12_security06iptables 防火墙+filter 表控制+扩展匹配+nat 典型应用

```
hostc ~]# systemctl status firewalld.service

hostc ~]# systemctl stop firewalld.service

hostc ~]# systemctl disable firewalld.service

hostc ~]# yum -y install iptables.services

hostc ~]# systemctl start iptables.service    #参见 1.5.2
```

一 iptables 防火墙-概述

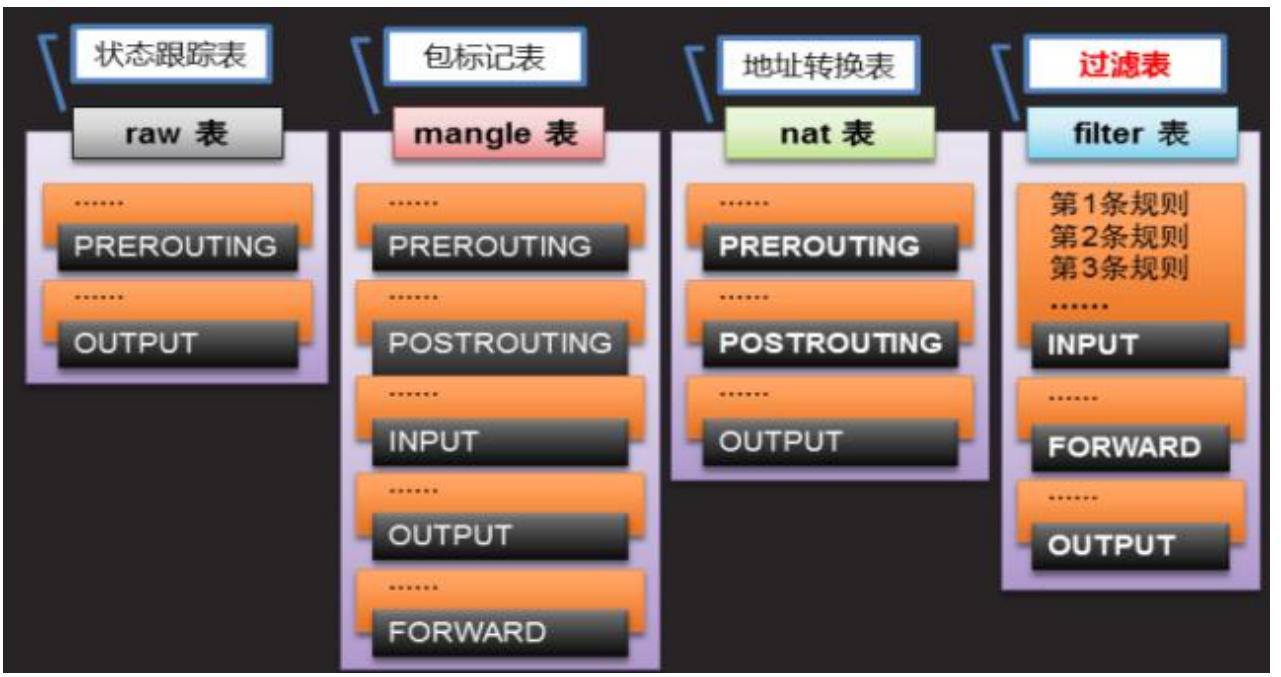
1.1 什么是防火墙

一道保护性的安全屏障;起保护\隔离公网和私网的作用的一种软件服务

1.2 Linux 包过滤型防火墙

RHEL7 默认使用 Firewall 作为防火墙,但 Firewalld 底层还是调用包过滤防火墙 iptables(RHEL5 6 使用的是 iptables)

1.3 iptables 的表\链结构



1.3.1 4 张表: 表是服务的功能分类

raw 表	对数据包做状态跟踪
mangle 表	对到访数据包打标记
nat 表	地址转换
filter 表	过滤表

1.3.2 5 条链: 链是数据包传输的方向(以防火墙主机为参照物定义的方向)

INPUT	匹配进入防火墙主机的数据包
OUTPUT	匹配从防火墙主机出去的数据包
FORWARD	匹配经过防火墙主机的数据包
POSTROUTING	路由后处理
PREROUTING	路由前处理

1.4 包过滤匹配流程

链内规则的匹配顺序:

顺序对比,匹配即停止(**LOG** 除外)

若无任何匹配,则按该链的默认策略处理

1.5 防火墙主机类型

1.5.1 主机型防火墙(自己开防火墙保护自己),使用 **filter** 表 **INPUT** 链

1.5.2 网络型防火墙(控制数据包是否允许经过自己),使用 **filter** 表 **FORWARD** 链

hostC eth0 192.168.4.51 eth1 192.168.2.51 防火墙主机

hostA eth0 192.168.4.50 hostB eth1 192.168.2.52 网络主机

二 iptables 用法解析

2.1 iptables 基本用法

2.2.1 管理程序位置: /sbin/iptables

2.2.2 指令组成

iptables [-t 表名] 选项 [链名] [匹配条件] [-j 处理操作]

