Создание таблицы файервола на основе iptables

1 Введение

Прежде, чем пытаться настраимать правила, надо иметь понятия, используемые при настройке файерврла:

Правило — состоит из критерия, действия и счетчика. Если пакет соответствует критерию, к нему применяется действие, и он учитывается счетчиком. Критерия может и не быть — тогда неявно предполагается критерий «все пакеты». Указывать действие тоже не обязательно — в отсутствие действия правило будет работать только как счетчик. Правила для каждой цепочки срабатывают в порядке их следования, поэтому порядок важен.

- Критерий — логическое выражение, анализирующее свойства пакета и/или соединения и определяющее, подпадает ли данный конкретный пакет под действие текущего правила. Критерии соединяются логическим «И».
- Действие — описание действия, которое нужно проделать с пакетом и/или соединением в том случае, если они подпадают под действие этого правила. О действиях более подробно будет рассказано ниже.
- Счетчик — компонент правила, обеспечивающий учет количества пакетов, которые попали под критерий данного правила. Также счетчик учитывает суммарный объем таких пакетов в байтах.

Цепочка — упорядоченная последовательность правил. Цепочки можно разделить на пользовательские и базовые.

- Базовая цепочка — цепочка, создаваемая по умолчанию при инициализации таблицы. Каждый пакет, в зависимости от того, предназначен ли он самому хосту, сгенерирован им или является транзитным, должен пройти положенный ему набор базовых цепочек различных таблиц. Кроме того, базовая цепочка отличается от пользовательской наличием «действия по умолчанию» (default policy). Это действие применяется к тем пакетам, которые не были обработаны другими правилами этой цепочки и вызванных из нее цепочек. Имена базовых цепочек всегда записываются в верхнем регистре (PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING).

- Пользовательская цепочка — цепочка, созданная пользователем. Может использоваться только в пределах своей таблицы. Рекомендуется не использовать для таких цепочек имена в верхнем регистре, чтобы избежать путаницы с базовыми цепочками и встроенными действиями.

Таблица — совокупность базовых и пользовательских цепочек, объединенных общим функциональным назначением. Имена таблиц (как и модулей критериев) записываются в нижнем регистре, так как в принципе не могут конфликтовать с именами пользовательских цепочек. При вызове команды iptables таблица указывается в формате -t имя_таблицы. При отсутствии явного указания, используется таблица filter.

1.1 Цепочки

Существует 5 типов стандартных цепочек, встроенных в систему:

- PREROUTING — для изначальной обработки входящих пакетов.
- INPUT — для входящих пакетов адресованных непосредственно локальному процессу (клиенту или серверу).
- FORWARD — для входящих пакетов перенаправленных на выход (заметьте, что перенаправляемые пакеты проходят сначала цепь PREROUTING, затем FORWARD и POSTROUTING).
- OUTPUT — для пакетов генерируемых локальными процессами.
- POSTROUTING — для окончательной обработки исходящих пакетов.

Также можно создавать и уничтожать собственные цепочки при помощи утилиты iptables.

1.2 Таблицы

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.

Цепочки с одинаковым названием, но в разных таблицах — совершенно независимые объекты. Например, raw PREROUTING и mangle PREROUTING обычно содержат разный набор правил; пакеты сначала проходят через цепочку raw PREROUTING, а потом через mangle PREROUTING.

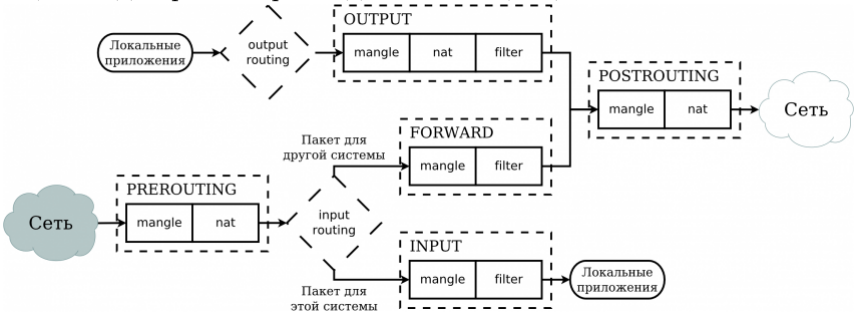
1.3 Состояния

В системе netfilter, каждый пакет проходящий через механизм определения состояний, может иметь одно из четырёх возможных состояний:

