

# 基于 SPI 的数据包过滤转发的设计与实现

袁 超, 黄本雄

(华中科技大学 电子与信息工程系, 湖北 武汉 430074)

**摘 要:**文中介绍了 Windows 环境下根据 Winsock 2 服务提供者接口(Service Provider Interface, SPI)开发数据包过滤程序的技术,并提出一种将此技术与 Socks5 协议相结合,在客户端实现 TCP,UDP 数据包代理转发的方案。重点阐述了 SPI 技术实现自定义基础服务提供者或分层服务提供者的原理,以及以分层服务提供者方式对 SPI 截获的 TCP,UDP 数据包分别进行代理转发的实现。此技术可广泛适用于单机透明代理客户端、基于代理服务器的计费系统等方面。

**关键词:**服务提供者接口;数据包过滤、转发;Socks5 协议;代理服务器

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2006)06-0045-03

## Design and Implementation of Packet Filtrating and Forwarding Based on SPI

YUAN Chao, HUANG Ben-xiong

(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

**Abstract:** Simply introduces the study result of technique of packet filtrating based on SPI in Windows operating system. Then this technology is put forward innovatively to integrate with Socks5 protocol for the design and implementation of a solution related to TCP and UDP packet filtrating and forwarding. This paper focuses on some knowledge of user-defined basic service provider and layered service provider, and the key point of forwarding TCP and UDP packet to Socks5 proxy server by layered service provider. This technique is useful in the applications of Socks5 proxy client and billing system based on proxy server technique.

**Key words:** SPI; packet filtrating and forwarding; Socks5 protocol; proxy server

### 0 引 言

随着计算机网络的普及和相关技术的迅速发展,代理服务器技术得到了广泛的应用。很多时候网络终端用户需要通过代理服务器穿透防火墙连接到特定服务器,而目前很多应用层的网络计费系统也是依托于代理服务器机制的。诸如此类的很多应用需要一种解决方案,能够对网络终端需要网络上传的数据包进行过滤和分析操作,有选择地将其中发送到特定远端的数据包拦截,封装报头,通过代理协议(如 SOCKS5 协议)实现代理转发。文中提出了一种将 WINSOCK 2 SPI 技术与 SOCKS5 协议相结合,实现数据包代理转发的解决方案。此方案已经在一种网络计费商业平台中应用。

### 1 包拦截的多种方法

根据 TCP/IP 协议在 Windows 网络架构中实现的特

点,在 Windows 环境下有多种方式实现数据包的截获。工作在内核级的有 NDIS 中间驱动程序、NDIS Hook Driver、TDI(传输驱动接口)过滤驱动程序等技术;工作在用户级的有 WINSOCK2000 包过滤接口、API HOOK 拦截、WINSOCK 2 SPI 等技术<sup>[1]</sup>。

作为用户态的数据包过滤技术,WINSOCK 2 SPI 的缺陷是它仍然可以被底层的数据包绕过,但对于目前绝大部分基于 TCP,UDP 协议的网络应用程序(如网络游戏、浏览器、FTP 客户端等),WINSOCK 2 SPI 技术完全可以胜任数据包的拦截,而且 WINSOCK 2 SPI 技术针对应用层,在用户态中工作,效率高、实现简单、CPU 占用率小、程序可移植性强。因此本方案选择 WINSOCK 2 SPI 技术实现数据包的拦截。

### 2 WINSOCK 2 SPI 技术原理

WINSOCK 2 服务提供者接口(Service Provider Interface, SPI)是一种基于 Windows 的网络编程接口,它是围绕着 Windows 开放系统架构(Windows Open System Architecture, WOSA)来设计的。WINSOCK 的一端是 API,另一端则是 SPI, WOSA 在 WINSOCK 和 WINSOCK 应用

