

Глава 4. Пакетный фильтр Netfilter (Iptables)

1. Общие сведения о Netfilter

Общие сведения о Netfilter



- Пакетный фильтр в ядрах Linux 2.4 и выше называется Netfilter, а утилита для его настройки – `iptables` или более современная утилита `nft`
- Netfilter – первый пакетный фильтр для Linux, который относится к классу фильтров с проверкой состояния (stateful filter)
- Команда `iptables (nft)` добавляет, удаляет или изменяет правила фильтрации пакетов, записанные в специальной таблице ядра. Данные этой таблицы сбрасываются при любой перезагрузке ОС
- Чтобы таблицы фильтрации сохранялись нужна специальная служба, которая восстановит правила при запуске ОС

Пакетный фильтр в ядрах Linux 2.4 называется Netfilter, а утилита для его настройки – `iptables` (домашняя страница в Интернете <http://netfilter.org>).

Netfilter – первый пакетный фильтр для Linux, который относится к классу фильтров с проверкой состояния (stateful filter)

Примечание: фильтры данного класса являются гибридом шлюза приложений и пакетного фильтра. Такие фильтры анализируют содержимое пакетов, хотя и не так подробно, как шлюзы приложений.

Netfilter хранит информацию о всех соединениях в памяти, позволяет обнаруживать и блокировать сканирование, DoS атаки.

Netfilter – сложнее и надежнее, чем его предшественник `Ipchains`

Команда `iptables` добавляет, удаляет или изменяет правила фильтрации пакетов, записанные в специальной таблице ядра. Данные этой таблицы сбрасываются при любой перезагрузке ОС.

Команда `nft` современная альтернатива `iptables`.

Для обеспечения сохранения правил фильтрации в файлах предназначены утилиты `iptables-save` и `iptables-restore`.

Для автоматической загрузки правил фильтрации нужно использовать специальную службу или скрипт.

Netfilter – часть пакетного фильтра, находящаяся в ядре ОС GNU/Linux, `iptables (nft)` – пользовательская утилита (пользовательская часть пакетного фильтра)

Общие сведения о Netfilter

- Netfilter поддерживает пять видов таблиц для хранения правил, с которыми сверяются все сетевые пакеты:
 1. **filter** — фильтрация пакетов
 2. **nat** — трансляция сетевых адресов
 3. **mangle** — используется для изменения битов качества обслуживания (QoS) в заголовках TCP
 4. **raw** — исключения из правил отслеживающих соединения
 5. **security** — применение MAC (Mandatory Access Control) для сетевых соединений, применяется после таблицы **filter**
- В **nft** можно создавать свои таблицы

Netfilter поддерживает три вида таблиц для хранения правил, с которыми сверяются все сетевые пакеты:

- **filter** — используется для фильтрации пакетов
- **nat** — отвечает за таблицу трансляции сетевых адресов
- **mangle** — используется для изменения битов качества обслуживания (QoS) в заголовках TCP (применяется редко)
- **raw** — исключения из правил отслеживающих соединения
- **security** — применение MAC (Mandatory Access Control) для сетевых соединений, применяется после таблицы **filter**

Опция **-t** (**--table**) позволяет указать таблицу, с которой будет работать утилита **iptables**.

- **-t filter** — используется таблица фильтрации (по умолчанию);
- **-t nat** — используется таблица сетевых трансляции адресов
- **-t mangle** — используется таблица специализированных модификаций пакетов и т. д.

В некоторых ОС семейства Linux могут быть созданы другие таблицы или отсутствовать указанные выше.

Для создания собственных таблиц нужно использовать **nft**.

2. Фильтрации пакетов



Фильтрации пакетов

- Правила фильтрации пакетов записываются в виде цепочек
- Каждое правило заканчивается действием (target, целью)
- Если пакет прошел всю цепочку правил, не встретив ни одного правила, которому он удовлетворяет, он подвергается обработке в соответствии с установленной политикой (policy) для данной цепочки

Правила фильтрации пакетов записываются в виде цепочек (как и в IPchains)

Каждое правило заканчивается действием (target, целью), обычно, **ACCEPT** (пропустить) или **DROP** (отбросить)

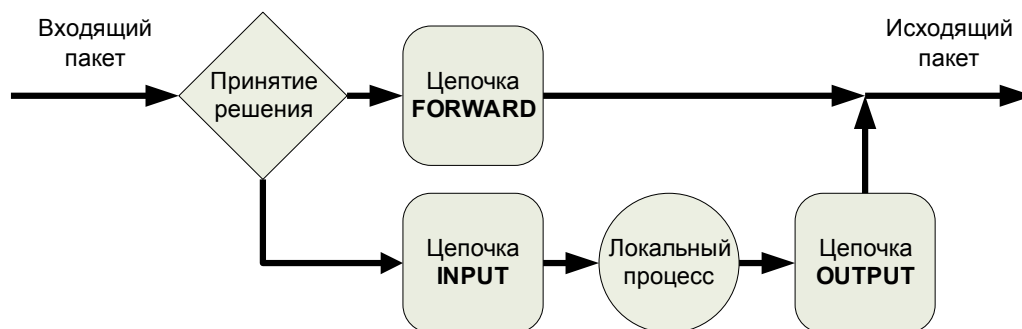
Заранее определено три цепочки правил: **INPUT**, **OUTPUT** и **FORWARD** для, соответственно, обработки входящих, исходящих и перенаправленных пакетов.

