第29卷 第10期 计 算 机 工 程 2003年6月
Vol.29 № 10 Computer Engineering June 2003
·安全技术· 文章编号：1000—3428(2003)10—0129—03 文献标识码：A 中图分类号： TP393.08

在Linux下用Iptables建立防火墙的方法

刘 华，颜国正，丁国清
(上海交通大学电子信息学院信息检测与仪器系820研究所，上海 200030)

摘 要：介绍了在Linux操作系统下管理IP包的工具——iptables，详细介绍了iptables的用法，利用iptables建立功能强大的防火墙，并给出了具体的实例。

关键词：Linux；Iptables；防火墙

Creation for Firewall Using Iptables in Linux

LIU Hua，YAN Guozheng，DING Guoqing
(820 Institute of Information Measurement and Instruments of School
of Electronics & Information Technology, Shanghai Jiaotong University,Shanghai 200030)

【Abstract】 This paper presents the tool-iptables for administrating IP packets in Linux. The paper presents the detail usage of the iptables. A firewall can be created using iptables. An example is presented

【Key words】 Linux；Iptables；Firewall

Iptables(NetFilter)应用程序被认为是Linux中实现包过滤功能的第四代应用程序，第一代是Linux内核1.1版本所使用的Alan Cox从BSD Unix中移植过来的ipfw；在Linux2.0版本内核中Jos Vos其他一些程序员对ipfw进行了扩展，并且添加了ipfwadm用户工具；在Linux2.2版内核中，Russell和Michael Neuling做了一些非常重要的改造，也就是该内核中Russell添加了帮助用户控制过滤规则的ipchains工具，现在Russell又完成了其名为NetFilter的内核框架。

NetFilter的目的是为用户提供一个专门用于包过滤的底层结构，并且用户和开发人员可以将其内建在Linux内核中，Iptables是一个内建在NetFilter框架中的模块，它可以让用户访问内核过滤规则和命令。

1 IP包如何穿越过滤器

内核用3条规则来实现对包的过滤，这些规则叫作防火墙链或就叫链。这3条链分别是INPUT、OUTPUT和FORWARD链。内核2.4所带的iptables工具与内核2.0或 2.2ipchains工作完全不同，3条链的安排(见图1)。

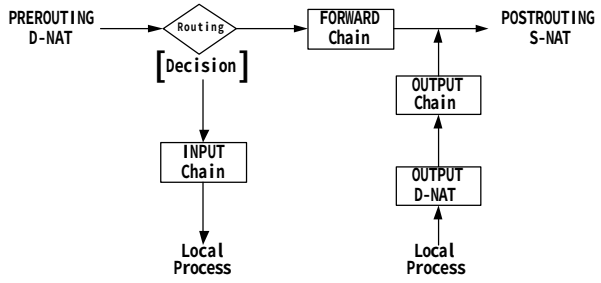

图1 3条链的安排

3个方框代表上面说到的3条链。当一个包到达这个图表的一个方框时，内核检查这个链以决定如何处理这个包，如果链决定丢弃这个包，这个包就会在这里丢弃；但是如果链决定接受这个包，那这个包将继续穿过方框。

链是检查规则的集合，如果包匹配这个规则，那就由这个规则来决定如何处理这个包；如果包不匹配这个规则，内核将检查链中下一个规则，直至最后一个规则，如果仍然没有规则匹配这个包，那内核就用默认的规则来处理这个包；在一个安全的系统，默认的规则应该是丢弃这个包。

(1)当一个包进入时(通过以太网卡进入内核)，内核首先判断这个包的目标地址以决定送到哪一个链进行检查，这个过程就叫“Routing(路由)”；

(2)如果包的目的地址是本计算机，包将在图中向下送到方框INPUT Chain，如果INPUT Chain决定接受这个包，那包将穿越这个链，任何等待这个包的进程将接受这个包。

(3)如果包的目的地址不是本计算机，则内核将转发这个包。IP包转发功能如果没打开，内核将不知道如何转发这个包，这个包将被丢弃；如果包的转发功能打开，这个包将被转发到指向目的地址的另一个网络设备上(如果有另外一个网络设备)，那包将直接转到图中方框FORWARD Chain，如果FORWARD Chain接受这个包，它将被送出去。

