使用 Linux 构筑企业自己的防火墙

北京华夏证券有限公司(100027) 邱 承

摘 要：分析了在配置 Linux 防火墙时，如何明确自己的需求，并在此基础上如何制定规则，给出了配置防火墙的具体方法。

关键词：Linux 防火墙 IP 伪装 端口转发

1 基本概念

1.1 IP 伪装(IP MASQUERADE)

IP 伪装允许很多机器通过 IP 伪装防火墙隐蔽地访问 Internet。对于 Internet 上的机器来说，所有这些对外的访问都像是从防火墙上发出的，加上它的其它附加功能，IP 伪装提供了一个建立非常安全的网络环境的基础。攻破一个配置良好的 IP 伪装防火墙是极其困难的。

IP 伪装与很多商业防火墙和路由器支持的 NAT 功能有些相似，是在 Linux 内核中支持的一项网络功能。例如，如果一个 Linux 主机与 Internet 连接，那么 IP 伪装使它与该主机相连的内部主机也可以访问 Internet 而不需要合法的 Internet 地址。

1.2 端口转发(Port Forward)

端口转发自动把发向某一主机某一端口的数据包转发到预先定义好的某主机的某一端口。它也是 Linux 内核支持的功能，与 IP 伪装配合使用可大大增强 Linux 防火墙的灵活性和安全性。

2 如何配置 Linux 防火墙

2.1 明确自己的需求

安装防火墙首先要弄清自己的需求，可以从下面二个方面考虑：

(1)对防火墙外的用户提供什么？

(2)防火墙内的用户需要什么？

如对外站点提供 WWW、邮件、域名解析服务，对内满足客户通过防火墙访问 Internet 的需求，包括：浏览 Web、收发邮件、使用 FTP 等等。制定安全规则的目标就是要最大限度地满足内部网用户的需求和最大限度地限制外部网用户对内部网的行动。一个极端安全的情况就是：内部网用户可随意访问外部网而外部网的用户不能访问内部网。

2.2 制定安全规则

明确了需求，就可以开始制定安全规则了。这里以图 1 所示为例进行说明。图 1 是一个对外提供 Web 服务并自己解析域名的网站典例，它设计方案灵活，其它复杂的站点和服务都可参照图 1。

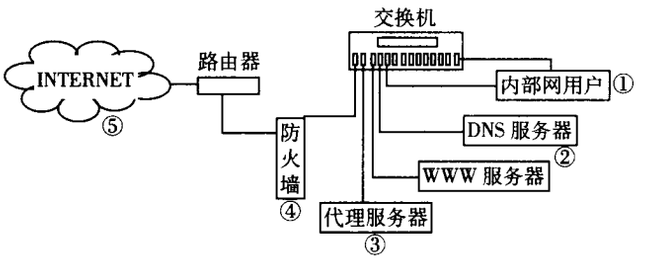

图 1 对外提供 Web 服务并自己解析域名的网站

应用 Linux 制定安全规则以实现：(1)隐藏所有主机，只暴露对外服务主机的对外服务端口，而且外部用户认为这些端口是 Linux 防火墙上的端口。(2)在内部只允许代理服务器 1 台主机自由对外访问，其它主机必须通过代理服务器来实现对外访问。也就是说，从外向里访问网站的 Web 和 DNS 服务，如果满足规定的条件，Linux 防火墙才能放行，走的路线是 5→4→2。而从内部向外的访问，只能经过代理服务器作一级代理再发出，即路线为 1→3→4→5，1→3→5 则走不通。这样做的好处是：(1)外部主机只能接触到内部有限的几个端口，安全性很高。(2)内部主机对外访问又多了代理服务器这一道屏障，加大了安全性。(3)不必为每台主机设置安全规则，减轻了配置负担和出错机会。这是一个典型的例子，虽然对外服务很少，可其它的服务都可照此办理。在这个例子中只需要 2 个 IP 地址，让 DNS 服务器单独使用 1 个 IP，而其它的服务公用 1 个 IP 地址。这样需要在防火墙的外部网卡上设置 2 个地址，假如它们是 202.166.66.1 和 202.166.66.2，那么使用命令：

ifconfig 网卡名称:202.166.66.1 netmask 子网掩码

ifconfig 网卡名称:0 202.166.66.2 netmask 子网掩码

假设这样分配内部网地址：

防火墙	10.0.0.1
DNS 服务器	10.0.0.2

代理服务器　　10.0.0.3

WWW 服务器　　10.0.0.4

那么 Internet 地址和内部网地址的对应关系如表 1 所示。

表 1　地址、端口分配表

地址对应	端口号	协议名称	数据包类型	方向
202.166.66.1↔10.0.0.2	53	DNS	TCP+UDP	进+出
202.166.66.2↔10.0.0.4	80	HTTP	TCP	进
10.0.0.3	所有	所有	所有	出