Если пакет прошел всю цепочку правил, не встретив ни одного правила, которому он удовлетворяет, он подвергается обработке в соответствии с установленной политикой (policy) для данной цепочки :
пропустить - **ACCEPT** или
отклонить - **DROP**

Примечание: текущую политику можно определить, выполнив команду `iptables -L` или `nft`

Фильтрации пакетов

- Заранее определено три цепочки правил: INPUT, OUTPUT и FORWARD



Когда появляется входящий пакет, он сначала проверяется на предмет его назначения. Эта процедура называется маршрутизацией (routing).

Если данный пакет предназначен для данного хоста, то он попадает на начало цепочки **INPUT**, и в случае ее успешного прохождения передается ожидающему его локальному процессу.

В случае, если пакет предназначается куда-либо на другой интерфейс, он передается (при разрешенном форвардинге в ядре `/proc/sys/net/ipv4/ip_forward = 1`) на вход цепи **FORWARD**.

Если форвардинг в ядре не включен, то пакет отклоняется (DROP).

Если пакет принимается (ACCEPT) правилом в цепочке FORWARD или политика это допускает, то он передается на заданный интерфейс.

Программа, работающая на компьютере, может посылать пакеты. Они попадают на вход цепочки **OUTPUT**. Если пакеты проходят эту цепь, то они покидают систему через заданный интерфейс.

3. Основные операции с цепочками и правилами



Основные операции с цепочками и правилами

- Для манипуляции цепочками правил можно применять следующие опции команды `iptables`:
 - `-N (--new-chain)` - создание новой цепочки;
 - `-X (--delete-chain)` - удаление пустой цепочки;
 - `-P (--policy)` - установка политики для встроенной цепочки;
 - `-L (--list)` - получение списка правил в цепочке;
 - `-F (--flush)` - очистка цепочки (сброс правил);
 - `-Z (--zero)` - обнуление счетчиков байтов и пакетов во всех правилах цепочки.

Для манипуляции цепочками правил можно применять следующие опции команды `iptables`:

- `-N (--new-chain)` - создание новой цепочки;
- `-X (--delete-chain)` - удаление пустой цепочки;
- `-P (--policy)` - установка политики для встроенной цепочки;
- `-L (--list)` - получение списка правил в цепочке;
- `-F (--flush)` - очистка цепочки (сброс правил);
- `-Z (--zero)` - обнуление счетчиков байтов и пакетов во всех правилах цепочки.

Основные операции с цепочками и правилами

- Опции команды `iptables`, которые позволяют управлять правилами в цепочке:
 - `-A` (`--append`) - добавление нового правила в цепочку (по умолчанию последним);
 - `-I` (`--insert`) - вставка в заданную позицию правила в цепочку (по умолчанию первым);
 - `-R` (`--replace`) - замена в цепочке заданного правила;
 - `-D` (`--delete`) - удаление правила в заданной позиции или первого правила, подходящего специфицированному условию
- Опция `-j` (`--jump`) указывает действие

Следующие опции команды `iptables` позволяют управлять правилами в цепочке:

- `-A` (`--append`) - добавление нового правила в цепочку (по умолчанию последним);
- `-I` (`--insert`) - вставка в заданную позицию правила в цепочку (по умолчанию первым);
- `-R` (`--replace`) - замена в цепочке заданного правила;
- `-D` (`--delete`) - удаление правила в заданной позиции или первого правила, подходящего специфицированному условию

Опция `-j` (`--jump`) указывает действие (`DROP` или `ACCEPT`), которое должно быть произведено с пакетами, удовлетворяющими данному правилу.

Пример: Эта команда добавляет в цепочку `INPUT` правило, которое отбрасывает любые `icmp` пакеты, направленные с `loopback` интерфейса

```
[root@linux1]# iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP
[root@linux1]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      icmp --  linux1                anywhere
Chain FORWARD (policy ACCEPT)
target     prot opt source                destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@linux1]# ping localhost
PING linux1 (127.0.0.1) 56(84) bytes of data.
--- linux1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2006ms
```

Пример: удаляем первое правило в цепочке `INPUT`

```
[root@linux1]# iptables -D INPUT 1
[root@linux1]# iptables -L
```

Глава 4. Пакетный фильтр Netfilter (Iptables)

```
Chain INPUT (policy ACCEPT)
target      prot opt source      destination
```

```
Chain FORWARD (policy ACCEPT)
target      prot opt source      destination
```

```
Chain OUTPUT (policy ACCEPT)
target      prot opt source      destination
```

Пример: удалить можно также, явно указав содержимое правила

```
[root@linux1]# iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP
```

4. Составление простейших правил фильтрации



Составление простейших правил фильтрации

- `-s (--src)` — адрес отправителя пакета
- `-d (--dst)` — адрес получателя пакета
- `-p (--protocol)` — протокол
- `-i (--in-interface)` — входящий интерфейс
- `-o (--out-interface)` — исходящий интерфейс
- `--dport (--destination-port)` — порт назначения
- `--sport (--source-port)` — порт источника
- `!` — отрицание
- `-f` — обработка фрагментов

Опция `-s` или `--src` применяется для указания адреса отправителя пакета

Опция `-d` или `--dst` применяется для указания адреса получателя пакета.

Адреса отправителя и получателя могут быть указаны либо в виде IP адреса, либо в виде доменного имени.

При необходимости указать группу адресов можно привести адрес сети и через косую черту либо маску сети, либо количество битов в сетевой части IP адреса.

Пример: 192.168.1.0/24 или 199.95.207.0/255.255.255.0.

Если в качестве адреса указана конструкция 0/0 - это означает все адреса.

