Broadview
www.broadview.com.cn

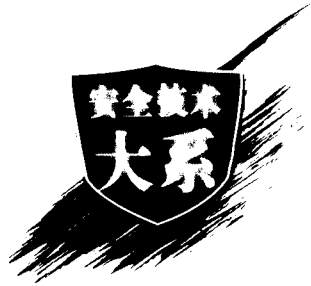
安全技术大系

Hacker Debugging Uncovered

黑客调试技术揭秘

[美] Kris Kaspersky 著
周长发 译

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



Hacker Debugging Uncovered

黑客调试技术揭秘

[美] Kris Kaspersky 著
周长发 译

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

内 容 简 介

本书是帮助应用程序员和系统程序员理解调试过程的指南，揭示了各种调试器的实用使用技巧，说明了如何操作调试器以及如何克服障碍和修复调试器，介绍了黑客利用调试器和反汇编器来寻找程序弱点和实施攻击的方法。通过本书，程序员将学会如何弄清楚计算机系统内部的结构、如何重建没有提供源程序的程序的运行算法、如何修改程序，以及如何调试驱动程序。本书还详细介绍了在 Windows 和 UNIX 操作系统中调试应用程序和驱动程序的方法。对于各种调试技术，书中都给出了带有详尽解释的源代码。如果你是具有 C/C++或者 Pascal/Delphi 语言实际编程经验的程序员，那么本书就是使你的技术升华至一个新的台阶的宝典。

Hacker Debugging Uncovered, ISBN:1931769400

版权贸易合同登记号　图字：01-2004-4689

图书在版编目（CIP）数据

黑客调试技术揭秘 /（美）卡斯帕克尔（Kaspersky, K.）著；周长发译. —北京：电子工业出版社，2006.7
（安全技术大系）
书名原文：Hacker Debugging Uncovered
ISBN 7-121-02802-6

Ⅰ. 黑…　Ⅱ. ①卡… ②周…　Ⅲ. 计算机网络－安全技术　Ⅳ. TP393.08

中国版本图书馆 CIP 数据核字（2006）第 068164 号

责任编辑：顾慧芳
印　　刷：北京智力达印刷有限公司
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
经　　销：各地新华书店
开　　本：787×980　1/16　印张：33.5　字数：548 千字
印　　次：2006 年 7 月第 1 次印刷
定　　价：59.00 元（含光盘 1 张）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。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。

译者序

当电子工业出版社的郭立女士委托我翻译这本《黑客调试技术揭秘》时，我内心是有几分踌躇的。因为我虽然编了多年程序，但是却从来没有做过一天黑客。我怀疑我是否有兴趣将这本600多页的“黑客书”翻译出来，也怀疑这本书对我这样的程序员是否有帮助。但是在走马观花地浏览了一下这本书后，我就发现我被这本书的书名欺骗了。本书实际上是一本帮助应用程序员和系统程序员理解调试过程的书籍。当然，本书名为“黑客”，也不是为了哗众取宠，而是为了突出本书的写作方式是从“黑客”的角度来看待程序（特别是保护机制）的弱点，介绍了黑客利用调试器和反汇编器来寻找程序弱点和实施攻击的方法，提醒程序员应该如何克服程序的弱点、如何抵御“黑客”的各种攻击。但是，正如作者所言，本书既不是一本教授黑客技术的手册，也不是一本关于防御黑客的保护指南，而是一本探索商业程序的保护机制以及学习调试器的工作原理和使用方法的指南。

本书结合带有详尽解释的源代码，揭示了各种调试器实用使用技巧，说明了如何操作调试器以及如何克服障碍和修复调试器。通过本书，程序员将学会如何弄清楚计算机系统内部的结构、如何重建没有提供源程序的程序的运行算法、如何修改程序以及如何调试驱动程序。本书还详细介绍了在Windows和UNIX操作系统中调试应用程序和驱动程序的方法。

本书的作者Kris Kaspersky是一位经验丰富的“代码挖掘者”，他解决了许多与安全和系统编程有关的问题，包括编译器的开发、优化技术、安全机制研究、实时操作系统内核的创建、软件保护，以及反病毒程序的创建等等。他还是一位著作颇丰的技术作家，出版了大量涉及破解、反汇编和代码优化的文章和书籍。

译者通过翻译本书获益匪浅。我编写程序多年，也使用了多年的调试器，但是本书讲述的调试技术中有许多都是我过去没有接触过，甚至没有听说过的。本书的许多内容也使我有恍然大悟之感，解决了多年来的疑惑。本书还有一些内容使我有胆战心惊之感：我们习以为常的许多代码原来都是极不安全的，可能被黑客利用来攻击我们的系统。幸运的是，

