

老男孩教育-综合架构-iptables防火墙

老男孩教育-综合架构-iptables防火墙

1. 防火墙概述

2. 防火墙

2.1 防火墙种类及使用说明

2.2 必须熟悉的名词

2.3 iptables 执行过程※※※※※

2.4 表与链※※※※※

2.4.1 简介

2.4.2 每个表说明

1) filter表

2) nat表

2.5 环境准备及命令

3.5.2 iptables命令参数

3.6 配置filter表规则※※※※※

3.6.1 禁止访问22端口

3.6.2 封ip,屏蔽某个ip

3.6.3 禁止网段连入 (禁止10.0.0.0/24网段访问 8888端口)

3.6.4 只允许指定网段连入 (允许172.16.1.0网段)

3.6.4 指定多个端口

3.6.5 匹配ICMP类型

3.6.6 匹配网络状态 (TCP/IP连接状态)

3.6.7 限制并发及速率

3.6.8 防火墙规则的保存与恢复☆☆☆☆

3.6.9 filter表小结

3.7 实际生产用法

3.8 nat

3.8.1 实现共享上网※※※※※

1. 防火墙配置

2. web配置

3. 完成后 在web01 发出 ip r和ping 外网ip的结果

3.8.2 实现端口转发※※※※※

3.8.3 实现ip映射

3.8.4 nat表总结

4. 总结

- 5、请写出查看iptables当前所有规则的命令。
- 6、禁止来自10.0.0.188 ip地址访问80端口的请求
- 7、如何在命令行执行的iptables规则永久生效?
- 8、实现把访问10.0.0.3:80的请求转到172.16.1.17:80
- 9、实现172.16.1.0/24段所有主机通过124.32.54.26外网IP共享上网。

1. 防火墙概述

目标:

封端口,封ip

实现NAT功能

共享上网

端口映射(端口转发),ip映射

2. 防火墙

2.1 防火墙种类及使用说明

- 硬件: 整个企业入口
 - 三层路由: H3C 华为 Cisco(思科)
 - 防火墙: 深信服, 绿盟, 奇安信.....
 - Juniper
- 软件: 开源软件 网站内部 封ip 封ip
 - iptables 写入到Linux内核中,以后服务docker 工作在 4层(大部分)
 - firewalld C7
 - nftalbes C8
 - ufw (ubuntu firewall) Ubuntu
- 云防火墙(公有云)
 - 阿里云:
 - 安全组 (封ip,封端口)
 - NAT网关(共享上网,端口映射....)
 - waf应用防火墙
- waf防火墙(应用防火墙,处理7层的攻击) SQL注入,等攻击.
 - 书写规则(描述攻击过程,关键提示,关键操作.)

中小企业：使用公有云,安全组,waf防火墙,态势感知。

访问量巨大：使用硬件防火墙,waf防火墙,硬件服务器+云服务器

2.2 必须熟悉的名词

- **容器**：瓶子 罐子 存放东西
- **表(table)**： 存放**链**的容器,防火墙最大概念
- **链(chain)**： 存放**规则**的容器
- **规则(policy)**： 准许或拒绝规则 ,未来书写的防火墙条件就是各种防火墙**规则**

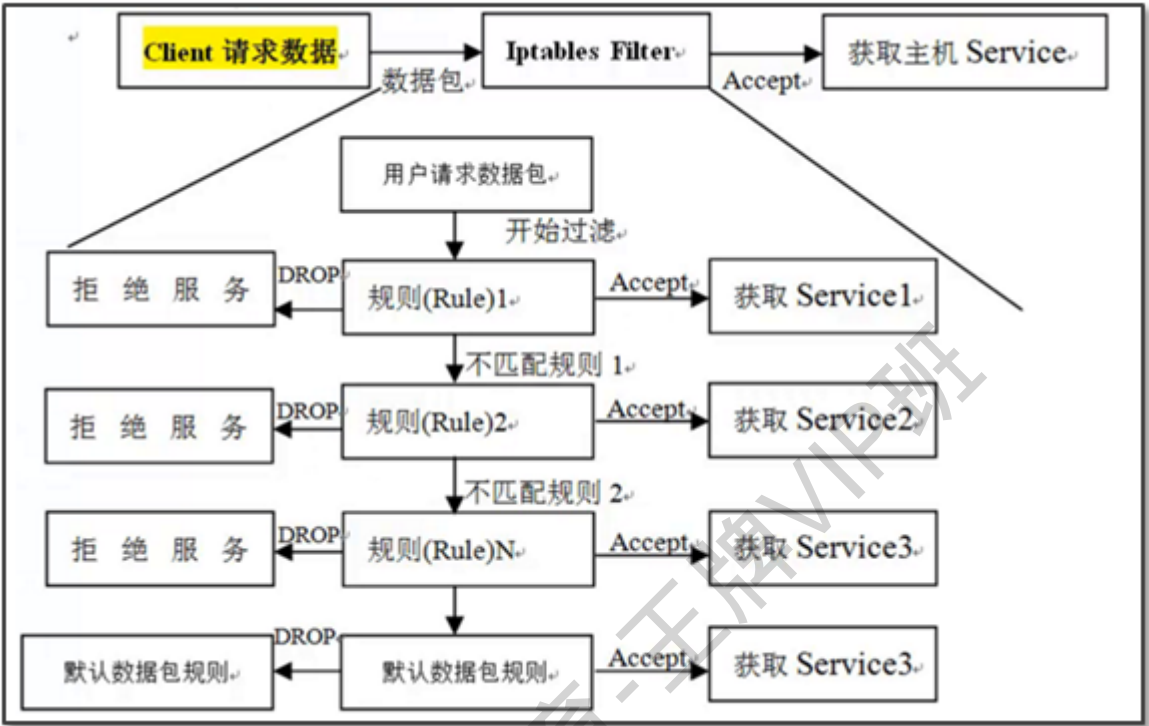


Netfilter	表 (tables)	链 (chains)	规则 (Policy)
一栋楼	楼里的房子	房子里的柜子	柜子里衣服, 摆放规则

2.3 iptables 执行过程*****

工作流程小结: *****

- 1. 防火墙是层层过滤的，实际是按照配置规则的顺序从上到下，从前到后进行过滤的。
- 2. 如果匹配成功规则，即明确表示是拒绝(DROP)还是接收(ACCEPT)，数据包就不再向下匹配新的规则。
- 3. 如果规则中没有明确表明是阻止还是通过的，也就是没有匹配规则，向下进行匹配，直到匹配默认规则得到明确的阻止还是通过。
- 4. 防火墙的默认规则是所有规则都匹配完才会匹配的。



Q: 如果配置了一条拒绝的规则,该放在哪里???

2.4 表与链*****

2.4.1 简介

- 表(table)是对功能的分类,防火墙功能(filter表),共享上网,端口转发(nat表)
- 链对数据流进行处理,需要使用不同的链(数据流入(INPUT),数据流出(OUTPUT))
- iptables 是4表伍链
- 4表: filter 表 nat表 raw表 mangle表
- 伍链: INPUT OUTPUT FORWARD PREROUTING POSTROUTING

pre.... 之前

post之后

2.4.2 每个表说明

1) filter表

- 是iptables默认的表,filter表示过滤.
- 实现防火墙功能:(对数据包的filter过滤)屏蔽或准许,端口,ip.

filter** 表**	强调: 主要和主机自身相关, 真正负责主机防火墙功能的 (过滤流入流出主机的数据包) filter表示iptables默认使用的表, 这个表定义了三个链 (chains) 企业工作场景: 主机防火墙
INPUT	负责过滤所有目标地址是本机地址的数据包 通俗来说: 就是过滤进入主机的数据包 (能否让数据包进入服务器)
FORWARD	路过: 负责转发流经主机的数据包.起转发的作用, 和NAT关系很大, 后面会详细介绍 LVS NAT模式, net.ipv4.ip_forward=0
OUTPUT	处理所有源地址是本机地址的数据包 通俗的讲: 就是处理从主机发出去的数据包