现在可以按照表 1 来配置防火墙了。

2.3 配置防火墙

在配置防火墙之前，必须具备 KERNEL 2.2.X 和 IPCHAINS1.3.X，并且仔细阅读 KERNEL-HOWTO、IPCHAINS-HOWTO 和 IP MASQUERADE MINI HOWTO(这里由于篇幅所限，不详细描述如何安装和配置这些软件，这些在上面所列的文档里都有详细的描述)。

下面给出防火墙的配置源程序。

下面所编写的工具是用来帮助管理员在防火墙上增加 1 个端口转发的链路，取名为“Enable”，其程序的代码如下：

```
#!/bin/bash
# file to enable a hole in the firewall
universe="0.0.0.0/0"
chainname=$1
internet_addr=$2
internet_port=$3
intranet_addr=$4
intranet_port=$5
protcp=$6
proudp=$7
out_interface=eth0
in_interface=eth1
echo chainname=$chainname
echo "Adding $internet_addr $internet_port<-> $intranet_
                    addr $intranet_port.."

#adding a new chain
/sbin/ipchains -N $chainname
/sbin/ipchains -F $chainname
#adding new chain's rules
if [ $protcp=tcp ];then
    /sbin/ipchains -A $chainname -j ACCEPT -p
tcp -s $universe 1024:-d $internet_addr/32 $internet_port
    /sbin/ipchains -A $chainname -j ACCEPT -p
tcp -s $universe 1024:-d $intranet_addr/32 $intranet_port
    /sbin/ipchains -A $chainname -j ACCEPT -p
tcp -s $internet_addr/32 $internet_port -d $universe 1024:
    /sbin/ipchains -A $chainname -j ACCEPT -p
tcp -s $intranet_addr/32 $intranet_port -d $universe 1024:
    /sbin/ipchains -A forward -b -p tcp -s $universe
        -d $intranet_addr $intranet_port -j MASQ
    /usr/sbin/ipmasqadm portfw -a -P tcp -L $internet_
        addr $internet_port -R $intranet_addr $intranet_
        port
    echo                    TCP OK.
    fi
    if[ $proudp=udp ];then
    /sbin/ipchains -A $chainname -j ACCEPT -b -p
udp -s $universe 1024: -d $internet_addr/32 $inter-
net_port
    /sbin/ipchains -A $chainname -j ACCEPT -b -p
udp -s $universe 1024: -d $intranet_addr/32 $in-
tranet_port
    /sbin/ipchains -A forward -b -p udp -s $universe
-d $intranet_addr $intranet_port -j MASQ
    /usr/sbin/ipmasqadm portfw -a -P udp -L $inter-
net_addr $internet_port -R $intranet_addr $intranet_port
    echo                    UDP OK.
    fi
    #connect to input and output chains
    /sbin/ipchains -A input -j $chainname
    /sbin/ipchains -A output -j $chainname
```

这个程序的用法如下：

Enable 链名 外部主机地址范围 外部主机端口范围 对应的内部主机地址 对应的内部主机端口 tcp [udp]