2.2.3 注意事项/整体规律

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

如果没有匹配的规则,则使用防火墙默认规则

选项\链名\处理动作用大写字母,其余都小写

2.2 基本的处理操作

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(此目标操作是匹配即停止的唯一例外)

SNAT: 源地址转换,转换数据包中的源地址

DNAT: 目标地址转换,转换数据包中的目标地址

2.3 常用的管理选项

类别	选项	用途
添加规则	-A	在链的末尾追加一条规则
	-I	在链的开头（或指定序号）插入一条规则
查看规则	-L	列出所有的规则条目
	-n	以数字形式显示地址、端口等信息
	--line-numbers	查看规则时，显示规则的序号
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
默认策略	-P	为指定的链设置默认规则

2.4 常用匹配条件

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

协议匹配需要取反条件时,用叹号!

三 iptables 规则管理示例

3.1 查看规则列表

-L 查看

```
hostc ~]# iptables -t filter -L

hostc ~]# iptables -t filter -nL                    #以数字显示网络

hostc ~]# iptables -t filter -nL --line-numbers  #列出规则序号

hostc ~]# iptables -t nat -nL --line-numbers     #默认规则为空

hostc ~]# iptables -t mangle -nL --line-numbers  #默认规则为空

hostc ~]# iptables -t raw -nL --line-numbers     #默认规则为空
```

3.2 删除\清空规则

-D 删除 **-F** 清空

```
hostc ~]# iptables -t raw -F           #临时清空规则

hostc ~]# iptables -t mangle -F        #临时清空规则

hostc ~]# iptables -t nat -F           #临时清空规则

hostc ~]# iptables -t filter -F        #临时清空规则

hostc ~]# iptables-save > /etc/sysconfig/iptables

      #将临时清空规则写入配置文件,达到永久清空的目录

hostc ~]# systemctl restart iptables
```

此时再按 3.1 查看 4 张表,4 张表中链内的规则全部为空

3.3 添加新的规则

-A 在链末尾追加一条规则　-I 在链的开头插入一条规则

3.4 设置默认规则

所有链的初始默认规则均为 ACCEPT

通过-P 选项可重置默认规则

　ACCEPT 或者 DROP

案例 1:主机型防火墙-仅允许连接到本机(192.168.4/2.51)的 ssh 服务

先允许连接到本主机的 ssh 服务,再拒绝其他所有的连接到本主机的服务请求,最后保存为永久生效

```
hostc ~]# iptables -t filter -A INPUT -p tcp --dport 22 -j ACCEPT
hostc ~]# iptables -t filter -nL INPUT --line-numbers
hostc ~]# iptables -t filter -P INPUT DROP
hostc ~]# iptables -t filter -nL INPUT --line-numbers
hostc ~]# iptables-save > /etc/sysconfig/iptables
```

测试本机的 ssh 连接服务,显示成功;测试 ping 本机,失败

```
room9pc01 ~]$ ssh -X root@192.168.2.51
room9pc01 ~]$ ssh -X root@192.168.4.51
hosta ~]# ping 192.168.4.51     #ping 不通
hostb ~]# ping 192.168.2.51     #ping 不通
```

案例 2:主机型防火墙-设置控制只能通过 192.168.2.0/24 网段连接到本机(192.168.4/2.51)的 ssh 服务

先在 filter 表 INPUT 链插入第 1 条规则,设置允许通过 2.51 连接本机的 ssh 服务

```
hostc ~]# iptables -t filter -I INPUT 1 -s 192.168.2.0/24 -p tcp --dport 22 -j ACCEPT  #-I,链开头插入,1 表示插入为第 1 条规则
```

查看 filter 表 INPUT 链规则,刚才插入的规则在此时为第 1 条

```
hostc ~]# iptables -t filter -nL INPUT --line-numbers
Chain INPUT (policy DROP)
num  target  prot opt source          destination
1    ACCEPT tcp --   192.168.2.0/24  0.0.0.0/0  tcp dpt:22
2    ACCEPT tcp  --    0.0.0.0/0      0.0.0.0/0  tcp dpt:22
```

删除 filter 表 INPUT 链第 2 条规则,该规则允许所有网段连接本机的 ssh 服务

```
hostc ~]# iptables -t filter -D INPUT 2
```

删除后查看 filter 表 INPUT 链规则

```
hostc ~]# iptables -t filter -nL INPUT --line-numbers
Chain INPUT (policy DROP)
num  target prot opt   source          destination
1   ACCEPT tcp   --   192.168.2.0/24  0.0.0.0/0     tcp dpt:22
```

此时,通过 4.51 连接 192.168.4.51 的 ssh 服务断开了

```
hostc ~]# iptables-save > /etc/sysconfig/iptables  #保存为永久
```

