Linux 防火墙 iptables 学习笔记（五）

本文旨在用为公司做防火墙的实例，让大家对 Linux+iptables 做防火墙的安装和配置有一个大致的了解，希望能起到抛砖引玉的作用。

系统环境与网络规化

先了解一下公司的环境，公司利用 2MADSL 专线上网，电信分配公用 IP 为 218.4.62.12/29,网关为 218.4.62.13,公司有电脑五十多台，使用 DHCP，IP 是 192.168.2.XXX，DHCPServer 建在 iptablesServer 上；另公司有一电脑培训中心，使用指定固定 IP，IP 为 192.168.20.XXX，为了更加快速的浏览网页，我们架了一台 SquidServer，所有电脑通过 SquidServer 浏览网页，公司还另有一台 WEBServer+MailServer+FtpServer。其 IP 为 218.4.62.18。以上电脑和服务器要求全架在防火墙内。我们规化如下：

IptablesServer 上有三块网卡,eth0 上加有二个 IP，218.4.62.14 和 218.4.62.18。

其中 218.4.62.14 为共享上网，218.4.62.18 为 WEBServer 专用，Eth1 的 IP 为 192..168.2.9；为了使培训中心 PC 与公司 PC 之间互不访问，所以直接从 IptablesServer 接到 Switch-B,eth2 接至 Switch-A，连接培训中心 PC 和 SquidServer,WebServer。

网络规化好了后，就开始装服务器了，IptablesServer 用的系统为 RedhatLinuxV7.3。在装服务器时要注意选上防火墙的安装包。

IPTABLES 基础

Iptables 语法：

```
Iptables[-tTABLE]ACTION[PATTERN][-jTARGET]
```

TABLE:

有 filter,nat,mangle；若无指定，预设为 filtertable.

ACTION(对 Chains 执行的动作):

ACTION 说明

```
-LChain 显示 Chain 中的所有规则

-AChain 对 Chain 新增一条规则

-DChain 删除 Chain 中的一条规则

-IChain 在 Chain 中插入一条规则

-RChain 替换 Chain 中的某一条规则

-PChain 对 Chain 设定的预设的 Policy

-FChain 清除 Chain 中的所有规则

-NChain 自订一个 Chain
```

```
-X 清除所有的自订 Chain
```

CHAINS:

Iptables 有五条默认的 Chains(规则链)，如下表：

```
Chains 发生的时机

PREROUTING 数据包进入本机后，进入 RouteTable 前

INPUT 数据包通过 RouteTable 后，目地为本机

OUTPUT 由本机发出，进入 RouteTable 前

FORWARD 通过 RouteTable 后，目地不是本机时

POSTROUTING 通过 RouteTable 后，送到网卡前

PATTERN(设定条件部份):
```

参数内容说明

```
-pProtocol 通讯协议，如 tcp,udp,icmp,all 等。。。

-sAddress 指定的 SourceAddress 为 Address

-dAddress 指定的 DestinationAddress 为 Address

-IInterface 指定数据包进入的网卡

-oInterface 指定数据包输出的网卡

-mMatch 指定高级选项，如 mac,state,multiport 等。。。
```