2) nat表

- 实现nat功能
 - 实现共享上网(内网服务器上外网)
 - 端口映射和ip映射

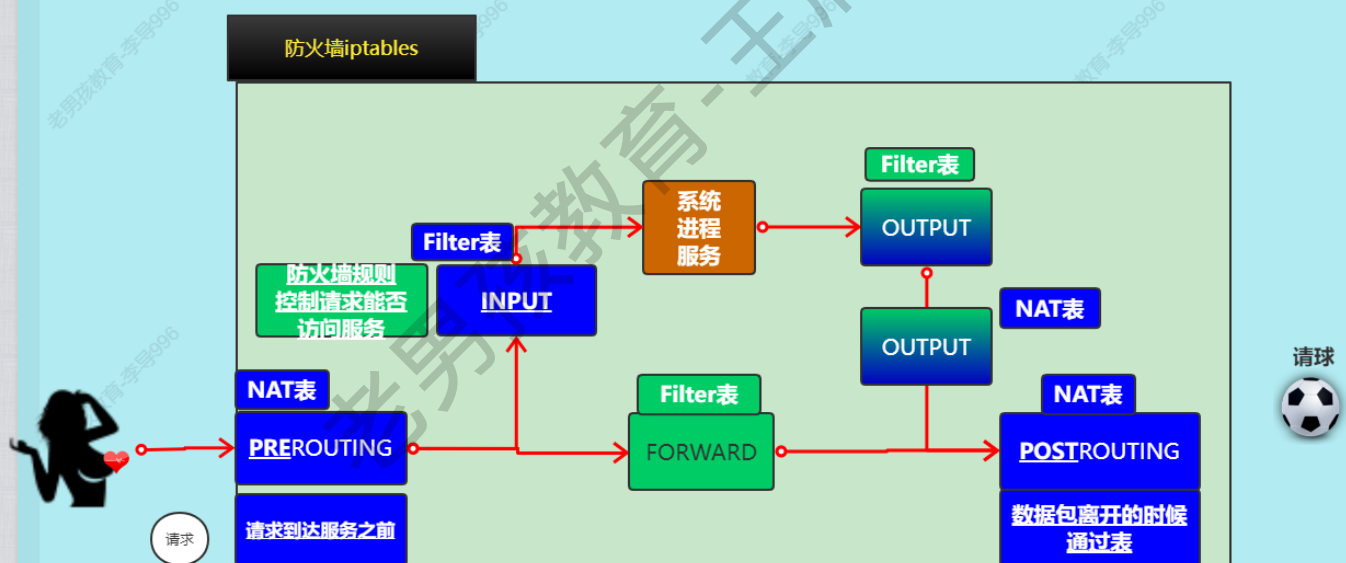
nat	负责网络地址转换的, 即来源与目的IP地址和port的转换。 应用: 和主机本身无关, 一般用于局域网共享上网或者特殊的端口转换服务相关。 工作场景: 1. 用于企业路由 (zebra) 或网关 (iptables), 共享上网 (POSTROUTING) 2. 做内部外部IP地址一对一映射 (dmz), 硬件防火墙映射IP到内部服务器, ftp服务 (PREROUTING) 3. WEB, 单个端口的映射, 直接映射80端口 (PREROUTING) 这个表定义了3个链, nat功能相当于网络的acl控制。和网络交换机acl类似。
OUTPUT	和主机放出去的数据包有关, 改变主机发出数据包的目的地址。
PREROUTING	在数据包到达防火墙时, 进行路由判断之前执行的规则, 作用是改变数据包的目的地址、目的端口等就是收信时, 根据规则重写收件人的地址。 例如: 把公网IP: xxx.xxx.xxx.xxx映射到局域网的xx.xx.xx.xx服务器上。 如果是web服务, 可以报80转换为局域网的服务器9000端口上 10.0.0.61 8080(目标端口) ----nat---à 10.0.0.7 22
POSTROUTING	在数据包离开防火墙时进行路由判断之后执行的规则, 作用改变数据包的源地址, 源端口等。 写好发件人的地址, 要让家人回信时能够有地址可回。 例如: 默认笔记本和虚拟机都是局域网地址, 在出网的时候被路由器将源地址改为了公网地址。 生产应用: 局域网共享上网。



