2-iptables

作者：Carey
原文链接：https://ld246.com/article/1611572894172
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20191211.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h2 id="3-iptables">3 iptables</h2>
<h2 id="3-1-iptables-规则说明">3.1 iptables 规则说明</h2>
<h3 id="3-1-1-iptables-规则组成">3.1.1 iptables 规则组成</h3>
规则 rule：根据规则的匹配条件尝试匹配报文，对匹配成功的报文根据规则定义的处
动作作出处理，规则在链接上的次序即为其检查时的生效次序
匹配条件：默认为与条件，同时满足
基本匹配：IP，端口，TCP 的 Flags（SYN,ACK 等）
扩展匹配：通过复杂高级功能匹配
处理动作：称为 target，跳转目标

内建处理动作 ACCEPT（允许）,DROP（抛弃）,REJECT（拒绝）,SNAT,D
AT,MASQUERADE,MARK,LOG...
自定义处理动作：自定义 chain，利用分类管理复杂情形

规则要添加在链上，才生效；添加在自定义链上不会自动生效
白名单:只有指定的特定主机可以访问,其它全拒绝
黑名单:只有指定的特定主机拒绝访问,其它全允许,默认方式
<h3 id="3-1-2-iptables规则添加时考量点">3.1.2 iptables 规则添加时考量点</h3>

要实现哪种功能：判断添加在哪张表上
报文流经的路径：判断添加在哪个链上
报文的流向：判断源和目的
匹配规则：业务需要

<h3 id="3-1-3-环境准备">3.1.3 环境准备</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">Centos7，8：
systemctl disable
-now firewalld

Centos6：
service stop iptabl
s
chkconfig iptables
off
</code></pre>
<h2 id="3-2-iptables用法说明">3.2 iptables 用法说明</h2>
帮助：man 8 iptables
格式：
<pre><code class="highlight-chroma"><span class="highlight
cl">iptables [-t table] {-A|-C|-D} chain rule-specification
iptables [-t table] -
 chain [rulenum] rule-specification
iptables [-t table]
R chain rulenum rule-specification
iptables [-t table]
D chain rulenum
iptables [-t table]
S [chain [rulenum]]
iptables [-t table] {
F|-L|-Z} [chain [rulenum]] [options...]
iptables [-t table]

N chain
iptables [-t table]
X [chain]
iptables [-t table]
P chain target
iptables [-t table]
E old-chain-name new-chain-name
rule-specification
= [matches...] [target]
match = -m matc
name [per-match-options]
target = -j targetn
me [per-target-options]
</code></pre>
范例：Filter 表中 INPUT 规则
<img src="https://ld246.com/images/img-loading.svg" alt="clipboard.png" data-src="ht
ps://b3logfile.com/file/2021/01/clipboard-6aebf24a.png?imageView2/2/interlace/1/format/j
g">
iptables 命令格式详解：
<pre><code class="highlight-chroma"><span class="highlight
cl">iptables [-t table] SUBCOMMAND chain [-m matchname [per-match-options]]
-j targetname [per
target-options]
</code></pre>

-t table：指定表

raw, mangle, nat, [filter]默认

<ol start="2">
SUBCOMMAND：子命令

<h3 id="3-2-1-链管理类-">3.2.1 链管理类：</h3>
<pre><code class="highlight-chroma"><span class="highlight
cl">-N：new, 自定义一条新的规则链
-E：重命名自定义
；引用计数不为0的自定义链不能够被重命名，也不能被删除
-X：delete，删除
定义的空的规则链
-P：Policy，设置
认策略；对filter表中的链而言，其默认策略有：ACCEPT：接受, DROP：丢弃
</code></pre>
范例：自定义链的使用，正常很少使用
<pre><code class="highlight-chroma"><span class="highlight
cl">[14:32:04 root@centos8 ~]#iptables -N web_chain #创建链，不写表名称默认filter表
[14:35:43 root@c
ntos8 ~]#iptables -N web_chain -t nat #把链关联到表
[14:36:55 root@c
ntos8 ~]#iptables -E web_chain WEB_CHAIN #修改链名称
[14:38:02 root@c
ntos8 ~]#iptables -A WEB_CHAIN -s 192.168.10.71 -p tcp -m multiport --dports 80,443,8080 -j
REJECT
[14:43:15 root@c

```
ntos8 ~]#iptables -vnL WEB_CHAIN
[14:43:48 root@centos8 ~]#iptables -AINPUT -j WEB_CHAIN
[14:44:38 root@centos8 ~]#iptables -X WEB_CHAIN
[14:46:24 root@centos8 ~]#iptables -F WEB_CHAIN
[14:47:14 root@centos8 ~]#iptables -X WEB_CHAIN
[14:47:53 root@centos8 ~]#iptables -D INPUT 1
[14:48:17 root@centos8 ~]#iptables -X WEB_CHAIN
```

3.2.2 查看类

```
-L：list, 列出指定鏈上的所有规则，本选项须置后
-n：numberic，数字格式显示地址和端口号
-v：verbose，详信息
-vv 更详细
-x：exactly，显示数器结果的精确值,而非单位转换后的易读值
--line-numbers：示规则的序号
-S selected,以iptables-save 命令格式显示链上规则
```

常用组合：

```
[17:49:50 root@centos7 ~]#iptables -nvL
[17:49:13 root@centos7 ~]#iptables -vnL --line-numbers
```

