Internet 防火墙及其 Linux 实现

吕华锋　吴秋峰
清华大学自动化系　(北京 100084)
E-mail:lvhf98@mails.tsinghua.edu.cn

摘　要　该文阐述了 Internet 防火墙技术的原理,介绍了若干种防火墙的体系结构,给出了 Linux 操作系统下的防火墙软件及其使用实例。

关键词　防火墙 包过滤　代理服务　包过滤防火墙　应用程序网关　双穴网关防火墙　屏蔽主机防火墙　屏蔽子网防火墙　透明代理　Linux　ipchains/iptables　Squid

文章编号 1002-8331-(2001)08-0041-04　**文献标识码** A　**中图分类号** TP393.08

Internet Firewall and its Linux Implementation

Lü Huafeng　Wu Qiufeng
(Automation Department,Tsinghua University,Beijing 100084)

Abstract: This paper explains the basic theory of the Internet firewall technology,introduces the architectures of several kinds of firewall,and gives out some firewall softwares under Linux operating system and their applications.

Keywords: firewall,packet filtering,proxy service,packet filtering gateway,application gateway,transparent proxy,Linux,ipchains/iptables,Squid

Internet 的广泛应用也带来了极其敏感的副作用，这就是安全问题。随着网络在商业和政府机关的日益普及,如何兼顾安全与方便，并防止恶意人士入侵本企业或组织的内部网络,正日益引起重视。为了有效隔离来自 Internet 的外部入侵,防火墙(firewall)技术正在普及中,而防火墙技术也早在 1994 年被 RFC1636 列为信息系统安全机制中不可缺少的一项措施。

目前,防火墙技术已经较为成熟,实用防火墙产品也有不少。文章将阐述防火墙技术的原理,给出若干种防火墙的体系结构,并给出 Linux 操作系统下的防火墙实例。

1　防火墙的概念

防火墙是一种行之有效且应用广泛的网络安全机制。“防火墙”这个词来自建筑物中的同名机构，从字面意思上理解，它可以防止火灾从建筑物的一部分蔓延到其他部分。Internet 防火墙也要起同样的作用，防止 Internet 上的不安全因素蔓延到自己企业或组织的内部网。

Internet 防火墙在一个特定控制点上根据一定的规则控制特定类型的数据包的通过与否。应用防火墙时,首先要明确防火墙的缺省策略，是接受还是拒绝。如果缺省策略是接受,那么,没有显式拒绝的数据包可以通过防火墙;如果缺省策略是拒绝,那么,没有显式接受的数据包不能通过防火墙。显然后者的安全性能更高。例如,假如已经设置防火墙禁止了对某系统提供的 ftp 服务(21 端口)的访问,而该系统又在另外的高端端口重开了 ftp 服务。如果缺省策略是接受，那么高端端口上的 ftp 服务是可以访问的;否则,如果缺省策略是拒绝,则该 ftp 服务不可访问。使用时应该根据自己的实际情况选择这两种策略。

2　防火墙的分类

防火墙从原理上可以分为两大类：包过滤 (packet filtering)型和代理服务(proxy service)型。

2.1　包过滤型防火墙

包过滤型防火墙根据数据包的包头中某些标志性的字段,对数据包进行过滤。当数据包到达防火墙时,防火墙根据包头中下列字段中的一些或全部进行判断,决定接受还是丢弃数据包:

·源地址、目的地址
·协议类型(TCP、UDP、ICMP)
·TCP/UDP 协议的源端口号、目的端口号
·ICMP 类型
等等。

不同的防火墙产品还可能附加其他的过滤准则，如 TCP 协议标志(SYN、ACK、FIN 等)、进入或外出防火墙时所经过的网络接口等等。

根据 TCP 或 UDP 端口进行过滤带来了很大的灵活性。特定的服务/协议是在特定的端口上提供的，阻塞了与特定端口相关的连接也就禁止了特定的服务/协议。应该根据自己的网络访问策略决定要阻塞哪些协议,不过,有些服务天生容易被滥用,应该小心对待。他们 TFTP(69 号端口)、X Windows 和 OpenWindows (6000+和 2000 号端口)、RPC (111 号端口)、rlogin、rsh 和 rexec(513、514、512 号端口)。另外的服务,通常也要给予限制,只允许那些确实需要它们的系统访问。这些常用的服务包括:TELNET (23)、FTP (21 和 20)、HTTP(80)、SMTP (25)、RIP(520)、DNS(53)等。

