第 36 卷 第 9 期 计 算 机 工 程 2010 年 5 月
Vol.36 No.9 Computer Engineering May 2010
·安全技术· 文章编号：1000—3428(2010)09—0150—03 文献标识码：A 中图分类号：TP393.08

# 基于网络驱动技术的木马通信检测系统

钟明全，李焕洲，唐彰国，张 健
(四川师范大学网络与通信技术研究所，成都 610066)

**摘 要：**为提高木马程序的网络通信检测率，在比较各种包截获技术优缺点的基础上，设计并实现一种基于 NDIS Hook 驱动的木马通信检测系统，给出主要模块和数据结构，提出基于网络通信行为分析技术的木马通信识别模型。测试结果表明，该模型能降低误报率和漏报率，可截获所有网络通信数据包，识别新的木马通信。

**关键词：**木马；包截获；NDIS Hook 驱动；木马通信识别

# Trojan Communication Detection System Based on Network Drive Technology

ZHONG Ming-quan, LI Huan-zhou, TANG Zhang-guo, ZHANG Jian
(Institute of Network and Communication Technology, Sichuan Normal University, Chengdu 610066)

**【Abstract】**To enhance network communication detection of Trojan program, this paper designs and realizes a Trojan communication detection system based on NDIS Hook drive on the base of comparing advantages and disadvantages of various packet capture technology. It gives main modules and data structures, proposes Trojan communication identification model based on network communication behavior analysis technology. Test results show that this model can decrease false positive rate and negative positive rate, acquire all network communication data packet and identify new Trojan communication.

**【Key words】**Trojan; packet capture; NDIS Hook drive; Trojan communication identification

## 1 概述

计算机网络应用不断丰富，给人们日常工作和学习带来很大方便。与此同时，各种病毒程序，尤其是木马程序通过计算机网络渗透到 PC 机中，窃取个人隐私或秘密信息，给信息系统或个人带来很大威胁。现有杀毒软件和个人防火墙在木马检测中发挥了重要作用，但是漏报率和误报率较高。本文采用内核态包截获技术，对木马工作周期中的网络通信阶段进行实时监视，通过对进程运行时产生的网络通信数据进行分析和判定，达到识别木马通信的目的。

## 2 网络通信数据包截获技术

### 2.1 TCP/IP通信协议与Windows系统分层对应关系

TCP/IP 通信协议层次见图 1，大致可分为 3 个层次：通信应用层，协议层和驱动层。通信应用层是一些网络应用程序，如 IE 浏览器、电子邮件收发程序、QQ 聊天软件等，通过调用 Socket 进行通信编程，运行在操作系统的用户态下。协议层是 TCP/IP 的通信数据的发送、接收、处理等实际通信功能的实现，工作在操作系统的核心态下。驱动层是完成 TCP/IP 通信数据在通信物理硬件上的发送和接收功能，也工作在操作系统的核心态下，但层次更低，优先级别更高。

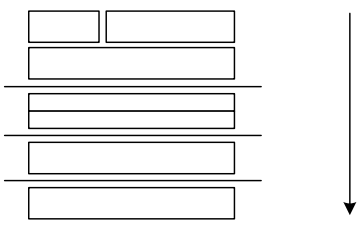

**图 1 TCP/IP 通信协议层次**

### 2.2 各种包截获技术及其比较

目前，常用的网络封包截获技术有 Winsock 2 SPI、NDIS 中间层驱动、NDIS Hook 驱动、WinPcap 等技术。

从工作层次方面，NDIS 中间层驱动、NDIS Hook 驱动、WinPca 都是工作在内核态的，可以截获所有通过网卡的数据包，其中，NDIS 中间层驱动、NDIS Hook 驱动对系统依赖性高；Winsock 2 SPI 工作在应用层，CPU 占用率低、效率高、程序可移植到微软新一代操作系统 Vista 下，但可能被底层的数据包绕过[1]。

从实现的难度方面，Winsock 2 SPI 是针对应用层中使用的 FTP, HTTP, POP3 和 SMTP 等协议的数据包进行操作，包较完整，容易提取包内容进行过滤。WinPcap 能够捕获原始数据包，收集网络通信过程中的统计信息，但不能将数据包和应用程序相关联[2]。NDIS 中间层驱动程序工作在网络层，下面是网卡驱动程序，驱动程序的设计较为底层，安装过于复杂。相对而言，NDIS Hook 驱动也工作在网络层，编程相对容易一些，并且安装比较简单。因此，本文系统实现选择 NDIS Hook 驱动技术。