3.2.3 规则管理类

```
-A：append，追加
-I：insert, 插入，指明插入至的规则编号，默认为第一条
-D：delete，删除
(1) 指明规则序号
(2) 指明规则本身
-R：replace，替指定链上的指定规则编号
-F：flush，清空指的规则链
-Z：zero，置零
iptables的每条规都有两个计数器
(1) 匹配到的报文个数
(2) 匹配到的所有文的大小之和
```

</code></pre>
范例：
<pre><code class="highlight-chroma"><span class="highlight
cl">[17:53:31 root@centos8 ~]#iptables -F OUTPUT
</code></pre>
<ol start="3">
chain（链）

<pre><code class="highlight-chroma"><span class="highlight
cl">PREROUTING，INPUT，FORWARD，OUTPUT，POSTROUTING
</code></pre>
<ol start="4">
匹配条件

基本：通用的，PARAMETERS
扩展：需加载模块，MATCH EXTENTIONS

<ol start="5">
处理动作

<pre><code class="highlight-chroma"><span class="highlight
cl">-j targetname [per-target-options]
</code></pre>
简单动作：
<pre><code class="highlight-chroma"><span class="highlight
cl">ACCEPT　拒绝，会给对方发送回应拒绝包
DROP　　丢弃，
会给对方返回任何包
</code></pre>
扩展动作：
<pre><code class="highlight-chroma"><span class="highlight
cl">REJECT：--reject-with:icmp-port-unreachable默认
RETURN：返回调
链
REDIRECT：端口
定向
LOG：记录日志，
mesg
MARK：做防火墙
记
DNAT：目标地址
换
SNAT：源地址转换
MASQUERADE：
址伪装
自定义链
</code></pre>
<h2 id="3-3-iptables-基本匹配条件">3.3 iptables 基本匹配条件</h2>
基本匹配条件：无需加载模块，由 iptables/netfilter 自行提供
<pre><code class="highlight-chroma"><span class="highlight
cl">[!] -s, --source address[/mask][,...]：源IP地址或者不连续的IP地址
[!] -d, --destinatio
 address[/mask][,...]：目标IP地址或者不连续的IP地址

[!] -p, --protocol protocol：指定协议，可使用数字如0（all）
protocol: tcp, udp, icmp, icmpv6, udplite,esp, ah, sctp, mh or “all “
参看：/etc/protocols
[!] -i, --in-interface name：报文流入的接口；只能应用于数据报文流入环节，只应用于INPUT、FORWARD、PREROUTING链
[!] -o, --out-interface name：报文流出的接口；只能应用于数据报文流出的环节，只应用于FORWARD、OUTPUT、POTROUTING链
</code></pre>
范例：
<pre><code class="highlight-chroma">[18:18:02 root@centos8 ~]#iptables -AINPUT -s 192.168.10.81 -j REJECT
[18:18:38 root@centos8 ~]#iptables -IINPUT -i lo -j ACCEPT
[18:20:27 root@centos8 ~]#curl 127.0.0.1
zhangzhuo.org
[18:20:50 root@centos8 ~]#curl 192.168.10.81
zhangzhuo.org
</code></pre>
<h2 id="3-4-iptables-扩展匹配条件">3.4 iptables 扩展匹配条件</h2>
扩展匹配条件：需要加载扩展模块（/usr/lib64/xtables/*.so），方可生效
扩展模块的查看帮助： man iptables-extensions
扩展匹配条件：

隐式扩展
显式扩展

<h3 id="3-4-1-隐式扩展">3.4.1 隐式扩展</h3>
iptables 在使用-p 选项指明了特定的协议时，无需再用-m 选项指明扩展模块的扩展机制，不需手动加
载扩展模块
tcp 协议的扩展选项
<pre><code class="highlight-chroma">[!] --source-port, --sport port[:port]：匹配报文源端口,可为端口连续范围
[!] --destination-port,--dport port[:port]：匹配报文目标端口,可为连续范围
[!] --tcp-flags mask comp
mask 需检查的标位列表，用,分隔 , 例如 SYN,ACK,FIN,RST
comp 在mask列中必须为1的标志位列表，无指定则必须为0，用,分隔tcp协议的扩展选项
</code></pre>
范例:
<pre><code class="highlight-chroma">--tcp-flags SYN,ACK,FIN,RST SYN 表示要检查的标志位为SYN,ACK,FIN,RST四个，其中SYN必
须为1，余下的必为0，第一次握手
</code></pre>

--tcp-flags SYN,AK,FIN,RST SYN,ACK 第二次握手
#错误包
--tcp-flags ALL AL

--tcp_flags ALL NNE
</code></pre>
[!] --syn：用于匹配第一次握手, 相当于：--tcp-flags SYN,ACK,FIN,RST SYN
udp 协议的扩展选项
<pre><code class="highlight-chroma">[!] --source-port, --sport port[:port]：匹配报文的源端口或端口范围
[!] --destination-prt,--dport port[:port]：匹配报文的目标端口或端口范围
</code></pre>
icmp 协议的扩展选项
<pre><code class="highlight-chroma">[!] --icmp-type {type[/code]|typename}
type/code
0/0   echo-reply icmp应答
8/0   echo-request icmp请求
</code></pre>
范例：禁止到本机访问 http 服务 80 端口
<pre><code class="highlight-chroma">[18:35:42 root@centos8 ~]#iptables -A INPUT -d 192.168.10.81 -p tcp --dport 80 -j REJEC