案例 3: 主机型防火墙-设置仅允许通过 4.51 访问本机的 http 服务(tcp 协议,80 端口)

关闭 iptables 服务,安装 httpd 服务,创建测试文件,并访问测试

```
hostc ~]# systemctl stop iptables
hostc ~]# yum -y install httpd
hostc ~]# systemctl start httpd
hostc ~]# echo "xxx" > /var/www/html/test.html
hosta ~]# curl http://192.168.4.51/test.html  #返回内容 xxx
hostb ~]# curl http://192.168.2.51/test.html  #返回内容 xxx
```

开启 iptables 服务,查看 filter 表 INPUT 链规则,并访问测试及抓包

```
hostc ~]# systemctl start iptables
hostc ~]# iptables -t filter -nL --line-numbers  #确认有规则
hosta ~]# curl http://192.168.4.51/test.html  #无回应
hostb ~]# curl http://192.168.2.51/test.html  #无回应
hostc ~]# tcpdump -i eth0/eth1 tcp port 80
#能抓到 192.168.4.50(2.52)访问本机 http 服务的 80 端口的包
```

在 filter 表 INPUT 链添加规则,设置能通过 4.51 访问本机的 http 服务

```
hostc ~]# iptables -t filter -A INPUT -d 192.168.4.51 -p tcp --dport
80 -j ACCEPT
hosta ~]# curl http://192.168.4.51/test.html    #有结果 xxx
```

```
hostb ~]# curl http://192.168.2.51/test.html    #无回应
```

再在 **filter** 表 **INPUT** 链添加规则,设置能通过 **2.51** 访问本机的 **http** 服务

```
hostc ~]# iptables -t filter -A INPUT -d 192.168.2.51 -p tcp --dport
80 -j ACCEPT
```

查看在 **filter** 表 **INPUT** 链内规则,此时有 **3** 条规则

```
hostc ~]# iptables -t filter -nL --line-numbers
Chain INPUT (policy DROP)
1 ACCEPT tcp  -- 92.168.2.0/24  0.0.0.0/0       tcp dpt:22
2 ACCEPT tcp  -- 0.0.0.0/0      192.168.4.51    tcp dpt:80
3 ACCEPT tcp  -- 0.0.0.0/0      192.168.2.51    tcp dpt:80
```

测试通过 **2.51** 访问本机 **http** 服务,成功

```
hostb ~]# curl http://192.168.2.51/test.html    #有结果 xxx
hostc ~]# iptables-save > /etc/sysconfig/iptables  #保存为永久
```

案例 4:主机型防火墙-允许本机(192.168.4.51/2.51)ping 其他主机,但是,禁止及其他主机 ping 本机

停止 **iptalbes** 服务,抓取 **ping** 的数据包

```
hostc ~]# systemctl stop iptables
hosta ~]# ping 192.168.4.51           #50ping51
hostc ~]# tcpdump -i eth0 -c 2 icmp   #抓取 50ping51 的 2 个包
```

IP 192.168.4.50 > hostc: ICMP echo request #50ping51 请求包

IP hostc > 192.168.4.50: ICMP echo reply   #51 回应 50 包

hostc ~]# ping 192.168.4.50   #51ping50

hostc ~]# tcpdump -i eth0-c 2 icmp   #51ping50 的 2 个包

IP hostc > 192.168.4.50: ICMP echo request #51ping50 请求包

IP 192.168.4.50 > hostc: ICMP echo reply   #50 回应 51 包

启动 iptables 服务,查看 filter 表 INPUT 链规则

hostc ~]# systemctl start iptables

hostc ~]# iptables -t filter -nL --line-numbers

#此时,默认放行从本机出的数据包,允许 tcp 的 22 和 80 端口的请求进入

在 filter 表 INPUT 链添加规则,放行本机 ping 其他主机的回应包(echo-reply)

