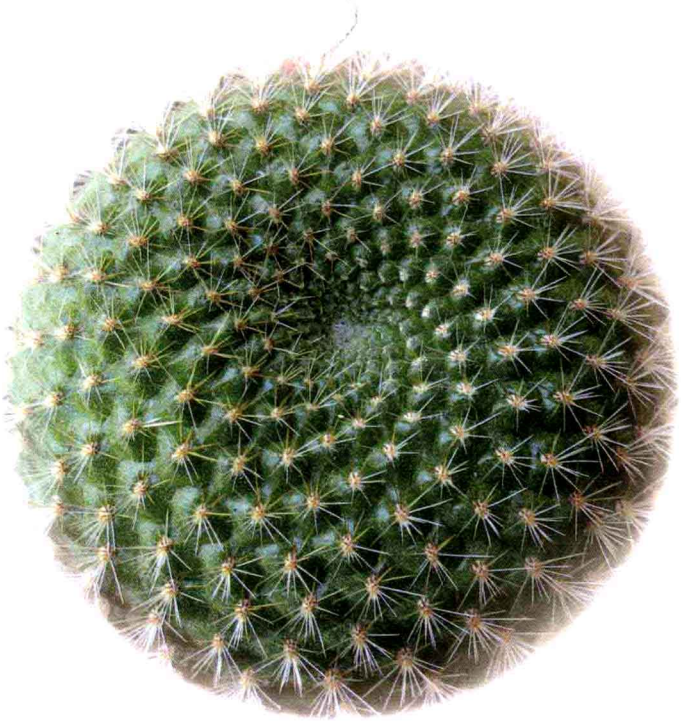
Broadview
www.broadview.com.cn

网络安全入门与提高

# Windows 内核安全编程 从入门到实践

《黑客防线》编辑部 组编　林聚伟 编著



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

网络安全入门与提高

# Windows 内核安全编程

## 从入门到实践



《黑客防线》编辑部 组编　林聚伟 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

## 内 容 简 介

本书详细介绍了 Windows 平台下的内核安全编程知识。首先简单介绍了驱动编程的基本方法；然后详细介绍了 Windows 各个系统组件的工作原理，如文件系统、网络系统自上而下的执行流程。同时还介绍了各个组件涉及的安全问题，如文件隐藏、键盘记录等，并通过工程项目让读者从代码层级了解这些信息安全问题及解决方法；最后介绍了驱动编程本身的安全问题，如安全编码的注意事项和脆弱代码的检测手段。另外本书还介绍了简单的调试和逆向技术，帮助解决开发过程中遇到的技术难题。通过阅读本书，可以帮助读者更深层次地了解内核态下的信息安全知识。

本书适合大专院校计算机系的学生、Windows 程序员、从事信息安全行业的工程师以及所有对 Windows 内核安全编程感兴趣的爱好者使用。



**图书在版编目（CIP）数据**

Windows 内核安全编程从入门到实践 / 《黑客防线》编辑部组编. —北京：电子工业出版社，2012.4
（网络安全入门与提高）
ISBN 978-7-121-16098-1

Ⅰ. ①W… Ⅱ. ①黑… Ⅲ. ①Windows 操作系统－程序设计 Ⅳ. ①TP316.7

中国版本图书馆 CIP 数据核字（2012）第 030617 号

策划编辑：毕 宁 bn@phei.com.cn
责任编辑：徐津平
印 刷：
装 订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×980 1/16 印张：26.5 字数：678 千字
印 次：2012 年 4 月第 1 次印刷
印 数：4000 册 定价：65.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

# 前　言

记得第一次在内核中简单调用 DbgPrint 函数输出“hello world”时兴奋得手舞足蹈，也记得之后开发 ARP 防火墙时遭遇无数次蓝屏而无限接近崩溃的状态。时至今日，接触 Windows 内核安全编程已有三年时间了，期间断断续续地写了一些小程序，直到偶然一天收到《黑客防线》杂志的邀请而萌生了写一本书记录自己一路走来经历的想法，对于和我一样正走在学习 Windows 内核安全编程之路上的菜鸟们来说，有很大的阅读价值，尤其对于那些正准备上路或者刚刚上路的新晋菜鸟而言，通过阅读本书，可以少走很多无谓的弯路。