## 3 基于NDIS Hook驱动的木马通信检测系统

### 3.1 模块结构

3.1.1 应用程序

应用程序实现通信数据的收集、加工与显示，其主要功

**作者简介：**钟明全(1975－)，男，讲师、硕士，主研方向：通信与信息安全；李焕洲，副教授、博士；唐彰国，讲师、硕士；张 健，讲师、博士研究生

**收稿日期：**2009-12-07 **E-mail：**mqzhong@sina.com

能模块结构如图 2 所示。

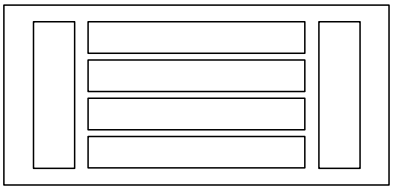

**图 2　应用程序功能模块结构**

由图 2 可知，封包采集模块采用循环工作模式，周期性地从封包缓冲区中读取最新通信数据；获取进程路径名模块根据驱动程序返回的进程 ID 从应用层获得进程的路径信息；流量统计模块对网络中的通信数据进行统计，获得 TCP 流量、UDP 流量和 ICMP 流量等信息；木马通信判定模块是本系统的关键模块，根据进程产生的通信数据，参照自定义判定规则，对进程危险等级进行判定；公共模块包含一些公用的数据结构和公共函数。

3.1.2　驱动程序

驱动程序实现网络通信原始数据包的截获、解析，并存放于内存空间中供应用程序读取，其主要功能模块结构如图 3 所示。

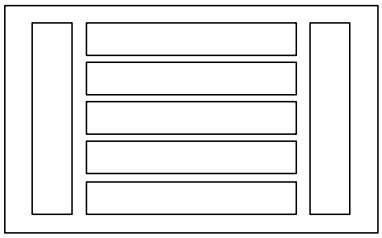

**图 3　驱动程序功能模块结构**

由图 3 可知，实现 Hook 模块用于 Hook 网络通信时的 ProtocolSendHandler 和 ProtocolReceive 系统核心函数，以实现网络封包的截获功能；协议解析模块根据网络封包结构对不同协议数据包进行解析，以获得网络通信使用的协议、IP 地址、端口等信息；获取进程名模块用于获取通信进程的进程 ID 和进程名信息；封包缓冲区处理模块用于申请内存空间，以存储截获的封包信息；通信接口模块实现应用程序对驱动程序的辅助控制，如通知驱动程序删除封包缓冲区中已经结束的封包数据等。

3.1.3　应用程序与驱动程序之间的接口

应用程序与驱动程序主要进行封包数据的共享，另外要进行一些辅助控制操作，它们之间的接口函数为 OnW32 Deviceiocontrol。应用程序通过 CreateFile 打开驱动程序，然后通过 DeviceIoControl 完成与驱动程序的通信。实现内存共享的原理是：首先在驱动程序中申请内存空间，由于在驱动程序中申请的内存空间位于系统地址空间(≥0x80000000)上，为让应用程序能够访问这个内存空间，需要将系统地址空间映射到用户地址空间(<0x80000000)。

### 3.2　木马通信识别模型

在 NDIS 接口中，网络封包结构常用 NDIS_PACKET 表示。对于不同 NDIS 版本，NDIS_PACKET 的结构内容略有不同，这里主要针对 Windows 2000 操作系统。NDIS_PACKET 的定义可以在 DDK 中找到，NDIS_PACKET 中保存一个 NDIS_BUFFER 结构的链表，封包数据用 NDIS_BUFFER 表示。NDIS_BUFFER 的成员变量 VirtualAddress 表示封包的缓冲区指针，Length 表示 VirtualAddress 的长度，Next 是下一个封包缓冲区，这些封包缓冲区的数据联合起来是一个完整的网络封包。在 Windows 2000 下，一般不能直接引用结构的成员变量，而是使用 Windows 2000 定义的 NdisQueryBuffer, NdisGetNextBuffer 等函数来得到有用的成员信息。

