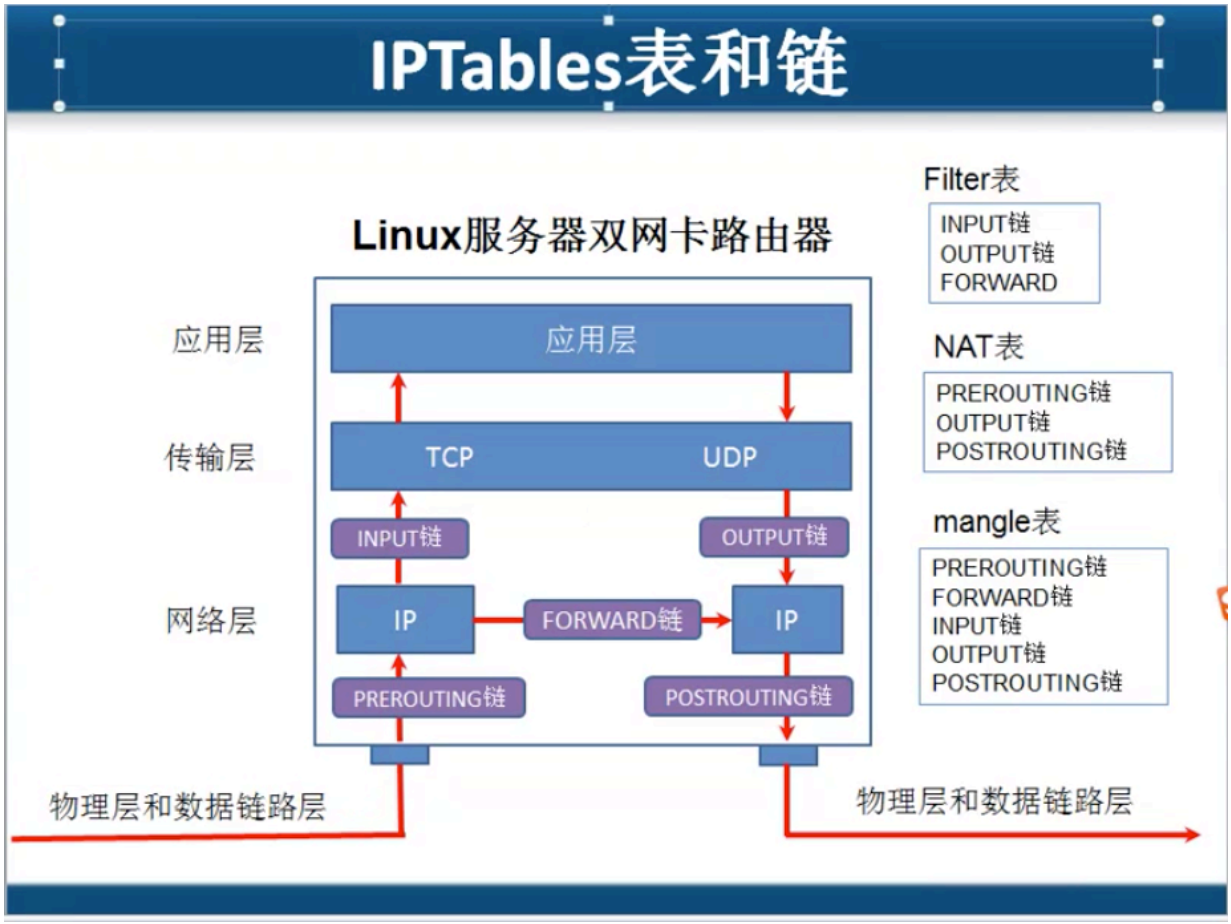


Iptables

保存 规则：
/etc/rc.d/init.d/iptables save
##或者
iptables-save > /etc/sysconfig/iptables

Iptables -t table 命令 chain rules -j target

table 可以是 filter nat mangle 默认 filter

命令
-P 或 --policy 定义 默认 策略

iptables -t filter —policy FORWARD DROP
iptables -t filter -P FORWARD ACCEPT

-A 或 —append 在规则列表 的 最后 一条 增加一条 规则

-I 或 — instert 在指定 的位置 插入一条规则 ,如果不指定规则 那就 是 最上边

iptables -t filter -I INPUT -p icmp -j DROP
iptables -t filter -I INPUT 3 -p icmp -j DROP #插入在第三个位置