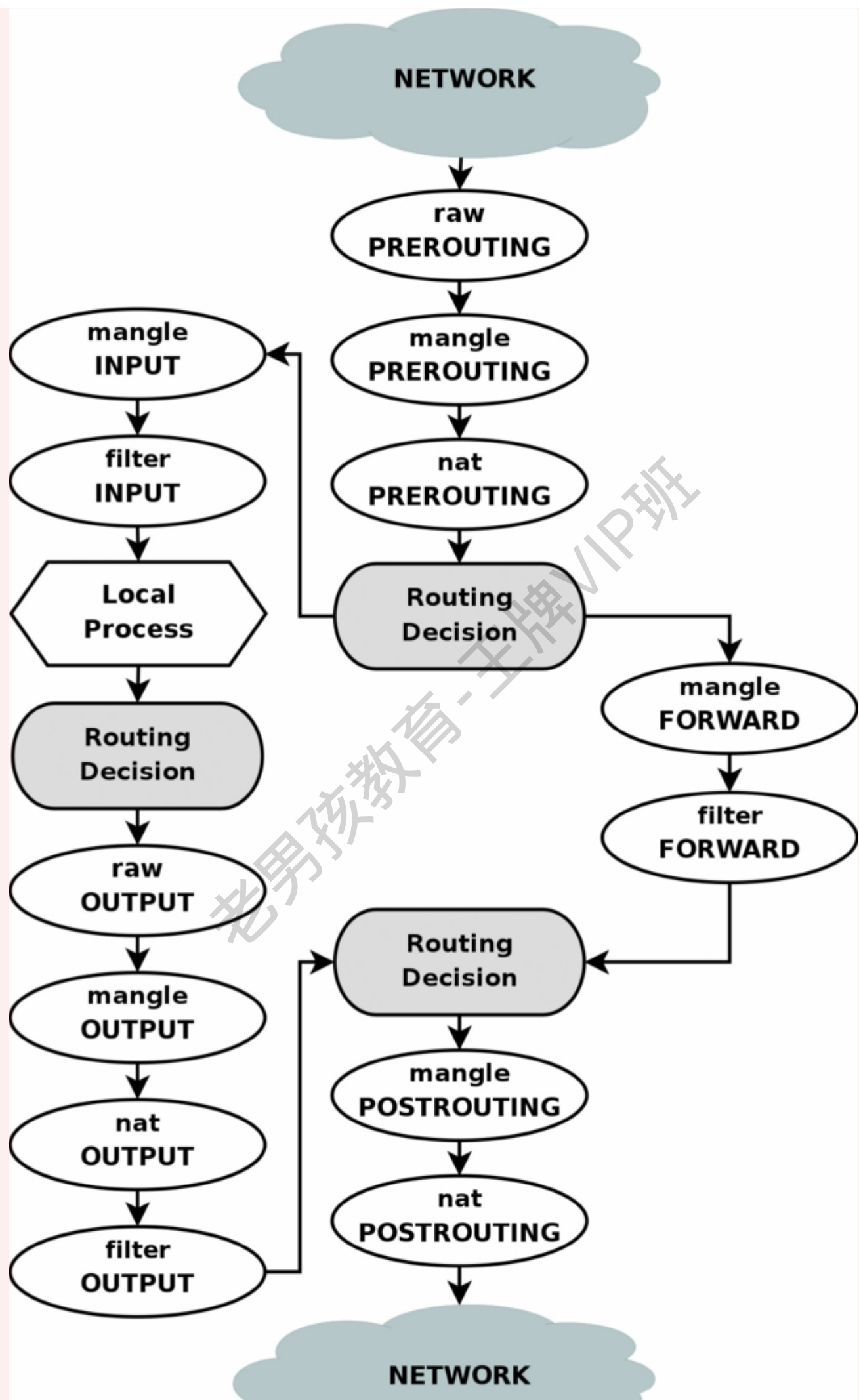


高清无码传送门: <https://www.processon.com/view/link/5d1b0c51e4b079b9e2177b3c>

老男孩教育-最新架构-iptables 4表5链规则



课外了解: 官方最全超级详解传送门



2.5 环境准备及命令

iptables iptables启动或关闭的命令

1	m01	10.0.0.61	172.16.1.61
2	web01	10.0.0.7	172.16.1.7
3	web02	10.0.0.8	172.16.1.8

```
1 yum install -y iptables-services
2
3
4 [root@oldboy-m01 ~]# uname -r
5 3.10.0-862.el7.x86_64
6 [root@oldboy-m01 ~]# yum install -y iptables-
services
7
8
9
10 [root@m01 ~]# rpm -ql iptables-services
11 /etc/sysconfig/ip6tables
12 /etc/sysconfig/iptables #
   防火墙的配置文件
13 /usr/lib/systemd/system/ip6tables.service
14 /usr/lib/systemd/system/iptables.service #
   防火墙服务配置文件(命令) systemctl start iptables
15
16
17 [root@m01 ~]# rpm -ql iptables
```



```
18 /usr/sbin/iptables          #iptables 命令 添加/删
   除/查看 规则(4表伍链)
19
20 /usr/sbin/iptables-save     #iptables规则 输出(保存)
21 /usr/sbin/iptables-restore  # 恢复
22
23
24
25
26
27
28
29 #防火墙相关模块 加载到内核中
30 #写入到开机自启动.
31 modprobe ip_tables
32 modprobe iptable_filter
33 modprobe iptable_nat
34 modprobe ip_conntrack
35 modprobe ip_conntrack_ftp
36 modprobe ip_nat_ftp
37 modprobe ipt_state
38
39 #永久
40 cat >>/etc/rc.local<<EOF
41 modprobe ip_tables
42 modprobe iptable_filter
43 modprobe iptable_nat
44 modprobe ip_conntrack
45 modprobe ip_conntrack_ftp
46 modprobe ip_nat_ftp
47 modprobe ipt_state
48 EOF
49
```

```
50
51 [root@m01 ~]# lsmod |egrep 'filter|nat|ipt'
52 nf_nat_ftp                12770    0
53 nf_conntrack_ftp          18638    1 nf_nat_ftp
54 iptable_nat               12875    0
55 nf_nat_ipv4               14115    1 iptable_nat
56 nf_nat                    26787    2
    nf_nat_ftp,nf_nat_ipv4
57 nf_conntrack              133053    6
    nf_nat_ftp,nf_nat,xt_state,nf_nat_ipv4,nf_conntrack
    _ftp,nf_conntrack_ipv4
58 iptable_filter            12810    0
59 ip_tables                 27126    2
    iptable_filter,iptable_nat
60 libcrc32c                 12644    3
    xfs,nf_nat,nf_conntrack
61
62
63
64
65 [root@m01 ~]# systemctl stop firewalld
66 [root@m01 ~]# systemctl disable firewalld
67
68
69
70 [root@m01 ~]# systemctl start iptables.service
71 [root@m01 ~]# systemctl enable iptables.service
72 Created symlink from
    /etc/systemd/system/basic.target.wants/iptables.ser
    vice to /usr/lib/systemd/system/iptables.service.
73 [root@m01 ~]# systemctl status iptables.service
74 ● iptables.service - IPv4 firewall with iptables
```

```
75     Loaded: loaded
        (/usr/lib/systemd/system/iptables.service; enabled;
        vendor preset: disabled)
76     Active: active (exited) since Fri 2021-05-28
        09:59:53 CST; 5s ago
77     Process: 7971
        ExecStart=/usr/libexec/iptables/iptables.init start
        (code=exited, status=0/SUCCESS)
78     Main PID: 7971 (code=exited, status=0/SUCCESS)
79
80 May 28 09:59:53 m01 systemd[1]: Starting IPv4
        firewall with iptables...
81 May 28 09:59:53 m01 iptables.init[7971]: iptables:
        Applying firewall rules: [ OK ]
82 May 28 09:59:53 m01 systemd[1]: Started IPv4
        firewall with iptables.
83
84
85
86
87
88 #查看filter表中的规则 ,默认查看的是filter表
89
90 [root@m01 ~]# iptables -nL
91 Chain INPUT (policy ACCEPT)
92 target     prot opt source
        destination
93 ACCEPT     all  --  0.0.0.0/0          0.0.0.0/0
        state RELATED,ESTABLISHED
94 ACCEPT     icmp --  0.0.0.0/0          0.0.0.0/0
95 ACCEPT     all  --  0.0.0.0/0          0.0.0.0/0
```

```
96 ACCEPT      tcp    --  0.0.0.0/0          0.0.0.0/0
    state NEW tcp dpt:22
97 REJECT      all    --  0.0.0.0/0          0.0.0.0/0
    reject-with icmp-host-prohibited
98
99 Chain FORWARD (policy ACCEPT)
100 target      prot opt source
    destination
101 REJECT      all    --  0.0.0.0/0          0.0.0.0/0
    reject-with icmp-host-prohibited
102
103 Chain OUTPUT (policy ACCEPT)
104 target      prot opt source
    destination
105
106 #查看指定表中的规则
107 [root@m01 ~]# iptables -t nat -nL
108 Chain PREROUTING (policy ACCEPT)
109 target      prot opt source
    destination
110
111 Chain INPUT (policy ACCEPT)
112 target      prot opt source
    destination
113
114 Chain OUTPUT (policy ACCEPT)
115 target      prot opt source
    destination
116
117 Chain POSTROUTING (policy ACCEPT)
118 target      prot opt source
    destination
119
```

120

121

122

老男孩教育-最新架构-iptables命令结果

指定表	
链	[root@m01 ~]# iptables -nL -t filter
链中的规则	Chain INPUT (policy ACCEPT) 默认规则:允许
	target prot opt source destination
	ACCEPT all -- 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
	ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
	ACCEPT all -- 0.0.0.0/0 0.0.0.0/0
	ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22
	REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited
链	Chain FORWARD (policy ACCEPT)
	target prot opt source destination
	REJECT all -- 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited
链	Chain OUTPUT (policy ACCEPT)
	target prot opt source destination
	[root@m01 ~]#

3.5.2 iptables命令参数

参数	含义
-L	显示表中的所有规则
-n	不要把端口或ip反向解析为名字
指定表	
-t	指定表, 不指定默认是filter表
指定连接(加入/追加/删除)	
-A	append 追加把规则写入到链的末尾. 加入允许类规则 使用-A
-I	insert 把规则加在链的第1条 拒绝类规则放在所有规则最上面 拒绝类 -I
-D	delete 删除 -D INPUT 1

参数	含义	
-p	指定协议 protocol tcp/udp/icmp/all	
--dport	目标端口 dest destination ▲ 指定端口的时候加上协议 -p tcp	
--sport	源端口 source 源	
-s	--source 源ip ▲ 如果只屏蔽/准许ip,网段,不用加上协议.	
-d	--destination 目标ip	
-m	指定模块 multiport	
-i	input 输入的时候 从哪个网卡进来	
-o	ouput 输出的时候 从哪个网卡出去	

参数	含义	
-j	满足条件后的动作： DROP(拒绝)/ACCEPT(准许)/REJECT(拒绝)	
	DROP REJECT拒绝 DROP 把数据丢掉 不会返回信息给用户 REJECT 拒绝 返回拒绝信息	

参数	含义	
-F flush	清除指定表中所有的规则,备份.	
-X	删除用户 自定义的链	
-Z zero	链的计数器清零（数据包计数器与数据包字节计数器） iptables	
-v	显示数据包,数据量	

iptables命令及选项	指定表	指定链 (插入/追加/删除)	ip	具体要求 (端口, ip, 协议)	端口	动作
iptables	-t filter	-A INPUT	-s	-p tcp/udp/icmp	--dport 目标端口	-j DROP
		-I	-d		--sport 源端口	-j REJECT
		-D				-j ACCEPT

3.6 配置filter表规则※※※※※

- 正式配置之前 先备份，清空规则

```
1 [root@m01 ~]# iptables -F
2 [root@m01 ~]# iptables -X
3 [root@m01 ~]# iptables -Z
4
5 [root@m01 ~]# iptables -nL
6 Chain INPUT (policy ACCEPT)
7 target                prot opt source                destination
8
9 Chain FORWARD (policy ACCEPT)
10 target                prot opt source                destination
11
12 Chain OUTPUT (policy ACCEPT)
13 target                prot opt source                destination
```

3.6.1 禁止访问22端口

```
1 #拒绝用户访问22端口
2 iptables -t filter -A INPUT -p tcp --dport 22 -j DROP
3 #查看规则并加上序号
4 iptables -t filter -nL --line-number
5 #删除规则
6 iptables -t filter -D INPUT 1 #根据序号删除
```

