# 1.tcp\_wrappers

在第四层(传输层)的安全工具通过 /etc/hosts.allow 和/etc/hosts.deny文件管控:

只有由对supper daemon(xinetd)管理的服务或者含有libwrap.so模块的服务起作用

[root@www ~]# ldd `which sshd` | grep libwrap.so

libwrap.so.0 => /lib64/libwrap.so.0 (0x00007f4dc946f000)

root@www xinetd.d]# ldd `which xinetd` | grep libwrap.so

libwrap.so.0 => /lib64/libwrap.so.0 (0x00007fa4362dd000)

规则:

先以/etc/hosts.allow为优先对比,符合条件就放行

再以/etc/hosts.deny对比,符合条件就抵挡

如果不在这两个文件中,则放行

注意写法:

程序名:ip

程序名:网段/子网掩码

ALL:xxxxxxx

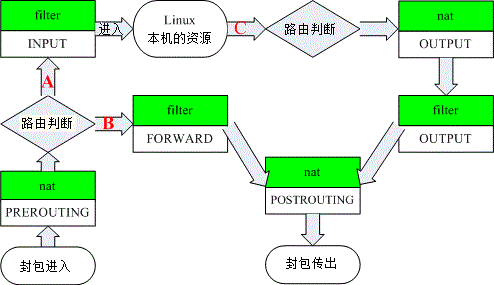
程序名:ALL

例如:/etc/hosts.deny

sshd:192.168.200.146

# 2.iptables

## 2.1简介



filter(过滤器):主要跟进入Linux本机的数据包有关

INPUT:主要与想进入到本机的数据包有关

OUPUT:主要与主机所要送出的数据包有关

FORWARD:和主机本身没关系,可以传递包到后端机器中

nat(地址转换):进行来源和目的ip或port的转换和本机无关,和后端机器相关

PREROUTING:在进行路由判断之前所要进行的规则

POSTROUTING:在进行路由判断之后所要进行的规则

OUTPUT:与发出去的包有关

## 2.2命令

查看格式化后的规则:

[root@www ~]# **iptables [-t tables] [-L] [-nv]**

选项与参数：

-t ：后面接 table ，例如 nat 或 filter ，若省略此项目，则使用默认的 filter

-L ：列出目前的 table 的规则

-n ：不进行 IP 与 HOSTNAME 的反查，显示讯息的速度会快很多！

-v ：列出更多的信息，包括通过该规则的封包总位数、相关的网络接口等

iptables -L -n

iptables -t nat -L -n

查看完整的非格式化的规则:

iptables-save

清楚规则:

[root@www ~]# **iptables [-t tables] [-FXZ]**

选项与参数：

-F ：清除所有的已订定的规则；

-X ：杀掉所有使用者 "自定义" 的 chain (应该说的是 tables ）啰；

-Z ：将所有的 chain 的计数与流量统计都归零

范例：清除本机防火墙 (filter) 的所有规则

配置规则:

root@www ~]# **iptables [-AI 链名] [-io 网络接口] [-p 协议] \**

> **[-s 来源IP/网域] [-d 目标IP/网域] -j [ACCEPT|DROP|REJECT|LOG]**

选项与参数：

-AI 链名：针对某的链进行规则的 "插入" 或 "累加"

-A ：新增加一条规则，该规则增加在原本规则的最后面。例如原本已经有四条规则，

使用 -A 就可以加上第五条规则！

-I ：插入一条规则。如果没有指定此规则的顺序，默认是插入变成第一条规则。

例如原本有四条规则，使用 -I 则该规则变成第一条，而原本四条变成 2~5 号

链 ：有 INPUT, OUTPUT, FORWARD 等，此链名称又与 -io 有关，请看底下。

-io 网络接口：设定封包进出的接口规范

-i ：封包所进入的那个网络接口，例如 eth0, lo 等接口。需与 INPUT 链配合；

-o ：封包所传出的那个网络接口，需与 OUTPUT 链配合；

-p 协定：设定此规则适用于哪种封包格式

主要的封包格式有： tcp, udp, icmp 及 all 。

-s 来源 IP/网域：设定此规则之封包的来源项目，可指定单纯的 IP 或包括网域，例如：

IP ：192.168.0.100

网域：192.168.0.0/24, 192.168.0.0/255.255.255.0 均可。

若规范为『不许』时，则加上 ! 即可，例如：

-s ! 192.168.100.0/24 表示不许 192.168.100.0/24 之封包来源；

-d 目标 IP/网域：同 -s ，只不过这里指的是目标的 IP 或网域。

-j ：后面接动作，主要的动作有接受(ACCEPT)、丢弃(DROP)、拒绝(REJECT)及记录(LOG)

