iptables: документация от iptables.ru

IPTables

Fix Me!

Источник

Порядок прохождения таблиц и цепочек

Когда пакет приходит на наш брандмауэр, то он сперва попадает на сетевое устройство, перехватывается соответствующим драйвером и далее передаётся в ядро. Далее пакет проходит ряд таблиц и затем передаётся либо локальному приложению, либо переправляется на другую машину. Порядок следования пакета приводится ниже:

Порядок движения транзитных пакетов

Шаг	Таблица	Цепочка	Примечание
1			Кабель (т.е. Интернет)
2			Сетевой интерфейс (например, eth0)
3	`mangle`	`PREROUTING`	Обычно эта цепочка используется для внесения изменений в заголовок пакета, например для изменения битов TOS и пр.
4	`nat`	`PREROUTING`	Эта цепочка используется для трансляции сетевых адресов (Destination Network Address Translation). Source Network Address Translation выполняется позднее, в другой цепочке. Любого рода фильтрация в этой цепочке может производиться только в исключительных случаях
5			Принятие решения о дальнейшей маршрутизации, т.е. в этой точке решается куда пойдёт пакет – локальному приложению или на другой узел сети.
6	`mangle`	`FORWARD`	Далее пакет попадает в цепочку FORWARD таблицы mangle, которая должна использоваться только в исключительных случаях, когда необходимо внести некоторые изменения в заголовок пакета между двумя точками принятия решения о маршрутизации.
7	`Filter`	`FORWARD`	В цепочку FORWARD попадают только те пакеты, которые идут на другой хост Вся фильтрация транзитного трафика должна выполняться здесь. Не забывайте, что через эту цепочку проходит трафик в обоих направлениях, обязательно учитывайте это обстоятельство при написании правил фильтрации.
8	`mangle`	`POSTROUTING`	Эта цепочка предназначена для внесения изменений в заголовок пакета уже после того как принято последнее решение о маршрутизации.
9	`nat`	`POSTROUTING`	Эта цепочка предназначена в первую очередь для Source Network Address Translation. Не используйте ее для фильтрации без особой на то необходимости. Здесь же выполняется и маскарадинг (Masquerading).
10			Выходной сетевой интерфейс (например, eth1).
11			Кабель (пусть будет LAN).

Как вы можете видеть, пакет проходит несколько этапов, прежде чем он будет передан далее. На каждом из них пакет может быть остановлен, будь то цепочка iptables или что либо ещё, но нас главным образом интересует iptables. Заметьте, что нет каких либо цепочек, специфичных для отдельных интерфейсов или чего либо подобного.

Цепочку `FORWARD` проходят ВСЕ пакеты, которые движутся через наш брандмауэр/ роутер.

Не используйте цепочку `INPUT` для фильтрации транзитных пакетов, они туда просто не попадают! Через эту цепочку движутся только те пакеты, которые предназначены данному хосту!

Установка политик по-умолчанию

Прежде, чем приступить к созданию набора правил, необходимо определиться с политиками цепочек по-умолчанию.

Политика по-умолчанию устанавливается командой:

```
iptables [-P {chain} {policy}]
```

Политика по-умолчанию представляет собой действие, которое применяется к пакету, не попавшему под действие ни одного из правил в цепочке. (Небольшое уточнение, команда `iptables -P` применима ТОЛЬКО К ВСТРОЕННЫМ цепочкам, т.е. `INPUT`, `FORWARD`, `OUTPUT` и т.п., и не применима к пользовательским цепочкам).

Нужно быть осторожным с установкой политик по-умолчанию для цепочек из таблиц, не предназначенных для фильтрации, так как это может приводить к довольно странным результатам.