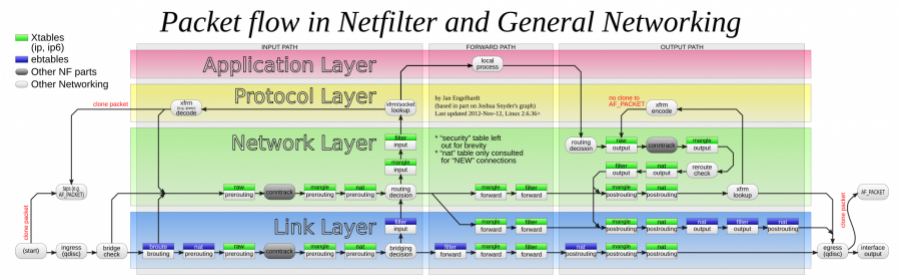
- NEW — пакет открывает новый сеанс. Классический пример — пакет TCP с флагом SYN.
- ESTABLISHED — пакет является частью уже существующего сеанса.
- RELATED — пакет открывает новый сеанс, связанный с уже открытым сеансом. Например, во время сеанса пассивного FTP, клиент подсоединяется к порту 21 сервера, 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).
- INVALID — все прочие пакеты.

1.4 Диаграммы

Урощённая диаграмма прохождения таблиц и цепочек:



Детальная диаграмма по уровням OSI:



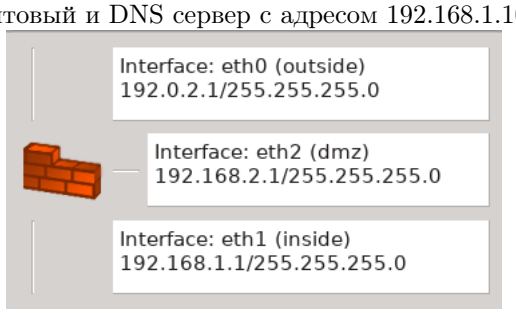
2 Примеры

Пример со статическими адресами

Имеются три интерфейса:

- eth0 - внешний с адресом 192.0.2.1/24
- eth1 - внутренний с адресом 192.168.1.1/24
- eth3 - DMZ с адресом 192.168.2.1/24

В DMZ-зоне имеется почтовый сервер с адресом 192.168.2.10. Во внутренй зоне имеется почтовый и DNS сервер с адресом 192.168.1.10.



Необходимо предоставить доступ из внутренней сети наружу, с наружной сети доступ к почтовому серверу, находящемуся в DMZ, к внутреннему DNS-серверу из DMZ, а также предоставить возможность соединяться почтовым серверам, находящимся в DMZ и во внутренней сети.

	Source	Destination	Service	Interface	Direction	Action	Time	Options	Comment
0	Test net-192.168.1.0 net-192.168.2.0	Any	Any	outside	Inboun	Deny	Any	log (optic	anti spoofing rule
1	Any	Any	Any	loopback	Both	Accept	Any		
2	net-192.168.1.0	Test	ssh	Any	Both	Accept	Any		SSH Access to firewall is permitted
3	Test	internal server	DNS	Any	Both	Accept	Any		Firewall uses one of the machines
4	Any	Test	Any	Any	Both	Deny	Any	log (optic	All other attempts to connect to the firewall are denied and logged
5	Any	Any	auth	Any	Both	Reject	Any		Quickly reject attempts to connect
6	Any	server on dmz	smtp	Any	Both	Accept	Any		Mail relay on DMZ can accept
7	server on dmz	internal server	smtp	Any	Both	Accept	Any		this rule permits a mail relay
8	server on dmz	net-192.168.1.0	DNS smtp	Any	Both	Accept	Any		Mail relay needs DNS and can connect to mail servers on the Internet
9	net-192.168.2.0	net-192.168.1.0	Any	Any	Both	Deny	Any	log	All other access from DMZ to
10	net-192.168.1.0	Any	Any	Any	Both	Accept	Any		This permits access from internal net
11	Any	Any	Any	Any	Both	Deny	Any	log	

