Linux防火墙

# Linux防火墙的概述

防火墙是连接内部和外部网络的一组硬性装置。可以对内外网进行访问控制，并且抵御网络攻击和提供网络地址翻译。防火墙分为包过滤防火墙、应用级网关、和代理服务型防火墙。

包过滤防火墙：在网络层对数据包进行选择，选择的依据是系统设置的访问控制表。例如iptables

应用级防火墙：在应用层建立协议过滤和转发功能。例如TCP\_wrappers

# Iptables简介

Netfilter/Iptables是集成在2.4.X版本以上内核linux上的包过滤防火墙系统。它包括Netfilter和Iptables两部分，Netfilter是包过滤器，称为内核空间。负责存储规则和包过滤匹配。Iptables是用户空间，是Netfilter的外部匹配工具，用户通过Iptables对包过滤进行增、删、改等操作。

## Netfilter框架

在NetFilter框架中，提供了filter、nat和mangle三个表，系统默认使用的是filter。每个表中包含若干个链（chains）,每个链中包含若干个规则。Iptables网络限制的策略由规则、链和表构成，表是链的容器，链是规则的容器。Netfilter/Iptables的结构：

filter------>LOCAL\_IN,LOCAL\_OUT,FORWARD------->rules

NAT--->PRE\_ROUTING,LOCAL\_IN,LOCAL\_OUT,POST\_ROUTING-->rules

mangle---->ALL HOOKS------->rules

raw

***表***：是Iptables内建块，它描述其功能的大类。

* filter表用于应用防火墙过滤规则；
* nat表包含源和目的地址及端口转换使用的规则；
* mangle表用于修改分组数据的特定规则(标记和修改包)。

***链***：链是数据包传输路径。

* INPUT链：当一个数据包由内核中的路由计算确定为本地的Linux系统后，它会通过INPUT链的检查。
* OUTPUT链：保留给系统自身生成的数据包。
* FORWARD链：经过Linux系统路由的数据包（即当iptables防火墙用于连接两个网络时，两个网络之间的数据包必须流经该防火墙）。
* PREROUTING链：用于修改目标地址（DNAT）。
* POSTROUTING链：用于修改源地址（SNAT）。

***规则***：判断一个包的特性及匹配时所进行的操作。

* ACCEPT：接收数据包。
* DROP：丢弃数据包。
* REDIRECT：将数据包重定向到本机或另一台主机的某个端口，通常能实现透明代理或对外开放内网的某些服务。
* REJECT：拦截该数据封包，并发回封包通知对方。
* SNAT：源地址转换，即改变数据包的源地址。例如，将局域网的IP（10.0.0.1/24）转换为广域网的IP（203.93.236.141/24），在NAT表的POSTROUTING链上进行该动作。
* DNAT：目的地址转换，即改变数据包的目的地址。例如，将广域网的

IP（203.93.236.141/24）转换为局域网的IP（10.0.0.1/24），在NAT表的PREROUTING链上进行该动作。

* MASQUERADE：IP伪装，即常说的NAT技术，MASQUERADE只能用于ADSL等拨号上网的IP伪装，也就是主机的IP是由ISP分配动态的，如果主机的IP地址是静态固定的，就要使用SNAT。
* LOG：日志功能，将符合规则的数据包相关信息记录在日志中，以便管理员分析和排错。

## Iptables工作原理

OUT

IN

POSTROUTING

contrack

nat-MASQUERADE

nat-SNAT

PREROUTING

contrack

mangle

nat-ONAT

FORWARD

filter

OUTPUT

Filter

Mangle

conntrack

INPUT

Filter

conntrack

localhost

# Iptables的基础知识

## Iptables的状态

Iptables支持状态过滤，常见的状态解释如下：

* NEW：如果你的主机向远程主机发出一个连接请求，这个数据包的状态是NEW。
* ESTABLISHED：在连接建立之后（完成TCP的三次握手后），远程主机和你的主机通信数据的状态为ESTABLISHED。
* RELATED：和现有联机相关的新联机封包。像FTP这样的服务，用端口21传送命令，而用端口20（port模式）或其他端口（PASV模式）传送数据。在已有的端口21上建立好连接后发送命令，用20或其他端口传送数据（FTP-DATA），其状态是RELATED。
* INVALID：无效的数据包，不能被识别属于哪个连接或没有任何状态，通常这种状态的数据包会被丢弃。

|  |
| --- |
| #这个规则将丢弃所有发送到你机器的数据包（状态是NEW的包）。也就是不允许其他的机器主动发起对你机器的连接，但是你却可以主动地连接其他的机器，不过仅仅是连接而已。  iptables -A INPUT-m state --state NEW -j DROP  #允许所有已经建立连接或者与之相关的数据通过  iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT |

## Iptables的Conntrack记录

/proc/net/ip\_conntrack里的conntrack记录。这些记录表示的是

当前被跟踪的连接。如果安装了ip\_conntrack模块，我们可以查看ip\_conntrack记录，命令：cat /proc/net/ip\_conntrack

|  |
| --- |
| tcp　　6 117 SYN\_SENT src=192.168.1.6 dst=192.168.1.9 sport=32775　dport=22 ［UNREPLIEDtcp　　6 431967 ESTABLISHED src=192.168.18.120 dst=192.168.18.98 sport=1059 dport=packets=3287 bytes=262840 src=192.168.18.98 dst=192.168.18.120 sport=22 dport=1059  packets=2863 bytes=439784 ［ASSURED］ mark=0 secmark=0 use= |

首先显示的是协议，这里是TCP，接着是十进制的6（TCP的协议类型

代码是6）。之后的117是这条conntrack记录的生存时间，它会有规律地被消耗，直到收到这个连接更多的包。那时，这个值就会被设为当时那个状态的默认值。接下来的是这个连接在当前时间点的状态。上面的例子说明这个包处在状态SYN\_SENT，这个值是iptables显示的，以便我们理解，而内部用的值稍有不同。SYN\_SENT说明我们正在观察的这个连接只在一个方向发送了一个TCP SYN包。再下面是源地址、目的地址、源端口和目的端口。其中有个特殊的词UNREPLIED，说明这个连接还没有收到任何回应。最后，是希望接收的应答包的信息，它们的地址和端口和前面是相反的。

连接跟踪记录的信息依据IP所包含的协议不同而不同，所有相应的值都是在头文件linux/include/netfilter-ipv4/ip\_conntrack\*.h中定义的。IP、TCP、UDP、ICMP协议的默认值是在linux/include/netfilter-ipv4/ip\_conntrack.h里定义的。具体的值可以查看相应的协议，但我们这里用不到它们，因为它们大都只在conntrack内部使用。随着状态的改变，生存时间也会改变。当一个连接在两个方向上都有传输时，conntrack记录就删除［UNREPLIED］标志，然后重置。在末尾有［ASSURED］的记录说明两个方向已没有流量。这样的记录是确定的，在连接跟踪表满时，是不会被删除的，没有［ASSURED］的记录就要被删除。连接跟踪表能容纳多少记录是被一个变量控制的，它可由内核中的ip-sysctl函数设置。默认值取决于你的内存大小，128MB可以包含8192条目录，256MB对应16376条，如果大家在生产服务器上通过加载模块的方法开启了ip\_conntract功能，就要注意内存方面的使用情况，此模块是极消耗内存的，对系统效率非常有影响，所以我这里不建议在生产服务器上开启conntrack功能。也可以在/proc/sys/net/ipv4/ip\_conntrack\_max里查看和设置。

## Iptables初始化的注意事项

在跟一些系统管理员进行交流时，我发现大家经常遇到的一个问题就是：误操作iptables而将自己也拦截在机器之外了，没办法只有去机房重启iptables。其实这个问题是有办法解决的，我们可以配置一计划任务Crontab，每5分钟运行一次，即\* \*/5 \* \* \* /etc/init.d/iptables stop；，这样即使你的脚本存在错误设置（或丢失的）规则时，也不至于将你锁在计算机外而无法返回与计算机的连接，这样你就可以放心大胆地调试脚本了。鉴于许多同事在学习及

调试iptables脚本时也是用自己内部的机房，所以推荐用此方法，免得受进出机房之苦。

# Iptables使用方法

## 语法格式

Iptables [-t 表名] –命令 -匹配 -j 动作/目标

-t：用来指定操作的表，其中默认的是filter表。

命令：指定进行的规则的操作命令

匹配：指定信息包与规则匹配所有的特征（源地址，目的地址等）

-j：指定对匹配的包采取的动作

编写步骤：指定表，指定链，对匹配的包进行某种操作。

## 相关选项

### 表名选项

filter：用于信息包过滤。

* input :负责过滤所有目标地址是本机地址的数据包，就是过滤进入主机的数据包;
* forward :负责转发流经主机但不进入本机的数据包，和NAT关系很大;
* output :负责处理源地址的数据包，就是对本机发出的数据包;

nat：用于网络地址转换

* snat :地址转换
* dnat :标地址转换
* pnat :标端口转换

mangle：标记、修改包

### 命令选项

* -A：将规则附加到链末尾。
* -D：指定要匹配的规则或则规则在链中的位置编号，从链中删除该规则
* -P：与链任何规则不匹配的信息都降至使用此链的策略（缺省目标策略）
* -N：创建新链
* -F：指定列名，则删除链中所有规则；未指定列名则删除所有链的所有规则。
* -I：插入一条规则，原本该位置上的规则将会往后顺序移动，如果没有指定规则编号，则在第一条规则前插入。
* -L：列出所有规则