Команды

Команда	Пример	Описание
`-A, –append`	`iptables -A INPUT ...`	Добавляет новое правило в конец заданной цепочки.
`-D, –delete`	`iptables -D INPUT --dport 80 -j DROP, iptables -D INPUT 1`	Удаление правила из цепочки. Команда имеет два формата записи, первый – когда задается критерий сравнения с опцией -D (см. первый пример), второй – порядковый номер правила. Если задается критерий сравнения, то удаляется правило, которое имеет в себе этот критерий, если задается номер правила, то будет удалено правило с заданным номером. Счет правил в цепочках начинается с 1.
`-R, –replace`	`iptables -R INPUT 1 -s 192.168.0.1 -j DROP`	Эта команда заменяет одно правило другим. В основном она используется во время отладки новых правил.
`-I, –insert`	`iptables -I INPUT 1 --dport 80 -j ACCEPT`	Вставляет новое правило в цепочку. Число, следующее за именем цепочки указывает номер правила, перед которым нужно вставить новое правило, другими словами число задает номер для вставляемого правила. В примере выше, указывается, что данное правило должно быть 1-м в цепочке INPUT.
`-L, –list`	`iptables -L INPUT`	Вывод списка правил в заданной цепочке, в данном примере предполагается вывод правил из цепочки INPUT. Если имя цепочки не указывается, то выводится список правил для всех цепочек. Формат вывода зависит от наличия дополнительных ключей в команде, например -n, -v, и пр.
`-F, –flush`	`iptables -F INPUT`	Сброс (удаление) всех правил из заданной цепочки (таблицы). Если имя цепочки и таблицы не указывается, то удаляются все правила, во всех цепочках. (Хочется от себя добавить, что если не указана таблица ключом -t (–table), то очистка цепочек производится только в таблице filter, прим. перев.)
`-Z, –zero`	`iptables -Z INPUT`	Обнуление всех счетчиков в заданной цепочке. Если имя цепочки не указывается, то подразумеваются все цепочки. При использовании ключа -v совместно с командой -L, на вывод будут поданы и состояния счетчиков пакетов, попавших под действие каждого правила. Допускается совместное использование команд -L и -Z. В этом случае будет выдан сначала список правил со счетчиками, а затем произойдет обнуление счетчиков.
`-N, –new-chain`	`iptables -N allowed`	Создается новая цепочка с заданным именем в заданной таблице В выше приведенном примере создается новая цепочка с именем allowed. Имя цепочки должно быть уникальным и не должно совпадать с зарезервированными именами цепочек и действий (такими как DROP, REJECT и т.п.)

Команда	Пример	Описание
-X, —delete-chain	`iptables -X allowed`	Удаление заданной цепочки из заданной таблицы. Удаляемая цепочка не должна иметь правил и не должно быть ссылок из других цепочек на удаляемую цепочку. Если имя цепочки не указано, то будут удалены все цепочки заданной таблице кроме встроенных.
-P, —policy	`iptables -P INPUT DROP`	Задает политику по-умолчанию для заданной цепочки. Политика по-умолчанию определяет действие, применяемое к пакетам не попавшим под действие ни одного из правил в цепочке. В качестве политики по умолчанию допускается использовать DROP и ACCEPT.
-E, —rename-chain	`iptables -E allowed disallowed`	Команда -E выполняет переименование пользовательской цепочки. В примере цепочка allowed будет переименована в цепочку disallowed. Эти переименования не изменяют порядок работы, а носят только косметический характер.

Команда должна быть указана всегда. Список доступных команд можно просмотреть с помощью команды iptables -h или, что тоже самое, iptables –help. Некоторые команды могут использоваться совместно с дополнительными ключами. Ниже приводится список дополнительных ключей и описывается результат их действия. При этом заметьте, что здесь не приводится дополнительных ключей, которые используются при построении критериев (matches) или действий (targets). Эти опции мы будем обсуждать далее.

