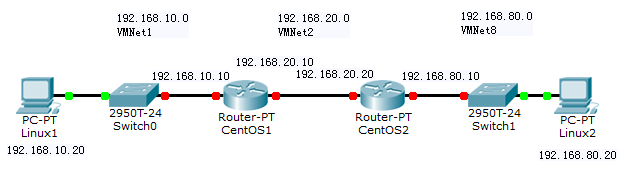
# 第10章 配置Linux防火墙

网络畅通的条件：有去有回，计算机要设置网关，沿途路由器要有到达目标网络的路由和返回源网络的路由。

## 本章实验环境能

本章第09章搭建的环境



需要4个CentOS Linux。

其中两个作为路由器CentOS1和CentOS2，添加两个网卡，网卡按着拓扑设计指定到相应的网络。

## 查看默认IPtables表和链

-L参数可以显示iptables表和链，后面会详细介绍iptables命令的用法

[root@WebServer ~]# iptables -t filter –L

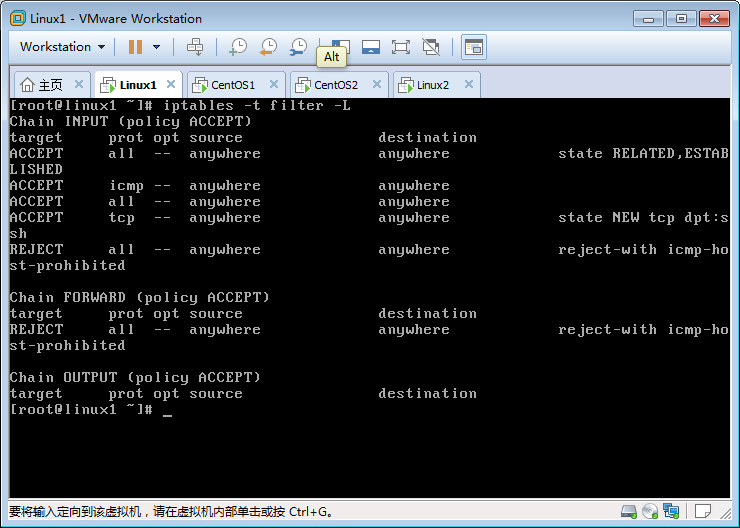
可以指明显示哪个链的设置

[root@WebServer ~]# iptables -t filter -L INPUT

如果不指名什么表 默认就是filter表

[root@LinuxServer ~]# iptables –L

### 显示Linux1的filter表



## Iptables实现网络安全

### Iptables命令

Iptables -t table 命令 chain rules –j target

table 可以是 filter nat mangle 默认 filter

命令

-P或--policy 定义默认策略

-L或—list 查看iptables规则列表

-A或—append 在规则列表的最后增加1条规则

-I或--insert 在指定的位置插入1条规则

-D或--delete 从规则列表中删除1条规则

-R或—replace 替换规则列表中的某条规则

-F或—flush 删除表中所有规则

### Iptables匹配选项

-i或--in-interface 指定数据包从哪个网络接口进入，如ppp0、eth0和eth1等

-o或--out-interface 指定数据包从哪块网络接口输出，如ppp0、eth0和eth1等

-p或--protocol协议类型 指定数据包匹配的协议，如TCP、UDP和ICMP等

-s或—source 指定数据包匹配的源地址

-d或--destination 指定数据包匹配的目标地址

--sport 指定数据包匹配的源端口号，可以使用“起始端口号:结束端口号”的格式指定一个范围的端口

--dport目标端口号：指定数据包匹配的目标端口号，可以使用“起始端口号:结束端口号”的格式指定一个范围的端口

匹配选项示例

拒绝从eth3网卡进入ICMP协议数据包

[root@WebServer ~]# iptables -t filter -I INPUT -p icmp -i eth3 -j DROP

[root@WebServer ~]# iptables -t filter -I FORWARD -p tcp -s 192.168.80.123/32 -d 192.168.10.123/32 --dport 3389 -j DROP

[root@WebServer ~]# iptables -t filter -A FORWARD -s 192.168.80.0/24 -d 192.168.10.0/24 -j ACCEPT