Кроме самих правил также понадобится построить NAT-таблицу:

	Original Src	Original Dst	Original Srv	Translated Src	Translated Dst	Translated Srv	Interface In	Interface Out
0	net-192.168.2.0	net-192.168.1.0	Any	Original	Original	Original	Auto	Auto
1	net-192.168.1.0 net-192.168.2.0	Any	Any	outside	Original	Original	Auto	Auto
2	Any	outside	Any	Original	server on dmz	Original	Auto	Auto

По правилам iptables, первыми правилами записываются NAT-правила, а затем следуют основные правила фильтров:

```
# ================ Reset all rules
#
iptables -P OUTPUT  DROP
iptables -P INPUT   DROP
iptables -P FORWARD DROP
```

```
#
# ================ Table 'filter'
# accept established sessions
iptables -A INPUT   -m state --state ESTABLISHED,RELATED -j
    ACCEPT
iptables -A OUTPUT  -m state --state ESTABLISHED,RELATED -j
    ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j
    ACCEPT
#
# ================ Table 'nat', rule set NAT
#
# Rule 0 (NAT)
#
echo "Rule 0 (NAT)"
#
# no need to translate
# between DMZ and
# internal net
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -d
    192.168.1.0/24 -j ACCEPT
iptables -t nat -A PREROUTING -s 192.168.2.0/24 -d
    192.168.1.0/24 -j ACCEPT
#
# Rule 1 (NAT)
#
echo "Rule 1 (NAT)"
#
# Translate source address
# for outgoing connections
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.1.0/24 -j
    SNAT --to-source 192.0.2.1
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.2.0/24 -j
    SNAT --to-source 192.0.2.1
#
# Rule 2 (NAT)
#
echo "Rule 2 (NAT)"
#
iptables -t nat -A PREROUTING -d 192.0.2.1 -j DNAT --to-
    destination 192.168.2.10
```