**本书的结构**

本书共包括 10 章，其中：

第 1～3 章为基础篇，介绍基于 Windows 编程的基础知识；

第 4～8 章为提升篇，通过具体示例介绍各 Windows 组件相关的编程方法；

第 9～10 章为辅助篇，简单介绍安全编码及逆向与调试程序的方法。

本书的理论部分主要以 WDK 文档作为支撑，全书关于理论介绍的知识绝大部分来源于 WDK，另外一小部分知识直接或间接来源于其他文档（如 Windbg 文档等）、书籍（如 Windows Internal 4 等）、网络及个人经验等。

本书的实践部分来源于 WDK 提供的例子及个人项目，建议读者在虚拟机中运行示例。

**本书的阅读说明**

读者可以根据需要选择阅读感兴趣的章节，也可以从头至尾完整地阅读全书。

除了特别说明适用于 Windows 7 或其他系统的内容除外，本书所有的内容默认适用于 Windows XP 系统。

每章提供的示例建议在虚拟机环境下运行。由于本人能力和时间有限，并没有在真机环境中测试过，不能保证所提供的示例程序稳定运行在各种环境下，若直接运行于真机环境中，可能会对您的计算机财产造成不必要的损害。

**致谢**

感谢赵跃华教授对我的悉心培养；感谢张翼（xyzreg)学长对我多年的指导和照顾；感谢全体 535 实验室的师弟师妹们对本书的文字修正；感谢黑客防线在合作过程中的细致帮

助；最后，感谢我的父母对我无尽的给予。

如果您在阅读过程中发现本书的技术性错误，或者有好的建议，欢迎致信ifsecurity588@gmail.com。

作者

2011/12/28

# 目　录

## 第一部分　基　础　篇

## 第三部分　辅　助　篇

PART 1

# 第一部分　基　础　篇

CHAPTER 1

# 第 1 章　前置要求与环境搭建

## 1.1　驱动编程的语言

所有的驱动程序应当是可移植的，即能够支持所有基于 Windows 系统的硬件平台。为了实现跨平台特性，驱动程序的编写者应当做到：

**1．使用 C 语言编码（不用反汇编语言）**

所有内核模式的驱动程序应当使用 C 语言进行编码，这样就能够使用系统兼容的 C 编译器重新编译、链接，从而不用重写或替换任何代码就能够在不同的 Windows 平台上运行。大多数操作系统组件是用 C 语言进行编码的，只有少量的内核组件及 HAL 驱动的小部分使用了汇编语言进行编码，这意味着操作系统很容易实现跨平台。在内核模式驱动程序中应当少用 C++ 语言。

**2．只使用 WDK 提供的编程接口和 Windows 交互**

Windows NT 的每一个系统执行体组件都会导出一系列内核模式函数，这些内核模式函数可以被驱动程序及其他的内核模式组件调用。尽管这些内核模式函数的实现代码可能不时地发生变化，但是其使用者仍然能够保持可移植性，因为接口是不变的。只使用 WDK 提供的编程接口和 Windows 交互，可以确保驱动程序的跨平台特性。

## 1.2　开发环境搭建

### 1.2.1　Visual Studio 2005/2008 的安装与配置

Visual Studio 2005/2008 是常用的编程工具，可以通过购买获得。另外微软制定了 DreamSpark 计划，该计划为在校学生提供免费正版软件，包括 Visual Studio 2005/2008/2010 等，可以访问 www.dreamspark.com 进一步了解该计划，页面如图 1.1 所示。