通过设置包过滤规则,可以阻塞来自或去往特定地址的连

作者简介:吕华锋,硕士研究生,研究方向为计算机网络,尤其是计算机网络安全。吴秋峰,教授,博士生导师,研究方向为工业自动化控制、现场总线网络、计算机网络应用、计算机网络安全等。

接。例如,从收费方面考虑,校园网可能需要阻塞来自或去往国外站点的连接。从安全方面考虑,某组织的内部网可能需要阻塞所有来自外部站点的连接。

包过滤防火墙的弱点主要在于规则的复杂性。通常,确定了基本策略(例如"拒绝"),然后设置一系列相反的("接受")规则;但很多情况下,需要对已经设立的规则设定一些特例;这样的特例越多,规则就越不容易管理。例如,已经设立了一条规则允许对 TELNET 服务(23 号端口)的访问,后来又要禁止某些系统对 TELNET 服务进行访问,那么,只能为每个系统添加一条相应规则。有时,这些后来添加的补丁规则会与整个防火墙策略产生冲突。同时,过于复杂的规则也不易测试。

从概念上讲,防火墙和网关(路由器)是不同的。但在具体实现中,包过滤防火墙通常还具有网关的功能,对数据包进行过滤后再转发到相应的网络。这样的包过滤防火墙或者网关称为"包过滤网关"。事实上,在 Unix 世界里,包过滤功能与 IP 转发功能都是系统内核的内置功能,在同一台主机上把它们结合起来使用是再正常不过的事情了。

2.2 代理服务型防火墙

代理服务防火墙可以解决包过滤防火墙的规则复杂性问题。

所谓代理服务,是指在防火墙上运行某种软件(称为代理程序),如果内部网需要与外部网通信,首先要建立与防火墙上代理程序的连接,把请求发送到代理程序;代理程序接收该请求,建立与外部网相应主机的连接,然后把内部网的请求通过新连接发送到外部网相应主机。反过来也是一样。内部网和外部网的主机之间不能建立直接的连接,而要通过代理服务进行转发。

一个代理程序一般只能为某几种协议提供代理服务,其他所有协议的数据包都不能通过代理服务程序(从而不可能在防火墙上开后门以提供未授权服务),这就相当于进行了一次过滤;代理程序还有自己的配置文件,其中对数据包的其他一些特征(如协议、目的地址、源地址等)进行了过滤,有时这种过滤条件甚至比纯粹包过滤的功能还要强大。

包过滤和代理服务结合起来使用,可以有效地解决规则复杂性问题。包过滤防火墙只需要让那些来自或去往代理服务器的包通过,同时简单地丢弃其他包。其他进一步的过滤由代理服务程序进行。

使用代理服务可以根据协议/服务的某些细节进行过滤。例如,有些代理服务程序对 FTP 连接进行过滤,禁止 FTP 的 put 命令的使用。这可以视为一定意义上的"基于内容的过滤"。在包过滤中,这是不能或不容易做到的。

代理服务禁止了源主机到目的主机的直接连接,可以由此达到一定程度的信息隐藏:内部系统的名字不通过 DNS 被外界知道;外界可以只知道代理服务器的名字,并通过它使用内部系统。

代理服务器在转发数据包之前,可以预先进行身份验证和日志纪录。代理服务器提供的这些功能往往比标准主机系统自带的功能强大。

代理服务的转发与(包过滤)网关的转发层次不同。(包过滤)网关只是在 IP 层次上简单地转发数据包,代理服务器则在应用程序的层次上转发某种协议/服务的数据流。从这个意义上说,代理服务主机也经常被称为"应用程序网关(Application Gateway)"。

除了"应用程序级网关",一些文献中还提出了"线路级网关"。这种"网关"直接转发 TCP 连接,不进行其他的处理或过滤。不过,也许没有必要分得这么细,把它看成应用程序级网关的一种特例就可以了。

代理服务也有一些缺陷。在诸如 TELNET 这样的客户/服务器型的协议中,需要两个步骤进行连接。有些代理服务程序要求对客户端程序进行一些修改,以适应代理服务的要求。

