Iptables 防火墙讲义

- What’ s Iptables ?
- iptables 是一种基于包过滤的防火墙
- Iptables 需要 2.4 以上版本的内核支持
- 2.6 内核仍然支持 iptables
- iptables 和内核的关系

撰稿人：马路遥

RHCE™

iptables 命令

```
iptables –t filter –A INPUT  \
–p tcp --dport 23  –j REJECT

iptables –L
```

iptables 的表

- 包过滤中包含 3 个表
- filter table ，过滤表
- Nat 表，用于地址转换
- mangle 表，俗称矫正表，本课程不包括 mangle 的内容，仅仅简单介绍其含义

filter 表 (1)

包含
INPUT 、 OUTPUT 和 FORWARD 链，用于处理输入、输出和转发包。

filter 表是缺省的表。

filter 表 (2)

在我们使用

iptables statement 的时候多数等同于下面的命令：

iptables -t filter statement

nat 表

用于处理网络地址翻译。（包含与 masquerading 相关的功能)

1 包含 PREROUTING (路由前)

2.POSTROUTING 路由后

3.OUTPUT 输出 (很少用到)

共 3 个链

mangle 表

用于处理特殊包的矫正，包含两个链
PREROUTING (路由前)
POSTROUTING （路由后）

关于 Mangle 的讨论就此为止，以下不再研究

在流量控制和 Qos 应用中 , 经常会用到 mangle 表。

规则和链 (1)

- 每个链中的规则是按顺序的，处理一个包时，从第一条规则到最后一条规则，依次匹配。
- 顺序很重要
- 可以创建自定义规则。 (大体相当于子函数)

规则和链 (2)

可以动态添加、删除和修改规则
可以查看当前规则

简单的 iptalbes 命令

iptables -F 清除所有规则
iptables -X 清除所有自定义规则
iptables -L 列出当前所有规则

在学习过程中，使用自定义脚本，最好以
iptables -F 和 iptables -X 开头
确保不受到其他因素的干扰。

iptables 命令

iptables –t filter –A INPUT \
–p tcp --dport 23 –j REJECT

红色部分定义使用的表
紫色部分定义匹配的规则
绿色部分定义了采取的措施

filter 表中采取的措施（ 1 ）

- ACCEPT 接受 , 等于不进行过滤
- DROP 丢弃 , 弃之不理 . 别人可以判断出您的系统使用了防火墙 . 还记得 Sendmail 中的 DISCARD 么？
- REJECT 弹回，通常貌似跟本没有打开这端口。
- LOG 进行日志， /var/log/message
- User Chain ，用自定义规则进行处理，等同于用子函数进行处理

措施中的措施

-j LOG
--log-prefix YOUR_STRING
-j REJECT --reject-with
tcp-reset icmp-port-unrearchable
icmp-host-unreachable

简单地添加规则 (1)

根据源地址进行匹配 的 -s 参数
[!] addr[net mask]
根据目的地址进行匹配 -d
[!] addr[net mask]

使用 “ !” 的时候，需要在两端加空格 (下同)

简单地添加规则 (2)

根据协议进行匹配的 -p 参数

- [!] icmp
- [!] tcp
- [!] udp

简单地添加规则 (3)

根据端口进行匹配，这时必须指定协议，必须是 tcp 或 udp 协议。

根据封包来源的端口进行匹配 的

--sport [!] port 。

--source-port = --sport

port 可以用 /etc/services 中的协议名来代替

简单地添加规则 (3)

根据封包的目的端口进行匹配

--dport [!] port

--dport = --destination-port

port 可以用 /etc/services 中的协议名来代替

实验及准备工作

1. 彻底关闭 tcp_wrappers 防火墙
2. 彻底地、永久地停止 ipchains 服务
   rpm -e ipchain
3. 打开 telnet ipop3 和 sendmail 服务，实验中将使用上述服务的端口进行练习 .
4. 实验中全部使用 -A INPUT 的方式
5. 不要使用内核中禁止 ping 的参数 .

