基于 NDIS 构造 Windows 下入侵检测系统技术分析*)

张岳公　刘　伟　李大兴
(山东大学网络信息安全研究所　济南 250100)

摘　要　文中介绍了 Windows 平台下基于 NDIS 中间层技术实现入侵检测的方法。并与一般的使用 WinPcap 技术实现入侵检测系统进行了比较,NDIS 中间层技术在功能和性能方面具有比较突出的优势。我们采用该技术实现了一个模型系统,试验证明它具有较高的性能。

关键词　入侵检测系统,NDIS,中间层驱动

Analysis of Implementing of IDS Based on NDIS in Windows

ZHANG Yue-Gong　LIU Wei　LI Da-Xing
(Institute of Network Information Security, Shandong University, Jinan 250100)

Abstract　An implementing method of IDS Based on NDIS is stated in this article, and the differences between methods which based on NDIS and that based on WinPcap are discussed. By comparison, the approach based on NDIS is more flexible and has higher performance than WinPcap. We have implemented a model system to prove our opinion.

Keywords　Intrusion detection system, NDIS, Intermediate layer driver

1　引言

入侵检测(Intrusion Detection, ID)是指监控并分析计算机系统或者网络上发生的事件,以寻找入侵迹象的过程。所谓的入侵可以被定义为一系列试图损害数据完整性、机密性和可用性的企图。入侵检测系统(Intrusion Detection System, IDS)是对入侵自动进行监控和分析的软、硬件产品。

入侵检测系统可以根据其数据源划分为主机入侵检测系统和网络入侵检测系统两种。它们各有其侧重点和优缺点,在一个安全设计严密,部署得当的系统中,主机入侵检测系统与网络入侵检测系统是相辅相成,缺一不可的。

主机入侵检测系统根据操作系统不同有不同的实现方式。当今,Windows 操作系统占据了桌面操作系统的主流。所以,探讨 Windows 平台下入侵检测系统的实现方法具有很重要的现实意义和应用价值。

在 Windows 操作系统与在 Unix、Linux 等操作系统上实现主机入侵检测的主要差别在于网络数据包的截获机制上。由于在数据包分析算法方面,已经有比较成熟的算法,所以在进行 Windows 平台入侵检测系统的设计时,我们主要考虑网络数据的截获问题。

目前,多数设计采用 WinPcap 作为网络数据截获工具,它具有免费、易用、避免二次开发等优点,同时,也存在着性能和效率低下的问题。与使用 NDIS 中间层技术实现的入侵检测系统相比,后者在整体性能和效率上具有明显的优势。

2　NDIS 与中间层技术

2.1　NDIS 体系结构

1989 年微软与 3Com 公司联合开发了网络驱动程序接口规范(Network Driver Interface Specification, NDIS),它可以以设备无关的方式实现协议驱动程序与网络接口卡驱动程序的通信。NDIS 结构如图 1 所示。

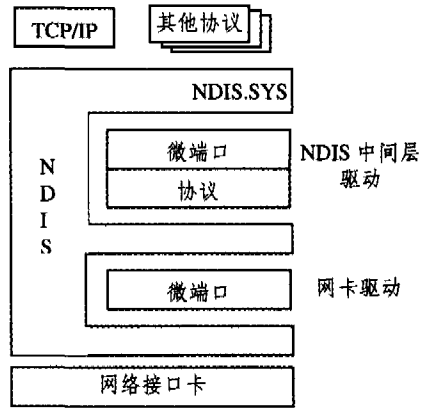

图 1

NDIS 库是一个帮助文件库,NDIS 驱动程序客户用它格式化发送给 NDIS 驱动程序的命令。通过 NDIS 驱动程序与 NDIS 库接口,实现了接收请求和返回应答。

NDIS 库为 NDIS 驱动程序提供了一个完整的运行环境,NDIS 支持三种类型的网络驱动:①微端口驱动(Miniport Driver):即网络接口卡驱动程序。②中间层驱动(Intermediate Driver):位于微端口驱动和协议驱动之间,用来实现数据包在不同网络介质间的转换,数据包过滤和多网卡数据流量的负载平衡。③协议驱动(Protocol Driver):实现具体的传输协议如 TCP/IP, IPX/SPX 等。