注册成功后就可以下载 DreamSpark 提供的免费正版软件，不过注册需要一个 edu 邮箱以证明注册者的学生身份。

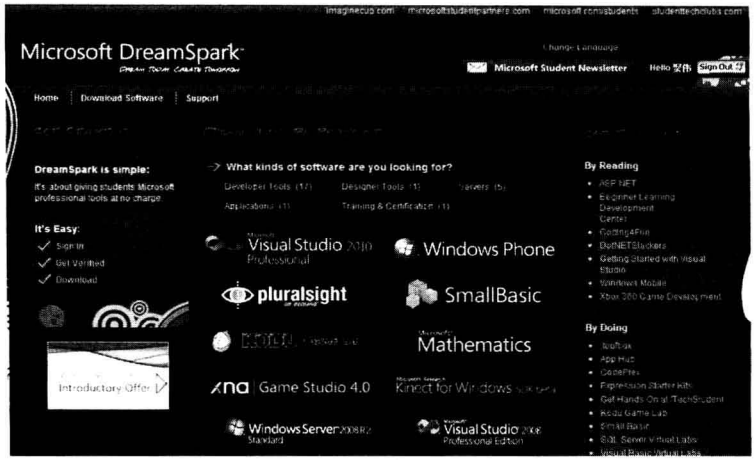



图 1.1　DreamSpark 计划

另外，还有部分学校参与了 DreamSpark 计划，这些学校的学生可以直接下载而不需要先使用 edu 邮箱注册。可以访问 http://dreamspark.eol.cn/查看详细信息，如图 1.2 所示。


