Команда ipfw (IP Firewall)

Содержание

Описание

IP Firewall – механизм фильтрации проходящих через узел IP-сети пакетов в соответствии с различными критериями. Администратор сети может определить набор входных и выходных фильтров. Входные фильтры определяют, какие пакеты могут быть приняты узлом, выходные – какие пакеты могут быть перенаправлены узлом в процессе маршрутизации. Каждый фильтр описывает класс пакетов и определяет, как эти пакеты должны быть обработаны (отклонить и зарегистрировать, принять, принять и зарегистрировать).

Пакеты могут быть отфильтрованы на основе следующих критериев:

- Протокол (IP, TCP, UDP, ICMP, ARP).
- IP-адрес отправителя и/или IP-адрес получателя (и номера портов для TCP и UDP).
- Входной сетевой интерфейс.
- Является ли пакет запросом TCP/IP-соединения (пакет пытается запустить TCP/IP-сессию).
- Является ли пакет первым или последующим фрагментом фрагментированного пакета.
- Имеет ли пакет определенные IP-опции.
- MAC-адрес получателя или отправителя.

Синтаксис:

```
list
show | reset
rearrange [STEP]
flush
quiet | -quiet
del  RULE_NUMBER
dump RULE_NUMBER
mov  RULE_A RULE_B
add[out] [NUM] [IFNAME] [chN] rules...

rules: [{setpri|addpri}=[N]] accept|reject|rpfilter|pass [log]
   [vlan={N|any|$ACL}] [dot1p=N] [swg=N] [ether={X|any}] [dscp=N|tos=N] [prf]
   -f "pcap filter expression"
   |
   PROTO from [not] ADDR [PORTs] to [not] ADDR [PORTs]

   PROTO: [all] | tcp | udp | icmp | arp | proto NUMBER
   ADDR: IP | $LOCAL | $ROUTE | $ACL | mac x:x:x:x:x:x }
   PORTS: NUM[:NUM] [NUM] ...
```

Параметры

Параметр	Описание
list	Выводит текущий набора фильтров.
show / reset	Вывод/сброс статистики созданных фильтров.

rearrange [STEP]	Перенумерация всех фильтров с шагом "*STEP*" (по умолчанию 5).
flush	Все определенные в данный момент фильтры удаляются. Механизм фильтрации отключается.
quiet | -quiet	Отключение регистрации отброшенных пакетов. По умолчанию регистрация включена, для отключения её следует выполнить команду "*ipfw quiet*".
del RULE_NUMBER	Удаление указанного фильтра из списка. Порядковый номер фильтра можно посмотреть командой "*ipfw list*".
dump RULE_NUMBER	Отображает скомпилированный псевдо-код фильтра, заданный в формате PCAP. Позволяет визуально оценить сложность /оптимальность, либо корректность полученного фильтра.
mov RULE_A RULE_B	Перемещает фильтр с номером "*A*" в позицию "*B*".
add[out] [NUM] [IFNAME] [chN] rules...	Используются для добавления фильтра в набор для обработки входящих пакетов, при добавлении части "*out*" фильтр затронет обработку исходящих с устройства пакетов. • "*NUM*" – указание порядкового номера правила в списке (необязательный параметр). • "*IFNAME*" – имя сетевого интерфейса, через который пакет попадает в систему (дополнительный параметр). • *"chN"* – аргумент, при совпадении с которым пакету будет назначен канал обслуживания N, созданный комадой "qm chN".