[18:36:19 root@centos8 ~]#iptables -vnL --line-numbers
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target   prot opt in   out   source   destination
1   1   60 REJECT   tcp -- *   *   0.0.0.0/0   192.168.10.81   tcp dpt:80 reject-with icmp-port-unreachable

Chain FORWARD policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target   prot opt in   out   source   destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target   prot opt in   out   source   destination
</code></pre>
范例：匹配第一次握手的请求拒绝
<pre><code class="highlight-chroma">[18:39:23 root@centos8 ~]#iptables -AINPUT -p tcp --syn -j REJECT
</code></pre>
范例：拒绝 ICMP 的请求包

<pre><code class="highlight-chroma">[18:40:45 root@centos8 ~]#iptables -AINPUT -s 192.168.10.71 -p icmp --icmp-type 8 -j R JECT
</code></pre>
<h3 id="3-4-2-显示扩展及相关模块">3.4.2 显示扩展及相关模块</h3>
显示扩展即必须使用-m 选项指明要调用的扩展模块名称，需要手动加载扩展模块</st ong>
<pre><code class="highlight-chroma">[-m matchname [per-match-options]]
</code></pre>
扩展模块的使用帮助：

CentOS 7,8: man iptables-extensions
CentOS 6: man iptables

<h4 id="3-4-2-1-multiport扩展">3.4.2.1 multiport 扩展</h4>
以离散方式定义多端口匹配,最多指定 15 个端口
<pre><code class="highlight-chroma">#指定多个源端口
[!] --source-ports, -sports port[,port|,port:port]...
指定多个目标端

[!] --destination-p rts,--dports port[,port|,port:port]...
#多个源或目标端
[!] --ports port[,po t|,port:port]
</code></pre>
范例：禁止多个端口访问
<pre><code class="highlight-chroma">[18:44:47 root@centos8 ~]#iptables -A INPUT -s 192.168.10.0/24 -d 192.168.10.1 -p tcp m multiport --dports 20:22,80 -j REJECT
[18:46:27 root@c ntos8 ~]#iptables -AINPUT -s 192.168.10.71 -p tcp -m multiport --dports 445,139 -j REJECT
</code></pre>
<h4 id="3-4-2-2-iprange扩展">3.4.2.2 iprange 扩展</h4>
指明连续的（但一般不是整个网络）ip 地址范围
<pre><code class="highlight-chroma">[!] --src-range from[-to] 源IP地址范围
[!] --dst-range fr m[-to] 目标IP地址范围
</code></pre>
范例：禁止 192.168.10.71-192.168.10.72 访问本机的 80 端口
<pre><code class="highlight-chroma">[18:50:13 root@centos8 ~]#iptables -AINPUT -d 192.168.10.81 -p tcp --dport 80 -m ipran e --src-range 192.168.10.71-192.168.10.75 -j DROP
</code></pre>
<h4 id="3-4-2-3-mac扩展">3.4.2.3 mac 扩展</h4>
mac 模块可以指明源 MAC 地址,，适用于：PREROUTING, FORWARD，INPUT chains
<pre><code class="highlight-chroma">[!] --mac-source XX:XX:XX:XX:XX:XX
</code></pre>
范例：禁止地址为 192.168.10.71 且 mac 地址为 00:0c:29:c1:8a:39 访问本机</stron

>
<pre><code class="highlight-chroma"><span class="highlight
cl">[18:52:25 root@centos8 ~]#iptables -AINPUT -s 192.168.10.71 -m mac --mac-source 00:0
:29:c1:8a:39 -j RE
JECT
</code></pre>
<h4 id="3-4-2-4-string扩展">3.4.2.4 string 扩展</h4>
对报文中的应用层数据做字符串模式匹配检测
<pre><code class="highlight-chroma"><span class="highlight
cl">--algo {bm|kmp} 字符串匹配检测算法
bm：Boyer-Moor

kmp：Knuth-Pratt
Morris
--from offset 开
偏移
--to offset 结束
移
[!] --string pattern
要检测的字符串模式
[!] --hex-string pat
ern要检测字符串模式，16进制格式
</code></pre>
范例：
<pre><code class="highlight-chroma"><span class="highlight
cl">[18:59:35 root@centos8 ~]#iptables -AOUTPUT -ptcp --sport 80 -m string --algo bm --fr
m 62 --string "zhangzhuo" -j REJECT
</code></pre>
<h4 id="3-4-2-5-time扩展">3.4.2.5 time 扩展</h4>
注意：CentOS 8 此模块有问题
根据将报文到达的时间与指定的时间范围进行匹配
<pre><code class="highlight-chroma"><span class="highlight
cl">--datestart YYYY[-MM[-DD[Thh[:mm[:ss]]]]] 日期
--datestop YYYY[
MM[-DD[Thh[:mm[:ss]]]]]
--timestart hh:mm
:ss] 时间
--timestop hh:mm
:ss]
[!] --monthdays d
y[,day...] 每个月的几号
[!] --weekdays day
,day...] 星期几，1 – 7 分别表示星期一到星期日
--kerneltz：内核
区（当地时间），不建议使用，CentOS 7版本以上系统默认为 UTC
注意： centos6
支持kerneltz ， --localtz指定本地时区(默认)
</code></pre>
范例: CentOS 8 的 time 模块问题
<pre><code class="highlight-chroma"><span class="highlight
cl">[root@centos8 ~]#rpm -ql iptables |grep time
/usr/lib64/xtables/
ibxt_time.so
[root@centos8 ~]