图 1.2　查看详细信息

如图 1.2 所示，只要计算机的 IP 地址在参与该计划的学校的范围内，就可以直接下载 DreamSpark 提供的任何免费正版软件。

下载完成后，双击执行安装，选择合适的安装位置及需要安装的组件，单击“下一步”按钮即可。

安装完成后，双击快捷方式即可开始使用，Visual Studio 的起始界面如图 1.3 所示。

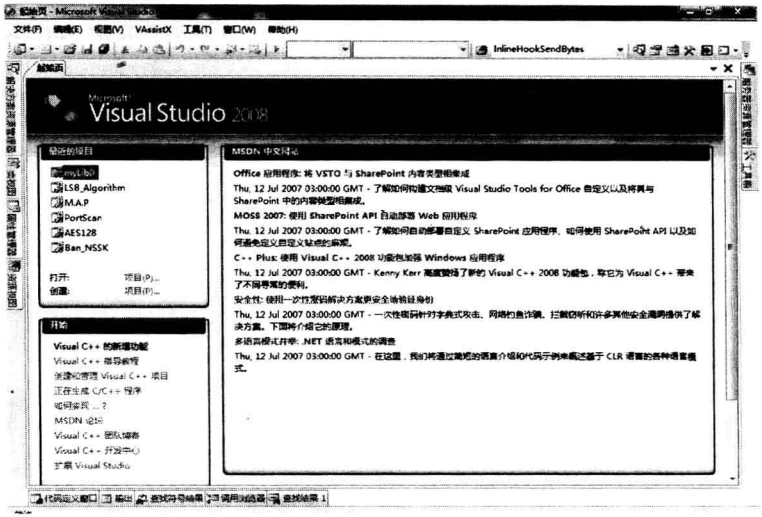



图 1.3　Visual Studio 的起始界面

启动后，可以创建或打开项目，随后可以开始编码、编译、链接和调试等。

## 1.2.2　WDK 的安装与配置

Windows 驱动开发包（WDK）包含工具、代码示例、文档、编译器、头文件及库文件，开发者通过使用 WDK 可以完成驱动程序的开发。

最新的 WDK 可以在微软的下载中心获得，如图 1.4 所示。

下载完成后，得到的是一个 ISO 镜像文件，推荐使用虚拟光驱加载镜像文件，如 Virtual CloneDriver。加载完成后，即可在虚拟光驱中执行安装。

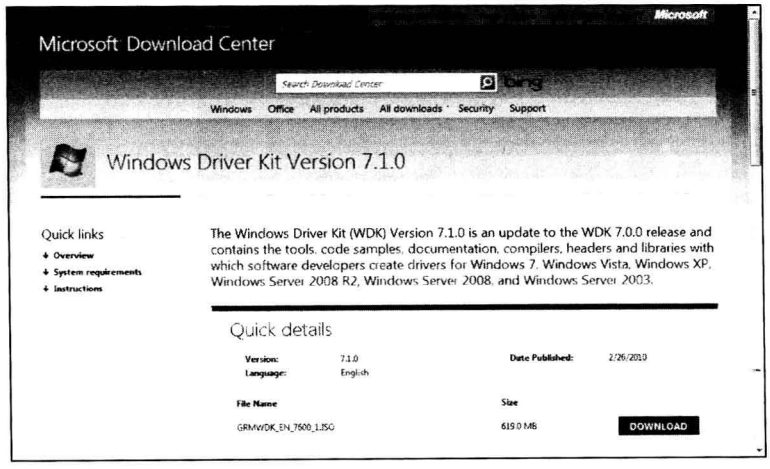



图 1.4　WDK 下载页面

### 1.2.3　VisualDDK 的安装与配置

VisualDDK（可以在 http://visualddk.sysprogs.org/下载）是一个 Visual Studio 的插件，可以帮助开发者直接在 Visual Studio 中编译甚至调试驱动程序（前提是已经安装了 WDK）。