Первым правилом фильтра делается правило для anti-spoofing. Следующим правилом надо создать полное разрешение для интерфейса loopback. Последнее правило запрещяет всё:

```
# ================ Table 'filter', rule set Policy
#
# Rule 0 (eth0) #
echo "Rule 0 (eth0)"
#
# anti spoofing rule
iptables -N In_RULE_0
iptables -A INPUT -i eth0 -s 192.0.2.1 -m state --state NEW
    -j In_RULE_0
```

```
iptables -A INPUT -i eth0 -s 192.168.1.1 -m state --state
    NEW -j In_RULE_0
iptables -A INPUT -i eth0 -s 192.168.2.1 -m state --state
    NEW -j In_RULE_0
iptables -A INPUT -i eth0 -s 192.168.1.0/24 -m state --state
     NEW -j In_RULE_0
iptables -A INPUT -i eth0 -s 192.168.2.0/24 -m state --state
     NEW -j In_RULE_0
iptables -A FORWARD -i eth0 -s 192.0.2.1 -m state --state
    NEW -j In_RULE_0
iptables -A FORWARD -i eth0 -s 192.168.1.1 -m state --state
    NEW -j In_RULE_0
iptables -A FORWARD -i eth0 -s 192.168.2.1 -m state --state
    NEW -j In_RULE_0
iptables -A FORWARD -i eth0 -s 192.168.1.0/24 -m state --
    state NEW -j In_RULE_0
iptables -A FORWARD -i eth0 -s 192.168.2.0/24 -m state --
    state NEW -j In_RULE_0
iptables -A In_RULE_0 -j LOG --log-level info --log-prefix "
    RULE 0 -- DENY "
iptables -A In_RULE_0 -j DROP
#
# Rule 1 (lo) #
echo "Rule 1 (lo)"
#
iptables -A INPUT -i lo -m state --state NEW -j ACCEPT
iptables -A OUTPUT -o lo -m state --state NEW -j ACCEPT
#
# Rule 2 (global)
#
echo "Rule 2 (global)"
#
# SSH Access to firewall is permitted
# only from internal network
iptables -A INPUT -p tcp -m tcp -s 192.168.1.0/24 --dport 22
     -m state --state NEW -j ACCEPT
#
# Rule 3 (global)
#
echo "Rule 3 (global)"
#
# Firewall uses one of the machines
# on internal network for DNS
iptables -A OUTPUT -p tcp -m tcp -d 192.168.1.10 --dport 53
    -m state --state NEW -j ACCEPT
iptables -A OUTPUT -p udp -m udp -d 192.168.1.10 --dport 53
    -m state --state NEW -j ACCEPT
#
# Rule 4 (global)
#
echo "Rule 4 (global)"
#
# All other attempts to connect to
# the firewall are denied and logged
```

```
iptables -N RULE_4
iptables -A OUTPUT -d 192.0.2.1 -m state --state NEW -j
    RULE_4
iptables -A OUTPUT -d 192.168.1.1 -m state --state NEW -j
    RULE_4
iptables -A OUTPUT -d 192.168.2.1 -m state --state NEW -j
    RULE_4
iptables -A INPUT -m state --state NEW -j RULE_4
iptables -A RULE_4 -j LOG --log-level info --log-prefix "
    RULE 4 -- DENY "
iptables -A RULE_4 -j DROP
#
# Rule 5 (global)
#
echo "Rule 5 (global)"
#
# Quickly reject attempts to connect
# to ident server to avoid SMTP delays
iptables -A OUTPUT -p tcp -m tcp --dport 113 -j REJECT
iptables -A INPUT -p tcp -m tcp --dport 113 -j REJECT
iptables -A FORWARD -p tcp -m tcp --dport 113 -j REJECT
#
# Rule 6 (global)
#
echo "Rule 6 (global)"
#
# Mail relay on DMZ can accept
# connections from hosts on the
# Internet
iptables -A OUTPUT -p tcp -m tcp -d 192.168.2.10 --dport 25
    -m state --state NEW -j ACCEPT
iptables -A FORWARD -p tcp -m tcp -d 192.168.2.10 --dport 25
     -m state --state NEW -j ACCEPT
#
# Rule 7 (global)
#
echo "Rule 7 (global)"
#
# this rule permits a mail relay
# located on DMZ to connect
# to internal mail server
iptables -A FORWARD -p tcp -m tcp -s 192.168.2.10 -d
    192.168.1.10 --dport 25 -m state --state NEW -j ACCEPT
#
# Rule 8 (global)
#
echo "Rule 8 (global)"
#
# Mail relay needs DNS and can
# connect to mail servers on the
# Internet
iptables -A INPUT -p tcp -m tcp -m multiport -s 192.168.2.10
     -d ! 192.168.1.0/24 --dports 53,25 -m state --state NEW
    -j ACCEPT
```