### 扩展匹配选项

**参数 -m state**

基于状态检测的包过滤，指定检测那种状态

--state {NEW，ESTATBLISHED，INVALID，RELATED}

说明 用来比对连接状态，连接状态共有四种：INVALID、ESTABLISHED、NEW 和 RELATED。

INVALID 表示该封包的连接编号（Session ID）无法辨识或编号不正确。

ESTABLISHED 表示该封包属于某个已经建立的连接。

NEW 表示该封包想要起始一个连接（重设连接或将连接重导向）。

RELATED 表示该封包是属于某个已经建立的连接，所建立的新连接。例如：FTP-DATA 连接必定是源自某个 FTP 连接。

示例：

单方向建立会话，保护服务器安全

[root@WebServer ~]# iptables -t filter -F FORWARD

[root@WebServer ~]# iptables -t filter –L

在路由器上设置，禁止192.168.10.0/24网段向192.168.80.0/24建立会话，FORWARD链

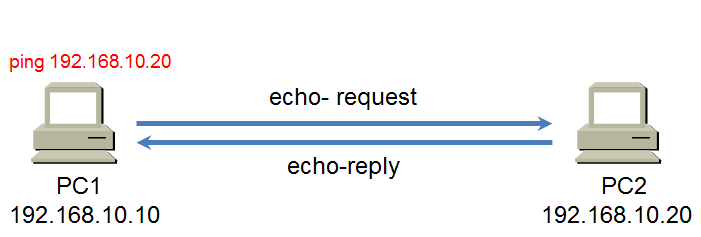
[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -d 192.168.80.0/24 -m state --state NEW -j DROP

在Linux服务器上设置，禁止Linux服务器向外主动建立会话，OUTPUT链。

[root@LinuxServer ~]# iptables -t filter -I OUTPUT -m state --state NEW -j DROP

参数-m icmp --icmp-type

Ping命令使用icmp协议测试网络是否畅通，Icmp有两种常用类型的数据包即icmp-type，常用的类型为echo-reply和echo- request。如下图所示PC1 ping PC2，发出去的数据包是icmp协议echo-reply类型的数据包，PC2返回来的数据包是icmp协议的echo-reply类型的数据包。



示例

禁止192.168.10.0/24ping通192.168.80.0/24

允许192.168.80.0/24ping通192.168.10.0/24

[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -d 192.168.80.0/24 -p icmp -m icmp --icmp-type echo-request -j DROP

禁止其他计算机ping 本地Linux

[root@LinuxServer ~]# iptables -t filter -I INPUT -p icmp -m icmp --icmp-type echo-request -j DROP

Icmp数据包的类型也可以使用数字表示，如

类型8：echo-request。

类型0：echo-reply。

也可以写成

[root@LinuxServer ~]# iptables -t filter -I INPUT -p icmp -m icmp --icmp-type 8 -j DROP

### 处理动作

-j 参数用来指定要进行的处理动作，常用的处理动作包括：ACCEPT、REJECT、DROP、REDIRECT、MASQUERADE、LOG、DNAT、SNAT、MIRROR、QUEUE、RETURN、MARK。

Filter表能使用的主要动作：

ACCEPT： 将封包放行，进行完此处理动作后，将不再匹配其它规则，直接跳往下一个规则链。

REJECT：拦阻该封包，并传送封包通知对方，可以传送的封包有几个选择：ICMP port-unreachable、ICMP echo-reply 或是tcp-reset（这个封包会要求对方关闭 连接），进行完此处理动作后，将不再匹配其它规则，直接中断过滤程序。

j REJECT --reject-with icmp-port-unreachable

[root@linux1 ~]# iptables -t filter -A INPUT -p tcp -s 0.0.0.0/0 -d 0.0.0.0/0 -j REJECT --reject-with tcp-reset

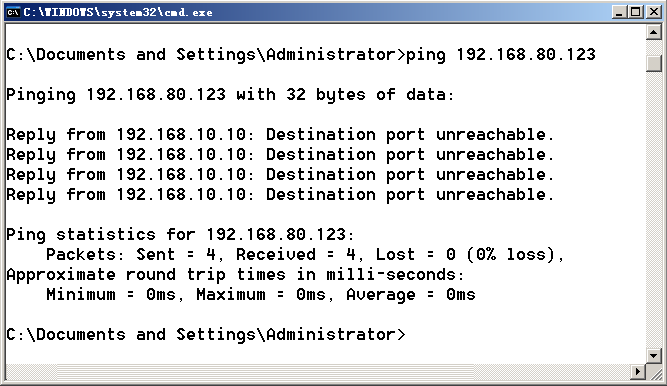
DROP：丢弃封包不予处理，进行完此处理动作后，将不再匹配其它规则，直接中断过滤程序。

LOG： 将封包相关讯息纪录在 /var/log 中，详细位置请查阅 /etc/syslog.conf 配置文件，进行完此处理动作后，将会继续匹配其规则。

REJECT示例

[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -p icmp -j REJECT

[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -p tcp --dport 3389 -j REJECT



例如：

[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -p tcp --dport 3389 -j DROP

[root@WebServer ~]# iptables -t filter -I FORWARD -s 192.168.10.0/24 -p tcp --dport 3389 -j LOG --log-prefix "iptables-rdp"