下载完成后直接双击执行安装，安装成功后即可在 Visual Studio 环境中编写、编译甚至调试驱动程序。

打开 Visual Studio，创建一个新的工程，将会发现一个新的选项“VisualDDK”，如图 1.5 所示。

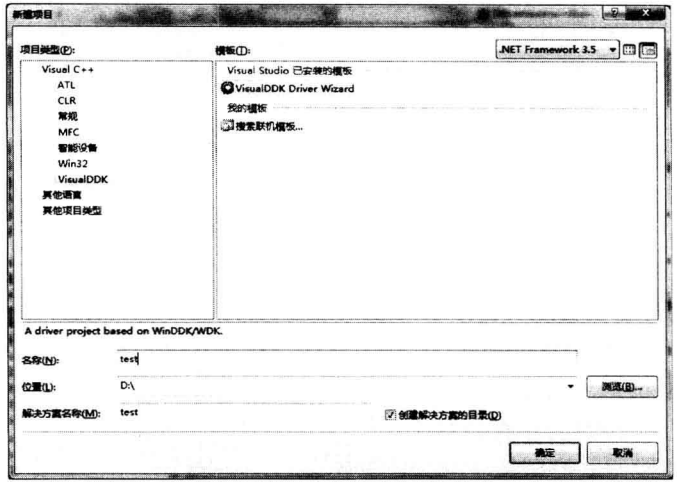



图 1.5　VisualDDK 选项

选择 VisualDDK 选项，填写项目名称，确定后如图 1.6 所示。

如图 1.6 所示，按需选择后即可创建一个驱动程序项目。

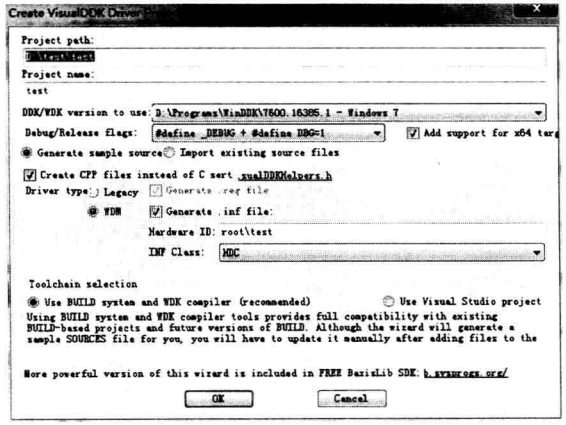



图 1.6　建一个 VisualDDK 项目

## 1.3 常用工具介绍

除了 1.2 节中介绍的 Visual Studio、WDK 及 VisualDDK 之外，还有很多其他常用的工具。

- VMware：常用的虚拟机软件，可以在 VMware 中安装多个操作系统。测试软件尤其是驱动程序时，VMware 是必不可少的工具。
- MSDN：MSDN 是一部编程“辞海”，所有文档化的接口资料都可以查询到，另外，MSDN 还包含大量的系统知识，也是一本非常好的“书”。简单地说，通过索引、搜索功能可以将 MSDN 当做“辞海”用，而通过目录功能可以将 MSDN 当做一本“书”来读。MSDN 是程序员（Windows）必不可少的开发辅助工具。
- IDA：常用的静态反汇编工具。通过 IDA 可以逆向感兴趣的程序，挖掘程序机理。
- Windbg：强大的调试工具。通常使用 Windbg 调试驱动程序，当然也可以调试应用程序。
- Source InSight：一款面向工程项目的代码浏览器，可以有效地组织源码并提供快速访问能力。
- VirtualKD：可以加快主机与虚拟机之间的双机调试速度。
- WinHex：拥有多项功能，包括十六进制数文件查看、编辑功能。
- Wireshark：常用的抓包工具。
- SysinternalsSuite：一组工具，包括 procexp.exe、Procmon.exe 等，很有用。
- LoadPE：常用的 PE 文件查看器。
- Dbgview：捕捉调试信息并显示，非常有用。
- 除了上述工具外，还有很多其他的工具同样可以帮助开发者更加高效地开发程序，程序开发者可以按需、按个人习惯选择使用。

CHAPTER 2

# 第 2 章　内核编程基础知识

## 2.1　Windows 主要系统组件

### 2.1.1　对象管理器

Windows 对象管理器用来管理对象。文件、设备、同步机制、注册表键值等在内核中都使用对象来表示。每个对象都有一个对象头（包含对象的一些信息，如名词、类型和位置等）和一个对象体（包含与各种对象类型相关的数据）。

Windows 包含超过 25 种类型的对象，如文件、设备、线程、进程、事件、互斥体、信号量、注册表键、作业、内存区、符号链接等。

对象管理器执行如下的操作来管理对象：

- 负责对象的创建和销毁；
- 维护一个对象名字空间以跟踪对象信息；
- 跟踪分配给每一个进程的资源；
- 跟踪具体对象的访问权限；
- 在对象的生命期内管理对象，判断对象何时可以被销毁。

对象管理器导出的接口函数通常包括一个“Ob”前缀，如 ObGetObjectSecurity 函数。

### 2.1.2　内存管理器

Windows 内存管理器用来管理物理内存，主要执行如下的操作：

- 管理内存的分配和销毁；
- 支持内存映射文件、共享内存及写时复制的实现。

内存管理器导出的接口函数通常包括一个“Mm”前缀，如 MmGetPhysicalAddress 函数。

### 2.1.3　进程和线程管理器

进程即当前系统中正在运行的软件程序，每个进程都有一个 ID；线程则代表了程序的哪部分正在运行，每个线程也有一个 ID。

一个进程可能包括多个线程，每个线程通过占用 CPU 时间获得执行。在单核处理器的机器

上，虽然一个进程可能拥有多个线程，但在某一时刻只有一个线程正在运行，且每个线程只运行很短的时间便切换到另一个线程继续执行，从而让用户产生错觉，似乎同一时间有多个程序在执行；在多核处理器机器上，在某一时刻可以有多个线程同时运行。如果一个程序包含有多个线程，那么多个线程可以在不同的处理器上同时运行。

Windows 的进程和线程管理器负责处理进程中所有线程的执行问题。无论机器是单核或是多核，在驱动编程中都必须仔细地处理所有的线程，无论这些线程按照什么顺序获得执行，驱动程序都会稳定地运行。

进程和线程管理器导出的接口函数通常包括一个“Ps”前缀，如 PsCreateSystemThread 函数。

### 2.1.4 I/O 管理器

计算机包含多个输入/输出设备，这些输入/输出设备负责计算机和外界的信息交互。常见的输入/输出设备包括键盘、鼠标、磁盘、网络端口等，这些输入/输出设备的驱动程序负责硬件设备与操作系统之间的连接。

Windows 的 I/O 管理器负责管理应用程序和 I/O 设备驱动程序接口之间的交互，而应用程序和 I/O 设备之间的交互主要通过 IRP（I/O 请求包）完成，这些 I/O 请求包从操作系统传递到指定的驱动程序，又从一个驱动程序传递到另一个驱动程序。I/O 管理器负责维护这些 I/O 请求包。