准备脚本

#!/bin/bash
iptables -F
iptables -X
第一行
....
最后一行

思考题

当甲连接乙的 telnet server 时，甲和乙各使用那个端口？

练习

1. 使得只有 192.168.0.x　和 192.168.0.y 的机器可以连接您的 pop3 服务器。但同一网段的其他人不行。
2. 每执行一个 iptables 命令后。使用 lsmod |grep ip 命令，查看内核中 iptables modules 的载入情况。
3. 禁止某一人 ping 到你 , 但别人可以

练习

1. 使用 iptables –L 命令来查看您的当前规则
2. 使用 iptables --help 命令来查看帮助
3. 可否用 ! icmp 来进行 port 方面的定义
4. 使得你可以 telnet 某人，但他不能 telnet 您。
5. 用 telnet 和 nmap 的方式，体会 REJECT 和 DROP 的差别。

练习

1. 本网段中的所有人都可以访问你的 pop3 服务，但禁止某三人。
2. 本网段中之有某三人可以访问你的 25 端口。其他人不行。
3. 以上两个练习分别做，争取能用最少的规则完成之。

练习

1. 使用 -j LOG 参数或
   -j LOG --log-prefix YOURSTRING 参数，
   查看 /var/log/messages 文件中的内容
2. 观察 LOG 和 ACCEPT 、 DROP 以及
   REJECT 参数使用不同顺序的结果。

INPUT 和 OUTPUT 的差别 (1)

对于 INPUT 而言

--dport -d 都是指你自己的端口和地址

--sport 和 -s 指的是发起连接者的端口和地址

INPUT 和 OUTPUT 的差别 (2)

对于 OUTPUT 而言
--sport -s 都是指你自己的端口和地址
--dport 和 -d 指的 dest 地址

INPUT 和 OUTPUT 的差别 (3)

任务：使得您不能 telnet 到 192.168.0.50 ，分别用 INPUT 和 OUTPUT 方法实现。

INPUT 和 OUTPUT 的差别 (4)

方法一

iptables -t filter -A INPUT –s \
192.168.0.50 -p tcp --sport 23 -j REJECT

方法二

iptables -t filter -A OUTPUT -d \
192.168.0.50 –dport 23 -j REJECT

INPUT 和 OUTPUT 的差别 (5)

方法一中，你发起的 telnet 请求被服务器所接收，但服务器返回给你的封包被 iptables 所阻挡。

方法二中，你发出的 telnet 请求本身就被 iptables 所阻挡。服务器当然收不到你的请求。

理解下面的脚本

- #!/bin/bash

```
iptables -F;iptables -X; IP=”  192.168.1”
iptables -A INPUT -p tcp --dport 110 -s $IP.101  -j ACCEPT
iptables -A INPUT -p tcp --dport 110 -s $IP.102  -j ACCEPT
iptables -A INPUT -p tcp --dport 110 -s $IP.0/24  -j REJECT
iptables -A INPUT -p icmp -s $IP.101 -j REJECT

iptables -A INPUT -s  $IP.101 -p tcp --dport 23 -j REJECT
iptables -A INPUT -s $IP.0/24 -p tcp --dport 23 -j ACCEPT
```

练习：用 OUTPUT 链改写上述脚本 , 并实现相同的功能

常用参数

iptables -A
增加一条规则，
iptables -t filter -A INPUT -s \
192.168.0.1 -j DROP
iptables -t filter -A INPUT -s \
192.168.0.2 -j DROP
用 iptables -L 可以看到两条规则

再增加两条规则

**iptables -t filter -A OUTPUT -d **
** 192.168.0.3 -j DROP**

**iptables -t filter -A OUTPUT -d **

192.168.0.4 -j DROP

现在就有了 **4** 条规则

iptables -L 命令可以看到 4 条规则
iptables -D 可以删除规则

方法 1:
iptables -t filter -D INPUT -s \
192.168.0.2 -j DROP
　　要和原来一样 , 但把 **-A** 换为 **-D**

方法二

iptables -D INPUT 2

可以删除第二条 **INPUT** 规则

iptables -D OUTPUT 1

可删除第一条输出规则

iptables -L 命令

iptables -L , 列出所有 filter 表的规则

等同于 iptables -t filter -L

iptables -t nat -L 列出所有 nat 表的规则

iptables -L INPUT 列出所有 filter 表中的 INPUT 规则

iptables -L OUTPUT 什么意思？

插入规则 iptables -I

```
iptables -t filter -I INPUT \
-s 192.168.0.5 -j REJECT
```

插入这条规则并做为第一条 INPUT 规则.
其余 INPUT 规则下移一行

```
iptables -t filter -I INPUT 3 \
-s 192.168.0.6 -j REJECT
```