在Windows2003上远程桌面连接XP

查看记录的日志

[root@WebServer ~]# tail -n 2 /var/log/messages

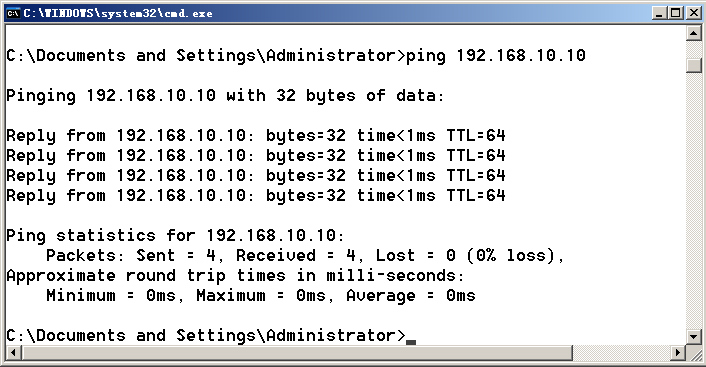
May 21 04:04:40 WebServer kernel: iptables-rdpIN=eth2 OUT=eth3 SRC=192.168.10.140 DST=192.168.80.123 LEN=48 TOS=0x00 PREC=0x00 TTL=127 ID=586 DF PROTO=TCP SPT=1035 DPT=3389 WINDOW=64240 RES=0x00 SYN URGP=0

May 21 04:04:43 WebServer kernel: iptables-rdpIN=eth2 OUT=eth3 SRC=192.168.10.140 DST=192.168.80.123 LEN=48 TOS=0x00 PREC=0x00 TTL=127 ID=587 DF PROTO=TCP SPT=1035 DPT=3389 WINDOW=64240 RES=0x00 SYN URGP=0

记录ping Router1的数据包

[root@WebServer ~]# iptables -t filter -I INPUT -p icmp -j LOG --log-prefix "iptables-icmp"

在Windows2003上ping Router1



查看某类日志 使用前缀过滤信息

[root@WebServer ~]# grep "iptables-rdp" /var/log/messages

[root@WebServer ~]# grep "iptables-icmp" /var/log/messages

更改日志文件，需要更改日志服务的配置文件，将日志指定到新的位置。注释掉原来的位置，在末尾增加一行kern.warning /var/log/iptables.log

[root@WebServer ~]# vi /etc/rsyslog.conf

# Log anything (except mail) of level info or higher.

# Don't log private authentication messages!

#\*.info;mail.none;authpriv.none;cron.none /var/log/messages

kern.warning /var/log/iptables.log

保存配置文件，重启rsyslog服务

[root@WebServer ~]# service rsyslog restart

可以看到新的日志文件产生，里面内容为空。

[root@WebServer ~]# cat /var/log/iptables.log

在Win2003上ping Router1。再次查看新的日志文件，可以看到新增加的记录。

[root@WebServer ~]# tail –n 2 /var/log/iptables.log

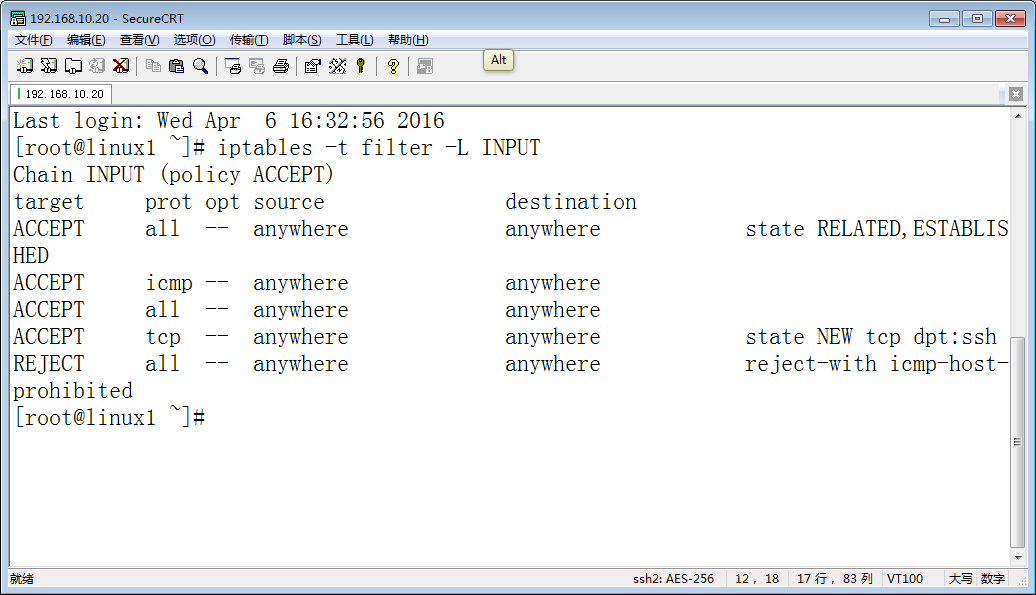
May 21 05:08:42 WebServer kernel: iptables-icmpIN=eth2 OUT= MAC=00:0c:29:ef:78:5a:00:0c:29:db:32:6f:08:00 SRC=192.168.10.140 DST=192.168.10.10 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=668 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=4864