TARGET(常用的动作)：

TARGET 说明

ACCEPT 让这个数据包通过

DROP 丢弃数据包

RETURN 不作对比直接返回

QUEUE 传给 User-Space 的应用软件处理这个数据包

SNATnat 专用：转译来源地址

DNATnat 专用：转译目地地址

MASQUERADEnat 专用：转译来源地址成为 NIC 的 MAC

REDIRECTnat 专用：转送到本机的某个 PORT

用/etc/rc.d/init.d/iptablessave 可在/etc/sysconfig/中产生一 iptables 文件，大家可以看到，它有三个*号开始的行，其每一个以*号开始的行对应一个 table，以 COMMIT 表示此 table 的结束。可将要定的规则加入到对应的 table 中，如下：

```
[root@jiaoyuanginit.d]#./iptablessaveSavingcurrentrulesto/etc/sysconfig/iptables:[OK][root@jiaoyuanginit.d]#cat/etc/sysconfig/iptables
```

```
#Generatedbyiptables-savev1.2.4onSatSep2816:51:222002
*mangle
:PREROUTINGACCEPT[61522:8074850]
:OUTPUTACCEPT[1079:79301]
COMMIT
#CompletedonSatSep2816:51:222002
#Generatedbyiptables-savev1.2.4onSatSep2816:51:222002
*nat
:PREROUTINGACCEPT[31850:5091703]
:POSTROUTINGACCEPT[20:1240]
:OUTPUTACCEPT[12:776]
COMMIT
#CompletedonSatSep2816:51:222002
#Generatedbyiptables-savev1.2.4onSatSep2816:51:222002
*filter
:INPUTACCEPT[61444:8070296]
:FORWARDACCEPT[34:1984]
:OUTPUTACCEPT[1079:79301]
COMMIT
```

安装并启动 IPTABLES

在安装 RedHatLinuxV7.3 后，iptables 就已经被安装了，但默认启动的是 ipchains。你在安装时所定义的一些规则也在/etc/sysconfig/ipchains 中被定义。我们需要将其停止，才能启动 iptables(注意：虽然不停止 ipchains 也可以启动 iptables ,但这时 iptables 并没有真正的起作用。Ipchains 和 iptables 是两个防火墙，你只能选择一个)。

```
serviceipchainsstop(停止 ipchains)

chkconfig--level2345ipchainsoff(使 ipchains 系统启动时不自动启动)

chkconfig--level2345iptableson(使 iptables 在系统启动时自动启动)

vi/etc/rc.d/rc.local(编辑 rc.local，将下面四行加到最后)

ifconfigeth0add218.4.62.18netmask255.255.255.248

modprobeip_conntrack_ftp

modprobeip_nat_ftp

echo"1">/proc/sys/net/ipv4/ip_forward
```

（第一行是在 eth0 上再加一个 IP：218.4.62.18，因在安装时只能设一个 IP：218.4.62.14。Ip_conntrack_ftp 和 ip_nat_ftp 为 iptables 运得必须的两个模块；最后一行为使开启服务器 IP 转发功能。）

(如果你将 iptables 的模块加到了内核中，以上第二，三行可省略。)

配置 DHCPServer ,以便让公司 PC 自动获得 IP 和网关 ,网关为 192.168.2.9。具体的方法请参见相关资料，本文不作详述。

reboot

重新启动服务器后，Iptables 就已经开始运行了。

配置 IPTABLES

对 iptables 有了一个基本的了解后，我们就可以来配置我们的服务器了。首先要发布我们的 WEBServer,将以下二行加入/etc/sysconfig/iptables 中的 nattable 内：

```
-APREROUTING-d218.4.62.18-jDNAT--to-destination192.168.20.254


-APOSTROUTING-s192.168.2.254-jSNAT--to-source218.4.62.18
```

第一行为将至服务器的所有目地地址为 218.4.62.18 的包都 NAT 为 192.168.2.254 ,第二行为将至服务器的所有源地址为 192.168.2.254 的包为 NAT 到 218.4.62.18。请把 WEBServer 的网关设为 192.168.20.9。

下面我们将所有从服务器共享出去的包都 SNAT 为 218.4.62.14,就可完成共享上网的功能了：

```
-APOSTROUTING-s192.168.0.0/16-jSNAT--to-source218.4.62.14
```

将下面的规则加入到/etc/sysconfig/iptables 中的 filtertables 内：

```
    -AINPUT-picmp-micmp--icmp-type8-mlimit--limit6/min--limit-burs
t2-jACCEPT

    -AINPUT-picmp-micmp--icmp-type8-jREJECT--reject-withicmp-port-
unreachable
```

以上两行是为了防止 Dos 攻击做的一个简单的处理 ,大家对于各种攻击可做出相应的处理。

```
    -AINPUT-ieth0-mstate-stateESTABLISHED,RELATED-jACCEPT-AINPUT-i
eth0-jDROP
```