为检测木马程序网络通信，较好的方法是在木马程序工作的关键路径(网络层)截获其通信的数据包。木马程序为了达到窃取信息和控制的目的，其服务端(server)通常要与控制端进行网络通信，于是选择在操作系统底层编写 NDIS Hook 驱动程序获取网络通信数据。通过对截获的数据包按照以太网数据包结构进行解析，解析出通信过程中使用的协议、IP 地址、端口、封包大小、状态等信息，并将这些信息和通信的进程名关联起来，作为该进程的基本通信信息。对一些特殊的数据包，可以解析出通信中使用的域名信息，把它和进程关联起来。接着将解析出的信息存储于内存空间，应用程序以共享方式读取封包数据。根据预先定义的判定规则，由应用程序对网络通信数据进行流量统计和木马通信危险等级判定。整个木马通信识别流程如图 4 所示。

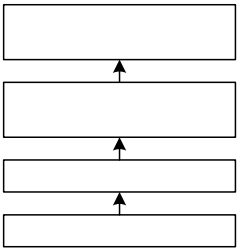

**图 4　木马通信识别流程**

在对常见木马经过深入研究和反复测试的基础上，对木马运行时的网络通信行为进行分析和归纳后，得到如下判定规则：

(1)如果进行网络通信的进程名与系统进程名(如 services.exe)相同，但其路径不在系统进程目录(system32)下，则等级为高。

(2)如果进行网络通信的进程名与系统进程名相似，如进程 svch0st.exe 与系统进程 svchost.exe 相似，则等级为高。

(3)如果进程运行时自动发送电子邮件，则等级为高。

(4)如果进程使用 ICMP 协议进行通信，则等级为中。

(5)如果进程主动向外进行连接，则等级为中。

第(1)条规则和第(2)条规则基于木马程序的进程隐藏原理进行判定；第(3)条规则基于木马程序传输信息的途径进行判定；第(4)条规则基于木马程序的通信隐藏原理进行判定，为了兼顾误报率与漏报率，在网络通信检测中危险等级定为中；第(5)条规则基于木马程序网络通信的端口及其状态进行判定[3]，同样为了兼顾误报率与漏报率，危险等级定为中。对于等级为中的进程需综合考虑其进线程行为数据、文件操作数据、注册表操作数据才能做进一步判定。

一个被检测程序可能同时匹配到以上多条规则，检测最终结果取其中最高等级。如果一个被检测程序不具有上文的 5 种情形之一时，则判定等级为低。需说明的是，随着被检测程序通信行为信息的增加，检测结论是动态变化的。

### 3.3　NDIS Hook驱动在木马通信检测系统中的实现

在 Windows 2000 操作系统下对木马通信检测系统进行实现，用到的开发工具有 VC++6.0、MSDN 2002 和 Windows 2003 DDK，同时使用 Windbg 和 Dbgview 作为调试分析工具。

检测系统的实现分为 2 个部分，其中 NDIS Hook 驱动程序的开发是一项较复杂的工作，可以利用微软 DDK 提供的例程作为参考，在其基础上进行改写。应用程序的实现难点在于将解析的各种信息准确地关联到对应进程上，并根据判定规则对该进程的网络通信行为数据进行判定，以得到被检测程序木马通信危险等级。因此，定义一个网络通信的数据结构，如表 1 所示。

**表 1　网络通信的数据结构**

| 进程名 | 进程 ID | 协议 | 目的 IP/端口 | 域名 | 源端 IP/端口 | 发送/接收 | 起止时间 | 状态 | 进程路径 |
|---|---|---|---|---|---|---|---|---|---|
| TCHAR | DWORD | TCHAR | TCHAR/WORD | TCHAR | TCHAR/WORD | DWORD/DWORD | TCHAR/TCHAR | TCHAR | TCHAR |

NDIS Hook 驱动通过修改 Ndis.sys 的导出表(Export Table)，实现对关键 NDIS API 的挂接[4]。由于协议驱动程序在系统启动时会调用 NdisRegisterProtocol 向系统进行协议注册，因此这种方法关键在于修改 Ndis.sys 所提供的 NdisSend, NdisRegisterProtocol/NdisDeRegisterProtocol, NdisOpenAdapter/NdisCloseAdapter 函数的起始地址[5]。图 5 为 NDIS Hook 的工作流程。

