文章编号:1006-2475(2002)05-0011-13

基于 Winpcap 的网络嗅探程序设计

庄春兴,彭奇志

(江南大学青山湾校区信息工程学院,江苏 无锡 214036)

摘要:数据包捕获与嗅探技术是设计网络分析软件的基础,而 Winpcap 是在 Windows 系统中实现的一个优秀的包捕获架构,本文对该软件包的结构与功能进行了详细的分析,介绍了应用 Winpcap 设计嗅探程序的方法。

关键词:包捕获;网络嗅探;BPF;NPF;NDIS

中图分类号:TP393.07　　**文献标识码**:A

The Design of Network Sniffing Software Using Winpcap

ZHUANG Chun-xing, PENG Qi-zhi

(College of Information Engineering, Southern Yangtze River University, Wuxi 214036, China)

Abstract: Packet capturing and sniffing technology is the basis for designing network analysing software. Winpcap is one of the excellent packet capture architectures. This paper introduces the structure and functions of the software packet and presents the method of sniffer designing.

Keywords: packet capture; sniffer; BPF; NPF; NDIS

0 引　言

当今计算机网络及远程通信技术得到了广泛的应用,Internet 已经成功走进了学校、公司、政府部门乃至许多家庭,每天都有新的应用与技术产生。在这种情况下,计算机网络的设计、维护的难度日益增加。因此,人们对于能够分析、诊断及测试网络功能及安全性的工具软件的需求也越来越迫切。

网络嗅探技术一方面是黑客窃听网络的工具,另一方面也是设计网络分析与管理软件的基础。因此,许多科研组织、公司或个人都致力于对该技术的研究。网络嗅探可获得网络上传输的数据包,它是建立在数据包捕获的基础上的。目前,大多数 Unix 平台都有支持捕获数据包的内核模块,而 Windows 平台除 2000 带有一个非常简单的 IP 过滤器外,其它系统都没有包含该方面的功能,现有的一些包捕获系统要么功能有限(如 Netmon API),要么象 PCAUSA 接口一样是商业性质的软件包。Winpcap 是为数不多的功能强大且可免费获得的包捕获接口软件之一。

1 Winpcap 的结构分析

Winpcap 是由意大利人 Fulvio Risso 和 Loris Degioanni 等人提出并实现的。它的主要思想来源于 Unix 系统中最著名的 BSD 包捕获架构,Winpcap 的基本结构如图 1 所示。

Winpcap 由三个模块组成,一个是工作在内核级的 NPF 包过滤器;另外两个在用户级,即用户级的 wpcap.dll 模块以及一个动态连接库 packet.dll。

NPF 是架构的核心(在 Win95/98 中是一个 VXD 文件,在 NT/2000 中是一个 SYS 文件),它的主要功能是过滤数据包,在包上附加时间戳、数据包长度等信息。第二个模块 packet.dll 在 Win32 平台上提供了与 NPF 的一个通用接口,基于 packet.dll 的应用程序可以在没有重新编译的情况下用于不同的 Win32 平台如 Windows 95/98、Windows NT/2000。Packet.dll 还有几个附加功能,它可用来取得适配器名称、动态驱动器加载以及获得主机掩码及以太网冲突次数等。第三个模块 wpcap.dll 是通过调用 packet.dll 提供的函数生成的,它包括了过滤器生成等一系列可以被用户

