iptables

iptables 参数详解

iptables 优先级顺序

iptables 多条规则有冲突的时候，排在上面的规则优先。

比如我们已经设置了 iptables -A INPUT -p udp –dport 53 -j REJECT

那么如果再执行iptables -A INPUT -p udp –dport 53 -s 180.169.223.10 -j ACCEPT ，则不会生效

-A参数是append，添加的规则会放在追后面，而前面已经有REJECT 该端口所有的访问了，那么这条ACCEPT就不会生效。

所以这里-A要改成-I，也就是insert的意思，插入一条记录，那么这条就会放在最前面，就在那条REJECT前面了，这样就能生效。

```
iptables -I INPUT -p udp --dport 53 -s 180.169.223.10 -j ACCEPT
```

这样我们就能在拒绝所有地址访问我们的udp 53端口之后，指定给180.169.223.10能访问了。

那如果我不想把新的规则加入到最前面，也不想加在最后，我要放到一个中间指定的地方，怎么做呢？ 使用-I 的同时，加入编号就可以了。

示例：

```
iptables -I INPUT 3 -p tcp --dport 80 -s 180.168.233.10 -j ACCEPT
```

以上命令中，我们使用iptables -I INPUT 3 xxxxxxxx

这里就是讲后面的规则插入到INPUT链中的第三条里面去了，后面的规则编号依次+1.

删除指定iptables规则

查询当前iptables的规则number

这里我们使用了这样几条命令

```
iptables -L -n --line-numbers    ##所有链的规则number

iptables -L INPUT --line-numbers ## 查看INPUT的

iptables -L OUTPUT --line-numbers ## 查看OUTPUT的

iptables -L FORWARD --line-numbers ##查看FORWARD的
```

```
[root@natasha ~]# iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     icmp --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    DOCKER-ISOLATION  all  --  0.0.0.0/0            0.0.0.0/0
2    DOCKER     all  --  0.0.0.0/0            0.0.0.0/0
3    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
4    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
5    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     tcp  --  0.0.0.0/0            125.39.240.113      tcp dpt:80 reject-with icmp-port-
unreachable
2    REJECT     tcp  --  0.0.0.0/0            61.135.157.156      reject-with icmp-port-unreachable

Chain DOCKER (1 references)
num  target     prot opt source               destination
1    ACCEPT     udp  --  0.0.0.0/0            172.17.0.2          udp dpt:4500
2    ACCEPT     udp  --  0.0.0.0/0            172.17.0.2          udp dpt:500
3    ACCEPT     tcp  --  0.0.0.0/0            172.17.0.3          tcp dpt:443
4    ACCEPT     tcp  --  0.0.0.0/0            172.17.0.3          tcp dpt:80

Chain DOCKER-ISOLATION (1 references)
num  target     prot opt source               destination
1    RETURN     all  --  0.0.0.0/0            0.0.0.0/0

[root@natasha ~]# iptables -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     icmp --  anywhere             anywhere            reject-with icmp-port-unreachable
[root@natasha ~]# iptables -L OUTPUT --line-numbers
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     tcp  --  anywhere             no-data             tcp dpt:http reject-with icmp-
port-unreachable
2    REJECT     tcp  --  anywhere             61.135.157.156      reject-with icmp-port-unreachable
[root@natasha ~]# iptables -L FORWARD --line-numbers
Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    DOCKER-ISOLATION  all  --  anywhere             anywhere
2    DOCKER     all  --  anywhere             anywhere
3    ACCEPT     all  --  anywhere             anywhere            ctstate RELATED,ESTABLISHED
4    ACCEPT     all  --  anywhere             anywhere
5    ACCEPT     all  --  anywhere             anywhere
```

根据编号删除规则

默认不指定表的时候，就是找的filter表，那这个时候我们要删除filter表里OUTPUT链里第二条规则，则需要执行iptables -D OUTPUT 2，如下所示：

```
[root@natasha ~]# iptables -L OUTPUT -n -t filter --line-numbers
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     tcp  --  0.0.0.0/0            125.39.240.113      tcp dpt:80 reject-with icmp-port-
unreachable
2    REJECT     tcp  --  0.0.0.0/0            61.135.157.156      reject-with icmp-port-unreachable
[root@natasha ~]# iptables -D OUTPUT 2
[root@natasha ~]# iptables -L OUTPUT -n -t filter --line-numbers
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     tcp  --  0.0.0.0/0            125.39.240.113      tcp dpt:80 reject-with icmp-port-
unreachable
[root@natasha ~]#
```

成功删除完成。

example iptables

```
#!/bin/bash
iptables -F
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p icmp -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp -s 192.168.105.4 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p udp --dport 53 -j ACCEPT
iptables -A INPUT -p tcp --dport 2049 -j ACCEPT
iptables -A INPUT -s 192.168.105.0/24 -j ACCEPT
iptables -A INPUT -j REJECT
iptables -A INPUT -j REJECT --reject-with icmp-host-prohibited
iptables -A FORWARD -j REJECT --reject-with icmp-host-prohibited

service iptables save
```

- 禁止ping

ping命令使用的是icmp协议，所以如果要禁止别人来ping我们的服务器，我们可以做如下设置。

正常情况下可以ping通目标主机

```
[root@alvin ~]# ping 192.168.127.51
PING 192.168.127.51 (192.168.127.51) 56(84) bytes of data.
64 bytes from 192.168.127.51: icmp_seq=1 ttl=64 time=0.232 ms
64 bytes from 192.168.127.51: icmp_seq=2 ttl=64 time=0.317 ms
```

