基于 Liunx 系统的双宿主主机防火墙的设计

李立峰　郭东辉　刘瑞堂　吴伯僖　（厦门大学物理系　361005）

摘要：本文在说明了防火墙的三种基本体系结构的基础上，着重阐述了双宿主主机防火墙的原理及 Linux 的实现方案，并给出了一个简单的配置实例。

关键词：Linux　防火墙　TCP/IP

一、前言

近几年来因特网越来越普及，已渐渐成了人们日常生活的一部分，同时，越来越多的商业应用也是建立在基于因特网的平台上。然而，不管是个人网络还是企业网络连接在开放的因特网上都需要考虑到网络安全问题，如个人的隐私、各商家的秘密都不希望被他人窥探或窃取。其中，设立网络防火墙是保护个人或企业内部网免受来自外部因特网恶意侵入的一种即有效又经济的选择。

防火墙是指安装在内部网络与因特网之间或者网络与网络之间一种可以限制相互访问的部件或设备。它一般可以实现多种预防和监视服务，如：限定人们从一个特别控制的点进入，防止侵袭者接近其他的内部网的信息，限定人们从一个特别控制的点离开，有效记录因特网的活动记录等。在逻辑上，它可视为分离器、限制器和分析器。它的兑现硬件可以是专用设备、路由器、计算机或这些设备的组合。但总的来看，放火墙可归纳为三种结构类型[1]：双宿主主机结构、被屏蔽主机结构、被屏蔽子网结构。其中，双重宿主主机结构的防火墙是利用一台计算机或专用硬件充当防火墙，该计算机的两个网络接口分别连接因特网和内部网，因特网与内部网的所有通信信息都经过该计算机控制；被屏蔽主机结构的防火墙是通过路由器来实现数据信息包的过滤以屏蔽内部网内的主机，它可以允许某些特定的内部主机开放到因特网上或特定内部主机的特定服务开放到因特网上；被屏蔽子网结构的防火墙是通过在被屏蔽主机结构防火墙的基础上添加额外的安全层如路由器来进一步把内部网络和因特网隔离。

从防火墙的结构来看，相对于被屏蔽子网结构来说，双宿主主机结构式的防火墙与屏蔽主机结构式的防火墙一样存在着单点突破问题，即当主机被侵入后造成内部网的完全暴露。不过，双宿主主机结构式防火墙的网络体系结构相对比较简单，所需的成本比较低，只要通过适当的配置管理，一样具有相当高的安全性。为此，本文将结合我们实验室内部网的防火墙设计经验，介绍一种基于 Linux 系统平台的双宿主主机式防火墙的设计和配置方法。

二、双宿主主机式防火墙的实现原理

如图 1 所示，双宿主主机式防火墙主要是利用一台计算机将因特网和内部网隔离开来，网络间的通信信息流可完全由做为防火墙的计算机过滤控制。根据网络通信 OSI（开放系统互联）参考模型，网络间的通信信息流可细分成七个层次来具体管理和控制。

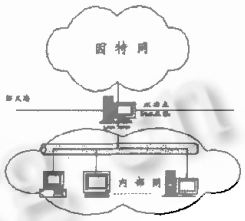

图 1　双宿主主机防火墙结构图

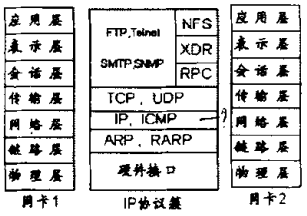

图 2　IP 协议簇与 OSI 的对应关系

对于以 TCP/IP 协议为基础的互连网络来说，用双

© 中国科学院软件研究所 http://www.c-s-a.org.cn

网卡实现因特网与内部网隔离的双宿主主机式防火墙可能处理的信息包与 OSI 参考模型的对应关系如图 2 所示。其中,位于高层的信息包可以加载在低层的信息包里由一个网卡转发到另一网卡进入另一个网络,因此,要实现网络的隔离必须从低的层次做起,例如:若不加任何控制地转发以太网帧,就根本无法实现 IP 包的过滤控制。对于双宿主主机,除了在物理层上两块网卡的接口配置是完全隔开的外,它可以实现链路层以上各种层次协议包的转发,因此,它还可以用来做网桥和路由等设备如图 3 所示。