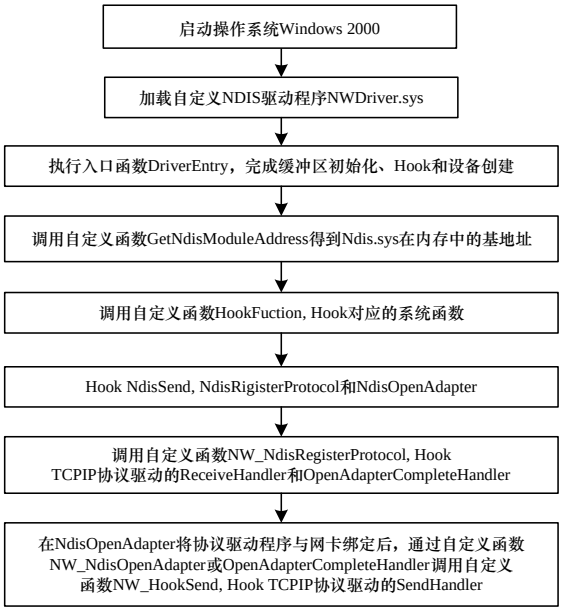


**图 5　NDIS Hook 工作流程**

由本系统的功能和网络通信数据结构可知，进程名是通信数据关联的主体，进程路径是判定木马通信危险等级的重要信息，然而驱动程序在底层获取进程名和路径名比较困难。其原因在于包含路径名的 EPROCESS 结构未公开，偏移地址错误或非法访问某些核心内存将导致计算机的蓝屏死机。因此，采取的解决方案是在驱动程序中获取进程 ID，然后在应用态根据返回的进程 ID 获得进程的路径名信息。

## 4　测试结果

由于被检测程序运行后可能会破坏主机操作系统，因此测试工作是在虚拟机操作系统中进行。测试环境的主要配置为：主机操作系统 Windows XP Professional SP2，虚拟机软件 VMware Workstation V6.0，虚拟机操作系统 Windows 2000 Professional SP4，虚拟机中安装常用应用软件如 Office, Adobe Reader, Winrar 等。为了检测被检测程序的网络通信，要求测试环境必须联网，主机上网方式为局域网代理上网，虚拟机选择桥接方式共享主机的 Internet 连接。木马通信检测系统运行主界面如图 6 所示。

| 序号 | 应用程序 | 进程ID | 协议/连线 | 目的IP:端口 | 目标域名 | 本地IP:端口 |
|---|---|---|---|---|---|---|
| 85 | IEXPLORE.EXE | 312 | HTTP/连出 | 121.194.0.206:80 | pfp.sina.com.cn | 0.0.0.0:1089 |
| 86 | IEXPLORE.EXE | 312 | HTTP/连出 | 218.60.1.48:80 | i3.sinaimg.cn | 0.0.0.0:1099 |
| 87 | IEXPLORE.EXE | 312 | HTTP/连出 | 121.194.0.206:80 | pfp.sina.com.cn | 0.0.0.0:1091 |
| 88 | IEXPLORE.EXE | 312 | HTTP/连出 | 121.194.0.217:80 | i0.sinaimg.cn | 0.0.0.0:1092 |
| 89 | services.exe | 220 | UDP/连出 | 192.168.198.2:53 | hq.sinajs.cn | 192.168.198.131:1096 |
| 90 | IEXPLORE.EXE | 312 | HTTP/连出 | 121.194.0.217:80 | www.sinaimg.cn | 0.0.0.0:1094 |
| 91 | IEXPLORE.EXE | 312 | HTTP/连出 | 202.205.3.144:80 | hq.sinajs.cn | 0.0.0.0:1097 |
| 92 | services.exe | 220 | UDP/连出 | 192.168.198.2:53 | js.icast.cn | 192.168.198.131:1101 |
| 93 | IEXPLORE.EXE | 312 | HTTP/连出 | 61.200.81.143:80 | fpdownload2.macromedia.com | 0.0.0.0:1098 |
| 94 | IEXPLORE.EXE | 312 | HTTP/连出 | 121.194.0.221:80 | d1.sina.com.cn | 0.0.0.0:1103 |
| 95 | services.exe | 220 | UDP/连出 | 192.168.198.2:53 | dcads.sina.com.cn | 192.168.198.131:1105 |
| 96 | services.exe | 220 | UDP/连入 | 192.168.198.2:53 | | 0.0.0.0:1105 |
| 97 | services.exe | 220 | UDP/连出 | 192.168.198.2:53 | sina.wrating.com | 192.168.198.131:1107 |
| 98 | services.exe | 220 | UDP/连出 | 192.168.198.2:53 | secure-cn.imrworldwide.com | 192.168.198.131:1108 |
| 99 | IEXPLORE.EXE | 312 | HTTP/连出 | 61.136.62.250:80 | secure-cn.imrworldwide.com | 0.0.0.0:1110 |