收稿日期:2002-01-05

作者简介:庄春兴(1966-),男,江苏江阴人,江南大学信息工程学院讲师,研究方向:计算机网络安全。

级调用的高级函数，另外还有诸如数据包统计及发送功能。

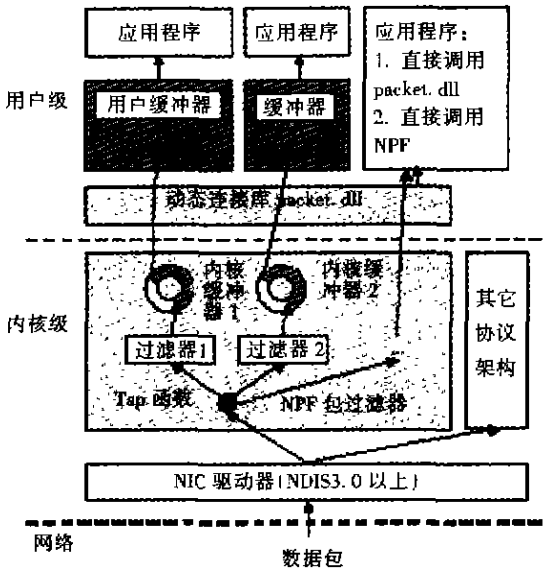

图 1 Winpcap 及 NPF 结构

整个包捕获架构的基础是 NDIS(网络驱动器接口规范)，它是 Windows 中最低端的与连网有关的软件，主要是为各种应用协议与网卡之间提供的一套接口函数，包驱动器的 tap 函数就是通过调用这些函数实现其数据采集功能的。

Wpcap.dll 模块与 Unix 系统下的 BSD 捕获架构提供的 Libpcap 库完全兼容，在功能上它还增加了统计及数据包发送功能。统计功能可以快速实现对网络数据的统计，如一定时间内流经总线的数据包数目，数据字节数等。发送数据包功能使得应用程序不仅可以嗅探网络，还可以实现向网络发送数据。在 Unix 系统下使用 Libpcap 编制的程序，经过重新编译后，可以在 Win32 平台上直接运行。由于这个原因我们通常用 Wpcap.dll 模块来设计嗅探程序。

2 Wpcap.dll 库函数介绍

Wpcap.dll 为包捕获应用提供了一个高级的接口程序，它是从 Libpcap 发展而来并且相互兼容，所以 Wpcap.dll 包含了所有 Libpcap 具有的函数。另外，由于它还增加了统计及数据包发送功能，所以还增加了以下的函数：

Pcap_setbuff：该函数用来设置包驱动器缓冲器的大小，一个适当大小的缓冲器不仅可以减少丢包率，还可以提高包捕获的速度。

Pcap_setmode：该函数可把网络适配器设置为统计方式。

Pcap_stats：这个函数用于获得包捕获过程的统计数据。

Wpcap 为了要与硬件接口，还直接调用了 Packet.dll 提供的函数，具体的内容可参考文献[2]。

另外，以下是 Wpcap.dll 和 Libpcap 中共有的几个主要的函数：

Pcap_open_live：这是最重要的一个函数。首先，它打开网络适配器，把网卡设置成“混杂”模式，使它能够接收来自网络的所有数据包；其次，它为应用程序设置一个缺省大小为 256kB 的缓冲器；最后，它为包捕获驱动器分配一个缺省为 1MB 的内核缓冲器，用户以后可以根据需要，改变该缓冲器的大小。

Pcap_read：这个函数从包捕获驱动器中读取一组数据包并针对每一个包运行包过滤程序，然后把过滤后的数据送应用程序缓冲器。

Pcap_setfilter：该函数在包捕获驱动器中设置一个新的包过滤器。过滤器程序则在一个名为 bpf_program 的结构中定义。

Pcap_loop：该函数可以连续读取并处理指定数目的包，对每个包进行相关的处理，处理程序名由所带参数指定，如果指定的数目为 0，则处理到包捕获过程的结束或者遇到了某个错误。

