

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

| 参数 | 作用 |
|-------------|----------------------------------|
| -P | 设置默认策略:iptables -P INPUT (DROP |
| -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)。

POSTROUTING 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，但是你必须要有身份认证，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：

filter 定义允许或者不允许的，只能做在 3 个链上：INPUT ， FORWARD ， OUTPUT

nat 定义地址转换的，也只能做在 3 个链上：PREROUTING ， OUTPUT ， POSTROUTING

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。

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，但是你必须要有身份认证，否则不能进，所以通，是要全通，而堵，则是要选择。

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。

net：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。

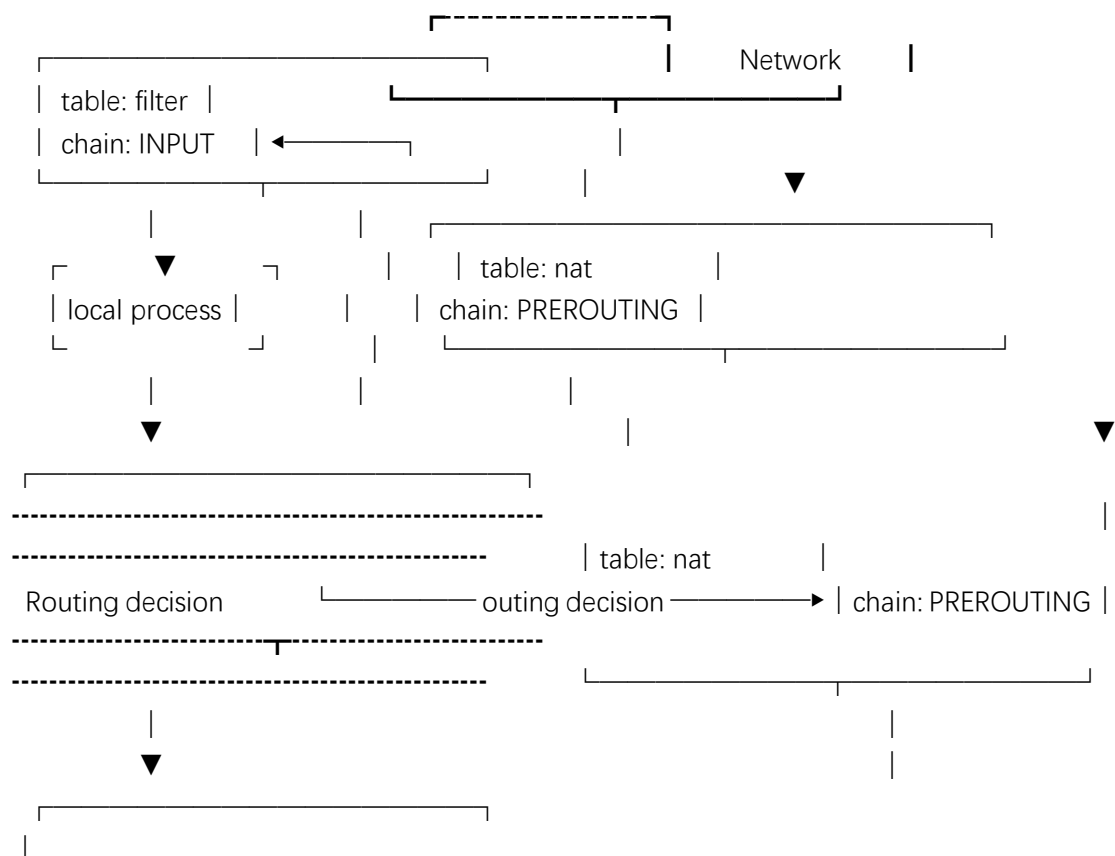
REDIRECT：重定向、映射、透明代理。

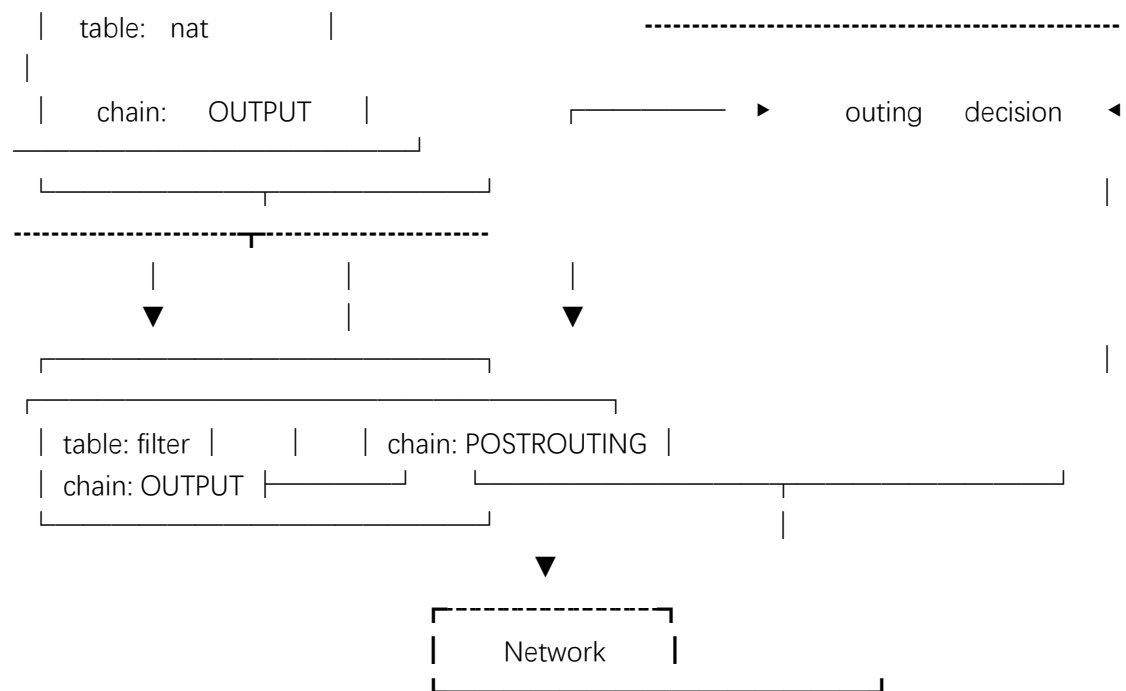
SNAT：源地址转换。

DNAT：目标地址转换。

MASQUERADE：IP 伪装（NAT），用于 ADSL。

LOG：日志记录。





实例

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

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
```