Дополнительные ключи

Ключ	Команды, с которыми используется	Описание
-v, —verbose	—list, —append, —insert, —delete, —replace	Используется для повышения информативности вывода и, как правило, используется совместно с командой –list. В случае использования с командой –list, в вывод этой команды включаются так же имя интерфейса, счетчики пакетов и байт для каждого правила. Формат вывода счетчиков предполагает вывод кроме цифр числа еще и символьные множители K (x1000), M (x1,000,000) и G (x1,000,000,000). Для того, чтобы заставить команду –list выводить полное число (без употребления множителей) требуется применять ключ -x, который описан ниже. Если ключ -v, –verbose используется с командами –append, –insert, –delete или –replace, то будет выведен подробный отчет о произведенной операции.
-x, —exact	—list	Для всех чисел в выходных данных выводятся их точные значения без округления и без использования множителей K, M, G. Этот ключ используется только с командой –list и не применим с другими командами.
-n, —numeric	—list	Заставляет iptables выводить IP-адреса и номера портов в числовом виде предотвращая попытки преобразовать их в символические имена. Данный ключ используется только с командой –list.
—line-numbers	—list	Ключ –line-numbers включает режим вывода номеров строк при отображении списка правил командой –list. Номер строки соответствует позиции правила в цепочке. Этот ключ используется только с командой –list.
-c, —set-counters	—insert, —append, —replace	Этот ключ используется для установки начального значения счетчиков пакетов и байт в заданное значение при создании нового правила. Например, ключ –set-counters 20 4000 установит счетчик пакетов = 20, а счетчик байт = 4000.
—modprobe	Все	Ключ –modprobe определяет команду загрузки модуля ядра. Данный ключ может использоваться в случае, когда модули ядра находится вне пути поиска (search path). Этот ключ может использоваться с любой командой.

Критерии

Общие критерии

Общие критерии допустимо употреблять в любых правилах, они не зависят от типа протокола и не требуют подгрузки модулей расширения. К этой группе я умышленно отнес критерий –protocol несмотря на то, что он

используется в некоторых специфичных от протокола расширениях. Например, мы решили использовать TCP критерий, тогда нам необходимо будет использовать и критерий –protocol которому в качестве дополнительного ключа передается название протокола – TCP. Однако критерий –protocol сам по себе является критерием, который используется для указания типа протокола.