-t 用于指定表,不写默认就是filter表


```
[root@m01 ~]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      tcp  --  0.0.0.0/0              0.0.0.0/0          tcp dpt:22

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@m01 ~]# _
```

```
[root@m01 ~]# iptables -nL --line-number
Chain INPUT (policy ACCEPT)
num target     prot opt source                destination
1  DROP      tcp  --  0.0.0.0/0              0.0.0.0/0          tcp dpt:22

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

Chain OUTPUT (policy ACCEPT)
num target     prot opt source                destination
[root@m01 ~]# iptables -D INPUT 1
[root@m01 ~]# iptables -nL
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
```

```
[root@m01 ~]# iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      tcp  --  0.0.0.0/0              0.0.0.0/0          tcp dpt:22

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@m01 ~]# iptables -nL --line-number
Chain INPUT (policy ACCEPT)
num target     prot opt source                destination
1  DROP      tcp  --  0.0.0.0/0              0.0.0.0/0          tcp dpt:22

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

Chain OUTPUT (policy ACCEPT)
num target     prot opt source                destination
[root@m01 ~]# iptables -t filter -D INPUT 1
[root@m01 ~]# _
```

3.6.2 封ip,屏蔽某个ip

```
1 [root@oldboy-m01 ~]# iptables -I INPUT -s 10.0.0.7
   -j DROP
2 [root@oldboy-m01 ~]# iptables -I INPUT -s
   172.16.1.7 -j DROP
3 [root@oldboy-m01 ~]#
4 [root@oldboy-m01 ~]# iptables -nL
5 Chain INPUT (policy ACCEPT)
6 target      prot opt source                destination
7 DROP        all  --  172.16.1.7             0.0.0.0/0
8 DROP        all  --  10.0.0.7                0.0.0.0/0
9
10 Chain FORWARD (policy ACCEPT)
11 target      prot opt source                destination
12
13 Chain OUTPUT (policy ACCEPT)
14 target      prot opt source                destination
15
16 iptables -I INPUT -s 172.16.1.0/24 -j DROP
17
```

3.6.3 禁止网段连入（禁止10.0.0.0/24网段访问8888端口）

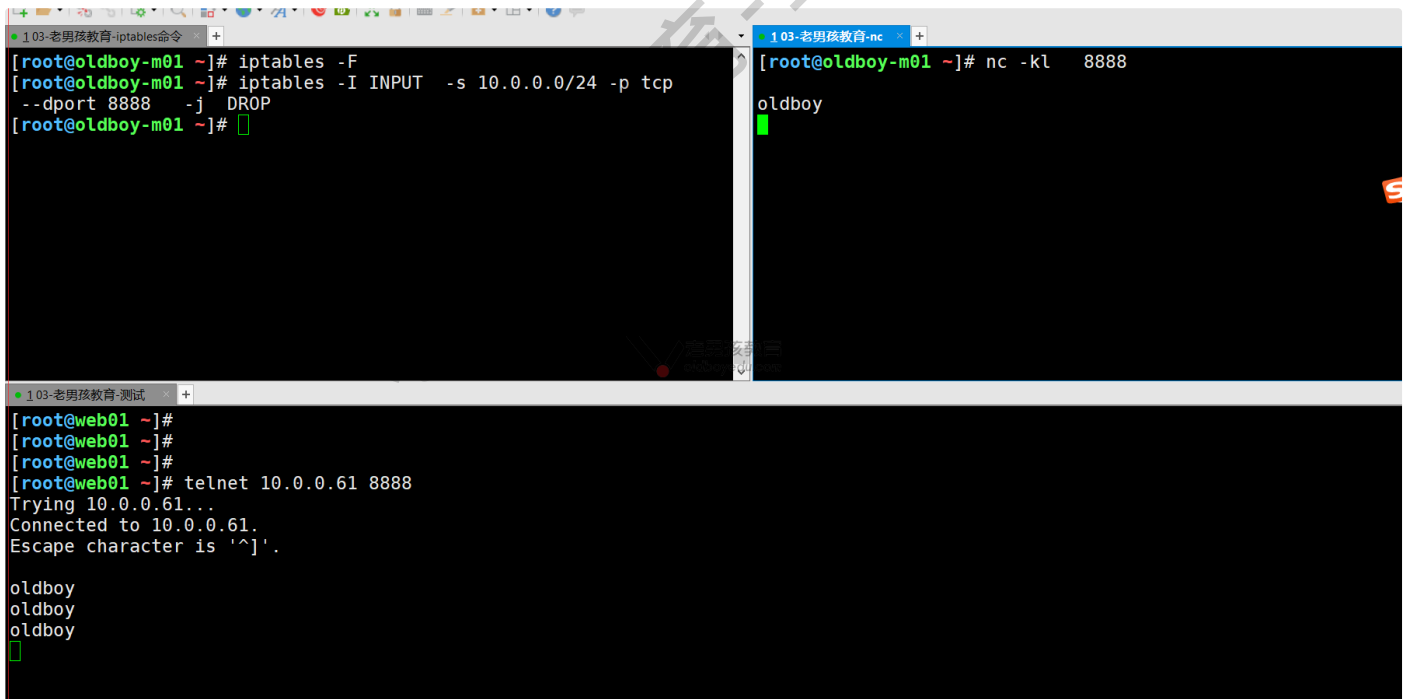
```
nc ncat netcat
```

```
nc -lk
```

```
telnet
```

```
1 iptables -I INPUT -s 10.0.0.0/24 -p tcp --dport 8888 -j DROP
```

```
1 [root@web01 ~]# ssh 10.0.0.61 hostname
2 ^C
3 [root@web01 ~]# ssh 172.16.1.61 hostname
4 root@172.16.1.61's password:
5 m01
```



The screenshot shows three terminal windows. The top-left window shows the configuration of iptables rules to block traffic from 10.0.0.0/24 to port 8888. The top-right window shows a netcat listener on port 8888 receiving a connection from 'oldboy'. The bottom window shows a telnet client attempting to connect to 10.0.0.61 on port 8888, which successfully connects and shows a prompt.

```
103-老男孩教育-iptables命令
[root@oldboy-m01 ~]# iptables -F
[root@oldboy-m01 ~]# iptables -I INPUT -s 10.0.0.0/24 -p tcp --dport 8888 -j DROP
[root@oldboy-m01 ~]#

103-老男孩教育-nc
[root@oldboy-m01 ~]# nc -kl 8888
oldboy

103-老男孩教育-测试
[root@web01 ~]#
[root@web01 ~]#
[root@web01 ~]#
[root@web01 ~]# telnet 10.0.0.61 8888
Trying 10.0.0.61...
Connected to 10.0.0.61.
Escape character is '^['.
oldboy
oldboy
oldboy
```

3.6.4 只允许指定网段连入（允许172.16.1.0网段）

实现阿里云白名单功能：默认是拒绝 开放端口 网段

```
allow 10.0.0.0/24;
```

```
deny all;
```

```
1 #方法1: 利用 ! 进行排除
2 iptables -I INPUT ! -s 172.16.1.0/24 -j DROP
3
4 #只准许 10.0.0.0/24 访问 言外之意 除了 10.0.0.0/24 都
  拒绝
5 iptables -I INPUT ! -s 10.0.0.0/24 -j DROP
```

```
1 03-老男孩教育-iptables命令 x 2 03-老男孩教育-nc x +
[root@oldboy-m01 ~]# iptables -I INPUT ! -s 10.0.0.0/24 -j DROP
[root@oldboy-m01 ~]#
```

```
1 03-老男孩教育-测试 x + 老男孩教育
PING 172.16.1.61 (172.16.1.61) 56(84) bytes of data.
64 bytes from 172.16.1.61: icmp_seq=9 ttl=64 time=0.334 ms
64 bytes from 172.16.1.61: icmp_seq=10 ttl=64 time=0.436 ms
64 bytes from 172.16.1.61: icmp_seq=11 ttl=64 time=0.345 ms
64 bytes from 172.16.1.61: icmp_seq=12 ttl=64 time=0.363 ms
^C
--- 172.16.1.61 ping statistics ---
20 packets transmitted, 4 received, 80% packet loss, time 19004ms
rtt min/avg/max/mdev = 0.334/0.369/0.436/0.044 ms
[root@web01 ~]# ping 10.0.0.61
[root@web01 ~]# ping 10.0.0.61
PING 10.0.0.61 (10.0.0.61) 56(84) bytes of data.
64 bytes from 10.0.0.61: icmp_seq=1 ttl=64 time=0.235 ms
64 bytes from 10.0.0.61: icmp_seq=2 ttl=64 time=0.305 ms
^C
--- 10.0.0.61 ping statistics ---
```

```

1 #方法2： 修改链默认规则 修改为拒绝,添加准许
2 先配置好规则 准许规则
3 修改默认规则
4
5
6 iptables -P INPUT DROP #修改默认规则
7