Пример: запрещаем получение всех пакетов от всех хостов
`[root@linux1 root]# iptables -A INPUT -s 0/0 -j DROP`

Для инверсии значения опций указания адресов (`-s` и `-d`) можно использовать восклицательный знак, устанавливаемый перед адресом.

Пример: запрещаем получение всех пакетов от всех хостов, кроме локальных пакетов
`[root@linux1 root]# iptables -A INPUT -s ! localhost -j DROP`

Для указания протокола следует использовать опцию `-p`, после которой необходимо указать либо имя, либо номер протокола (`/etc/protocols`).

Если перед именем или номером протокола указан восклицательный знак, это обозначает все протоколы, кроме указанного.

Указать интерфейс, с которого должны приходить пакеты можно, используя опцию `-i (--in-interface)`.

Опция `-o (--out-interface)` указывает интерфейс, через который систему

Глава 4. Пакетный фильтр Netfilter (Iptables)

покидают исходящие пакеты.

Для указания всех интерфейсов, начинающихся с некоторого префикса, например, `eth`, следует указать этот префикс, добавив после него `+`

Пример: `eth+` – все Ethernet интерфейсы.

Перед именем интерфейса допустимо указывать восклицательный знак для указания всех интерфейсов, кроме заданного.

При использовании протоколов TCP или UDP можно указать порт назначения `--destination-port` или `--dport`, а также порт источника `--source-port` или `--sport`.

В случае фрагментации пакетов второй и следующие фрагменты не будут содержать такую информацию, как номера портов, тип ICMP сообщения.

Опция `-f` позволяет строить правила для второго и последующего фрагмента, а установка восклицательного знака перед `-f` (`! -f`) будет применять правило только для первого фрагмента или нефрагментированных пакетов.

5. Трансляция сетевых адресов (NAT)

Трансляция сетевых адресов (NAT)



- Трансляцию сетевых адресов принято разделять на две категории:
 - Source NAT (SNAT) - трансляция адреса отправителя.
 - Destination NAT (DNAT) - трансляция адреса получателя.

Применение трансляции сетевых адресов вызывает изменения в адресах отправителя или получателя пакетов. При этом происходит запоминание информации об измененном пакете так, чтобы для ответного пакета можно было провести обратное преобразование.

Трансляцию сетевых адресов принято разделять на две категории:

- Source NAT (**SNAT**) - трансляция адреса отправителя.
- Destination NAT (**DNAT**) - трансляция адреса получателя.

Трансляция сетевых адресов (NAT)

- Таблица NAT поддерживает три встроенные цепочки
 - PREROUTING – трансляция адресов до принятия решения о маршрутизации
 - POSTROUTING – трансляция адресов после принятия решения о маршрутизации
 - OUTPUT – трансляция адресов, созданных локальным процессом, до принятия решения о маршрутизации

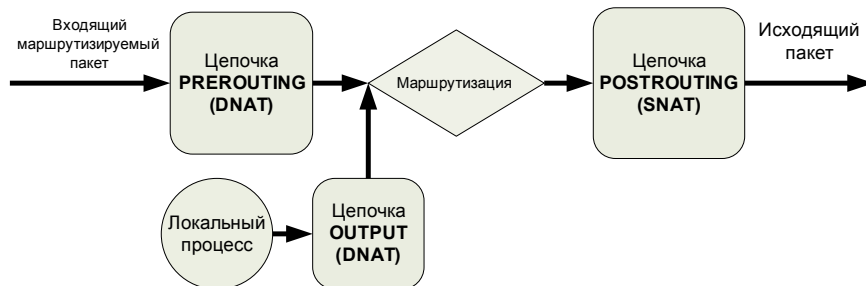


Таблица NAT поддерживает три встроенные цепочки

- **PREROUTING** – трансляция адресов до принятия решения о маршрутизации
- **POSTROUTING** – трансляция адресов после принятия решения о маршрутизации
- **OUTPUT** – трансляция адресов, созданных локальным процессом, до принятия решения о маршрутизации.

Трансляция адреса отправителя SNAT изменяет адрес отправителя пакета и всегда происходит после маршрутизации непосредственно перед тем, как пакет попадает в канал связи.

Маскарадинг (masquerading) - это форма SNAT.

Трансляция адреса назначения DNAT изменяет адрес назначения пакета и происходит до маршрутизации сразу после прихода пакета из канала связи.

Перенаправление портов, прозрачное проксирование и распределение нагрузки на серверы - это формы DNAT.

SNAT, используемая для подмены адреса источника пакета, производится в цепочке POSTROUTING непосредственно перед выходом пакета в линию связи.

Для включения SNAT необходимо

1. указать цель (target) –j SNAT
2. опцию --to, после которой необходимо указать адрес или диапазон адресов, которые будут использованы для подмены исходного адреса
3. опцию -o для указания внешний интерфейс.

Примеры: Изменяем адреса отправителя на 1.2.3.4.

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4
```

Изменяем адреса отправителя на диапазон 1.2.3.4, 1.2.3.5 или 1.2.3.6

Глава 4. Пакетный фильтр Netfilter (Iptables)

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4-1.2.3.6
```

Изменением адреса отправителя на диапазон 1.2.3.4 и порты 1-1023

```
iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to 1.2.3.4:1-1023
```

При использовании в цепи POSTROUTING цели MASQUERADE включается режим маскарadingа, являющегося частным случаем SNAT.

В режиме маскарadingа не нужно указывать адрес(а) для подмены адреса отправителя, поскольку адрес отправителя заменяется на адрес внешнего интерфейса.

