

时不待我!

<http://zengweidao.blog.51cto.com> 【复制】 【订阅】

主页 | linux日记 | linux运维实用命令及工具 | linux基本设置 | linux服务搭建 | shell脚本应用

zengweidao 的BLOG

博主的更多文章>>



发私信

加友情链接

博客统计信息

用户名: zengweidao
文章数: 20
评论数: 0
访问量: 10693
无忧币: 309
博客积分: 196
博客等级: 2
注册日期: 2013-12-11

热门专题 更多>>



Oracle零基础成长之路

阅读量: 1297



原来你也在这里(征文)

阅读量: 3317



从菜鸟到老鸟-教你玩转Mac操作系统

阅读量: 432596



QT学习之路: 从入门到精通

阅读量: 1120555

热门文章

利用xshell密钥管理服务..

yum源制作、U盘做yum源

shell运用中exec与文件描..

关于Linux字符集的查看及..

iptables详解加实战

nfs、dhcp、tftp组建远程..

在Linux上限制远程登陆的IP

DNS服务搭建、转发、主从..

原创 iptables详解加实战

2014-06-30 21:02:23

标签: iptables filter

原创作品，允许转载，转载时请务必以超链接形式标明文章 [原始出处](#)、作者信息和本声明。否则将追究法律责任。
任。<http://zengweidao.blog.51cto.com/8342699/1432696>

iptables 组件是一种工具，也称为用户空间（userspace），它使插入、修改和除去信息包过滤表中的规则变得容易。分为四个表和五个链，其中表是按照对数据包的操作区分的，链是按照不同的Hook点来区分的，表和链实际上是netfilter的两个维度。

4个表:filter,nat,mangle,raw，默认表是filter（没有指定表的时候就是filter表）。表的处理优先级: raw>mangle>nat>filter。

filter: 一般的过滤功能，如不-t指定表，则默认filter

nat:用于nat功能（端口映射，地址映射等）

mangle:用于对特定数据包的修改

raw:优先级最高，设置raw时一般是为了不再让iptables做数据包的链接跟踪处理，提高性能

5个链: PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING。

PREROUTING:数据包进入路由表之前

INPUT:通过路由表后目的地为本机

FORWARDING:通过路由表后，目的地不为本机

OUTPUT:由本机产生，向外转发

POSTROUTING:发送到网卡接口之前

用表格表示下他们之间关系

四个表/五个链	PREROUTING	INPUT	FORWARD	OUTPUT	POSTROUTING
rew	√			√	
mangle	√	√	√	√	√
nat	√			√	√
filter		√	√	√	

用图来表示他们的过滤匹配流程

搜索BLOG文章

搜索

最近访客



雾满天空



wangy..



inter..



power..



wy125629



goku0519



kxfzmm



赵东军



qishu..



Boom56



wx58f..



yaoko..

最新评论

51CTO推荐博文

更多>>

一位架构师用服务打动客户的故事

在线考试系统从Windows系统迁移到..

案例 - 一个IP切换引发的数据不一致

阿里云常见问题分析与解答

Web服务基础

一键自动化部署（定制rpm包）

基于etcd+confd通过nginx对docker..

如何防御DDOS等流量攻击

主流区块链技术特点及Fabric V0.6..

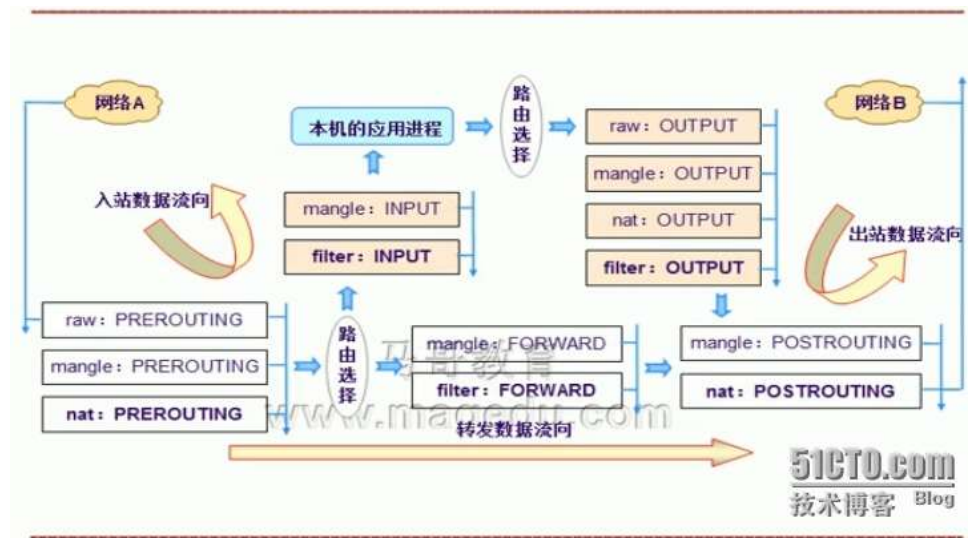
总编下午茶：挑战者心态能否帮助..

Powershell管理系列（三十八）Pow..