iptables -A INPUT -m time --timestart 12:30 --timestop 13:30 -j
ACCEPT
iptables v1.8.4 (nf_
ables): Couldn't load match `time':No such file or
directory
</code></pre>
范例：
<pre><code class="highlight-chroma"><span class="highlight
cl">[19:06:48 root@centos7 ~]#iptables -IINPUT -s 192.168.10.0/24 -p icmp --icmp-type 8 -m
time --timestart 11:00 --timestop 20:00 -j DROP
</code></pre>
<h4 id="3-4-2-6-connlimit扩展">3.4.2.6 connlimit 扩展</h4>
根据每客户端 IP 做并发连接数数量匹配
可防止 Dos(Denial of Service，拒绝服务)攻击
<pre><code class="highlight-chroma"><span class="highlight
cl">--connlimit-upto N #连接的数量小于等于N时匹配
--connlimit-above
N #连接的数量大于N时匹配
</code></pre>
范例:
<pre><code class="highlight-chroma"><span class="highlight
cl">[13:43:54 root@centos8 ~]#iptables -A INPUT -d 192.168.10.81 -p tcp --dport 22 -m conn
limit --connlimit-above 2 -j REJECT
</code></pre>
<h4 id="3-4-2-7-limit扩展">3.4.2.7 limit 扩展</h4>
基于收发报文的速率做匹配，令牌桶过滤器
<pre><code class="highlight-chroma"><span class="highlight
cl">--limit-burst number #前多少个包不限制
--limit #[/second
秒）|/minute（分）|/hour（时）|/day（天）] #前面加数字表示每多少过多少个包
</code></pre>
范例：
<pre><code class="highlight-chroma"><span class="highlight
cl">[13:46:19 root@centos8 ~]#iptables -A INPUT -p icmp -m limit --limit-burst 10 --limit 20
minute -j ACCEPT
[13:48:22 root@c
ntos8 ~]#iptables -AINPUT -p icmp -j REJECT
[13:48:34 root@c
ntos8 ~]#ping 192.168.10.81
</code></pre>
<h4 id="3-4-2-8-state扩展-重点-">3.4.2.8 state 扩展（重点）</h4>
state 扩展模块，可以根据”连接追踪机制“去检查连接的状态，较耗资源
conntrack 机制：追踪本机上的请求和响应之间的关系
状态类型

NEW：新发出请求；连接追踪信息库中不存在此连接的相关信息条目，因此，将其识为第一次发出的请求
ESTABLISHED：NEW 状态之后，连接追踪信息库中为其建立的条目失效之前期间内进行的通信状态
RELATED：新发起的但与已有连接相关联的连接，如：ftp 协议中的数据连接与命令接之间的关系
INVALID：无效的连接，如 flag 标记不正确
UNTRACKED：未进行追踪的连接，如：raw 表中关闭追踪

已经追踪到的并记录下来的连接信息库
<pre><code class="highlight-chroma"><span class="highlight
cl">[14:12:14 root@centos8 ~]#cat /proc/net/nf_conntrack
ipv4   2 tcp   6
6 CLOSE_WAIT src=192.168.10.81 dst=192.168.10.82 sport=22 dport=49282 src=192.168.10.
2 dst=192.168.10.81 sport=49282 dport=22 [ASSURED] mark=0 zone=0 use=2
ipv4   2 tcp   6
99 ESTABLISHED src=192.168.10.81 dst=192.168.10.1 sport=22 dport=57028 src=192.168.10.
 dst=192.168.10.81 sport=57028 dport=22 [ASSURED] mark=0 zone=0 use=2
</code></pre>
调整连接追踪功能所能够容纳的最大连接数量
<pre><code class="highlight-chroma"><span class="highlight
cl">[14:12:36 root@centos8 ~]#cat /proc/sys/net/netfilter/nf_conntrack_max
30720
[14:13:42 root@c
ntos8 ~]#cat /proc/sys/net/nf_conntrack_max
30720
</code></pre>
查看连接跟踪有多少条目
<pre><code class="highlight-chroma"><span class="highlight
cl">[14:13:54 root@centos8 ~]#cat /proc/sys/net/netfilter/nf_conntrack_count
1
</code></pre>
不同的协议的连接追踪时长
<pre><code class="highlight-chroma"><span class="highlight
cl">[14:15:16 root@centos8 ~]#ll /proc/sys/net/netfilter/
</code></pre>
说明：

连接跟踪，需要加载模块： modprobe nf_conntrack_ipv4
当服务器连接多于最大连接数时 dmesg 可以观察到：kernel: ip_conntrack: table full, droppi
g packet 错误,并且导致建立 TCP 连接很慢。
各种状态的超时后，链接会从表中删除

