NDIS - HOOK网络封包截获技术实现

刘 静,裘国永

(陕西师范大学计算机科学学院,西安 710062)

摘 要:首先介绍了 NDIS(网络驱动接口规范)的基本概念,引出 NDIS的层次结构,简要介绍各个层次的功能。然后在此基础上介绍 NDIS - HOOK技术的特点、工作原理和两种实现方法,并对各种网络封包截获技术进行比较。经过测试,NDIS - HOOK技术可以截获所有网络封包。最后,对本次研究和开发工作进行总结,并对基于 NDIS - HOOK的网络封包截获技术的应用作了探讨和展望。

关键词:网络驱动接口规范;NDIS - HOOK技术;网络封包;截获

中图分类号: TP393.08 **文献标识码**: A **文章编号**: 1002 - 2279(2008)05 - 0051 - 03

The Implementation of Network Packets Capturing Technology based on NDIS - HOOK

LIU Jing, QIU Guo - yong

(SNNU School of computer science, Xi'an 710062, China)

Abstract: This paper introduces basic conceptions of NDIS (Network Driver Interface Specification) and its system hierarchy and function. Then, based on this foundation, this paper introduces technology characteristic, operational principle and two implementation methods of NDIS - HOOK. Also, this paper compares several network packets capturing technology. It can capture all network packets and functions well. In the end, solutions are presented and perceptions as well as future work of NDIS - HOOK based network packet capturing are discussed.

Key words: NDIS; NDIS - HOOK technology; Network packets; Capture

1 引 言

随着计算机技术和通信技术的飞速发展,因特网迅速普及。网络信息系统已渗透到社会生活的各个领域,全球信息化已成为人类发展的大趋势。随之而来的安全问题也日益严重,一旦网络安全问题发生,可能会带来非常严重的后果。例如重要信息和数据的窃取,计算机资源的破坏等,将会给政府和企业蒙受巨大的损失。

为了有效的防范网络攻击,我们必须采取行之有效的措施来保障信息安全,防火墙当然作为人们首选的安全防卫工具,防火墙基本上是基于对数据包的截获技术实现的。当然在具体的实现方式上有很大的不同,总的来说可分为用户态和核心态数据包截获技术。其中用户态截获数据包包括 SPI接口,Winsock API HOOK技术等;而核心态主要是 TDI传输驱动程序,NDIS中间层驱动程序,NDIS - HOOK钩子驱动程序,这些都是利用网络驱动程序来实现的。

在设计实现中,重点讨论如何实现利用 NDIS - HOOK技术来截获网络封包,分析 NDIS - HOOK技术截获网络封包的技术特点、实现原理,以及与其他截获网络封包方法的比较。

2 NDIS系统结构

NDIS(Network Driver Interface Specification)是 Microsoft和 3Com公司开发的网络驱动程序接口规范。它为 Windows下网络驱序程序的开发带来许多方便,编写符合 NDIS规范的驱动程序时,只要调用 NDIS函数,而不用考虑操作系统的内核以及与其他驱动程序的接口问题,为操作系统对不同网络的支持提供了方便。

Windows使用 NDIS函数库实现 NDIS接口,所有的网络通信最终必须通过 NDIS完成。NDIS负责上下层驱动程序间服务原语与驱动程序入口之间的转换,分派消息通知,保证符合 NDIS的驱动程序无

作者简介:刘静(1983 -),女,河南省新乡市辉县人,硕士研究生,主研方向:网络安全,数据挖掘。
收稿日期: 2006 - 12 - 01

须知道其他驱动程序的入口就可以与之通信。NDIS横跨传输层、网络层和数据链路层。

NDIS现在已经发展成为网络标准的一个完整家族,为网络驱动的开发提供了一套标准接口,目前最新的NDIS是5.1版本,Windows2000及以后版本的NDIS是5.0,在本文的开发中使用NDIS5.0。

NDIS提供三个层次的接口:网络接口卡驱动程序(Miniport Network Interface Card drivers),中间层驱动程序(Intermediate driver,简称IMD)和协议驱动程序(Protocol driver),层次结构[2]见图1。

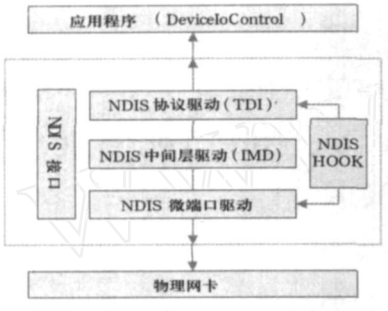

图1 NDIS层次结构图

NDIS网络接口卡驱动程序可以通过NDIS接口来完成对网卡的操作,实现网卡驱动,同时开放Miniport接口供上层驱动程序使用。协议驱动程序开放Protocol接口供底层程序调用,实现Protocol接口与Miniport接口的对接,可以用协议驱动程序来完成协议驱动。中间驱动程序处于小型端口驱动程序和协议驱动程序之间,同时具有两种驱动程序接口。

