# TTS 10.0 COOKBOOK

## (NSD SECURITY DAY06)

**版本编号 10.0**



**2019-06**

**达内 IT 培训集团**

# NSD SECURITY DAY06

## 1. 案例 1：iptables 基本管理

- **问题**

本案例要求练习 iptables 命令的使用，按照要求完成以下任务：

1) 关闭 firewalld，开启 iptables 服务
2) 查看防火墙规则
3) 追加、插入防火墙规则
4) 删除、清空防火墙规则

- **步骤**

实现此案例需要按照如下步骤进行。

**步骤一：关闭 firewalld，启动 iptables 服务**

1）关闭 firewalld 服务器

```
[root@proxy ~]# systemctl stop firewalld.service
[root@proxy ~]# systemctl disable firewalld.service
```

2）安装 iptables-services 并启动服务

```
[root@proxy ~]# yum -y install iptables-services
[root@proxy ~]# systemctl start iptables.service
```

**步骤二：熟悉 iptables 框架**

1）iptables 的 4 个表（区分大小写）：

iptables 默认有 4 个表，nat 表（地址转换表）、filter 表（数据过滤表）、raw 表（状态跟踪表）、mangle 表（包标记表）。

2）iptables 的 5 个链（区分大小写）：

INPUT 链（入站规则）

OUTPUT 链（出站规则）

FORWARD 链（转发规则）

PREROUTING 链（路由前规则）

POSTROUTING 链（路由后规则）

**步骤三：iptables 命令的基本使用方法**

1）iptabels 语法格式

```
[root@proxy ~]# iptables [-t 表名] 选项 [链名] [条件] [-j 目标操作]
[root@proxy ~]# iptables -t filter -I INPUT -p icmp -j REJECT
```

```
[root@proxy ~]# iptables -t filter -I  INPUT   -p  icmp  -j  ACCEPT
[root@proxy ~]# iptables  -I  INPUT  -p  icmp  -j  REJECT
//注意事项与规律:
//可以不指定表，默认为 filter 表
//可以不指定链，默认为对应表的所有链
//如果没有找到匹配条件，则执行防火墙默认规则
//选项/链名/目标操作用大写字母，其余都小写
########################################################################
//目标操作:
// ACCEPT：允许通过/放行
// DROP：直接丢弃，不给出任何回应
// REJECT：拒绝通过，必要时会给出提示
// LOG：记录日志，然后传给下一条规则
```

iptables 命令的常用选项如表-1 所示。

表-1 iptables 常用选项

| 类别 | 选项 | 描述 |
|---|---|---|
| 添加规则 | -A | 追加一条防火墙规则至链的末尾位置 |
| | -I | 插入一条防火墙规则至链的开头 |
| 查看规则 | -L | 查看 iptables 所有规则 |
| | -n | 以数字形式显示地址、端口等信息 |
| | --line-numbers | 查看规则时，显示规则的行号 |
| 删除规则 | -D | 删除链内指定序号（或内容）的一条规则 |
| | -F | 清空所有的规则 |
| 默认规则 | -P | 为指定的链设置默认规则 |

2）iptables 命令的使用案例

创建规则的案例：

```
[root@proxy ~]# iptables  -t  filter  -A  INPUT  -p tcp  -j  ACCEPT
//追加规则至 filter 表中的 INPUT 链的末尾，允许任何人使用 TCP 协议访问本机
[root@proxy ~]# iptables  -I  INPUT  -p  udp  -j  ACCEPT
//插入规则至 filter 表中的 INPUT 链的开头，允许任何人使用 UDP 协议访问本机
[root@proxy ~]# iptables  -I  INPUT 2  -p  icmp  -j  ACCEPT
//插入规则至 filter 表中的 INPUT 链的第 2 行，允许任何人使用 ICMP 协议访问本机
```

查看 iptables 防火墙规则

```
[root@proxy ~]# iptables  -nL  INPUT                    //仅查看 INPUT 链的规则
target     prot opt source               destination
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0
[root@proxy ~]# iptables  -L  INPUT  --line-numbers     //查看规则，显示行号
num  target     prot opt source               destination
1    ACCEPT     udp  --  anywhere             anywhere
2    ACCEPT     icmp --  anywhere             anywhere
3    ACCEPT     tcp  --  anywhere             anywhere
```