范例：面试题
<pre><code class="highlight-chroma"><span class="highlight
cl">[14:17:21 root@centos8 ~]#echo 1 >/proc/sys/net/netfilter/nf_conntrack_max
[14:17:51 root@c
ntos8 ~]#tail -f /var/log/messages
Jan 24 14:18:05 c
ntos8 kernel: nf_conntrack: nf_conntrack: table full, dropping packet
[root@centos6 ~]
tail /var/log/messages
Jul 8 09:51:16 cen
os6 kernel: nf_conntrack: table full, dropping packet.
</code></pre>
连接过多的解决方法两个：
(1) 加大 nf_conntrack_max 值
<pre><code class="highlight-chroma"><span class="highlight
cl">vi /etc/sysctl.conf
net.nf_conntrack
max = 393216
net.netfilter.nf_co
ntrack_max = 393216

</code></pre>
(2) 降低 nf_conntrack timeout 时间
<pre><code class="highlight-chroma">vi /etc/sysctl.conf
net.netfilter.nf_conntrack_tcp_timeout_established = 300
net.netfilter.nf_conntrack_tcp_timeout_time_wait = 120
net.netfilter.nf_conntrack_tcp_timeout_close_wait = 60
net.netfilter.nf_conntrack_tcp_timeout_fin_wait = 120
iptables -t nat -L -

</code></pre>
格式：
<pre><code class="highlight-chroma">[!] --state state
</code></pre>
范例: 不允许 192.168.82 访问本机,但本机可以访问 192.168.10.82
<pre><code class="highlight-chroma">[14:59:13 root@centos8 ~]#iptables -AINPUT -m state --state ESTABLISHED,RELATED -j ACEPT
[14:59:13 root@centos8 ~]#iptables -RINPUT 2 -s 192.168.10.82 -j REJECT
</code></pre>
<h2 id="3-5-target">3.5 target</h2>
target 包括以下类型：
<pre><code class="highlight-chroma">自定义链, ACCEPT， DROP， REJECT，RETURN,LOG，SNAT，DNAT，REDIRECT，MASQURADE
LOG：非中断target本身不拒绝和允许,放在拒绝和允许规则前，并将日志记录在/var/log/messages系
统日志中
--log-level level 级别： debug，info，notice, warning, error, crit, alert,emerg
--log-prefix prefix 日志前缀，用于区别不同的日志，最多29个字符
</code></pre>
范例:访问记录日志
<pre><code class="highlight-chroma">[15:07:37 root@centos8 ~]#iptables -AINPUT -s192.168.10.0/24 -p tcp --dport 80 -m stat --state NEW -j LOG --log-prefix "new connections:"
[15:07:50 root@centos8 ~]#tail -1 /var/log/messages
Jan 24 15:07:20 centos8 kernel: new connections:IN=eth0 OUT= MAC=00:0c:29:62:36:f5:00:0c:29:1a:4b:7f:08:00 RC=192.168.10.82 DST=192.168.10.81 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=61055 DF PROTO=TCP SPT=32886 DPT=80 WINDOW=29200 RES=0x00 SYN URGP=0
</code></pre>
<h2 id="3-6-规则优化最佳实践">3.6 规则优化最佳实践</h2>

安全放行所有入站和出站的状态为 ESTABLISHED 状态连接,建议放在第一条，效率更高
谨慎放行入站的新请求

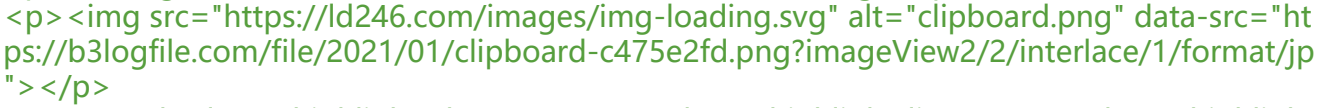
有特殊目的限制访问功能，要在放行规则之前加以拒绝
同类规则（访问同一应用，比如：http ），匹配范围小的放在前面，用于特殊处理
不同类的规则（访问不同应用，一个是 http，另一个是 mysql ），匹配范围大的放在前面，效更高

<pre><code class="highlight-chroma">-s 10.0.0.6 -p tcp --dport 3306 -j REJECT
-s 172.16.0.0/16 -p tcp --dport 80 -j REJECT
</code></pre>
<ol start="6">
应该将那些可由一条规则能够描述的多个规则合并为一条,减少规则数量,提高检查效率
设置默认策略，建议白名单（只放行特定连接）

iptables -P，不建议，容易出现“自杀现象”
规则的最后定义规则做为默认策略，推荐使用，放在最后一条

<h2 id="3-7-iptables规则保存">3.7 iptables 规则保存</h2>
使用 iptables 命令定义的规则，手动删除之前，其生效期限为 kernel 存活期限
持久保存规则：
CentOS 7,8
<pre><code class="highlight-chroma">iptables-save > /PATH/TO/SOME_RULES_FILE
</code></pre>
CentOS 6
<pre><code class="highlight-chroma">#将规则覆盖保存至/etc/sysconfig/iptables文件中
service iptables sa e
</code></pre>
加载规则
CentOS 7,8 重新载入预存规则文件中规则：
<pre><code class="highlight-chroma">iptables-restore < /PATH/FROM/SOME_RULES_FILE
</code></pre>
iptables-restore 选项
<pre><code class="highlight-chroma">-n, --noflush：不清除原有规则
-t, --test：仅分析成规则集，但不提交
</code></pre>
CentOS 6：
<pre><code class="highlight-chroma">#会自动从/etc/sysconfig/iptables 重新载入规则
service iptables r start
</code></pre>
开机自动重载规则

