HZ BOOKS
华章科技

深度揭秘软件逆向分析技术的流程与方法，理论与实践完美结合

由安全领域资深专家亲自执笔，看雪软件安全网站创始人段钢等多位安全领域专家联袂推荐

安全技术大系
SECURITY

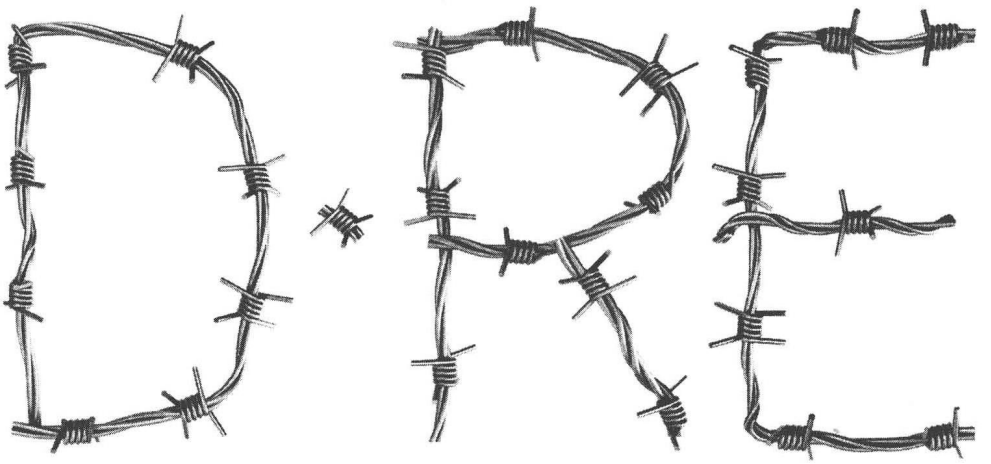
DISASSEMBLY AND REVERSE ENGINEERING FOR C++

C++反汇编与逆向分析技术揭秘

钱林松 赵海旭 著

机械工业出版社
China Machine Press

安全技术大系
SECURITY



DISASSEMBLY AND DISASSEMBLY INEERING FOR C++

C++反汇编与逆向分析技术揭秘

钱林松 赵海旭 著

机械工业出版社
China Machine Press

本书既是一本全面而系统地讲解反汇编与逆向分析技术的安全类专著，又是一部深刻揭示 C++ 内部工作机制的程序设计类著作。理论与实践并重，理论部分系统地讲解了 C++ 的各种语法特性和元素的逆向分析方法和流程，重在授人以渔；实践部分通过几个经典的案例演示了逆向分析技术的具体实施步骤和方法。

全书共分为三大部分：第一部分主要介绍了 VC++6.0、OllyDBG 和反汇编静态分析工具的使用，以及反汇编引擎的工作原理；第二部分以 C/C++ 语法为导向，以 VC++6.0 为例，深入解析了每个 C/C++ 知识点的汇编表现形式，包括基本数据类型、表达式、流程控制语句、函数、变量、数组、指针、结构体、类、构造函数、析构函数、虚函数、继承和多重继承、异常处理等，这部分内容重在修炼“内功”，不仅讲解了调试和识别各种 C/C++ 语句的方法，而且还深入剖析了各知识点的底层机制；第三部分是逆向分析技术的实际应用，通过对 PEiD、“熊猫烧香”病毒、OllyDBG 调试器等的逆向分析将理论和实践很好地融合在了一起。

本书适合所有软件安全领域的工作者、想了解 C++ 内部机制的中高级程序员，以及对 Windows 底层原理感兴趣的技术人员阅读。

图书在版编目（CIP）数据

C++ 反汇编与逆向分析技术揭秘 / 钱林松，赵海旭著 .—北京：机械工业出版社，2011.9

ISBN 978-7-111-35633-2

Ⅰ . C…　Ⅱ . ①钱…　②赵…　Ⅲ . C 语言－程序设计　Ⅳ . TP312

中国版本图书馆 CIP 数据核字（2011）第 162655 号

机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码　100037）

责任编辑：姜　影

北京京师印务有限公司印刷

2011 年 10 月第 1 版第 1 次印刷

186mm×240mm • 26.5 印张

标准书号：ISBN 978-7-111-35633-2

定价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

前　言

为什么写这本书

“时下的IDE很多都是极其优秀的，拜其所赐，职场上的程序员多出十几倍，但是又有多少能理解程序内部的机制呢？”

——侯捷[①]

随着软件技术的发展及其在各个领域的广泛应用，对软件进行逆向工程，然后通过阅读其反汇编代码来推断其数据结构、体系结构和程序设计思路的需求越来越多。逆向工程技术能帮助我们很好地研究和学习先进的软件技术，特别是当我们非常想知道某个软件的某些功能究竟是如何实现的，而手头又没有合适的资料的时候。