将此条插入作为第三行，原来第三行及以后下移一行

思考 , 执行下列脚本后的规则顺序

```
for i in nat filter mangle ;do
iptables -t  $i  -F;   iptables -t  $i -X;done
for i in 1 2 3 4 ;do
iptables -A INPUT -s 192.168.0.$i -j DROP
done
iptables  -I INPUT    -s 192.168.0.5 -j DROP
iptables  -I INPUT 3  -s 192.168.0.6 -j DROP
```

执行 iptables -L INPUT 发现第一行是 192.168.0.6, 然后规则依次是 192.168.0.

5 1 6 2 3 4

iptables -R 规则取代一条规则

iptables -R INPUT **3** -d 192.168.0.7 -j DROP

取代第三条 INPUT 规则 , 必需指定取代第几条规则。

iptables -F 命令

清除规则

iptables -t filter -F INPUT

iptables -t filter -F OUTPUT

iptables -t nat -F

Iptables -N , 插入自定义规则

iptables -N dalian

iptables -A dalian -d 192.168.0.9 -j DROP

iptables -A dalian -d 192.168.0.8 -j DROP

使用自定义规则

iptables -A INPUT -j dalian

查询自定义规则

iptables -L dalian

删除自定义规则

iptables -X dalian (空链)

iptables -F dalian

重命名自定义规则

iptables -E dalian shenyang

缺省策略

Iptables -P 缺省策略

iptables -P INPUT DROP
iptables -P Someting ACTION
Something 必需是 **INPUT** 或 **OUTPUT** 之一

一般放在脚本的最后一行

对比性匹配的扩展

通过 -m 参数来调用 . 主要用法有
基于状态的匹配 -m state
基于 Mac 地址的匹配 -m mac
基于封包数量的匹配 -m limit
基于 uid 、 gid 的限制 -m owner

基于状态的防火墙

第一种状态叫作 NEW 。输入 "ssh abc.com" 时，初始包或源自于您的机器并要发送到 abc.com 的包都处于 NEW 状态。但是，即使只从 abc.com 接收到一个应答包，那么就立即不再将其它作为此连接的一部分、发送至 abc.com 的包看作是 NEW 包 .

只有在建立新连接、并且还没有从远程主机接收到通信流时使用的包才被看作是 NEW（当然，这个包是此特定连接的一部分）。我们已经描述了外出 NEW 包，但还有可能会有进入 NEW 包（很常见）。进入 NEW 包通常来自远程机器，在启动与您的连接时使用。

您的 Web 服务器接收到的初始包（作为 HTTP 请求的一部分）将被看作是进入 NEW 包；但是，只要您应答了一个进入 NEW 包，所接收到的与此特定连接相关的其它包都不再被看作是处于 NEW 状态。

ESTABLISHED 状态

一旦连接看到两个方向上都有通信流，与此附加相关的其它包都被看作处于 ESTABLISHED 状态。 NEW 和 ESTABLISHED 之间的区别很重要

RELATED 状态

第三种连接状态类别叫作 RELATED。RELATED 包是那些启动新连接，但有与当前现有连接相关的包。RELATED 状态可以用于调整组成多重连接协议（如 ftp）的连接，以及与现有连接相关的错误包（如与现有连接相关的 ICMP 错误包）

INVALID 状态

最后是 INVALID 包 ，那些包不能归入以上三种类别。应当注意某个包是否被看作是

INVALID，因为这种包不会被自动废弃；因此您需要插入适当的规则，并设置链策略，以便可以正确处理这些包。

允许向外主动发出的包 (的应答回来)

iptables -A OUTPUT -j ACCEPT

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

禁止别人发起的主动连接

iptables -A INPUT -m state --state NEW,INVALID -j DROP

基于 MAC 地址的匹配

格式 ：-m mac --mac-source mac_addr

```
iptables -A INPUT -p tcp --dport 23 -m
  mac \
  --mac-source 00:0C:29:BC:BB:DB \
-j REJECT
```

