图文详解防火墙及NAT服务

2017-03-30 10:22 | 阅读 5.8k | 评论 0

一、简介

1. 关于防火墙

防火墙，其实就是用于实现Linux下访问控制的功能的，它分为硬件和软件防火墙两种。无论是在哪个网络中，防火墙工作的地方一定是在网络的边缘。而我们的任务就是需要去定义到底防火墙如何工作，这就是防火墙的策略、规则，以达到让它对出入网络的IP、数据进行检测。

目前市面上比较常见的有三、四层的防火墙，叫做网络层的防火墙，还有七层的防火墙，其实是代理层的网关。对于TCP/IP的七层模型来讲，我们知道第三层是网络层，三层的防火墙会在这层对源地址和目标地址进行检测。但对于七层的防火墙，不管你源端口或者目标端口，源地址或者目标地址是什么，都将对你所有的东西进行检查。所以，对于设计原理来讲，七层防火墙更加安全，但是这却带来了效率更低。所以市面上通常的防火墙方案，都是两者相互结合的。

2. iptables的发展

包括iptables及其前身在内，这些都是工作在用户空间中，定义规则的工具，本身并不算是防火墙。它们定义的规则，并且可以让在内核空间当中的“Netfilter”来读取，从而实现让防火墙工作。而放入内核的地方必须要是特定的位置，必须是TCP/IP的协议栈所经过的地方——Netfilter。

iptables只是防火墙的管理工具，在内核中真正实现防火墙功能的是Netfilter。

对Linux而言，TCP/IP协议栈存在于内核当中，这就意味着对数据报文的处理是在内核中处理的，也就是说防火墙必须在工作在内核中，防火墙必须在内核中完成TCP/IP报文所流进的位置，使用规则去检查，才真正能工作起来。

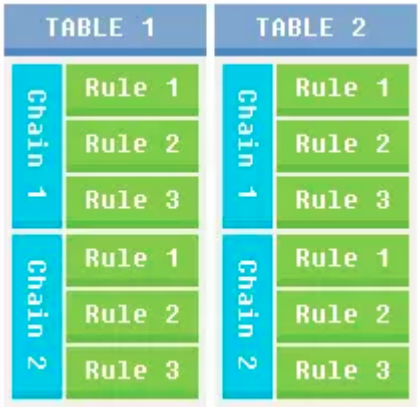
3. iptables的结构

从上面的发展我们知道了作者选择了五个位置，来作为控制的地方，但是你有没有发现，其实前三个位置已经基本上能将路径彻底封锁了，但是为什么已经在进出的口设置了关卡之后还要在内部设置关卡呢？由于数据包尚未进行路由决策，还不知道数据要走向哪里，所以在进出口是没有办法实现数据过滤的。所以要在内核空间里设置转发的关卡，进入用户空间的关卡，从用户空间出去的关卡。那么，既然他们没有什么用，我们为什么还要放置他们呢？因为在进行NAT/DNAT的情况下，目标地址转换必须在路由之前转换。所以我们必须在外网而后内网的接口处进行设置关卡。
Netfilter规定的这五个位置也叫五个规则链：

Chain / Table	PREROUTING	INPUT	FORWARDING	OUTPUT	POSTROUTING
Filter		○	○	○	
NAT	○			○	○
Mangle	○	○	○	○	○

规则链	说明
PREROUTING	在进行路由之前所要执行的规则。它会转换数据包中的目标IP地址，通常用于 DNAT 。注意：所有的数据包进入的时侯都先由这个链处理
INPUT	主要与要进入Linux主机的数据包有关，负责过滤所有目标地址是本机的数据包，即过滤进入主机的数据包。对防火墙来讲，这是一个非常重要的链。
FORWARD	与Linux本机关系不大，它可以将数据包进行转发，与NAT这个表有密切的关系。
OUTPUT	主要与Linux本机发送出的数据包有关。在生产环境中对出去的数据包并不十分关注。
POSTROUTING	在进行路由判断之后所要执行的规则。处理即将离开本机的数据包。它会转换数据包中的源IP地址，通常用于 SNAT 。注意：所有的数据包出来的时侯都先由这个链处理。

iptable的结构：在数据包过滤表中，规则被分组放在我们所谓的链中。链，就是一个规则的列表（如图所示）。



二、表和链

要设置一个Linux防火墙，就要使用规则，每个规则指定在包中与什么匹配，以及对包执行什么操作。那么什么是规则呢？因为iptables利用的是数据包过滤的机制，所以它会分析数据包的报头数据。根据报头数据与定义的规则来决定该数据包是否可以通过或者是被丢弃。也就是说，根据数据