用脚本保存各 iptables 命令；让此脚本开机后自动运行 /etc/rc.d/rc.local 文件中添加脚本路径 /ATH/TO/SOME_SCRIPT_FILE
用规则文件保存各规则，开机时自动载入此规则文件中的规则在/etc/rc.d/rc.local 文件添加

<pre><code class="highlight-chroma">iptables-restore < /PATH/FROM/IPTABLES_RULES_FILE
</code></pre>
<h2 id="3-8-网络防火墙">3.8 网络防火墙</h2>
iptables/netfilter 利用 filter 表的 FORWARD 链,可以充当网络防火墙：
注意的问题：
(1) 请求-响应报文均会经由 FORWARD 链，要注意规则的方向性
(2) 如果要启用 conntrack 机制，建议将双方向的状态为 ESTABLISHED 的报文直接放行
<h3 id="3-8-1-FORWARD-链实现内外网络的流量控制">3.8.1 FORWARD 链实现内外网络的流量制</h3>
范例: 实现内网访问可以访问外网,反之禁止

<pre><code class="highlight-chroma">#环境准备
[15:59:55 root@centos8 ~]#hostname -I
172.16.10.71
[16:01:41 root@centos8 ~]#route -n
Kernel IP routing table
Destination   Gateway   Genmask   Flags Metric Ref   Use Iface
0.0.0.0   172.16.10.81   0.0.0.0   UG   100   0   0 eth0
172.16.10.0   0.0.0.0   255.255.255.0   U   100   0   0 eth0

[15:57:40 root@centos8 ~]#hostname -I
192.168.10.81 172.16.10.81
[16:02:29 root@centos8 ~]#echo 1 >/proc/sys/net/ipv4/ip_forward

[15:59:23 root@centos7 ~]#hostname -I
192.168.10.71
[16:02:59 root@centos7 ~]#route -n
Kernel IP routing table
Destination   Gateway   Genmask   Flags Metric Ref   Use Iface
0.0.0.0   192.168.10.81   0.0.0.0   UG   100   0   0 eth0
192.168.10.0   0.0.0.0   255.255.255.0   U   100   0   0 eth0

[15:59:07 root@centos7 ~]#hostname -I</code></pre>

```
</span></span><span class="highlight-line"><span class="highlight-cl">192.168.10.72
</span></span><span class="highlight-line"><span class="highlight-cl">[16:03:07 root@centos7 ~]#route -n
</span></span><span class="highlight-line"><span class="highlight-cl">Kernel IP routing table
</span></span><span class="highlight-line"><span class="highlight-cl">Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
</span></span><span class="highlight-line"><span class="highlight-cl">0.0.0.0         192.168.10.81   0.0.0.0         UG    100    0        0 eth0
</span></span><span class="highlight-line"><span class="highlight-cl">192.168.10.0    0.0.0.0         255.255.255.0   U     100    0        0 eth0
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">#方法1 通过标准块实现内网访问外网特定服务http和icmp,反之禁止
</span></span><span class="highlight-line"><span class="highlight-cl">[16:15:31 root@firewall ~]#iptables -AFORWARD -j REJECT
</span></span><span class="highlight-line"><span class="highlight-cl">[16:15:58 root@firewall ~]#iptables -IFORWARD -s 192.168.10.0/24 -p tcp --sport 80 -j ACCEPT
</span></span><span class="highlight-line"><span class="highlight-cl">[16:16:46 root@firewall ~]#iptables -IFORWARD -d 192.168.10.0/24 -p tcp --dport 80 -j ACCEPT
</span></span><span class="highlight-line"><span class="highlight-cl">[16:17:04 root@firewall ~]#iptables -IFORWARD -s 192.168.10.0/24 -p icmp --icmp-type 8 -j ACCEPT
</span></span><span class="highlight-line"><span class="highlight-cl">[16:20:46 root@firewall ~]#iptables -IFORWARD -d 192.168.10.0/24 -p icmp --icmp-type 0 -j ACCEPT
</span></span><span class="highlight-line"><span class="highlight-cl">[16:21:52 root@firewall ~]#iptables -vnL --line-numbers
</span></span><span class="highlight-line"><span class="highlight-cl">Chain FORWARD policy ACCEPT 0 packets, 0 bytes)
</span></span><span class="highlight-line"><span class="highlight-cl">num   pkts bytes target     prot opt in     out     source               destination
</span></span><span class="highlight-line"><span class="highlight-cl">1        2   168 ACCEPT     icmp --  *      *       0.0.0.0/0            192.168.10.0/24      icmptype 0
</span></span><span class="highlight-line"><span class="highlight-cl">2        2   168 ACCEPT     icmp --  *      *       192.168.10.0/24      0.0.0.0/0            icmptype 8
</span></span><span class="highlight-line"><span class="highlight-cl">3       18  1191 ACCEPT     tcp  --  *      *       0.0.0.0/0            192.168.10.0/24      tcp dpt:80
</span></span><span class="highlight-line"><span class="highlight-cl">4       12  1431 ACCEPT     tcp  --  *      *       192.168.10.0/24      0.0.0.0/0            tcp spt:80
</span></span><span class="highlight-line"><span class="highlight-cl">5        2   168 REJECT     all  --  *      *       0.0.0.0/0            0.0.0.0/0            reject-with icmp-port-unreachable
</span></span><span class="highlight-line"><span class="highlight-cl">#方法2 利用state块实现内网访问可以访问外网,反之禁止,但是外网可以访问内网特定服务http服务
</span></span><span class="highlight-line"><span class="highlight-cl">[16:58:48 root@firewall ~]#iptables -AFORWARD ! -s 192.168.10.0/24 -d 192.168.10.0/24 -j REJECT
</span></span><span class="highlight-line"><span class="highlight-cl">[16:59:29 root@firewall ~]#iptables -IFORWARD ! -s 192.168.10.0/24  -m state --state ESTABLISHED,RELATED -jCCEPT
</span></span><span class="highlight-line"><span class="highlight-cl">[17:03:22 root@firewall ~]#iptables -IFORWARD 2 ! -s 192.168.10.0/24 -p tcp --dport 80 -j ACCEPT
</span></span></code></pre>
<h3 id="3-8-2-NAT表-重点-">3.8.2 NAT 表（重点）</h3>
<p><img src="https://ld246.com/images/img-loading.svg" alt="clipboard.png" data-src="https://b3logfile.com/file/2021/01/clipboard-fd54c3ec.png?imageView2/2/interlace/1/format/jp
```

