Broadview®
www.broadview.com.cn

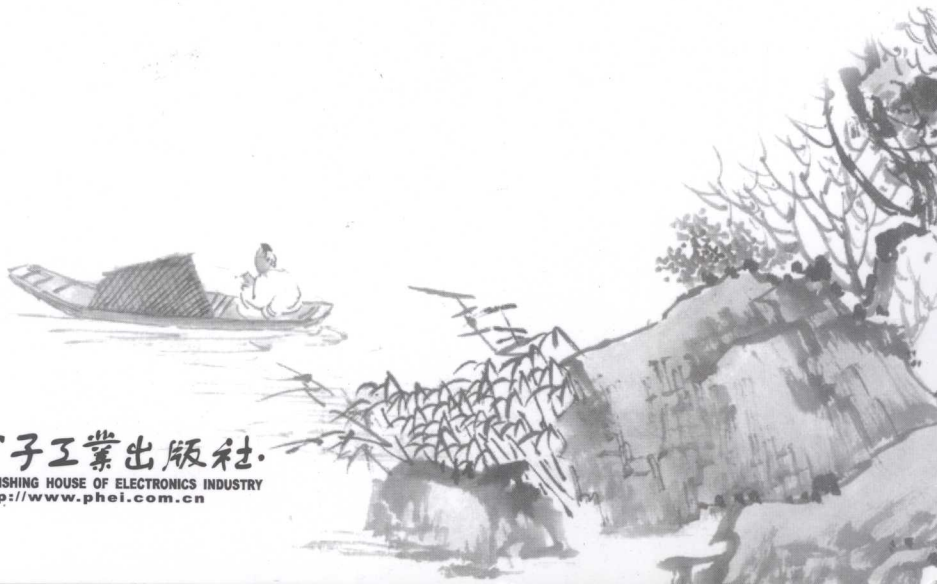
寒江独钓

驱网核心技术丛书

寒江独钓——Windows内核安全编程

Windows 内核安全编程

谭文 杨潇 邵坚磊 等著



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

驱网核心技术丛书

寒江独钓

——Windows内核安全编程

Windows 内核安全编程

谭文 杨潇 邵坚磊 等著

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

内 容 简 介

本书从Windows内核编程出发，全面系统地介绍了串口、键盘、磁盘、文件系统、网络等相关的Windows内核模块的编程技术，以及基于这些技术实现的输入密码保护、防毒引擎、文件加密、网络嗅探、网络防火墙等信息安全软件的核心组件的具体编程。主要知识重点包括：Windows串口与键盘过滤驱动、Windows虚拟存储设备与存储设备过滤驱动、Windows文件系统过滤驱动、文件系统透明加密/解密驱动、Windows各类网络驱动（包括TDI过滤驱动及三类NDIS驱动），以及最新的WDF驱动开发模型。有助于读者熟悉Windows内核驱动的体系结构，并精通信息安全类的内核编程技术。本书的大部分代码具有广泛的兼容性，适合从Windows 2000一直到目前最新的Windows 7 Beta版。

本书适合大专院校计算机系的学生、普通Windows程序员、Windows内核程序员、信息安全行业的程序员，以及希望了解Windows系统底层知识的计算机编程爱好者使用。阅读本书，需要读者有C语言、数据结构、操作系统和计算机网络的基础知识。

图书在版编目（CIP）数据

寒江独钓：Windows 内核安全编程 / 谭文，杨潇，邵坚磊著. —北京：电子工业出版社，2009.6
（驱网核心技术丛书）
ISBN 978-7-121-08796-7

Ⅰ. 寒… Ⅱ.①谭… ②杨… ③邵… Ⅲ. 窗口软件，Windows－安全技术 Ⅳ. TP316.7

中国版本图书馆 CIP 数据核字（2009）第 071333 号

策划编辑：李 冰
责任编辑：葛 娜
印 刷：北京智力达印刷有限公司
装 订：北京中新伟业印刷有限公司
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：787×1092 1/16 印张：32.25 字数：684 千字
印 次：2009 年 6 月第 1 次印刷
印 数：4000 册 定价：75.00 元（含光盘 1 张）

