文章编号:1005－0523(2005)05－0096－04

Sniffer 原理解析及其 WinPcap 实现

刘波涛[1], 赵 刚[1], 冯翠丽[2], 唐 乐[1]

(1.成都市新都区西南石油学院,四川 成都 610500;2.长江大学 计科院, 434102)

摘要:在分析 Sniffer 工作原理的基础上,阐述了有关 Sniffer 设计的一般流程,并用 VC++结合 WinPcap 实现了 Sniffer 的主要功能.

关 键 词:嗅探器;TCP/IP;WinPcap;VC++

中图分类号:TP206 **文献标识码**:A

随着网络的日益普及,网络对信息安全的要求也与日俱增.在网络安全中扮演着重要角色的 Sniffer,将越来越受到关注.熟悉 Sniffer 的原理、实现将有利于更好的利用 Sniffer 来维护计算机网络及解决网络安全问题.

1 Sniffer 的工作原理

要对 sniffer 的工作原理进行深入的了解,可以从网络通讯及网卡的基本工作原理着手.在实际的网络通讯 TCP/IP 模型中,数据以一定的单位(不妨称为数据包)在网络上进行传输.数据包在发送端一方的协议栈从上层至下层被依次封装,即将上层的整个数据包作为下层的数据部分,再加上下层的 Header,作为新的数据包.此过程如图 1 所示.

其中,网卡工作在 L1 层,在这里数据包被网卡驱动程序封装成帧,然后被发送到网线上,经网络路由到达目的机器.那么,包是怎么到达目的机器的呢? 这是通过两个地址(MAC 地址及 IP 地址)及路由协议实现的:首先网络根据此数据包的 IP 地址将其路由到目的机器所在网络的网关上(这一步由路由协议实现),然后由此网关将数据包发送到 MAC 地址所指定的目的机器.而 Ethernet(此处指的是共享式而非交换式的 Ethernet)的工作方式是,把要发送的数据包发送给连接在一起的所有主机,每台机器视网卡的工作模式来判断是否接收此数据

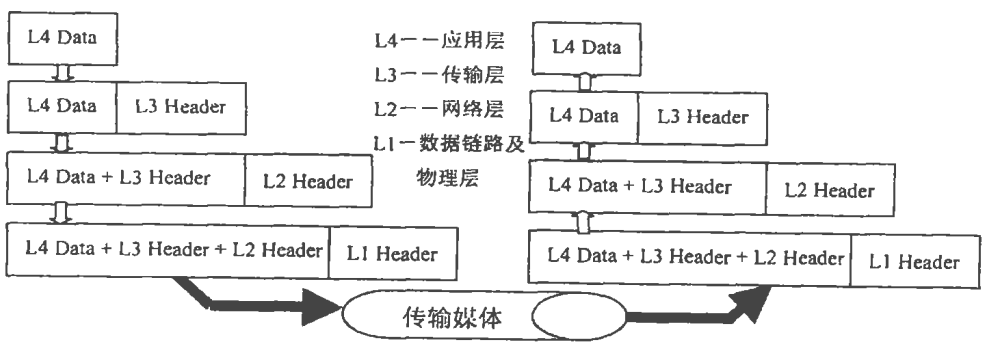

图 1 TCP/IP 模型的数据流图

收稿日期:2005－04－18

作者简介:刘波涛(1980－),男,湖北省孝感市大悟县,现就读于成都市新都区西南石油学院研究生院硕士.

包．若认为该接收，就在存储后产生中断信号通知 CPU；若认为不该接收，就简单地丢弃该数据包．也就是说，不该接收的数据包被网卡截断了，OS 根本就不知道．而接收后的数据包则经由接收端一方的协议栈从下层至上层被依次解包，即将下层的 Header 剥去，将数据部分根据 Header 的内容交付上层的协议或进程处理．