包的分析资料来与预先定义的规则内容进行“比对”，若数据包数据与规则内容相匹配则进行相应的处理，否则就继续下一条规则的比对。重点在于比对与比对的顺序。

什么是表和链呢？这得由iptables的名称说起，为什么称为iptables呢？因为它里面包含有多个表格（table），每个表格都定义出自己的默认策略与规则，且每个表格的用途都不相同。iptables包含四个表，五个链。其中表是按照对数据包的处理功能区分的，链是按照不同的Hook点来区分的，表和链实际上是netfilter的两个维度。

四个规则表分别为：Filter、NAT、Mangle、Raw，默认表是Filter（没有指定表的时候就是Filter表）。 表的处理优先级为：Raw>Mangle>NAT>Filter

常用的三个表：

表	说明
Mangle	用于对特定数据包的修改（较少使用，目前不予关注）
NAT	用于网络地址转换功能（端口映射，地址映射等）
Filter	主要和主机自身有关，真正负责主机的防火墙功能。一般的过滤功能，是默认的表。对于Filter表的控制是我们实现防火墙功能的重要手段，特别是对INPUT的控制。

三、工作流程

iptables采用的是数据包过滤机制工作的，所以它会对数据包的报头信息进行分析，并根据我们预先设定的规则进行匹配来决定是否对数据包的处理方式。

防火墙是层层过滤的，实际是按照匹配规则的顺序从上到下，从前到后进行过滤的。如果匹配上规则，即明确表明是阻止还是通过，数据包就不在向下继续进行匹配了。如果规则中没有明确判断出处理结果，也就是说不匹配当前规则，那么就继续向下进行匹配，直到匹配默认的规则，得到最后的处理结果。所以说规则的顺序至关重要。

防火墙的默认规则是所有的规则均不匹配时，才会执行的规则。


```
#启动防火墙服务:
[root@LB-N1 ~]# /etc/init.d/iptables start
iptables: Applying firewall rules:                         [  OK  ]
[root@LB-N1 ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
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
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
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
--------------------------
#显示相关的内核模块:
[root@LB-N1 ~]# lsmod|egrep "nat|filter|ipt"
ipt_REJECT              2351  2
iptable_filter          2793  1
ip_tables              17831  1 iptable_filter
#如果没有加载相关的模块，可以通过下面的指令进行加载:
modprobe ip_tables
modprobe iptable_filter
modprobe iptable_nat
modprobe ip_conntrack
modprobe ip_conntrack_ftp
modprobe ip_nat_ftp
modprobe ipt_state
```

指令常用操作选项:

选项	长选项	说明
-t	--tables	用来指定表，当未指定规则表时，则一律视为是Filter表
-A	--append	新增规则到某个规则链中，该规则将会成为规则链中的最后一条规则
-D	--delete	从某个规则链中删除一条规则，可以输入完整规则，或直接指定规则编号加以删除。
-I	--insert	插入一条规则，原本该位置上的规则将会往后移动一个顺位
-R	--replace	取代现行规则，规则被取代后并不会改变顺序
-L	--list	列出某规则链中的所有规则
-S	--lis-rules	列出某规则链中的所有规则
-F	--flush	删除某规则链中的所有规则
-Z	--zero	将封包计数器归零，也就是将所有的记数与流量统计都归零。封包计数器是用来计算同一封包出现次数，是过滤阻断式攻击不可或缺的工具
-N	--new-chain	定义新的规则链
-X	--delete-chain	删除某个规则链
-P	--policy	定义过滤策略，也就是未符合过滤条件的数据包，默认的处理方式
-E	--rename-chain	修改某自订规则链的名称
-h	--help	显示帮助信息