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。

序

大约在半年年前，谭文就和我谈过想写一本既能深刻介绍 Windows 内核架构，又能结合具体 Windows 驱动程序开发实例的书。那时候他的《天书夜读——从汇编语言到 Windows 内核编程》已经出版，《天书夜读》所涉及的内容很广，但就如同书名一样，它的内容不太适合刚刚涉猎 Windows 内核编程的程序员，反而更像一本供黑客学习的读物。书中翔实地介绍了很多反汇编的实战技巧，非常的精辟，但对于新手来说，容易对 Windows 内核编程产生畏惧感。

当我第一次读完《寒江独钓——Windows 内核安全编程》的初稿时，我觉得本书非常适合 Windows 内核程序的入门。Windows 内核程序一直被认为是只有高手才能涉及的领域，很多程序员对这种开发都觉得非常神秘。我觉得这是一种错觉，其中有一个很重要的原因就是国内很少出版这方面的书籍。这本书很好地弥补了这方面的空白，我相信大部分读者读完本书后，都会觉得 Windows 内核开发程序不再那么神秘。的确，微软自从 Windows 2000 版本以后，内核的架构变化不是很大。当然，这并不意味着你读完本书后，你就可以对内核开发游刃有余了，这需要你对每一个细节反复研究，并且多做试验。

编写 Windows 内核程序，就意味着这个程序可以执行任意指令，可以访问计算机所有的软件、硬件资源。因此，稍有不慎就有可能将系统变得不稳定。Windows 的设计者设计了各种驱动模型或者框架，如 NT 式内核驱动模型、WDM 框架和新推出的 WDF 框架。在这些模型框架下编程，就使内核编程变得简单，同样也降低了内核程序崩溃的机会。其实，Windows 驱动程序员和黑客都在写内核程序，唯一不同的是驱动程序员按照微软设计的模型写程序，而黑客可以不按照这些框架写。Windows 设计的这些框架，可以将操作系统的原理隐藏起来，只暴露一些接口，驱动程序员只要把这些接口写好就可以了。从这个角度看，驱动开发并不难，尤其是读完本书后，更会觉得不难了。但是想完成一些特殊的功能，如内核级隐藏进程等，Windows 的这些框架就没什么用处了，程序员就需要对 Windows 内核有全面的了解，通过直接修改 Windows 内核来实现这些目的。往往黑客对这种技术乐此不疲，通过修改 Windows 内核，你会发现你的程序几乎无所不能。

编写内核程序是一件很痛苦的事情，回想起这些年学习内核程序开发的经历，真是感慨万千。就如同谭文所说：编写内核程序的人从某种程度讲是孤独的。当一个经验并不丰富的小程序员面对庞大复杂的并且不开源的 Windows 框架时，那是一种怎样的无助感啊！谭文是我比较钦佩的程序员之一，他对技术非常执着，并且精力充沛。内核程序的知识涉及面非常广，不同类别的内核程序差别也特别大，他几乎都有所涉猎。相信读者在读完这本书后，能对 Windows 内核开发有比较详细的了解，同时也能结合书中的实例写出很优秀的内核程序了。

张　帆

2009 年 5 月 1 日于北京

关于本书作者和贡献者

- 主要作者：

谭文，C 程序员。1980 年生于湖南。2002 年毕业于西安交通大学自动控制系。毕业后一直从事各类系统底层软件的开发设计工作；目前在英特尔在上海紫竹科技园的研发中心参与不同平台二进制指令动态翻译系统的项目开发。爱好聊天、发帖。曾著有《天书夜读——从汇编语言到 Windows 内核编程》一书。编写本书的第 1～3,7,8,10~12 章，并统稿全书。

杨潇，C 程序员。1981 年生于陕西。2006 年毕业于西安交通大学自动控制研究所。一直从事各类系统软硬件接口部分的开发设计工作；目前在Comodo北京研发中心任职，负责各类 Windows/Linux 相关驱动的开发。爱好音乐、旅游和摄影。负责编写本书的第 5,6 章。

邵坚磊，前执业医生，现 C、汇编程序员。1976 年生于上海，毕业于上海交通大学计算机系，具有临床医学和计算机专业的双学位。长期致力于 x86 体系架构与 Windows 系统底层技术的研究与相关开发工作；目前在 NEC-AS 上海研发中心主持信息防泄密软件的 Windows 内核驱动的开发工作。是著名的反 rootkit 工具 DarkSpy 的作者之一。曾与谭文合著《天书夜读——从汇编语言到 Windows 内核编程》。爱好网游。编写本书的第 4 章。

