iptables

iptables的20道面试题

http://blog.51cto.com/oldboy/1637493

1）关闭firewalld服务器

[root@proxy ~]# systemctl stop firewalld.service

[root@proxy ~]# systemctl disable firewalld.service  
2）安装iptables-services并启动服务

[root@proxy ~]# yum -y install iptables-services

[root@proxy ~]# systemctl start iptables.service

熟悉iptables框架

1）iptables的4个表（区分大小写）：

iptables默认有4个表，nat表（地址转换表）、filter表（数据过滤表）、raw表（状态跟踪表）、mangle表（包标记表）。

2）iptables的5个链（区分大小写）：

INPUT链（入站规则）

OUTPUT链（出站规则）

FORWARD链（转发规则）

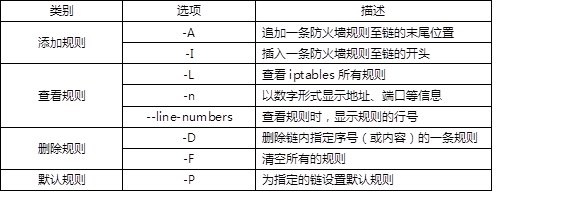
PREROUTING链（路由前规则）

POSTROUTING链（路由后规则）

iptabels语法格式

1. **[**root@proxy **~]**# iptables **[-**t 表名**]** 选项 **[**链名**]** **[**条件**]** **[-**j 目标操作**]**
2. **[**root@proxy **~]**# iptables **-**t filter **-**I INPUT **-**p icmp **-**j REJECT
3. **[**root@proxy **~]**# iptables **-**t filter **-**I INPUT **-**p icmp **-**j ACCEPT
4. **[**root@proxy **~]**# iptables **-**I INPUT **-**p icmp **-**j REJECT
5. //注意事项与规律：
6. //可以不指定表，默认为filter表
7. //可以不指定链，默认为对应表的所有链
8. //如果没有找到匹配条件，则执行防火墙默认规则
9. //选项/链名/目标操作用大写字母，其余都小写
10. ########################################################################
11. //目标操作：
12. // ACCEPT：允许通过/放行
13. // DROP：直接丢弃，不给出任何回应
14. // REJECT：拒绝通过，必要时会给出提示
15. // LOG：记录日志，然后传给下一条规则

iptables常用选项



创建规则的案例：

1. **[**root@proxy **~]**# iptables **-**t filter **-**A INPUT **-**p tcp **-**j ACCEPT
2. //追加规则至filter表中的INPUT链的末尾，允许任何人使用TCP协议访问本机
3. **[**root@proxy **~]**# iptables **-**I INPUT **-**p udp **-**j ACCEPT
4. //插入规则至filter表中的INPUT链的开头，允许任何人使用UDP协议访问本机
5. **[**root@proxy **~]**# iptables **-**I INPUT **2** **-**p icmp **-**j ACCEPT
6. //插入规则至filter表中的INPUT链的第2行，允许任何人使用ICMP协议访问本
7. **[**root@proxy **~]**# iptables **-**nL INPUT                    //仅查看INPUT链的规则
8. target prot opt source destination
9. ACCEPT udp **--** **0.0.0.0/0** **0.0.0.0/0**
10. ACCEPT icmp **--** **0.0.0.0/0** **0.0.0.0/0**
11. ACCEPT tcp **--** **0.0.0.0/0** **0.0.0.0/0**
12. **[**root@proxy **~]**# iptables **-**L INPUT **--**line**-**numbers        //查看规则，显示行号
13. num target prot opt source destination
14. **1** ACCEPT udp **--** anywhere anywhere
15. **2** ACCEPT icmp **--** anywhere anywhere
16. **3** ACCEPT tcp **--** anywhere anywhere