- [x] –reject-with icmp-host-prohibited

现在目标主机添加一条iptables规则，这里我们设置的是拒绝任何网段来ping 拒绝的方式是–reject-with icmp-host-prohibited

```
[root@zabbix ~]# sudo iptables -A INPUT -p icmp -s 0.0.0.0/0 -j REJECT  --reject-with icmp-host-prohibited
```

- 效果

然后再ping的时候，就发现ping不同了，显示Destination Host Prohibited

```
[root@alvin ~]# ping 192.168.127.51 -c 2
PING 192.168.127.51 (192.168.127.51) 56(84) bytes of data.
From 192.168.127.51 icmp_seq=1 Destination Host Prohibited
From 192.168.127.51 icmp_seq=2 Destination Host Prohibited

--- 192.168.127.51 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 999ms
```

- [x] –reject-with icmp-net-unreachable

那么现在我们再用另一种方式去禁止ping，那就是–reject-with icmp-net-unreachable

先删除之前的记录,查看规则的number后删除对应的规则

```
[root@zabbix ~]# iptables -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     icmp --  anywhere             anywhere            reject-with icmp-host-prohibited
[root@zabbix ~]# iptables -D INPUT 1
[root@zabbix ~]# iptables -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
[root@zabbix ~]#
```

添加新的纪录,使用–reject-with icmp-net-unreachable

```
iptables -A INPUT -p icmp -s 0.0.0.0/0 -j REJECT  --reject-with icmp-net-unreachable
```

那接下来，我们在访问该服务器的时候就是Unreachable了。

```
[root@alvin ~]# ping dhcp.alv.pub -c 2
PING dhcp.alv.pub (192.168.127.1) 56(84) bytes of data.
From 192.168.127.1 (192.168.127.1) icmp_seq=1 Destination Net Unreachable
From 192.168.127.1 (192.168.127.1) icmp_seq=2 Destination Net Unreachable
```

- [x] drop

或者其实我们还可以直接掉掉包，不做响应。

还是先删除之前的规则

```
# iptables -D INPUT 1
# iptables -A INPUT -p icmp -s 0.0.0.0/0 -j drop
```

那么这个时候客户端来ping这个服务器的时候就不会收到之前那种不可达之类的提示了。 下面我们是加了-c 2,表示只ping两次，如果没加那个，会一直那样等很久,得不到相应，这样的方式在防攻击的时候能起到一定的作用。

```
[root@alvin ~]# ping dhcp.alv.pub -c 2
PING dhcp.alv.pub (192.168.127.1) 56(84) bytes of data.

--- dhcp.alv.pub ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1000ms
```

NAT

linux系统下允许包转发

临时开启

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

永久开启

```
echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
sysctl -p
```

将本地所有tcp端口请求转发到目标IP地址上

这里我们本服务器IP地址是192.168.127.83， 目标服务器是一台vmware esxi，IP地址是192.168.127.60

进行如下设置后，就可以通过访问192.168.127.83来访问到我们的vmware esxi了。

```
iptables -t nat -I PREROUTING -d 192.168.127.83 -p tcp -j DNAT --to-destination 192.168.127.60
iptables -t nat -I POSTROUTING -s 192.168.127.0/24 -p tcp -j SNAT --to-source 192.168.127.83
```

本地端口转发为目标服务器器指定端口

转发一个80端口

将本地192.168.38.1端口上的80转发到192.168.127.51的80上。

```
iptables -t nat -I PREROUTING -d 192.168.38.1 -p tcp --dport 80 -j DNAT --to-destination
192.168.127.51:80
```

上面这条规则配置了如何过去转发本地80到目标服务器，但是数据回来之后还要伪装修改一下才能返回给客户端，需要还需要添加一条。

所有来自192.168.38.0网段的对于目标服务器192.168.127.51的tcp端口为80的请求，都伪装成本服务器

如果使用-s -d -p –dport -o 之类的参数，就是默认对所有都开放。不指定网段，不指定端口，那么所有通过该服务器装发出去的对所有端口的请求，都会变成该服务器发出的请求。

```
iptables -t nat -I POSTROUTING -s 192.168.38.0/24 -d 192.168.127.51 -p tcp --dport 80 -j MASQUERADE
```

或者可以用下面的命令，将-j MASQUERADE换成–to-source 192.168.127.1，效果是一样的，只是指定了ip。 这两条命令用其中一条就可以了，

```
iptables -t nat -I POSTROUTING -s 192.168.38.0/24 -d 192.168.127.51 -p tcp --dport 80 -j SNAT --to-
source 192.168.127.1
```

转发vmware esxi的三个端口

本地服务器IP 192.168.127.74， 目标服务器IP 192.168.127.60， 目标服务器是vmware esxi 服务器，我们需要转发三个端口。

```
iptables -t nat -I PREROUTING -d 192.168.127.74 -p tcp  --dport 902 -j DNAT --to-destination
192.168.127.60:902
iptables -t nat -I PREROUTING -d 192.168.127.74 -p tcp  --dport 80 -j DNAT --to-destination
192.168.127.60:80
iptables -t nat -I PREROUTING -d 192.168.127.74 -p tcp  --dport 443 -j DNAT --to-destination
192.168.127.60:443

iptables -t nat -I POSTROUTING -s 192.168.127.0/24 -p tcp -j SNAT --to-source 192.168.127.74
```

然后就可以通过访问192.168.127.74来访问到192.168.127.60的esxi服务了。

本地端口转发到本地其他端口

将80端口转发到8080

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
```