iptables

Linux上常用的防火墙软件

补充说明

iptables命令 是Linux上常用的防火墙软件，是netfilter项目的一部分。可以直接配置，也可以通过许多前端和图形界面配置。

语法

```
iptables(选项)(参数)
```

选项

```
-t, --table table 对指定的表 table 进行操作， table 必须是 raw， nat，filter，mangle 中的一个。如果不指定此选项，默认的是 filter 表。

# 通用匹配：源地址目标地址的匹配
-p，指定要匹配的数据包协议类型；
-s, --source [!] address[/mask] ，把指定的一个／一组地址作为源地址，按此规则进行过滤。当后面没有 mask 时，address 是一个地址，比如：192.168.1.1；当 mask 指定时，可以表示一组范
```

围内的地址，比如：192.168.1.0/255.255.255.0。
-d, --destination [!] address[/mask] 地址格式同上，但这里是指定地址为目的地址，按此进行过滤。
-i, --in-interface [!] <网络接口name> 指定数据包的来自来自网络接口，比如最常见的eth0 注意：它只对 INPUT FORWARD PREROUTING 这三个链起作用。如果没有指定此选项， 说明可以来自任何一个网络接口。同前面类似，"!" 表示取反。
-o, --out-interface [!] <网络接口name> 指定数据包出去的网络接口。只对OUTPUT FORWARD POSTROUTING 三个链起作用。

查看管理命令
-L, --list [chain] 列出链 chain 上面的所有规则，如果没有指定链，列出表上所有链的所有规则。

规则管理命令
-A, --append chain rule-specification 在指定链 chain 的末尾插入指定的规则，也就是说，这条规则会被放到最后，最后才会被执行。规则是由后面的匹配来指定。
-I, --insert chain [rulenum] rule-specification 在链 chain 中的指定位置插入一条或多条规则。如果指定的规则号是1，则在链的头部插入。这也是默认的情况，如果没有指定规则号。
-D, --delete chain rule-specification -D, --delete chain rulenum 在指定的链 chain 中删除一个或多个指定规则。
-R num Replays替换/修改第几条规则

链管理命令（这都是立即生效的）
-P, --policy chain target 为指定的链 chain 设置策略 target注意，只有内置的链才允许有策略，用户自定义的是不允许的。
-F, --flush [chain] 清空指定链 chain 上面的所有规则。如果没有指定链，清空该表上所有链的所有规则。
-N, --new-chain chain 用指定的名字创建一个新的链。
-X, --delete-chain [chain] 删除指定的链，这个链必须没有被其它任何规则引用，而且这条上必须没有任何规则。如果没有指定链名，则会删除该表中所有非内置的链。
-E, --rename-chain old-chain new-chain 用指定的新名字去重命名指定的链。这并不会对链内部造成任何影响。
-Z, --zero [chain] 把指定链，或者表中的所有链上的所有计数器清零。

-j, --jump target <指定目标> ：即满足某条件时该执行什么样的动作target 可以是内置的目标，比如 ACCEPT也可以是用户自定义的链。
-h显示帮助信息；

基本参数

-P	设置默认策略:iptables -P INPUT (DROP	ACCEPT)
-F	清空规则链	
-L	查看规则链	
-A	在规则链的末尾加入新规则	
-I	num 在规则链的头部加入新规则	
-D	num 删除某一条规则	
-s	匹配来源地址IP/MASK加叹号"!"表示除这个IP外。	
-d	匹配目标地址	
-i	网卡名称 匹配从这块网卡流入的数据	
-o	网卡名称 匹配从这块网卡流出的数据	
-p	匹配协议,如tcp,udp,icmp	
--dport num	匹配目标端口号	

--sport num 匹配来源端口号

命令选项输入顺序

```
iptables -t 表名 <-A/I/D/R> 规则链名 [规则号] <-i/o 网卡名> -p 协议名 <-s 源IP/源子网> --sport 源端口 <-d 目标IP/目标子网> --dport 目标端口 -j 动作
```

工作机制

规则链名包括(也被称为五个钩子函数（hook functions）)：

- **INPUT链** ：处理输入数据包。
- **OUTPUT链** ：处理输出数据包。
- **FORWARD链** ：处理转发数据包。
- **PREROUTING链** ：用于目标地址转换（DNAT）。
- **POSTOUTING链** ：用于源地址转换（SNAT）。