我从本书中了解到了这些知识，在将来的编程中就能充分利用这些调试技术、有意识地避免存在问题的编码方式。因此，如果你是一名程序员，本书将有益于你编写出健壮安全的程序；如果你对破解程序感兴趣，你将从本书中学到发现程序的弱点并通过这些弱点来破解程序的方法（但是千万不要有违法的行为）。

由于译者的水平所限，翻译这样的一本集“黑”（黑客或者破解者）“白”（应用程序员或者系统程序员）两道技术于一体的技术指南，真是一种挑战。译文中肯定存在错误和疏漏，敬请读者批评指正。

感谢电子工业出版社博文视点资讯有限公司为本书的出版付出的努力。特别感谢本书的责任编辑顾慧芳女士，她细心地改正了许多译者“脑是笔非”的谬误，她的努力和高效使本书得以快速出版。

周长发

2006 年 4 月

前　言

许多聪颖敏锐的人本能地渴求破解疑难。他们不辞辛劳地探索周围事物的本质，而不是刻意地进行破坏。只要放眼四顾就能发现：原子科学家分裂原子，分析家将分子分解为更小的分子，数学家积极地使用分解法。可是却没人责备他们！

破解并不等同于破坏。破解是本能的好奇心和渴望了解周围世界的一种行为表现。反汇编清单、机器指令、SoftIce 的黑屏等都让我们想起已成为过去的 MS-DOS 时代，它们至今依然引人入胜、充满魅力。除此之外，就是满世界的隐藏机制和保护代码。不要在地图上寻找它们，它们的世界仅仅存在于打印出来的纸片上、在向最令人着迷的职位敞开的技术手册中，以及在屏幕前度过的许多不眠之夜里。

本书既不是一本讲授破解技术的手册，也不是一本关于防御黑客的保护指南，诸类书籍已经多如牛毛。确切地说，本书是一个代码挖掘者的“旅行笔记”。读者将检验英特尔的编译器，探索商业程序的保护机制，并且学习调试器是如何工作的以及如何熟练地使用它们。一般而言，只要你没被吓得马上合上这本书并把它扔到一边的话，你就会学到许多新颖有趣的东西。

关于作者

笔者是一个不修边幅的年轻人（写作本书时 28 岁），既不关心周围的事物，也不照顾自己的身体，独自踯躅在机器代码丛林和技术规范的迷宫中。本人不善交际，像食肉类啮齿动物一样，过着一种与世隔绝的生活，除了偶尔出去看看星星之外，从不轻易离开自己的老鼠洞。个人生活很不幸（而且似乎也不会在将来变得幸运些），因此从早到晚惟一消磨时光的方式就是把自己完全沉浸在工作中。

从孩提时候（也可能更早，只是我记不起来了）开始，我就沉迷于计算机。主要的研究领域包括逆向工程（反汇编技术），寻找现有保护机制的弱点或漏洞，以及开发自己的保护系统。然而，计算机并非是我的惟一爱好，甚至也可能并不是我的主要嗜好。除了总是关心计算机硬件和漫游在保护代码的丛林中之外，我还一直着迷于夜晚的天空和我的望远镜。在最近一段时期，我的时间更多地花在写作上，用于阅读的时间反而较少。我并不是

简单随意地选择破解作为自己作品主题的。选择这类主题是源于本人对计算机“为什么会如此”的本能的好奇心，以及对使用撬棍和锤子（当然这是一种比喻）来破解事物的渴望。是否可能通过其他的方式来理解选择这一主题的理由呢？

如果说黑客是沉迷于探索宇宙的人，那么我就是一个黑客。

本书的写作目的和读者对象

本书原本是为专业人士量身定做的。然而，发表在因特网上的几个试读章节并未吸引专业人士的兴趣。他们不喜欢用简单的语言来阐述复杂的问题，认为这样水分太多。但是刚入行的代码挖掘者们并不赞成他们的观点，真可谓仁者见仁，智者见智。自然，每一个读者都要求书的格式最便于自己阅读。但是，要满足这种愿望是不可能的，一本书不可能同时满足不同类型的读者的兴趣（特别是对一本并非简单易懂的书来说，更是如此）。

我选择破解新手作为本书的最广泛和最急需的读者群。专业人士并不需要这类书籍。许多专业人士告诉我，他们仅对分散在全书不同章节中的若干页内容感兴趣，因此他们仅会粗略地翻阅本书。这类话并未使我觉得难受！相反，他们帮助我更好地了解本书的定位和目标。