收稿日期:2005-10-14

**作者简介:**袁 超(1980-),男,湖北武汉人,硕士研究生,研究方向为网络通信与网络安全;黄本雄,教授,博士生导师,研究方向为通信软件与交换技术。

程序之间有一个标准 API;在 WINSOCK 和 WINSOCK 服务提供者(比如 TCP/IP)之间有一个标准的 SPI。WS2\_32.DLL,如图 1 所示,是 WINSOCK 2 支持的动态链接库,其主要功能是提供 WINSOCK 2 应用程序调用接口到服务提供者接口的映射<sup>[2,3]</sup>。

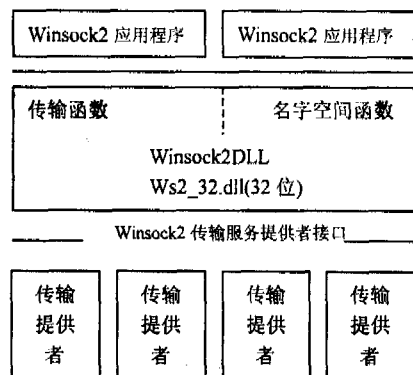


图 1 WINSOCK 2 体系结构

服务提供者就是 Win32 支持的 DLL, 挂靠在 WINSOCK 2 的 WS2\_32.Dll 模块下。对 WINSOCK 2 API 中定义的许多内部调用来说, 这些服务提供者都提供了它们的运作方式。

WINSOCK 2 SPI 允许开发两类服务提供者: 传输提供者和名字空间提供者。“传输提供者”(Transport providers, 一般称作协议堆栈, 比如 TCP/IP) 即能够提供建立通信、传输数据、日常数据流控制和错误控制等功能的服务。“名字空间提供者”(Name space providers) 则把一个网络协议的定址属性和一个或多个用户友好名关联到一起, 以便启用与协议无关的名字解析方案<sup>[2]</sup>。

文中主要涉及的是传输服务提供者。传输服务提供者又分为两类: 基础服务提供者和分层服务提供者。分层服务提供者是在基础服务提供者的上层, 依靠底层基础服务提供者实现更高级的自定义的通信服务; 基础服务提供者执行网络传输协议(比如 TCP/IP)的具体细节, 其中包括在网络上收发数据之类的核心网络协议功能, 它是相对于分层服务提供者而言的, 如图 2 所示<sup>[2,3]</sup>。本方案采用分层服务提供者的方式实现。

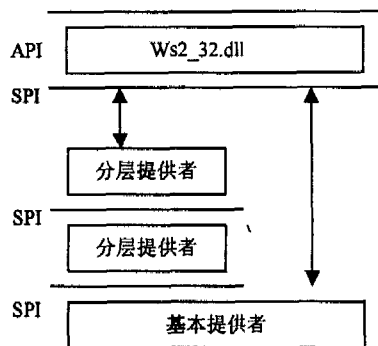


图 2 分层提供者的架构

Windows 中, 现有的系统服务提供者(系统自带)几乎已提供了所有基本的服务。因此在加入自定义的分层提供者, 对数据包进行特定的操作(本方案的操作是对特定

的 TCP, UDP 数据包按照 SOCKS 5 协议封装包头)之后, 还是调用系统服务提供者来完成绝大部分剩下的功能。

### 3 SOCKS5 协议技术简介

SOCKS 协议是用于 C/S 环境下的传输层代理协议, 它是提供在 TCP 或 UDP 上 C/S 应用的一种网络架构。SOCKS5 协议由 SOCKS4 协议扩展而来。SOCKS 4 主要针对基于 TCP 的 C/S 应用, 如 TELNET, FTP, HTTP 等。SOCKS 5 在前者的基础上扩展了 UDP 的报文中继和鉴别功能, 并支持域名和 IPv6 的寻址方法。

