包过滤型防火墙：

简单包过滤

TCP,IP:根据tcp ip首部数据进行判断

带状态检测的包过滤

NEW,ESTABLISHED,INVALID,RELATED

应用层网关防火墙

应用层：根据数据包传输实际数据进行判断

iptables：防火墙规则生成工具

Netfilter: 网络过滤器, 只是一个Framework 只有规则才过滤

NAT: Network Address Translation

DNAT

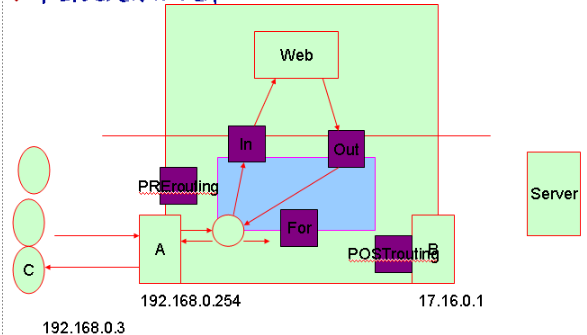
SNAT

过滤功能

地址转换功能

源地址转换在postrouting

目标地址转换在prerouting



filter: INPUT, OUTPUT, FORWARD // PREROUTING, POSTROUTING 不做过滤

nat: PREROUTING, POSTROUTING, OUTPUT

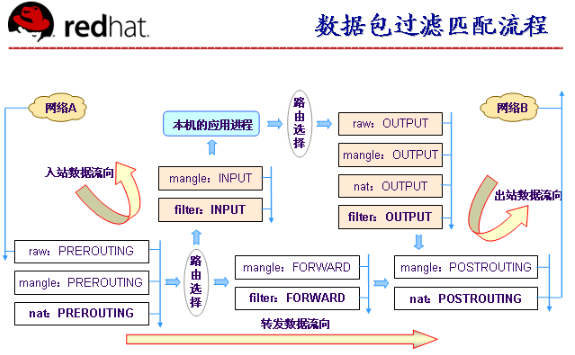
mangle: PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING

数据包首部拆开改改 撕裂 在哪可以撕裂数据包

还有不常用的raw

OUT:三个链都可用，但是如果这样 10个 filter, 5个 nat, 2: mangle

三种规则分类存放



是有优先级的：

raw-->mangle-->nat-->filter

这个门 就叫做：iptables

iptables [-t TABLE指定表] COMMAND CHAIN链 [creteria匹配条件] -j ACTION处理

-t {raw|mangle|nat|filter}, 默认filter

COMMAND:

1)规则管理类：

-A, append最后追加

-I # insert指定插入位置

-D # 删除第几条

-R # 替换某规则

2)链接管理类:

-F, flush, 清空链

-N, new, 新建链

-X, delete, 删除自定义的空链

-E, rename重命名

3)定义默认策略：

-P, policy

4)清空计数器：

-Z, zero

每条规则(包括默认策略)都有两个计数器：

被此规则匹配到的所有数据包的个数；

被此规则匹配到的所有数据包的大小之和；

5)查看类： -L, list

#iptables -L

-n, numeric 数字方式显示 不做主机名解析

-v, verbose

-vv

-vvv

-x, exactly

--line-numbers 行号

#iptables -t mangle -L -n -v

匹配条件：

基本匹配：

-s SOURCE:IP, NETWORK

-d 目标地址

-p {tcp|udp|icmp}协议

-i INTERFACE

-o INTERFACE

扩展匹配：(调用iptables的模块，以便扩展iptables的匹配功能, -m指定调用模块)

隐含扩展 不指定罢了

-p tcp (-m tcp只不过是可以省略了)

--sport PORT 源端口

--dport PORT 目标端口

--tcp-flags tcp的标志位 带两个参数 第一个 第二个参数位

ACK,SYN,RST,FIN(检查的标志位) SYN(此处必须为1) = --syn

含义：新发起的会话

--tcp-flags ACK,SYN,RST,FIN SYN,ACK,RST,FIN

所有包均为一，理论不存在，非法数据包

--sport 22:23

-p UDP

--sport PORT

--dport PORT

-p icmp

--icmp-type

8: echo-request

0: echo-reply

3：ping通的，很多情况(较少运用)

**显式扩展**: 必须使用-m指定的拓展

-m state --state NEW 指定的状态

四种状态：

NEW 新发起的请求syn=1的包

ESTABLISHED syn不等于1 ack=1 fyn不等于1 对服务端只有ESTABLISHED，没有NEW

INVALID

RELATED

-m multiport 合并时用

--source-ports 22,53,80 可取反[!]