May 21 05:08:43 WebServer kernel: iptables-icmpIN=eth2 OUT= MAC=00:0c:29:ef:78:5a:00:0c:29:db:32:6f:08:00 SRC=192.168.10.140 DST=192.168.10.10 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=669 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=5120

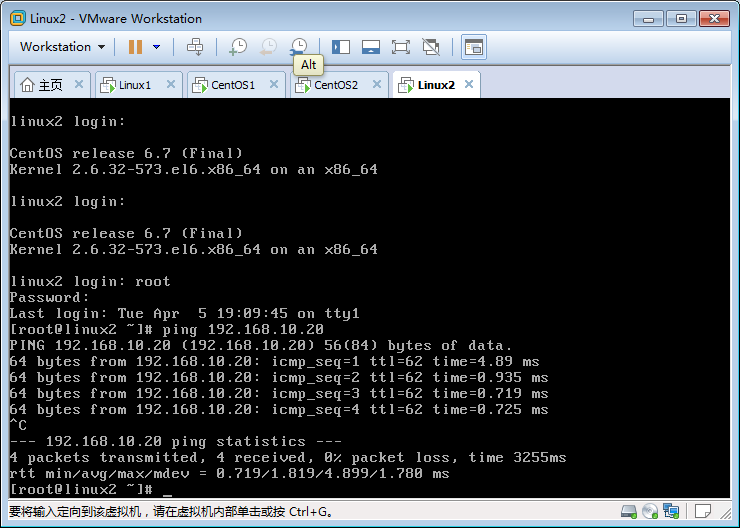
## IPtables应用示例：

### 查看iptables filter表INPUT链规则

查看Linux1的filter表INPUT链规则

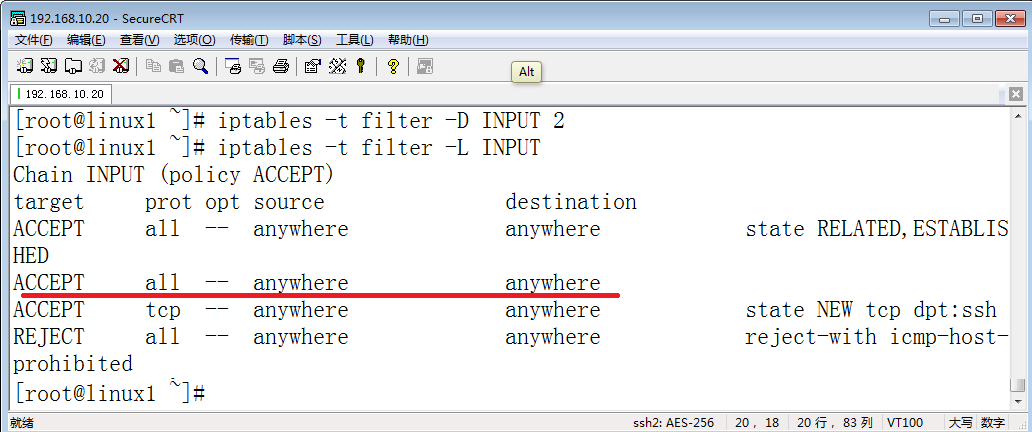
[root@linux1 ~]# iptables -t filter -L INPUT 

在Linux2上ping Linux1，通



删掉Linux1 filter表INPUT链第二条规则

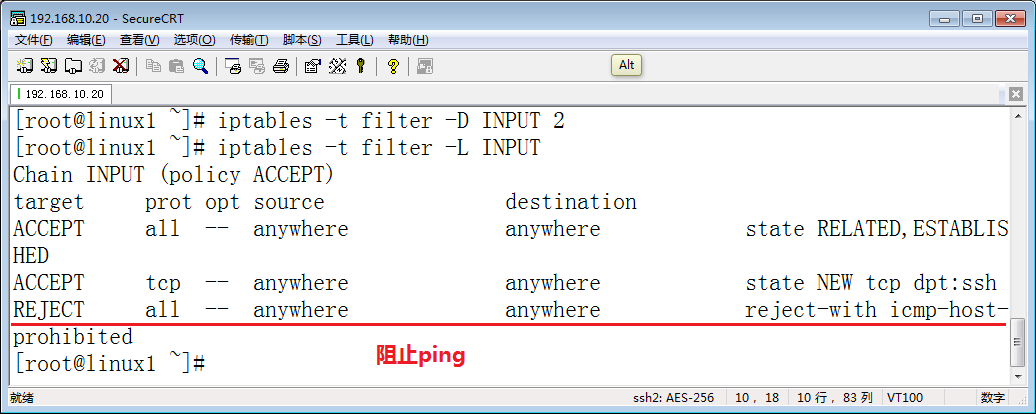
[root@WebServer ~]#iptables –t filter –D INPUT 2