rules: [{setpri|addpri}=[N]] accept|reject|rpfilter|pass [log] [vlan={N|any|$ACL}] [dot1p=N] [swg=N] [ether={X|any}] [dscp=N|tos=N] [prf] -f "pcap filter expression"	• "*setpri=[N]*" – назначение пакету нового приоритета независимо от того, какой приоритет он имел до этого.
	• "*addpri=[N]*" – повышение приоритета пакета до указанной величины, если он имел более низкий приоритет. С помощью параметра "*addpri*" приоритет можно только повысить.
	• "*accept*" – пакет обрабатывается системой, игнорируя другие правила IP Firewall.
	• "*reject*" – пакет отбрасывается.
	• "*rpfilter*" – позволяет убедиться, что отправитель пакета действительно доступен через тот сетевой интерфейс, через который пакет от него поступил в систему. Если доступность отправителя подтверждается, то обработка пакета модулем "*ipfw*" продолжается, иначе пакет уничтожается. Данный фильтр можно добавлять в список правил первым.
	• "*pass*" – позволяет "пропустить" правило, выполнив связанные с ним действия, и продолжить просмотр других правил по списку.
	• "*log*" – включение записи действий фильтра в системный журнал (необязательный параметр).
	• "*vlan=*" – классификатор, позволяющий анализировать VLAN ID ((допустимые значения 0-4095):
	  ◦ "*N*" – фильтр будет пропускать тегированные пакеты с указанной меткой N.
	  ◦ "*any*" – фильтр будет пропускать все тегированные пакеты с любым значением VLAN ID.
	  ◦ "*$ACL*" – фильтр будет пропускать тегированные пакеты с метками VLAN, указанными в виде списка "*$ACL*" (описание списков ACL см. в разделе «Списки контроля доступа (команда «acl»)).
	• "*dot1p=N*" – классификатор, позволяющий анализировать приоритет 802.1p (допустимые значения 0-7).
	• "*swg=N*" – классификатор, позволяющий анализировать номер группы коммутации.
	• "*ether={X|any}*" – классификатор, позволяющий анализировать тип пакета. При выборе опции "*any*" фильтр будет пропускать пакеты любых типов.
	• "*dscp=N*" – классификатор, позволяющий анализировать наличие метки DSCP (допустимые значения 0-63).
	• "*tos=N*" – классификатор, позволяющий анализировать наличие метки TOS.
	• "*prf*" – включает фильтрацию трафика, формируемого интерфейсами PRF.
	• "*-f "pcap filter expressio*n" – позволяет использовать фильтры pcap.

PROTO from [not] ADDR [PORTs] to [not] ADDR [PORTs]	Классификаторы уточняют определенное направление передачи от и/или к ID:
	• "*from*" – указатель IP-адреса отправителя.
	• "*to*" – указатель IP-адреса получателя.
	• "*not*" – отрицательная приставка, может использоваться после ключевых слов "*from*" и "*to*", при этом её действие будет распространяться только на соответствующий адрес(а), но не на порты, если они используются в команде.
	• "*ADDR*" – адрес отправителя или получателя (endpoint). Синтаксис этого поля зависит от значения поля "*proto*". Если "*proto*" указан как "*all*" или "*icmp*", то "*ADDR*" содержит адресную информацию. Если "*proto*" задан как "*udp*" или "*tcp*", то "*ADDR*" содержит адресную информацию и необязательный список портов. Адресная информация задаётся как IP-адрес и необязательная маска. IP-адрес должен задаваться в традиционном числовом формате (*nn.nn.nn.nn*). Необязательная маска может быть задана как длиной маски в битах, так и числовым значением в формате (*nnn.nnn.nnn.nnn*).
	
	Возможные форматы :
	`nn.nn.nn.nn`
	`nn.nn.nn.nn:xxx.xxx.xxx.xxx`
	`nn.nn.nn.nn/NN`
	
	Запись "*0/0*" описывает все возможные IP-адреса.