3 几种典型的防火墙体系结构

3.1 双穴网关(Dual-homed Gateway)

双穴网关是包过滤网关的一种替代。与普通(包过滤)网关一样,双穴网关也位于外界 Internet 与内部网络之间,并且通过两个网络接口分别与它们相连。但是,其 IP 转发功能被禁用,其网关功能是通过提供代理服务而不是通过 IP 转发来实现的。显然只有特定类型的协议请求才能被代理服务处理,于是,双穴网关实现了"缺省拒绝"策略,可以得到很高的安全性。

另外一种双穴网关的使用方法是,要求用户先远程登录到双穴网关,再从上面访问外界。这种方式不值得提倡,因为在防火墙上最好保留尽可能少的账户。

3.2 屏蔽主机型防火墙(Screened Host Firewall)

这种防火墙其实是包过滤和代理功能的结合,其中代理服务器位于包过滤网关靠近内部网的一侧。代理服务器只安装一个网络接口,它通过代理功能把某些服务传送到内部某些主机。而包过滤网关把那些天生危险的协议屏蔽/过滤掉,不让它们到达代理服务器那里。

从安全性考虑,内部网络与外界 Internet 之间的通信都通过代理服务器进行,包过滤网关再通过过滤对代理服务器进行保护。这样,包过滤网关应该只让那些来自或去往代理服务器的数据包通过,而丢弃其他所有数据包。

但是有的协议没有相应的代理服务。如果使用这些协议的风险经过考虑认为可以接受(则相应数据包称为"可信任的"),可以让该协议的数据包通过包过滤网关。这些包不经过代理服务器,直接去往相应的服务器。

如图 1 所示,包过滤网关为代理服务器和 E-mail 服务器提供保护,而代理服务器为 FTP 和 HTTP 协议提供代理服务。包过滤网关只放行两种类型的数据包:来自或去往代理服务器的;来自或去往 E-mail 服务器且使用 SMTP 协议的。

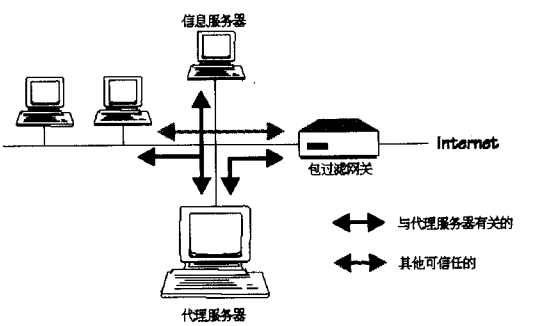

图 1 屏蔽主机型防火墙

这种类型的防火墙的主要安全问题也就出在"可信任"上。上面提到,包过滤方式的缺点在于规则配置比较复杂,规则数目增多时容易互相冲突。本来如果只放行与代理服务器有关的

数据包,包过滤网关的规则配置将非常简单;但对于每一种"可信任的"数据包,只好为它们分别配置包过滤规则,又把复杂性问题遗留了下来。而且,这些"可信任的"服务仍然有可能成为潜在的安全漏洞。解决这个问题的最好方法就是寻找"可信任的"服务的代理软件。好在各种服务的代理软件越来越多了。

3.3 屏蔽子网型防火墙(Screened Subnet Firewall)

这种防火墙是双穴网关和屏蔽主机防火墙的变形。

如图 2,该系统中使用了两个包过滤网关在内部网络和外界 Internet 之间隔离出一个受屏蔽的子网。有些文献称这个子网为"非军事区(DMZ)"。代理服务器、E-mail 服务器、各种信息服务器(包括 Web 服务器、FTP 服务器等)、Modem 池及其他需要进行访问控制的系统都放置在 DMZ 中。