## 2.3配置规则

|  |
| --- |
| #清楚默认规则  iptables -F  iptables -X  iptables -Z  iptables -t nat -F  iptables -t nat -X  iptables -t nat -Z  #filter表的默认规则  iptables -P INPUT DROP  iptables -P OUTPUT ACCEPT  iptables -P FORWARD DROP  #INPUT链的基本规则  iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT  iptables -A INPUT -m state --state INVALID -j ACCEPT  iptables -A INPUT -i lo -j ACCEPT  #FORWARD基本规则  iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT  iptables -A FORWARD -m state --state INVALID -j ACCEPT  #能被ping  iptables -A INPUT -p icmp -j ACCEPT  #sshd  iptables -A INPUT -p tcp --dport 22 -j ACCEPT  iptables -A INPUT -p udp --dport 22 -j ACCEPT  #nginx  iptables -A INPUT -p tcp --dport 80 -j ACCEPT  iptables -A INPUT -p udp --dport 80 -j ACCEPT  #mongodb  iptables -A INPUT -p tcp --dport 27017 -j ACCEPT  iptables -A INPUT -p udp --dport 27017 -j ACCEPT  iptables -A INPUT -p tcp --dport 28017 -j ACCEPT  iptables -A INPUT -p udp --dport 28017 -j ACCEPT  #mysql  iptables -A INPUT -p tcp --dport 3306 -j ACCEPT  #nginx  iptables -A INPUT -p tcp --dport 8000 -j ACCEPT  iptables -A INPUT -p udp --dport 8000 -j ACCEPT  #/usr/libexec/iptables/iptables.init save  service iptables save  #systemctl restart iptables |

注意:-p 和 –dport要一起使用,单独使用要报错

## 2.4端口重定向

iptables -t nat -A PREROUTING -p tcp --dport 8888 -j REDIRECT --to-ports 80

## 2.5端口转发

**此实验可能有问题**

iptables所在的服务器

eth0:192.168.100.10

eth1:192.168.200.158

web1:

eth0:192.168.100.254

web2:

eth0:192.168.100.253

客户端:

192.168.200.156

web1中httpd.conf里Listen:80

web2中httpd.conf里Listen:8888

目的:客户端访问192.168.100.10:80 端得到192.168.100.254的web界面,访问192.168.100.10:8888端得到的是192.168.100.253:8888的web界面,从而实现

端口转发功能

#清楚默认规则

iptables -F

iptables -X

iptables -Z

iptables -t nat -F

iptables -t nat -X

iptables -t nat -Z

#默认操作

iptables -P INPUT DROP

iptables -P OUTPUT ACCEPT

iptables -P FORWARD DROP

#INPUT基本规则

iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A INPUT -m state --state INVALID -j DROP

#FORWARD基本规则

iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -m state --state INVALID -j DROP

#放行所有通过lo的数据包

iptables -A INPUT -i lo -j ACCEPT

#icmp

iptables -A INPUT -p icmp --icmp-type any -s 192.168.200.0/24 -j ACCEPT

iptables -A INPUT -p icmp --icmp-type any -s 192.168.100/24 -j ACCEPT

#sshd

iptables -A INPUT -s 192.168.200.0/24 -i eth1 -p tcp --dport 22 -j ACCEPT

iptables -A INPUT -s 192.168.200.0/24 -i eth1 -p udp --dport 22 -j ACCEPT

#修改数据包目的地址

iptables -t nat -A PREROUTING -p tcp -d 192.168.200.158 --dport 80 -j DNAT --to 192.168.100.254:80

iptables -t nat -A PREROUTING -p tcp -d 192.168.200.158 --dport 8888 -j DNAT --to 192.168.100.253:8888