-D 或 - -delete 删除一个规则

iptables -t filter -D INPUT 1 # 删除第一条规则
iptables -t filter -D INPUT 3 # 删除 第三条 规则

-R 或 —replace 替换规则列表中的 某个规则
Iptables -t filter -R INPUT 2 -p icmp -j DROP

-F 或 - - flush 删除 表中所有规则

Iptables -t filter -F INPUT #删除链上的所有 规则

iptables 匹配选项

-i or —in-interface 指定数据包从哪个网络接口进入 如 ppp0 eth0 eth1 等
-o or —out-interface 指定数据包从那块网路接口输出
Iptables -t filter -I INPUT -p icmp -i eth3 -j DROP # 拒绝 从网卡 eth3 进入

Iptables -t filter -I FORWARD -p tcp -s 192.168.80.123/32 -d 192.168.10.123/32 —dport 3389 -j DROP
Iptables -t filter -I FORWARD -p tcp -s 192.168.80.0/24 -d 192.168.10.0/24 -j DROP

-p 或 - - protocol 协议类型 指定数据包 匹配的协议 如 tcp udp icmp 等

-s 或 - - source 指定数据包匹配的原地址
-d or - - destination 指定数据包匹配的目标地址

- - sport 指定数据包匹配 的源端口号 , 可以使用 1:1024 的格式 指定端口 范围\
- - dport 目标端口号 指定数据包匹配的目标端口号, 可以使用 …….

基于状态的 扩展 匹配选项
参数 -m state

基于 状态检测 的包过滤 , 指定检测哪种 状态
—state {NEW ,ESTATBLISHED, INVALID, RELATED}

说明 用来比对 连接状态 , 连接状态 共有 四种 : INVALID. ESTABLISHED, NEW 和 RELATED

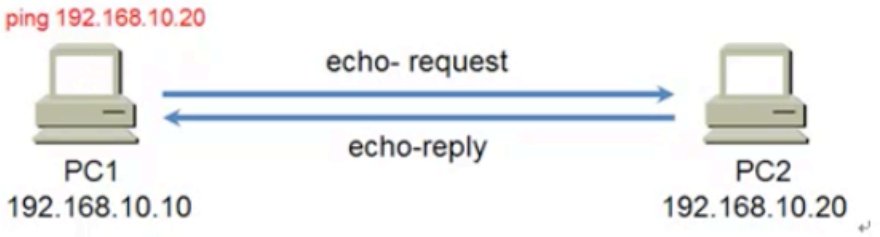
INVALID 表示 该封包 的连接 编号 (session ID) 无法辨识 或 编号 不正确
ESTABLISHED 表示 该封包 属于某个 已经建立 的连接
NEW 表示 该封包 想要 起始 一个连接 (重设 连接 或将 连接 重导向)
REALTED 表示 该封包 是 属于 某个已经建立 的连接 ,所建立的新连接 .
例如 FTP-data 连接 必定是 源自 某个 FTP 连接

iptables -t filter -I FORWARD -s 192.168.10.0/24 -d 192.168.80.0/24 -m state - -state NEW -j DROP # 在路由 端 设置

iptables -t filter -I OUTPUT -m state —state NEW -j DROP # 在服务器 本机

参数-m icmp --icmp-type

Ping 命令使用 icmp 协议测试网络是否畅通，Icmp 有两种常用类型的数据包即 icmp-type，常用的类型为 echo-reply 和 echo- request。如下图所示 PC1 ping PC2，发出去的数据包是 icmp 协议 echo-reply 类型的数据包，PC2 返回来的数据包是 icmp 协议的 echo-reply 类型的数据包。



示例

禁止 192.168.10.0/24ping 通 192.168.80.0/24

允许 192.168.80.0/24ping 通 192.168.10.0/24

```
[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -d 192.168.80.0/24 -p icmp -m icmp --icmp-type echo-request -j DROP
```