Критерий	Пример	Описание
-p, –protocol	`iptables -A INPUT -p tcp`	Этот критерий используется для указания типа протокола. Примерами протоколов могут быть TCP, UDP и ICMP. Список протоколов можно посмотреть в файле /etc/protocols. Прежде всего, в качестве имени протокола в данный критерий можно передавать один из трех вышеупомянутых протоколов, а также ключевое слово ALL. В качестве протокола допускается передавать число - номер протокола, так например, протоколу ICMP соответствует число 1, TCP – 6 и UDP – 17. Соответствия между номерами протоколов и их именами вы можете посмотреть в файле /etc/protocols, который уже упоминался. Критерию может передаваться и список протоколов, разделенных запятыми, например так: udp,tcp (Хотя автор и указывает на возможность передачи списка протоколов, тем не менее вам врят ли удастся это сделать! Кстати, man iptables явно оговаривает, что в данном критерии может быть указан только один протокол. Может быть это расширение имеется в patch-o-matic? прим. перев.) Если данному критерию передается числовое значение 0, то это эквивалентно использованию спецификатора ALL, который подразумевается по умолчанию, когда критерий –protocol не используется. Для логической инверсии критерия, перед именем протокола (списком протоколов) используется символ !, например –protocol ! tcp подразумевает пакеты протоколов, UDP и ICMP.
-s, –src, –source	`iptables -A INPUT -s 192.168.1.1`	IP-адрес(а) источника пакета. Адрес источника может указываться так, как показано в примере, тогда подразумевается единственный IP-адрес. А можно указать адрес в виде address/mask, например как 192.168.0.0/255.255.255.0, или более современным способом 192.168.0.0/24, т.е. фактически определяя диапазон адресов Как и ранее, символ !, установленный перед адресом, означает логическое отрицание, т.е. –source ! 192.168.0.0/24 означает любой адрес кроме адресов 192.168.0.x.
-d, –dst, –destination	`iptables -A INPUT -d 192.168.1.1`	IP-адрес(а) получателя. Имеет синтаксис схожий с критерием –source, за исключением того, что подразумевает адрес места назначения. Точно так же может определять как единственный IP-адрес, так и диапазон адресов. Символ ! используется для логической инверсии критерия.
-i, –in-interface	`iptables -A INPUT -i eth0`	Интерфейс, с которого был получен пакет. Использование этого критерия допускается только в цепочках INPUT, FORWARD и PREROUTING, в любых других случаях будет вызывать сообщение об ошибке. При отсутствии этого критерия предполагается любой интерфейс, что равносильно использованию критерия -i +. Как и прежде, символ ! инвертирует результат совпадения. Если имя интерфейса завершается символом +, то критерий задает все интерфейсы, начинающиеся с заданной строки, например -i PPP+ обозначает любой PPP интерфейс, а запись -i ! eth+ – любой интерфейс, кроме любого eth.
-o, –out-interface	`iptables -A FORWARD -o eth0`	Задает имя выходного интерфейса. Этот критерий допускается использовать только в цепочках OUTPUT, FORWARD и POSTROUTING, в противном случае будет генерироваться сообщение об ошибке. При отсутствии этого критерия предполагается любой интерфейс, что равносильно использованию критерия -o +. Как и прежде, символ ! инвертирует результат совпадения. Если имя интерфейса завершается символом +, то критерий задает все интерфейсы, начинающиеся с заданной строки, например -o eth+ обозначает любой eth интерфейс, а запись -o ! eth+ - любой интерфейс, кроме любого eth.
-f, –fragment	`iptables -A INPUT -f`	Правило распространяется на все фрагменты фрагментированного пакета, кроме первого, сделано это потому, что нет возможности определить исходящий/входящий порт для фрагмента пакета, а для ICMP-пакетов определить их тип. С помощью фрагментированных пакетов могут производиться атаки на ваш брандмауэр, так как фрагменты пакетов могут не отлавливаться другими правилами. Как и раньше, допускается использования символа ! для инверсии результата сравнения. только в данном случае символ ! должен предшествовать критерию -f, например ! -f. Инверсия критерия трактуется как «все первые фрагменты фрагментированных пакетов и/или нефрагментированные пакеты, но не вторые и последующие фрагменты фрагментированных пакетов».

Неявные критерии

Критерии, которые подгружаются неявно и становятся доступны, например при указании критерия –protocol tcp. На сегодняшний день существует три автоматически подгружаемых расширения, это TCP критерии, UDP критерии и ICMP критерии (при построении своих правил я столкнулся с необходимостью явного указания ключа -m tcp, т.е. о неявности здесь говорить не приходится, поэтому будьте внимательнее при построении своих правил, если что-то не идет – пробуйте явно указывать необходимое расширение. прим. перев.). Загрузка этих расширений может производиться и явным образом с помощью ключа -m, -match, например -m tcp. 6.4.2.1. TCP критерии

Этот набор критериев зависит от типа протокола и работает только с TCP пакетами. Чтобы использовать их, вам потребуется в правилах указывать тип протокола –protocol tcp. Важно: критерий –protocol tcp обязательно должен стоять перед специфичным критерием. Эти расширения загружаются автоматически как для tcp протокола, так и для udp и icmp протоколов. (О неявной загрузке расширений я уже упоминал выше прим. перев.).