```

```

[root@oldboy-m01 ~]# iptables -nL
Chain INPUT (policy DROP)
target     prot opt source                destination
ACCEPT     all  --  10.0.0.0/24            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@oldboy-m01 ~]#

103-老男孩教育-测试
[root@web01 ~]# ping 10.0.0.61
PING 10.0.0.61 (10.0.0.61) 56(84) bytes of data:
64 bytes from 10.0.0.61: icmp_seq=1 ttl=64 time=0.236 ms
64 bytes from 10.0.0.61: icmp_seq=2 ttl=64 time=0.656 ms
64 bytes from 10.0.0.61: icmp_seq=3 ttl=64 time=0.317 ms
^C
--- 10.0.0.61 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.236/0.403/0.656/0.181 ms
[root@web01 ~]# ping 172.16.1.61
PING 172.16.1.61 (172.16.1.61) 56(84) bytes of data.
^C
--- 172.16.1.61 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1000ms

[root@web01 ~]#

```

3.6.4 指定多个端口

```

1 iptables -I INPUT -p tcp --dport 8888 -j DROP
2 iptables -I INPUT -p tcp --dport 9999 -j DROP
3 [root@oldboy-m01 ~]# iptables -nL
4 Chain INPUT (policy ACCEPT)
5 target     prot opt source                destination

```

```
6 DROP          tcp -- 0.0.0.0/0          0.0.0.0/0
          tcp dpt:9999
7 DROP          tcp -- 0.0.0.0/0          0.0.0.0/0
          tcp dpt:8888
8
9 Chain FORWARD (policy ACCEPT)
10 target        prot opt source                destination
11
12 Chain OUTPUT (policy ACCEPT)
13 target        prot opt source                destination
14
15 #指定多个端口
16 -m multiport   -p tcp    --dport 80,443
17 多端口模块
18 iptables -A INPUT -m multiport -p tcp --dport
  80,443 -j ACCEPT
19 iptables -nL
20 iptables -A INPUT -p tcp --dport 1:1024 -j
  ACCEPT
21
22
23 #补充:
24 #
25 iptables -I INPUT -p tcp -m multiport ! --dport
  80,443 -j DROP
26 #如果是 连续的端口 可以不加上-m multiport 1:1024
27 iptables -I INPUT -p tcp --dport 1024:65535 -j
  DROP
```

3.6.5 匹配ICMP类型

- ICMP (Internet Control Message Protocol) Internet控制报文协议 ping
- 整个网站核心

通过防火墙规则 控制是否可以ping

```
1 iptables -I INPUT -p icmp --icmp-type 8 -j DROP
2 [root@m01 ~]# iptables -nL
3 Chain INPUT (policy ACCEPT)
4 target      prot opt source                destination
5 DROP        icmp -- 0.0.0.0/0              0.0.0.0/0
               icmp-type 8
6
7 Chain FORWARD (policy ACCEPT)
8 target      prot opt source                destination
9
10 Chain OUTPUT (policy ACCEPT)
11 target      prot opt source                destination
12
13
14 [root@m01 ~]# iptables -I INPUT -p icmp --icmp-
    type 255 -j DROP
15
```

温馨提示：更加精确的写法是

```
iptables -t filter -I INPUT -p icmp --icmp-type 8 -j DROP
```

简单写法

```
iptables -t filter -I INPUT -p icmp -j DROP
```

通过内核参数 控制 禁止被ping


```
1 [root@m01 ~]# cat /etc/sysctl.conf
2 #/proc/sys/net/ipv4/icmp_echo_ignore_all
3 #net网络 ipv4协议 icmp协议忽略所有
4 net.ipv4.icmp_echo_ignore_all = 1
5 #生效
6 sysctl -p
```

3.6.6 匹配网络状态 (TCP/IP连接状态)

-m state --state 状态即可。

NEW: 已经或将启动新的连接

ESTABLISHED: 已建立的连接

RELATED: 正在启动的新连接

INVALID: 非法或无法识别的

```
1 iptables -A INPUT -m state --state
  ESTABLISHED,RELATED -j ACCEPT
2
3 iptables -A OUTPUT -m state --state
  ESTABLISHED,RELATED -j ACCEPT
```

3.6.7 限制并发及速率

-m limit 限制模块

```
1 -m limit --limit 10/minute      #每分钟只能有10个数据包
  每6秒生成
```

-m limit --limit n/{second/minute/hour}:

解释: 指定时间内的请求速率“n”为速率, 后面为时间分别为: 秒 分 时

```

1 -m limit --limit 10/minute --limit-burst 5 每6秒释
  放工牌 给别人使用
2
3 #10个数据包
4 前5个 1个1个工牌 从第6个开始 每6秒 才能释放1个工牌

```

--limit-burst [n]

解释：在同一时间内允许通过的请求“n”为数字，不指定默认为5

- 测试 演示

```

1 #ping icmp 协议 进行测试
2 iptables -F
3 iptables -I INPUT -p icmp -m limit --limit
  10/minute --limit-burst 5 -j ACCEPT
4 iptables -A INPUT -p tcp --dport -j ACCEPT
5 iptables -P INPUT DROP
6

```

```

[root@oldboy-m01 ~]# iptables -nL
Chain INPUT (policy DROP)
target    prot opt source                destination
ACCEPT    icmp -- 0.0.0.0/0              0.0.0.0/0          limit: avg 10/min burst 5

Chain FORWARD (policy ACCEPT)
target    prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
[root@oldboy-m01 ~]#

```

- web01 进行ping测试

```

1 ping 10.0.0.61
2
3 [root@web01 ~]# ping 10.0.0.61
4 PING 10.0.0.61 (10.0.0.61) 56(84) bytes of data.
5 64 bytes from 10.0.0.61: icmp_seq=1 ttl=64
  time=0.240 ms

```

```
6 64 bytes from 10.0.0.61: icmp_seq=2 ttl=64
  time=0.376 ms
7 64 bytes from 10.0.0.61: icmp_seq=3 ttl=64
  time=0.474 ms
8 64 bytes from 10.0.0.61: icmp_seq=4 ttl=64
  time=0.693 ms
9 64 bytes from 10.0.0.61: icmp_seq=5 ttl=64 time=2.43
  ms
10 64 bytes from 10.0.0.61: icmp_seq=7 ttl=64
   time=0.351 ms      #7与1比 就是间隔6秒
11 64 bytes from 10.0.0.61: icmp_seq=13 ttl=64
   time=0.869 ms      #7 13 间隔6秒
12 64 bytes from 10.0.0.61: icmp_seq=19 ttl=64
   time=0.482 ms
13 64 bytes from 10.0.0.61: icmp_seq=25 ttl=64
   time=0.378 ms
14 64 bytes from 10.0.0.61: icmp_seq=31 ttl=64
   time=0.329 ms
15 64 bytes from 10.0.0.61: icmp_seq=37 ttl=64
   time=0.857 ms
16 64 bytes from 10.0.0.61: icmp_seq=43 ttl=64
   time=0.314 ms
17 64 bytes from 10.0.0.61: icmp_seq=49 ttl=64
   time=0.360 ms
18 64 bytes from 10.0.0.61: icmp_seq=55 ttl=64
   time=0.349 ms
19 64 bytes from 10.0.0.61: icmp_seq=61 ttl=64
   time=0.552 ms
20 64 bytes from 10.0.0.61: icmp_seq=67 ttl=64
   time=0.283 ms
21 64 bytes from 10.0.0.61: icmp_seq=73 ttl=64
   time=0.407 ms
```

```
22 64 bytes from 10.0.0.61: icmp_seq=79 ttl=64
    time=0.297 ms
23 64 bytes from 10.0.0.61: icmp_seq=85 ttl=64
    time=0.428 ms
24 64 bytes from 10.0.0.61: icmp_seq=91 ttl=64
    time=0.390 ms
25 64 bytes from 10.0.0.61: icmp_seq=97 ttl=64
    time=0.691 ms
26 64 bytes from 10.0.0.61: icmp_seq=103 ttl=64
    time=0.537 ms
27 64 bytes from 10.0.0.61: icmp_seq=109 ttl=64
    time=0.546 ms
28 64 bytes from 10.0.0.61: icmp_seq=115 ttl=64
    time=0.382 ms
29 64 bytes from 10.0.0.61: icmp_seq=121 ttl=64
    time=0.337 ms
```