- 其他有贡献的作者：

卢冠豪，C 程序员。1981 年生于中国台湾。毕业于辅仁大学资讯工程学系。长期从事 C、C++、网络与通信程序设计的工作；参与过“端点安全”、“资产管理”、“网络流量分析”等项目的开发与维护；擅长 Windows 项目开发。平时爱好吸收与科技电脑有关的新知识。受邀编写本书的第 9 章。

张佩，C 程序员，1982 年生于江苏扬中，毕业于苏州大学。近三四年来，一直从事底层软件开发，乐此不疲。因偶然的机会，在 NEC-AS 公司加入了 Infocage 安全软件开发项目组，从此与底层驱动开发结缘。现工作于上海一家公司，服务于美国 Avid 公司

音频驱动项目。此人好读书，好写文章，好交朋友。为人善，与人交善，诚善人也。受邀编写本书的第 13 章。

- 其他有贡献者：

杨瑾，西安交通大学在读博士生。在参与导师的项目过程中，因技术的需要阅读了本书的原稿，修改了书中的许多细节错误与纰漏，并按照本书介绍的步骤，编译、安装和测试了本书源码光盘上的部分代码。是本书最主要的审校者。

张帆，1980 年生于北京，毕业于北京理工大学电子工程系。目前在赛门铁克北京的研发中心从事信息安全方面的工作；著名畅销书《Windows 驱动开发技术详解》的作者。阅读了全书的原稿，并给出了许多指导性的意见，并为本书作序。

张银奎，国内著名的调试专家。目前和本书作者谭文在同一研发中心工作，著有《软件调试》一书。阅读了本书的全部原稿，修改了书中多处细节，给本书作者许多指导性的意见，并参与了本书的命名。

李庆颖，华南理工大学计算机系统研究所硕士研究生。阅读了本书的原稿，修改了书中的一些细节错误与纰漏，并按照本书介绍的步骤，编译、安装和测试了本书源码光盘上的部分代码。是本书重要的审校者。

马勇，驱动开发网的站长，国内知名的文件系统驱动专家。“驱网核心技术丛书”的组织和策划者之一，阅读了全书的原稿，并给出了许多指导性的意见。

当您碰到本书内容相关的问题，如环境配置不能成功、代码编译失败、运行出现错误等情况时，您可以通过 E-mail 直接与本书作者联系。“驱网核心技术丛书”所有作者统一使用公共邮箱如下：

ckrsoft@gmail.com

前　言

本书是一本专门介绍实时扫描的防毒软件、虚拟磁盘、硬盘还原、硬盘加密、文件系统保护、文件透明加密、防火墙、密码输入保护等软件的 Windows 内核模块的具体实现方法的编程技术书。本书的目的是使读者能够用 C 语言编写这些核心模块。

大学的时候，在 Windows 平台上我最初学的是 VB，然后是 Delphi。我的感觉是，无论想实现任何功能，都早已有工具的开发者给我们准备了良好的接口和文档，让我们学习和使用都非常的方便。因此觉得自己已经学到了终点。如果仅从“能实现功能”的角度讲，我没有必要再学习了，剩下的事情，只是去很舒适地使用那些接口就可以了。那又何必再学习 Windows 编程呢？

工作之后遇到了障碍。我的第一个任务是实现一个网络的虚拟磁盘。我虽然自以为无所不能，但是也找不到在 Windows 系统里增加一个虚拟磁盘的 API 在哪里。我每天都在使用虚拟光驱、杀毒软件、防火墙，但是我从未想过它们如何实现。不是因为我懂，而是因为我自以为任何功能的实现一定是简单而舒适的，等需要的时候再去研究，绝不会有什么困难。

但是实际编码的时候才明白：良好的接口、舒适的编码过程，绝对不是天生之道。天地万物自混沌而起，那些美好的表面，不过是在残酷的现实上重重包裹的包装纸罢了。

一辆新车的表面自然光彩照人，操作接口也人性而美好。但是一旦需要打开车身去修理内部某根漏油的管子，就没有那么容易和舒适了。造成这种情况，绝不是 Windows 的底层开发者们天生没有美学观念。那些多年积累和维护着并不断改进的无数行代码，已经是人类工程史上的奇迹了。如今要打开它的外壳去肆意修理，当然不是一件轻松的事情。