PROTO: [all] | tcp | udp | icmp | arp | proto NUMBER	Классификаторы ограничивают по признаку соответствия определенному протоколу. Возможные протоколы: TCP, UDP, ICMP, ARP, либо числовое значение номера протокола. ARP-пакеты будут всегда пропускаться для тех IP-адресов и диапазонов IP-адресов, которые указаны в разрешающих (accept) фильтрах, даже если эти фильтры созданы для других типов пакетов.
ADDR: IP | $LOCAL | $ROUTE | $ACL | mac x:x:x:x:x:x }	Если необходимо составить фильтр, который применяется к нескольким сетевым адресам или группам, то вместо повторения однотипных правил удобнее и эффективнее сгруппировать все фигурирующие адреса в соответствующий список доступа и указать имя списка в качестве IP-адреса (*$ACLRULE*). Существует несколько предопределённых динамических списков: • "*$LOCAL*" – список, включающий в себя все локальные адреса, принадлежащие данному маршрутизатору. Его можно использовать для удобной записи фильтров, ограничивающих/разрешающих доступ к самому устройству. • "*$ROUTE*" – список, содержащий текущую системную таблицу маршрутов, за исключением "*default route*". Совпадение адреса с этим списком означает, что для данного адреса существует точный маршрут и не будет использован маршрут по умолчанию. • "*$ACL*" – список IP-адресов или сетей, на которые будет распространяться это правило. • "*mac x:x:x:x:x:x*" – для интерфейсов, имеющих физические MAC-адреса в стандарте Ethernet, допускается использовать непосредственно цифровое значение MAC-адреса с предшествующим ключевым словом "*mac*". При этом для входящих фильтров можно указать только MAC-адрес отправителя, а для исходящих только адрес получателя. Кроме того, вместо числового значения, можно указывать ключевое слово "*$BS*". В этом случае в качестве числового значения будет использоваться реальный MAC-адрес сектора базовой станции, к которой подключен абонентский терминал (при наличии). ⚠ **ВНИМАНИЕ** Следует иметь ввиду, что правила, использующие MAC-адреса для входящих пакетов будут обработаны раньше всех остальных правил, а правила для исходящих пакетов, напротив, будут обработаны в самую последнюю очередь.
PORTS: NUM[: NUM] [NUM] ...	Используется для фильтрации трафика по номерам портов. Можно использовать список портов для указания сразу нескольких портов в одной команде. Первый элемент списка портов может задавать диапазон номеров от меньшего к большему, разделённый двоеточием.

Расширить действие фильтров можно при помощи pcap-выражений.

Правила фильтрации пакетов

Каждый пакет, приходящий на маршрутизатор, проходит через набор входных (блокирующих) фильтров. Пакеты, принимаемые набором входных фильтров, обрабатываются IP-уровнем ядра маршрутизатора. Если IP-уровень определяет, что пакет не относится к данному узлу и должен пройти дальше, пакет отправляется на выходные (маршрутизирующие) фильтры.

Информация о пакетах, отбрасываемых любым фильтром, отображается на терминале оператора, а сами пакеты отбрасываются без уведомления отправителя.

Пакеты, проходящие через набор фильтров, проверяются каждым фильтром от первого до последнего, или до тех пор, пока не дойдут до первого подходящего фильтра. Алгоритм следующий:

1. Если фильтры не заданы, пакет принимается.
2. Если фильтры заданы, первый подходящий фильтр решит судьбу пакета. Если фильтр разрешающий, пакет принимается, если запрещающий – пакет отбрасывается.
3. Если ни один из фильтров не подходит пакету, пакет принимается.

Примеры

Установим фильтр, запрещающий прохождение всех пакетов с IP-адреса "*1.1.1.1*" на адрес "*2.2.2.2*".

```
ipfw add reject all from 1.1.1.1 to 2.2.2.2
```

Установим фильтр, запрещающий прохождение пакетов из сети "*1.1.1.0/24*" на IP-адрес "*2.2.2.2*".

```
ipfw add reject all from 1.1.1.0/24 to 2.2.2.2
```

Либо.

```
ipfw add reject all from 1.1.1.1:255.255.255.0 to 2.2.2.2
```

Установим фильтр, полностью блокирующий все пакеты из сети "*1.1.1.0*" класса C, посланные на любой адрес (если они будут проходить через данный маршрутизатор).

```
ipfw add reject all from 1.1.1.0/24 to 0/0
```