3.6.8 防火墙规则的保存与恢复☆☆☆☆

- iptables-save 进行备份, 默认输出到屏幕
- iptables-restore 进行恢复, 加上文件
- 写入到/etc/sysconfig/iptables

```
1 [root@oldboy-m01 ~]# iptables-save
  >/etc/sysconfig/iptables
2 [root@oldboy-m01 ~]# cat /etc/sysconfig/iptables
3 # Generated by iptables-save v1.4.21 on Wed Feb 12
  15:31:43 2020
4 *filter
5 :INPUT DROP [92:7008]
6 :FORWARD ACCEPT [0:0]
7 :OUTPUT ACCEPT [127:14360]
8 -A INPUT -p icmp -m limit --limit 10/min -j ACCEPT
9 -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
```

```
10 -A INPUT -p tcp -m multiport --dports 80,443 -j
    ACCEPT
11 COMMIT
12 # Completed on Wed Feb 12 15:31:43 2020
13 # Generated by iptables-save v1.4.21 on Wed Feb 12
    15:31:43 2020
14 *nat
15 :PREROUTING ACCEPT [559:51047]
16 :INPUT ACCEPT [60:6814]
17 :OUTPUT ACCEPT [6801:408868]
18 :POSTROUTING ACCEPT [6801:408868]
19 COMMIT
20 # Completed on Wed Feb 12 15:31:43 2020
21 [root@oldboy-m01 ~]# iptables -nL
22 Chain INPUT (policy DROP)
23 target                prot opt source                destination
24 ACCEPT                icmp -- 0.0.0.0/0              0.0.0.0/0
                        limit: avg 10/min burst 5
25 ACCEPT                tcp -- 0.0.0.0/0              0.0.0.0/0
                        tcp dpt:22
26 ACCEPT                tcp -- 0.0.0.0/0              0.0.0.0/0
                        multiport dports 80,443
27
28 Chain FORWARD (policy ACCEPT)
29 target                prot opt source                destination
30
31 Chain OUTPUT (policy ACCEPT)
32 target                prot opt source                destination
33 [root@oldboy-m01 ~]# iptables -D 1
```

```
34 iptables: Bad rule (does a matching rule exist in
that chain?).
35 [root@oldboy-m01 ~]#
36 [root@oldboy-m01 ~]# iptables -D INPUT 1
37 [root@oldboy-m01 ~]# iptables -nL
38 Chain INPUT (policy DROP)
39 target      prot opt source                destination

40 ACCEPT      tcp  --  0.0.0.0/0              0.0.0.0/0
      tcp dpt:22
41 ACCEPT      tcp  --  0.0.0.0/0              0.0.0.0/0
      multiport dports 80,443
42
43 Chain FORWARD (policy ACCEPT)
44 target      prot opt source                destination

45
46 Chain OUTPUT (policy ACCEPT)
47 target      prot opt source                destination

48 [root@oldboy-m01 ~]# iptables-restore
</etc/sysconfig/iptables
49 [root@oldboy-m01 ~]# iptables -nL
50 Chain INPUT (policy DROP)
51 target      prot opt source                destination

52 ACCEPT      icmp --  0.0.0.0/0              0.0.0.0/0
      limit: avg 10/min burst 5
53 ACCEPT      tcp  --  0.0.0.0/0              0.0.0.0/0
      tcp dpt:22
54 ACCEPT      tcp  --  0.0.0.0/0              0.0.0.0/0
      multiport dports 80,443
55
```

```
56 Chain FORWARD (policy ACCEPT)
57 target      prot opt source                destination
58
59 Chain OUTPUT (policy ACCEPT)
60 target      prot opt source                destination
61
62 #@补充:
63 systemctl restart iptables      #读取
    /etc/sysconfig/iptables内容
64
65
66
67 [root@m01 ~]# iptables-save >
    /etc/sysconfig/iptables
68 [root@m01 ~]# iptables -F
69 [root@m01 ~]# iptables -nL
70 Chain INPUT (policy ACCEPT)
71 target      prot opt source                destination
72
73 Chain FORWARD (policy ACCEPT)
74 target      prot opt source                destination
75
76 Chain OUTPUT (policy ACCEPT)
77 target      prot opt source                destination
78
79 [root@m01 ~]# iptables-restore
    </etc/sysconfig/iptables
80 [root@m01 ~]# iptables -nL
81 Chain INPUT (policy ACCEPT)
```



```

81 target      prot opt source                destination

82 ACCEPT      tcp  --  0.0.0.0/0             0.0.0.0/0
      tcp dpt:22

83

84 Chain FORWARD (policy ACCEPT)

85 target      prot opt source                destination

86

87 Chain OUTPUT (policy ACCEPT)

88 target      prot opt source                destination

89

90

91

```

3.6.9 filter表小结

- 封ip 端口 网段 ❖❖❖❖❖
- 禁止ping ❖❖
- 限制速度和并发 ❖
- 防火墙规则的备份与恢复 ☆☆☆☆☆
- 补充：
 - iptables filter表 功能 可以在 云服务器使用
 - 云服务器应用：安全组控制端口，iptables控制ip

3.7 实际生产用法

- iptables配置方式
 - 逛公园模式：默认规则是 ACCEPT
 - 看电影模式：默认规则是 DROP 白名单模式
- 默认是拒绝 去电影院

1. ssh可以连接进来

```
1 iptables -F
2 iptables -X
3 iptables -Z
4 iptables -nL
5
6 iptables -A INPUT -p tcp --dport 22 -j ACCEPT
7
```

2. 1) 设置允许本机lo**通讯规则**

允许本机回环lo接口数据流量流出与流入

```
1 # -i 数据进入的时候
2 iptables -A INPUT -i lo -j ACCEPT
3 # -o 数据流出的时候
4 iptables -A OUTPUT -o lo -j ACCEPT
```

3. 配置默认规则及 放行 80 443端口

```
1
2
3
4 iptables -A INPUT -m multiport -p tcp --dport
  443,80 -j ACCEPT
5 [root@m01 ~]# iptables -nL
6 Chain INPUT (policy DROP)
7 target      prot opt source                destination
```

```
8 ACCEPT      tcp    --  0.0.0.0/0          0.0.0.0/0
      tcp dpt:22
9 ACCEPT      all    --  0.0.0.0/0          0.0.0.0/0
10 ACCEPT      tcp    --  0.0.0.0/0          0.0.0.0/0
      multiport dports 443,80
11
12 Chain FORWARD (policy DROP)
13 target      prot opt source                destination
14
15 Chain OUTPUT (policy ACCEPT)
16 target      prot opt source                destination
17 ACCEPT      all    --  0.0.0.0/0          0.0.0.0/0
18 [root@m01 ~]# iptables -A INPUT -s 10.0.0.0/24 -j
ACCEPT
19 [root@m01 ~]# iptables -A INPUT -s 172.16.1.0/24 -
j ACCEPT
20 #此处还可以添加 vpn网段 比如说 10.7.1.0/24
21
22 [root@m01 ~]# iptables -nL
23 Chain INPUT (policy DROP)
24 target      prot opt source                destination
25 ACCEPT      tcp    --  0.0.0.0/0          0.0.0.0/0
      tcp dpt:22
26 ACCEPT      all    --  0.0.0.0/0          0.0.0.0/0
27 ACCEPT      tcp    --  0.0.0.0/0          0.0.0.0/0
      multiport dports 443,80
```

```

28 ACCEPT      all  --  10.0.0.0/24          0.0.0.0/0
29 ACCEPT      all  --  172.16.1.0/24         0.0.0.0/0
30
31 Chain FORWARD (policy DROP)
32 target      prot opt source                destination

33
34 Chain OUTPUT (policy ACCEPT)
35 target      prot opt source                destination

36 ACCEPT      all  --  0.0.0.0/0            0.0.0.0/0

37 [root@m01 ~]# iptables-save
38 # Generated by iptables-save v1.4.21 on Wed Jul 24
   23:42:00 2019
39 *filter
40 :INPUT DROP [0:0]
41 :FORWARD DROP [0:0]
42 :OUTPUT ACCEPT [24:3008]
43 -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
44 -A INPUT -i lo -j ACCEPT
45 -A INPUT -p tcp -m multiport --dports 443,80 -j
   ACCEPT
46 -A INPUT -s 10.0.0.0/24 -j ACCEPT
47 -A INPUT -s 172.16.1.0/24 -j ACCEPT
48 -A OUTPUT -o lo -j ACCEPT
49 COMMIT
50 # Completed on Wed Jul 24 23:42:00 2019