--destination-ports

--ports

-m iprange

--src-range

--dst-range

iptables -A INPUT -m iprange --src-range 172.16.100.190-172.16.100.210 -p tcp --dport.,……

limit 速率限定：

-m limit

--limit 1/second

--limit-burst x

实验时注意apache的keepalive

iptables -A INPUT -d 172.16.100.1 -p tcp --dport 80 -m state --state NEW -m limit --limit 1/second --limit-burst 3 -j ACCEPT

-m string 字符串限定

--algo bm|kmp

--string "STRING"

-m time

--timestart 8:00

--timestop 18:00 -j DROP

--days LISTOFDAYS(Mon,Tue,Wed,Thu等)

--datestart date

--datestop date

**-j 处理方法**

ACCEPT

DROP 委婉的拒绝 泥牛入海

REJECT 直接拒绝 有回应的拒绝

SNAT

DNAT --to-destination

REDIRECT

PNAT:

--to-destination 192.168.100.2[:PORT]

LOG:（被处理的规则）

--log-prefix “DNAT for web:” /var/log/messages日志中









1、把默认策略设为DROP

INPUT, OUTPUT

iptables -P CHINE {ACCEPT|DROP}

2、允许本机的ssh服务能被172.16.0.0/16访问；

3、允许本机127.0.0.1到127.0.0.1的所有请求都能正常进行；

4、安装DNS服务，并开放给所有主机使用；

5、安装WEB服务，并开放给非172.16.100.200的所有主机访问；

22端口进来的可以是NEW或者ESTABLISHED

出来的只有ESTABLISHED

服务器端定义iptables规则：

iptables –A INPUT –d 172.16.100.1 –p tcp --dport 22 -m state --state NEW, ESTABLISHED -J ACCEPT(--s 0.0.0.0/任意地址可省略)

iptables –A OUTPUT –d 172.16.100.1 –p tcp --sport 22 -m state --state ESTABLISHED -J ACCEPT

iptables -P INPUT DROP

iptables -P OUTPUT DROP

从127.0.0.1到127.0.0.1的均放行：

iptables -I INPUT 1 -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT

iptables -I OUTPUT 1 -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT

iptables -A INPUT -d 172.16.100.1 -p tcp --dport 80 -m state --state NEW, ESTABLISHED -J ACCEPT

iptables -A OUTPUT -d 172.16.100.1 -p tcp --sport 80 -m state --state ESTABLISHED -J ACCEPT

**规则优化：**

把要求严的放前面

把规则合并：22和80端口合并

顺序影响

**放行DNS：**

iptables -A INPUT -d 172.16.100.1 -p udp --dport 53 -m state --state NEW, ESTABLISHED -J ACCEPT

iptables -A OUTPUT -d 172.16.100.1 -p udp --sport 53 -m state --state ESTABLISHED -J ACCEPT

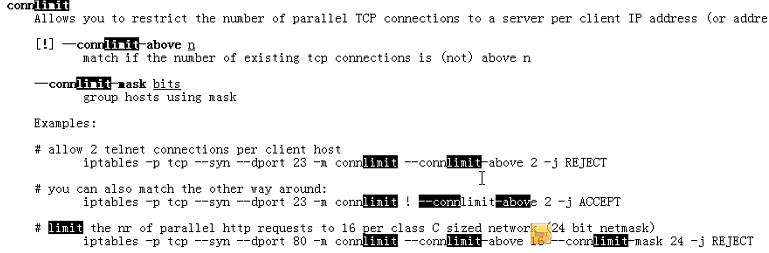
iptables -A INPUT -d 172.16.100.1 -p tcp --dport 53 -m state --state NEW, ESTABLISHED -J ACCEPT

iptables -A OUTPUT -d 172.16.100.1 -p tcp --sport 53 -m state --state ESTABLISHED -J ACCEPT