1、selinux（生产中也是关闭的），ids入侵检测，md5指纹。

2、iptables（生产环境中，内网关闭，外网打开），大并发的情况，不能开iptables，影响性能，硬件防火墙。

安全优化：

1、尽可能不给服务器配置外网IP。可以通过代理转发或者通过防火墙映射。

2、并发不是特别大情况再外网IP的环境，要开启iptables防火墙。

学好iptables的基础

1、OSI7层模型以及不同层对应哪些协议？

2、TCP/IP三次握手，四次断开的过程，TCP HEADER

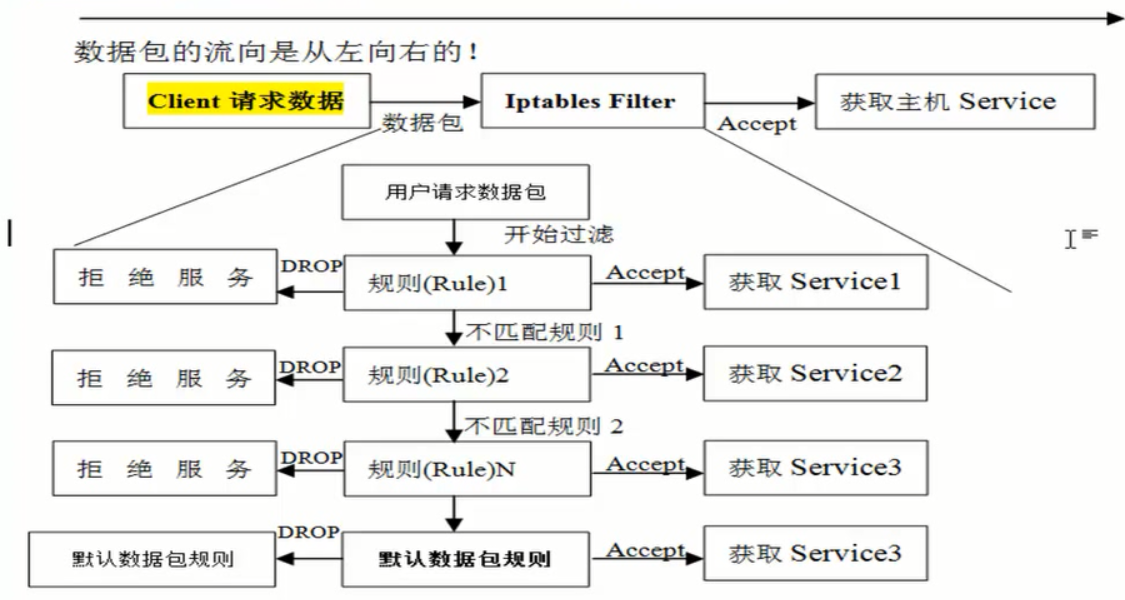
3、常用的服务端口要了如指掌

Iptables 是unix/linux自带的一款优秀且开放源代码的完全自由基于包过滤的防火墙工具。它的功能十分强大，使用非常灵活，可以对流入和流出服务器的数据包进行很精细的控制。特别是它可以在一台非常低的硬件配置下跑的非常好（本人曾经在赛扬500HZ cpu 64M内存的情况部署网关防火墙）提供近400人的上网丝毫不逊色企业级专业路由防火墙。）iptables+zebra+squid

iptables是linux2.4及2.6内核中集成的服务。其功能与安全性比其老一辈ipfwadm,ipchains强大得多，iptables主要工作在OSI七层的二、三、四层，如果重新编译内核，iptables也可以支持7层控制（squid代理+iptables）。

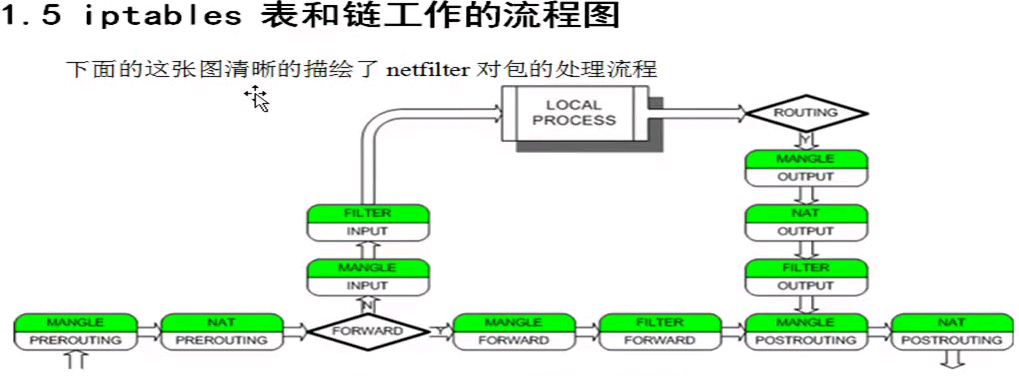
（iptables是采用数据包过滤机制工作的，它会对请求换数据包的包头数据进行分析，并根据我们预先设定的规则进行匹配来决定是否可以进入主机 ）