常用封包比对参数：

选项	长选项	说明
-p	--protocol	对比通讯协议类型是否相符，可以使用<!>表示非，如果要对比所有类型，则可以使用<all>关键词
-s	--source	用来对比封包的来源IP，可以比对单机或网络，对比网络时用数字来表示掩码，对比IP时可以使用<!>运算子进行反向比对
-d	--destination	用来比对封包的目标IP，设定方式同上
-j	--jump	用来指定要进行的处理动作
-i	--in-interface	用来比对封包是从哪块网卡进入，可以使用通配字符“+”来做大范围比对，也可以使用“!”运算子进行反向比对
-o	--out-interface	用来比对封包要从哪块网卡送出，设定方式同上

其它选项：

选项	长选项	说明
-v	--verbose	列出更多的信息，包括通过该规则的数据包总位数、相关的网络接口等
-n	--numeric	不进行IP与hostname的反查，加快信息显示的速度
	--line-numbers	显示规则的行号，即规则在规则链中的序号或位

```
[root@LB-N1 ~]# iptables -F   #没有指定规则链，则清空所有的规则，但无法清除默认的规则。
[root@LB-N1 ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
...
```

操作实例：禁止SSH远程登录

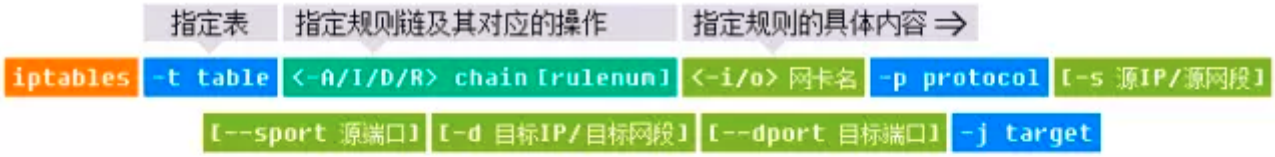
```
#因为是要针对『Filter』表，所以可以不指定-t选项，应为默认就是该表。
#另外需要注意，在实际的生产中，要考虑到管理员无法登陆的后果。
[root@LB-N1 ~]# ss -lntup|grep ssh
tcp    LISTEN     0      128                  :::22           :::*      users:(("sshd",922,4))
tcp    LISTEN     0      128                   *:22            *:*      users:(("sshd",922,3))
[root@LB-N1 ~]# iptables -t filter -A INPUT -p tcp --dport 22 -j DROP
--------------------------
#可以通过清除所有规则来清除上述规则：
[root@LB-N1 ~]# iptables -F
#或者可以直接删除对应的规则：
[root@LB-N1 ~]# iptables -t filter -D INPUT -p tcp --dport 22 -j DROP
```

注意：通过命令行添加的防火墙指令仅仅是临时生效的，系统重启即失效。

处理动作包括：

动作	说明
ACCEPT	接收数据包
REJECT	拒绝接收数据包，并传送封包通知对方，等于向外界表明，系统自身作为数据包的目标是存在的
DROP	丢弃数据包。丢弃并且无任何回应。从某种意义上将要优于 REJECT 处理
REDIRECT	重定向、映射、透明代理
SNAT	改写数据包源IP为某特定IP或IP范围。可以指定端口对应的范围
DNAT	改写数据包目的IP为某特定IP或IP范围，可以指定端口对应的范围
MASQUERADE	改写封包来源IP为防火墙IP，可以指定端口对应的范围。这个功能与SNAT略有不同，当进行IP伪装时，不需指定要伪装成哪个IP，IP会从网卡直接读，当使用拨接连线时，IP通常是由ISP公司的DHCP服务器指派的，这个时候 MASQUERADE 就特别有用了

指令格式示意图：




```
[root@LB-N1 ~]# iptables -t filter -A INPUT -p tcp --dport 80 -j DROP
[root@LB-N1 ~]# iptables -L -n --line-numbers
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:80
...
[root@LB-N1 ~]# iptables -t filter -D INPUT 1     #删除时，直接指定规则对应的序号即可，比较方便。
```