hostc ~]# iptables -t filter -A INPUT -p icmp --icmp-type

echo-reply -j ACCEPT

hostc ~]# ping 192.168.4.50   #能 ping 通

hostc ~]# ping 192.168.2.52   #能 ping 通

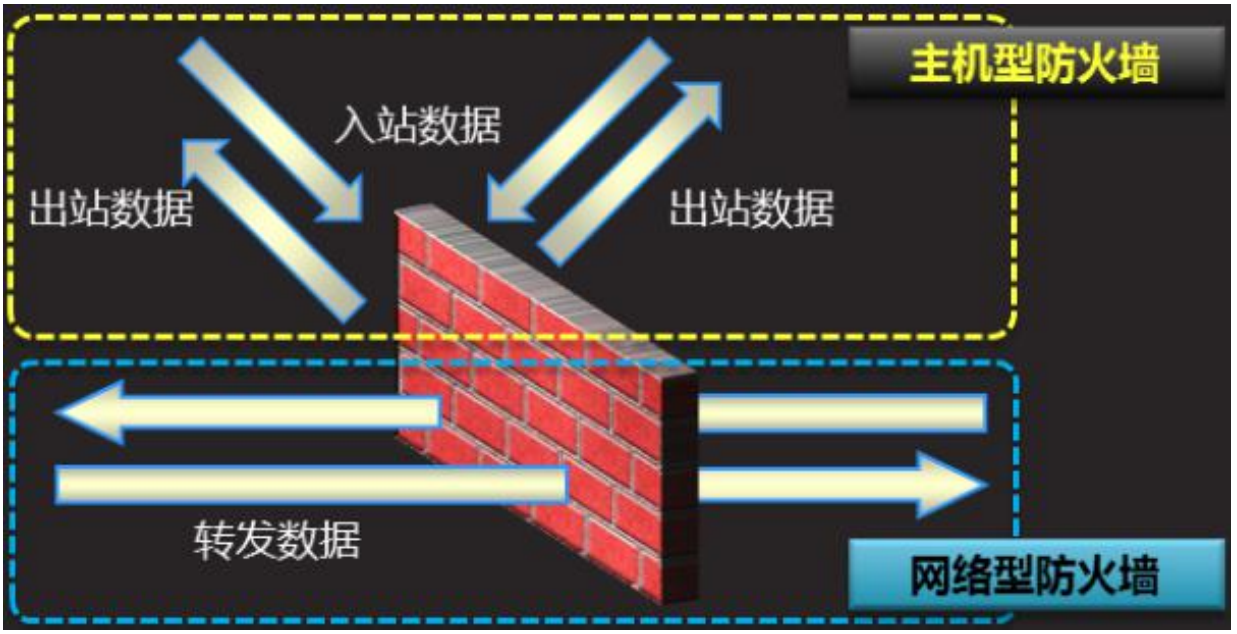
hosta ~]# ping 192.168.4.51   #无回应

hostb ~]# ping 192.168.2.51   #无回应

四 filter 表控制-防护类型及条件

4.1 主机/网络型防护

根据保护对象(本机\其他主机)区分



4.2 开启内核的 IP 转发

作为网关\路由的必要条件: 开启内核的 IP 转发

```
echo 1 > /proc/sys/net/ipv4/ip_forward                    #临时配置
echo “net.ipv4.ip_forward=1” >> /etc/sysctl.conf  #永久配置
```

Linux 网关:192.168.4.51/2.51

内网:192.168.2.52　　外网:192.168.4.50

案例:网络防火墙-FORWARD 功能设置演示

给 2.52 主机设置网关为 192.168.2.51

```
hostb ~]# route add default gw 192.168.2.51
hostb ~]# route -n
```

给 4.50 主机设置网关为 192.168.4.51

```
hosta ~]# route add default gw 192.168.4.51
hosta ~]# route -n
```

给 4.51/2.51 开启内核的 IP 转发(克隆主机默认开启了)

```
hostb ~]# sysctl -a | grep forward   #sysctl -a 查看所有内核配置
net.ipv4.ip_forward = 1
hostb ~]# echo 1 > /proc/sys/net/ipv4/ip_forward  #临时配置
hostb ~]# echo “net.ipv4.ip_forward=1” >> /etc/sysctl.conf
```

关闭 4.51/2.51 上的 iptables 服务, 用 4.50ping 2.52,在 4.51/2.51 上抓包

```
hosta ~]# ping 192.168.2.52
hostc ~]# systemctl stop iptables
hostc ~]# tcpdump -i eth0 -c 2 icmp
IP 192.168.4.50 > 192.168.2.52: ICMP echo request
IP 192.168.2.52 > 192.168.4.50: ICMP echo reply
```

删除 4.51/2.51 上的主机型防火墙的规则,并重启 iptables 服务

```
hostc ~]# iptables -t filter -P INPUT ACCEPT  #修改默认策略为允许
hostc ~]# iptables -t filter -F INPUT
hostc ~]# iptables-save > /etc/sysconfig/iptables
hostc ~]# systemctl stop iptables
hostc ~]# systemctl start iptables
```

```
hostc ~]# iptables -t filter -nL --line-numbers
```

#查看规则,无规则,允许所有包通过

外网主机 4.50 安装 http 服务,创建测试页面,启动 httpd 服务

```
hosta ~]# yum -y install httpd
hosta ~]# echo "i’m host 192.168.4.50" > /var/www/html/test.html
hosta ~]# systemctl restart httpd
```

内网主机 2.52 测试访问 4.50 的 http 服务

```
hostb ~]# curl http://192.168.4.50/test.html
i’m host 192.168.4.50
```

在网络防火墙主机 filter 表 FORWARD 链设置规则,丢掉所有转发包,并测试

```
hostc ~]# iptables -t filter -P FORWARD DROP
hostc ~]# iptables -t filter -nL
hostb ~]# curl http://192.168.4.50/test.html #测试包被丢掉,无回应
```