NAT: network address translation，支持 PREROUTING，INPUT，OUTPUT，POSTROUTIN 四个链
请求报文：修改源/目标 IP，由定义如何修改
响应报文：修改源/目标 IP，根据跟踪机制自动实现
NAT 的实现分为下面类型：

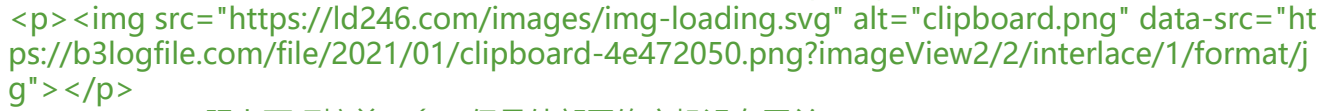
SNAT：source NAT ，支持 POSTROUTING, INPUT，让本地网络中的主机通过某一特定地址问外部网络，实现地址伪装,请求报文：修改源 IP
DNAT：destination NAT 支持 PREROUTING , OUTPUT，把本地网络中的主机上的某服务开给外部网络访问(发布服务和端口映射)，但隐藏真实 IP,请求报文：修改目标 IP
PNAT: port nat，端口和 IP 都进行修改

<h3 id="3-8-3-SNAT">3.8.3 SNAT</h3>
SNAT：基于 nat 表的 target，适用于固定的公网 IP
SNAT 选项：

--to-source [ipaddr[-ipaddr]][:port[-port]]
--random

<pre><code class="highlight-chroma">iptables -t nat -A POSTROUTING -s LocalNET ! -d LocalNet -j SNAT --to-source ExtIP
</code></pre>
注意: 需要开启 ip_forward
范例：
<pre><code class="highlight-chroma">iptables -t nat -A POSTROUTING -s 10.0.1.0/24 ! –d 10.0.1.0/24 -j SNAT --to-source 172.18 1.6-172.18.1.9
</code></pre>
MASQUERADE：基于 nat 表的 target，适用于动态的公网 IP，如：拨号网络，就是卡的公网 IP 不固定是动态获取的
MASQUERADE 选项：

--to-ports port[-port]
--random

<pre><code class="highlight-chroma">iptables -t nat -A POSTROUTING -s LocalNET ! -d LocalNet -j MASQUERADE
</code></pre>
范例：
<pre><code class="highlight-chroma">iptables -t nat -A POSTROUTING -s 10.0.1.0/24 ! –d 10.0.1.0/24 -j MASQUERADE
</code></pre>
范例：查看本地主机访问公网时使用的 IP
<pre><code class="highlight-chroma">[17:05:09 root@frewall ~]#curl ip.sb
110.17.5.83
</code></pre>
<h4 id="3-8-2-1-范例-SNAT-重点-">3.8.2.1 范例：SNAT（重点）</h4>

跟上面环境差不多，但是外部网络主机没有网关
<pre><code class="highlight-chroma">一定要启用路由转发

#针对专线静态公共P
[17:14:20 root@frwall ~]#iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -j SNAT --to-source 172.16.10.81
#针对拨号网络和线静态公共IP
[17:26:12 root@frwall ~]#iptables -t nat -APOSTROUTING -s 192.168.10.0/24 -j MASQUERADE
#查看监听端口
[17:26:52 root@frwall ~]#ss -ntl
State   Recv-Q Send-Q   Local Address:Port   Peer Address:Port
LISTEN   0   128   0.0.0.0:111   0.0.0.0:*
LISTEN   0   128   0.0.0.0:22   0.0.0.0:*
LISTEN   0   128   [::]:111   [::]:*
LISTEN   0   128   [::]:22   [::]:*
#内网可以访问外网
[17:31:36 root@centos7 ~]#curl 172.16.10.71
internet
#外网不可以访问网
[17:31:42 root@centos8 ~]#curl 192.168.10.71
curl: (7) Couldn't connect to server
#在外网服务器查到是firewalld的地址在访问
[17:32:34 root@centos8 ~]#tail /var/log/httpd/access_log
172.16.10.81 - - [2/Jan/2021:17:31:46 +0800] "GET / HTTP/1.1" 200 9 "-" "curl/7.29.0"
#查看转换状态信息
[17:34:06 root@frwall ~]#cat /proc/net/nf_conntrack
ipv4   2 tcp   6 117 TIME_WAIT src=192.168.10.71 dst=172.16.10.71 sport=52598 dport=80 src=172.16.10.71 st=172.16.10.81 sport=80 dport=52598 [ASSURED] mark=0 zone=0 use=2
</code></pre>

