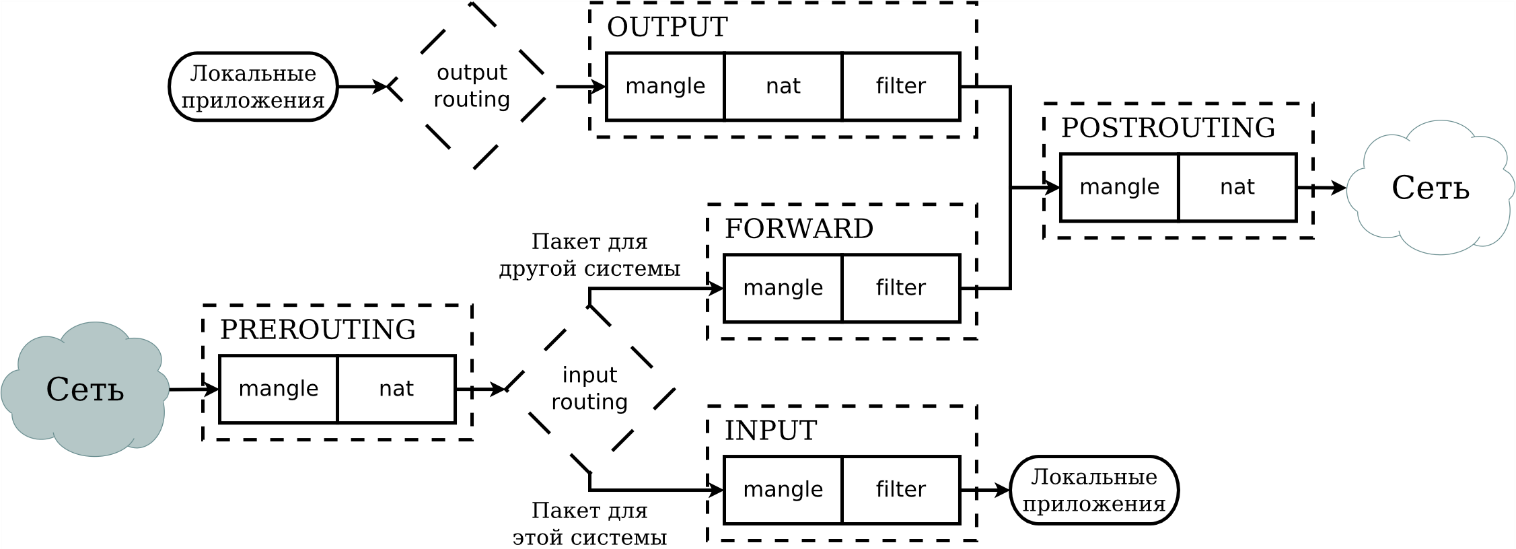


Рисунок 1 — Упрощённая диаграмма прохождения таблиц и цепочек.

Управление правилами, находящимися в цепочках и таблицах, осуществляется из терминала с использованием команд iptables, которые имеют следующую структуру:

iptables таблица операция цепочка критерии действие

Действие может быть одним из следующих:

* -A — добавить правило в цепочку;
* -С — проверить все правила;
* -D — удалить правило;
* -I — вставить правило с нужным номером;
* -L — вывести все правила в текущей цепочке;
* -S — вывести все правила;
* -F — очистить все правила;
* -N — создать цепочку;
* -X — удалить цепочку;
* -P — установить действие по умолчанию.

Дополнительные опции:

* -p — указать протокол, один из tcp, udp, icmp, icmpv6,esp, ah, sctp;
* -s — указать ip адрес устройства-отправителя пакета;
* -d — указать ip адрес получателя;
* -i — входной сетевой интерфейс;
* -o — исходящий сетевой интерфейс;
* -j — выбрать действие, если правило подошло.

Для разных таблиц существуют разные действия.

***Для таблицы raw:***

— NOTRACK — позволяет предотвратить обработку пакетов системой conntrack (трассировщика соединений). Разумеется, применять его стоит не ко всем пакетам подряд, а только к тем, для которых такая обработка не нужна и даже вредна, например, к пакетам, к которым впоследствии применяется действие TARPIT.

— CT — более функциональный инструмент, добавленный в версии Linux 2.6.34. Позволяет задать различные настройки conntrack, в соответствии с которыми будет обрабатываться соединение, открытое данным пакетом.

***Для таблицы mangle:***

— TOS — изменяет поле TOS (Type Of Service) данного пакета.

— TTL — изменяет поле TTL (Time To Live) данного пакета.

— MARK — устанавливает или изменяет маркировку пакета.

***Для таблицы filter:***

— ACCEPT — разрешить прохождение пакета дальше по цепочке правил;

— DROP — удалить пакет;

— REJECT — отклонить пакет, отправителю будет отправлено сообщение, что пакет был отклонен;

— LOG — сделать запись о пакете в лог файл;

— QUEUE — отправить пакет пользовательскому приложению.

***Для таблицы nat:***

— REDIRECT — выполняет перенаправление пакетов на другой порт того же самого компьютера.