在网络防火墙主机 filter 表 FORWARD 链设置规则,允许(tcp 80)的请求和回应通过,并测试

```
hostc ~]# iptables -t filter -A FORWARD -p tcp --dport 80 -j ACCEPT
hostc ~]# iptables -t filter -A FORWARD -p tcp --sport 80 -j ACCEPT
hostb ~]# curl http://192.168.4.50/test.html    #测试成功
i am host 192.168.4.50
```

在网络防火墙主机 filter 表 FORWARD 链设置规则,允许(tcp 22)的请求和回应通

过,并测试

```
hostc ~]# iptables -t filter -A FORWARD -p tcp --dport 22 -j ACCEPT

hostc ~]# iptables -t filter -A FORWARD -p tcp --sport 22 -j ACCEPT

hostc ~]# iptables -t filter -nL --line-numbers  #查看规则,有 4 条

Chain FORWARD (policy DROP)

num target  prot opt source    destination

1    ACCEPT tcp  --   0.0.0.0/0  0.0.0.0/0     tcp dpt:80

2    ACCEPT tcp  --   0.0.0.0/0  0.0.0.0/0     tcp spt:80

3    ACCEPT tcp  --   0.0.0.0/0  0.0.0.0/0     tcp dpt:22

4    ACCEPT tcp  --   0.0.0.0/0  0.0.0.0/0     tcp spt:22

hosta ~]# ssh -X root@192.168.2.52    #测试,能 ssh 连接到 2.52
```

六 扩展匹配-扩展条件概述

6.1 扩展条件的方法

前提条件:有对应的防火墙模块支持

基本用法: -m 扩展模块 --扩展条件 条件值

6.2 常见的扩展条件类型

类别	选项	用法
扩展匹配	MAC地址匹配	-m mac --mac-source MAC地址
	多端口匹配	-m multiport --sports 源端口列表 -m multiport --dports 目标端口列表
	IP范围匹配	-m iprange --src-range IP1-IP2 -m iprange --dst-range IP1-IP2

允许 4.50 和 2.52 互 ping

```
hostc ~]# iptables -t filter -A FORWARD -p icmp -j ACCEPT
hostc ~]# iptables -t filter -nL --line-numbers
Chain FORWARD (policy DROP)
num  target    prot opt source              destination
5    ACCEPT    icmp --  0.0.0.0/0           0.0.0.0/0
```

不允许 4.50-4.60 的 ping 包通过,用基本匹配需要写 10 条规则,匹配速度慢

ping 包中的源地址为 4.50-4.60

根据匹配即停止原则,该条规则要写在第 5 条规则之前

```
hostc ~]# iptables -t filter -I FORWARD 5 -p icmp \
-m iprange --src-range 192.168.4.50-192.168.4.60 -j DROP
hosta ~]# ping 192.168.4.52    #不能 ping 通
hostb ~]# ping 192.168.4.50    #不能 ping 通,回应包被 DROP 了
```

七 扩展匹配-扩展案例

7.1 根据 MAC 地址封锁主机,禁止 ping 其他主机

获取 hosta 192.168.4.50 的 MAC

```
hosta ~]# ifconfig eth0
ether 52:54:00:fd:81:ad
```

网络防火墙主机上在 filter 表 FORWARD 链中添加规则,禁止 MAC 地址为

52:54:00:fd:81:ad 的 ping 包通过

```
hostc ~]# iptables -t filter -I FORWARD 5 -p icmp -m mac \
--mac-source 52:54:00:fd:81:ad -j DROP
hostc ~]# iptables -t filter -D FORWDRD 6
hosta ~]# ping 192.168.2.52    #不能 ping 通
hostb ~]# ping 192.168.4.50    #不能 ping 通,回应包被 DROP 了
```

7.2 多端口案例

```
[root@hostc ~]# iptables -t filter -nL --line-numbers
Chain FORWARD (policy DROP)
num  target    prot opt source              destination
1 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:80
2 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp spt:80
3 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:22
4 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp spt:22
5 DROP icmp -- 0.0.0.0/0 0.0.0.0/0 MAC 52:54:00:FD:81:AD
6 ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
```

将以上规则的 22 端口和 80 端口合并,并添加 3306 端口

在网络防火墙主机 filter 表 FORWARD 链设置规则,放行 tcp 22 80 3306 请求包

```
hostc ~]# iptables -t filter -A FORWARD -p tcp -m multiport \
```

```
--dports 22,80,3306 -j ACCEPT
```

在网络防火墙主机 **filter** 表 **FORWARD** 链设置规则,放行 **tcp 22 80 3306** 回应包

```
hostc ~]# iptables -t filter -A FORWARD -p tcp -m multiport \
--sports 22,80,3306 -j ACCEPT
```