友情链接



开发



i>、进站数据流向：来自外界的数据包到达防火墙后，首先被PREROUTING规则链处理（是否修改数据包地址等），之后会进行路由选择（判断该数据包应该发往何处），如果数据包的目标地址是防火墙本机（如Internet用户访问防火墙主机中的Web服务的数据包），那么内核将其传递给INPUT链进行处理（决定是否允许通过等），通过以后再交给系统上层的应用程序（如httpd服务器）进行响应。

ii>、转发数据流向：来自外界的数据包到达防火墙后，首先被PREROUTING规则链处理，之后会进行路由选择，如果数据包的目标地址是其他外部地址（如局域网用户通过网关访问QQ站点的数据包），则内核将其传递给FORWARD链进行处理（是否转发或拦截），然后再讲给POSTROUTING规则链（是否修改数据包的地址等）进行处理。

iii>、出站数据流向：防火墙本机向外部地址发送的数据包（如在防火墙主机中测试公网DNS服务时），首先被OUTPUT规则链处理，之后进行路由选择，然后传递给POSTROUTING规则链（是否修改数据包的地址等）进行处理。

规则链内部各条防火墙规则之间的优先顺序：

在数据包经由各条规则链的处理过程中，依次按第1条规则、第2条规则、第3条规则....的顺序进行匹配和处理。如果找到一条能够匹配该数据包的规则，则不会继续检查后面的规则（使用LOG记录日志的规则除外，下面iptables配置说明）。如果比对完整个规则链，也找不到和数据包相匹配的规则，就按照该规则链的默认策略进行处理。

常用参数：

链中规则：

- A 添加一条规则
- I 根据给出的规则序号向所选链中插入一条或更多规则
- R 从选中的链中取代一条规则
- D 删除一条规则

链：

- N 新建一条自定义链
- X 删除一条自定义链
- E 重命名一条自定义链
- F 清空指定链，如果不指定链，则清空表中的所有链
- P 设定链的默认策略
- Z 把所有链的包及字节的计数器清空

查看：

- L 查看
- v 详细查看
- vv 更详细查看
- x 显示包和字节计数器的精确值，代替用K, M, G表示的约数。这个选项仅能用于 '-L -v' 命令
- n 数字输出。IP地址和端口会以数字的形式打印
- line-numbers 在每个列表前加上行号

服务脚本：/etc/rc.d/init.d/iptables

脚本配置文件：/etc/sysconfig/iptables-config

service iptables {status|start|stop|restart|save}

规则的保存位置/etc/sysconfig/iptables

匹配条件：

通用匹配

- s 源地址匹配
- d 目标地址匹配

（源地址和目标地址可以是ip地址，也可以是网络地址，网络地址需加上掩码，如：192.168.1.0/24。如需取反则加上!，如：!192.168.1.11，即除192.168.1.11以外的ip）

```
-p 协议匹配, 通常有三种协议 {icmp|tcp|udp}
-i IN_INTERFACE 流入接口, 注意: 通常接的链有 PREROUTING、INPUT、FORWARD
-o OUT_INTERFACE 流出接口, 注意: 通常接的链有 FORWARD、OUTPUT、POSTROUTING
```

扩展匹配

隐式扩展

```
-p tcp
    --dport PORT
    --sport PORT
```

(这里可以指定端口片, 如: {22: 67} 指定22端口到67端口, {22: } 指定22端口以上, { :22} 指定22端口以下, 也可以使用!取反 (!22:67})

--tcp-flags SYN,ACK,RST,FIN SYN 检查tcp标志位, SYN为1, 其他三个为0(tcp标识位有6个, 分别是SYN、ACK、URG、PSH、RST、FIN)——建立连接、确认、紧急、强迫、复位、结束)

```
-p udp
    --sport PORT
    --dport PORT
```

(这里同上tcp)

```
-p icmp
    --icmp-type typename
```

(typename有'8' 请求、'0' 应答等类型, 例如配置一个不让某网段ping本机: iptables -t filter -A INPUT -s 192.168.0.0/24 -p icmp --icmp-type 8 -j DROP)

显示扩展

(netfilter 扩展模块引入的扩展, 用于扩展匹配条件, 通常需要额外专用选项来定义)

```
-m state:用于实现连接的状态检测模块
    --state NEW (新建连接), ESTABLISHED (确认连接), RELATED (关联连接), INVALID (无效连接)
-m multiport :多端口
    --source-port :源端口组
    --destination-ports :目标端口组
    --ports
```