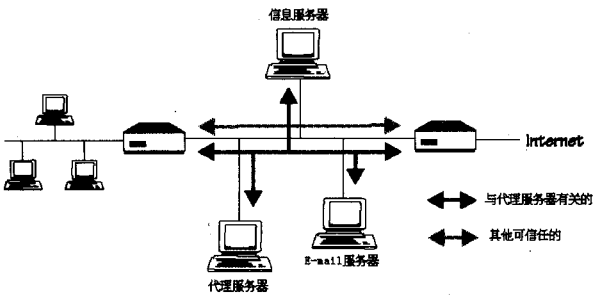

图 2 屏蔽子网型防火墙

与外界 Internet 相连的那个网关称为"外部路由器",它只让与 DMZ 中的代理服务器、E-mail 服务器、信息服务器有关的数据包通过,其他所有类型的数据包都被丢弃,从而把外界 Internet 对 DMZ 的访问限制在特定的服务器范围内。内部路由器的情况也是如此。

这样,内部网与外界 Internet 之间没有直接连接,它们之间的连接要通过 DMZ 的中转。这与双穴网关的情况是一样的。不同的是,屏蔽子网防火墙使用了包过滤网关把数据包转发到特定系统,使得代理服务器只需要安装一块网络接口。

通过配置,可以做到,只有 DMZ 中的代理服务器、E-mail 服务器和信息服务器是外界可见的,外界无法知道其他系统的存在,无法通过它们的名字直接访问它们。

与屏蔽主机防火墙一样,对于那些没有代理又需要使用的服务,要在外部路由器处网开一面,这造成了一定的安全问题。但是,在屏蔽子网结构中,可以把需要这些服务的系统直接放置在 DMZ 中,这些系统通过代理服务器与内部子网联系。这种方法并不完美,但对于需要高度安全性的系统来说,可以作为一个选择。

4 防火墙在 Linux 操作系统下的实现

上面介绍了防火墙的一般概念和几种体系结构。在实践中,已经有大量的硬件和软件防火墙产品上市销售。这里要介绍的是 Linux 操作系统下的软件防火墙,它们都可以免费获得。除了标准的包过滤和代理服务功能,这些软件一般还具有其他一些附加功能。

Linux 内核中内置了包过滤功能,包过滤软件一般只是通过命令行对内核中的包过滤功能进行操作。由于 Linux 的不同内核对包过滤的支持方式不完全相同,因此,每推出一个新的内核版本,相应的包过滤防火墙软件也要进行升级。

Linux 下的包过滤软件有很多。平常应用最广的是 ipchains、iptables 系列。这一系列最早是在 1994 年年底,从 BSD 的 ipfw 移植到 Linux 下的。它在不同内核版本下有不同的版本,分别是:ipfwadm(2.0 内核)、ipchains(2.2 内核)、iptables(2.4 内核),它们依次是前一个的直接升级版本。到目前为止,使用最广泛的是 ipchains(这主要是因为目前 Linux 的稳定内核版本是 2.2),一些重要的 Linux 发行版,如 RedHat、Turbo Linux 等都缺省安装了 ipchains;ipfwadm 由于年代过于久远,使用的人已经不多;iptables 本身应该是随 Linux 2.4 内核一起工作的,由于目前 Linux 2.4 内核还在紧张开发中(预计今年 9 月可以推出),所以使用的人还不多,但实际上 iptables 在内核 2.3.15 之后就可以使用了。目前,笔者使用 Linux 2.3.99pre8 内核,iptables 在上面运行正常。

除了这一系列,还有其他的包过滤软件。比较著名的有 IP Filter,它的命令行语法与 ipchains 系列完全不同,但 ipchains 可以做到的,它都可以做到,而且据称做得更"干净"。Mason,具有一定智能,可以在一个学习周期内根据流经的数据包自动动态生成防火墙规则,它甚至还可以通过分析 ipchains 或 ipfwadm 的日志纪录,生成 ipchains 或 ipfwadm 规则。还有其他一些形形色色的包过滤软件,这里不再介绍了。

Linux 下的代理服务软件同样数目众多,比较著名的有 Squid、SOCKS、The TIS Firewall Toolkit (FWTK)、delegate 等。Squid 是一个很好的 http、ftp 代理服务软件,同时还支持 Linux 的透明代理特性。

下面,将以 ipchains/iptables、Squid 为例,介绍它们的原理和使用方法。

4.1 IPChains/IPTables

前面说过,包过滤功能是内嵌在 Linux 内核中的,一些关键的数据结构是在内核中实现的。ipchains/iptables 只是通过命令行来操纵这些数据结构而已。这些数据结构主要是若干条规则表,称为"防火链"或直接简称"链(chain)"。(这也是 ipchains 名字的来由。)每条链由若干条规则(rule)和一个缺省策略(policy)构成。