1)网络接口卡驱动程序(Network Interface Card Drivers)[2]

网络接口卡驱动程序管理网络接口卡,NIC驱动程序在它的下端直接控制网络接口卡硬件,在它的上端提供一个较高层的驱动能够使用的接口,这个接口一般完成以下一些任务:初始化网卡,停止网卡,发送和接收数据包,设置网卡的操作参数等等。

2)中间层驱动程序(Intermediate Drivers)[3]

中间层驱动程序在协议驱动程序和微端口驱动程序之间。在高层的传输层驱动程序看来,中间层驱动程序像一个微端口驱动程序,而在底层的微端口驱动程序看来,它像一个协议驱动程序。使用中间层驱动程序的最主要原因可能是在一个已经存在的传输层驱动程序和一个使用新的,传输层驱动程序并不认识的媒体格式的微端口驱动程序中相互转换格式,即充当翻译的角色。

利用NDIS中间驱动程序,可以在网卡驱动程序与传输驱动程序之间插入一层自己的自定义驱动程序(类似Tcpip.sys),从而可以截获网络封包,并重新进行封包、加密、网络地址转换、过滤、认证等操作。由于NDIS中间驱动程序位于网卡与传输驱动程序之间,所以它可截获较为底层的封包,从而可以完成更为低级的操作。但是也正因为如此,它有一个弱点,就是编程接口复杂,而且编写出来的驱动自动化安装太困难,很容易造成整个网络瘫痪,所以目前个人防火墙产品还很少用到这种技术来截获网络封包。

3)协议驱动程序(Protocol Drivers)[5]

开放Protocol接口供底层驱动程序调用,实现Protocol接口与Miniport接口的对接。协议驱动程序执行具体的网络协议,如IPX/SPX,TCP/IP等,协议驱动程序为传输层提供Miniport接口,接收来自网卡或中间驱动程序的信息。

3 NDIS-HOOK技术设计与实现

3.1 NDIS-HOOK技术介绍

NDIS-HOOK可以有两种方法。一种是PE HOOK[1],通过修改NDIS.SYS的Export Table来实现。在Windows2000/XP下,可执行文件(包括DLL以及SYS)都是遵从PE(Portable Executable)格式的。所有向其他操作系统组件提供接口的驱动程序都有一个导出函数表,因此只要修改NDIS.SYS的导出函数表就可以实现对关键NDIS API的挂接。

另外一种是注册假协议(fake protocol)[1],当协议驱动调用NdisRegisterProtocol之后,NDIS总是会把新注册的协议放在协议链表的表头并返回这张表,所以只要我们注册一个新协议,通过新注册协议返回的链表头就可以轻而易举的遍历系统中所有协议表。但是,如果要成功地挂接派发函数,还需要对协议所对应的NDIS_OPEN_BLOCK结构里的派发函数进行挂接。因为NDIS并不是直接调用协议驱动在NDIS_PROTOCOL_CHARACTERISTICS所注册的派发函数地址,而是调用NDIS_OPEN_BLOCK里的派发函数。

NDIS-HOOK技术有以下特点[4]:①编程方便、接口简单、思路明确、性能稳定;②更灵活,可以仅仅截获自己需求的,不需要冗余的代码;③功能强大,可以截获所有NDIS和TDI函数完成的功能。当然比标准方式的功能强大许多。还可以用这项技术延伸到HOOK的所有系统函数;④安全性高,因为它是比较底层截获封包,不容易被穿透;⑤安装简单,利用DDK中的snetcfg.exe可以实现程序自动化安装。

3.2　NDIS-HOOK工作原理

NDIS-HOOK的工作原理是直接替换 NDIS函数库中的函数地址,这样只要发向 NDIS的请求就会先经过我们自己函数的处理,这样就非常简单,处理完转发给系统函数就完成了。

NDIS-HOOK安装前的结构示意图如图 2所示。

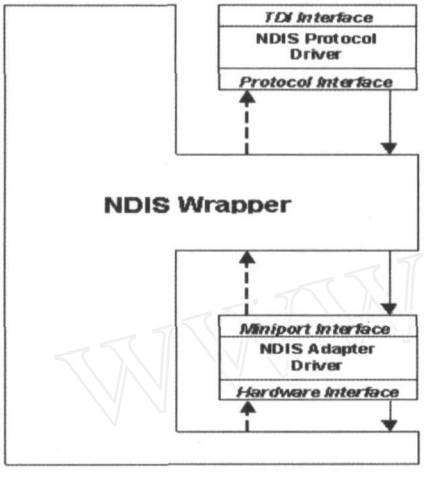

图 2　NDIS-HOOK安装前的 NDIS结构[6]