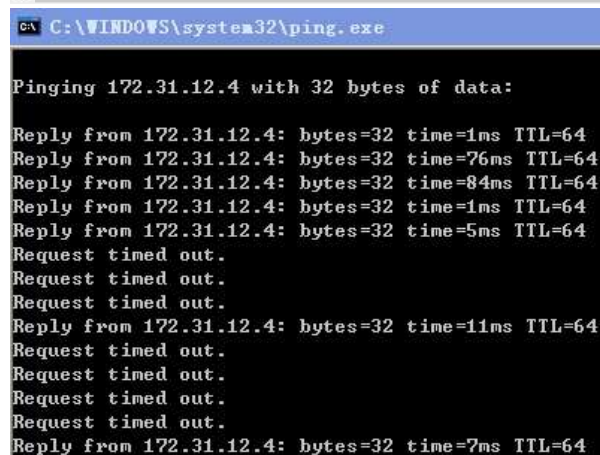
例如: 将同是tcp协议的22、80端口合并一条规则

```
1 [root@david ~]#iptables -L INPUT -n
2 Chain INPUT (policy DROP)
3 target prot opt source destination
4 ACCEPT tcp -- 0.0.0.0/0 172.31.12.4 tcp dpt:22
5 ACCEPT tcp -- 0.0.0.0/0 172.31.12.4 tcp dpt:80
6 [root@david ~]#iptables -F INPUT
7 [root@david ~]#iptables -I INPUT 1 -d 172.31.12.4 -p tcp -m multiport --destination-ports 22,80
```

```
-m limit :速率限制
    --limit 速率 (如: 3/second 表示每秒3个数据包)
    --limit-burst 峰值速率 (如100 表示不能超过100个数据包)
```

例如: 限制其他机器每分钟同时Ping本机eth0网卡的速率

```
1 [root@david ~ 15:52 &1]#iptables -A INPUT -i eth0 -d 172.31.12.4 -p icmp --icmp-type 8 -m limit -
2 [root@david ~ 15:56 &2]#iptables -A INPUT -i eth0 -d 172.31.12.4 -p icmp --icmp-type 8 -j DROP
```



```
C:\WINDOWS\system32\ping.exe

Pinging 172.31.12.4 with 32 bytes of data:

Reply from 172.31.12.4: bytes=32 time=1ms TTL=64
Reply from 172.31.12.4: bytes=32 time=76ms TTL=64
Reply from 172.31.12.4: bytes=32 time=84ms TTL=64
Reply from 172.31.12.4: bytes=32 time=1ms TTL=64
Reply from 172.31.12.4: bytes=32 time=5ms TTL=64
Request timed out.
Request timed out.
Request timed out.
Reply from 172.31.12.4: bytes=32 time=11ms TTL=64
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 172.31.12.4: bytes=32 time=7ms TTL=64
```

```
-m connlimit :连接上限
    --connlimit-above n 多于n个表示满足条件 取反要在选项前加!
-m iprange :ip地址范围
    --src-range ip-ip
    --dst-range ip-ip
```

(例如: -m iprange --src-range 172.31.12.1-172.31.12.50)

```
-m mac mac地址限制
    --mac-source
-m string 字符串编码特征匹配
```

```
--algo [bm|kmp] 匹配算法
--string "Pattern" 要匹配的字符串
-m recent 有点类似坏人名单
--set 添加源地址的包到列表中
--update 每次建立连接都更新列表
--seconds 必须与--rcheck或--update同时使用
--hitcount 必须与--rcheck或--update同时使用
```

例：限制ping 192.168.146.3主机的数据包数，平均2/s个，最多不能超过3个

```
iptables -A INPUT -i eth0 -d 192.168.146.3 -p icmp --icmp-type 8 -m limit --limit 2/second --limit-burst 3 -j ACCEPT
```

例：限制SSH连接速率(默认策略是DROP)

```
iptables -I INPUT 1 -p tcp --dport 22 -d 192.168.146.3 -m state --state ESTABLISHED -j ACCEPT
```

```
iptables -I INPUT 2 -p tcp --dport 22 -d 192.168.146.3 -m limit --limit 2/minute --limit-burst 2 -m state --state NEW -j ACCEPT
```

例：web服务器不响应内容含有yao页面的请求

```
iptables -I OUTPUT 1 -s 192.168.146.3 -p tcp --sport 80 -m string --algo kmp --string "yao" -j DROP
```

例：利用recent模块抵御DOS攻击

```
iptables -I INPUT -p tcp --dport 22 -m connlimit --connlimit-above 3 -j DROP 单个IP最多连接3个会话,
```

```
iptables -I INPUT -p tcp --dport 22 -m state --state NEW -m recent --set --name SSH
```

只要是新的连接请求，就把它加入到SSH列表中

```
Iptables -I INPUT -p tcp --dport 22 -m state NEW -m recent --update --seconds 300 --hitcount 3 --name SSH -j DROP
```

5分钟内你的尝试次数达到3次，就拒绝提供SSH列表中的这个IP任何服务。被限制5分钟后即可恢复访问。

本文出自 “时不待我!” 博客，请务必保留此出处<http://zengweidao.blog.51cto.com/8342699/1432696>

分享至:

收藏 

 0人

了这篇文章

类别: [linux运维实用命令及工具](#) | 阅读(99) | 评论(0) | [返回博主首页](#) | [返回博客首页](#)

[上一篇 nfs、dhcp、tftp组建远程无人值守系统安装](#) [下一篇 利用xshell密钥管理服务器远程登录](#)



文章评论

发表评论

昵 称:

[登录](#) [快速注册](#)

验证码:

请点击后输入验证码 [博客过2级，无需填写验证码](#)

内 容:

Copyright By 51CTO.COM 版权所有

51CTO 技术博客