对网卡来说，接收帧的目的 MAC 地址可以是六种 MAC 地址之一：1>To_Me，目的 MAC 地址就是自己；2>To_Others，目的 MAC 地址是别的主机；3>Multicast_In，目的 MAC 地址是组播地址，此时该 Host 在多播列表中已注册；4>Multicast_Not，目的 MAC 地址是组播地址，此时该 Host 还没有在多播列表中注册；5>Broadcast，广播地址；6>Group_Bit，目的 MAC 地址既不是 Broadcast 也不是 Multicast，其组位被置位，即 01－00－00－00－00－00．

以太网卡有两种工作模式：normal model 和 promiscuous model．这两种模式针对不同 MAC 地址的过滤方式如表 1 所示．

表 1　网卡的过滤方式

Model \ MAC	To_Me	To_Others	Multicast_In	Multicast_Not	Broadcast	Group_Bit
normal	Pass	Reject	Pass	Reject	Pass	Reject
promiscuous	Pass	Pass	Pass	Pass	Pass	Pass

通常，网卡都工作在 normal model 下，故对于不属于自己的数据包均作简单的丢弃处理．但如果让某个 Host 的网卡处于 promiscuous model，那么它就可以捕获网络上所有的数据包，此时的它（包括其软件）实际上就构成了一个嗅探器．

也就是说，Sniffer 的基本工作原理是：让网卡接收一切它所能接收到的数据．

2　Sniffer 的设计及实现

从以上 Sniffer 的工作原理可知，设计 Sniffer 的基本思想就是：置网卡于混杂模式、捕获数据包、分析数据包．一般来讲，Windows 环境下有三种方案可供选择以编程实现 Sniffer，表二对它们各自的优缺点进行了简要的比较．

表 2　Windows 下实现 Sniffer 的三种方案

方法	优点	缺点	补充
Raw Socket	实现简单	只能抓到 IP 包，对与 IP 同层（如 ARP）的其它数据包无能为力	只能获得包的一份拷贝
NDIS	可以对网卡进行任意的操作 可以抓取原始数据包	驱动程序开发代价较大	可截断数据流而不仅仅只获得一份拷贝
Packet32 或 WinPcap 库	可抓取原始数据 提供应用程序接口，复杂的内部操作由 DLL 完成	实现的 Sniffer 需要安装相应的链接库才能运行	只能获得包的一份拷贝

1）Raw Socket（套接字）起初是 Unix 上最流行的网络编程接口之一，后来微软将它引入到 Windows 中并得以实现．

2）NDIS（Network Driver Interface Specification）由 Microsoft 和 3Com 公司联合开发，是 Windows 中“通信协议程序”和“网络设备驱动器”之间通信的规范，它为协议通讯程序操作网络设备提供标准的接口，它支持计算机通过不同的协议栈与网络相连．

3）Packet32 和 WinPcap（windows packet capture）都是免费的基于 Windows 平台下的网络接口，为 win32 应用程序提供访问网络底层的能力，并且它们工作于驱动层、效率很高．前者是是微软的的一个实现版本，后者是 UNIX 下的 libpcap 移植到 windows 下的产物．

WinPcap 包提供的功能有：能够捕获原始数据包、方便地把数据写入到文件和从文件中读出、按照自定义的规则过滤、发送原始数据包、收集网络统计信息．用 VC++结合 WinPcap 包实现 Sniffer 的一般流程如图 2 所示．

