



Физически факултет

Филтриране на мрежови трафик с Iptables – Linux Firewall

Изготвил Стефано Огнянски

Въведение

Iptables е изключително гъвкавата помощна програма за защитна стена, създадена за Linux дистрибуции. Използва верига от правила за блокиране и разрешаване на трафик. Когато е наличен опит за остановяване на връзка с нашата система, Iptables проверява дали входящата връзка отговаря на правилата. Ако не бъдат открити такива, се остановява връзка по подразбиране.

Почти всяка линукс дистрибуция съдържа в себе си Iptables. Съществуват и алтернативни програми с графична среда, въпрос на предпочитания.

Видове вериги

Вход – Тази верига се използва за контрол на входящите връзки. Например, ако някой се опита да влезе чрез SSH в нашия компютър, iptables ще съпостави IP адреса и порта с правило във входящата верига.

Препращащи – Използват се за входящи връзки, които не са доставени локално. Подобно на рутер – данните винаги се изпращат до него, но рядко са предназначени за самия рутер. Тази верига се използва, в случай че искаме да извършим някаква маршрутизация, NAT или нещо друго в системата, което изисква препращане.

Можем да проверим дали нашата система използва препращаща верига със следната команда:

```
root@doomst:~# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
root@doomst:~#
```

1 Forward chain

На снимката от терминала виждаме, че нямаме създадени правила за филтриране на пакети.

Изходящи – използват се за изходящи връзки. Например, ако искаме да направим пинг към определен сайт, iptables ще провери правилата относно пинга към този сайт, след което ще прецени дали да позволи опита за свързване.

Когато пингуваме външен хост, освен че трябва да преминем през изходящата верига, за да получим обратно пакети те трябва да минат и през входящата. Голяма част от протоколите изискват двупосочна комуникация например SSH.

Правила по подразбиране

Преди за създадем специфични правила, трябва да преценим какво да е поведението по подразбиране на трите вериги. С други думи, как iptables да реагира когато връзката не съвпада с нито едно правило..

За да проверим как са конфигурирани веригите използваме командата:

```
DOOMST> iptables -L
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
DOOMST>
```

2 default policy

Виждаме че трите типа връзки са позволени. В случай че не са и искаме да ги разрешим можем да използваме следните команди:

```
DOOMST> iptables --policy INPUT ACCEPT
DOOMST> iptables --policy OUTPUT ACCEPT
DOOMST> iptables --policy FORWARD ACCEPT
```

3 Разрешаване на трафик

Обратното:

```
DOOMST> iptables --policy INPUT DROP
DOOMST> iptables --policy OUTPUT DROP
DOOMST> iptables --policy FORWARD DROP
```

4 Забраняване на трафик

Ако конфигурираме сървър, който желаем да се свързва само с определени IP-та, с оглед на сигурността е най-добре да се забранят всички връзки и след това да разрешим само тези, които биха били необходими.

След като сме конфигурирали правила по подразбиране, може да започваме да добавяме такива за определени IP адреси или портове. Ще разгледаме най-базовите отговори на заявки за връзка:

Accept – приемане на връзката.

Drop – игнориране на връзката. Използва се когато не искаме източника на заявката да знае за съществуването на нашата система.

Reject – не позволява връзката, но изпраща съобщение със грешка. Използва се когато не искаме да позволим дадена връзка, но искаме източника да знае, че е блокиран от защитната стена.