防火墙的策略

防火墙策略一般分为两种，一种叫通策略，一种叫堵策略，通策略，默认门是关着的，必须要定义谁能进。堵策略则是，大门是洞开的，但是你必须有身份认证，否则不能进。所以我们要定义，让进来的进来，让出去的出去，所以通，是要全通，而堵，则是要选择。当我们定义的策略的时候，要分别定义多条功能，其中：定义数据包中允许或者不允许的策略（filter过滤的功能，而定义地址转换的功能的则是nat选项。为了让这些功能交替工作，我们制定出了“ 表 ”这个定义，来定义、区分各种不同的工作功能和处理方式。

我们现在用的比较多个功能有3个：

1. filter 定义允许或者不允许的，只能做在3个链上（INPUT 、FORWARD 、OUTPUT
2. nat 定义地址转换的，也只能做在3个链上（PREROUTING 、OUTPUT 、POSTROUTING
3. mangle功能:修改报文原数据，是5个链都可以做（PREROUTING、INPUT、FORWARD、OUTPUT、POSTROUTING

我们修改报文原数据就是来修改TTL的。能够实现将数据包的元数据拆开，在里面做标记/修改内容的。而防火墙标记，其实就是靠mangle来实现的。

小扩展:

- 对于filter来讲一般只能做在3个链上（INPUT 、FORWARD 、OUTPUT
- 对于nat来讲一般也只能做在3个链上（PREROUTING 、OUTPUT 、POSTROUTING
- 而mangle则是5个链都可以做（PREROUTING、INPUT、FORWARD、OUTPUT、POSTROUTING

iptables/netfilter（这款软件）是工作在用户空间的，它可以让规则进行生效的，本身不是一种服务，而且规则是立即生效的。而我们iptables现在被做成了一个服务，可以进行启动，停止的。启动，则将规则直接生效，停止，则将规则撤销。

iptables还支持自己定义链。但是自己定义的链，必须是跟某种特定的链关联起来的。在一个关卡设定，指定当有数据的时候专门去找某个特定的链来处理，当那个链处理完之后，再返回。接着在特定的链中继续检查。