两种增加规则选项的差别：

-A chain rule-specification：添加规则到指定规则链的结尾，成为最后一条规则。

-I chain [rulenum] rule-specification：如果没有指定序号，则添加的规则将成为对应链中的第一条规则。如果指定了序号，则成为该序号上的规则，而原来位于该序号的规则将往后移一位。

```
#封掉8080端口，并将其插入到规则链中的第二行：
[root@LB-N1 ~]# iptables -I INPUT 2 -p tcp --dport 8080 -j DROP
------------------------
#阻止宿主机的数据包进入虚拟机，不但无法远程连接了，同时也PING不通了。
[root@LB-N1 ~]# iptables -t filter -I INPUT -i eth0 -s 172.16.1.1 -j DROP
#直接登录虚拟机，删除上面这条规则：
[root@LB-N1 ~]# iptables -t filter -D INPUT 1
------------------------
#也可以设定只有宿主机可以连接，其他主机都不能连接：
[root@LB-N1 ~]# iptables -t filter -A INPUT -i eth0 ! -s 172.16.1.1 -j DROP
```

操作实例：禁止PING本机（ping指令属于ICMP协议，其类型为“8”）

```
[root@LB-N1 ~]# iptables -t filter -I INPUT -p icmp --icmp-type 8 -i eth0 -s 10.0.0.0/24 -j DROP
#除了指定具体的类型，也可以将所有的类型全部封堵：
[root@LB-N1 ~]# iptables -t filter -A INPUT -p icmp -m icmp --icmp-type any -j DROP
```

```
#操作实例：封掉指定范围的端口
[root@LB-N1 ~]# iptables -I INPUT -p tcp --dport 5200:5500 -j DROP
[root@LB-N1 ~]# iptables -I INPUT -p tcp -m multiport --dport 21,22,23,24 -j ACCEPT
```

四、企业案例

1. 配置案例讲解

生产环境配置防火墙主要有两种模式：逛公园及看电影模式

- 逛公园模式：默认随便进出，对非法分子进行拒绝。企业应用：企业配置上网网关路由。
- 看电影模式：默认没票进不去，花钱买票才能够进入电影院。企业应用：服务器防火墙。

可以看出，还是第二种模式更加的严格和安全。其本质区别就是防火墙的默认规则是允许还是拒绝。

```
#配置一个最安全最严格的企业防火墙:
[root@LB-N1 ~]# iptables -F
[root@LB-N1 ~]# iptables -X
[root@LB-N1 ~]# iptables -Z
#第一步：首先要允许自己内部的人能够通过SSH进行远程登录连接。
iptables -A INPUT -p tcp  --dport 22 -s 10.0.0.0/24  -j ACCEPT
#第二步：其次，允许本机回环地址通信连接。
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
#第三步：重点操作，设置默认规则
[root@LB-N1 ~]# iptables -P INPUT DROP
[root@LB-N1 ~]# iptables -P OUTPUT ACCEPT
[root@LB-N1 ~]# iptables -P FORWARD DROP
#第四步：允许信任的网段进入
iptables -A INPUT -s 124.43.62.96/27 -p all -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -p all -j ACCEPT
iptables -A INPUT -s 10.0.0.0/24 -p all -j ACCEPT
iptables -A INPUT -s 203.83.24.0/24 -p all -j ACCEPT
iptables -A INPUT -s 201.82.34.0/24 -p all -j ACCEPT
#第五步：允许业务服务端口对外访问，例如允许HTTP服务无条件通过。
iptables -A INPUT -p tcp  --dport 80 -j ACCEPT
#第六步：允许外部能够PING通主机（非必要的操作）。
iptables -A INPUT -p icmp --icmp-type 8  -j ACCEPT
#第七步：允许关联的状态包
iptables -A INPUT  -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
----------------------
#我们上面配置的所有规则还都只存在于内存中，要进行保存，防止重启丢失。
[root@LB-N1 ~]# /etc/init.d/iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[  OK  ]
#还可以使用:
iptables-save >/etc/sysconfig/iptables
```