禁止其他计算机 ping 本地 Linux

```
[root@LinuxServer ~]# iptables -t filter -I INPUT -p icmp -m icmp --icmp-type echo-request -j DROP
```

Icmp 数据包 类型 可以使用数组 表达

echo-request 8
echo-reply 0

参数 -m multiport

指定多端口号

-m multiport

--sport

--dport

--ports

示例：

拒绝 192.168.80.0/24 访问 192.168.10.0/24 1-1024 3339

[root@WebServer ~]# iptables -t filter -I FORWARD -p tcp -d 192.168.10.0/24 -s 192.168.80.0/24 -m multiport --dports 1:1024,3389 -j DROP

指定 IP 段

-m iprange

--src-range ip-ip

--dst-range ip-ip

示例：

禁止 192.168.80.1-100 地址段 访问 192.168.10.0/24

[root@WebServer ~]# iptables -t filter -I FORWARD -m iprange --src-range 192.168.80.1-192.168.80.100 -j DROP

限速

-m limit --limit

说明 用来比对某段时间内封包的平均流量，上面的例子是用来比对：每小时平均流量是否超过一次 3 个封包。

除了每小时平均一次外，也可以每秒钟、每分钟、每小时或每天平均一次，默认值为每小时平均一次，参数如后： /second /minute /hour /day

范例

192.168.10.0/24 网段每秒钟向 192.168.80.0/24 发送数据包不能超过 300 个， 1500 字节 1500*300=450 000=450K

```
[root@WebServer ~]# iptables -F
[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -d 192.168.80.0/24 -m limit --limit 3000/second -j ACCEPT
[root@WebServer ~]# iptables -t filter -A FORWARD -s 192.168.10.0/24 -d 192.168.80.0/24 -j DROP
```

公司使用 linux 服务器来当网关，配置 iptables NAT 上网，虽然公司光纤带宽 30MB，但只要有一个人使用迅雷等 P2P 软件下载资料时，会把带宽跑满，如何使用 iptables 来对每个用户进行带宽限速呢，参见如下脚本即可：

vi /etc/sysconfig/limit.sh

```
for ((i=2; i<254; i++))
do
iptables -I FORWARD -s 192.168.10.$i -j DROP
iptables -I FORWARD -s 192.168.10.$i -m limit --limit 300/sec --limit-burst 400 -j ACCEPT
done
~
```

双击可显示空白

注：上面的脚本是对 192.168.1.2-254 限制每秒钟只允许最多 300 个数据包通过的限制；

使用 扩展 选项 限制 最大 连接 数

```
[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.80.0/24 -d 192.168.10.123/32 -p tcp --dport 3389 -m connlimit --connlimit-above 2 -j DROP
```

瞬间流量 控制

瞬间流量控制

参数 --limit-burst

说明 用来比对瞬间大量封包的数量，上面的例子是用来比对一次同时涌入的封包是否超过 5 个（这是默认值），超过此上限的封包将被直接丢弃。使用效果同上。

范例

允许 192.168.10.123/32 ping 192.168.80.123/32 4 个包

在路由器上的设置

```
[root@WebServer ~]# iptables -F
[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.123/32 -d 192.168.80.123/32 -p icmp -m limit --limit-burst 4 -j ACCEPT
[root@WebServer ~]# iptables -t filter -A FORWARD -s 192.168.10.123/32 -d 192.168.80.123/32 -p icmp -j DROP
root@WebServer ~]# iptables -t filter -Z
```

基于 Mac 的地址 过滤 流量

参数 -m mac --mac-source

说明 用来比对封包来源网络接口的硬件地址，这个参数不能用在 OUTPUT 和 Postrouting 规则链上，这是因为封包要送出到网卡后，才能由网卡驱动程序透过 ARP 通讯协议查出目的地的 MAC 地址，所以 iptables 在进行封包比对时，并不知道封包会送到哪个网络接口去。

示例：

拒绝 2003 计算机的 MAC 能够访问 VMNet8 这个网段

```
[root@WebServer ~]# iptables -t filter -I FORWARD -d 192.168.80.0/24 -m mac --mac-source 00-0C-29-DB-32-6F -j DROP
```