Пример: адрес внешнего интерфейса eth0 будет присвоен всем исходящим пакетам

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Трансляция адресов получателя DNAT осуществляется в цепи PREROUTING или OUTPUT. При этом требуется указать

1. цель -j DNAT
2. опцию --to, которая указывает на какой адрес(а) должен быть подменен адрес получателя
3. опцию -i для указания входного интерфейса

Пример: Изменяем адреса получателя на 5.6.7.8

```
iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 5.6.7.8
```

Изменяем диапазон адресов получателей на 5.6.7.8, 5.6.7.9 or 5.6.7.10.

```
iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 5.6.7.8-5.6.7.10
```

Изменяем адреса получателей для web трафика to 5.6.7.8, port 8080.

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth0 -j DNAT --to 5.6.7.8:8080
```

Для устройства прозрачного прокси сервера нужно воспользоваться специализированной формой DNAT, называемой перенаправлением (redirection). Перенаправление можно включить, указав исходный порт, цель REDIRECT и порт, на который будут перенаправлен трафик.

Пример: Пересылка входящего на 80-й порт web трафика на прозрачный прокси squid

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

iptables позволяет отображать диапазон адресов в один адрес

Пример:

```
iptables -t nat -A POSTROUTING -s 192.168.1.1 -o eth1 -j SNAT --to 1.2.3.1
```

```
iptables -t nat -A POSTROUTING -s 192.168.1.2 -o eth1 -j SNAT --to 1.2.3.1
```

Можно отображать сеть в сеть

Пример:

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth1 -j SNAT --to 1.2.3.0/24
```

Для обеспечения правильной работы FTP с сетью, использующей трансляцию адресов, следует использовать загружаемые модули ядра ip_conntrack_ftp.o и ip_nat_ftp.o.

При использовании трансляции адресов все пакеты, исходящие из локальной сети и входящие в сеть, должны проходить через хост, осуществляющий трансляцию адресов.

Если используется трансляция адресов отправителя SNAT, при которой некоторый адрес отображается на неиспользуемые адреса в той же сети, компьютер, осуществляющий NAT, должен отвечать за правильное разрешение ARP, чего проще всего добиться с помощью IP алиасов.

При отображении адресов на совершенно другую сеть следует позаботиться, чтобы компьютер, осуществляющий NAT, мог принимать ответные пакеты. Это достигается с помощью конфигурации таблицы маршрутизации. Если компьютер с NAT является маршрутизатором по умолчанию, то изменений в таблице маршрутизации не требуется.

Если осуществляется трансляция DNAT порта в той же сети, следует убедиться, что все

Глава 4. Пакетный фильтр Netfilter (Iptables)

пакеты и ответные пакеты проходят через компьютер с NAT.

Пример: Эта команда осуществляет трансляцию по назначению DNAT для внутренней сети, отображая адрес 1.2.3.4 в адрес внутреннего сервера 192.168.1.1 для всех http пакетов.

```
iptables -t nat -A PREROUTING -d 1.2.3.4 -p tcp --dport 80 -j DNAT --to 192.168.1.1
```

Для обеспечения получения ответных пакетов с веб-сервера необходимо либо перенаправить ответы на него с помощью DNS сервера, либо использовать SNAT для отображения на адрес сервера.

```
iptables -t nat -A POSTROUTING -d 192.168.1.1 -s 192.168.1.0/24 \
-p tcp --dport 80 -j SNAT --to 192.168.1.250
```

В этой команде предполагается, что адрес NAT компьютера 192.168.1.250, а все пакеты по порту web, направляющиеся по адресу 192.168.1.1, отображаются в адрес NAT сервера 192.168.1.250.

6. Загружаемые расширения



Загружаемые расширения

- Netfilter имеет модульную архитектуру
- Расширения Netfilter реализуются или в виде дополнительных модулей ядра или библиотек iptables
- Все расширения можно разделить на две категории:
 - расширения для составления новых шаблонов фильтрации
 - расширения, создающие новые цели (target)
- При работе с протоколами TCP, UDP и ICMP расширения будут использованы автоматически

Netfilter имеет модульную архитектуру, что обеспечивает хорошую расширяемость функциональности пакетного фильтра.

Расширения Netfilter реализуются или в виде дополнительных модулей ядра или библиотек Iptables.

Расширения Iptables являются разделяемыми библиотеками и находятся в каталоге /lib/iptables или /usr/lib/iptables.

Пример:

```
[root@rhe4 ~]# ls /lib/iptables/
libipt_ah.so          libipt_LOG.so        libipt_SAME.so
libipt_CLASSIFY.so   libipt_mac.so        libipt_sctp.so
libipt_connlimit.so  libipt_mark.so       libipt_SNAT.so
libipt_connmark.so   libipt_MARK.so       libipt_standard.so
libipt_CONNMARK.so   libipt_MASQUERADE.so libipt_state.so
libipt_conntrack.so  libipt_MIRROR.so     libipt_TARPIT.so
libipt_DNAT.so        libipt_multiport.so  libipt_tcpmss.so
libipt_dscp.so        libipt_NETMAP.so     libipt_TCPMSS.so
libipt_DSCP.so        libipt_NOTRACK.so    libipt_tcp.so
libipt_ecn.so         libipt_owner.so      libipt_tos.so
libipt_ECN.so         libipt_physdev.so    libipt_TOS.so
libipt_esp.so         libipt_pkttype.so    libipt_TRACE.so
libipt_helper.so     libipt_realm.so      libipt_ttl.so
libipt_icmp.so        libipt_recent.so     libipt_TTL.so
libipt_iprange.so    libipt_REDIRECT.so   libipt_udp.so
libipt_length.so     libipt_REJECT.so     libipt_ULOG.so
libipt_limit.so      libipt_rpc.so        libipt_unclean.so
```

Модули ядра, расширяющие функциональность Iptables, находятся в каталоге /lib/modules/`uname -r`/kernel/net/ipv4/netfilter.