Windows 的 I/O 系统提供了一种称为驱动设备栈的分层驱动模型，IRP 一般在一个驱动设备栈中从一个驱动程序被传递到另一个驱动程序。

I/O 管理器导出的接口函数通常包括一个“Io”前缀，如 IoCreateDevice 函数。

### 2.1.5 PnP 管理器

PnP（即插即用）结合了硬件技术和软件技术，PnP 技术使得系统能够检测到插入机器中的设备。PnP 管理器实际上是 I/O 管理器的一部分。

PnP 管理器也有相应的一系列函数，但需要注意的是，这些函数使用的并不是“Pp”前缀，而是“Io”前缀，如 IoRegisterDeviceInterface 函数。

### 2.1.6 电源管理器

Windows 使用电源管理技术来减少计算机的电源消耗，例如，Windows 系统可以进入睡眠或者休眠状态以节省电源。电源管理系统可以使得当计算机关机或者进入低电源消耗状态时，计算机上的设备也可以恰当地关闭电源而不会丢失数据。

Windows 的电源管理器能够为所有支持电源状态变化的设备管理其电源状态的有序变化。如果需要编写一个支持电源状态变化的驱动程序，应当考虑如下一些因素：

- 系统的活动级别；
- 系统的电池级别；
- 关机、睡眠及休眠的请求；
- 用户的动作，如按下了电源按钮；
- 控制面板设置，如电源指示 10%的时候自动关闭系统。

电源管理器使用了电源管理策略来处理电源事件并产生电源管理请求。电源管理器通过获取要求改变电源状态的请求，决定哪一种电源状态是恰当的，然后给恰当的驱动程序发送 IRP 指示其执行电源状态的变化。

电源管理器也是 I/O 管理器的一部分，不过它导出了自己的接口函数，前缀为“Po”，如 PoSetPowerState 函数。

### 2.1.7　配置管理器

早期 Windows 将配置信息存储在“INI”文件中，但是，随着后来 Windows 环境变得复杂起来，需要一种新的存储软件和操作系统信息的方法，Windows 注册表应运而生。

Windows 配置管理器用来管理注册表。如果一个驱动程序需要知道注册表发生的变化，可以使用配置管理器提供的函数，在指定的注册表位置注册一个回调函数，随后，当该指定位置处的注册表值发生变化时，注册的回调函数将被触发，从而驱动程序获得注册表变化的信息。

配置管理器导出的接口函数通常包括一个“Cm”前缀，如 CmRegisterCallback 函数。

### 2.1.8　安全引用监视器

安全是操作系统非常重要的一部分，在实施一个操作前，操作系统必须首先确保该操作没有违反系统策略。Windows 使用了访问控制列表（ACL）来判断哪些对象拥有何种安全性，而安全引用监视器提供了一些函数用于访问控制。

安全引用监视器导出的接口函数通常包括一个“Se”前缀，如 SeAccessCheck 函数。

## 2.2　常见名词解释

本节简单介绍一下内核编程中常见的名词，包括内核名词、文件名词及网络名词。

### 2.2.1　内核名词

#### 1. NT 式和 WDM 式驱动程序模型

NT 式驱动程序模型：NT 式驱动程序模型是一种比较老式的驱动程序模型，但适用于现有的 Windows 系统。NT 式驱动模型没有固定的形式，最简单的 NT 式驱动程序可以仅仅包括一个 DriverEntry 入口函数。正是由于 NT 式驱动程序模型的这一特点，程序开发者可以编写一个完全不支持硬件工作的驱动程序，却可以将代码运行在内核模式中。

WDM 式驱动程序模型：WDM 式驱动程序在 NT 式驱动程序的基础上，还必须：

- 包括 wdm.h 头文件，而不是 ntddk.h（wdm.h 是 ntddk.h 的一部分）；
- 被设计为一种 WDM 驱动程序类型，如总线驱动、功能驱动、过滤驱动等；
- 创建的设备对象属于 WDM 设备对象类型（物理设备对象、功能设备对象、过滤设备对象）；
- 支持即插即用（PnP）；