删除原来的 **tcp 22 80** 的规则

```
hostc ~]# iptables -t filter -D FORWARD 1  #执行 4 次
```

网络防火墙主机查看 **filter** 表 **FORWARD** 链规则

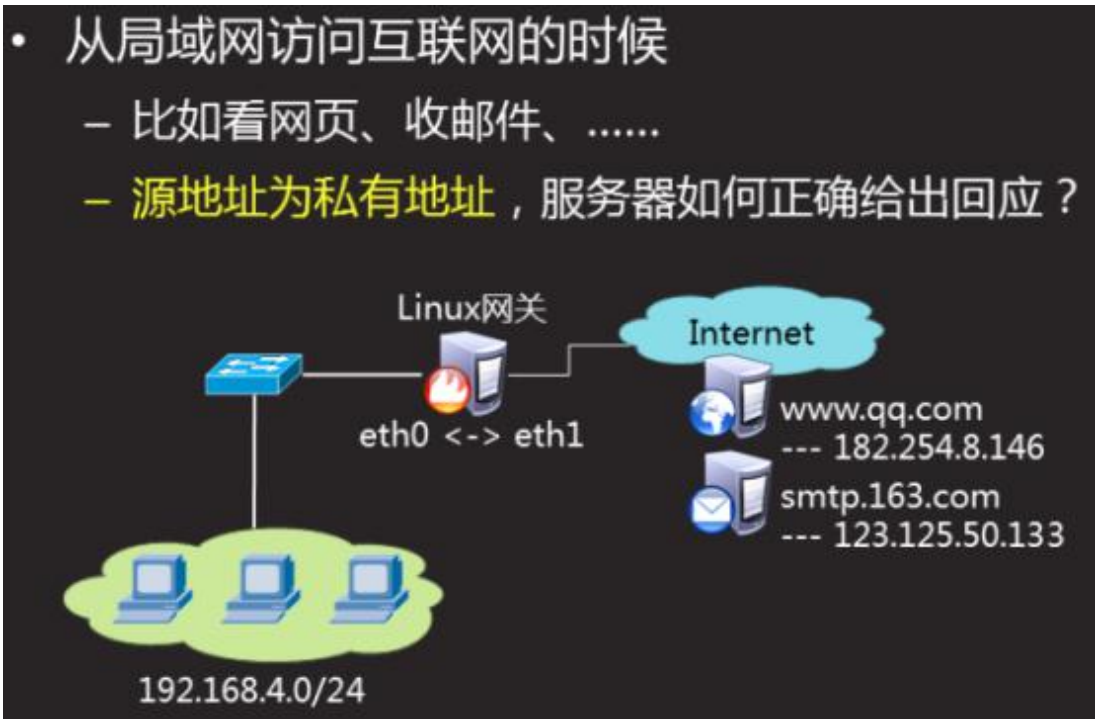
```
hostc ~]# iptables -t filter -nL --line-numbers
Chain FORWARD (policy DROP)
num  target    prot opt source              destination
1 DROP icmp -- 0.0.0.0/0 0.0.0.0/0          MAC 52:54:00:FD:81:AD
2 ACCEPT icmp -- 0.0.0.0/0 0.0.0.0/0
3 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 multiport dports 22,80,3306
4 ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 multiport sports 22,80,3306
```

测试

```
hosta ~]# ssh -X root@192.168.2.52   #成功
hostb ~]# ssh -X root@192.168.4.50   #成功
hostb ~]# curl http://192.168.4.50/test.html   #成功
i am host 192.168.4.50
```