TCP критерии

Критерий –sport, –source-port Пример iptables -A INPUT -p tcp –sport 22 Описание Исходный порт, с которого был отправлен пакет. В качестве параметра может указываться номер порта или название сетевой службы. Соответствие имен сервисов и номеров портов вы сможете найти в файле /etc/services. При указании номеров портов правила отрабатывают несколько быстрее. однако это менее удобно при разборе листингов скриптов. Если же вы собираетесь создавать значительные по объему наборы правил, скажем порядка нескольких сотен и более, то тут предпочтительнее использовать номера портов. Номера портов могут задаваться в виде интервала из минимального и максимального номеров, например –source-port 22:80. Если опускается минимальный порт, т.е. когда критерий записывается как –source-port :80, то в качестве начала диапазона принимается число 0. Если опускается максимальный порт, т.е. когда критерий записывается как –source-port 22:, то в качестве конца диапазона принимается число 65535. Допускается такая запись –source-port 80:22, в этом случае iptables поменяет числа 22 и 80 местами, т.е. подобного рода запись будет преобразована в –source-port 22:80. Как и раньше, символ ! используется для инверсии. Так критерий –source-port ! 22 подразумевает любой порт, кроме 22. Инверсия может применяться и к диапазону портов, например –source-port ! 22:80. За дополнительной информацией обращайтесь к описанию критерия multiport. Критерий –dport, –destination-port Пример iptables -A INPUT -p tcp –dport 22 Описание Порт или диапазон портов, на который адресован пакет. Аргументы задаются в том же формате, что и для –source-port. Критерий –tcp-flags Пример iptables -p tcp –tcp-flags SYN,FIN,ACK SYN Описание Определяет маску и флаги tcp-пакета. Пакет считается удовлетворяющим критерию, если из перечисленных флагов в первом списке в единичное состояние установлены флаги из второго списка. Так для вышеуказанного примера под критерий подпадают пакеты у которых флаг SYN установлен, а флаги FIN и ACK сброшены. В качестве аргументов критерия могут выступать флаги SYN, ACK, FIN, RST, URG, PSH, а так же зарезервированные идентификаторы ALL и NONE. ALL – значит ВСЕ флаги и NONE - НИ ОДИН флаг. Так, критерий –tcp-flags ALL NONE означает – «все флаги в пакете должны быть сброшены». Как и ранее, символ ! означает инверсию критерия Важно: имена флагов в каждом списке должны разделяться запятыми, пробелы служат для разделения списков. Критерий –syn Пример iptables -p tcp –syn Описание Критерий –syn является по сути реликтом, перекочевавшим из ipchains. Критерию соответствуют пакеты с установленным флагом SYN и сброшенными флагами ACK и FIN. Этот критерий аналогичен критерию –tcp-flags SYN,ACK,FIN SYN. Такие пакеты используются для открытия соединения TCP. Заблокировав такие пакеты, вы надежно заблокируете все входящие запросы на соединение, однако этот критерий не способен заблокировать исходящие запросы на соединение. Как и ранее, допускается инвертирование критерия символом !. Так критерий ! –syn означает – «все пакеты, не являющиеся запросом на соединение», т.е. все пакеты с установленными флагами FIN или ACK. Критерий –tcp-option Пример iptables -p tcp –tcp-option 16 Описание Удовлетворяющим условию данного критерия будет будет считаться пакет, TCP параметр которого равен заданному числу. TCP Option - это часть заголовка пакета. Она состоит из 3 различных полей. Первое 8-ми битовое поле содержит информацию об опциях, используемых в данном соединении. Второе 8-ми битовое поле содержит длину поля опций. Если следовать стандартам до конца, то следовало бы реализовать обработку всех возможных вариантов, однако, вместо этого мы можем проверить первое поле и в случае, если там указана неподдерживаемая нашим брандмауэром опция, то просто перешагнуть через третье поле (длина которого содержится во втором поле). Пакет, который не будет иметь полного TCP заголовка, будет сброшен автоматически при попытке изучения его TCP параметра. Как и ранее, допускается использование флага инверсии условия !. Дополнительную информацию по TCP Options вы сможете найти на Internet Engineering Task Force 6.4.2.2. UDP критерии