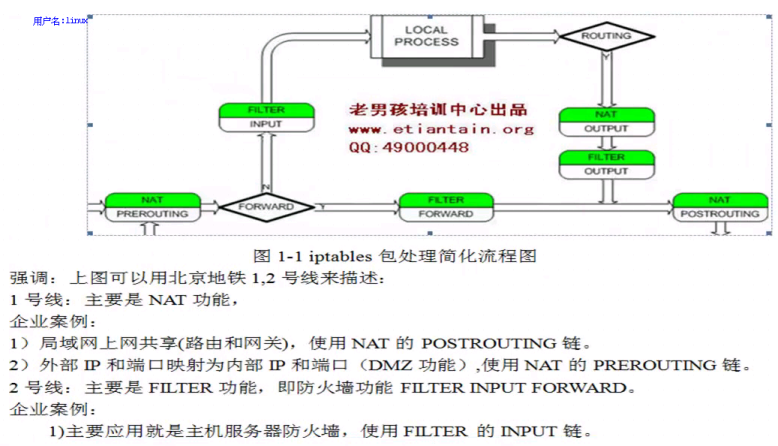
**iptables工作流程**



**iptables表和链的工作流程**

**简化后（主要学习）**





**iptables工作流程小结**：

1. 防火墙是一层层过滤的。实际是按照配置规则的顺序从上到下，从前到后进行过滤的。

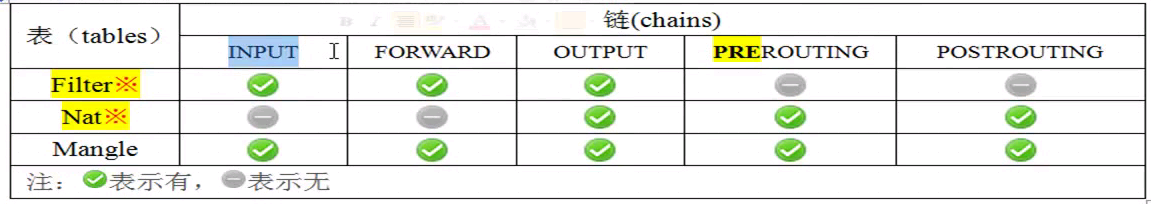
2、如果匹配上规则，即明确表明是阻止还是通过，此时数据包就不在向下匹配新规则了。