2.2　NDIS 中间层驱动

如图 1 所示,NDIS 中间层驱动位于微端口驱动和协议驱动之间。为了做到中间层驱动对传输协议驱动和网卡驱动的

*)基金项目:国家科技部"973"项目,项目编号:G1999035802。张岳公　博士研究生,主要研究方向:入侵检测,网络信息安全;刘　伟　硕士研究生,主要研究方向:入侵监测,网络信息安全;李大兴　教授,博士生导师,主要研究方向:密码学,网络信息安全。

透明，在逻辑上，将中间层驱动分成两层：上层的微端口驱动模块和下层的协议驱动模块。

对 NDIS 上面的 TCP/IP 及其它传输协议来说，NDIS 中间层驱动上层的微端口驱动模块表现为虚拟的网卡驱动；对底层的网卡驱动程序来讲，NDIS 中间层驱动下层的协议驱动模块表现为虚拟的协议层驱动。

NDIS 中间层驱动处于协议驱动和网卡驱动之间，可以直接对数据包进行过滤处理，因此可以用来实现诸如网络地址翻译(NAT)、防火墙(Firewall)、IPSec VPN(Virtual Private Network)、入侵检测等与网络安全相关的应用系统。

3 NDIS 中间层技术实现 IDS

3.1 实现原理

在 Windows 平台中实现 IDS，采用 NDIS 中间层驱动对网络数据包进行捕获，采用高效的模式匹配算法对数据包进行分析。设计结构图如图 2。

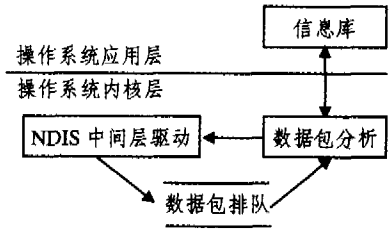

图 2

NDIS 中间层驱动中的协议驱动模块和微端口驱动模块分别完成网络数据包的收发操作。我们实现的中间层驱动程序名称为 IDSIMD，它与底层的网卡实施绑定，主机发送和接收的任何网络数据包都要经过被绑定的网卡和 IDSIMD 的处理。数据流图如图 3 所示。

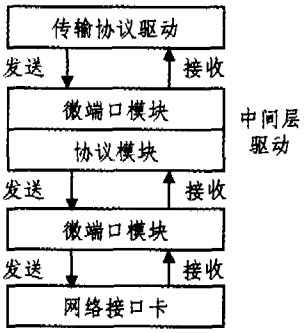

图 3

为了提高 IDS 的性能，降低丢包率，NDIS 中间层驱动在获取到完整的数据包后，直接放入数据包队列。如果是分片数据包，则进入分片重组队列，在数据包重组完成后放入数据包队列。

内核中运行的系统线程对队列中的数据包进行调度，对所有的数据包，系统线程将其派遣给内核中的数据包分析模块进行分析。内核中数据包分析模块通过与应用层的信息库进行交互，完成对数据包的分析。对有入侵特征的数据包，数据包分析模块将作丢弃处理，正常数据包将无害通过。

3.2 实现方法

3.2.1 *驱动模块注册* 我们实现了数据包截获功能的 NDIS 中间层驱动 IDSIMD。在其中的主例程 DriverEntry 例程中，定义了协议模块和微端口模块的数据结构：

NDIS_PROTOCOL_CHARACTERISTICS 对应于协议模块；

NDIS_MINIPORT_CHARACTERISTICS 对应于微端口模块。在这两个模块的数据结构中，定义了相应的数据包处理例程。

在微端口模块中，主要定义了如下例程的处理句柄：

MiniportSend：当上层的协议驱动程序要发送数据时，通过 NDIS 接口库调用此例程发送相应的网络数据包。

MiniportReturnPacket：在上层协议驱动程序接收到中间层驱动程序接收上来的数据包后，上层协议驱动程序通过调用 NDIS 接口库调用此例程，用来释放在数据包接收过程中分配给数据包的未分页内存。