Установим фильтр, позволяющий всем пакетам TCP обращаться к сервису smtp (почтовый агент) на устройстве с IP-адресом "*192.5.42.1*". Сервис "*smtp*" определяется номером порта 25.

```
ipfw add accept tcp from 0/0 to 192.5.42.1 25
```

Установим фильтр, который разрешит прохождение TCP-пакетов на IP-адрес "*1.1.1.1*", если целевой порт при этом попадает в диапазон от 900 до 5000 или равен 25 (*smtp*) или 113 (*ident*).

```
ipfw add accept tcp from 0/0 to 1.1.1.1 900:5000 25 113
```

В предыдущих примерах IP-адрес отправителя использовался как главный и единственный критерий проверки надежности источника. К сожалению, есть возможность посылать пакеты с ненадежного IP-адреса, подставляя в качестве обратного адреса тот, которому вы доверяете (этот вид атаки называется IP-spoofing). Помимо проверок IP-адреса необходимо также проверить, каким путём шёл пакет или, что более практично, через какой сетевой интерфейс он был принят.

Все подсети внутренней сети, включая и IP-адрес внутреннего хоста "*innerhost*", принадлежат одной сети (или группе сетей). Предположим, что во внешней сети нет ни одного хоста, принадлежащего диапазону, выделенному для внутренней сети. Следовательно, любые пакеты, которые принимаются через интерфейс "*rf5.0*" маршрутизатора, на котором запущен Firewall, и имеют IP-адрес отправителя, принадлежащий диапазону адресов внутренней сети, должны блокироваться. Следующая команда позволяет сделать это, причём этот фильтр будет применяться только к пакетам, приходящим через интерфейс "*rf5.0*". Пакеты, поступающие через любой другой интерфейс, блокироваться не будут.

```
ipfw add rf5.0 reject all from innerhost/16 to 0/0
```

Дополнительно можно заблокировать прохождение пакетов с IP-адресом отправителя из закольцованной сети (loopback) "*127.0.0.0*".

```
ipfw add rf5.0 reject all from 127.0.0.0/8 to 0/0
```

Title

TCP/IP-клиенты обычно используют порты в диапазоне от 900 до 5000, а сервера обычно обслуживают порты с номерами ниже 900 или выше 5000. Следовательно, следующая пара фильтров позволит запретить любым внешним клиентам работу с нашими серверами (предполагается, что "*rf5.0*" - это интерфейс, соединяющий нас с внешней сетью).

Установим фильтр, который будет пропускать все входящие извне пакеты, посылаемые на сервисы с портами в диапазоне от 900 до 5000 (обычно используемые клиентами), а второй фильтр будет отбрасывать всё остальное.

```
ipfw add rf5.0 accept tcp from 0/0 to 0/0 900:5000
ipfw add rf5.0 reject tcp from 0/0 to 0/0
```

В отличие от протокола TCP, который ориентирован на установление соединений, UDP-протокол использует отдельные пакеты (датаграммы). В этом протоколе каждый пакет передаётся независимо от других, и логические сеансы, которые могли бы устанавливаться между UDP/IP-клиентом и сервером, существуют только на уровне приложений и не видны на уровне UDP. Поскольку все пакеты независимы, по заголовку пакета невозможно определить, посылается пакет от сервера клиенту или наоборот (фактически, в протоколе UDP, участники действуют как равноправные партнёры и термины сервер и клиент явно не определены). Поэтому лучшее, что можно сделать, это как можно более точно определить диапазон портов UDP, который может использоваться для связи с внешним миром.

Заблокируем весь трафик UDP, приходящий на сетевой интерфейс "*rf5.0",* установив при этом фильтр, разрешающий взаимодействие между нашим и внешними DNS-серверами (для обмена данными с сервером DNS используется порт 53).

```
ipfw add accept udp from 0/0 53 to 0/0 53
ipfw add rf5.0 reject udp from 0/0 to 0/0
```