```

修改默认的规则为拒绝INPUT

```
1 iptables -P INPUT DROP
2 iptables -P FORWARD ACCEPT
3 iptables -P OUTPUT ACCEPT
```

汇总

```
1 [root@oldboy-m01 ~]# iptables-save
2 # Generated by iptables-save v1.4.21 on Wed Feb 12
   15:51:48 2020
3 *filter
4 :INPUT DROP [0:0]
5 :FORWARD DROP [0:0]
6 :OUTPUT ACCEPT [1:60]
7 -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT      #放行
   22端口
8 -A INPUT -i lo -j ACCEPT                          #lo网
   卡放行
9 -A OUTPUT -o lo -j ACCEPT                         #lo网
   卡放行
10 -A INPUT -p tcp -m multiport --dports 443,80 -j
   ACCEPT      #放行80,443端口
11 -A INPUT -s 10.0.0.0/24 -j ACCEPT
   #放行10.0.0.0/24 和172.16.1.0/24网段
12 -A INPUT -s 172.16.1.0/24 -j ACCEPT
13 -A INPUT -s 10.7.1.0/24 -j ACCEPT
14 -A INPUT -m state --state RELATED,ESTABLISHED -j
   ACCEPT      #放行tcp连接状态
15 -A OUTPUT -m state --state RELATED,ESTABLISHED -j
   ACCEPT
16 COMMIT
17 # Completed on Wed Feb 12 15:51:48 2020
```

```
18 # Generated by iptables-save v1.4.21 on Wed Feb 12
    15:51:48 2020
19 *nat
20 :PREROUTING ACCEPT [2:458]
21 :INPUT ACCEPT [0:0]
22 :OUTPUT ACCEPT [417:25020]
23 :POSTROUTING ACCEPT [417:25020]
24 COMMIT
25 # Completed on Wed Feb 12 15:51:48 2020
26
```

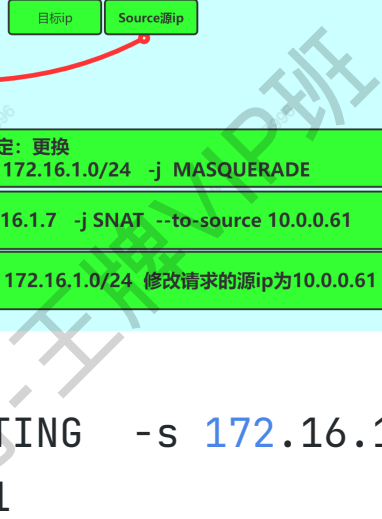
3.8 nat

nat表用于实现nat功能。nat网络地址转换。

共享上网
端口转发/端口映射
ip映射

3.8.1 实现共享上网*****

172.16.1.7 内网服务器实现通过iptables实现共享上网原理详解。



- ING -s 172.16.1.7 -j
- L
- 链
- 行共享上网,如果是多台(-s
- .
- p地址改为防火墙公网的ip地址

61 可以写为 `-j MASQUERADE` 伪装成公网ip.

eth0网卡关闭(ONBOOT=no)

eth0网卡关闭(ONBOOT)

1. 防火墙配置

- 1 配置防火墙规则,改为默认是准许.
- 2 清空其他规则.
- 3 配置防火墙共享上网规则.
- 4
- 5 `iptables -t nat -A POSTROUTING -s 172.16.1.0/24 -j SNAT --to-source 10.0.0.61`
- 6 防火墙上开启ip_forward功能(内核转发功能)
- 7 `echo 'net.ipv4.ip_forward = 1' >> /etc/sysctl.conf`
- 8 `sysctl -p`
- 9

注意事项:

公网ip不固定:

```
iptables -t nat -A POSTROUTING -s 172.16.1.0/24 -j MASQUERADE
```

2. web配置

关闭eth0网卡,仅开启eth1网卡,配置网关指向m01(172.16.1.61)

- 1 `[root@web01 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0`
- 2 `TYPE=Ethernet`
- 3 `BOOTPROTO=none`
- 4 `NAME=eth0`
- 5 `DEVICE=eth0`
- 6 `ONBOOT=no`
- 7 `IPADDR=10.0.0.7`
- 8 `PREFIX=24`
- 9 `GATEWAY=10.0.0.2`
- 10 `DNS1=223.5.5.5`
- 11 `[root@web01 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth1`

```
12 TYPE=Ethernet
13 IPADDR=172.16.1.7
14 PREFIX=24
15 NAME=eth1
16 DEVICE=eth1
17 ONBOOT=yes
18 GATEWAY=172.16.1.61
19 [root@web01 ~]# systemctl restart network
20 [root@m01 ~]# ssh 172.16.1.7
21 Last login: Wed Jul 24 23:06:58 2019 from 10.0.0.1
22 [root@web01 ~]# ip a
23 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc
    noqueue state UNKNOWN group default qlen 1000
24     link/loopback 00:00:00:00:00:00 brd
    00:00:00:00:00:00
25     inet 127.0.0.1/8 scope host lo
26         valid_lft forever preferred_lft forever
27     inet6 ::1/128 scope host
28         valid_lft forever preferred_lft forever
29 2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc
    pfifo_fast state DOWN group default qlen 1000
30     link/ether 00:0c:29:b2:e3:7e brd
    ff:ff:ff:ff:ff:ff
31 3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    qdisc pfifo_fast state UP group default qlen 1000
32     link/ether 00:0c:29:b2:e3:88 brd
    ff:ff:ff:ff:ff:ff
33     inet 172.16.1.7/24 brd 172.16.1.255 scope global
    eth1
34         valid_lft forever preferred_lft forever
35     inet6 fe80::20c:29ff:feb2:e388/64 scope link
36         valid_lft forever preferred_lft forever
37 [root@web01 ~]# ping baidu.com
```

```
38 PING baidu.com (220.181.38.148) 56(84) bytes of
   data.
39 64 bytes from 220.181.38.148 (220.181.38.148):
   icmp_seq=1 ttl=127 time=8.90 ms
40 64 bytes from 220.181.38.148 (220.181.38.148):
   icmp_seq=2 ttl=127 time=7.52 ms
41 64 bytes from 220.181.38.148 (220.181.38.148):
   icmp_seq=3 ttl=127 time=9.28 ms
42 64 bytes from 220.181.38.148 (220.181.38.148):
   icmp_seq=4 ttl=127 time=9.36 ms
43 ^C
44 --- baidu.com ping statistics ---
45 4 packets transmitted, 4 received, 0% packet loss,
   time 3006ms
46 rtt min/avg/max/mdev = 7.528/8.769/9.364/0.746 ms
47 [root@web01 ~]# ping 1.2.4.8
48 PING 1.2.4.8 (1.2.4.8) 56(84) bytes of data.
49 64 bytes from 1.2.4.8: icmp_seq=1 ttl=127 time=76.4
   ms
50 64 bytes from 1.2.4.8: icmp_seq=2 ttl=127 time=76.8
   ms
51 ^C
52 --- 1.2.4.8 ping statistics ---
53 2 packets transmitted, 2 received, 0% packet loss,
   time 1002ms
54 rtt min/avg/max/mdev = 76.440/76.637/76.834/0.197 ms
```