Пример:

Глава 4. Пакетный фильтр Netfilter (Iptables)

```
root@rhe4 ~]# ls /lib/modules/2.6.9-5.EL/kernel/net/ipv4/netfilter/
arptable_filter.ko      ipt_addrtype.ko      ipt_NETMAP.ko
arp_tables.ko          ipt_ah.ko            ipt_NOTRACK.ko
arpt_mangle.ko         ipt_CLASSIFY.ko      ipt_owner.ko
ip_conntrack_amanda.ko ipt_comment.ko        ipt_physdev.ko
ip_conntrack_ftp.ko    ipt_conntrack.ko     ipt_pkttype.ko
ip_conntrack_irc.ko    ipt_dscp.ko          ipt_realm.ko
ip_conntrack.ko        ipt_DSCP.ko          ipt_recent.ko
ip_conntrack_proto_sctp.ko ipt_ecn.ko            ipt_REDIRECT.ko
ip_conntrack_tftp.ko   ipt_ECN.ko           ipt_REJECT.ko
ip_nat_amanda.ko       ipt_esp.ko           ipt_SAME.ko
ip_nat_ftp.ko          ipt_helper.ko        ipt_sctp.ko
ip_nat_irc.ko          ipt_iprange.ko       ipt_state.ko
ip_nat_snmp_basic.ko   ipt_length.ko        ipt_tcpmss.ko
ip_nat_tftp.ko         ipt_limit.ko         ipt_TCPMSS.ko
ip_queue.ko            ipt_LOG.ko           ipt_tos.ko
iptable_filter.ko      ipt_mac.ko           ipt_TOS.ko
iptable_mangle.ko     ipt_mark.ko          ipt_ttl.ko
iptable_nat.ko         ipt_MARK.ko          ipt_ULOG.ko
iptable_raw.ko         ipt_MASQUERADE.ko
ip_tables.ko           ipt_multiport.ko
```

Все расширения можно разделить на две категории:

1. расширения для составления новых шаблонов фильтрации
2. расширения, создающие новые цели (target).

При работе с протоколами TCP, UDP и ICMP расширения будут использованы автоматически.

При использовании опции `-p` (`--protocol`) команды `iptables` будет загружен модуль расширения для нового шаблона фильтрации. После опции `-p` указывается имя протокола, а модуль будет загружен автоматически.

При необходимости загрузить модуль расширения явно следует использовать опцию `-m` команды `iptables`, после которой следует указать имя модуля.

Для получения помощи по загружаемому расширению следует использовать опцию `-h` (`--help`) после опции `-p` или `-m`, указывающих на модуль, который предполагается загрузить.

Пример:

```
[root@linux1 root]# iptables -p tcp --help
....
TCP v1.2.8 options:
--tcp-flags [!] mask comp      match when TCP flags & mask == comp
                                (Flags: SYN ACK FIN RST URG PSH ALL NONE)
[!] --syn                      match when only SYN flag set
                                (equivalent to --tcp-flags SYN,RST,ACK SYN)
--source-port [!] port[:port]
--sport ...                    match source port(s)
--destination-port [!] port[:port]
--dport ...                    match destination port(s)
--tcp-option [!] number       match if TCP option set
```

Загружаемые расширения для TCP

- Данные расширения загружаются, если указано `-p tcp` (`--protocol tcp`)
 - `--tcp-flags`
 - `--syn`
 - `--sport`
 - `--dport`
 - `--tcp-option`

Данные расширения загружаются, если указано `-p tcp` (`--protocol tcp`).

Опция `--tcp-flags` загружает расширение, позволяющее обрабатывать пакеты определенного типа, например, ACK или SYN.

Опция `--syn` предназначена для обработки пакетов SYN.

Опция `--sport` указывает порт или порты источника пакетов. Порты могут быть указаны именами (`/etc/services`), либо их номерами.

Опция `--dport` предназначена для указания порта или портов назначения пакетов. Порты указываются аналогично `--sport`.

Диапазоны портов удобно задавать двумя именами портов, разделенными символом двоеточие.

Если после имени порта указано двоеточие, то это обозначает данный порт и все порты, номера которых больше.

Если двоеточие стоит перед именем порта, то это обозначает все порты, с номерами меньше, либо равными номеру данного порта.

Опция `--tcp-option` позволяет указать биты, установленные в заголовке TCP пакета.

Пример: отбрасываются TCP пакеты, в которых установлены флажки SYN и ACK

```
[root@linux1]# iptables -A INPUT -p tcp --tcp-flags ALL SYN,ACK -j DROP
```

```
[root@linux1]# iptables -L INPUT
```

```
Chain INPUT (policy ACCEPT)
```

```
target    prot opt source      destination
```

```
DROP      icmp -- linux1     anywhere
```

```
DROP      all  -- !linux1    anywhere
```

```
DROP      tcp  -- anywhere  anywhere      tcp flags:FIN,SYN,RST,PSH,ACK,URG/SYN,ACK
```


Загружаемые расширения для UDP и ICMP

- Расширения UDP загружаются при включении опции `-p udp`, а расширения для протокола ICMP - `-p icmp`
- UDP
 - `--dport`
 - `--sport`
- ICMP
 - `--icmp-type`

Расширения UDP загружаются при включении опции `-p udp`, а расширения для протокола ICMP - `-p icmp`.