#修改数据包来源地址

iptables -t nat -A POSTROUTING -p tcp -d 192.168.100.254 --dport 80 -j SNAT --to 192.168.100.10

iptables -t nat -A POSTROUTING -p tcp -d 192.168.100.253 --dport 8888 -j SNAT --to 192.168.100.10

#FORWARD设置

iptables -A FORWARD -p tcp -d 192.168.100.254 --dport 80 -j ACCEPT

iptables -A FORWARD -p tcp -d 192.168.100.253 --dport 8888 -j ACCEPT

/etc/sysctl.conf里:

net.ipv4.ip\_forward = 1;

这样才能转发

分析:

1)当客户端192.168.200.156访问192.168.200.158:80时,首先客户端发的数据包会经过PREROUTING链,根据上面iptables配置可知，目的地址被改为192.168.100.254:80

再经过路由判断数据包会经过上图B,然后经过FORWARD链,有iptables配置知此数据包会被放行,然后经过POSTROUTING链,有iptables配置可知,来源被修改为192.168.100.10

然后数据包通过iptables所在的服务器转发到了web1

2)web1通过分析客户端发来的数据包,可知web1发送的数据包目的地址为192.168.100.10,首先此数据包会经过PREROUTING链,此时Iptables配置里没有对来自

192.168.100.254:80的规则,则放行,经过路由判断,目的地址为192.168.100.10正好是iptables所在的机器此时经过端口好等一系列的映射就能返回到客户端手中(具体过程不清楚，web返回的数据只要知道目的地址192.168.100.10就能返回)

## 2.6nat服务器

注意:

/etc/sysctl.conf:

net.ipv4.ip\_forward = 1

原理:修改来源Ip地址

**环境:**

客户端:

192.168.1.1

nat服务器:

192.168.200.239(通过此ip可以访问外网)

192.168.1.10

|  |
| --- |
| iptables -F  iptables -X  iptables -Z  iptables -t nat -F  iptables -t nat -X  iptables -t nat -Z  iptables -P INPUT DROP  iptables -P OUTPUT ACCEPT  iptables -P FORWARD DROP  #INPUT链的基本规则  iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT  iptables -A INPUT -m state --state INVALID -j ACCEPT  iptables -A INPUT -i lo -j ACCEPT  #FORWARD基本规则  iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT  iptables -A FORWARD -m state --state INVALID -j ACCEPT  #能被ping  iptables -A INPUT -p icmp -j ACCEPT  #sshd  iptables -A INPUT -p tcp --dport 22 -j ACCEPT  iptables -A INPUT -p udp --dport 22 -j ACCEPT  iptables -A INPUT -p tcp --dport 80 -j ACCEPT  #访问外网  iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT --to 192.168.200.239  #iptables -A FORWARD -j ACCEPT  iptables -A FORWARD -s 192.168.1.0/24 -j ACCEPT |

## 2.7内网开启sshd

修改目的ip地址

和2.6环境一样,目的:使200网段的能ssh到192.168.1.1

|  |
| --- |
| iptables -F  iptables -X  iptables -Z  iptables -t nat -F  iptables -t nat -X  iptables -t nat -Z  iptables -P INPUT DROP  iptables -P OUTPUT ACCEPT  iptables -P FORWARD DROP  #INPUT链的基本规则  iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT  iptables -A INPUT -m state --state INVALID -j ACCEPT  iptables -A INPUT -i lo -j ACCEPT  #FORWARD基本规则  iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT  iptables -A FORWARD -m state --state INVALID -j ACCEPT  #能被ping  iptables -A INPUT -p icmp -j ACCEPT  #sshd  iptables -A INPUT -p tcp --dport 22 -j ACCEPT  iptables -A INPUT -p udp --dport 22 -j ACCEPT  iptables -A INPUT -p tcp --dport 80 -j ACCEPT  #内网开启sshd服务  iptables -t nat -A PREROUTING -d 192.168.200.239 -j DNAT --to 192.168.1.1  #iptables -A FORWARD -j ACCEPT  iptables -A FORWARD -s 192.168.200.0/24 -j ACCEPT |