— SNAT (Source Network Address Translation) — используется для трансляции сетевых адресов, то есть изменения исходящего IP адреса в заголовке IP пакета. Например, это действие можно использовать для предоставления выхода в интернет другим компьютерам из локальной сети, имея лишь один уникальный IP адрес. Для этого необходимо включить пересылку пакетов (forwarding) в ядре и затем создать правила, которые будут транслировать исходящие IP адреса локальной сети в реальный внешний адрес.

— DNAT (Destination Network Address Translation) — используется для преобразования адреса назначения в заголовке IP пакета. Если пакет подпадает под критерий правила, выполняющего DNAT, то этот пакет, и все последующие пакеты из этого же потока, будут подвергнуты преобразованию адреса назначения и переданы на требуемое устройство, хост или сеть. Данное действие реализует механизм, часто называемый перенаправлением портов, и может, к примеру, успешно использоваться для предоставления доступа к внутренним серверам, находящимся в локальной сети, и не имеющих реального IP адреса.

— MASQUERADE — во многом аналогичен SNAT, но не требует задания внешнего адреса, в который будут транслироваться внутренние адреса, а определяет его автоматически. Используется в случаях, когда IP адрес присваивается устройству динамически по протоколу DHCP. Такой способ работает дольше, чем SNAT и создаёт дополнительную нагрузку на вычислительные устройства компьютера, поэтому использовать его рекомендуется только при динамическом подключении. Если же IP адреса назначены статически, то более подходящим выбором будет использование действия SNAT.

Наличие трассировщика позволяет создавать более надежные наборы правил по сравнению с брандмауэрами, которые не имеют поддержки такого механизма.

В пределах iptables, соединение может иметь одно из 4-х базовых состояний: NEW, ESTABLISHED, RELATED и INVALID.

— Признак NEW сообщает о том, пакет открывает новый сеанс. Это означает, что это первый пакет в данном соединении, который увидел модуль трассировщика. Классический пример — пакет TCP с флагом SYN.

— Состояние ESTABLISHED означает, что пакет является частью уже существующего сеанса. Единственное требование, предъявляемое к соединению, заключается в том, что для перехода в состояние ESTABLISHED необходимо чтобы узел сети передал пакет и получил на него ответ от другого узла (хоста). После получения ответа состояние соединения NEW или будет изменено на ESTABLISHED.

— Статус RELATED получает соединение, если оно связано с другим соединением, имеющим признак ESTABLISHED. То есть соединение получает признак RELATED тогда, когда оно инициировано из уже установленного соединения. Например, во время сеанса пассивного FTP, клиент подсоединяется к порту 21 сервера, сервер сообщает клиенту номер второго, случайно выбранного порта, после чего клиент подсоединяется ко второму порту для передачи файлов. В этом случае второй сеанс (передача файлов по второму порту) связан с уже существующим сеансом (изначальное подсоединение к порту 21).

— Признак INVALID говорит о том, что пакет не может быть идентифицирован и поэтому не может иметь определенного статуса.

Для вывода всех существующих правил iptables используется команда:

iptables –L

а для получения более подробной информации:

iptables –L -v

Полная очистка всех правил выполняется командой:

sudo iptables –F

отдельной цепочки:

sudo iptables -F ЦЕПОЧКА

Если для пакета не подходит ни одно правило, то для него применяется действие по умолчанию. Его можно задать с помощью команды:

sudo iptables -p ЦЕПОЧКА ДЕЙСТВИЕ

Например, чтобы полностью запретить пересылку пакетов, необходимо написать:

sudo iptables -p FORWARD DROP

Ниже приведены несколько примеров команд для решения наиболее типичных задач по фильтрации трафика.

Блокировка ip-адреса:

iptables –A FORWARD –d 8.8.8.8 –j DROP

Блокировка диапазона ip-адресов (подсети):

iptables –A FORWARD –d 8.8.0.0/16 –j DROP

Блокировка протокола (ICMP):

iptables –A FORWARD –p ICMP –j DROP

# Задание

Настройте сетевые экраны на виртуальных машинах таким образом, чтобы они могли взаимодействовать только с помощью тех протоколов, которые были использованы в предыдущих лабораторных работах, а именно:

Сервер может принимать соединения только по

* telnet (порт 23),
* SSH (порт 22),
* FTP (порты 20 (исходящие соединения в активном режиме), 21 (входящие соединения)),
* SMB (порт 445);

и отвечает на пинги.

Клиент может устанавливать соединения с сервером по перечисленным выше протоколам и посылать пинги.

После создания всех правил их необходимо сохранить, иначе они исчезнут после перезагрузки операционной системы. Это можно сделать командами iptables-save, iptables-restore, однако, проще использовать утилиту iptables-persistent. Её необходимо установить, командой:

sudo apt install iptables-persistent

Однако, для установки потребуется скачать соответствующий пакет из репозиториев Ubuntu, а для доступа к ним нужно разрешить работу протокола DNS (UDP-порт 53). Запишите самостоятельно два правила:

— для разрешения уже установленных UDP-соединений,

— для разрешения соединений с портом назначения, используемым DNS.

# Контрольные вопросы

1. Что такое iptables?

2. Какие стандартные цепочки существуют в iptables? От чего зависит набор цепочек, пройденных конкретным пакетом?

3. Что такое порт?

4. Что такое состояние соединения? В каких протоколах есть состояния, в каких нет? От чего это зависит?