Для протокола UDP можно указать опции `--dport` и `--sport`, имеющие такой же смысл, как и для TCP.

Опция `--icmp-type` позволяет указать тип пакетов ICMP, используя либо название типа (например, `host-unreachable`), либо номер типа.

При необходимости указания типа и кода пакета ICMP одновременно, следует указать их через черту, например: `3/3`.

Другие загружаемые расширения

- Все расширения, не относящиеся к протоколам TCP, UDP или ICMP, следует загружать явно, используя ключ `-m`
 - `multiport` — перечисление портов
 - `mac` — анализ MAC адресов
 - `limit` — ограничение количества событий
 - `owner` — анализ идентификаторов пользователей
 - `state` — отслеживание состояний соединения
 - `unclean` — проверки пакетов

Все расширения, не относящиеся к протоколам TCP, UDP или ICMP, следует загружать явно, используя ключ `-m`.

Модуль `multiport` позволяет через запятую перечислять порты в адресе отправителя и/или получателя. Используется вместе с опциями `--sport`, `--dport` и `--ports`

Пример:

```
iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o eth1 -p TCP \  
--sport 1024:65535 -m multiport --dport 80,443 -j ACCEPT
```

Модуль `mac`, загружаемый опцией `-m mac` (`--match mac`), позволяет указывать MAC адреса сетевых карт для цепочек INPUT и FORWARD. Используется только с опцией `--mac-source`.

Опция `--mac-source`, используемая при загруженном модуле `mac` предназначена для указания после нее MAC адреса сетевого интерфейса.

Модуль `limit` используется для ограничения некоторых событий, например, частоты записи в журнал. Может использоваться с опциями `--limit` и `--limit-burst`.

Опция `--limit` с последующим цифровым значением предназначена для указания максимальной средней частоты событий в секунду. Если через черту после числа указано `/second`, `/minute`, `/hour` или `/day`, то это значит, что, соответственно, указано количество событий за секунду, минуту, час или день. По умолчанию - 3 события в час.

Опция `--limit-burst` обозначает максимальное количество подряд идущих событий, после которого предел будет достигнут. По умолчанию используется значение 5.

Пример: согласно приведенному ниже правилу информация о перенаправленных на другой интерфейс пакетах будет записана при достижении следующих параметров. Будет записаны первые пять таких пакетов (счетчик `limit-burst` обнулится), после этого 20 минут (60/3) такие пакеты записываться не будут. Каждые следующие 20 минут, если пакет такого типа приходить не будет, то счетчик `limit-burst` будет увеличиваться на единицу. Если пакет не приходит в течение 100 минут, то счетчик `limit-burst` будет восстановлен в первоначальное значение (в нашем случае – 5)

Глава 4. Пакетный фильтр Netfilter (Iptables)

```
[root@linux1 root]# iptables -A FORWARD -m limit -j LOG
```

Модуль `owner` используется в цепочке `OUTPUT` и позволяет создавать правила в зависимости от имени какого пользователя отправлен пакет.

Допустимые опции модуля `owner`:

- `--uid-owner` – соответствие UID
- `--gid-owner` – соответствие GID
- `--pid-owner` – соответствие PID процесса
- `--sid-owner` – соответствие id сессии
- `--cmd-owner` – соответствие имени команды

Модуль `state` очень полезен для отслеживания состояний сетевых соединений

В модуле `state` существует одна опция – `--state`, которая может отследить 4 состояния

- `NEW` – пакет создает новое соединение
- `ESTABLISHED` – пакет принадлежит установленному соединению
- `RELATED` – пакет имеет отношение к соединению, но не является его частью (например, диагностическое ICMP сообщение или ftp data соединение, сопровождающее ftp сессию)
- `INVALID` – пакет, который не может быть идентифицирован

Модуль `unclean` позволяет осуществлять различные случайные проверки целостности пакета. `Unclean` – экспериментальный модуль и его не следует использовать на промышленных системах.

Расширения iptables для действий (target)

- Модуль LOG предназначен для журналирования информации о пакетах, удовлетворяющих правилу
- Модуль REJECT позволяет использовать действие (цель) REJECT в правилах

Модуль LOG предназначен для журналирования информации о пакетах, удовлетворяющих правилу. Может использоваться с опциями `--log-level`, `--log-prefix`, `--log-tcp-sequence`, `--log-tcp-options`, `--log-ip-options`.

Опция `--log-level` предназначена для указания уровня (приоритет) сообщения для журналирования с помощью службы syslogd (категория (facility) сообщений – kern)

Опция `--log-prefix` позволяет установить строку префикса длиной до 29 символов, которая будет выводиться перед каждым сообщением.

Для записи номеров TCP пакетов следует использовать опцию `--log-tcp-sequence`.

Опция `--log-tcp-options` позволяет записывать опции из заголовков TCP пакетов.

Опция `--log-ip-options` нужна для записи опций из заголовков IP пакетов.

Модуль REJECT позволяет использовать действие (цель) REJECT в правилах. Это действие действует аналогично DROP, но при этом производится посылка сообщения ICMP 'port unreachable'.

Цель REJECT может использоваться только в цепях INPUT, FORWARD и OUTPUT.