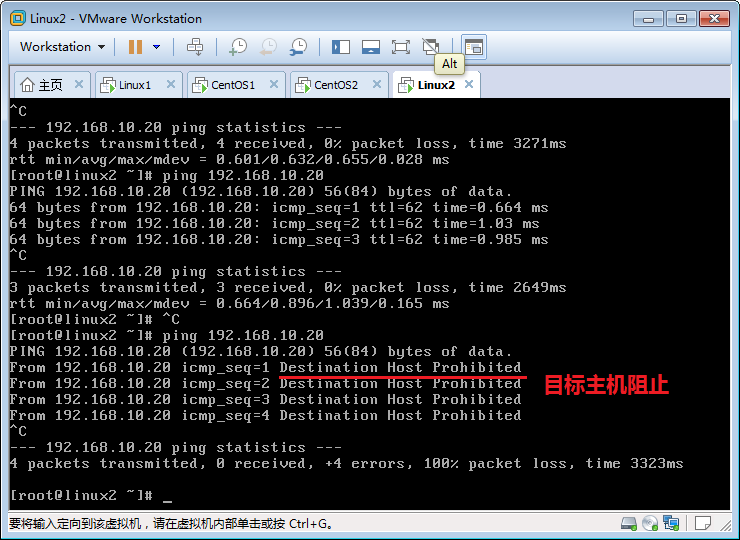


[root@linux1 ~]# iptables -t filter -D INPUT 2 

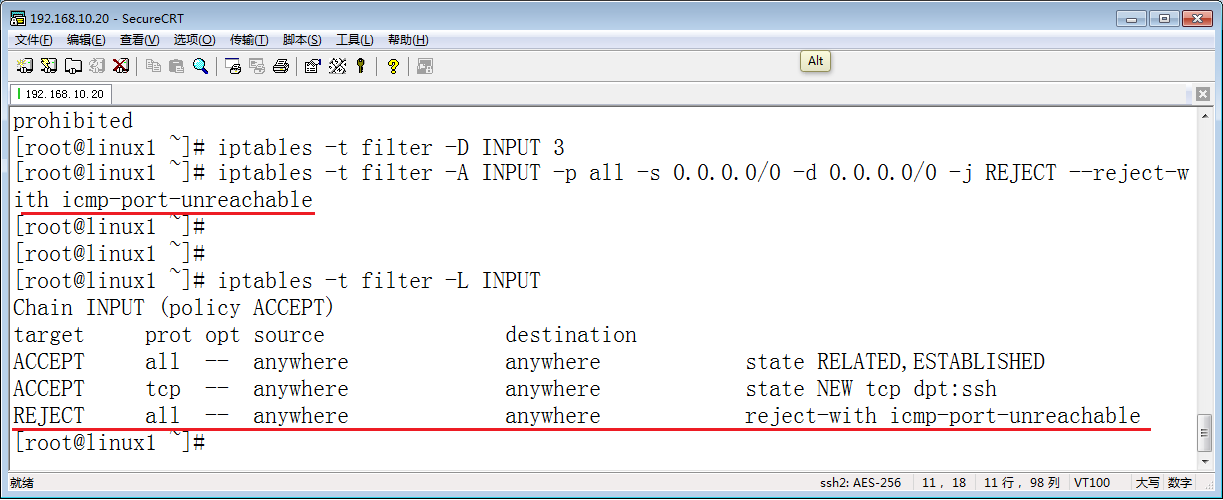
设置filter表INPUT链默认丢弃数据包

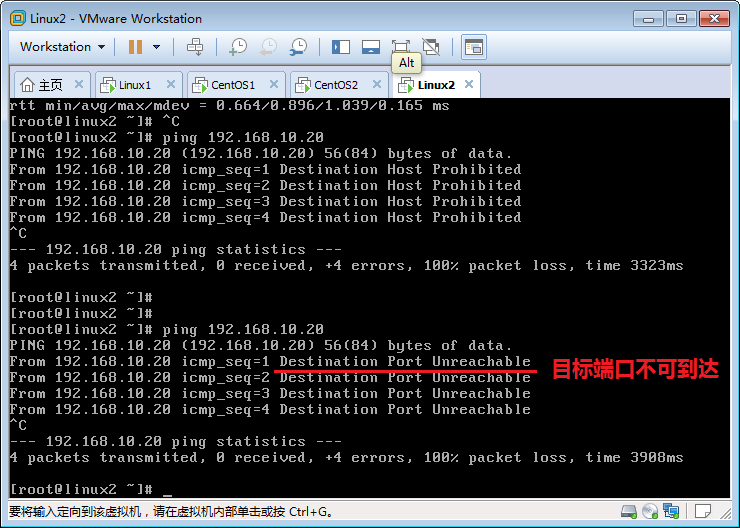
[root@CentOS1 ~]# iptables -t filter -P INPUT DROP