### 匹配选项

* -p：用于检查某些特定的协议，如tcp,udp,icmp
* -s：用来与信息包源地址匹配
* -d：用来与信息包目标地址匹配
* -i：接收数据包的网卡接口
* -o：发送数据包的网卡接口
* (-m tcp) --sport：匹配源端口
* (-m tcp) --dport：匹配目标端口
* -m iprange --src-range：匹配源地址范围
* -m iprange --dst-range：匹配目标地址范围
* -m limit --limit：匹配数据表速率
* -m mac --mac-source：匹配源MAC地址
* -m multiport --sports：匹配源端口段
* -m multiport --dports：匹配目标端口段
* -m state --state ：(匹配状态INVALID,ESTABLISHED,NEW,RELATED)
* -m srting --string "…" --algo {bm|kmp} ：匹配应用层字串

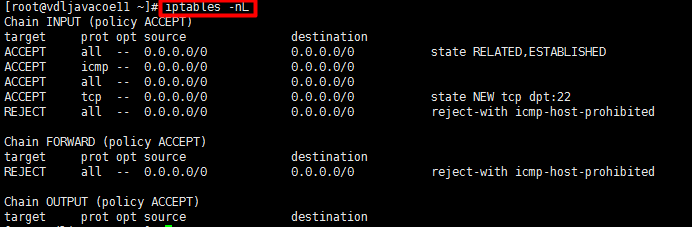
### 目标操作选项

* ACCEPT：当信息包ACCEPT目标规则完全匹配时，会接受并停止遍历链。
* DROP：当信息包DROP目标规则完全匹配时，会阻塞信息包，并不会做处理。
* REJECT：当信息包REJECT目标规则完全匹配时，会将错误信息发送给发送方。
* SNAT：源地址改写
* DNAT：目标地址改写
* LOG：信息记录

## Iptables常用使用

### 查看iptables规则

iptables　［-t　表名］　<-L>　［链名］



### 清除规则和计数器

iptables　［-t 表名］　<-F| Z>

参数说明：

* -F：删除指定表中的所有规则。
* -Z：将指定表中的数据包计数器和流量计数器归零。

|  |
| --- |
| iptables -F  iptables -X  iptables -Z  iptables -F -t nat  iptables -X -t nat #删除链  iptables -Z -t nat  iptables -X -t mangle |

### 定义默认的策略

iptables　［-t 表名］　<-P>　<链名>　<动作>

|  |
| --- |
| //设置FORWARD默认的策略是DROP  iptables -P INPUT DROP  iptables -P FORWARD DROP  iptables -P OUTPUT DROP |

### 增加、插入、删除、替换iptables规则

-A（追加一条规则）-I(插入一条规则)-D（删除一条规则）-R（替换一条规则）

|  |
| --- |
| #开放指定ip或者网段  -A INPUT -s 10.0.0.0/8 -j ACCEPT  #开放指定端口（22）  -A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT  #开放多个端口（22）  -A INPUT -p tcp -m state --state NEW -m multiport --dports 22,80 -j ACCEPT  #开放指定mac地址  -A INPUT -m mac --mac-source 00:E0:4C-29:9E:C6 -j ACCEPT  #限制发送包个数(每秒500个包)  -A INPUT -m limit --limit 500/sec -j ACCEPT |

### Iptables保存和恢复命令

|  |
| --- |
| service iptables save //保存对当规则到/etc/sysconfig/iptables  iptables-save > firewall.bak #防火墙的备份  iptables-restore < firewall.bak #防火墙的还原 |