下面是利用这个小工具编写的一个防火墙的配置源程序：

```
#!/bin/bash
# 为外部网卡加上双地址
/sbin/ifconfig eth1 202.166.66.1 netmask 255.255.
                255.0 broadcast 202.166.66.255
/sbin/ifconfig eth1:0 202.166.66.2 netmask 255.255.
                255.0 broadcast 202.166.66.255
internal_interface="eth0"       # 内部网卡名
loopback="lo"                   #Loopback 名
internal_ip="10.0.0.1"          # 内部网卡的 IP
internal_net="10.0.0.0/24"      # 内部网的 IP 范围
external_interface="eth1"       # 外部网卡名
external_ip="202.106.75.97"     # 定义拥有的 Internet
                                # 合法 IP 地址
external_net="202.106.75.0/24"
universe="0.0.0.0"  # 所有的 IP
primarydns="202.106.75.97"  # 内部 DNS 的对内对外
                            #IP 地址
protfwdns="10.101.1.225"
internet_web="202.106.75.98"  #Web 的对内对外 IP
```

```
                                    #地址
portfwweb="10.101.1.229"
dhcp_server="10.101.3.229"  #内部的 DHCP 服务器
                                   #地址
proxy="10.101.1.220"  #代理服务器的地址
#--------------------------------------
/sbin/ipchains -M -S 7200 600 600  #设定 MASQ
                                    #Timeouts 值
echo "Loading MASQ modules.."  #调用 Masq 的附
                                   #加模块
/sbin/modprobe ip_masq_cuseeme
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_irc
/sbin/modprobe ip_masq_quake
/sbin/modprobe ip_masq_vdolive
/sbin/modprobe ip_masq_radio
#设置 telnet,ftp,ftp-data 的 TOS 值
/sbin/ipchains -A output -p tcp -d 0.0.0.0/0 www
                                    -t 0x01 0x10
/sbin/ipchains -A output -p tcp -d 0.0.0.0/0
                                 telnet -t 0x01 0x10
/sbin/ipchains -A output -p tcp -d 0.0.0.0/0 ftp
                                    -t 0x01 0x10
/sbin/ipchains -A output -p tcp -d 0.0.0.0/0
                                ftp-data -t 0x01 0x08
/sbin/ipchains -A output -p tcp -d 0.0.0.0/0 nntp
                                    -t 0x01 0x02
/sbin/ipchains -A output -p tcp -d 0.0.0.0/0
                                  pop-3 -t 0x01 0x02
#设置所有的链路规则策略为 REJECT 并删除所有的旧
#规则。
/sbin/ipchains -F
/sbin/ipchains -P input REJECT
/sbin/ipchains -P output REJECT
/sbin/ipchains -P forward REJECT
#对来自内部网卡、内部网的机器之间的数据交换一律
#放行。
/sbin/ipchains -A input -j ACCEPT -i $inter-
nal_interface -s $internal_net -d $internal_net
/sbin/ipchains -A output -j ACCEPT -i $inter-
nal_interface -s $internal_net -d $internal_net
/sbin/ipchains -A forward -j ACCEPT -i $inter-
nal_interface -s $internal_net -d $internal_net
#对来自外部网卡,从自己的合法 IP 到其他 Internet 地
#址的数据包给予放行。
/sbin/ipchains -A input -j ACCEPT -b -i $exter-
nal_interface -s $external_net -d ! $internal_net
/sbin/ipchains -A output -j ACCEPT -b -i $ex-
ternal_interface -s $external_net -d ! $internal_net

#防止外部主机假冒内部主机进入内部网
/sbin/ipchains -A input -j REJECT -i $exter-
nal_interface -s $internal_net -d $universe/0 -l

#允许 loopback
/sbin/ipchains -A input -j ACCEPT -i $loopback
-s $universe/0 -d $universe/0
#允许 icmp 访问
/sbin/ipchains -A input -j ACCEPT -b -p icmp
-s $universe/0 -d $external_net
/sbin/ipchains -A output -j ACCEPT -b -p icmp
-s $universe/0 -d $external_net
/sbin/ipchains -A forward -j MASQ -b -p icmp
-s $internel_net -d $universe/0

#允许代理服务器访问任何主机的任何端口,但只准内
#部主机使用代理服务器。
/sbin/ipchains -A input -j REJECT -p tcp -y -l
-s ! $internal_net -d $proxy/32
/sbin/ipchains -A input -j ACCEPT -b -s $proxy/
32 -d $universe/0
/sbin/ipchains -A output -j ACCEPT -b -s $prox-
y/32 -d $universe/0
/sbin/ipchains -A forward -j MASQ -b -s $proxy/
32 -d $universe/0
#允许 DNS 查询和区域传输
/sbin/Enable dns $primarydns dns $protfwdns dns
tcp udp
/sbin/ipchains -j ACCEPT -A input -b -p udp -s
$portfwdns 53 -d $universe/0 53
/sbin/ipchains -j ACCEPT -A input -b -p tcp -s
$portfwdns 1024:-d $universe/0 53
/sbin/ipchains -j ACCEPT -A output -b -p udp -
s $portfwdns 53 -d $universe/0 53
/sbin/ipchains -j ACCEPT -A output -b -p tcp -s
$portfwdns 1024:-d $universe/0 53
/sbin/ipchains -j MASQ -A forward -p udp -s
$portfwdns 53 -d $universe/0 53
/sbin/ipchains -j MASQ -A forward -p tcp -s
$portfwdns 1024:-d $universe/0 53
#Enable http
/sbin/Enable www $internet_web 80 $portfwweb 80
tcp noudp
#忽略 DHCP 数据包
/sbin/ipchains -A input -j REJECT -i $inter-
nal_interface -s 0.0.0.0/0 68 -d 255.255.255.255/32
```

```
67 -p udp
    /sbin/ipchains -A input -j REJECT -i $inter-
nal_interface -s $dhcp_server/32 67 -d 255.255.255.
255/32 68 -p udp
    # 纪录所有被拒绝的包供分析
    /sbin/ipchains -A input -j REJECT -l
    /sbin/ipchains -A output -j REJECT -l
    /sbin/ipchains -A forward -j REJECT -l
```