SOCKS5 协议是一种需要进行身份认证的协议, 它可以根据对用户的身份认证进行监视和访问控制; 在客户机与主机之间建立一条虚电路; 实现数据转发。

执行 SOCKS5 标准的客户端将标准的 TCP/UDP 连接请求按照一定格式封装后发给 SOCKS5 代理服务器。客户端建立与 SOCKS5 服务器的连接(通常其服务端口为 1080), 客户端首先发送一个版本标识/方法选择的 TCP 报文给 SOCKS5 服务器, SOCKS5 服务器收到客户端的请求后, 根据自身系统的实现返回告诉客户端采用哪种验证方法; 接着该服务器根据选择方法验证客户身份。验证成功后就可以进行数据通信了。客户请求的有 3 种不同的命令类型, CONNECT 模式: 表示客户请求的是去连接远端的套接字; BIND 模式: 表示客户建立套接字并等待外面的程序的连入; UDP ASSOCIATE 模式: UDP 是增加的新功能, 可以发送和接收 UDP 数据包。对于 CONNECT 命令, 在接收到确认后, 就可以把数据发送到 SOCKS5 服务器, 服务器会自动发送到远端。而基于 UDP 的数据每次发送都必须加上一定的头信息。服务器并不接收所有的 UDP 包, 而只接收客户登记过的端口的数据包<sup>[4]</sup>。

### 4 方案的实现和应用

本套方案的关键在于, 要在对网络终端用户透明的前提下, 将本机发往特定目标服务器的 TCP, UDP 包通过 SOCKS5 代理服务器转发。这就要求本系统客户端程序要能根据数据包的目的地址来区分本机发出的所有 TCP, UDP 包, 将特定的数据包筛选出来, 对其进行 SOCKS5 包头封装, 将其发往指定的 SOCKS5 服务器, 依照 SOCKS5 协议建立代理通道, 进行转发。另一方面, 放行其他无关的数据包。

在客户端具体实现中, 构建两个程序: 一个是可执行文件用来安装传输服务提供者; 另一个就是 DLL 形式的数据传输服务提供者。WS2\_32.Dll 是使用标准的动态链接库来加载服务提供者接口的 DLL 到系统中去的, 并调用 WSPStartup 来初始化<sup>[5]</sup>。WSPStartup 的参数 LP\_WSAPROTOCOL\_INFOW 指针提供应用程序所期望的协议信息, 然后通过这个结构指针可以获得所保存的系统服务提供者的 DLL 名称和路径, 加载系统服务提供者后

查找到系统 SPI 程序的 WSPStartup 函数的指针,通过这个指针就可以将自己服务提供者的 WSPStartup 函数和系统 SPI 程序的 WSPStartup 函数相关联,进而调用系统的各个服务提供者函数。在插入的这一层传输服务提供者中,实现对特定数据包的过滤与封装<sup>[5]</sup>。

应用程序在调用 WINSOCKET API 函数时,WSPStartup 通过一个被当作参数投送的函数派遣表打开另外的 30 个 SPI 函数,传输服务提供者便由这 30 个函数组成。每个 API 函数都映射到对应的 SPI 函数。而我们只关注建立 TCP 连接,发送 UDP 包所调用的几个重要的 API 函数,如 Connect, sendto 等,在插入的 SOCKS5 传输服务提供者的相应 SPI 函数中做特殊处理。而其他 WINSOCKET API 函数所对应的 SPI 函数不需要重新编写。

客户端运行时,首先将设置好的特定目标服务器 IP 表读入共享内存,所有目的 IP 在此表中的数据包即是需要代理转发的数据包。在 Connect 函数对应的 SPI 函数 WSPConnect 中,首先进行这样的判断,如果目的 IP 在表中,说明该连接的数据包需要代理转发,则发起到指定 SOCKS5 服务器的连接,进行身份认证,启动 SOCKS5 协议的 Connect 模式,从而实现 TCP 包的代理转发。对于不需要转发的数据包,则直接调用下一层基础服务者 SPI 接口,由系统服务提供者完成默认的功能。在 sendto 函数对应的 WSPSendTo 函数中,也要进行类似的对 UDP 包的判断和处理,对于需要代理转发的数据包,依照 SOCKS5 协议执行身份认证,启动 UDP ASSOCIATE 模式,对每一个 UDP 数据包加上特定的头信息。而值得注意的是,SOCKS5 协议必须通过 TCP 方式进行通信。要发送穿透