注意：规则的次序非常关键，谁的规则越严格，应该放的越靠前，而检查规则的时候，是按照从上往下的方式进行检查的。

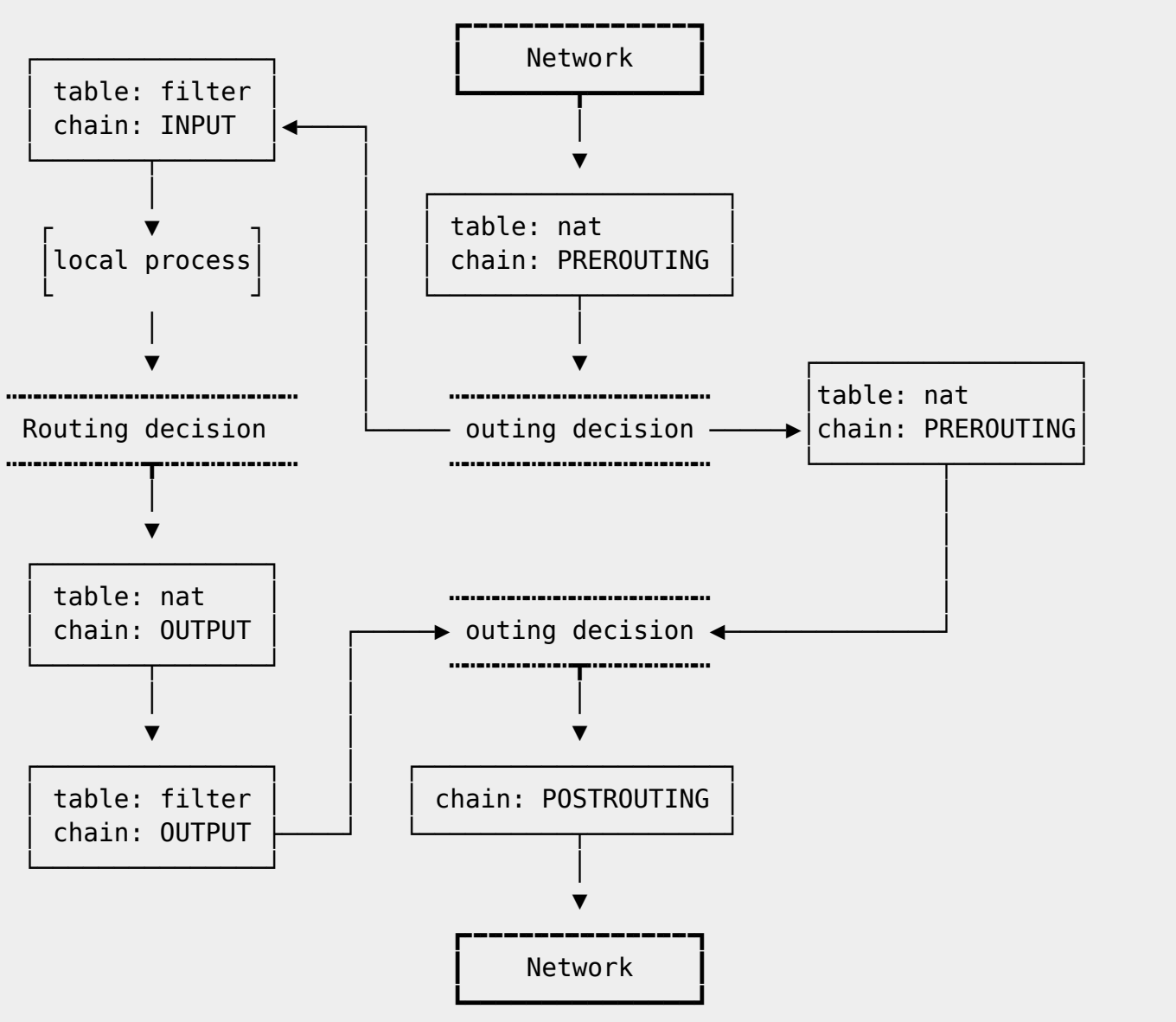
表名包括：

- **raw** ：高级功能，如：网址过滤。

- **mangle** ：数据包修改（QOS），用于实现服务质量。
- **nat** ：地址转换，用于网关路由器。
- **filter** ：包过滤，用于防火墙规则。

动作包括：

- **ACCEPT** ：接收数据包。
- **DROP** ：丢弃数据包。
- **REDIRECT** ：重定向、映射、透明代理。
- **SNAT** ：源地址转换。
- **DNAT** ：目标地址转换。
- **MASQUERADE** ：IP伪装（NAT），用于ADSL。
- **LOG** ：日志记录。
- **SEMARK**：添加SEMARK标记以供网域内强制访问控制（MAC）



实例

清空当前的所有规则和计数

```
iptables -F  # 清空所有的防火墙规则
iptables -X  # 删除用户自定义的空链
```

```
iptables -Z  # 清空计数
```

配置允许ssh端口连接

```
iptables -A INPUT -s 192.168.1.0/24 -p tcp --dport 22 -j ACCEPT
# 22为你的ssh端口  -s 192.168.1.0/24表示允许这个网段的机器来连接，其它网段的ip地址是登陆不了你的机器的  -j ACCEPT表示接受这样的请求
```

允许本地回环地址可以正常使用

```
iptables -A INPUT -i lo -j ACCEPT
#本地圆环地址就是那个127.0.0.1，是本机上使用的,它进与出都设置为允许
iptables -A OUTPUT -o lo -j ACCEPT
```

设置默认的规则

```
iptables -P INPUT DROP # 配置默认的不让进
iptables -P FORWARD DROP # 默认的不允许转发
iptables -P OUTPUT ACCEPT # 默认的可以出去
```

配置白名单

```
iptables -A INPUT -p all -s 192.168.1.0/24 -j ACCEPT  # 允许机房内网机器可以访问
iptables -A INPUT -p all -s 192.168.140.0/24 -j ACCEPT  # 允许机房内网机器可以访问
iptables -A INPUT -p tcp -s 183.121.3.7 --dport 3380 -j ACCEPT # 允许183.121.3.7访问本机的3380端口
```

开启相应的服务端口

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT # 开启80端口，因为web对外都是这个端口
iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT # 允许被ping
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT # 已经建立的连接得让它进来
```

保存规则到配置文件中

```
cp /etc/sysconfig/iptables /etc/sysconfig/iptables.bak # 任何改动之前先备份，请保持这一优秀的习惯
iptables-save > /etc/sysconfig/iptables
cat /etc/sysconfig/iptables
```

列出已设置的规则

> iptables -L [-t 表名] [链名]

- 四个表名 raw，nat，filter，mangle
- 五个规则链名 INPUT，OUTPUT，FORWARD，PREROUTING，POSTROUTING
- filter表包含INPUT，OUTPUT，FORWARD三个规则链