删除第三条规则，追加一条，目标端口不可到达





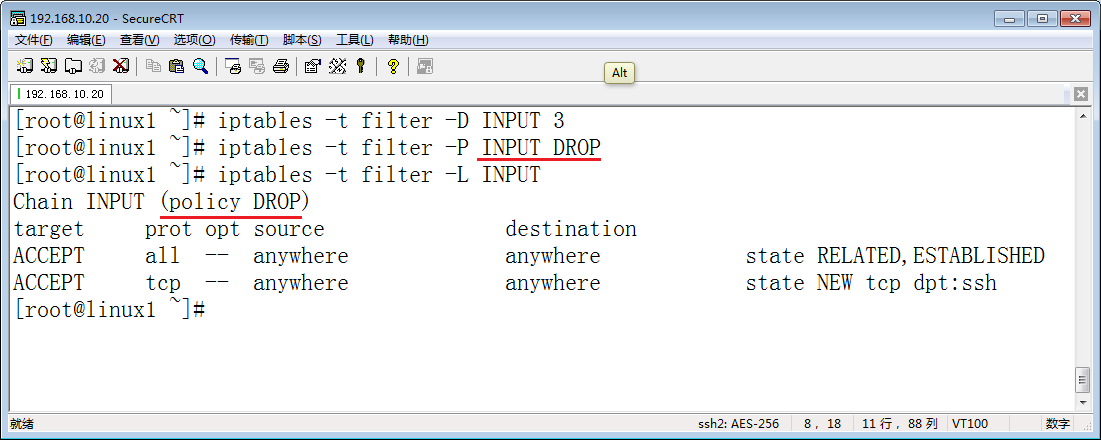
删除最后一条记录，

[root@linux1 ~]# iptables -t filter -D INPUT 3

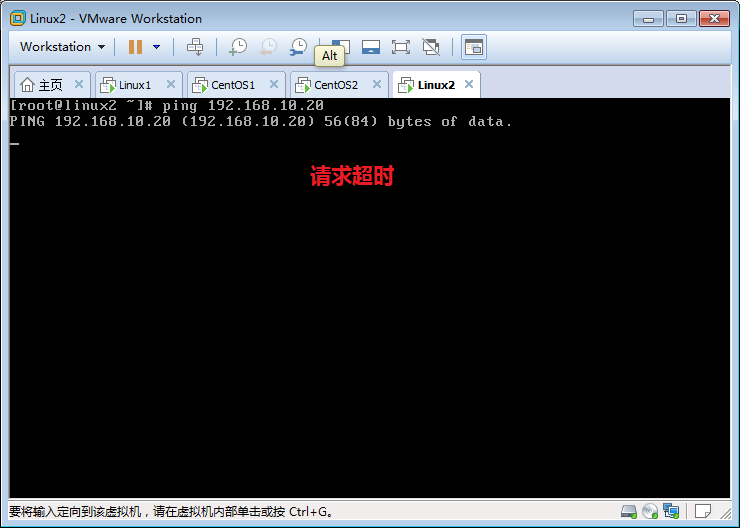
更改默认策略为DROP

[root@linux1 ~]# iptables -t filter -P INPUT DROP

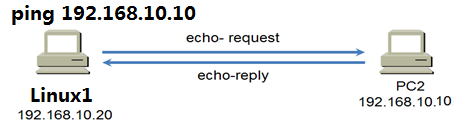
[root@linux1 ~]# iptables -t filter -L INPUT