Сообщение ICMP 'port unreachable' не посылается в случаях (RFC1122):

- фильтруемый пакет имеет в начале сообщение об ошибке ICMP или неизвестный тип пакета ICMP;
- фильтруемый фрагмент пакета не является первым;
- для данного адресата производится посылка слишком большого количества пакетов ICMP с сообщениями об ошибках.

Цель REJECT можно использовать с опцией `--reject-with`, позволяющей указывать

Глава 4. Пакетный фильтр Netfilter (Iptables)

тип ICMP сообщения об ошибке.

7. Пользовательские цепочки правил



Пользовательские цепочки правил

- Пользователь может создавать свои цепочки правил, отличные от встроенных (INPUT, OUTPUT, FORWARD)
- Пакет попадает на обработку в пользовательскую цепочку тогда, когда она указана в качестве действия (цели) в каком-либо правиле, которому удовлетворил данный пакет
- Если пакет проходит всю цепь, определенную пользователем, и не встречает ни одного правила, которому он подходит, он возвращается для обработки в следующее правило той цепи, которую он покинул

Пользователь может создавать свои цепочки правил, отличные от встроенных (INPUT, OUTPUT, FORWARD).

Создание своих цепочек позволяет упростить настройку сложных и/или иерархических правил фильтрации.

Пример:

```
[root@nb ~]# iptables -N ssh_access
[root@nb ~]# iptables -I ssh_access -s 1.2.3.4/32 -j ACCEPT
[root@nb ~]# iptables -I ssh_access -s 9.8.7.6/32 -j ACCEPT
[root@nb ~]# iptables -I ssh_access -s 192.168.1.0/24 -j DROP
[root@nb ~]# iptables -A ssh_access -j DROP
[root@nb ~]# iptables -I INPUT -p tcp --dport 22 -j ssh_access
[root@nb ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ssh_access tcp  --  0.0.0.0/0              0.0.0.0/0              tcp dpt:22
...
```

```
Chain ssh_access (1 references)
target     prot opt source                destination
ACCEPT     all  --  127.0.0.1              0.0.0.0/0
DROP       all  --  192.168.1.0/24         0.0.0.0/0
ACCEPT     all  --  9.8.7.6                0.0.0.0/0
ACCEPT     all  --  1.2.3.4                0.0.0.0/0
DROP       all  --  0.0.0.0/0              0.0.0.0/0
```

Примечание: В примере выше создается новая цепочка правил с названием `ssh_access`. Затем цепочка заполняется правилами фильтрации. Обратите внимание на то, что последнее правило в цепочке добавляется с опцией `-A`. Затем в цепочке `INPUT` создается правило для перехода в цепочку при обращении на порт 22.

Глава 4. Пакетный фильтр Netfilter (Iptables)

Пакет попадает на обработку в пользовательскую цепочку тогда, когда она указана в качестве действия (цели) в каком-либо правиле, которому удовлетворил данный пакет.

Если пакет проходит всю цепь, определенную пользователем, и не встречает ни одного правила, которому он подходит, он возвращается для обработки в следующее правило той цепи, которую он покинул.

8. Специальные действия (цели)

Специальные действия (цели)



- Цель RETURN позволяет немедленно покидать цепочку
- Цель QUEUE ставит пакеты в очередь для обработки на пользовательском уровне

Цель RETURN позволяет немедленно покидать цепочку. При этом, для встроенной цепочки управление передается политике, а для пользовательской следующему правилу в цепочке, из которой был произведен переход.

Цель QUEUE ставит пакеты в очередь для обработки на пользовательском уровне (user side), а не на уровне ядра (kernel side).

Если в пользовательском пространстве нет процесса, который ожидает данный сетевой пакет, то пакет отбрасывается

Библиотека libipq предоставляет API для создания таких приложений.

9. Защита сервера Linux с помощью межсетевого экрана

Защита сервера с помощью брандмауэра



- Для защиты сервера Linux вы можете применить:
 - Локальный брандмауэр
 - Установить сервер в демилитаризованной зоне или внутренней сети, созданной сетевым брандмауэром
- Дополнительно можно контролировать трафик с помощью прокси сервера

Устанавливая сервер для обслуживания клиентских запросов обдумайте вариант защиты сетевых соединений к этому серверу.

Вы можете установить сервер напрямую подключенным к интернету, тогда для его защиты вам необходимо применять локальный брандмауэр, основанный на Netfilter.

Лучшим, с точки зрения безопасности, будет установить на границе вашей сети специализированный брандмауэр для контроля сетевого трафика всей сети. Такие системы могут, в том числе, использовать в качестве основы Linux. При этом установка централизованного решения не отменяет настройку локальной защиты.

Преимущества специализированных решений:

- Поддержка продвинутых механизмов маршрутизации.
- Настройка ориентированная на обеспечение безопасности.
- Продвинутая поддержка VPN.
- В некоторых случаях контентная фильтрация.

Одной из мер защиты может рассматриваться применение прокси серверов для фильтрации и оптимизации трафика.



Служба netfilter-persistent

- Для управления правилами фильтрации трафика можно применить службу netfilter-persistent
- Правила находятся в файлах `/etc/iptables/rules.v4` и `/etc/iptables/rules.v6`
- Формат файла такой же как вывод команды `iptables-save`

В Debian подобных системах в качестве базового решения для фильтрации трафика можно применить службу netfilter-persistent.

Это простая служба, которая при своем старте загружает правила фильтрации трафика из файла.

Конфигурационный файл `/etc/iptables/rules.v4` содержит правила фильтрации для IPv4.