```
iptables -L -t nat                    # 列出 nat 上面的所有规则
#            ^ -t 参数指定，必须是 raw， nat，filter，mangle 中的一个
iptables -L -t nat  --line-numbers  # 规则带编号
iptables -L INPUT

iptables -L -nv  # 查看，这个列表看起来更详细
```

清除已有规则

```
iptables -F INPUT  # 清空指定链 INPUT 上面的所有规则
iptables -X INPUT  # 删除指定的链，这个链必须没有被其它任何规则引用，而且这条上必须没有任何规则。
       # 如果没有指定链名，则会删除该表中所有非内置的链。
iptables -Z INPUT  # 把指定链，或者表中的所有链上的所有计数器清零。
```

删除已添加的规则

```
# 添加一条规则
iptables -A INPUT -s 192.168.1.5 -j DROP
```

将所有iptables以序号标记显示，执行：

```
iptables -L -n --line-numbers
```

比如要删除INPUT里序号为8的规则，执行：

```
iptables -D INPUT 8
```

开放指定的端口

```
iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT               #允许本地回环接口(即运行本机访问本机)
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT    #允许已建立的或相关连的通行
iptables -A OUTPUT -j ACCEPT         #允许所有本机向外的访问
iptables -A INPUT -p tcp --dport 22 -j ACCEPT    #允许访问22端口
iptables -A INPUT -p tcp --dport 80 -j ACCEPT    #允许访问80端口
iptables -A INPUT -p tcp --dport 21 -j ACCEPT    #允许ftp服务的21端口
iptables -A INPUT -p tcp --dport 20 -j ACCEPT    #允许FTP服务的20端口
iptables -A INPUT -j reject       #禁止其他未允许的规则访问
iptables -A FORWARD -j REJECT     #禁止其他未允许的规则访问
```

屏蔽IP

```
iptables -A INPUT -p tcp -m tcp -s 192.168.0.8 -j DROP  # 屏蔽恶意主机（比如，192.168.0.8
iptables -I INPUT -s 123.45.6.7 -j DROP        #屏蔽单个IP的命令
iptables -I INPUT -s 123.0.0.0/8 -j DROP       #封整个段即从123.0.0.1到123.255.255.254的命令
iptables -I INPUT -s 124.45.0.0/16 -j DROP     #封IP段即从123.45.0.1到123.45.255.254的命令
```

```
iptables -I INPUT -s 123.45.6.0/24 -j DROP     #封IP段即从123.45.6.1到123.45.6.254的命令是
```

指定数据包出去的网络接口

只对 OUTPUT、FORWARD、POSTROUTING 三个链起作用。

```
iptables -A FORWARD -o eth0
```

查看已添加的规则

```
iptables -L -n -v
Chain INPUT (policy DROP 48106 packets, 2690K bytes)
 pkts bytes target     prot opt in     out     source               destination
 5075  589K ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
 191K   90M ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
1499K  133M ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
4364K 6351M ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
 6256  327K ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0

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

Chain OUTPUT (policy ACCEPT 3382K packets, 1819M bytes)
 pkts bytes target     prot opt in     out     source               destination
 5075  589K ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0
```

启动网络转发规则

公网210.14.67.7让内网192.168.188.0/24上网

```
iptables -t nat -A POSTROUTING -s 192.168.188.0/24 -j SNAT --to-source 210.14.67.127
```

端口映射

本机的 2222 端口映射到内网 虚拟机的22 端口

```
iptables -t nat -A PREROUTING -d 210.14.67.127 -p tcp --dport 2222  -j DNAT --to-dest 192.168.188.115:22
```

字符串匹配

比如，我们要过滤所有TCP连接中的字符串test，一旦出现它我们就终止这个连接，我们可以这么做：

```
iptables -A INPUT -p tcp -m string --algo kmp --string "test" -j REJECT --
reject-with tcp-reset
iptables -L

# Chain INPUT (policy ACCEPT)
# target     prot opt source               destination
# REJECT     tcp  --  anywhere             anywhere            STRING match
"test" ALGO name kmp TO 65535 reject-with tcp-reset
#
# Chain FORWARD (policy ACCEPT)
# target     prot opt source               destination
#
# Chain OUTPUT (policy ACCEPT)
# target     prot opt source               destination
```

阻止Windows蠕虫的攻击

```
iptables -I INPUT -j DROP -p tcp -s 0.0.0.0/0 -m string --algo kmp --string
"cmd.exe"
```

防止SYN洪水攻击

```
iptables -A INPUT -p tcp --syn -m limit --limit 5/second -j ACCEPT
```

添加SECMARK记录