处理 动作 reject 详解

.处理动作

-j 参数用来指定要进行的处理动作，常用的处理动作包括：ACCEPT、REJECT、DROP、REDIRECT、MASQUERADE、LOG、DNAT、SNAT、MIRROR、QUEUE、RETURN、MARK。

Filter 表能使用的主要动作：

ACCEPT： 将封包放行，进行完此处理动作后，将不再匹配其它规则，直接跳往下一个规则链。

REJECT：拦阻该封包，并传送封包通知对方，可以传送的封包有几个选择：ICMP port-unreachable、ICMP echo-reply 或是 tcp-reset（这个封包会要求对方关闭 连接），进行完此处理动作后，将不再匹配其它规则，直接中断过滤程序。

DROP：丢弃封包不予处理，进行完此处理动作后，将不再匹配其它规则，直接中断过滤程序。

双击可显示空白

LOG： 将封包相关讯息纪录在 /var/log 中，详细位置请查阅 /etc/syslog.conf 配置文件，进行完此处理动作后，将会继续匹配其规则。

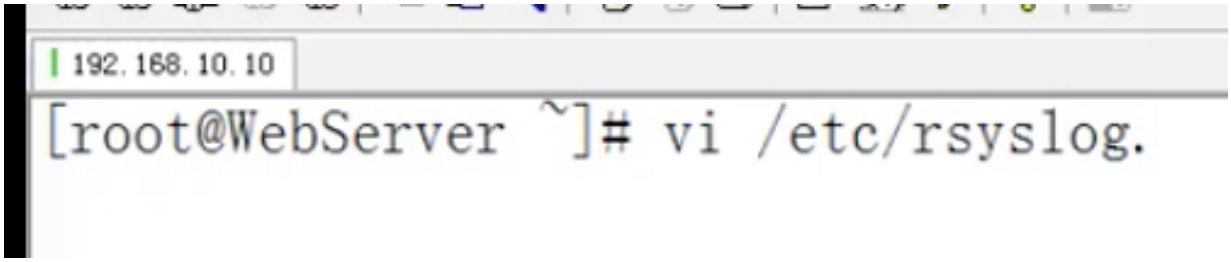
处理 动作 LOG 详解

```
[root@WebServer ~]# iptables -F
[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -p tcp --dport 3389 -j DROP
[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -p tcp --dport 3389 -j LOG --log-prefix "i
ptables-rdp"
[root@WebServer ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
LOG        tcp  --  192.168.10.0/24      anywhere            tcp dpt:ms-wbt-server LOG level warning prefix `ip
tables-rdp'
DROP       tcp  --  192.168.10.0/24      anywhere            tcp dpt:ms-wbt-server

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

修改 默认 日志 存放位置



保存 和 还原 iptables 设置

/etc/sysconfig/iptables

iptables-save
iptables-restore

常规网络安全设置

示例:
Web 服务器对 Internet 开放 TCP 80 端口

SSH 端口 只允许 192.168.10.123 访问
默认拒绝所有进入的流量

安装 httpd 服务
[root@WebServer ~]# yum install httpd
[root@WebServer ~]# chkconfig httpd on
[root@WebServer ~]# service httpd restart

设置防火墙规则
[root@WebServer ~]# iptables -t filter -P INPUT DROP
[root@WebServer ~]# iptbales -F
[root@WebServer ~]#iptables –t filter –I INPUT –p tcp --dport 80 -j AACEPT
[root@WebServer ~]# iptables –t filter –I INPUT –p tcp --dport 22 -s 192.168.10.123/32 -j AACEPT

recent 模块实现服务器网络安全

recent 这个模块很有趣，善加利用可充分保证您服务器安全。

设定常用参数：

--name #设定列表名称，默认 DEFAULT。

--rsource #源地址，此为默认。

--rdest #目的地址

--seconds #指定时间内

--hitcount #命中次数

--set #将地址添加进列表，并更新信息，包含地址加入的时间戳。

--rcheck #检查地址是否在列表，以第一个匹配开始计算时间。

--update #和 rcheck 类似，以最后一个匹配计算时间。

--remove #在列表里删除相应地址，后跟列表名称及地址。

示例 1

SSH 连接一个客户端 60 秒内只允许连接 2 次

用来设置多长时间 几次 拒绝

[root@WebServer ~]# iptables -t filter -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --name SSHPOOL --rcheck --seconds 60 --hitcount 2 -j DROP

连接计数

[root@WebServer ~]# iptables -t filter -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --name SSHPOOL --set –j ACCEPT

建立了会话的数据包 就允许进入

[root@WebServer ~]# iptables -t filter -A INPUT -m state --state ESTABLISHED -j ACCEPT

示例 2

限制 TCP 80 端口 60 秒内每个 IP 只能发起 10 个新连接，超过记录日记及丢失数据包，可防 CC 及非伪造 IP 的 syn flood。

iptables -A INPUT -p tcp --dport 80 --syn -m recent --name webpool --rcheck --seconds 60 --hitcount 10 -j LOG --log-prefix 'DDOS:' --log-ip-options

iptables -A INPUT -p tcp --dport 80 --syn -m recent --name webpool --rcheck --seconds 60 --hitcount 10 -j DROP

iptables -A INPUT -p tcp --dport 80 --syn -m recent --name webpool --set -j ACCEPT

设置打开端口的钥匙

#记录日志，前缀 SSHOPEN

iptables -A INPUT -p icmp --icmp-type echo-request -m length --length 1078 -j LOG --log-prefix "SSHOPEN: "

#指定数据包 1078 字节，包含 IP 头部 20 字节，ICMP 头部 8 字节。

iptables -A INPUT -p icmp --icmp-type echo-request -m length --length 1078 -m recent --set --name sshopen --rsource -j ACCEPT

#检查 sshopen 表中 60 秒内的记录如果有源地址则允许访问 TCP 的 22 端口

iptables -A INPUT -p tcp --dport 22 --syn -m recent --rcheck --seconds 60 --name sshopen --rsource -j ACCEPT

#发送 ping 包指定数据包 1178 个字节，将客户端址从 sshopen 表中移除，客户端将不能使用 tcp 的 22 端口建立连接

iptables -A INPUT -p icmp --icmp-type echo-request -m length --length 1178 -m recent --set --name sshopen –rmove -j ACCEPT

iptables –A INPUT –m state –state ESTABLISHED –j ACCEPT

Linux 下解锁

ping –s 1050 192.168.10.10

Windows 下解锁

ping -l 1050 192.168.10.10

保护整个 网段 的 安全