注 :-m mac 仅仅对 PREROUTING 和 INPUT 链起作用

限制别人 ping

允许每秒通过一个 icmp 包，默认触发条件是 5 个 icmp 包

```
iptables -A INPUT -p icmp -m limit \
--limit 1/s -j ACCEPT
```

超过部分全部拒绝

```
iptables -A INPUT -p icmp -j DROP
```

根据 uid 或者 gid 进行限制

-m owner 参数

-m owner [!] --uid-owner $AN_UID

```
iptables -A OUTPUT -p tcp --dport 23
 -m owner --uid-owner 500 -j REJECT
iptables -A OUTPUT -p tcp --dport 23
 -m owner --gid-owner 500 -j REJECT
```

注 :-m owner 仅仅对 OUTPUT 链有效

练习题

- 缺省策略为 INPUT 和 OUTPUT 拒绝全部, 但 : 设置
- 允许在本机上以 root 用户的身分对你自己的 ip 地址 (192 和 127) 进行任何访问 .
- 只允许某 2 台机器 telnet 你
- 允许某 2 台机器每秒钟 ping 你 1 次 .
- 允许 root 用户主动访问任何机器 .

练习题

- Test01 用户可以使用的自己的 ip(127 和 192) 地址作测试用途 , 但不能访问其他的任何 ip 。以 Test01 用户登陆。尝试是否可以 ping 到别人。为什么？
- Test02 用户可以访问任意地址
- 对于老师的笔记本电脑 , 不论老师如何改变 ip 地址 , 都不能使用你的 pop3 服务器 . 但可以使用你的 sendmail 服务 .

练习题

- 定义一个自定义规则，对所有访问您 23 端口的动作进行日志

What’ s NAT

NAT= Network Address Translation

在两个网站之间，如 sohu.com 和 163.com 之间传递封包，要经过若干个 router ，他们不会改变你的封包，仅仅是传递而已 .

NAT 所做的工作，是将传出去的封包重新包装。然后再将传回来的封包进行反包装

Why NAT

1. 你只有一个 ip 地址，但想带动多台主机上网。这时您需要 source-NAT
2. 一个 ip 的多个服务器 . 有时候您想让人连接到您真实 ip 后边的服务器。这时您需要的是 Destination-NAT 。不严格的说法叫映射 (map)
3. 透明代理

Nat 表

使用 nat 表之前 , 必须调整内核参数

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

应该写将此值写入 /etc/sysctl.conf 以使之在重新启动后仍然生效

小窍门：用 sysctl –p 命令使该文件立即生效

普通的地址伪装

SNAT 源地址转换

iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE

或者

**iptables -t nat -A POSTROUTING -j **

SNAT --to ONE_OF_YOUR_EXT_IP

前者在使用动态 **IP** 的时候使用。后者一般在使用静态地址的时候使用。

解决网关的 ftp 和 irc 问题

modprobe -a ip_conntrack_ftp ip_nat_ftp
modprobe -a ip_conntrack_irc ip_nat_irc
解决 QQ 的聊天问题？Sorry

透明代理

iptables -t nat -A PREROUTING -s 192.168.10.0/24 -p tcp -m multiport --dport 80,443 -j REDIRECT --to-port 3128

(squid 服务需要特别的配置)

Squid 的配置

在能正常工作的 squid.conf 中增加下列四行即可 :

- *httpd_accel_host virtual*
- *httpd_accel_port 80*
- *httpd_accel_with_proxy on*
- *httpd_accel_uses_host_header on*

穿透防火墙 . 提供服务

```
iptables -t nat -A PREROUTING -p tcp -i \
ppp0 --dport 80 -j DNAT –to \
192.168.10.156:80
```

新参数 -i -o

-i 参数指定网卡，适用于 INPUT 和 PREROUGING

-o 参数指定网卡，适用于 OUTPUT 和 POSTROUGING

几种流向的顺续为

PREROUTING --->INPUT--> OUTPUT--> POSTROUTING

Iptables with Redhat

/etc/rc.d/init.d/iptables start

启动了当前在 /etc/sysconfig/iptables 中保存了的规则 .

/etc/rc.d/init.d/iptables save

将当前规则保存在 /etc/sysconfig/iptables 中