但这正是 Windows 内核编程的魅力所在。

只有极少的程序员会需要参与微软的 Windows 内核开发，也只有极少的读者会自己试图从头开发一个类 Windows 的操作系统内核（有这方面兴趣的读者，建议参考开源项目 ReactOS）。单纯地讲解 Windows 内核编程对大多数读者都没有意义。但是，信息安全类的软件是内核编程的极好的应用实例。病毒实时监控、防火墙、入侵检测、数

据保护还原、数据即时备份、数据加密、数据防止泄密、反外挂等，都不同程度地涉及到内核编程；或者，内核编程可以让它们工作得更好。这些就是本书的内容，因此本书的副标题为“Windows 内核安全编程”。“寒江独钓”则表明了这个领域的寒冷与寂寥。

本书和《天书夜读——从汇编语言到 Windows 内核编程》的不同之处在于：《天书夜读》一书介绍的是自己调试 Windows 内核、获取知识、解决问题的技巧。因此《天书夜读》一书介绍的内容大部分是没有文档可循的，容易走火入魔。

本书则基本上介绍的是正统的内核编程技术，是微软在内核编程中给信息安全软件开发者提供的相关接口的大集合，是名门正派的技术，不沾邪气。一个好的内核程序员，“正邪兼修”是有必要的。

本书既适合于有志于成为软件程序员的学生使用，也适合于希望加强自己的技术实力的 Windows 程序员阅读，同时更适合于从事信息安全行业的 Windows 软件的开发者作为手头参考。

本书对改善病毒横行的网络现状也有一定益处。虽然无助于劝说那些孜孜不倦的病毒开发者们弃恶从善，但是至少有助于他们提高技术素养，学会更认真地编写程序，以免总是写出导致程序崩溃和系统蓝屏的代码，影响无辜者的正常工作。

本书假定读者了解 C 语言，能理解 C 语言的基本语法，并且学习过操作系统、计算机网络和数据结构的基础知识。一般来说，如果读者听说过“进程”、“文件系统”、“中断”、“TCP 协议”、“以太网包”、“链表”、“哈希表”、“加密算法”这些名词，则足够阅读此书了。

有些读者可能会关心作为一个程序员的就业前景。这也是我非常关心的一个问题。我曾经在杭州的核新软件公司为证券营业部开发防火墙和虚拟磁盘，一共 3 年的时间；后来在日电卓越软件（北京）的上海分公司开发部信息安全开发课工作了 3 年。我认识的业界朋友们，大多在赛门铁克、趋势、瑞星、EMC、华赛这样的公司就职。现在是我工作的第 7 个年头了，我在 Intel 在上海的紫竹中心参与动态二进制翻译项目。最有价值的是，我参与的每一个项目都让我学习到更多的知识，面对许多前所未有的考验，每一步都让人充满了精神上的成就感。

本书的读者未来很可能会从事底层编码的工作，而不是一个上层的设计和管理人员。从事底层编码的程序员，常常被同事称为“牛人”。这个牛人不是“牛皮哄哄的人”的意思，而是“像牛一样辛苦工作的人”的意思。想从事这个行业的读者，我抄我的前同事钱铮最喜欢的一首古诗《代牛言》献给您：

渴饮颍水流，饿喘吴门月。黄金如可种，我力终不竭。

谭　文

2009 年 1 月 1 日

阅读注意

如何阅读本书

请注意，本书基本上不可以跳读。虽然书中的内容是按照应用的领域进行划分，但是并没有采用一种分类介绍所有基础知识，然后分各个领域介绍的模式。而是从简单的应用到复杂的应用依次进行的。

本书首先最简单地介绍了串口的过滤，然后是键盘的过滤（用于密码保护）。先介绍这些内容是因为它们在驱动中最简单。但是并不是意味着只关心网络过滤的读者，可以跳过它们，直接翻到 NDIS 中间层驱动的章节进行阅读。

本书所有的章节内容，采用的都是基于 WDK 的 C 语言编程。在编程方法上是统一而且一脉相承的。对于所有的内核 API 函数的介绍、特殊名词、基础概念的说明，本书都是在具体范例中，第一次出现时做详细的介绍，并举出例子。而以后再出现时，就直接使用而不做介绍了。因此，要对后面介绍的复杂的内核模块的章节能顺利地阅读，必须以前面的简单的章节为基础。