具体到编程实现上，除去必要的出错处理外，关键功能的实现如下：

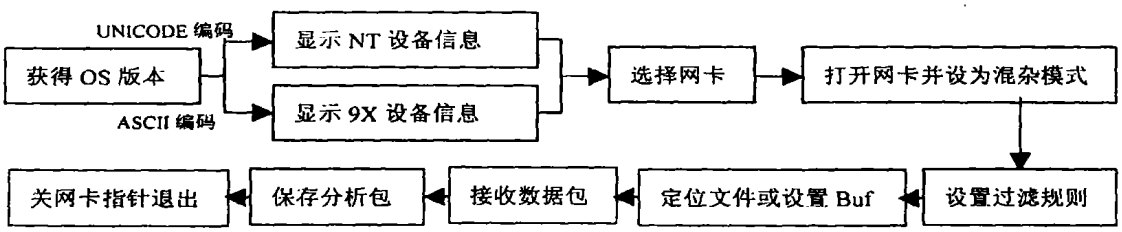

图 2 实验的 Sniffer 的一般流程

1) 打开网卡列表 DevPointer＝pcap_findalldevs(&AllDevsList, ErrBuf).其中,ErrBuf 是出错缓存,用来写入出错信息;而 AllDevsList 是一个用来描述网卡数据结构的 pcap_if_t 型指针.值得注意的是,在实际捕获数据包之前必须调用 pcap_freealldevs(AllDevsList)来释放该指针.对于多穴机而言,此步旨在选择要检测的网卡接口,并返回 pcap_if_t 型指针.

2) 打开对应的网卡 DevHandle＝pcap_open(DevPointer－＞name, 65536, 1, 1000, ErrBuf).在有些系统中,第一个参数为 NULL 表示监视所有网卡.第二个参数表明了要捕获数据包的长度,65536 保证了所有数据包能被捕获.参数"1"表示设置成混杂模式,"1000"表示阻塞时间是 1 秒.该函数返回一个指向网卡的设备句柄 DevHandle,它是后面几个函数的关键参数.

3) 设置过滤规则,可用 pcap_compile()和 pcap_setfilter()完成,以过滤掉自己不感兴趣的数据包,保证不会因为数据包总量太多而丢失需要监视的数据包.

4) 文件的读写.在一个繁忙的网络中,可能在一秒钟内得到上千个数据包,故有必要将数据存放在文件中以备将来分析使用.

①打开文件由 DumpFile＝ap_dump_open(DevHandle, Path)实现,其中的 Path 存放要打开文件的完整路径.返回值 DumpFile 是一个与设备句柄 DevHandle 相关联的文件句柄.

②调用 pcap_dump(DumpFile, PktHeader, PktData) 将数据存储到文件.其中 DumpFile 是 u-char 型指针,即是上面提到的文件句柄.PktData 是一个 u_char 型指针,指向原始数据包.该函数在 PacketHandle()内部被调用.

③用 pcap_open_offline()打开一个堆文件,之后用 pcap_loop()循环从文件中读取数据.通过这种方式,读取脱机的数据几乎和实时从网卡上读取数据一模一样.

④程序退出前,必须调用 pcap_dump_close(DumpFile)来关闭所关联的文件句柄.

5) 捕获数据包 pcap_loop(DevHandle, 0, PacketHandle, DumpFile),并将接收的数据包写入一个文件(用 pcap_dump()实现),便于以后分析.一般有两种捕获方法:

①用 pcap_loop()或是 pcap_dispatch().两者功能十分相似,不同的是前者在没有数据包到达时将阻塞,而后者可不被阻塞.

②循环调用 pcap_next_ex 或 pcap_next(),前者只在 Win32 环境下才能够被调用,而后者则效率很低,因为它隐藏了回调方法,并且不能够识别文件结束标志 EOF,所以对来自文件的数据流它几乎无能为力.

6) 解析数据包由回调函数 PacketHandle(Param, PktHeader, PktData)实现.其中 Param 是 u_char 型指针,多用于传递参数,如在 PacketHandle 函数中调用 pcap_dump()函数就用它来传送文件句柄 DumpFile.PktData 是一个 u_char 型指针,指向原始数据包.

除去出错处理、校验和验证等必要的附加程序外,每层协议处理模块的程序结构都是一个多分支结构.例如,L1 层以太包处理函数 EtherHandle(char *)的关键部分如下:

```
EtherHeader＝(ETH-HEADER * )CharPointer;
SwapETHHeader(EtherHeader);
switch (EtherHeader－＞frm_typ)
{ case ARP_PACKET: //ARP & RARP
case RARP_PACKET: ARPHandle(CharPointer); break;
case IP_PACKET: IPHandle(CharPointer); break; //IP
……
}// switch (eth－＞frame_type)
```