删除规则，清空所有规则

```
[root@proxy ~]# iptables  -D  INPUT  3
//删除 filter 表中 INPUT 链的第 3 条规则
[root@proxy ~]# iptables  -nL  INPUT                    //查看规则，确认是否删除
[root@proxy ~]# iptables  -F
//清空 filter 表中所有链的防火墙规则
[root@proxy ~]# iptables  -t  nat  -F
//清空 nat 表中所有链的防火墙规则
[root@proxy ~]# iptables  -t  mangle  -F
//清空 mangle 表中所有链的防火墙规则
[root@proxy ~]# iptables  -t  raw  -F
//清空 raw 表中所有链的防火墙规则
```

设置防火墙默认规则

```
[root@proxy ~]# iptables  -t  filter  -P  INPUT  DROP
[root@proxy ~]# iptables  -nL
Chain INPUT (policy DROP)
… …
```

## 2. 案例 2：filter 过滤和转发控制

- **问题**

本案例要求创建常用主机防火墙规则以及网络防火墙规则：

1) 针对 Linux 主机进行出站、入站控制
2) 利用 ip_forward 机制实现 Linux 路由/网关功能
3) 在 Linux 网关上实现数据包转发访问控制

- **方案**

根据防火墙保护的对象不同，防火墙可以分为主机型防火墙与网络型防火墙，如图-1 所示。

主机型防火墙，主要保护的是服务器本机（过滤威胁本机的数据包）。

网络防火墙，主要保护的是防火墙后面的其他服务器，如 web 服务器、FTP 服务器等。

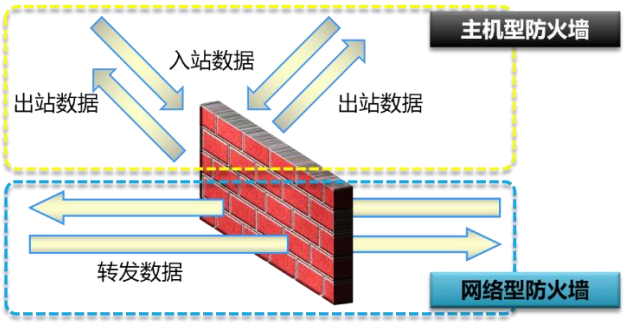

图-1

- **步骤**

实现此案例需要按照如下步骤进行。

**步骤一：iptables 防火墙规则的条件**

iptables 防火墙可以根据很多很灵活的规则进行过滤行为，具体常用的过滤条件如表-2 所示。

表-2 iptables 过滤条件

| 类别 | 选项 | 用法 |
| --- | --- | --- |
| 通用匹配 | 协议匹配 | -p 协议名称 |
| | 地址匹配 | -s 源地址、-d 目标地址 |
| | 接口匹配 | -i 接受数据的网卡、-o 发送数据的网卡 |
| 隐含匹配 | 端口匹配 | --sport 源端口号、--dport 目标端口号 |
| | ICMP 类型匹配 | --icmp-type ICMP 类型 |

1）主机型防火墙案例

```
[root@proxy ~]# iptables -I INPUT -p tcp --dport 80 -j REJECT
[root@proxy ~]# iptables -I INPUT -s 192.168.2.100 -j REJECT
[root@proxy ~]# iptables -I INPUT -d 192.168.2.5 -p tcp --dport 80 -j REJECT
[root@proxy ~]# iptables -I INPUT -i eth0 -p tcp --dport 80 -j REJECT
[root@proxy ~]# iptables  -A  INPUT  -s 192.168.4.100  -j  DROP
//丢弃 192.168.4.100 发给本机的所有数据包
[root@proxy ~]# iptables  -A  INPUT -s  192.168.2.0/24  -j  DROP
//丢弃 192.168.2.0/24 网络中所有主机发送给本机的所有数据包
[root@proxy ~]# iptables -A  INPUT -s  114.212.33.12  -p tcp --dport 22 -j  REJECT
//拒绝 114.212.33.12 使用 tcp 协议远程连接本机 ssh（22 端口）
```