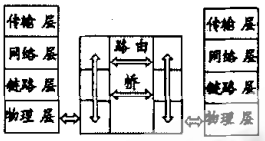

图 3　防火墙过滤(OSI 参考模型图

一般来讲,防火墙所要防范的网络攻击大多为如 Trojan Horse 程序、嗅包(比如侦听以太网包)、冒充 IP 地址、以及拒绝服务式攻击等几种形式[2],这些攻击分别是以链路层、网络层或传输层中相应协议的信息包在网络中传递的,因此,双宿主主机式防火墙的关键就是设计包括网桥和路由协议在内的链路层、网络层与传输层之间所有可能存在的协议包的过滤控制过程。一方面,要保证网络的安全,另一方面,还需要考虑到网络的流畅和使用方便灵活。

根据防火墙的这种设计原理和要求,我们可以首先令内部网和外部网完全隔开,即不转发任意网络协议包,再用 NAT(网络地址转换)方式实现内部网对因特网的自由访问,而因特网对内部主机的访问通过双宿主主机的端口映射来实现。为了说明实现这种双宿主主机式的防火墙配置,以 Linux 操作系统为平台来具体说明其兑现实例。

三、Linux 系统下防火墙实现方案

Linux 是在 1991 年由年轻的芬兰大学生 Linus Torvalds 在一台 Intel386 微机上开发出的一个类似 Unix 的操作系统。由于它是属于 GPL (General Public License) 软件系统,因此当 Linux 系统的内核源代码在 Internet 上公布之后,就赢得了许多专业人员的支持,使 Linux 在很短的时间内形成了一个完整安全的操作系统。同时,它在因特网上得到发展壮大,使它对网络的支持比较完善。其中,Linux 系统中的 IP-Masq 功能就可以用来实现以双网卡连接内部网和因特网的双宿主主机式防火墙。

在 Linux 系统中,IP-Masq 其实是 NAT 的一种实现方式[3]。由于在 Linux 系统的实现中,TCP 协议一般不予分配 32k 以上的端口号,所以,系统核心保留了 61000 以上的 4K 个端口号供 IP-Masq 使用,它的具体数值是在文件 ip-masq.h 里面定义的,用户根据需要可以对它进行修改。当 Linux 系统配置了 IP-Masq 功能后,它跟踪网络上的每个活动,如果内部的其他机器把双宿主主机的内部网络界面设为缺省路由,则内部机器要建立与外面的连接,双宿主主机就在系统内部为这个连接建立一个 IP-Masq 对照表,并且对该数据包的源地址,端口号,序列号等信息进行必要转换,其中源地址改为它的外部网络界面的地址,再通过另一个网络界面转发出去。相反,当对外的网络界面收到目标端口号在 61000 以上的信息包时,则在系统内部的表中查找相应的项,实现反变换,这样就实现了内部主机跟外面机器的无缝连接。

下面就针对以 Slackware 版本的 Linux 系统下 IP-Masq 功能配置防火墙的具体步骤为例进行具体说明:

1.在 Linux 系统平台配置支持双网卡功能,假设两块网卡都是 ISA 总线的 Ne2000 兼容卡,要使内核支持该类网卡,则先在编译内核时打开选项 Network Device Support, Ethernet, Other ISA Cards, Ne2000/Ne1000 ISA Support;其次在/etc/lilo.conf 加进两块网卡的参数,若我们的网卡参是网卡 1 中断设为 3,I/O 端口设为 0x300,网卡 2 中断设为 10,I/O 端口设为 0x320,则我们的具体配置如下:

append = "ether=3,0x300,eth0 ether=10,0x320,eth1"

在系统启动时内核应该正确检测出双网卡并显示如下信息:

ne.c:v1.10 9/23/94 Donald Becker (becker@cesdis.gsfc.nasa.gov)

NE*000 ethercard probe at 0x300: 00 00 21 01 39 33

eth0: NE2000 found at 0x300, using IRQ 3.

NE*000 ethercard probe at 0x320: 00 00 21 01 38 13

eth1: NE2000 found at 0x320, using IRQ 10.