以上两行是做了一个 INPUT 状态防火墙的处理 ,其主要作用为防止外部的连接和攻击，因其接受 ESTABLISHED,RELATED 状态(一个包分为 NEW，ESTABLISHED,RELATED ,INVALID 四种状态)的包 ,故又不妨碍从本机出去的连接。

由于并不是所有的电脑都可以上网，所以还要对共享上网的电脑做一个限制：

IP 限制：

```
    -AFORWARD-s192.168.2.0/29-pudp-mmultiport-port53-jACCEPT

    -AFORWARD-s192.168.2.0/29-ptcp-mmultiport-port3128,110,25-jACC
EPT

    -AFORWARD-s192.168.20.253-jACCEPT
```

充许 192.168.2.0~192.168.2.7 和 192.168.20.253(squidserver)的电脑可上网和发邮件。3128 是 squidserver 的 proxyport。我们用它去共享上网，110 为 pop3,25 为 smtp。Udp 的 53 为 DNS 所要的 port。不过由于使用的是 DHCP，可能每次得到的 IP 都不一样，所以我们就要用下面一种 MAC 限制的方法了。

MAC 限制：

```
-AFORWARD-mmac--macXX:XX:XX:XX:XX:XX-pudp-mmultiport-port53-jACCEPT

-AFORWARD-mmac--macXX:XX:XX:XX:XX:XX-ptcp-mmultiport-port3128,110,25-jACCEPT
```

如上就可通过网卡来控制上网了，但现在电脑高手多多，改一个 MAC 的地址好像也不是什么难事了，怎么办呢？那就用我们的第三种方法吧。

MAC+IP 限制：

更改/etc/dhcpd.conf,如果 MAC 与 IP 绑定：

```
subnet192.168.2.0

netmask255.255.255.0{

range192.168.2.30192.168.2.230;

optionbroadcast-address192.168.2.255;

optionrouters192.168.2.9;

optiondomain-name-servers212.132.16.163;
```

```
    hostmeeting-room{

    hardwareethernet00:50:ba:c8:4b:3a;

    fixed-address192.168.2.35;

    }}
```

我们的 Iptables 改为：0

```
    -AFORWARD-s192.168.2.35-mmac--macXX:XX:XX:XX:XX:XX-pudp-mmulti
port-port53-jACCEPT

    -AFORWARD-s192.168.2.35-mmac--macXX:XX:XX:XX:XX:XX-ptcp-mmulti
port-port3128,110,25-jACCEPT
```

这样做之后，高手也无能为力了，不过公司有位 MM 是兄台的 GF，上班的时候想和她聊聊天，培养培养感情；怎么办呢？我们知道 QQ 用的是 udp 的 4000 端口，如占用则 4002,4003。。。那么就如下了：

```
    -AFORWARD-s192.168.2.35-mmac--macXX:XX:XX:XX:XX:XX-pudp-mmulti
port-port53,4000,4001,4002,4003,4004,4005-jACCEPT

    -AFORWARD-s192.168.2.35-mmac--macXX:XX:XX:XX:XX:XX-ptcp-mmulti
port-port3128,110,25-jACCEPT
```

最后加一句：

```
    -AFORWARD-s192.168.0.0/16-jDROP
```

由于前面应该开的都开了，所以最后全部禁止。呵呵，到此大功告成。

总结

世界上没有绝对安全的防火墙，安全永远是相对的。配置 iptables 的思路是先 ACCEPT 再 DROP。共享上网的办法还有一个就是用 iptablesserver 的 Owner，但由于 linux 没有像 win2k 那样的验证模式，在验证 owner 时有些困难。本人正在测试，但目前还没有比较好的解决办法，哪位兄弟搞定的话请 Mail 小弟，小弟将不胜感激。值得注意的是在做 NAT 时，客户端的网关一定要是 iptagles 的 IP。