В данном разделе будут рассматриваться критерии, специфичные только для протокола UDP. Эти расширения подгружаются автоматически при указании типа протокола –protocol udp. Важно отметить, что пакеты UDP не ориентированы на установленное соединение, и поэтому не имеют различных флагов которые дают возможность судить о предназначении датаграмм. Получение UDP пакетов не требует какого либо подтверждения со стороны получателя. Если они потеряны, то они просто потеряны (не вызывая передачу ICMP сообщения об ошибке). Это

предполагает наличие значительно меньшего числа дополнительных критериев, в отличие от TCP пакетов. Важно: Хороший брандмауэр должен работать с пакетами любого типа, UDP или ICMP, которые считаются не ориентированными на соединение, так же хорошо как и с TCP пакетами. Об этом мы поговорим позднее, в следующих главах.

UDP критерии

Критерий –sport, –source-port Пример iptables -A INPUT -p udp –sport 53 Описание Исходный порт, с которого был отправлен пакет. В качестве параметра может указываться номер порта или название сетевой службы. Соответствие имен сервисов и номеров портов вы сможете найти в файле other/services.txt. При указании номеров портов правила отрабатывают несколько быстрее. однако это менее удобно при разборе листингов скриптов. Если же вы собираетесь создавать значительные по объему наборы правил, скажем порядка нескольких сотен и более, то тут предпочтительнее использовать номера портов. Номера портов могут задаваться в виде интервала из минимального и максимального номеров, например -source-port 22:80. Если опускается минимальный порт, т.е. когда критерий записывается как –source-port :80, то в качестве начала диапазона принимается число 0. Если опускается максимальный порт, т.е. когда критерий записывается как –source-port 22: , то в качестве конца диапазона принимается число 65535. Допускается такая запись –source-port 80:22 , в этом случае iptables поменяет числа 22 и 80 местами, т.е. подобного рода запись будет преобразована в –source-port 22:80 . Как и раньше, символ ! используется для инверсии. Так критерий –source-port ! 22 подразумевает любой порт, кроме 22. Инверсия может применяться и к диапазону портов, например –source-port ! 22:80. Критерий –dport, –destination-port Пример iptables -A INPUT -p udp –dport 53 Описание Порт, на который адресован пакет. Формат аргументов полностью аналогичен принятому в критерии –source-port.

ICMP критерии

Этот протокол используется, как правило, для передачи сообщений об ошибках и для управления соединением. Он не является подчиненным IP протоколу, но тесно с ним взаимодействует, поскольку помогает обрабатывать ошибочные ситуации. Заголовки ICMP пакетов очень похожи на IP заголовки, но имеют и отличия. Главное свойство этого протокола заключается в типе заголовка, который содержит информацию о том, что это за пакет. Например, когда мы пытаемся соединиться с недоступным хостом, то мы получим в ответ сообщение `ICMP host unreachable`. Полный список типов ICMP сообщений, вы можете посмотреть в приложении Типы ICMP. Существует только один специфичный критерий для ICMP пакетов. Это расширение загружается автоматически, когда мы указываем критерий `–protocol icmp`. Заметьте, что для проверки ICMP пакетов могут употребляться и общие критерии, поскольку известны и адрес источника и адрес назначения и пр.

Критерий `–icmp-type`

Пример

```
iptables -A INPUT -p icmp --icmp-type 8
```

Описание
Тип сообщения ICMP определяется номером или именем. Числовые значения определяются в RFC 792. Чтобы получить список имен ICMP значений выполните команду

```
iptables --protocol icmp --help
```

,

или посмотрите приложение Типы ICMP. Как и ранее, символ ! инвертирует критерий, например

```
--icmp-type ! 8
```

.

Ссылки

Источник

http://sysadminmosaic.ru/iptables/iptables_ru

2020-08-25 09:27