Формат конфигурационного файла такой же как и вывод команды `iptables-save`.

Пример: Сначала очистим все правила добавим свои:

```
root@rl:~# iptables -F
root@rl:~# iptables -I INPUT -i lo -j ACCEPT
root@rl:~# iptables -t nat -I POSTROUTING -s 10.0.100.0/24 -o enp0s3 -j
MASQUERADE
```

Далее сохраним правила:

```
root@rl:~# /etc/init.d/netfilter-persistent save
Saving netfilter rules...run-parts: executing
/usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
done.
```

Проверим результат:

```
root@rl:~# cat /etc/iptables/rules.v4
# Generated by iptables-save v1.8.9 (nf_tables) on Sat Oct 26 21:48:28 2024
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
```

Глава 4. Пакетный фильтр Netfilter (Iptables)

```
COMMIT
# Completed on Sat Oct 26 21:48:28 2024
# Generated by iptables-save v1.8.9 (nf_tables) on Sat Oct 26 21:48:28 2024
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 10.0.100.0/24 -o enp0s3 -j MASQUERADE
COMMIT
# Completed on Sat Oct 26 21:48:28 2024
```

10. Служба firewalld

Служба firewalld



- Особенности демона firewalld

- Динамический фильтр
- Сетевые интерфейсы поделены на зоны
- Поддержка IPv4 и IPv6, а также Ethernet мостов и IP Set
- Разделение на активную (runtime) и постоянную (permanent) конфигурацию
- Взаимодействие с другими сервисами и приложениями, через D-Bus
- Авторизация пользователей через polkit

Имеются несколько сценариев, в которых применение службы iptables является нецелесообразным. Например: вы хотите применять разные правила фильтрации трафика для разных интерфейсов или разных сетей. Или возможно вы захотите делегировать полномочия на управление брандмауэром. Для решения этих задач предназначена служба firewalld, которая управляет правилами Netfilter.

Основные характеристики службы firewalld:

- Динамический управление межсетевым экраном.
- Сетевые интерфейсы поделены на зоны. Зоны определяют наборы применяемых правил фильтрации.
- Поддержка IPv4 и IPv6, а также Ethernet мостов и IP Set.
- Разделение на активную (runtime) и постоянную (permanent) конфигурацию.
- Взаимодействие с другими сервисами и приложениями, через D-Bus.
- Авторизация пользователей через polkit.

Управление firewalld

- Конфигурация находится в XML файлах в каталогах /etc/firewalld и /usr/lib/firewalld
- Основные утилиты управления:
 - firewall-config графическая утилита
 - firewall-cmd интерфейс управления через командную строку
 - firewall-offline-cmd командный интерфейс для управления постоянной конфигурацией

Настройки демона firewalld находятся в XML файлах, за исключением основного конфигурационного файла /etc/firewalld/firewalld.conf.

Пример: Просмотр состояния демона firewalld, и его запуск.

```
[root@sl0 ~]# firewall-cmd --state
not running
[root@sl0 ~]# systemctl enable firewalld
[root@sl0 ~]# systemctl start firewalld
[root@sl0 ~]# firewall-cmd --state
running
```

Пример: получение информации о зоне. И определение интерфейса в зону.

```
[root@sl0 ~]# firewall-cmd --get-default-zone
public
[root@sl0 ~]# firewall-cmd --zone=public --list-services
ssh dhcpv6-client
[root@sl0 ~]# firewall-cmd --get-active-zones
[root@sl0 ~]# firewall-cmd --get-zone-of-interface=eth0
no zone
[root@sl0 ~]# firewall-cmd --add-interface=eth0 --zone=home
The interface is under control of NetworkManager, setting zone to 'home'.
success
[root@sl0 ~]# firewall-cmd --get-zone-of-interface=eth0
no zone
[root@sl0 ~]# systemctl restart NetworkManager
[root@sl0 ~]# firewall-cmd --get-zone-of-interface=eth0
home
[root@sl0 ~]# firewall-cmd --add-interface=eth0 --permanent --zone=work
The interface is under control of NetworkManager, setting zone to 'work'.
success
[root@sl0 ~]# firewall-cmd --get-zone-of-interface=eth0
work
[root@sl0 ~]# firewall-cmd --list-services --zone=work
```

Глава 4. Пакетный фильтр Netfilter (Iptables)

```
ssh dhcpv6-client  
[root@sl0 ~]# firewall-cmd --list-ports --zone=work
```

Пример: добавление порта TCP в нужную зону.

```
[root@sl0 ~]# firewall-cmd --add-port=22222/tcp --permanent --zone=work  
success  
[root@sl0 ~]# firewall-cmd --list-ports --zone=work  
  
[root@sl0 ~]# firewall-cmd --add-port=22222/tcp --zone=work  
success  
[root@sl0 ~]# firewall-cmd --list-ports --zone=work  
22222/tcp  
[root@sl0 ~]# firewall-cmd --list-ports --zone=work --permanent  
22222/tcp
```

Пример: создание политики взаимодействия зон

(<https://firewalld.org/documentation/concepts.html>)

```
[root@sl0 ~]# firewall-cmd --permanent --new-policy=wrktoext  
[root@sl0 ~]# firewall-cmd --permanent --policy=wrktoext --add-ingress-  
zone=work  
[root@sl0 ~]# firewall-cmd --permanent --policy=wrktoext --add-egress-  
zone=external  
[root@sl0 ~]# firewall-cmd --permanent --policy=wrktoext --set-target=ACCEPT
```