3. 完成后 在web01 发出 ip r和ping 外网ip的结果

```
1
2
3
```

```

4
5 [root@web01 ~]# ip r
6 default via 172.16.1.61 dev eth1
7 169.254.0.0/16 dev eth1 scope link metric 1003
8 172.16.1.0/24 dev eth1 proto kernel scope link src
  172.16.1.7
9 [root@web01 ~]# route -n
10 Kernel IP routing table
11 Destination          Gateway                Genmask
   Flags Metric Ref    Use Iface
12 0.0.0.0                172.16.1.61           0.0.0.0                UG
   0         0      0 eth1
13 169.254.0.0            0.0.0.0               255.255.0.0            U
   1003     0      0 eth1
14 172.16.1.0             0.0.0.0               255.255.255.0          U
   0         0      0 eth1
15 [root@web01 ~]# ping baidu.com
16 PING baidu.com (39.156.69.79) 56(84) bytes of data.
17 64 bytes from 39.156.69.79 (39.156.69.79):
   icmp_seq=1 ttl=127 time=21.7 ms
18 64 bytes from 39.156.69.79 (39.156.69.79):
   icmp_seq=2 ttl=127 time=32.6 ms
19 ^C
20 --- baidu.com ping statistics ---
21 2 packets transmitted, 2 received, 0% packet loss,
   time 1002ms
22 rtt min/avg/max/mdev = 21.781/27.214/32.647/5.433 ms

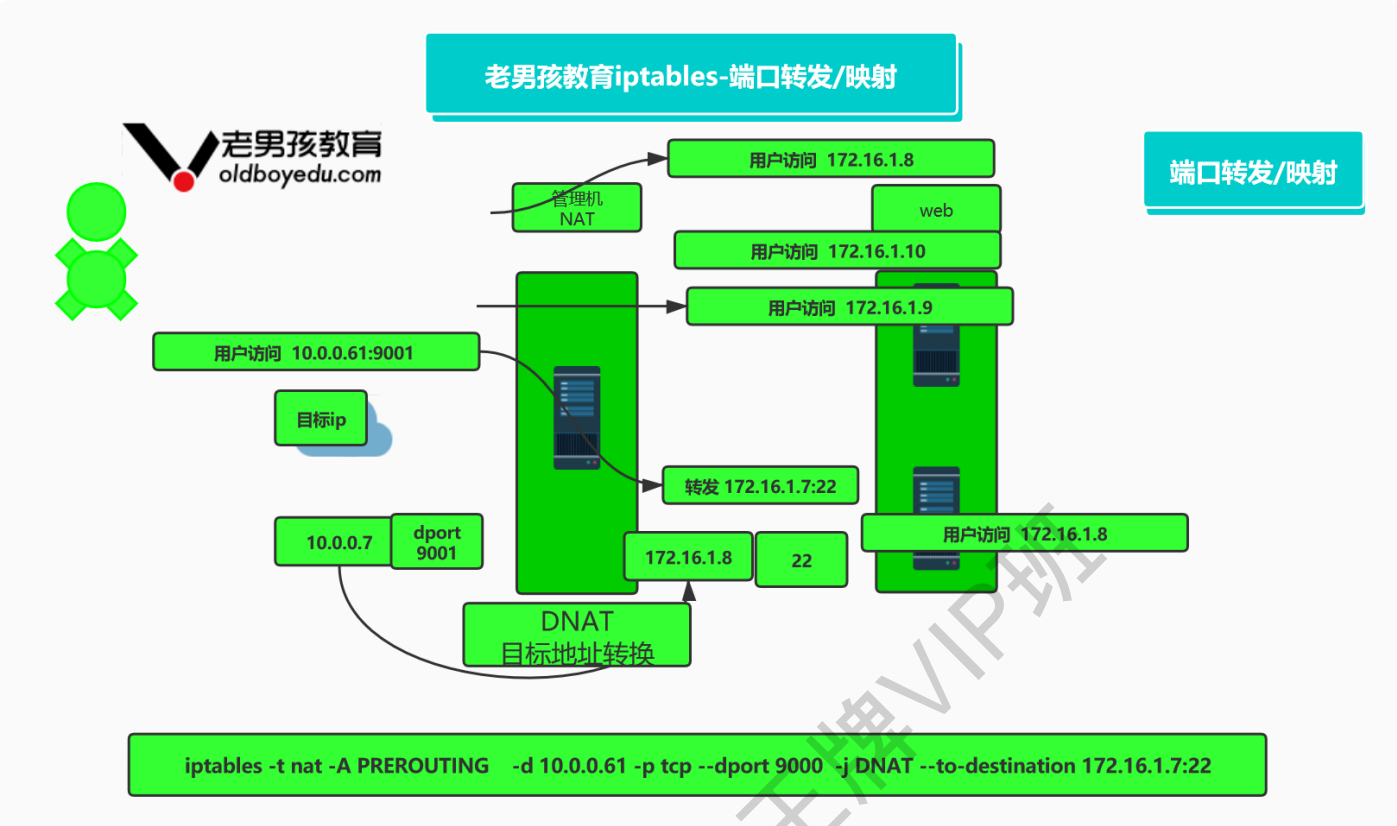
```

共享上网流程:

1. 防火墙添加规则 SNAT\规则 nat表 POSTROUTING
2. 防火墙服务器 开启ip转发功能
3. 后端节点,配置网卡,让网卡网关指向防火墙
4. 后端节点的网卡中配置DNS1=223.5.5.5 ,DNS2=223.6.6.6

3.8.2 实现端口转发*****

端口映射为了解决用户进来的问题。 外部的用户访问内网的某个服务器,端口 ..



映射传入端口

主机端口(H): 9000

类型: ☒ TCP(T) ☐ UDP(U)

虚拟机 IP 地址(A): 10 . 0 . 0 . 61

虚拟机端口(P): 80

描述(D):

确定 取消 帮助

```
1 [root@m01 ~]# iptables -t nat -A PREROUTING -d 10.0.0.61 -p tcp --dport 9000 -j DNAT --to-destination 172.16.1.7:22
2 [root@m01 ~]# iptables -nL -t nat
3 Chain PREROUTING (policy ACCEPT)
4 target      prot opt source                destination
```

```

5 DNAT          tcp -- 0.0.0.0/0           10.0.0.61
               tcp dpt:9000 to:172.16.1.7:22
6
7 Chain INPUT (policy ACCEPT)
8 target        prot opt source                destination
9
10 Chain OUTPUT (policy ACCEPT)
11 target        prot opt source                destination
12
13 Chain POSTROUTING (policy ACCEPT)
14 target        prot opt source                destination
15 SNAT          all -- 172.16.1.0/24         0.0.0.0/0
               to:10.0.0.61

```

测试与检查

本地shell中

```
1 [d:\~]$ ssh root@10.0.0.61 9000
```

3.8.3 实现ip映射

```

1 ip a add 10.0.0.62/24 dev eth0 label eth0:0
2
3 [root@m01 ~]# iptables -t nat -A PREROUTING -d
  10.0.0.62 -j DNAT --to-destination 172.16.1.7
4 [root@m01 ~]# iptables -nL -t nat
5 Chain PREROUTING (policy ACCEPT)
6 target        prot opt source                destination

```

```

7 DNAT      tcp -- 0.0.0.0/0          10.0.0.61
            tcp dpt:9000 to:172.16.1.7:22
8 DNAT      all -- 0.0.0.0/0          10.0.0.62
            to:172.16.1.7
9
10 Chain INPUT (policy ACCEPT)
11 target    prot opt source                destination
12
13 Chain OUTPUT (policy ACCEPT)
14 target    prot opt source                destination
15
16 Chain POSTROUTING (policy ACCEPT)
17 target    prot opt source                destination
18 SNAT      all -- 172.16.1.0/24      0.0.0.0/0
            to:10.0.0.61

```

3.8.4 nat表总结

*实现共享上网*****

端口转发 *****

nat功能在 云服务器无法使用 替代品叫: NAT网关

4. 总结

面试题: 防火墙4表伍链, 处理流程.

防火墙filter表, 禁用ip, 端口.

防火墙nat表实现: 共享上网, 端口映射.

防火墙备份与恢复.

练习题:

5、请写出查看iptables当前所有规则的命令。

1

6、禁止来自10.0.0.188 ip地址访问80端口的请求

1

7、如何在命令行执行的iptables规则永久生效?

1

8、实现把访问10.0.0.3:80的请求转到172.16.1.17:80

1

9、实现172.16.1.0/24段所有主机通过124.32.54.26外网IP共享上网。