书中的代码

书中及附带光盘中的代码，仅供学习与研究使用。这些代码有一部分是笔者所写，有一部分是修改或者直接引用了业内公开的、可供研究使用的代码。这些代码在笔者的测试环境下都可以正常运行。但是笔者并不保证这些代码都有相应的授权可以应用于商业开发中，也不保证这些代码在所有版本的 Windows、与其他任何软件并存都能可靠运行。商业级的内核代码需要经过详尽的测试，本书中的示例代码并不具备这个过程。

读者在运行这些代码时，应该自己使用合理的安全手段（保存未保存的文件、使用虚拟机，或者在运行之前准备硬盘还原映像），以便万一系统损坏时可以恢复。至于具体的操作方法，请读者参考本书第 1 章“内核上机指导”。笔者对读者因为运行这些代码时发生的意外而导致的损失不负责任。

如果本书的读者将这些代码应用到商业软件开发中，由此所引发的一切后果（如版权上的侵犯和技术上的问题，以及相应造成的损失），本书的作者都不承担任何责任。

如果读者发现测试时某个示例程序无法运行，请做以下的事：

（1）在虚拟机上安装一个干净的 Windows XP，不要安装任何其他软件。

（2）确保虚拟机模拟的是单核的 CPU。

如果程序依然崩溃，请保留 dump 文件，并用 E-mail 联系本书的作者。

阅读前的准备

读者必须先准备编译和调试环境。内核编程的环境不像应用编程只需要安装一个软件那么简单。具体的方法在本书第 1 章有详细的介绍。

随书光盘上并没有提供运行、调试本书示例代码的完整工具集。这是因为有一部分软件的授权要求为“不得拷贝分发”，所以需要读者自己在网上下载。但是幸运的是，本书涉及的所有必要工具软件都是免费的，第 1 章中提供了相应的说明和下载的网址，下载的网址可能有时效性。同时由于工具软件版本的升级，这些说明可能和实际的情况有所不同，读者应该在网上搜索最新的信息。

关于本书读者所需要的基础知识，笔者认为应该精通 C 语言（汇编与 C++是不必要的，本书只使用 C 语言编程），至少熟悉 Windows 下 C 语言的一种应用编程方式（比如使用过 VC 或者 C++Builder）。如果从事过驱动开发，那是非常好的基础。但是本书尽量面向没有从事过驱动开发的读者。

操作系统、计算机网络、数据结构与算法的基础知识对于理解本书的内容很有帮助，完全没有学习过这方面知识的读者会难以理解书中的一些细节。

技术细节的说明

本书是由实际从业的程序员所写，总体而言，是实践而不是理论之书。所以和一些常见的编程技术类书籍不同，本书对一些具体的技术细节的描述可能不详细，比如对内核函数的参数使用的说明。一般来说，Windows 的应用层 API 函数和内核 API 函数的参数都非常复杂，详细地说明它们需要较大的篇幅。但是更重要的是，在绝大多数情况下，使用的参数组合仅仅是常见的几种。本书以笔者实际的编程经验，详细介绍在开发中实际使用的参数配置；而对于笔者从未使用或者极少使用的情况，则往往以“笔者也从未使用过”进行说明。因为笔者认为，知道哪些细节有用，远比了解有哪些细节更加重要。所以读者应该理解这一点：本书提到的情况，往往是必须掌握的；而本书未提及的情况，则往往是很不常用的。如果需要用到的细节本书没有提供，大部分都可以在帮助文档中查到。

与应用编程不同，内核编程（尤其是信息安全软件涉及的内核编程）总有部分技术点没有文档可循，有时虽然有文档，也语焉不详。有些问题，笔者也并没有搞清其原因，

但是在实际开发中却是必须解决的，因此很有可能是通过经验解决。为此，书中虽然说“请务必这样做”，或者说“笔者是这样做的”，但是并不能详尽地说出原因。如果有读者对某个问题有更深入的了解，敬请与笔者联络，以便在重印或再版时加入更明确的说明。

词汇的翻译

在这个行业中有太多词汇来源于英文文档，因此如何翻译为中文是一个难题。一般地说，应当沿用业界最常用的翻译方式。但是由于笔者所读过的翻译书籍毕竟有限，不大可能使各个词汇都符合业界最标准的翻译方式。为了稳妥起见，那些重要的、可能引起疑惑的词汇在本书正文第一次出现时，都使用“中文（英文）”的方式。