对于 2.2 内核,系统启动之后就存在了三条链:input (进入)、output(外出)、forward(转发)。当一个数据包从网卡进入防火墙,系统使用 input 链来检验,如果通过检验,系统对该包进行路由;如果该包的目的地址就是本机,系统将其转交给相应进程处理;如果目的地址不是本机而是另外一台主机(也就是说,本机是该包旅途中的一个网关),系统再使用 forward 链进行判断;最后,一个包被发送出去之前,要接受 output 链的检验。

那么,具体是如何运用链进行检验的呢?一条链由若干规则(rule)和一个缺省策略(policy)组成。其中,一条规则包含一些匹配条件和一个目标 (target) 行为,目标行为通常是接受(ACCEPT)或拒绝(DENY、REJECT,这两者有些细微差别)。而缺省策略只有接受和拒绝两个选择,这与我们前面提到的一致。一个包进入某链后,要依次与每一条规则进行比较;如果匹配了某条规则的条件,就执行其目标动作,接受或丢弃该包。如果不匹配一条规则的条件,就与下一条规则进行比较;如果所有规则都不匹配,那么,由这条链的缺省策略(接受或拒绝)来决定该包的命运。

下面举例说明。这里不准备详细介绍语法,但读者可以通过简单介绍有个感性认识。

ipchains -P forward ACCEPT

把 forward 链的缺省规则设为接受；

ipchains -A forward -s 192.168.1.0/24 -p icmp -j DENY

向 forward 链中添加一条规则：来自网络 192.168.1.0/24 的 icmp 包，拒绝；

ipchains -A forward -d www.tsinghua.edu.cn -p tcp --destination-port 80 -j DENY

再添加一条规则：去往 www.tsinghua.edu.cn 服务器的 www 服务(tcp 协议 80 端口)的包，拒绝；

对一个包进行一系列的检验，最终目的是为了决定如何对待这个包：接受，或者丢弃。这是防火墙的标准选项。除此之外，ipchains(以及 iptables)还有其他扩展动作，如伪装(masquerade)、重定向(redirect)、标记(mark)等，其具体情况参见有关资料。

iptables 是 ipchains 的直接升级。在 2.4 内核下，包过滤功能的实现方法有了根本性的变化。内核提供了一个称为 NetFilter 的框架结构，所有有关包过滤、NAT(网络地址转换)的功能都在这个框架内实现。这个框架还提供了很好的可扩展性，可以很容易地根据框架提供的接口以模块形式添加新的匹配规则或新的目标动作。事实上，iptables 中对 mac 地址的匹配、对本地用户的匹配，以及做日志(LOG)目标动作就是作为扩展模块实现的。这样的扩展在 ipchains 中是比较困难的。

这里只从使用角度考虑 iptables 与 ipchains 的区别。从包过滤方面，仍然是三条链：INPUT、OUTPUT、FORWARD(名字变成了大写)，但在处理过程上有很大不同。一个包到达系统后，首先要进行路由；如果该包的目的地址是本机，则由 INPUT 链进行检验，结果或者被丢弃，或者通过检验，交给一个本地进程处理；如果该包的目的地址不是本机，则由 FORWARD 链进行检验，或者被丢弃，或者通过检验，被转发出去；OUTPUT 链仅对本地产生的外出包进行处理。

iptables 的这种处理方式使得任何类型的包只要经过一条链的检验：来自外地、目的地址是本地的包，由 INPUT 链处理；来自外地、目的地址不是本地的包，由 FORWARD 链处理；本地产生、去往外地的包，由 OUTPUT 链处理。这比 ipchains 要简捷得多。相比之下，来自外地、目的地址不是本地的包在 ipchains 的情况下，要依次经过 input、forward、output 链。这样导致了不必要的复杂性。

上面的例子在 iptables 中如下：

iptables -P FORWARD ACCEPT

iptables -A FORWARD -s 192.168.1.0/24 -p icmp -j DROP

iptables -A FORWARD -d www.tsinghua.edu.cn -p tcp --dport 80 -j DROP

4.2 Squid

Squid 是一个高性能的代理缓存服务器。它支持 ftp、gopher、http 数据对象。Squid 的配置文件主要是 squid.conf，下面提及的配置都在该文件中进行。