“专业人士”和“新手”只是一种习惯性的叫法，而且许多的新手都可能在自己的专长上轻易地击败某些专业人士。真正的专业人士是很少的。因此，阅读本书之前不可能知道是否能在本书中找到你所不了解的知识。而阅读本书是让你知道这一点的惟一方法。

本书并非为黑客而写！虽然书中描述了攻击某些众所周知的系统的方法，并将它们看成是现成的完整技术，但这些只是信息，而不是行动的系统指南。没有人可以豁免恶意破坏计算机系统行为的法律责任。因此，在应用新学到的知识之前，你应该先阅读刑事法典并认识一些律师。在自由民主的社会中，法律仅仅表明为一种描述彼此之间关系的系统，这种系统早已形成体系，并且保护大多数人的利益。然而什么是大多数人的愿望呢？你说对了！就是面包和马戏（译者注：泛指统治者为了笼络人心所施展的一种小恩小惠的手段）。对于面包，每一个人都或多或少地明白其意义。但是马戏的情形就要复杂一些。音频和视频工业在不断地悲惨衰退，在这种背景下，反对侵犯版权运动的规模是十分可怕的，甚至侵害了个人用户和整个世界的利益。有些信息安全领域的工程和科学研究被部分或者完全地禁止了。用户甚至被剥夺了用反汇编器查看卖给他们的产品的代码的权利。可看不可摸！可摸不可尝！可尝不可咽！信息就如同空气和水一样，是公共资源。我们的思想和观点，虽然我们切实地认为它们是“我们自己的”，但是实际上都是很久以前就已经创造和发表的一些思想和观点的组合。新颖的发明和聪明的观点都是在理解了曾经读到和听到的知识之后产生的。

研究保护机制的黑客和开发人员并非只是对手，他们也是同事。如果你认为黑客是一种寄生虫，他们寄生于建造高质量的保护机制的程序员的能力缺陷，那么你也必须承认程序员也是寄生虫，他们寄生于用户没有能力自己编写程序！

破解与编程具有许多的共同点。创建高质量且可靠的保护机制需要：（1）低级语言编程技巧；（2）协调运用操作系统、驱动程序和设备的能力；（3）具备如下知识，即现代处理器的体系结构、特定编译器的典型代码生成特征，以及所使用的程序库的整体结构。在这种编程级别上，编程和破解之间的差别是非常小的，甚至很难在它们之间划出明确的界限。

首先需要指出的是，与其他的软件组件一样，每种保护都需要仔细且全面的测试，以评测其可用性。这里所说的“可用性”可以理解为它抵御合格的用户使用破解工具（加密盘拷贝软件、虚拟设备仿真器、窗口和消息侦探程序、文件和注册表监视程序，等等）来试图攻破它的能力。保护质量并不是用强度来评测的，而是以实现保护机制所需的人时（译注：即一个人一天完成的工作量）与攻破它所用的人时之间的关系来衡量的。经过长时间的运行，任何保护系统都可被攻破，因为破解只是时间、金钱、黑客的水平和努力的综合结果。然而，高水平的保护系统必须防止被轻易破解的可能性。下面的实例有助于理解这一论断：考虑使用坏扇区（在每一种存储介质中都是罕有的）来实现的保护机制，如果它因为错误的 EDC 和 ECC 字段而不能识别这类粗糙的仿真，那么该保护机制将是无效的。另一个例子是：通过创建能仿真源盘结构所有特征的虚拟 CD-ROM 驱动器，将实现利用 CD 螺旋形轨迹的几何特征来实现的保护机制，即使其实现不存在漏洞。注意，即使你不是一个黑客也能够做到这一点，只要运行自动破解这类保护机制的程序 Alcohol 120%就足够了。

保护机制的设计错误会使其开发人员付出惨重的代价。但是，没有人能保证可以避免这类错误。试图在软件保护的开发中应用科学方法是可笑的。黑客们嘲笑学术型的工作。实际上，任何的此类保护都可以无须过分花费心思地在 15 分钟内被攻破。下面是一个粗略而直观的例证：任何人使用即便是最古老的飞机（就像 WDB 这类飞机）都可攻占防守策略上没有考虑防空的堡垒，更不用说是使用战斗机和轰炸机了（SoftIce 像战斗机，而 IDA 就如轰炸机）。