# CentOS7中使用iptables

CentOS 7.0默认使用的是firewall作为防火墙，这里改为iptables防火墙。

## 关闭firewall

systemctl stop firewalld.service #停止firewall

systemctl disable firewalld.service #禁止firewall开机启动

## 安装iptables防火墙

yum install iptables-services -y #安装

vi /etc/sysconfig/iptables #编辑防火墙配置文件

|  |
| --- |
| # Firewall configuration written by system-config-firewall  # Manual customization of this file is not recommended.  \*filter  :INPUT ACCEPT [0:0]  :FORWARD ACCEPT [0:0]  :OUTPUT ACCEPT [0:0]  -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  -A INPUT -p icmp -j ACCEPT  -A INPUT -i lo -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT  -A INPUT -j REJECT --reject-with icmp-host-prohibited  -A FORWARD -j REJECT --reject-with icmp-host-prohibited  COMMIT |

systemctl restart iptables.service #最后重启防火墙使配置生效

systemctl enable iptables.service #设置防火墙开机启动

systemctl restart/start/stop/status iptables.service#来查看状态

## Iptables参考举例

|  |
| --- |
| modprobe ipt\_MASQUERADE  modprobe ip\_conntrack\_ftp  modprobe ip\_nat\_ftp  iptables -F  iptables -t nat -F  iptatles -X  iptables -t nat -X  #主机input默认规则为drop，允许主机访问外网，开放回环地址  iptables -P INPUT DROP  iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  iptables -A INPUT -i lo -j ACCEPT    #开放常用服务端口  iptables -A INPUT -p tcp -m multiport --dports 110,80,25 -j ACCEPT  iptables -A INPUT -p tcp -s 10.10.0.0/24 --dport 139 -j ACCEPT   #邮件  iptables -A INPUT -i eth1 -p udp -m multiport --dports 53 -j ACCEPT #DNS  iptables -A INPUT -p tcp --dport 1723 -j ACCEPT #VPN  iptables -A INPUT -p gre -j ACCEPT  iptables -A INPUT -s 192.168.0.0/24 -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT  iptables -A INPUT -i ppp0 -p tcp --syn -m connlimit --connlimit-above 15 -j DROP  iptables -A INPUT -p icmp -j DROP  #防止外部的SYN洪水攻击  iptables -t nat -A POSTROUTING -o ppp0 -s 10.10.10.0/24 -j MASQUERADE  iptables -N syn-flood  iptables -A sys-flood -p tcp -m limit --limit 3/s --limit-burst 6 -j RETURN  iptables -A sys-flood -j REJECT  #forward链配置  iptables -P FORWARD DROP  iptables -A FORWARD -p tcp -s 10.10.0.0/24 -m multiport --dport 80,110,21,25,1723 -j ACCEPT  iptables -A FORWARD -p udp -s 10.10.0.0/24 --dport 53 -j ACCEPT  iptables -A FORWARD -p gre -s 10.10.0.0/24 -j ACCEPT  iptables -A FORWARD -p icmp -s 10.10.0.0/24 -j ACCEPT  iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT  ###过滤一些关键词  iptables -I FORWARD -p udp --dport 53 -m string --string "tencent" -m time --timestart 8:15 --timestop 12:30 --days Mon,Wed -j DROP  iptables -I FORWARD -s 10.10.0.0/24 -m string --string "ay.net" -j DROP  ###修改内核  sysctl -w net.ipv4.ip\_forward=1&>/dev/null  sysctl -w net.ipv4.tcp\_syncookies=1&>/dev/null  ###允许本机  iptables -I INPUT -s 10.10.0.50 -j ACCEPT  iptables -I FORWARD -s 10.10.0.50 -j ACCEPT |

## 生产中配置iptables方法

### 设置白名单

iptables -A INPUT -s 10.XX.XX.XX -j ACCEPT

### 安全防护措施

#防止外部的SYN洪水攻击

|  |
| --- |
| iptables -N synflood  iptables -A synflood -m limit --limit 10/s --limit-burst 100 -j RETURN  iptables -A synflood -p tcp -j REJECT --reject-with tcp-reset  iptables -A INPUT -p tcp -m state --state NEW -j synflood |

#防止ping洪水攻击，限制每秒的ping包不超过5个

|  |
| --- |
| #--limit-burst 5初始为5，然后--limit 1/s  iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/s --limit-burst 5 -j ACCEPT |

### recent模块限制同一ip连接数

|  |
| --- |
| #允许一个客户端每分钟内可以接上服务器20个新连接  iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/s --limit-burst 5 -j ACCEPT  iptables -A INPUT -j REJECT --reject-with icmp-host-prohibited  iptables -A FORWARD -j REJECT --reject-with icmp-host-prohibited |