企业面试题：自定义链处理“syn”攻击

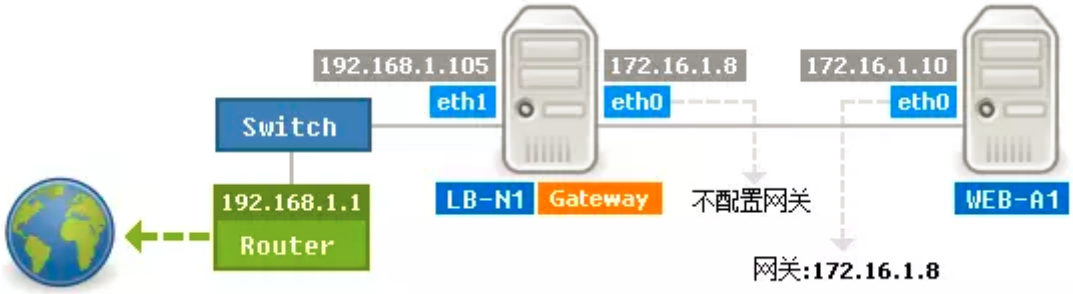
```
iptables -N syn-flood
iptables -A INPUT -i eth0 -syn -j syn-flood
iptables -A syn-flood -m limit -limit 5000/s -limit-burst 200 -j RETURN
iptables -A syn-flood -j DROP
```

2. 工作中如何维护防火墙

实际生产中，一般第一次添加规则是以命令行或者脚本的方式进行，然后一次性的保存成配置文件，之后的维护工作就是围绕着对该配置文件的修来来进行。

```
[root@LB-N1 ~]# vim /etc/sysconfig/iptables
# Generated by iptables-save v1.4.7 on Thu Jan  8 17:17:42 2015
*filter
:INPUT ACCEPT [25:1676]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [16:2464]
-A INPUT ! -s 172.16.1.1/32 -p tcp -m tcp --dport 22 -j DROP
COMMIT
# Completed on Thu Jan  8 17:17:42 2015
--------------------
#修改完配置文件后可以重新加载以使之生效:
[root@LB-N1 ~]# /etc/init.d/iptables reload
iptables: Trying to reload firewall rules:                 [  OK  ]
--------------------
#发现一个问题，如果存下面的两行配置的话，宿主机（172.16.1.1）是无法SSH远程连接的:
-A INPUT ! -s 172.16.1.1/32 -p tcp -m tcp --dport 22 -j DROP
-A INPUT ! -s 172.16.1.5/32 -p tcp -m tcp --dport 22 -j DROP
#可以好好理解一下规则的匹配：实际上真实机被拦截在第二条规则上，而第一条规则实际上并未匹配上。
```

3. 配置网关



第一步：首先，作为网关的主机除了要具备双网卡并且能够连接互联网等物理条件外，还要确保将内核的转发功能打开。
另外，还要求Filter表的“FORWARD”链允许通过。

```
[root@LB-N1 ~]# vim /etc/sysctl.conf
net.ipv4.ip_forward = 1
----------------
#使修改立即生效:
[root@LB-N1 ~]# sysctl -p
----------------
[root@LB-N1 ~]# iptables -P INPUT ACCEPT
[root@LB-N1 ~]# iptables -P FORWARD ACCEPT
[root@LB-N1 ~]# iptables -L -n|grep FORWARD
Chain FORWARD (policy ACCEPT)
#不需要Filter防火墙功能，共享上网，因此最好暂时关闭防火墙进行测试。
[root@LB-N1 ~]# /etc/init.d/iptables stop
```