3、如果所有规则中没有明确表明是阻止还是通过这个数据包，也就是没有匹配上规则，向下进行匹配，直到匹配默认规则得到明确的阻止还是通过。

4、防火墙是默认规则是对应链的所有的规则执行完才会执行的。

iptable（国家）--->tables表（省）--->chains链（市）--->规则（县）

4表5链



（主要是前两个filter和nat表的使用，后两个几乎没有用到）

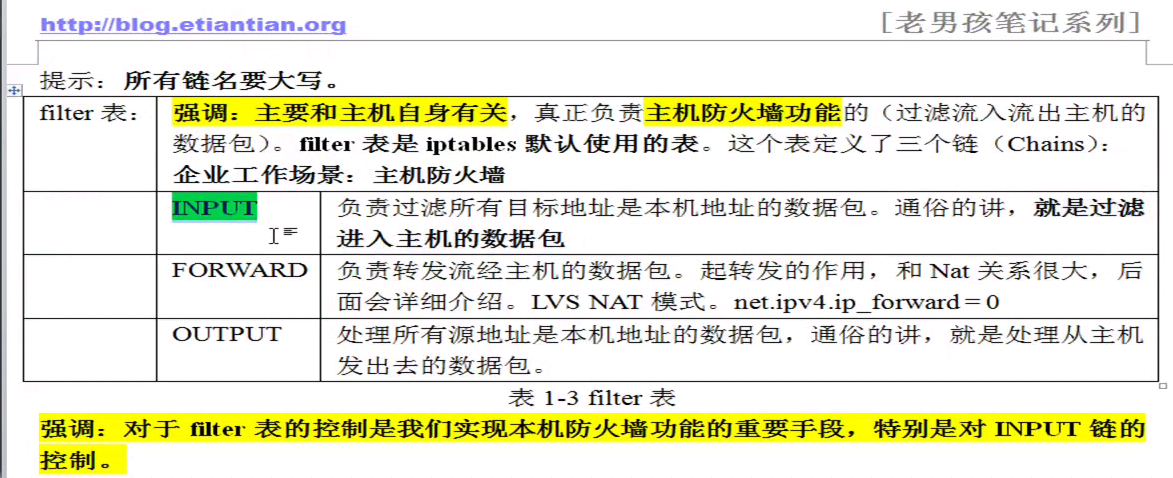
Filter表是真正的防火墙功能（iptables默认使用的表），有INPUT,FORWARD,OUTPUT的3条链，工作中一般对流入的数据包控制。

Nat表是负责数据包转发（改写），网关共享上网、IP和端口映射

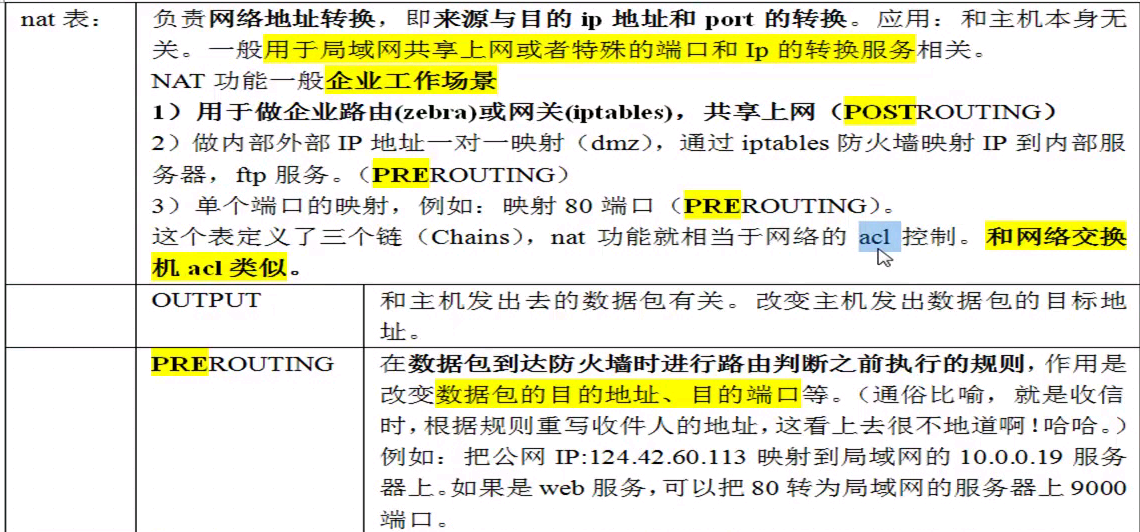
mangle表是做路由标记,这个表有5链。

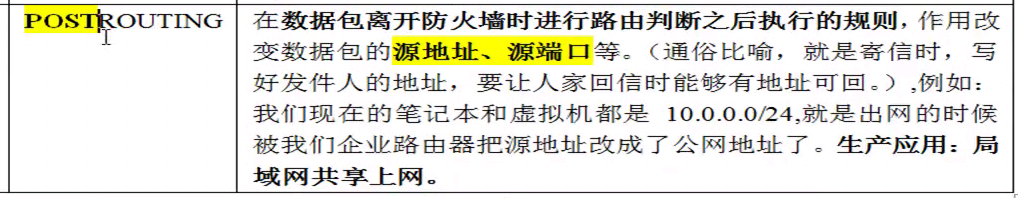
raw表用做连接跟踪

filter表中的3链



nat表中的3链





**iptables命令**

**iptables案例**