八 NAT 表典型应用-NAT 转化原理

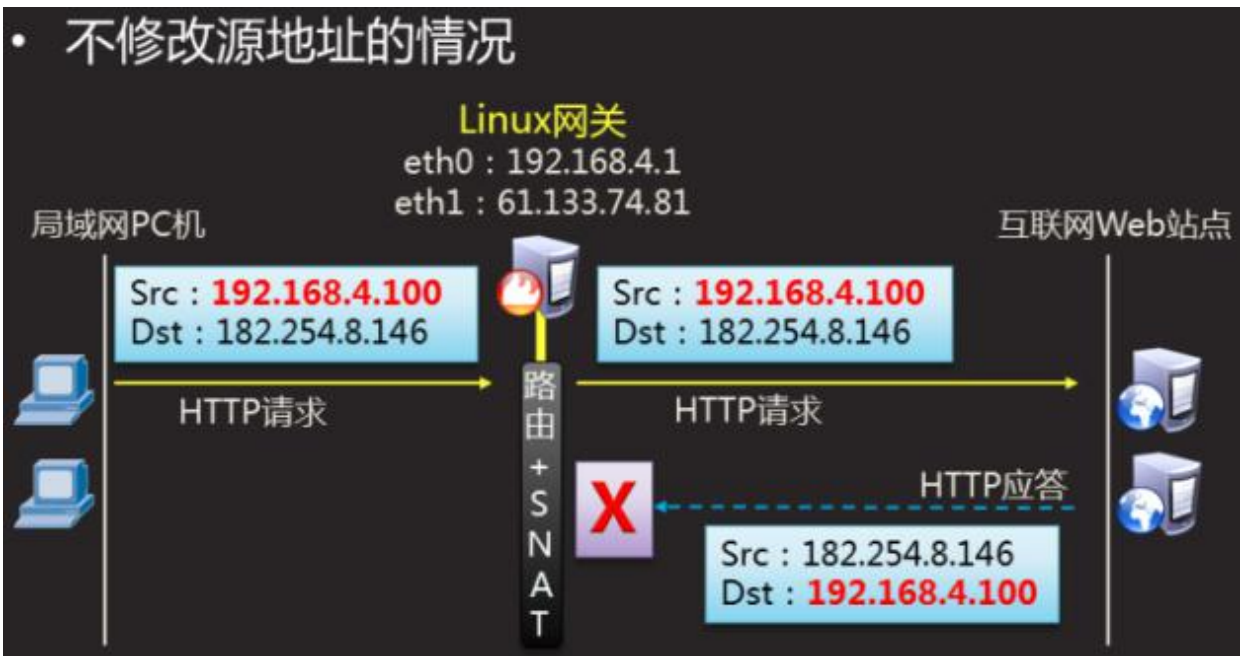
8.1 私有地址的局限性

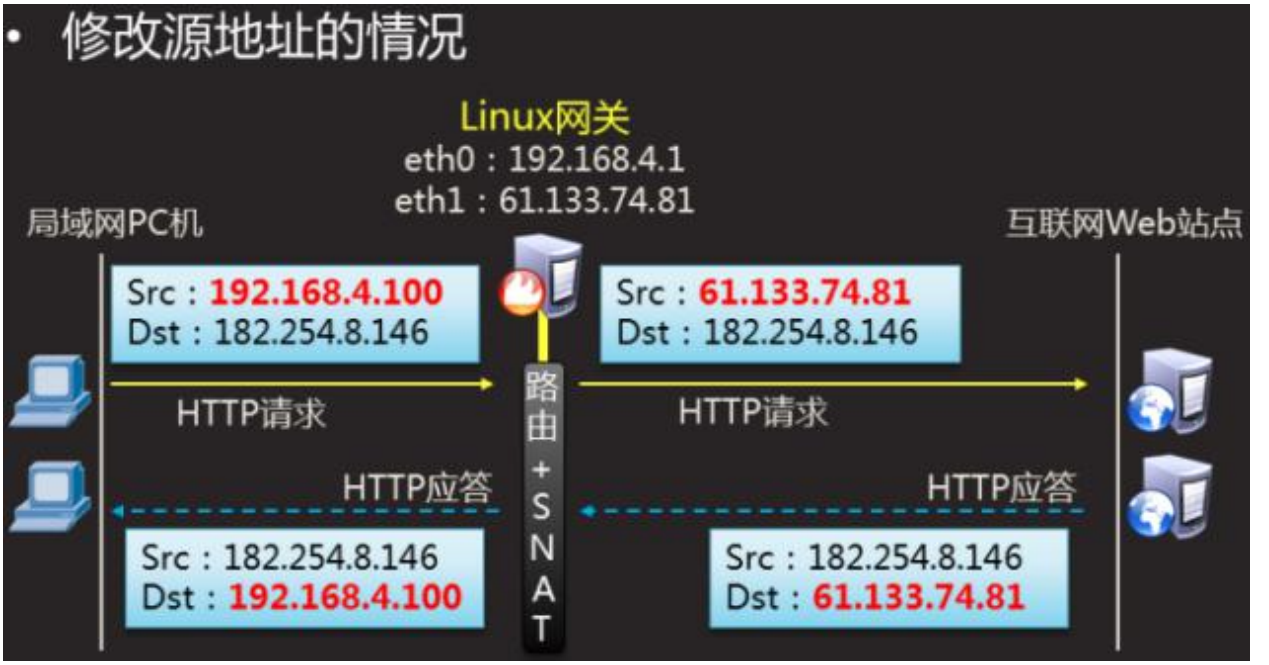


8.2 SNAT 源地址转换

Source Network Address Translation

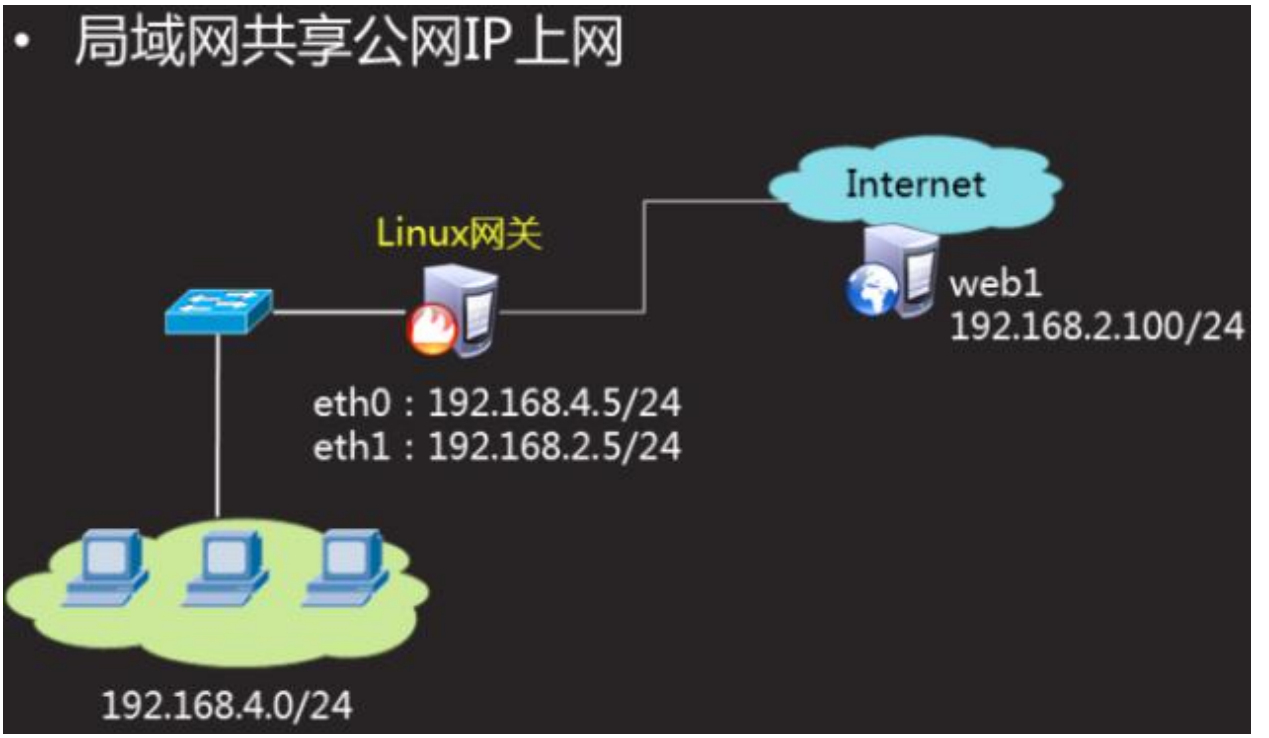
修改数据包的源地址,仅用于 nat 表的 POSTROUTING 链





九 NAT 表典型应用-SNAT 策略应用

9.1 案例环境



9.2 配置 SNAT 共享上网

9.2.1 删除 4.50 2.52 的网关,停止 4.50 2.52 的 iptables 服务

```
hosta ~]# route del default gw 192.168.4.51
```

```
hostb ~]# route del default gw 192.168.2.51
```

9.2.2 2.52 上安装 **http,**创建测试页面,启动 **httpd** 服务

```
hostb ~]# yum -y install httpd
hostb ~]# echo "i'm host 192.168.2.52" > /var/www/html/test.html
hostb ~]# systemctl restart httpd
```

9.2.3 网络防火墙主机上清除所有工作并保存

```
hostc ~]# iptables -t filter -F
hostc ~]# iptables -t raw -F
hostc ~]# iptables -t nat -F
hostc ~]# iptables -t mangle -F
hostc ~]# iptables-save > /etc/sysconfig/iptables