(4)最后本地计算机上的程序可以送出IP包，这些包立即被送到OUTPUT Chain，如果OUTPUT Chain接受这些包，那这些包将继续被送到它们的目的地。

2 Iptable的基本语法

Iptables在Linux内核中用于建立、维护和检查IP包过滤规则的表的集合。可以定义几种不同的表，每种表包含许多内建链也可能包含用户定义的链。

每条链是可以匹配一系列包的规则的集合。每个规则指明如何处理匹配的包。处理这个包的方法就叫作目标，在同一表中目标可以跳到用户定义的链上。

(1)TARGETS(目标)

防火墙规则指明处理包的标准和处理包的目标，如果包不匹配，就检查链中的下一条规则；如果下一条规则匹配，那处理包的目标就是由下一条规则的目标决定，这个目标可以是用户定义链的名字或特定值：ACCEPT、DDROP、QUEUE或RETURN中的一个。

作者简介：刘 华(1974－)，男，博士生，主研方向为网络、机器人；颜国正，教授、博导；丁国清，副教授

收稿日期：2002-06-14

ACCEPT意思是让包通过规则；DROP意思是丢弃包；QUEUE意思是将包送到用户的空间(如果内核支持)；RETURN意思是停止匹配当前规则而从前一条链的第二条规则重新开始匹配；如果匹配检查到了内建链的最后或匹配的内建链的目标值是RETURN，则包的目标值由链默认的规则来决定。

(2)TABLES(表)

可以使用3个独立表(使用哪一个表取决于内核配置选项和加载了哪些模块)。

-t, --table

这个选项指明包匹配命令将要起作用的表。如果内核用自动模块加载方式配置，那将自动为表加载合适的缺少的模块。表可以取下列值中的一个。

filter 这是默认的表。它包含内建链INPUT(用于包进入内核)、FORWARD(用于包通过内核)和OUTPUT(用于本地计算机产生的包)。

nat 遇到一个产生新的连接的包时需要检查的表。它包含三条内建的链：PREROUTING(用于包一进入时改变包)、OUTPUT(用于在路由前改变本地计算机产生的包)和POSTROUTING(用于送出包时改变包)。

mangle 这个表用于指定包的改变动作。有两个内建链：PREROUTING(用于在路由前改变进入的包)和OUTPUT(用于在路由前改变由本地计算机产生的包)。mangle 这个表用于指定包的改变动作。它有两条内建链：PREROUTING(在路由前改变进入的包)和OUTPUT(在路由前改变本地产生的包)。

Iptables还有分成不同类别的选项命令选项、参数选项和其它选项。

(1)命令选项　用于指明要进行的特定的动作。

```
iptables -[ADC] chain rule-specification [options]
iptables -[RI] chain rulenum rule-specification [options]
iptables -D chain rulenum [options]
iptables -[LFZ] [chain] [options]
iptables -[NX] chain
iptables -P chain target [options]
iptables -E old-chain-name new-chain-name
```

1)增加一个规则到链中(-A)；

2)在链中的适当位置删除一个规则(-D)；

3)在链的适当位置替换一个规则(-R)；

4)在链的适当位置插入一个新的规则(-I)；

5)产生一个新的链(-N)；

6)删除一个空的链(-X)；

7)改变一个内建链的规则(-P)；

8)列出一个链的规则(-L)；

9)清空一个链里所有规则(-F)；

10)清零一个链里所有规则的包和字节计数器(-Z)。

(2)参数选项　参数选项组成一个规则的说明(表在增加、删除、插入和替换命令选项之中)。

-p, --protocol [!]protocol 协议说明选项

-s, --soruce [!] address[/mask] 源地址说明选项

-d, --destination [!] address[/mask] 源地址说明选项

-j --jump target 指明规则的目标

-i --in-interface [!] [name] 指明进入界面的值

-o, --out-interface [!] [name] 指明送出界面的值

-c,--set-counters PKTS BYTES 管理员的初始包和字节计数器

(3)其它选项　其它选项里包含有匹配扩展、目标扩展和附加选项。

1)匹配扩展

Iptables使用扩展包匹配模块，可以通过两种方法来加载扩展包匹配模块：间接加载，当-p或-protocol指定后扩展包匹配模块自动加载，或用-m带上扩展区匹配模块名，有了这些扩展包匹配模块后可以附加各种命令。