为了设计保护机制，程序员至少必须概略地了解对手的工作方法和技术工具。精通这类工具的水平不低于对手当然更好。破解程序的实际经验是非常有用的，因为这些经验能帮助程序员仔细地研究攻击方的战略和战术，有利于组织最佳的防守，还能使程序员检测并加固黑客可能攻击的目标，并将主要的可用资源集中到这些目标上。这意味着开发保护机制的程序员必须了解黑客的心理，并且开始像黑客一样地思考问题。

因此，精通了信息保护技术也就等于精通了破解技术。如果你不知道保护机制是如何

被破解的，不清楚它们易被攻击的弱点，也不了解黑客所用的工具，那么你肯定不能创建出坚固、廉价且易于实现的保护机制。那些仅从保护的观点来考虑安全的书籍其缺点是显而易见的，就像认为存储设备只能写信息一样，它们没有实用性。

有一种普遍的观点认为，公开出版安全系统的漏洞弊大于利，应该禁止。换句话说，支持这种观点的人认为自己没有能力创建一种有价值的防拷贝机制，却又不愿意承认自己的错误。因此，为了不破坏第一只迷途的啄木鸟所建立的文明，就必须射杀所有其他的啄木鸟。套用一句众所周知的谚语“预先得到警报就能预先做好准备”，这里我再举个例子，考虑制药工业。假设有一个广告吹嘘一种未经测试但有效的药物，声称它包治百病，并坚持强制使用它，专业人士会如何看待这类促销？同时，局外人无权对其进行化学分析，无权公开发表调查结果，以揭露该“万能药”只不过是一种副作用很大的低质阿司匹林。哈！发表这类真相会很大程度地降低用户的购买热情，他们会更喜欢其他药厂的产品。

在这种情形中，应该谴责谁呢？是欺骗用户的公司还是揭露真相的研究人员？如果你认为这种比拟是不对的，那么请回答如下的问题：什么是保护机制的目标？这类保护机制应该满足什么样的要求？每一种技术都有它的局限和副作用，声称保护坚不可破的广告口号是荒谬的。一种媒体如果可以播放，那么它也是可以拷贝的，惟一的问题是怎样拷贝。禁止破解并不能改变什么。这样的禁令并不能阻止受利益驱使的人们进行批量非法复制受保护的磁盘之类的活动。另一方面，我们这些合法用户将遭殃。有一些人无法抵制窥视黑匣子的诱惑而试图猜测它是如何工作的，对此我们无能为力。在保护机制领域，垃圾和精品并不是一目了然的。只有依靠供应商的权威信息，或者有系统地采购各种产品，这甚至还不能保证可以从市场上获得任何有价值的保护。这是真实的！商业保护机制的质量是如此糟糕，它们甚至不能抵抗数百万普通用户中的任何一位所运行的一个自动程序复制软件的攻击。是否好意思说阻止用自动复制软件拷贝产品是任何一种保护的最低合理要求呢？一种理想的保护必须经受住装备了强大的软件和硬件破解工具的高水平黑客的攻击。高质量的保护机制是存在的，但是在市场上找不到。为什么呢？缺乏值得信赖的关于特定保护产品的功效信息，因而并不能帮助用户做出清醒的选择。因此，劣质产品的生产商并不需要担心。

版权法律的倡导者们必须明白，保护机制越被激烈地破解，他们在该领域开发所取得的进步就越大！在这种条件下，开发人员具有很强的动力来创建高质量和有竞争性的保护产品。我将从头开始说明这一点。基尔霍夫规则（Kirchhoff's rule，即所有密码系统的基本规则）表明：任何密码的强度都取决于密钥的秘密性。这说明密码破译者知道加密和解密过程的所有细节，只是不知道秘密的密钥。每一种高质量保护的不同特征就在于其算法的细节描述。不必掩盖真相，因为每一个黑客都能利用调试器和反汇编器来揭示真相。毕竟，事物的信息完整性是不能抹杀的。试图掩盖明显的瑕疵和缺点是非法的。公开真相无妨于

好的事物，但是坏的事物就像瘟疫一样，害怕被公开。

现在已经处于“数字时代”，我们要求修订相关的法律，因为它们只是空谈而没有任何实质内容。有一些法律诉讼的对象是使用保护程序的合法用户，许多这类诉讼都使人莫名惊诧。虽然大多数这类诉讼都友好地和解了，但这种趋势却是很可怕的。谁知道会不会有一天，由于<Shift>键能让用户阻止 Windows 的自动运行功能且有些保护机制以此为基础，而导致<Shift>键被禁止使用呢。因此，有些事情在今天看来是疯人的胡言乱语，明天就可能成为真实的法律诉讼。