在协议模块中，主要定义了如下例程的处理句柄：

ProtocolReceivePacket：在底层的网卡驱动程序接收到数据包后，通过调用 NDIS 接口库调用此例程进行数据包的接收。

ProtocolReceive：其作用如 ProtocolReceivePacket。此例程与 ProtocolReceivePacket 的不同之处在于：此例程是网卡驱动程序接收到完整的数据包或者部分数据包(不完整)的时候，通过 NDIS 接口库调用此例程，将接收到的数据包传送给中间层驱动，如果是中间层驱动感兴趣的数据包，需要调用 NdisTransferData，让网卡驱动程序将剩余的数据包接收上来。而 ProtocolReceivePacket 则是将整个数据包接收到中间层驱动程序中。从性能的角度看，使用 ProtocolReceivePacket 例程要比使用 ProtocolReceive 例程高效。值得说明的是，在新型的网卡如 Intel、3Com 的驱动程序中，此例程不会被调用；但是在使用比较广泛、价格低廉的 Rtl8139 网卡驱动程序中，此例程会被调用。

ProtocolSendComplete：在网卡驱动程序发送完数据包后，通过调用 NDIS 接口库调用此例程，用来释放在数据包发送过程中分配给数据包的未分页内存。

ProtocolTransferDataComplete：在 NDIS 接口库调用了 ProtocolReceive 接收到一个不完整的数据包时，为了得到整个数据包，需要调用 NdisTransferData 例程将剩余数据包接收上来，在接收操作完成后，NDIS 接口库调用此例程。

在定义了中间层驱动相关模块的处理例程后，调用 NdisRegisterProtocol 进行协议模块的注册，调用 NdisIMRegisterLayeredMiniport 进行微端口模块的注册。

3.2.2 *数据包截获* 在 NDIS 驱动程序中，数据的发送和接收都是以数据包为单位进行的。数据包通过包描述符来标识，其中包括数据包占用的物理页面的数量、数据包的长度、指向第一个和最后一个缓冲区描述符的指针等。而缓冲区描述符则描述一片存储区域，包括起始地址、存储区域大小、下一个缓冲区的指针等。

在数据发送或者接收的过程中，都是通过 NDIS 接口库调用注册时的相关处理例程，主要参数就是数据包的包描述符。通过包描述符，可以通过 NDIS 接口库的相关例程如 NdisGetFirstBufferFromPacket，NdisQueryBufferSafe 等得到数据包中的网络数据，将此网络数据存放在一块连续的未分页内存中，将原来的包描述符和相应的缓冲区描述符释放。

如果网络数据是一个完整的数据包，接下来重新构造一个缓冲区描述符和包描述符，将此数据包放入以链表形式构造的数据包 FIFO 队列，等待数据包分析模块的调度。

如果此网络数据是一个分片数据包，则将此未分页内存

的指针放入分片重组链表，重用分片包 IP 头中的 TTL 字段，设定重组定时器为 60 秒。此分片重组链表为一个十字双向循环链表，其结构图如图 4。在定时器时间范围内，如果分片重组失败，则丢弃并释放此数据包对应的整条分片链。如果重组成功，则构造一个缓冲区描述符和包描述符，并将此数据包放入 FIFO 队列。