3.8.4 DNAT

DNAT：nat 表的 target，适用于端口映射，即可重定向到本机，也可以支持重定向至不同主机不同端口，但不支持多目标，即不支持负载均衡功能

DNAT 选项：

- --to-destination [ipaddr[-ipaddr]][:port[-port]]

<pre><code class="highlight-chroma">帮助查看
[17:34:12 root@fr

wall ~]#man iptables-extensions
</code></pre>
DNAT 格式：
<pre><code class="highlight-chroma">iptables -t nat -A PREROUTING -d ExtIP -p tcp|udp --dport PORT -j DNAT --to-destination InterSeverIP[:PORT]
</code></pre>
注意需要开启 ip_forward
范例：DNAT

/p>
<pre><code class="highlight-chroma">[17:41:04 root@frewall ~]#iptables -t nat -A PREROUTING -d 172.16.10.81 -p tcp --dport 0 -j DNAT --to-destination 192.168.10.71:80
[17:41:12 root@frewall ~]#ss -ntl
State   Recv-Q Send-Q   Local Address:Port   Peer Address:Port
LISTEN   0   128   0.0.0.0:111   0.0.0.0:*
LISTEN   0   128   0.0.0.0:22   0.0.0.0:*
LISTEN   0   128   [::]:111   [::]:*
LISTEN   0   128   [::]:22   [::]:*
[17:33:03 root@cntos8 ~]#curl 172.16.10.81
192.168.10.71 httd
[17:34:10 root@cntos7 ~]#tail /var/log/httpd/access_log
172.16.10.71 - - [2/Jan/2021:16:10:29 +0800] "GET / HTTP/1.1" 200 20 "-" "curl/7.61.1"、
[17:43:49 root@frewall ~]#cat /proc/net/nf_conntrack
ipv4   2 udp   17 27 src=192.168.10.71 dst=192.168.10.81 sport=45053 dport=123 src=192.168.10.81 dst=192.168.10.71 sport=123 dport=45053 mark=0 zone=0 use=2
ipv4   2 tcp   6 117 TIME_WAIT src=172.16.10.71 dst=172.16.10.81 sport=57170 dport=80 src=192.168.10.71 st=172.16.10.71 sport=80 dport=57170 [ASSURED] mark=0 zone=0 use=2
ipv4   2 tcp   6 99 ESTABLISHED src=192.168.10.81 dst=192.168.10.1 sport=22 dport=59552 src=192.168.10. dst=192.168.10.81 sport=59552 dport=22 [ASSURED] mark=0 zone=0 use=2
</code></pre>
范例：
<pre><code class="highlight-chroma">如果目标访问的端口和自己主机的端口一致最后可以不写
[17:48:02 root@frewall ~]#iptables -t nat -A PREROUTING -s 0/0 -d 172.16.10.81 -p tcp --dport 80 -j DNAT --to-estination 192.168.10.71
也可以把原本的端

代理到其他端口
[17:49:22 root@frwall ~]#iptables -t nat -R PREROUTING 2 -s 0/0 -d 172.16.10.81 -p tcp --dport 80 -j DNAT --t-destination 192.168.10.72:8080
注意代理端口的话用ss命令是看不到端口监听的只要应用或者进程才可以打开监听端口，iptables是内核中的操作
</code></pre>
<h3 id="3-8-5-REDIRECT转发">3.8.5 REDIRECT 转发</h3>
REDIRECT，是 NAT 表的 target，通过改变目标 IP 和端口，将接受的包转发至同一个主机的不端口，可用于 PREROUTING OUTPUT 链
REDIRECT 选项：

--to-ports port[-port]

注意: 无需开启 ip_forward，一般用于本机端口转发
范例：
<pre><code class="highlight-chroma">iptables -t nat -A PREROUTING -d 172.16.100.10 -p tcp --dport 80 -j REDIRECT --to-ports 8080
</code></pre>
范例：
<pre><code class="highlight-chroma">[17:42:27 root@centos7 ~]#ss -ntl
State   Recv-Q Send-Q   Local Address:Port   Peer Address:Port
LISTEN   0   128   *:111   *:*
LISTEN   0   128   *:22   *:*
LISTEN   0   100   127.0.0.1:25   *:*
LISTEN   0   128   [::]:111   [::]:*
LISTEN   0   128   [::]:80   [::]:*
LISTEN   0   128   [::]:22   [::]:*
LISTEN   0   100   [::1]:25   [::]:*
[18:06:02 root@centos7 ~]#iptables -t nat -APREROUTING -p tcp --dport 8000 -j REDIRECT --to-ports 80
[17:45:54 root@centos7 ~]#curl 192.168.10.71:8000
192.168.10.71 httpd
</code></pre>