2.4 测试防火墙

防火墙配置完成后一定要测试一下它是否能达到设计要求。可用如 ISS、SATAN 等工具对防火墙进行测试，以便在正式投入使用之前排除所有问题。

2.5 远程控制

还有一个很重要的环节就是远程控制。在计算机网络如此发达的今天，不可能什么改动都到防火墙去做，必须有一个绝对安全的远程控制工具。推荐使用 SSH，它是一种安全、功能强大的远程控制工具。FOR UNIX 的版本可免费获得，FOR WINDOWS 的版本则需要付费。

(收稿日期：2000-03-21)

(上接第 14 页)

信号量的操作。因为 ISR 具有最高优先级，如果在 ISR 中执行了信号量操作而被挂起，则整个系统将会死锁。

在中断处理上，一般的操作系统与嵌入式操作系统的不同之处是现场保护。一般的操作系统的中断现场保护是由操作系统来完成的，在中断处理完成之后，也由操作系统恢复现场。在嵌入式操作系统中，有时由于受到代码量的限制，中断现场的保护往往由中断处理程序来完成。在中断处理程序的入口要保护在中断处理程序中用到的寄存器，在中断处理完成后恢复。这样一方面减少了代码量，另一方面提高了中断响应时间，但是却损失了系统的安全性，同时也增加了调试的难度。这是在嵌入式操作系统的设计中应该予以关注的问题。

(4)操作系统与用户的接口

操作系统提供给用户使用的有二类接口。一类是人机界面，无论是视窗形式还是命令行形式，这个接口确切地说并不能做为操作系统的一部分，而仅仅是操作系统的一个外壳，这个界面是为了方便用户使用操作系统，而这个接口在嵌入式操作系统中是不存在的。这里要讨论的是另一个接口，操作系统提供给用户开发自己的应用程序接口(API)，也就是系统调用。无论是一般的操作系统还是嵌入式操作系统都应具有这个接口。每一个操作系统提供的系统调用的功能和种类都不同，当然，对于一个操作系统来说，它提供的系统调用越多，则功能越强，对于应用程序的开发，也就越能提供高效而简单的支持，同时也会减少应用程序的维护量。相反，一个操作系统的系统调用越少越单一，那么应用程序相对就要做更多的工作，应用程序也就越复杂。为了适应不断复杂的应用程序开发的需求，操作系统中设计的系统调用也就越来越多，越来越复杂，功能越来越强大。但是这一规律并不适用于嵌入式操作系统，嵌入式操作系统的应用领域非常广，简单的可以应用在调制解调器上，复杂的可以应用在卫星地面通信接收站。这就决定了嵌入式操作系统所提供的系统调用的数量和功能是因应用不同而不同的。尽管前文提到的可裁剪性是嵌入式操作系统的一个非常重要的特性，但是任何一个嵌入式操作系统都不可能从具有各种完善功能、代码达几百 KB 的操作系统，裁剪到只具有实时调度和信号量操作的几 KB 代码。所以嵌入式操作系统只能面向实际的被嵌入系统的具体需求，确定系统调用，以便达到在提供最有效的系统调用的同时具有最小的代码量。

最后，在系统调用的形式上要提到 POSIX。由于各个操作系统提供自己的系统调用，其类型、功能和调用格式各不相同，这样给应用程序的移植带来了很大困难。POSIX 标准的提出正是试图解决这一问题。POSIX 试图定义一些标准的系统调用接口和功能，尽管各个操作系统的实现方式各不相同。POSIX 是以类 Unix 为基础开发的，同时，它试图将实时和非实时的情况统一化，这样就丧失了一定的效率和增加了代码量，所以有些操作系统在提供 POSIX 兼容的系统调用的同时，也提供了非 POSIX 兼容的系统调用。

4 结 论

以上从构成一个操作系统的几个最重要的组成部分讨论了嵌入式操作系统与一般的操作系统的相同与不同之处。当然，构成一个完整的操作系统，还需要其它组成部分，如时钟、同步/互斥、进程间通信等。从以上分析可以看出，要设计一个好的嵌入式操作系统，必须充分考虑被嵌入的系统，要根据实际的应用来设计、选择操作系统。虽然嵌入式操作系统是可裁剪的，但这种裁剪也是有限的。

嵌入式系统正在蓬勃发展，存在着无限的商机，在这一领域，机遇与挑战并存。嵌入式操作系统的设计将在竞争中起决定性作用。

参考文献

1 Ghosh K.A Survey of Real-Time Operating Systems-Draft.GIT-CC-93/18,1994;(2)

(收稿日期：2000-04-14)