За да видим разликата между тези три правила, за всеки от случаите, ще отправим ping към Linux виртуална машина от Windows:

🚦 DROP

```
C:\Users\DOOMST6>ping 192.168.0.105
Pinging 192.168.0.105 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.105:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

5 Drop connection

ACCEPT

```
Pinging 192.168.0.105 with 32 bytes of data:
Reply from 192.168.0.105: bytes=32 time<1ms TTL=64
Reply from 192.168.0.105: bytes=32 time=1ms TTL=64
Reply from 192.168.0.105: bytes=32 time=1ms TTL=64
Reply from 192.168.0.105: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.0.105:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

6 *Allow the connection*

REJECT

```
C:\Users\DOOMST6>ping 192.168.0.105
inet 127.0.0.1 netmask 255.0.0.0
Pinging 192.168.0.105 with 32 bytes of data:
Reply from 192.168.0.105: Destination port unreachable.
Reply from 192.168.0.105: Destination port unreachable.
Reply from 192.168.0.105: Destination port unreachable.
Reply from 192.168.0.105: Destination port unreachable.
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
Ping statistics for 192.168.0.105:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    times = 0, INPUT = 8, 192.168.0.102 -i REJECT
```

7 *REJECT*

Някой основни параметри

-p, --protocol

С него задаваме протокола на правилото или на пакета който искаме да бъде проверен.

-s, --source *address[/mask]*

Спецификация на източника. Адресът може да бъде име на мрежата, име на хоста, IP адрес, IP адрес с маска

-i, --in-interface *name*

Задава името на интерфейса, през който даден пакет да бъде е приет.

-o, -- out-interface

Задава името на интерфейса ,по който да бъде изпратен пакета.

Разрешаване или блокиране на специфични връзки

Iptables ни позволява да филтрираме трафика по IP, диапазон от IP-та и порт.

Блокиране на връзка за едно IP:

```
Iptables -A INPUT -s 192.168.0.102 -j DROP
```

За да се добавят правила към вече съществуващите използваме аргумента “-A”. Iptables чете листа с правила отгоре надолу, линия по линия докато не открие такова което съвпада.

Ако искаме да вмъкнем правило над друго използваме “-i” вместо “A”.

Блокиране на IP-та в определен обхват:

```
iptables -A INPUT -s 192.168.0.102/255 -j DROP
```

Или

```
iptables -A INPUT -s 192.168.0.102/255.255.255.0 -j DROP
```

Блокиране на SSH връзка от 192.168.0.102

```
Iptables -A INPUT -p tcp -dport ssh -s 102.168.0.102 -j DROP
```

В този случай можем да заменим “ssh” със друг протокол или порт. Вида на връзката която използва протокола задаваме чрез “-p tcp”. За да блокираме ssh за всички IP адреси премахваме “-s 0.0.0.0”:

```
iptables -A INPUT -p tcp -dport ssh -j DROP
```

Двупосочна комуникация

Тъй като мрежовият трафик обикновено трябва да бъде двупосочен - входящ и изходящ, необходимо е да се създаде правило за защитната стена, което позволява *установен(established)* и *свързан(related)* входящ трафик, така че сървърът да позволява връщане на трафик към изходящите връзки, инициирани от самия сървър:

```
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

ESTABLISHED - означава, че пакетът е свързан с връзка, която е виждала пакети в двете посоки.

RELATED - Пакетът стартира нова връзка, но е свързан с вече съществуваща, като например FTP трансфер на данни или ICMP(Internet Control Message Protocol) грешка.

Запазване на промените

Промените, които правим в правилата на iptables, ще бъдат премахнати следващия път, когато услугата iptables бъде рестартирана, освен ако не изпълните команда за запазване на промените. Тази команда може да е различава в зависимост от дистрибуцията която използваме:

Ubuntu:

```
sudo /sbin/iptables-save
```

Red Hat / CentOS:

```
/sbin/service iptables save  
  
/etc/init.d/iptables save
```

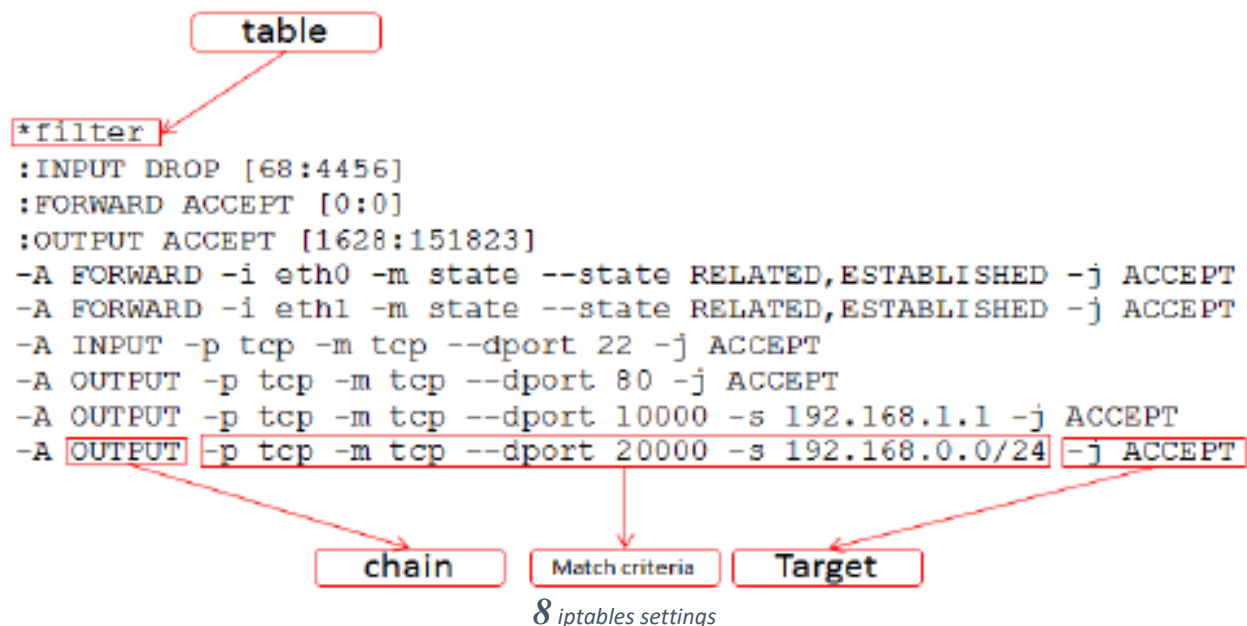
В някой дистрибуции, като Slackware, е необходимо да се конфигурира защитната стена. За целта ще използваме „Easy Firewall Generator for IPTables“. След като генерираме скрипта го запазваме в */etc/rc.d/rc.firewall* и го правим изпълним:

```
chmod a+x /etc/rc.d/rc.firewall
```

За да изтрием всички конфигурирани правила можем да използваме командата:

```
iptables -F
```

Как изглежда примерен файл в който се съхраняват настройките на iptables









Заклучение


Iptables се използва за настройка, поддръжка и проверка на таблиците на правилата на IP пакетния филтър в ядрото на Linux. Могат да бъдат дефинирани няколко различни таблици. Всяка таблица съдържа редица вградени вериги и такива добавени от потребителя.

Всяка верига е списък с правила, които могат да съвпадат с набор от пакети. Всяко правило определя какво да се прави с пакета, който съвпада.

Iptables изисква root привилегии за да функционира. В повечето Linux системи Iptables е инсталиран в директорията /usr/sbin/iptables и е документиран в своята man страница.

Източници

-  <https://unix.stackexchange.com/questions/46029/why-cant-i-use-the-reject-policy-on-my-iptables-output-chain>
-  <https://www.howtogeek.com/177621/the-beginners-guide-to-iptables-the-linux-firewall/>
-  <https://linux.die.net/man/8/iptables>
-  <http://www.slackware.com/~alien/efg/>
-  https://docs.slackware.com/howtos:security:basic_security
-  <https://serverfault.com/questions/371316/iptables-difference-between-new-established-and-related-packets>

 <https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands>

 https://www.researchgate.net/publication/272184105_Parallel_Implementation_of_Linux_Packet_Filtering/figures?lo=1