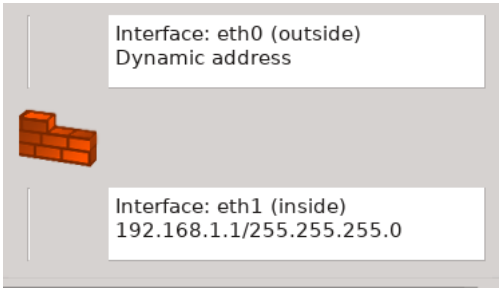
```
iptables -A INPUT -p udp -m udp -s 192.168.2.10 -d !
    192.168.1.0/24 --dport 53 -m state --state NEW -j ACCEPT
iptables -A FORWARD -p tcp -m tcp -m multiport -s
    192.168.2.10 -d ! 192.168.1.0/24 --dports 53,25 -m state
    --state NEW -j ACCEPT
iptables -A FORWARD -p udp -m udp -s 192.168.2.10 -d !
    192.168.1.0/24 --dport 53 -m state --state NEW -j ACCEPT
#
# Rule 9 (global)
#
echo "Rule 9 (global)"
#
# All other access from DMZ to
# internal net is denied
iptables -N RULE_9
iptables -A OUTPUT -s 192.168.2.0/24 -d 192.168.1.0/24 -j
    RULE_9
iptables -A INPUT -s 192.168.2.0/24 -d 192.168.1.0/24 -j
    RULE_9
iptables -A FORWARD -s 192.168.2.0/24 -d 192.168.1.0/24 -j
    RULE_9
iptables -A RULE_9 -j LOG --log-level info --log-prefix "
    RULE 9 -- DENY "
iptables -A RULE_9 -j DROP
#
# Rule 10 (global)
#
echo "Rule 10 (global)"
#
# This permits access from internal net
# to the Internet and DMZ
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -j
    ACCEPT
iptables -A OUTPUT -s 192.168.1.0/24 -m state --state NEW -j
     ACCEPT
iptables -A FORWARD -s 192.168.1.0/24 -m state --state NEW -
    j ACCEPT
#
# Rule 11 (global)
#
echo "Rule 11 (global)"
#
iptables -N RULE_11
iptables -A OUTPUT -j RULE_11
iptables -A INPUT -j RULE_11
iptables -A FORWARD -j RULE_11
iptables -A RULE_11 -j LOG --log-level info --log-prefix "
    RULE 11 -- DENY "
iptables -A RULE_11 -j DROP
```

Пример с динамическим адресом

Имеются два интерфейса:

- eth0 - внешний с динамическим адресом, получаемым от провайдера
- eth1 - внутренний с адресом 192.168.1.1/24

В данном примере рассматривается доступ из локальной сети 192.168.1.0 в интернет.



Правила файервола будут следующие:

	Source	Destination	Service	Interface	Direction	Action	Time	Options	Comment
0	Test net-192.168.1.0	Any	Any	outside	Inboun	Deny	Any	log	anti spoofing rule
1	Any	Any	Any	loopback	Both	Accept	Any		
2	net-192.168.1.0	Test	ssh	Any	Both	Accept	Any		SSH Access to firewall
3	Test	net-192.168.1.0	DNS	Any	Both	Accept	Any		Firewall uses one of th
4	Any	Test	Any	Any	Both	Deny	Any	log	All other attempts to c
5	net-192.168.1.0	Any	Any	Any	Both	Accept	Any		
6	Any	Any	Any	Any	Both	Deny	Any	log	

Правилом NAT необходимо предоставить трансляцию внутренних адресов сети в адрес внешнего интерфейса. В связи с тем, что адрес внешнего интерфейса не постоянный, надо в правиле использовать MASQERADING.

	Original Src	Original Dst	Original Srv	Translated Src	Translated Dst	Translated Srv	Interface In	Interface Out	Action
0	net-192.168.1.0	Any	Any	outside	Original	Original	Auto	Auto	Trar

Так как исходно адрес интерфейса eth0 неизвестен, его необходимо определить и добавить в качестве переменной “i_eth0_list”.

Это можно сделать, например, следующим образом:

```
dev="eth0"
name="i_eth0"
af="-4"
L=$(ip $af addr show dev $dev | sed -n '/inet/{s!.*inet6*
    !!;s!/.*!!p}' | sed 's/peer.*//')
test -z "$L" && {
  eval "$name=''"
  return
}
eval "${name}\_list=\"$L\""
```