Squid 功能强大，有多种工作模式可以选择。这里只介绍两种。

普通代理服务。这是平常惯用的方式，需要在 web 浏览器显式指定代理服务器的主机地址和端口号。浏览网页时，浏览器先连接代理服务器，代理服务器再连接真正的目的地址，并把用户的请求转发过去。Squid 缺省工作在这种方式下，缺省的代理端口(就是需要在浏览器中设置的代理端口)是 3128。这由 squid.conf 文件中的 http_port 指令来定义。

当然，如果只是无条件地转发请求，其防火墙功能将大打折扣。Squid 提供了功能强大的过滤手段，由称为 ACL(Access List，访问控制列表)的机制实现。ACL 首先定义一系列的访问控制条件，然后由 http_access 指令对符合各个条件的连接进行 deny 或 allow。

ACL 指令的格式如下：

acl aclname acltype string

其中，aclname 是用来标识该访问列表的名字，acltype 是匹配类型，包括：src、dst(源地址、目的地址)、srcdomain、dstdomain(源域、目的域)、srcdom_regex、dstdom_regex(对源域/目的域进行正则表达式匹配)、url_regex、urlpath_regex (对 URL/URL 路径进行正则匹配)、time (时间)、port (端口)、proto (协议)、user(用户)等，其中对域的匹配和正则匹配是 ipchains/iptables 没有的。string 则是这些列表的具体描述。例如：

acl EDU dstdomain .edu.cn

表示一个名为 EDU 的访问列表，满足其条件的包的目的地址的域必须是.edu.cn。

http_access 指令根据已经定义的 acl 列表来允许和拒绝访问。

例如：

acl ALL src 0.0.0.0/0.0.0.0

acl EDU dstdomain .edu.cn .edu

http_access allow EDU

http_access deny ALL

允许所有目的地址域为.edu.cn 或.edu 的访问，其他访问一概拒绝。

透明代理服务。顾名思义，代理服务的设置对用户是透明的，用户不需要知道代理服务是如何进行的。在这种情况下，用户不用在浏览器中显式指定代理服务主机，只要把网关设置为代理服务所在主机；代理服务主机收到浏览请求后，把这些数据包重定向到本机代理服务所在的端口，由代理服务程序进行处理。不过，数据包的重定向显然是代理服务程序做不了的，需要其他程序的参与。

下面的实例显示了如何用 ipchains/iptables 与 squid 配合使用，完成透明代理功能。

网络配置如图 3 所示。

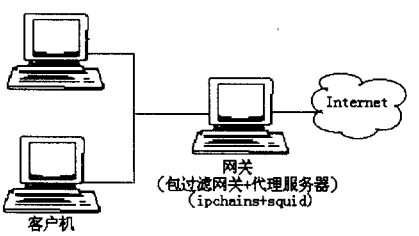

图 3 透明代理示意

其中用作网关的主机上安装了 ipchains 和 squid，由 ipchains 完成端口重定向功能。

(下转 103 页)

(2)在客户端,用户连入站点,如果是首次访问,则 Applet 会被下载到客户端。在用户选择了他所关心的股票后,由 Java 服务器通过 JDBC 查询数据库,获得股票的初始价格,并通过网络连接传递给 Applet.

(3)整个系统处于就绪状态。

(4)当服务器端的所关心的数据发生变化时,数据库触发器发生作用,数据通过命名管道从数据库服务进程传送到守护进程。守护进程则找出那些客户需要数据,然后通过网络连接将该数据传送给客户端的将数据发送到客户端的 Applet。如果没有该股票对应的端口,则说明目前没有人关心该支股票,可以不必发送。