**步骤二：开启 Linux 的路由转发功能**

1）Linux 内核默认支持软路由功能，通过修改内核参数即可开启或关闭路由转发功能。

```
[root@proxy ~]# echo 0 > /proc/sys/net/ipv4/ip_forward          //关闭路由转发
[root@proxy ~]# echo 1 > /proc/sys/net/ipv4/ip_forward          //开启路由转发
//注意以上操作仅当前有效，计算机重启后无效
[root@proxy ~]# echo 'net.ipv4.ip_forward=1' >> /etc/sysctl.conf
//修改/etc/sysctl.conf 配置文件，可以实现永久有效规则
```

**步骤四：网络型防火墙案例**

1）网络型防火墙案例

部署如表-3 所示的网络拓扑，一定要把 proxy 主机的路由转发功能打开。

表-3 实验拓扑

| 主机名要求 | 网卡、IP 地址以及网关设置要求 |
| --- | --- |
| client | eth0:192.168.4.100 |

| | |
|---|---|
| | 网关：192.168.4.5 |
| proxy | eth0:192.168.4.5<br>eth1:192.168.2.5 |
| web1 | eth1:192.168.2.100<br>网关：192.168.2.5 |

添加网关的命令

```
[root@client ~]# nmcli connection modify eth0 ipv4.gateway 192.168.4.5
[root@client ~]# nmcli connection up eth0

[root@web1 ~]# nmcli connection modify eth1 ipv4.gateway 192.168.2.5
[root@web1 ~]# nmcli connection up eth1
```

确认不同网络的联通性

```
[root@client ~]# ping 192.168.2.100
[root@web1 ~]# ping 192.168.4.100
```

在 web1 主机上启动 http 服务

```
[root@web1 ~]# yum -y install httpd
[root@web1 ~]# echo "test page" > /var/www/html/index.html
[root@web1 ~]# systemctl restart httpd
```

没有防火墙的情况下 client 访问 web 服务

```
[root@client ~]# curl http://192.168.2.100                    //成功
```

设置 proxy 主机的防火墙规则，保护防火墙后面的 Web 服务器

```
[root@proxy ~]# iptables -I FORWARD -s 192.168.4.100 -p tcp --dport 80 -j DROP
```

设置完防火墙规则后，再次使用 client 客户端访问测试效果

```
[root@client ~]# curl http://192.168.2.100                    //失败
```

**步骤三：禁 ping 的相关策略**

1）默认直接禁 ping 的问题?

```
[root@proxy ~]# iptables -I INPUT -p icmp -j DROP
//设置完上面的规则后，其他主机确实无法 ping 本机，但本机也无法 ping 其他主机
//当本机 ping 其他主机，其他主机回应也是使用 icmp，对方的回应被丢弃
```

2）禁止其他主机 ping 本机，允许本机 ping 其他主机

```
[root@proxy ~]# iptables  -A  INPUT  -p icmp  \
> --icmp-type echo-request  -j  DROP
//仅禁止入站的 ping 请求，不拒绝入站的 ping 回应包
```

注意：关于 ICMP 的类型，可以参考 help 帮助，参考命令如下：

```
[root@proxy ~]# iptables -p icmp --help
```

## 3. 案例 3：防火墙扩展规则

- **问题**

本案例要求熟悉使用 iptables 的扩展规则，实现更丰富的过滤功能，完成以下任务：

1) 根据 MAC 地址封锁主机
2) 在一条规则中开放多个 TCP 服务
3) 根据 IP 范围设置封锁规则

- **方案**

iptables 在基本过滤条件的基础上还扩展了很多其他条件，在使用时需要使用-m 参数来启动这些扩展功能，语法如下：

iptables 选项 链名称 -m 扩展模块 --具体扩展条件 -j 动作

- **步骤**

实现此案例需要按照如下步骤进行。

**步骤一：根据 MAC 地址过滤**

1）根据 IP 过滤的规则，当对方修改 IP 后，防火墙会失效

```
[root@proxy ~]# iptables -F
[root@proxy ~]# iptables -I INPUT -s 192.168.4.100 -p tcp --dport 22 -j DROP
//设置规则禁止 192.168.4.100 使用 ssh 远程本机
```

但是，当 client 主机修改 IP 地址后，该规则就会失效，注意因为修改了 IP，对 client 主机的远程连接会断开，需要使用 virt-manager 开启虚拟机操作：

```
[root@client ~]# ifconfig eth0  192.168.4.101
[root@client ~]# ssh 192.168.4.5                    //依然成功
```