NDIS-HOOK安装后的结构示意图如图 3所示。

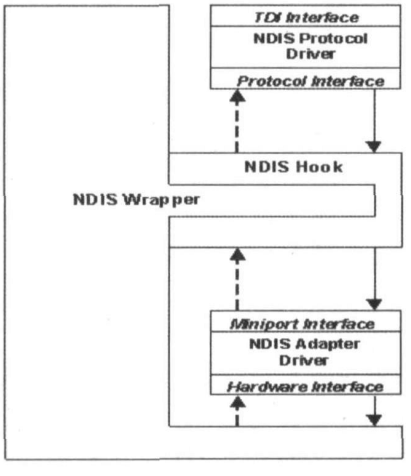

图 3　NDIS-HOOK安装后的 NDIS结构[6]

3.3　NDIS-HOOK的实现

3.3.1　PE HOOK的实现

在 PE HOOK的实现中,替换了 _NDIS_PROTOCOL_BLOCK中保存的发送和接收函数地址,主要完成对 NdisSend、NdisOpenAdapter、ProtocolSend和 ProtocolReceive函数的 HOOK,由于 ProtocolRecvive是在 RegisterProtocol时注册的,所以需要对 NdisRegisterProtocol函数进行 Hook。在 Windows2000/XP下协议驱动不再利用 NdisSend发送数据,而是通过协议驱动与网卡绑定后 NDIS为其分配的 SendHandler来进行数据发送,所以还需要对这个 SendHandler进行 HOOK。通过函数地址的替换,可以截获较为底层的网络封包,为数据包的分析提供条件。详细的 PE HOOK实现流程如图 4所示。

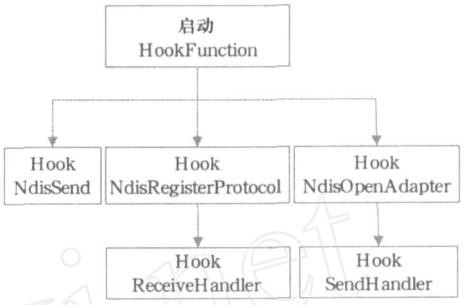

图 4　PE HOOK实现流程

3.3.2　注册假协议 HOOK的实现

把中间层驱动和注册假协议 HOOK结合起来使用,通过注册假协议并对整个协议链表注册协议的 HOOK替换发送和接收函数地址,这样可以先使用我们自己的协议驱动程序来截获网络封包,然后再交给原有的协议驱动处理函数处理。下面给出注册假协议的 HOOK实现流程,如图 5所示。

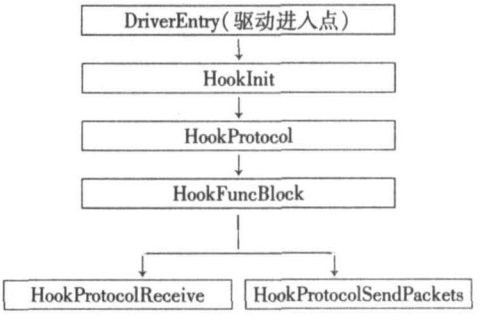

图 5　注册假协议 HOOK实现流程

3.4　网络封包截获技术比较

下面对目前已有的几种截获网络封包的技术进行比较,包括核心态和用户态技术,对它们的工作层次,截获封包的层次,安装方法以及截获方式进行比较,如表 1所示。

4　结束语

NDIS-HOOK技术可以很好的实现网络封包的截获。测试表明利用 NDIS-HOOK技术可以截获所有网络层的封包,而对于像 Winpcap这类工具采用 TDI技术,不能截获网络层的封包,例如它不能截获 ICMP包。网络封包截获技术作为防火墙最基本的技术,必然会得到重视和发展,采用先进的网络封包截获技术对于防火墙乃至网络安全来说都有其重要的应用价值。NDIS-HOOK技术有许多优越性,必然会在今后的防火墙,入侵检测以及 VoIP等更多领域得到长足的发展。　(下转第 56页)

取得一个样本;发送一个包;接收一个包。第二项是做这个动作的时间。当不需要做其中的任何动作时,节点可以处于睡眠状态,节省能量,仅当需要时节点才被唤醒。具体算法中当确保在初始阶段建立进度表必需的控制包不与在稳定阶段发送的数据包冲突的时候,就需要 MAC层协议保证通信链路的畅通。仿真的结果显示在同样的条件下没用能量节省算法的网络生命期如果是 8.3天,那么用了能量节省算法的网络生命期就能达到 24.2月。

