**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**

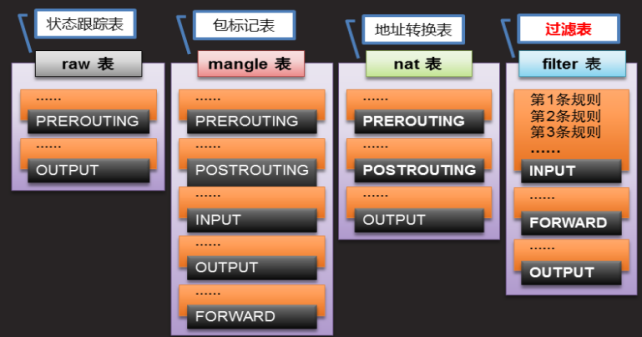
一道保护性的安全屏障;起保护\隔离公网和私网的作用的一种软件服务

**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**



**2.4 常用匹配条件**



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](mailto: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

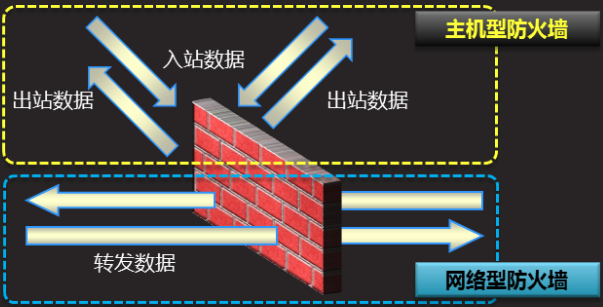
hosta ~]# ping 192.168.4.51 #无回应

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](mailto: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



**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](mailto:root@192.168.2.52) #成功

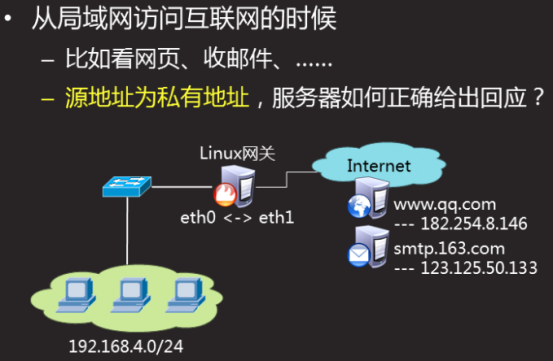
hostb ~]# ssh -X [root@192.168.4.50](mailto:root@192.168.4.50) #成功

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**

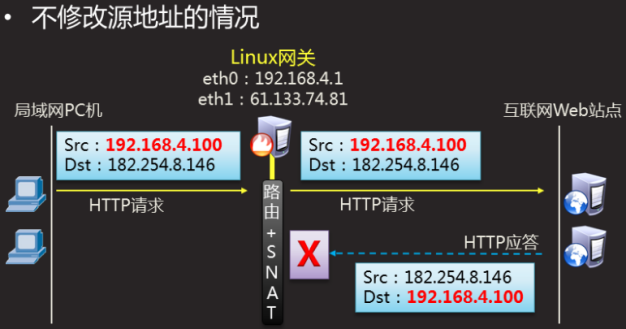
**8.1 私有地址的局限性**

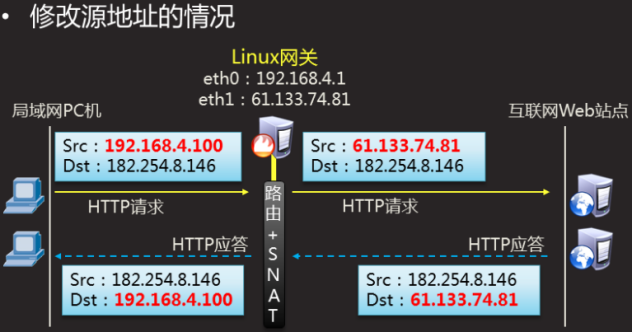


**8.2 SNAT源地址转换**

**S**ource **N**etwork **A**ddress **T**ranslation

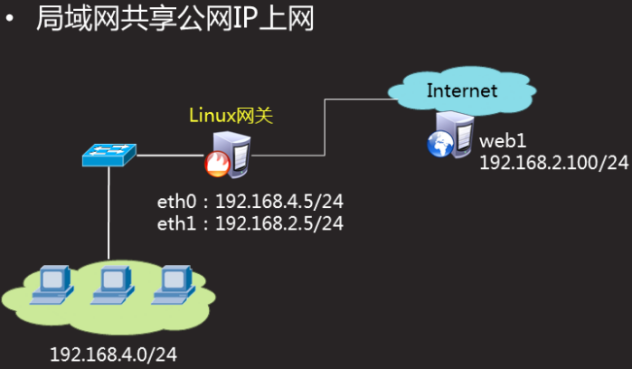
修改数据包的源地址,仅用于nat表的POSTROUTING链





**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