```

9.2.4 4.50 上设置网关,并查看;尝试访问 **2.52** 的测试页面

```
hosta ~]# route add default gw 192.168.4.51
hosta ~]# route -n
hosta ~]# curl http://192.168.2.52/test.html    #无法连接,超时
```

9.2.5 抓包验证

```
hosta ~]# curl http://192.168.2.52/test.html
hostc ~]# tcpdump -i eth0 -t tcp port 80
IP 192.168.4.50.51174 > 192.168.2.52
```

9.2.6 nat 表 **POSTROUTING** 链书写规则,将 **192.168.4.0/24** 转换为

192.168.2.51

```
hostc ~]# iptables -t nat -nL
hostc ~]# iptables -t nat -A POSTROUTING -s 192.168.4.0/24 -p tcp
--dport 80 -j SNAT --to-source 192.168.2.51
hostc ~]# iptables -t nat -nL
```

9.2.7 连接验证

```
hosta ~]# curl http://192.168.2.52/test.html
```

9.2.8 查看 **2.52** 网页访问日志

```
hostb ~]# tail -f /etc/httpd/logs/access_log
```

9.3 地址伪装策略

```
hostc ~]# iptables -t nat -A POSTROUTING -s 192.168.4.0/24 -o eth1
-j MASQUERADE
```