根据 MAC 地址过滤，可以防止这种情况的发生

```
[root@client ~]# ip link show eth0                  //查看 client 的 MAC 地址
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode
DEFAULT qlen 1000
link/ether 52:54:00:00:00:0b brd ff:ff:ff:ff:ff:ff

[root@proxy ~]# iptables  -A  INPUT  -p tcp --dport 22\
> -m   mac --mac-source  52:54:00:00:00:0b  -j  DROP
//拒绝 52:54:00:00:00:0b 这台主机远程本机
```

**步骤二：基于多端口设置过滤规则**

1）一次需要过滤或放行很多端口时会比较方便

```
[root@proxy ~]# iptables  -A  INPUT  -p tcp   \
> -m  multiport --dports  20:22,25,80,110,143,16501:16800  -j  ACCEPT
//一次性开启 20,21,22,25,80,110,143,16501 到 16800 所有的端口
```

提示，多端口还可以限制多个源端口，但因为源端口不固定，一般不会使用，限制多个源端口的参数是--sports.

**步骤三：根据 IP 地址范围设置规则**

1）允许从 192.168.4.10-192.168.4.20 登录

```
[root@proxy ~]# iptables  -A  INPUT  -p tcp  --dport  22  \
> -m  iprange  --src-range  192.168.4.10-192.168.4.20   -j  ACCEPT
```

注意，这里也可以限制多个目标 IP 的范围，参数是--dst-range,用法与--src-range一致。

2）禁止从 192.168.4.0/24 网段其他的主机登录

```
[root@proxy ~]# iptables -A INPUT -p tcp --dport 22  -s 192.168.4.0/24  -j  DROP
```

## 4. 案例 4：配置 SNAT 实现共享上网

- **问题**

本案例要求设置防火墙规则，允许位于局域网中的主机可以访问外网，主要包括下列服务：

1) 搭建内外网案例环境
2) 配置 SNAT 策略实现共享上网访问

- **步骤**

实现此案例需要按照如下步骤进行。

**步骤一：搭建内外网案例环境**

表-4 实验拓扑

| 主机名要求 | 网卡、IP 地址以及网关设置要求 |
|---|---|
| client | eth0:192.168.4.100<br>网关：192.168.4.5 |
| proxy | eth0:192.168.4.5<br>eth1:192.168.2.5 |
| web1 | eth1:192.168.2.100<br>网关：192.168.2.5 |

这里，我们设定 192.168.2.0/24 网络为外部网络，192.168.4.0/24 为内部网络。

现在，在外部网络中有一台 web 服务器 192.168.2.100，因为设置了网关，client 已经可以访问此 web 服务器了。但，如果查看 web1 的日志就会发现，日志里记录的是 192.168.4.100 在访问网页。

我们需要实现的效果是，client 可以访问 web 服务器，但要伪装为 192.168.2.5 后

再访问 web 服务器(模拟所有位于公司内部的电脑都使用的是私有 IP，希望访问外网，就需要伪装为公司的外网 IP 后才可以)。

**步骤二：设置防火墙规则，实现 IP 地址的伪装（SNAT 源地址转换）**

1）确保 proxy 主机开启了路由转发功能

```
[root@proxy ~]# echo 1 > /proc/sys/net/ipv4/ip_forward          //开启路由转发
```

2）设置防火墙规则，实现 SNAT 地址转换

```
[root@proxy ~]# iptables  -t  nat  -A POSTROUTING \
> -s  192.168.4.0/24 -p tcp --dport 80  -j SNAT  --to-source 192.168.2.5
```

3）登陆 web 主机查看日志

```
[root@proxy ~]# tail  /var/log/httpd/access_log
.. ..
192.168.2.5  -  -  [12/Aug/2018:17:57:10  +0800]  "GET  /  HTTP/1.1"  200  27  "-"
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)"
```

通过日志会发现，客户端是先伪装为了 192.168.2.5 之后再访问的 web 服务器!

4）扩展知识，对于 proxy 外网 IP 不固定的情况可以执行下面的地址伪装，动态伪装 IP。

```
[root@proxy ~]# iptables  -t  nat  -A POSTROUTING \
> -s  192.168.4.0/24 -p tcp --dport 80 -j MASQUERADE
```

最后，所有 iptables 规则都是临时规则，如果需要永久保留规则需要执行如下命令：

```
[root@proxy ~]# service  iptables save
```