2)目标扩展

Iptables可以使用标准的目标扩展模块。其中重要的目标扩展模块有：

REJECT

目标值用于返回错误的包，这个目标类似于DROP，这个目标只在INPUT、FORWARD和OUTPUT链和只是从用户定义的调用这些目标值的链里有效。

SNAT

这个目标只在nat 表的POSTROUTING链里有效，它指明包的源地址将被修改(在这个连接里的所有的包将被修改)，并且将停止规则的检查。它只有一个选项：

--to-source <ipaddr>[-<ipaddr>][:port-port]

这个选项将指明一个新的IP地址、一个IP地址范围端口是可选项，如果没有指明端口，那在512以下的源端口将被映射到端口在512以下的其它端口，在512～1023端口之间的将被映射到1024端口以下，其它的端口将被映射到1024端口之上有可能不改变端口。

DNAT

这个目标只在nat表的PREROUTING链、OUTPUT链和用户定义的引用PEROUTING和OUTPUT链中有效。它指明包被修改目的地址(在这个连接中的所有包都被修改)，并且停止检查其它规则，它有一个选项

--to-destination <ipaddr>[-<ipaddr>][:port-port]

这个选项指明一个新的目的地址，端口选项是可选的，如果不指定端口，那端口将不会被修改。

MASQUERADE

这个目标只在nat表的POSTROUTING链中有效，它只用于动态分配的IP地址(括号)连接上，如果有一个静态IP地址，应该使用SNAT目标，Masquerading等同于指定一个包将通过的界面的IP映射。但是仍然有当界面停止时连接将忘记的特点，它有一个选项：

--to-ports <port>[-<port>]

这指明将用到的源端口的范围，屏蔽默认的SNAT源端口选择。

REDIRECT

这个目标只在nat表的PREROUTING链、OUTPUT链和用户引用这两个链效，它改变了目的IP地址来发送包到机器本身(本地产生的包将被映射到127.0.0.1地址),有一个选项：

--to-ports <port>[-<port>]

这个选项指明要用到的目标端口或端口范围，没有这个选项，端口将不会被改变。

3)附加选项

3 单个链的用法

计算机启动后，在任何iptables命令运行之前(一些版本中的iptables运行于初始化的脚本里)，在内建("INPUT"、"FORWARD"和"OURPUT")没有任何的规则，在INPUT和

OUTPUT内建链的规则规定是ACCEPT，而FORWARD链里的规定是DROP(可以通过在iptables模块里提供"forward=1"来更改默认的规定DROP)。

单个链是包过滤和管理规则的最基本用法。通常可以用增加(-A)和删除(-D)命令来使用单个链，其它(插入 -I和替换 -R)用于这概念的简单扩展。根据Iptables的用法可以定制出合适的IP包的检查规则和目标，以构筑所需要的防火墙。

4 利用Iptables建立防火墙

通过一个局域网防火墙设置来介绍如何利用Iptables定制防火墙。

4.1局域网结构

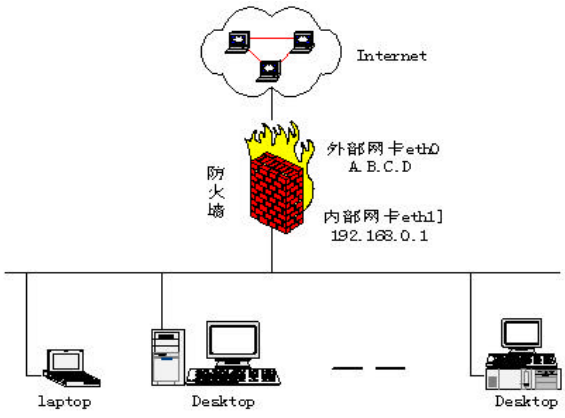

图2 局域网设置

局域的拓扑结构如图2，整个局域网对外只有一个IP地址：A.B.C.D，内部有若干台计算机，对外访问Internet网络只能通过IP地址为A.B.C.D的计算机，为了实现局域网所有计算机访问外部Internet，需将IP地址为A.B.C.D的计算机设置为代理服务器，局域网内部计算机将这台计算机设置为网关，通过这台计算机访问外部Internet，内部计算机可以访问外部网络，但同时要采取措施防止外部计算机直接访问内部网络，以保证局域网运行的安全，为此要在IP地址为A.B.C.D的计算机上定制一个合适的防火墙，在这台计算机上有两张以上的网卡：一张对内网卡(IP地址：192.168.0.1)；另外一张对外网卡(IP地址为: A.B.C.D)。局域网所有计算机通过HUB与服务器的对内网卡(IP地址：192.168.0.1)连接起来，对外网卡直接连接到外部Internet网络上，确保所有硬件连接正确后，就可以进行防火墙设置。