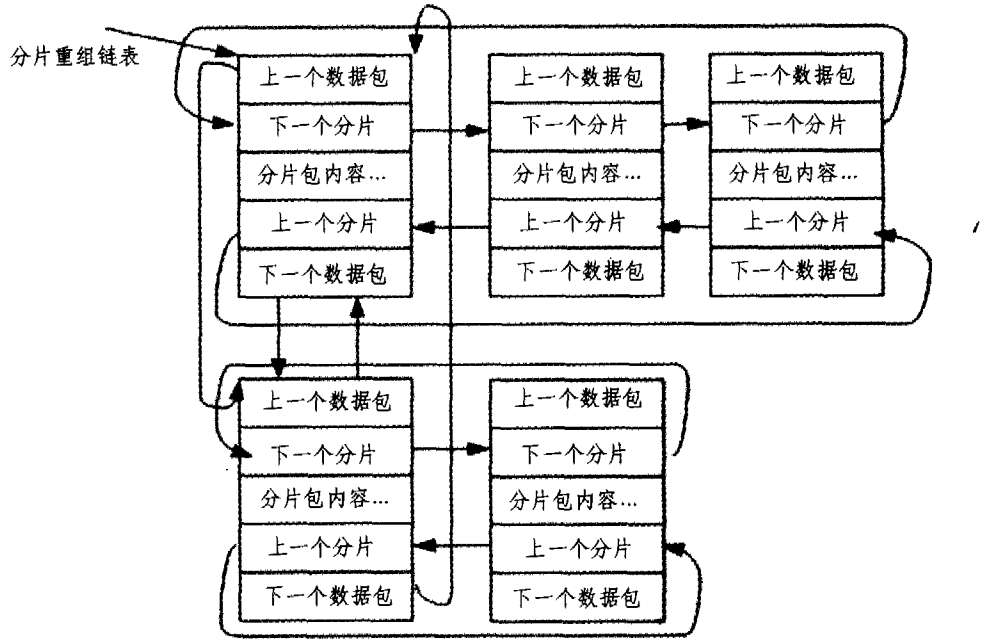

图 4

3.2.3 *数据包分析调度* 在内核中间层驱动 IDSIMD 的 DriverEntry 中，调用 PsCreateSystemThread 产生一个系统线程。此线程不断轮询数据包 FIFO 队列，一旦有数据包出现，立即转入数据包分析模块进行分析；如果队列中一直没有数据包，则在等待 10ms 后进入下一次轮询。

4 WinPcap 与 NDIS 实现 IDS 的比较

WinPcap 的设计思想来源于 Unix 系统中数据包的截获架构，它在 Windows 系统中的实现主要包括三个部分：一是工作在内核层的 NPF 数据包过滤器；另外两个是工作在应用层的 packet. dll 和 wpcap. dll。内核层的 NPF 数据包过滤器将截取到的数据包原封不动地传递给应用层，由应用层对数据包进行处理。

与使用 NDIS 中间层技术实现的 IDS 相比，使用 WinPcap 存在性能和效率上面的不足。具体表现在：

1)模块耦合度低：使用 WinPcap 技术实现 IDS，在模块耦合度上要低于 NDIS 中间层驱动的方式。这是由 WinPcap 的实现机制决定的，WinPcap 将截获的数据包不做任何改动的传递给应用层处理。而使用 NDIS 中间层驱动的方式，在内核中截获的数据包直接交给内核中的数据分析模块处理。不需要 WinPcap 内核与应用层通信的处理过程。

2)分片处理难题：使用 WinPcap 在对分片包进行处理时将面临不可避免的麻烦。WinPcap 智能将分片数据包交由应用层，由应用层对分片数据重组后才能进行入侵特征分析；否则，直接对分片数据进行特征分析容易产生误报。而 NDIS 中间层驱动对分片包在内核中直接进行重组，然后进行入侵分析。

3)丢包率和内存效率问题：WinPcap 为了降低丢包率，分配了固定为 1MB 的未分页内存，对数据包进行截获处理。这对于一般的网络流量是可以满足的，但是对于网络攻击等数据流量异常的情形，此 1MB 的未分页内存往往不能满足要求，势必造成丢包。如果网络流量很小，对于系统来说 1MB 未分页内存有些浪费。NDIS 中间层驱动对数据包进行排队处理的时候，使用链表的形式，对数据包所使用的未分页内存实行按需分配，极大地降低了丢包率，并提高了未分页内存的使用效率。

4)自主知识产权问题：WinPcap 的技术和源代码是开放的，在使用其开发产品的时候面临知识产权的问题。而使用 NDIS 中间层驱动技术实现 IDS，完全是我国的自主知识产权，不会受制于人。

结束语 本文中介绍的 Windows 主机入侵检测系统是我们基于 NDIS 中间层驱动技术实现的具有完全自主知识产权的系统。在实际使用中，其性能和效率明显高于使用 WinPcap 技术的入侵检测系统。但是，由于 NDIS 处于操作系统内核层，它的开发难度和调试难度要高于使用 WinPcap。并且，对于 Windows 不同架构的操作系统，其网络截获程序的具体实现也有所不同。如：在 Windows 9x/Me 中，我们采用 NDIS Hook 实现了 NE 格式的虚拟设备驱动程序；在基于 NT 架构的 Windows 2000/XP/2003 中我们使用 NDIS 中间层驱动实现了 PE 格式的驱动程序。

参 考 文 献

1 [美]Solomon A, Russinovich M E 著. 詹剑峰，张文耀，黄艳，等译. Windows 2000 内部揭秘. 北京：机械工业出版社，2001
2 Windows 2000 DDK
3 [美]Stevens R W 著. 范建华，胥光辉，张涛，等译. TCP/IP 详解，卷 1：协议. 北京：机械工业出版社，2000
4 [美]Wright G R, Stevens W R 著. 陆雪莹，蒋慧，等译. TCP/IP 详解，卷 2：实现. 北京：机械工业出版社，2000