(5)当客户端的 Applet 得到数据后,进行处理并显示。

4.6 系统结构图

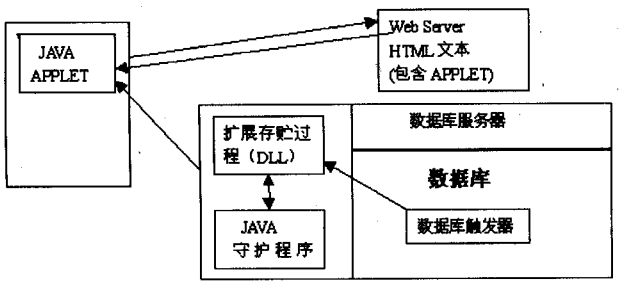

图 3

4.7 为什么选用 JAVA 技术

对这个系统中的守护程序之所以采用 JAVA 实现,首先是由于使用 Java 编写功能性的服务器更加快捷,特别是设计网络通信的情况下。另一方面是由于 Java 服务器的移植性很强[3]。

5 结束语

该文提出的模型,在数据的实时性和发布的主动性方面有明显的优点,但也有它的不足,网络通信的负载问题没有完全解决。在实际应用中,为了编制出更加出色的网络应用程序,使其既能适应各种复杂的情况,又能使网络安全的运行,必须考虑将多种模型在应用系统中结合起来,只有这样的系统才能更高效,可靠、安全。(收稿日期:2000 年 4 月)

参考文献

1.李昭原.数据库技术新进展[M].清华大学出版社
2.周世雄.NT 网络数据库速成-解决方案篇[M].中国铁道出版社,1998.9
3.王克宏.JAVA 语言编程技术[M].清华大学出版社,1997.8
4.陈滢等.DBMS WEB 支撑框架研究[J].计算机研究与发展,1998.6
5.赵洪彪等.Internet 环境中的数据库信息发布途径[J].软件学报,1998.8
6.LaurenceVanhelsuwe 著,邱仲潘译.JavaBeans 从入门到精通[M].电子工业出版社,1998.1
7.NEIL BARTLETT 等著,杨武杰译.WEB 推拉新技术[M].电子工业出版社,1998.6

(上接 44 页)

为了使 squid 支持透明代理功能,需要在 squid。conf 文件中进行如下配置:

```
http_port 8080
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

为了把浏览请求重定向到本机的代理服务 (8080 端口),其他请求一概拒绝,需要如下设置 ipchains:

```
ipchains -P input DENY
ipchains -A input -p tcp -d any/0 80 -j REDIRECT 8080
```

如果使用 iptables,则设置如下:

```
iptables -t nat -P FORWARD DENY
iptables -t nat -A FORWARD -p tcp --dport 80 -j REDIRECT --to 8080
```

这样,所有目的端口是 80 的 tcp 数据包被重定向到本机的 8080 端口,由 squid 处理。由 acl 和 http_access 指令定义的访问控制仍然是有效的。

这其实可以看成屏蔽主机式防火墙的一种变形。限于篇幅,其他例子不再举出,请参阅其他资料。

(收稿日期:2000 年 7 月)

参考文献

1.Elizabeth D Zwicky,Simon Cooper,D Brent Chapman.Building Internet Firewalls[M].2nd Eidition,2000
2.John P Wack,Lisa J Carnahan.Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls
3.Linux IPChains HOWTO.来自 www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html
4.Linux 2.4 Packet Filtering HOWTO.来自 http://netfilter.kernelnotes.org/
5.Linux 2.4 NAT HOWTO.来自 http://netfilter.kernelnotes.org/
6.Linux Netfilter Hacking HOWTO.来自 http://netfilter.kernelnotes.org/
7.SQUID Frequently Asked Questions.来自 http://www.squid.org/

(上接 60 页)

4 结束语

该文介绍了用 UDP 协议和参考 RTP 协议标准在 IP 网传输多媒体数据的方案,并给出单播、广播、组播的编程方法。作者在开发基于 IP 的远程实时教学系统过程中,采用以上方案传输视频音频流的效果比较理想。因此,这种方案在开发基于 IP 的多媒体系统有一定的参考价值。(收稿日期:2000 年 7 月)

参考文献

1.Marcus Goncalves,Kitty Niles.Ip Multicasting Concepts and Applications[M].电子工业出版社,2000
2.周明天,汪文勇.TCP/IP 网络原理与技术[M].清华大学出版社,1997
3.Kris Jamsa,Ken Cope.Internet Programming[M].电子工业出版社,1997
4.蒋林涛.多媒体通信网[M].电子工业出版社,1999