“Digital Millennium Copyright Art”法案禁止传播用于绕开现存保护机制的技术、设备和服务。律师们试图通过反对罪犯和坏蛋来保护世界。可是，有必要在信息技术领域区分破解和研究活动。怀有恶意的破解者至少应该受到谴责、罚款甚至被判入狱。然而惩罚的程度应该由破解所造成的破坏来确定。将黑客与恐怖分子相提并论是没有用的！

没有什么书只从保护的观点来考虑安全的各个方面，只能写入而不能播放的设备也没有任何的实际应用价值。但是，本书并非是专为保护机制的开发人员而写的。我不想强调任何的这类观点，因为保护问题更多的是组织的而非技术的问题。保护机制软件开发人员不需要这类书籍，也很少听取实现保护机制的意见。至于黑客，在阅读这类书籍后，他们就丧失了进一步自学的动因。

术语“黑客”具有多种解释。将这些解释列举出来并强调其中的一些是没有意义的。最好的解释是，黑客就是试图理解任何问题的本质的个人。这些人构成了本书预计的读者。只要可能，我都将概括主要的思想，而不强调特定的实现。这并不是说不能在本书中找到完整并且可以直接使用的方法。相反，书中包括许多这样的方法。可是，我并不想向你提供可以直接使用的“钓鱼工具”。

我不会试图教你“如何钓鱼”，而是会更进一步，教你培养一些自学的技巧。在任何情形中都能找到解决问题的方法，即使可用的工具是有限的。典型和标准的方法是软弱而无用的，这已成为一条准则。任何的技巧都与特定的环境相关。采用传统的方法来学习已经太迟了，因为许多人都没有时间。大部分程序员都不得不边学边干，当详细地学会一种技术时，它已变得陈旧了。

请注意，保护系统进步得慢一些。在过去的几年中，并没有出现任何重要创新，也没有重要的、新的保护机制得到广泛的应用。漏洞暴露了，其特别的背景是流行的且得到广泛使用的保护系统的质量在下降。不过，这段时间还是出现了一些关于该论题的有价值的教科书和出版物。由此导致的后果是市场上充斥了几乎没有经过很好地设计、规划和实现的保护机制。许多作者通过回忆在学校学到的代数（并不很容易）来创造他们自己的算法。如果仔细考虑，就能发现这些算法都不具备强壮的密码保护，并没有用到现代数学中的宝贵知识。然而，还是出现了一些高质量的出版物。我将在本书的适当章节加以引用。

本书的主要目的是教你如何自行获取必要的知识和技巧，有时甚至不需要使用相关的文献和信息。现代丰富的知识导致自行获取知识的技巧萎缩了。乍看这似乎很荒谬，但是缺乏文献而训练大脑，要好过信息的丰富多彩。比如，我利用 Debug.com 来学习 8086 汇编语言。除了该工具和大量的空闲时间之外，我没有任何其他东西。我学习指令运算逻辑的方法是分析其对寄存器和内存的影响。这真是单调乏味的的工作。而在将近 3 个月之后，我融会贯通地获得了汇编语言的知识（不算其中的一些不重要的指令）。如果当时我有一本手册，这一过程就能缩短为 2 天或 3 天（因为我知道其他平台的汇编语言）。但是，如果这样的话，我就不能像当时那样获得那些如此有用的技巧。

很少有手册是简明易懂或者容易阅读的。很久以前获得的技巧直到现在还是非常重要的，因为手册的厚度在不断增加，而手册的质量却变得越来越晦涩难懂。大多数用户面对计算机的心理障碍是一种无助的感觉。这些用户感觉就像正在考试却运气不好的学生，盲目地猜测面前的机器如何操作。然而，用户应该能够掌控这种情形。设想被调用的函数的行为总是与文档中描述的不一样，函数也不工作——这是大家都熟悉的情形，不是吗？

在这种情形下，通过反汇编来分析问题是非常有用的。也许另一种方法是最好的。问题可能就出在重要的变量上。只有新手才会认为方法（函数）在任何情形下都是工作的，好像黑客按下某个按钮就能得到奖金一样。如果不是这种情形又该如何做？如果任务的方法根本就不对又该如何做？这时你应该学会因地制宜。本书的主要目标就是：教你如何因地制宜地解决问题。

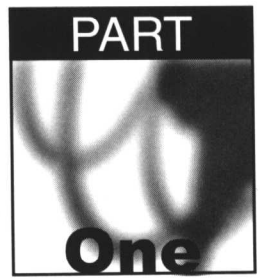
目　录

第 1 部分　调试工具入门

第 2 部分　调试工具入门

第 3 部分　反调试技术

第 5 部分　PE 文件



第 1 部分
调试工具入门