```
iptables -t mangle -A INPUT -p tcp --src 192.168.1.2 --dport 443 -j SECMARK
--selctx system_u:object_r:myauth_packet_t
# 向从 192.168.1.2:443 以TCP方式发出到本机的包添加MAC安全上下文
system_u:object_r:myauth_packet_t
```

更多实例

> 用iptables搭建一套强大的安全防护盾 http://www.imooc.com/learn/389

iptables: linux 下应用层防火墙工具

iptables 5链: 对应 Hook point netfilter: linux 操作系统核心层内部的一个数据包处理模块 Hook point: 数据包在 netfilter 中的挂载点; PRE_ROUTING / INPUT / OUTPUT / FORWARD / POST_ROUTING

iptables & netfilter

iptables 4表5链

iptables rules

- 4表

filter: 访问控制 / 规则匹配 **nat**: 地址转发 mangle / raw

- 规则

数据访问控制: ACCEPT / DROP / REJECT 数据包改写(nat -> 地址转换): snat / dnat 信息记录: log

使用场景实例

- 场景一

开放 tcp 10-22/80 端口 开放 icmp 其他未被允许的端口禁止访问

存在的问题: 本机无法访问本机; 本机无法访问其他主机

- 场景二

ftp: 默认被动模式(服务器产生随机端口告诉客户端, 客户端主动连接这个端口拉取数据) vsftpd: 使 ftp 支持主动模式(客户端产生随机端口通知服务器, 服务器主动连接这个端口发送数据)

- 场景三

允许外网访问: web http -> 80/tcp; https -> 443/tcp mail smtp -> 25/tcp; smtps -> 465/tcp pop3 -> 110/tcp; pop3s -> 995/tcp imap -> 143/tcp

内部使用: file nfs -> 123/udp samba -> 137/138/139/445/tcp ftp -> 20/21/tcp remote ssh -> 22/tcp sql mysql -> 3306/tcp oracle -> 1521/tcp

- 场景四

nat 转发

- 场景五

防CC攻击

```
iptables -L -F -A -D # list flush append delete
# 场景一
iptables -I INPUT -p tcp --dport 80 -j ACCEPT # 允许 tcp 80 端口
iptables -I INPUT -p tcp --dport 10:22 -j ACCEPT # 允许 tcp 10-22 端口
iptables -I INPUT -p icmp -j ACCEPT # 允许 icmp
iptables -A INPUT -j REJECT # 添加一条规则, 不允许所有

# 优化场景一
iptables -I INPUT -i lo -j ACCEPT # 允许本机访问
iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT # 允许访问外网
iptables -I INPUT -p tcp --dport 80 -s 10.10.188.233 -j ACCEPT # 只允许固定ip访问80

# 场景二
vi /etc/vsftpd/vsftpd.conf # 使用 vsftpd 开启 ftp 主动模式
port_enable=yes
```

```
connect_from_port_20=YES
iptables -I INPUT -p tcp --dport 21 -j ACCEPT

vi /etc/vsftpd/vsftpd.conf # 建议使用 ftp 被动模式
pasv_min_port=50000
pasv_max_port=60000
iptables -I INPUT -p tcp --dport 21 -j ACCEPT
iptables -I INPUT -p tcp --dport 50000:60000 -j ACCEPT

# 还可以使用 iptables 模块追踪来自动开发对应的端口

# 场景三
iptables -I INPUT -i lo -j ACCEPT # 允许本机访问
iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT # 允许访问外网
iptables -I INPUT -s 10.10.155.0/24 -j ACCEPT # 允许内网访问
iptables -I INPUT -p tcp -m multiport --dports 80,1723 -j ACCEPT # 允许端口, 80 -> http, 1723 -> vpn
iptables -A INPUT -j REJECT # 添加一条规则, 不允许所有

iptables-save # 保存设置到配置文件

# 场景四
iptables -t nat -L # 查看 nat 配置

iptables -t nat -A POST_ROUTING -s 10.10.177.0/24 -j SNAT --to 10.10.188.232 # SNAT
vi /etc/sysconfig/network # 配置网关

iptables -t nat -A POST_ROUTING -d 10.10.188.232 -p tcp --dport 80 -j DNAT --to 10.10.177.232:80 # DNAT

#场景五
iptables -I INPUT -p tcp --syn --dport 80 -m connlimit --connlimit-above 100 -j REJECT # 限制并发连接访问数
iptables -I INPUT -m limit --limit 3/hour --limit-burst 10 -j ACCEPT # limit 模块; --limit-burst 默认为5
```

From:
https://rd.irust.top/ - **学习笔记**

Permanent link:
https://rd.irust.top/doku.php?id=command:iptables

Last update: **2021/10/15 14:58**