第二步：确保网关主机的相关模块已经加载

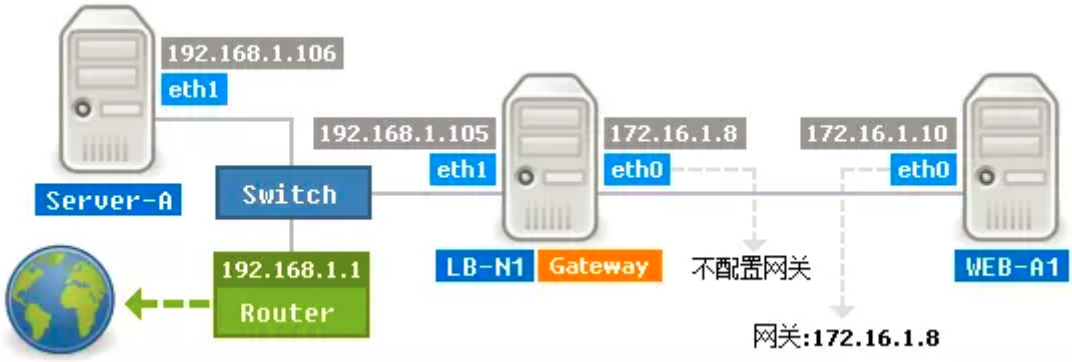
```
[root@LB-N1 ~]# lsmod|egrep ^ip
ipt_REJECT              2351  0
ipv6                  317340  136
#载入模块:
modprobe ip_tables
modprobe iptable_filter
modprobe iptable_nat
modprobe ip_conntrack
modprobe ip_conntrack_ftp
modprobe ip_nat_ftp
modprobe ipt_state
```

第三步：内网服务器要能够Ping通网关主机的内外网卡。
第四步：在网关主机上配置规则（两种方法）。

```
#方法一：适合于有固定外网地址
iptables -t nat -A POSTROUTING -s 172.16.1.0/24 -o eth1 -j SNAT --to-source 192.168.1.105
#方法二：适合变化外网地址（ADSL）：
iptables -t nat -A POSTROUTING -s 172.16.1.0/24 -j MASQUERADE
----------------
[root@LB-N1 ~]# /etc/init.d/iptables start
```

至此，Linux网关主机配置完毕。

还有一种应用，就是把外部IP地址及端口映射到内部服务器的地址及端口（和共享上网的环境一样）。



要求：


```
[root@LB-N1 ~]# iptables -t nat -A PREROUTING -d 192.168.1.0/24 -p tcp --dport 80 -j DNAT --to-destination 172.16.1.10:80
[root@LB-N1 ~]# iptables -L -n -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       tcp  --  0.0.0.0/0            192.168.1.0/24      tcp dpt:80 to:172.16.1.10:80
Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  172.16.1.0/24        0.0.0.0/0           to:192.168.1.105
...
----------------
[root@Server-A ~]# curl 192.168.1.105:80
bbs.etiantian.org@Apache 172.16.1.10
```

处理	参数	说明
SNAT	--to-source ipaddr[-ipaddr][:port[-port]]	意思是将 -s 后面的地址及端口，转换成--to后面的地址及端口，即对数据包的源IP进行转换。
DNAT	--to-destination ipaddr[-ipaddr][:port[-port]]	同上，只是将目的地址及端口进行转换。

企业应用场景：

- 把访问外网IP及端口的请求映射到内网某台服务器的地址及指定端口上（企业内部）。
- 硬件防火墙，把访问LVS/Nginx外网VIP及80端口的请求映射到IDC负载均衡服务器内部IP及指定端口上（IDC机房的操作）

iptables在企业中的应用小结：

- Linux主机防火墙（表：Filter
- 最为内网共享上网的网关（表：NAT，链：POSTROUTING）
- 由外到内的端口映射（表：NAT，链：PREROUTING）

```
#映射多个外网IP地址上网:
iptables -t nat -A POSTROUTING -s 10.0.0.0/255.255.240.0 -o eth0 -j SNAT --to-source 124.42.60.11-124.42.60.16
iptables -t nat -A POSTROUTING -s 172.16.1.0/255.255.255.0 -o eth0 -j SNAT --to-source 124.42.60.103-124.42.60.106
```

指定地址段

```
iptables -A INPUT -m iprange --src-range 192.168.30.25-192.168.30.48 -j ACCEPT
```

4. 端口映射

```
iptables -t nat -A PREROUTING -d 192.168.30.102 -p tcp -m tcp --dport 8090 -j DNAT --to-destination 192.168.30.102:8080
```

连接跟踪表已满，开始丢包的解决办法：

一、关闭防火墙。 简单粗暴，直接有效

二、加大防火墙跟踪表的大小，优化对应的系统参数