在调用 EtherHandle()之前,有一个编程技巧就是用一个 char 型指针 CharPointer 指向原始数据包,

即 CharPointer＝(char ＊)PktData．随后所有各层协议处理函数都引用这个指针，以便在上层协议在需要下层协议相关信息时(例如计算 TCP 校验和时需要 IP 头部信息)可以方便地获取，不过这种做法打破了协议之间的独立封装性．在 EtherHandle()中首先用以太包头指针 EtherHeader 指向数据包头，以便对以太包做进一步的处理．因为原始数据包的字节顺序是网络字节顺序即 Big－endian，而与主机字节顺序有可能不同，原因是主机字节顺序与具体 CPU 有关，如 X86 系列的 CPU 就是 Little－endian ，因此要用 SwapETHHeader (EtherHeader)来交换字节顺序．随后根据“帧类型”的域值(EtherHeader－＞frm＿typ)来判断该将数据包交给哪个模块作进一步的处理：如果是 IP 包就交给 IP 模块处理，如果是 ARP 包就交给 ARP 模块处理．因为 ARPR 及 RARP 包头格式一样，只是域值不同而已，故可以将这两个数据包放在一起进行分析．IPHandle(CharPointer)的功能就是要提取以太包的 Data (即 IP 包)交由 IP 处理模块处理．该函数首先由 IpHeader＝(IP＿HEADER ＊)(CharPointer＋14)取得 IP 数据包头，随后对此 IP 数据包进行解析，其结构与 EtherHandle()大体相同．数据包就是这样从下层至上层被依次解包而解析的．

编译之前，要将 WinPcap 包中 include、lib 目录下的文件分别拷贝到 VC 对应的文件中，还需要在工程中加入宏 WPCAP．为了使程序链接成功，还必须包含头文件 winsock2．h、pcap．h 及连接库 wpcap．lib、ws2-32．lib、wsock32．lib．调试过程中为防止结构体产生空洞而与实际的网络通讯不一样，可以在工程中加入＃pragma pack(1)．

3　优化及不足

为提高性能，本文使用了两个线程来实现数据包的捕获及分析：首先在程序中建立一个公共的 LILO 缓冲池，捕获线程将过滤后的数据包添加到队列的头部；另一个分析线程则专门根据头部信息分析数据包．考虑到应尽可能少地丢失数据包，故必须设置捕获线程为较高优先级．

为了使本文实现的 Sniffer 具有一定的攻击力，实现中加入了发送数据包的“发包”功能，这是用 pcap-sendpacket (DevHandle， SendBuf， SendBufLen)来实现的．

本文所介绍的 Sniffer 代码在 Windows XP 及 VC＋＋ 6.0 环境下成功地得到了调试和运行．其运行结果表明，此实现具备了 Sniffer 的捕获包、分析包、发包的基本功能，但尚未实现支持所有的 TCP/IP 协议及 IPX、DECNet 等其他非 TCP/IP 族协议，故有待进一步地完善．

参考文献：

[1] W．Richard Stevens，范建华，等．TCP/IP 祥解 卷一：协议[M]．北京：机械工业出版社．

[2] sniffer 技术原理及应用．http://dev．csdn．net/develop/article/22/22362．shtm．

[3] 监听以太网(一～四)．http://www．vckbase．net/document/viewdoc/？ id＝729．

[4] 深入学习 sniffer，http://www．linuxeden．com/edu/doctext．php？ docid＝911 ．

[5] sniffer 安全技术专题，http://www．chinaitlab．com/www/special/sniffer．asp．

[6] 手把手教你玩转 ARP 包，http://blog．csdn．net/superhacker110/．

[7] 用 WinPcap 实现 Sniffer，http://kaka．rootcn．com/shadowstar/essay/security/sniffer2．htm．

Sniffer's Principle Resolution and Realization in WinPcap

LIU Bo-tao[1]， ZHAO Gang[1]， FENG Cui-li[2]， TANG Le[1]

(1．SWPI，Xindu District，Chengdu 610500；2．Computer Science Department in ChangJiang University， 434102， China)

Abstract：On the basis of explained the Sniffer operation principle， the paper has expounded the general procedure of designing a Sniffer， and has realized the Sniffer's primary function in VC＋＋ combined with WinPcap．

Key words：sniffer；TCP/IP；WinPcap；VC＋＋