В результате получим следующее:

```
# ================ Reset all rules
#
iptables -P OUTPUT  DROP
iptables -P INPUT   DROP
iptables -P FORWARD DROP
#
# ================ Table 'filter'
```

```
# accept established sessions
iptables -A INPUT   -m state --state ESTABLISHED,RELATED -j
    ACCEPT
iptables -A OUTPUT  -m state --state ESTABLISHED,RELATED -j
    ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j
    ACCEPT
#
# ================ Table 'nat',  rule set NAT
#
# Rule 0 (NAT)
#
echo "Rule 0 (NAT)"
#
iptables -t nat -A POSTROUTING -o eth0   -s 192.168.1.0/24
    -j MASQUERADE
#
# ================ Table 'filter', rule set Policy
#
# Rule 0 (eth0)
#
echo "Rule 0 (eth0)"
#
# anti spoofing rule
iptables -N In_RULE_0
for i_eth0 in $i_eth0_list
do
  test -n "$i_eth0" && iptables -A INPUT -i eth0   -s
      $i_eth0   -j In_RULE_0
done
iptables -A INPUT -i eth0   -s 192.168.1.1   -j In_RULE_0
iptables -A INPUT -i eth0   -s 192.168.1.0/24   -j In_RULE_0
for i_eth0 in $i_eth0_list
do
  test -n "$i_eth0" && iptables -A FORWARD -i eth0   -s
      $i_eth0   -j In_RULE_0
done
iptables -A FORWARD -i eth0   -s 192.168.1.1   -j In_RULE_0
iptables -A FORWARD -i eth0   -s 192.168.1.0/24   -j
    In_RULE_0
iptables -A In_RULE_0  -j LOG  --log-level info --log-prefix
     "RULE 0 -- DENY "
iptables -A In_RULE_0  -j DROP
#
# Rule 1 (lo)
#       echo "Rule 1 (lo)"
#
iptables -A INPUT -i lo   -m state --state NEW  -j ACCEPT
iptables -A OUTPUT -o lo   -m state --state NEW  -j ACCEPT
#
# Rule 2 (global)
#
echo "Rule 2 (global)"
#
```

```
# SSH Access to firewall is permitted
# only from internal network
iptables -A INPUT -p tcp -m tcp  -s 192.168.1.0/24   --dport
    22  -m state - -state NEW  -j ACCEPT
#
# Rule 3 (global)
#
echo "Rule 3 (global)"
#
# Firewall uses one of the machines
# on internal network for DNS
iptables -A OUTPUT -p tcp -m tcp  -d 192.168.1.0/24   --
   dport 53  -m state --state NEW  -j ACCEPT
iptables -A OUTPUT -p udp -m udp  -d 192.168.1.0/24   --
   dport 53  -m state --state NEW  -j ACCEPT
#
# Rule 4 (global)
#
echo "Rule 4 (global)"
#
# All other attempts to connect to
# the firewall are denied and logged
iptables -N RULE_4
for i_eth0 in $i_eth0_list
do
  test -n "$i_eth0" && iptables -A OUTPUT  -d $i_eth0   -j
     RULE_4
done
iptables -A OUTPUT  -d 192.168.1.1   -j RULE_4
iptables -A INPUT  -j RULE_4
iptables -A RULE_4  -j LOG  --log-level info --log-prefix "
   RULE 4 -- DENY "
iptables -A RULE_4  -j DROP
#
# Rule 5 (global)
#
echo "Rule 5 (global)"
#
iptables -A INPUT  -s 192.168.1.0/24   -m state --state NEW
    -j ACCEPT
iptables -A OUTPUT  -s 192.168.1.0/24   -m state --state NEW
     -j ACCEPT
iptables -A FORWARD  -s 192.168.1.0/24   -m state --state
   NEW  -j ACCEPT
#
# Rule 6 (global)
#      echo "Rule 6 (global)"
#
iptables -N RULE_6
iptables -A OUTPUT  -j RULE_6
iptables -A INPUT  -j RULE_6
iptables -A FORWARD  -j RULE_6
iptables -A RULE_6  -j LOG  --log-level info --log-prefix "
   RULE 6 -- DENY "
```

```
iptables -A RULE_6  -j DROP
```