这样,系统就已经可以认出了两块网卡,下一步就是使内核也支持 IP-Masq,通过在编译内核时选择下列选项: Network Firewalls, IP: Forwarding/Gatewaying, IP: Firewalling 和 IP:Masquerading 使内核支持 IP-Masq。

2.配置两个网络界面,假设与互联网相连的网络界面是 eth0,而 eth1 则连到内部网。连到互联网的网络界

© 中国科学院软件研究所 http://www.c-s-a.org.cn

面 eth0 的配置要参考实际的情况进行配置，而内部网 IP 地址范围一般用 RFC1918 中推荐的私有网络的地址范围：A 类地址 10.0.0.0 到 10.255.255.255，B 类地址的 172.16.0.0 到 172.31.255.255 或 C 类地址 192.168.0.0 到 192.168.255.255。我们假设它的 IP 地址范围为 C 类私有地址范围：192.168.1.0。我们在/etc/rc.d/rc.inet1 文件里加入下面配置 eth1 的语句：

```
# config eth1
IPADDR="192.168.1.1"
NETMASK="255.255.255.0"
NETWORK="192.168.1.0"
BROADCAST="192.168.1.255"

/sbin/ifconfig eth1 ${IPADDR} broadcast ${BROADCAST} netmask ${NETMASK}

if [ ! $? = 0 ]; then
  cat << END
Your ethernet card eth1 no work proper.
END
fi
/sbin/route add -net ${NETWORK} netmask ${NETMASK} eth1
```

重新启动机器即可。

3. 设置基本防火墙规则，假设我们的防火墙规则是允许内部子网的机器可以自由访问外面的资源，而外部因特网的机器不允许访问内部网的资源，则对上述规则在 2.2 版本以前的内核用 ipfwadm 的配置如下：

```
/sbin/ipfwadm -F -p deny
/sbin/ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
```

在 2.2 的内核由于对 IP-Masq 部分改写了代码，增强了功能，改用 ipchains 作为 IP-Masq 的配置工具：

```
/sbin/ipchains -P forward DENY
/sbin/ipchains -A forward -j MASQ -s 192.168.1.0/24 -d 0.0.0.0/0
```

如果上述配置不能生效，则可以查看内核 IP 转发的功能是否打开，在 2.2.* 版本的核心中缺省是关闭的，可以通过下述命令打开：

```
echo 1 >/proc/sys/net/ipv4/ip-forward
```

这样，内部子网的机器(192.168.1.0)只要把缺省路由设为 192.168.1.1 就可以透明地访问外面的机器，而外面的机器就看不到内部子网的机器，达到了防火墙的目的。

四、Linux 系统下防火墙增强功能的配置

从上面防火墙的基本实现方案来看，Linux 系统防火墙很容易实现网络隔离，阻止外部主机对内部资源的窥探，而不影响内部机器对因特网的自由访问，但是，当希望让外部因特网的机器能访问内部特定机器资源时，那么按 Linux 系统的 IP-Masq 功能是无法实现的。不过，可以通过 Linux 系统中的 ipportfw 功能来实现内部网络资源对外的安全共享。

Ipportfw 的作用相当于 TCP/IP 的端口映射，即把特定端口的数据包都转发到设定的机器和端口。在 Linux 系统中 2.0.x 版本的内核是不包含对 ipportfw 功能的支持，因此，需要自己对内核打补丁，可以从 http://www.monmouth.demon.co.uk/ipsubs/portforwarding.html 下载到内核的补丁，重新编译内核并打开选项 ipportfw masq support 即可。在 2.2.x 版本的内核中已经包含了对 ipportfw 功能的支持。在确信内核包含对 ipportfw 功能的支持后用前端控制工具 ipportfw 来设定端口映射规则，比如执行如下命令：

```
ipportfw -A -t [eth0 的 IP]/80 -R 192.168.1.10/80
```

或将上述命令加到系统的启动文件/etc/rc.d/rc.local 中，则外部互联网的用户就可以通过访问 Linux 主机对外界面的 80 端口直接访问内部网中 192.168.1.10 机器的 80 端口的资源，而无法访问内部网的其他资源。

五、结论

根据上面双宿主主机式防火墙的设计方案，我们用一台 486 主机配双网卡将实验室内部网与校园网相接，以 Slackware3.5 的 Linux 系统配置了防火墙，实践证明，它完全隐藏了内部网的信息，有效地控制了外部用户对内部资源的访问，同时不影响内部网的用户正常访问校园网的资源，具有较高的吞吐量和较小的延迟。不但是个良好的防火墙，而且较好地解决了 IP 地址短缺的问题。

参考文献

[1] D.B. Chapman 等，构筑因特网防火墙，电子工业出版社，1998 年 1 月.

[2] 肖明等，基于 Linux 的路由器和防火墙技术，计算机应用研究，pp.54－56，1999 年 5 期.

[3] 赵海波等，IP 网络地址映射技术的分析和实现，电子技术应用，Vol. 25，No.5 pp.44－46，1999

(来稿时间：1999 年 6 月)

© 中国科学院软件研究所 http://www.c-s-a.org.cn