3.3 链路层、MAC层、网络层的联合

文章[10]中联合了链路层、MAC层和路由层以及硬件设计能量消耗最小化方案,这个方案使用了3.1中提到的变长 TDMA作为底层支持,当由这个 TDMA方案分配时间片给相应的节点时,这个节点就处于激活模式,完成了数据发送,它就关闭所有的电路处于睡眠模式,节省能量。结果显示这几层的联合能量有效性设计比传统的基于分层结构的 MAC层和路由层的设计节省很多能量,同时在文章中还指出 MAC层和路由层联合自适应链路层的组合比 MAC层和路由层联合固定链路层的组合节省73%的能量。

4 结 束 语

本文中说明了使用 Cross-layer协议结构来达到能量有效性。Cross-layer通过网络状态信息数据库实现数据共享,统一实现协议栈的交互,达到减少整个网络能耗的目的。然而,无效的 Cross-layer交互也是很可能发生的,因为每一次的修改是要贯穿整个协议栈的,这就有可能带来一些意想不到的交互,例如:循环修改等,这样会导致网络性能的降低。另外,哪几层的交互更加有效也是一个未知的问题,所以在设计中克服和探究这些潜在的问题还需要进一步的研究。

参考文献:

[1] 任丰原,黄海宁,林闯. 无线传感器网络[J]. 软件学报, 2003, 14(7): 1282-1291.

[2] 李建中,李金宝,石胜飞. 传感器网络及其数据管理的概念、问题与进展[J]. 软件学报, 2003, 14(10): 1717-1727.

[3] Sanjay Shakkottai, Theodore S Rappaport, Peter C Karlsson, Cross-layer Design for Wireless Networks[J]. IEEE Communications, October 2003.

[4] Ahmed Safwat, Hossam Hassanein, and Hussein Mouftah. Optimal Cross-Layer Designs for Energy-Efficient Wireless Ad hoc and Sensor Networks[R]. IPCCC, 2003.

[5] Mihail L Sichitiu. Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks, Proceedings[J]. IEEE INFOCOM, v3, IEEE INFOCOM 2004-Conference on Computer Communications-Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies, 2004: 1740-1750.

[6] XU Li, ZHENG Bao-yu, Study on Cross-layer Design and Power Conservation in Ad Hoc Network[R]. PDCAT Proceedings, 2003: 324-328.

[7] Wendi Beth Heinzelman, Application-Specific Protocol Architectures for Wireless Networks[D]. Ph. D. thesis, 2000.

[8] Marco Conti, Gaia Maselli, Giovanni Turi, Silvia Giordano. Cross-layering in Mobile Ad Hoc Network Design[J]. IEEE Computer Society, February 2004.

[9] Shuguang Cui, Andrea Goldsmith, and Ahmad Bahai. Joint modulation and multiple access optimization under energy constarints[R]. at proceedings of Globecom'04, 2004.

[10] Shuguang Cui, Ritesh Madan, Andrea J Goldsmith, and Sanjay Lall. Joint Routing, MAC, and Link Layer Optimization in Sensor Networks with Energy Constraints [R]. to appear at ICC'05, South Korea, May, 2005.

(上接第 53页)

表 1 网络封包截获技术比较

比较内容 / 截获技术	工作层次	截获封包层次	安装方法	截获方式
TDI	传输层	传输层以上	自动	监听,共享
中间层驱动	数据链路层和网络层之间	所有(以太网就是 MAC帧)	手动	阻塞,过滤
NDIS-HOOK	数据链路层和网络层之间	所有(以太网就是 MAC帧)	自动	阻塞,过滤
SPI	传输层	传输层以上	自动	阻塞,过滤
Winsock API HOOK	应用层	针对 socket通信	自动	监听,共享

参考文献:

[1] 朱雁辉. Window防火墙与网络封包截获技术[M]. 北京:电子工业出版社, 2002.

[2] Microsoft Windows 2000 Driver Development Kit[S]. Microsoft Press, 2001.

[3] 易青松. 基于 NDIS的网络监控系统的设计与实现[J]. 计算机工程与设计, 2006, 27(15): 2816-2817.

[4] 高泽胜,陶宏才. 基于 NDIS-HOOK与 SPI的个人防火墙研究与设计[J]. 计算机应用研究, 2004(11): 279-281.

[5] 吴莹,彭丽芳. Windows下开发网络嗅探器程序[J]. 微处理机, 2002(2): 32-33.

[6] 驱动程序开发网技术社区[EB/OL]. http://bbs.zndev.com/.