3 利用 Wpcap.dll 设计嗅探程序

在编制嗅探程序之前，首先必须从 Winpcap Web 站点下载 Winpcap 的自动安装程序，然后按照提示运行该程序，相应的包捕获驱动器架构就准备就绪了。

如果你的机器上已经安装好 Visual C++ 6.0 软件，接下来就可以开始编程。

下面是作者编写的一个用来在屏幕上显示数据包部分数据的非常简单的包捕获程序。

```
# include <stdlib.h>
# include <stdio.h>
# include <pcap.h>
# define MAX_PRINT 80
# define MAX_LINE 16
void dispatcher_handler(u_char *, const struct pcap_pkthdr
*, const u_char *);
void main(int argc, char **argv)
{
pcap_t *fp;
char error[PCAP_ERRBUF_SIZE];
if (argc < 3)
{    printf("\n\t pktdump[-n adapter][-f file_name]\n
     \n");
```

```
        return;
    }
    if ( (fp= pcap_open_live(argv[2], 100, 1, 20, error) ) ==
= NULL) //打开并初始化驱动器
    {   fprintf(stderr," \ nError opening adapter \ n");
        return;
    }
    pcap_loop(fp, 0, dispatcher_handler, NULL); // 读取并显
示
    }
    void dispatcher_handler(u_char * temp1, const struct pcap_
pkthdr * header, const u_char * pkt_data){
    u_int i=0;
        //显示数据包时间戳和长度
    printf("%ld:%ld (%ld) \ n", header->ts.tv_sec, header-
>ts.tv_usec, header->len);
    while ( (i< MAX_PRINT) && (i< header->len) )
    {   i++;
        printf("%x ", pkt_data[i-1]);
        if((i%MAX_LINE) == 0) printf(" \ n");
    }
        printf(" \ n \ n");
    }
```

本程序在机器上已经过编译,并能正确运行。证明了 Wpcap.dll 是一个简单但功能强大的包捕获工具。

4　包捕获应用程序的开发展望

包捕获应用程序不仅仅能用来做像上述程序一样的嗅探工作,我们还可以对捕获的数据包进行更加深入地分析,如黑客可用来分析网络中 SMTP 数据包,截取用户的邮件内容,也可以用来分析网络中的口令信息,并与其它手段结合起来,入侵到相关的主机。对一些已知的网络攻击方法,我们还可以编制专门的分析程序,监听该类型的数据包,找到攻击者,及时保护网络安全。另外,我们也可以用它来监测网络中的数据流量,分析网络故障。网络包过滤、网络日志等也是相关的应用领域。

总之深入研究网络嗅探技术,可以帮助我们找到网络中存在的问题,更好地进行网络维护及安全管理。

参考文献:

[1] 庄春兴,彭奇志. 网络窃听及其防范[J]. 网络安全技术与应用,2001,7.

[2] Fulvio Risso, Loris Degioanni. Development of an Architecture for Packet Capture and Network Traffic Analysis[DB/OL]. http://netgroup-serv.polito.it/winpcap, 2000-05-02.

[3] S McCanne and V Jacobson. The BSD Packet Filter: A New Architecture for User-level Packet Capture[A]. Proceedings of the 1993 Winter USENIX Technical Conference[C]. San Diego, CA, 1993, 1.

[4] V Jacobson, C Leres and S McCanne. TCPdump[R]. Lawrence Berkeley Laboratory, Berkeley, CA, 1989, 6.

[5] Gary R Wright, W Richard Stevens. TCP-IP(Volume 2, Chapter 31)[A]. Addison-Wesley Professional Computing Series[M], 1994.

文后参考文献表编排格式

参考文献按在正文中出现的先后次序列表于文后:表上以"**参考文献:**"(左顶格)作为标识:参考文献的序号左顶格,并用数字加方括号表示,如[1]、[2],…,以与正文中的指示序号格式一致。每一参考文献条目的最后均以"."结束。各类参考文献条目的编排格式及示例如下:

1. 专著、论文集、学位论文、报告

[序号]主要责任者. 文献题名[文献类型标识]. 出版地:出版者,出版年. 起止页码(任选).

[1] 刘国钧,陈绍业,王凤翥. 图书馆目录[M]. 北京:高等教育出版社,1957. 15~18.

[2] 辛希孟. 信息技术与信息服务国际研讨会论文集:A 集[C]. 北京:中国社会科学出版社,1994.

[3] 张筑生. 微分半动力系统的不变集[D]. 北京:北京大学系数学研究所,1983.

[4] 冯西桥. 核反应堆压力管道与压力容器的 LBB 分析[R]. 北京:清华大学核能技术设计研究院,1997.

2. 期刊文章

[序号]主要责任者. 文献题名[J]. 刊名,年,卷(期):起止页码.

[5] 何龄修. 读顾城《南明史》[J]. 中国史研究,1998,(3):167~173.

[6] 金显贺,王昌长,王忠东,等. 一种用于在线检测局部放电的数字滤波技术[J]. 清华大学学报(自然科学版),1993,33(4):62~67.

(下转第 15 页)