有少数词汇是笔者在阅读前人的书籍时，曾经深感疑惑的。为了浅显起见，有意地使用了本人自认为更浅显的翻译方式。最典型的就是“System Routine”，所有在 WDK 中提供的开发者可以调用的函数（类似于 SDK 中提供的 API 函数）都称为“System Routine”。笔者见过以前的书中翻译为“系统例程”。既然它就是一个函数，笔者倒是觉得翻译为“内核 API”或者“内核函数”，对于习惯应用程序编程的读者会更好理解。同理，“Dispatch Routine”也被笔者翻译成了“分发函数”。读者如果认为有不妥或者错误的地方，敬请批评指正，以便在重印或再版时修改。

驱动开发模型的选择

WDF 是 Windows 驱动编程模型的发展趋势，但是传统的 NT 式驱动、WDM 模型依然是理解驱动开发的基础。目前，完全抛弃传统是不可能的。一方面，传统的模型编写的驱动程序依然有效，而且有大量现成的例子可以参考；另一方面，WDF 虽然已经出来很久了，但是有许多的基础例子找不到 WDF 的范本。很多无处不用的驱动程序是人类经过多久都不愿意去重新开发的，虽然微软把大部分旧的驱动程序的例子改为了 WDF 模型，并在 WDK 中和旧例子一起提供给开发者参考，全面取代 WDM 的时代依然没有到来。这个过程还有赖于 WDF 自身的进一步完善。

如何区分驱动模型？下面是本书第 2 章中的一段：

“Windows 的模型概念，本来是就驱动程序的行为而言的。比如 WDM 驱动，必须要满足提供 *n* 种被要求的特性（如电源管理、即插即用）才被称为 WDM 驱动。但是如果不提供这些功能，那么统一称为 NT 式驱动。

本书采用简单的区分方法。将一切在 Windows 2000 到 Windows Vista 下能正常运作且未调用 WDF 相关的内核 API 函数的驱动都称为传统型驱动（包括 NT 式和 WDM）。如果调用了 WDF 相关的内核 API 则称为 WDF 驱动。

从目前笔者的理解来看，WDF 的编程方式是对已有的在 WDM 中广泛使用的内核

API 函数的一次封装，尤其是对以 Io 开头的系列函数，包括驱动对象（Driver Object）、设备对象（Device Object）、和请求（IRP）相关的 API 进行了封装，而且旧的 API 完全可以调用。已经有太多的内核程序依赖于旧的接口，微软在很长一段时间内都不可能要求只能使用 WDF 编程。

因此本书采用二者并重的方式。大部分例子用传统型驱动方式编码，另一些例子则使用 WDF 的方式编码（虚拟磁盘与虚拟网卡）。读者会发现这二者是一脉相承的，只要熟练掌握传统型的编程方法，了解 WDF 会是非常轻松和愉快的过程。

本书的习题

有些读者不喜欢用习题来练习的方式，他们认为“这只是一些知识性的东西，记下来是浪费大脑的空间”或者“这些东西在需要的时候再去查阅资料就可以了”等。确实，有许多细节是不需要去记忆的。对一个程序员来说，最重要的就是在需求产生、得到问题时，明白应该从哪里入手去解决这个问题。这就要有一些在大脑中已经建立的概念。如果是处在完全蒙昧无知的状态下，那么就算把整个图书馆都摆在面前，也完全没有意义。

本书每章后都附有少量的习题，这些习题的目的在于让读者自我检查，读完一章后是否已经建立了正确的概念。此外，还有一些常识性的习题（比如指令 int 3 的意义是什么），是会对读者在进行实际工作时有很大帮助的，建议读者不要忽略。

目　录

键盘是很重要的输入设备！这是因为我们用键盘录入信息、用键盘输入密码，甚至用键盘编程，也用键盘著书立说。对于黑客来说，使用庞大的计算机资源去破解那些坚不可摧的加密算法，哪如偷偷地记下用户用键盘输入的密钥更加简单呢？本章专注于键盘的保护。

CPU 是计算机的核心，但是它不保存信息。如果它被窃，我们可以简单地购买一个新的。但是如果装满了机密信息的硬盘被窃了，那可就不是买一个新的就能弥补得了的。本章介绍硬盘内核魔术：虚拟硬盘。虚拟硬盘可以不被盗窃者利用吗？良好的设计可以做到这一点。