```
[root@WebServer ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@WebServer ~]# iptables -t filter -I FORWARD -p tcp --dport 3389 -d 192.168.10.110/32 -o eth2 -j ACCEPT
[root@WebServer ~]# iptables -t filter -I FORWARD -m state --state ESTABLISHED -j ACCEPT
[root@WebServer ~]#
```

使用 nat 表 实现 网段 地址 转换

```
192.168.10.10
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@WebServer ~]# ifconfig eth3:1 192.168.20.11 netmask 255.255.255.0
[root@WebServer ~]# iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o eth3 -j SNAT --to-source 192.168.20.10
-192.168.20.11
[root@WebServer ~]# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  192.168.10.0/24      anywhere            to:192.168.20.10-192.168.20.11

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

端口 映射 (在配置 好 net 的情况 下)

```
[root@WebServer ~]# iptables -t nat -A PREROUTING -i eth3 -d 192.168.20.10/32 -p tcp --dport 3389 -j DNAT --to
192.168.10.110:3389
[root@WebServer ~]#
```

·mangle 表的应用案例

–ttl-inc 1

--ttl-dec 2

--ttl-set 45

设置 TTL 的值

```
[root@ftpLinux ~]# iptables -t mangle -A PREROUTING -i eth0 -j TTL --ttl-set 30
```

删除

```
[root@ftpLinux ~]# iptables -t mangle -D PREROUTING 1
[root@ftpLinux ~]# iptables -t mangle -A PREROUTING -i eth0 -j TTL --ttl-inc 4
[root@ftpLinux ~]# iptables -t mangle -D PREROUTING 1
[root@ftpLinux ~]# iptables -t mangle -A PREROUTING -i eth0 -j TTL --ttl-dec 10
```