### 生产配置的完整例子

--iptables配置

|  |
| --- |
| yum install iptables-services -y #安装  echo "1" > /proc/sys/net/ipv4/tcp\_syncookies  cat > /etc/sysconfig/iptables <<EOF  # Firewall configuration written by system-config-firewall  # Manual customization of this file is not recommended.  \*filter  :INPUT ACCEPT [0:0]  :FORWARD ACCEPT [0:0]  :OUTPUT ACCEPT [0:0]  :synflood - [0:0]  -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  -A INPUT -p icmp -j ACCEPT  -A INPUT -i lo -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT  #设置主机白名单  -A INPUT -s 122.225.227.163 -j ACCEPT  -A INPUT -s 122.224.66.34 -j ACCEPT  -A INPUT -s 10.161.166.225 -j ACCEPT  -A INPUT -s 10.162.94.58 -j ACCEPT  -A INPUT -s 10.161.219.27 -j ACCEPT  -A INPUT -s 10.161.220.172 -j ACCEPT  -A INPUT -s 10.162.57.241 -j ACCEPT  -A INPUT -s 10.161.171.195 -j ACCEPT  -A INPUT -s 10.161.158.22 -j ACCEPT  -A INPUT -s 10.161.132.101 -j ACCEPT  -A INPUT -s 10.161.189.96 -j ACCEPT  -A INPUT -s 10.161.185.101 -j ACCEPT  -A INPUT -s 10.162.61.168 -j ACCEPT  -A INPUT -s 10.161.226.140 -j ACCEPT  -A INPUT -s 10.161.189.223 -j ACCEPT  -A INPUT -s 10.162.87.169 -j ACCEPT  -A INPUT -s 10.162.57.147 -j ACCEPT  -A INPUT -s 10.161.214.153 -j ACCEPT  -A INPUT -s 10.161.218.32 -j ACCEPT  -A INPUT -s 10.161.219.44 -j ACCEPT  -A INPUT -s 10.161.164.55 -j ACCEPT  -A INPUT -s 10.161.212.254 -j ACCEPT  -A INPUT -s 10.161.216.212 -j ACCEPT  -A INPUT -s 10.161.227.194 -j ACCEPT  -A INPUT -s 10.161.175.154 -j ACCEPT  #防止外部的SYN洪水攻击  -A INPUT -p tcp -m state --state NEW -j synflood  -A synflood -m limit --limit 10/sec --limit-burst 100 -j RETURN  -A synflood -p tcp -j REJECT --reject-with tcp-reset  -A INPUT -j REJECT --reject-with icmp-host-prohibited  #防止ping洪水攻击，限制每秒的ping包不超过5个  -A INPUT -p icmp -m icmp --icmp-type 8 -m limit --limit 1/sec -j ACCEPT  -A FORWARD -j REJECT --reject-with icmp-host-prohibited  COMMIT  EOF  service iptables restart |

--TCP wrappers限制访问ssh的ip

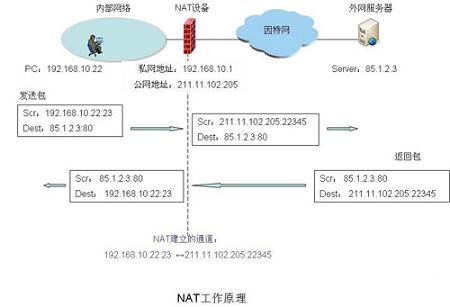
|  |
| --- |
| cat > /etc/hosts.allow <<EOF  sshd:122.225.227.163  EOF  cat >/etc/hosts.deny <<EOF  sshd:ALL  EOF |

# NAT配置

## NAT简介

NAT（Network Address Translation）中文名字是网络地址转换，它是一个IETF标准，允许整个机构以一个公用的ip地址出现在一个Internet上面。他就是一种把内部私有网络地址翻译成公有合法的ip地址。在一定程度上解决ip地址不足的问题。

## NAT工作原理



NAT基本工作原理是：当私有网络主机和公共网络主机通讯的ip包经过NAT网关后，将ip包的源代码地址或者目的ip地址在私有ip和公共ip之间进行装换。发送数据包的时候，nat会将ip包的源ip地址改成nat的公网地址。Nat接收到返回的数据包时候，nat会将目的地址改成内网私有地址。

## NAT分类

源NAT(SNAT)：修改数据包的源地址。

目的NAT(DNAT)：修改数据包的目的地址

伪装(MASQUERADE)：用发送数据的网卡上的ip来替换源ip.

## Iptables配置NAT规则

|  |
| --- |
| #开启路由转发  echo "1" > /proc/sys/net/ipv4/ip\_forward  #在NAT服务器上编写SNAT，实现内网访问外网#将内网地址段转换成10.10.10.1  iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.0/24 -j SNAT --to 10.10.10.1  ＃在NAT服务器上编写DNAT,实现外网访问内网将目的地址为10.10.10.1转换成内网www服务器地址192.168.0.2  iptables -t nat -A PREROUTING -i eth0 -s 10.10.10.0/24 -d 10.10.10.1 -p tcp --dport 80 -j DNAT --to 192.168.0.2 |