在Linux2ping Linux1不通。



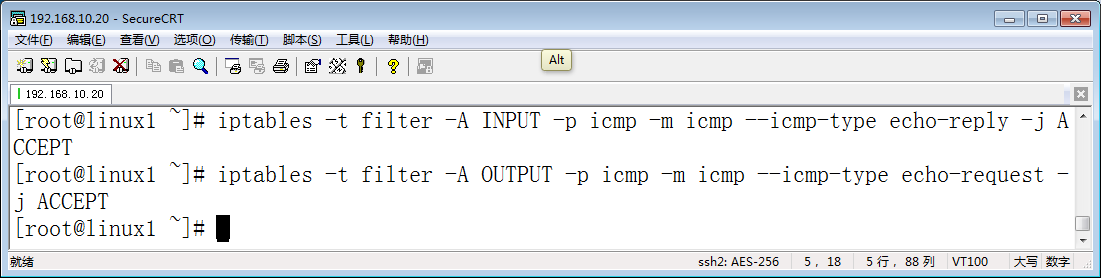
### 拒绝icmp协议数据包进入Linux



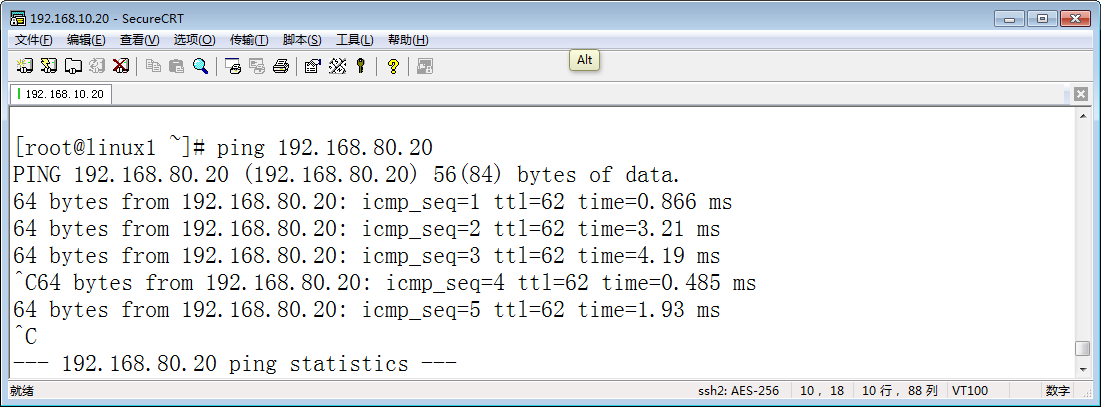
以下的设置允许Linux1ping其他计算机 不允许其他计算机ping通Linux1

[root@linux1 ~]# iptables -t filter -A INPUT -p icmp -m icmp --icmp-type echo-reply -j ACCEPT

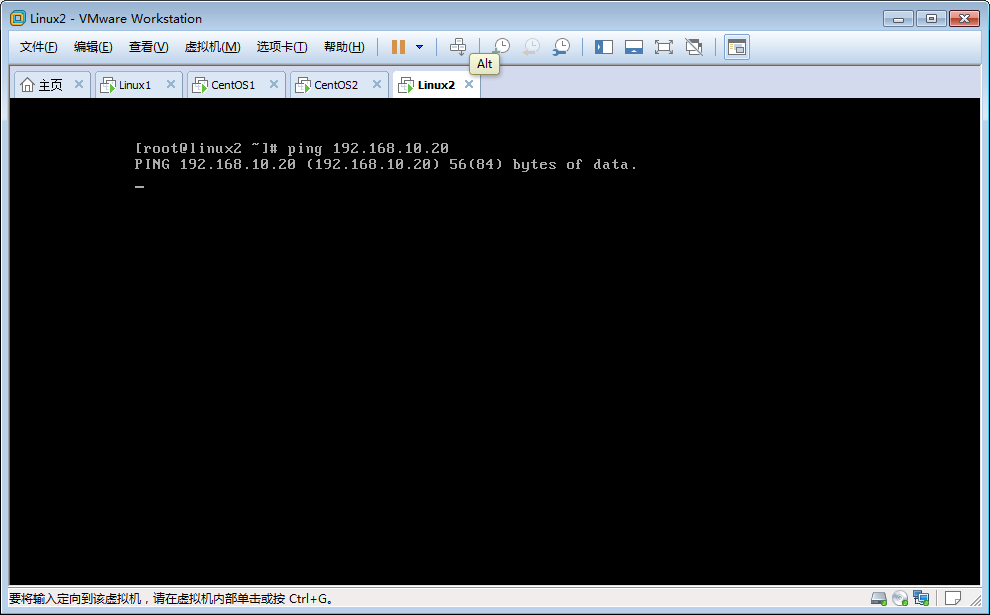
[root@linux1 ~]# iptables -t filter -A OUTPUT -p icmp -m icmp --icmp-type echo-request -j ACCEPT