**图 6　木马通信检测系统运行主界面**

被检测程序分为：木马程序和普通应用程序，木马通信检测系统测试结果见表 2,3 种检测工具的测试结果比较见表 3。

**表 2　木马通信检测系统测试结果**

| 被检测程序 | 行为数据 | | | | |
|---|---|---|---|---|---|
| | 伪造系统进程 | 模仿系统进程 | 自动发送电子邮件 | ICMP 通信 | 主动对外连接 |
| 灰鸽子木马 | 无 | 无 | 无 | 无 | 有 |
| 流萤木马 | 无 | 无 | 无 | 无 | 有 |
| 波尔木马 | 无 | 无 | 无 | 无 | 有 |
| Iexp1ore.exe | 无 | 有 | 无 | 无 | 有 |
| Foxmail.exe | 无 | 无 | 有 | 无 | 有 |
| Ping.exe | 无 | 无 | 无 | 有 | 有 |

**表 3　3 种检测工具的检测结果比较**

| 被检测程序 | 检测工具 | | | | | |
|---|---|---|---|---|---|---|
| | 木马通信检测系统 | | Sniffer 软件 | | Tcpview 工具 | |
| | 进程信息 | 通信检测 | 进程信息 | 通信检测 | 进程信息 | 通信检测 |
| 灰鸽子木马 | 有 | 能 | 无 | 能 | 有 | 能 |
| 流萤木马 | 有 | 能 | 无 | 能 | 有 | 能 |
| 波尔木马 | 有 | 能 | 无 | 能 | 有 | 能 |
| Iexp1ore.exe | 有 | 能 | 无 | 能 | 有 | 能 |
| Foxmail.exe | 有 | 能 | 无 | 能 | 有 | 能 |
| Ping.exe | 有 | 能 | 无 | 能 | 无 | 否 |

在木马通信检测系统运行过程中，同时开启 NAI 公司的协议分析软件 Sniffer 和 sysinternals 公司的 Tcpview 工具，检测情况如表 3 所示。通过对比发现，木马通信检测系统检测到的信息最多、检测率最高；Sniffer 软件不能检测到进程信息；Tcpview 软件不能检测采用 ICMP 协议的网络数据。测试与检测结果表明，基于 NDIS Hook 的木马通信检测系统能较准确获取被检测程序网络通信信息，能截获采用各种通信协议的网络通信数据包，能通过其网络通信行为数据分析出一些木马程序，对木马通信过程具有较高的检测率，对被检测程序具有一定程度的木马识别率。同时，该系统对普通应用程序的网络通信能够进行监视与检测。

## 5　结束语

本文提出一种基于网络通信行为分析的木马通信检测机制，实现一种基于 NDIS Hook 技术的木马通信检测系统，给出基于判定规则的木马通信识别模型。下一步改进方向是：加上进程行为数据、文件操作行为数据和注册表操作行为数据进一步提高木马识别结果的精度。

## 参考文献


[1] 王万龙, 周育人. 微软新一代操作系统下的个人防火墙研究[J]. 计算机工程与设计, 2007, 28(23): 5613-5615.

[2] 苍志刚, 潘爱民. Windows 平台下的网络监听技术[J]. 计算机工程与设计, 2004, 25(2): 248-251.

[3] 罗改龙, 程胜利. 基于端口复用技术的木马研究[J]. 计算机工程, 2007, 33(15): 165-169.

[4] 杨志程, 舒　辉, 董卫宇. 基于 NDIS 隐蔽通信技术的木马病毒分析[J]. 计算机工程, 2008, 34(10): 147-149.

[5] 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, 21(11): 279-281.


编辑　陆燕菲