4.2 编写防火墙脚本

在Linux操作系统下确认两张网卡完全可以正常工作后，就可以编写防火墙来定制防火墙，可以将防火墙脚本放置在/etc/rc.local系统的启动脚本里，系统启动后就可以自动执行脚本，编辑/etc/rc.local文件。

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
#The following lines are the scripts for the firewall!
echo 1>/proc/sys/net/ipv4/ip_forward     //打开内核IP包转发功能
iptables -F INPUT                //清除默认的INPUT内建链规则
iptables -F FORWARD      //清除默认的FORWARD内建链规则
iptables -F POSTROUTING -t nat      // 清除默认的POSTROUTING //内建链规则
iptables -P FORWARD DROP
 //设置默认的FORWARD规则为丢弃所有的包DROP，当一个 //包转发不能应用到任何一条转发规则上则应用默认规则
iptables -A FORWARD -s 192.168.0.0/24 -j ACCEPT
//对于来自192.168.0.0/24内部局域网来的所有包实施转发
iptables -A FORWARD  -i eth1 -m state --state ESTABLISHED, RELATED -j ACCEPT //利用了有状态的能力，只要是对先前从防火 //墙外部接口出去的请求包的回复都允许。ESTABLISHED指TCP连 //接，RELATED指象主动FTP、ICMP ping 请求等。当回复包到达  //时，实际上是检查文件/proc/net/ip_conntrack看是否在里面，如果  //在表中，则不检查任何链，包允许通过。
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24 -j  //MASQUERADE //打开内核包的IP伪装功能，从eth1出去的包被重  //写源地址后伪装出去，是源地址SNAT的选择特例。需注意的是选 //项为-o指从一个接口出去的包，-i指从一个接口进来的包。
#port:80 for http
iptables -A INPUT -p tcp -i eth0 -d 192.168.0.1 --destination-port 80 -j ACCEPT
iptables -A INPUT -p tcp -i eth1 -d 202.120.16.87 -destination-port 80 -j ACCEPT
#port:23 for telnet
iptables -A INPUT -p tcp -i eth0 -d 192.168.0.1 --destination-port 23 -j ACCEPT
iptables -A INPUT -p tcp -i eth1 -d 202.120.16.87 -destination-port 23 -j ACCEPT
#port:21 for ftp
iptables -A INPUT -p TCP -i eth0 -d 192.168.0.1 -destination-port 21 -j ACCEPT
iptables -A INPUT -p TCP -i eth1 -d 202.120.16.87 -destination-port 21 -j ACCEPT
#Reject any machine visit noe-http, non-ftp, non-telnet
iptables -A INPUT -p TCP -j DROP
iptables -A INPUT -p UDP -syn -j DROP
```

将上面的防火墙脚本加入/etc/rc.local文件之后保存文件并重新启动系统，则利用iptables 就实现了局域网服务器和一个较好的防火墙的结合。

5 结论

利用Linux操作系统下的Iptables可以非常方便地编写功能完全的防火墙，Iptables模块化结构，方便管理、易于排除错误，是一种功能强大、实用的防火墙工具。

参考文献

1 杨 辉防火墙北京国防工业出版社,2001

2 黄凌云防火墙安全技术的设计与实现上海:上海交通大学出版社, 1999

3 Russell R.Linux Iptables HOWTO.http://www.linuxguruz.org/iptables/

4 吴阿亭.Linux 2.4 内核中的Iptables 新增功能指南.http://www.ccw.com.cn/htm/app/salon/01_10_18_2.asp

5 Linux 2.4 Netfilter FAQ.http://netfilter.samba.org/netfilter-faq.html

6 Linux 2.4 Packet Filtering HOWTO.http://netfilter.samba.org/unreliab- leguides/packet-filtering-HOWTO/index.html

7 Linux 2.4. NAT HOWTO.http://netfilter.samba.org/unreliable-guides/NAT-HOWTO/index.html

8 The Journey of a Packet Through the Linux 2.4 Network Stack.http: //www.gnumonks.org/ftp/pub/doc/packet-journey-2.4.html