测试Linux1pingLinux2，能通。



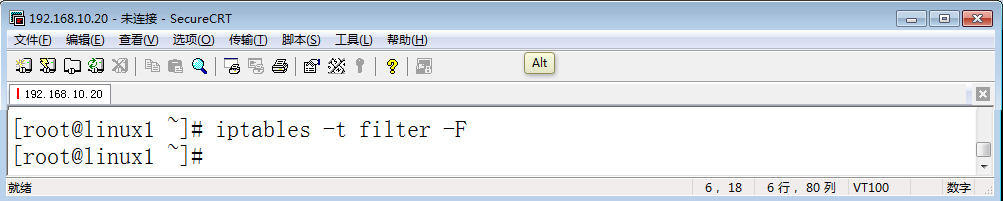
在Linux2 ping Linux1 不通。



### 清除防火墙规则

清除filter表全部链规则

[root@WebServer ~]#iptables –t filter –F

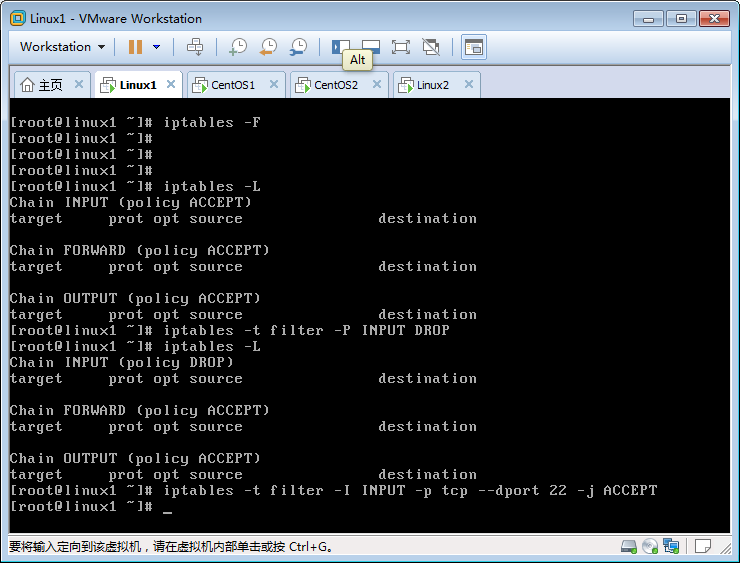


清除filter表INPUT链规则

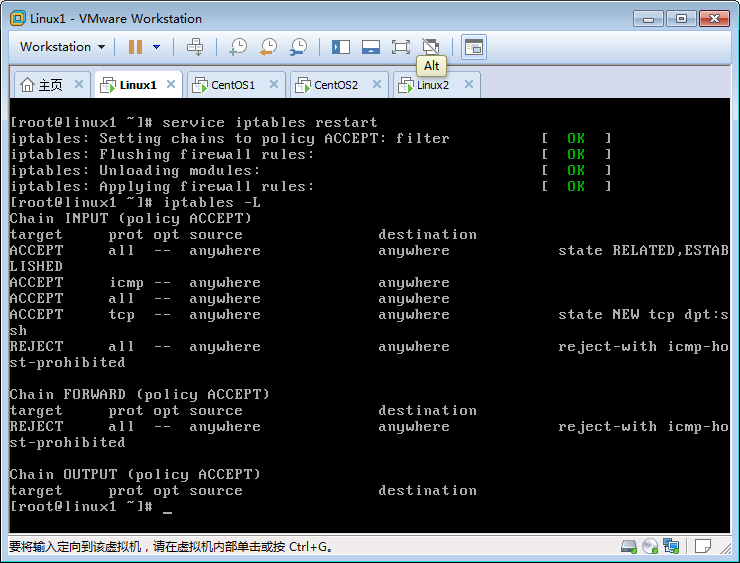
[root@WebServer ~]#iptables –t filter –F INPUT

### 设置Iptables防火墙只打开指定端口

下面的设置清除了iptables filter表的全部规则。只允许ssh能够连接Linux。Linux不能主动发送数据包出去。



### 重启iptables服务还原配置



### 保存还原防火墙设置

Iptables设置存放位置

[root@WebServer ~]# cat /etc/sysconfig/iptables

保存设置

[root@WebServer ~]# /etc/rc.d/init.d/iptables save

iptables：将防火墙规则保存到 /etc/sysconfig/iptables：[确定]

将iptables服务从配置文件读取设置，重启iptables服务即可。

[root@WebServer ~]# service iptables restart

iptables：清除防火墙规则：[确定]

iptables：将链设置为政策 ACCEPT：filter [确定]

iptables：正在卸载模块：[确定]

iptables：应用防火墙规则：[确定]

将设置保存到其他文件

[root@WebServer ~]# iptables-save > iptables.conf1

应用保存的设置

[root@WebServer ~]# iptables-restore < iptables.conf1

## 保护服务器安全

### 常规网络安全设置

示例:

Web服务器对Internet开放TCP 80端口

SSH端口 只允许192.168.10.123访问

默认拒绝所有进入的流量

安装httpd服务

[root@WebServer ~]# yum install httpd

[root@WebServer ~]# chkconfig httpd on

[root@WebServer ~]# service httpd restart

设置防火墙规则

[root@WebServer ~]# iptables -t filter -P INPUT DROP

[root@WebServer ~]# iptbales -F

[root@WebServer ~]#iptables –t filter –I INPUT –p tcp --dport 80 -j AACEPT

[root@WebServer ~]# iptables –t filter –I INPUT –p tcp --dport 22 -s 192.168.10.123/32 -j AACEPT

## 保护子网的安全

示例：

[root@WebServer ~]# iptables -t filter -I FORWARD -p tcp --dport 3389 -d 192.168.10.110/32 -o eth2 -j ACCEPT

[root@WebServer ~]# iptables -t filter -I FORWARD -m state --state ESTABLISHED -j ACCEPT