合并：

iptables -A INPUT 2 -d 172.16.100.1 -p tcp --dport 53 -m state --state NEW, ESTABLISHED -J ACCEPT

iptables -I OUTPUE 1 -m state --state ESTBSLISH…………



**例子：对ssh连接数量进行限制**

iptables -I INPUT 2 -d 172.16.100.1 -p tcp --dport 22 -m state --state NEW -m 

**如何开放本机的ftp：**

FTP在主动模式下：TCP 20 21

被动模式下：TCP的21命令连接端口 数据连接在大于1023的随机端口

那么iptables需要开放大于1023的所有端口，但那就没意义了

其实只需要开放RELATED状态的端口即可

RELATED

开放被动模块下的ftp

iptables直接生效：关机重启不生效，只能写到配置文件中来生效

不是服务

**保存规则的文件：**

/etc/sysconfig/iptables

向iptables脚本提供配置文件的文件：

/etc/sysconfig/ipconfig-config

/etc/rc.d/init.d/iptables

service iptables start:保存规则的的文件 原有规则会被覆盖

service iptables stop:清空

service iptables save:

iptables-save > /etc/sysconfig/iptables.test

**iptables-restore** < /etc/sysconfig/iptables.test 恢复原来的命令

启用iptables的变量：例如两个网卡，颠倒之后就会出现错误或者需要手动修改：

自动生效

#!/bin/bash

INNET=eth0

OUTNET=eth1

iptables -F

iptables -f nat -F

iptables -t mangle -F

iptables -P aINPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

iptables -A INPUT -I lo -j ACCEPT

iptables -A OUTPUT -o lo -j ACCEPT

/etc/rc/local下自动执行

#########################

以上都是主机防火墙

下面是网络防火墙

利用iptables的**recent模块**来抵御DOS攻击

iptables -I INPUT -p tcp --dport 22 -m connlimit --connlimit-above 3 -j DROP

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

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

1.利用connlimit模块将单IP的并发设置为3；会误杀使用NAT上网的用户，可以根据实际情况增大该值；

2.利用recent和state模块限制单IP在300s内只能与本机建立3个新连接。被限制一分钟后即可恢复访问。

下面对最后两句做一个说明：

1.第一句是记录访问tcp 22端口的新连接，记录名称为SSH

--set 记录数据包的来源IP，如果IP已经存在将更新已经存在的条目

2.第三句是指SSH记录中的IP，300s内发起超过3次连接则拒绝此IP的连接。

--update 是指每次建立连接都更新列表；

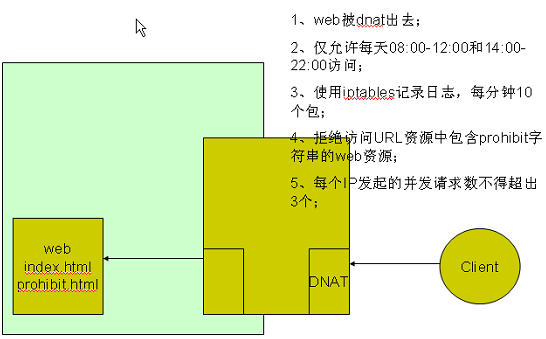
--seconds必须与--rcheck或者--update同时使用

--hitcount必须与--rcheck或者--update同时使用

3.iptables的记录：/proc/net/ipt\_recent/SSH

也可以使用下面的这句记录日志：

iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --name SSH --second 300 --hitcount 3 -j LOG --log-prefix "SSH Attack"



# iptables -A clean\_in -d 255.255.255.255 -p icmp -j DROP

# iptables -A clean\_in -d 172.16.255.255 -p icmp -j DROP

# iptables -A clean\_in -p tcp ! --syn -m state --state NEW -j DROP

# iptables -A clean\_in -p tcp --tcp-flags ALL ALL -j DROP

# iptables -A clean\_in -p tcp --tcp-flags ALL NONE -j DROP

# iptables -A clean\_in -d 172.16.100.1 -j RETURN

-i数据流入的接口

-o数据流出的接口

# iptables -A INPUT -i lo -j ACCEPT

# iptables -A OUTPUT -o lo -j ACCEPT

# iptables -A INPUT -i eth0 -m multiport -p tcp --dports 53,113,135,137,139,445 -j DROP

# iptables -A INPUT -i eth0 -m multiport -p udp --dports 53,113,135,137,139,445 -j DROP

# iptables -A INPUT -i eth0 -p udp --dport 1026 -j DROP

# iptables -A INPUT -i eth0 -m multiport -p tcp --dports 1433,4899 -j DROP

# iptables -A INPUT -p icmp -m limit --limit 10/second -j ACCEPT

# iptables -A INPUT ! --syn -p tcp -m state --state NEW -j DROP

# iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP

# iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP

# iptables -A INPUT -p icmp -d 255.255.255.255 -j DROP

# iptables -A INPUT -p icmp -d 172.16.255.255 -j DROP

--tcp-flags SYN,ACK,RST,FIN SYN

NEW

iptables -A INPUT -p tcp --dport 80 -d 172.16.100.1 -i eth0 -j DROP

-o, OUTPUT, FORWARD,POSTROUTING

-i, INPUT, FORWARD, PREROUTING