SOCKS5 代理服务器的 UDP 数据包,其实首先需要建立客户端到代理服务器的 TCP 连接,通过一系列的交互,获得代理服务器的许可才能够发送出去(同时代理服务器也记录下客户端 IP 和端口),也确保从远端发回的数据能够通过代理服务器发回给某个 UDP 客户端(因为它登记了一个关于 SOCKET UDP 的通路映射)。所以为了发送 UDP 数据,必须建立和保持这个 TCP 连接。不能在取得代理服务器的通道后就关闭 TCP 连接。

本方案已经成功应用于一种商用的网络游戏集中计费系统中,即在客户机上运行该客户端,将所有发往需要计费的网络游戏服务器 IP 的数据包截取并转发到指定 SOCKS5 服务器上,SOCKS5 服务器通过监控记录 SOCKS5 认证用户名,客户机 IP,建立连接的时间、时长,目标服务器 IP,端口等信息,完成分析和计费。而对客户完全透明。该方案为类似的基于有选择的代理服务器透明转发的相关应用提供了一种思路。

#### 参考文献:

- [1] 朱雁辉. Windows 防火墙与网络封包截获技术[M]. 北京:电子工业出版社,2002.
- [2] Jones A, Ohlund J. Network Programming for Microsoft Windows[M]. US: Microsoft Press, 2000.
- [3] 汪国洋, 王景中. 基于 SPI 的访问控制技术[J]. 计算机应用, 2003, 23(6): 267-269.
- [4] Microsoft Corp. MSDN[Z]. 2001.
- [5] 胡滨. Windows 下使用 SPI 过滤网络数据包[J]. 华中科技大学学报(自然科学版), 2003, 31(增刊): 169-170.

(上接第 44 页)

```
PRIMARY KEY (id),
KEY UserName (UserName(32)) );
CREATE TABLE radgroupcheck ( //创建组检查表
id int(11) unsigned NOT NULL auto-increment,
GroupName varchar(64) NOT NULL default '', //可以按照
楼层来划分组
Attribute varchar(32) NOT NULL default '', //属性值,包
括组号、NAS 标识
op char(2) NOT NULL DEFAULT '=',
Value varchar(253) NOT NULL default '',
PRIMARY KEY (id),
KEY GroupName (GroupName(32)) );
CREATE TABLE radreply ( //创建由 RADIUS 返回给客户端
的属性表
id int(11) unsigned NOT NULL auto-increment,
UserName varchar(64) NOT NULL default '', //认证账号
Attribute varchar(32) NOT NULL default '', //返回属性,
包括给客户端返回的 IP
op char(2) NOT NULL DEFAULT '=',
Value varchar(253) NOT NULL default '',
```

```
PRIMARY KEY (id),
KEY UserName (UserName(32)) );
```

#### 3 结束语

文中设计了一种基于 RADIUS 协议的校园网认证计费管理系统的方法,经过茂名学院校园网的长期测试表明,本系统长时间运行稳定、可靠,处理能力达到每秒 250 个接入认证。同时由于采用了后台数据库技术,增加了系统的可靠性、可用性和可伸缩性,能够实现在 FreeBSD, Linux 等各种平台上运行。

#### 参考文献:

- [1] Rigney C. RFC 2866, Radius Accounting[S]. 2000.
- [2] RFC 2865, Remote Authentication Dial In User Service (RADIUS)[S]. 2000.
- [3] 陶智勇. 综合宽带接入技术[M]. 北京:北京邮电大学出版社, 2002. 75-83.
- [4] 唐磊, 金连坤. 大型拨号认证计费服务器的设计与实现[J]. 计算机工程与设计, 2004(7): 160-161.
- [5] 赵宇, 刘刚, 杨宗凯. 一种基于 Linux 的 Radius 客户端的设计与实现[J]. 计算机工程, 2003(9): 113-114.