我国的软件产业落后于西方，甚至在某些方面落后于邻国的印度和日本。如果我们能够利用逆向技术去研究国外的一些一流软件的设计思想和实现方法，那么我国的软件技术将会得到极大的提升。目前，国内关于逆向分析技术的资料实在是少之又少，大中专院校的计算机相关专业对此项技术也尚未有足够的重视。

有很多人认为研究程序的内部原理会破坏“黑盒子”封装性，但是如果我们只是在别人搭建好的平台上做开发，那么始终只能使用别人提供的各种未开源的SDK，会一直被别人的技术牵制着。如果我们能够充分掌握逆向分析的方法，就可以洞悉各种SDK的实现原理，

---

① 著名技术专家和IT教育工作者，尤其精通C++和MFC，计算机图书作家、译者和书评人。

学习各种一流软件所采用的先进技术，取长补短，为我所用。若能如此，实为我国软件产业之幸。

我当初学习逆向技术时完全靠自学，且不说这方面的书籍，就连相关的文档和资料也都极度匮乏。在这种条件下，虽然在很努力地钻研，但学习进度却非常缓慢，花费几天几夜完成对一个软件的关键算法的分析是常有的事。如果当初有一本全面讲解反汇编与逆向分析技术的书供参考，我当年不仅能节省很多时间和精力，而且还能少走很多弯路。因为有这段经历，我斗胆争先，决定将自己多年来在反汇编与逆向分析技术领域的一些经验和心得整理出来与大家分享，希望更多的开发人员在掌握这些技术后能更好地将其应用到软件开发的实践中，从而提高我国软件行业的整体水平。由于个人能力有限，书中的疏漏在所难免，还请各位同行和读者多多批评和指正。

本书适合的读者

首先，无论大家从事哪个行业，在开始阅读本书之前，都需要具备以下几个方面的基础知识：

- 数据结构的基础知识，如栈结构存取元素的特点等。
- 汇编的基础知识，如寻址方式和指令的使用等。
- C/C++ 语言的基础知识，如指针、虚函数和继承的概念等。
- 熟悉 Microsoft Visual C++ 6.0 的常用功能，如观察某变量的地址、单步跟踪等。

具备了上面这些基础知识，就能根据自己的实际需求来学习本书的内容。

如果你是一位软件研发人员，你将通过本书更深入地了解 C++ 语法的实现机制，对产品知其然更知其所以然，能够在熟练阅读反汇编代码后，使调试技术也有质的提升。

如果你是一位反病毒分析或者电子证据司法取证分析人员，通过逆向恶意软件样本，可以进行取证分析处理，例如，可以归纳开发者的编写习惯，推定开发者的编程水平，甚至可以进一步判定某病毒样本是否与其他某些病毒为同一作者所为。

如果你是高等院校计算机相关专业的教师或学生（本科或本科以上），软件逆向分析技术可以给你带来崭新的职业空间，使你有足够的技术竞争力面对软件研发行业。同时，信息安全行业也会是你新的求职方向。

本书内容及特色

在本书的内容结构上，笔者结合自己的学习经历和对 C++ 反汇编与逆向分析技术的了解进行了较为周详的设计，将全书划分为三个部分。

第一部分　准备工作（第 1 章）

在软件开发过程中，程序员会使用一些调试工具，以便高效地找出软件中存在的错误。

在逆向分析领域，分析者也要利用相关的工具来分析软件行为和验证分析结果。本书第一部分简单介绍了几款常用的逆向分析辅助工具和软件。

第二部分　C++ 反汇编揭秘（第 2~13 章）

如果要评估一位软件开发者的能力，一是看设计能力，二是看调试水平。一般来说，大师级的程序员对软件逆向分析技术的理解都很深入，他们在编写高级语言代码的同时，心里还会浮现出对应的汇编代码，他们在写程序时就已经非常了解最终产品的真正模样，达到人机合一的境界，所以在调试 Bug 的时候游刃有余。逆向分析技术重在代码的调试和分析，如果你本来就是一个不错的程序员，学习这部分内容就是对你“内功”的锻炼，这部分内容可以帮助你彻底掌握 C/C++ 的各种特性的底层机制，不仅能做到知其然，而且还能知其所以然。这个部分以 C/C++ 语法为导向，以 VC++ 6.0 为例，解析每个 C/C++ 知识点的汇编表现形式，通过整理其反汇编代码来梳理其流程和脉络。这部分内容重在讲方法，授人以渔，不重剑招，但重剑意。如果大家照此“精修”，可达到看反汇编代码如同看武侠小说的境界。

第三部分　逆向分析技术应用（第 14~17 章）

这是本书的最后一部分，以理论与实践相结合的方式，通过对具体程序的分析来加深大家对前面所学理论知识的理解，从而快速积累实战经验。第 14 章分析了 PE 文件分析工具 PEiD 的工作原理；第 15 章对“熊猫烧香”病毒进行了逆向分析；第 16 章分析了调试器 OllyDBG 的工作原理；第 17 章讲解了反汇编代码的重建与预编译。通过对这部分内容的学习，大家可以通过实际应用领略逆向分析技术的魔力。

如何阅读本书

逆向分析技术具有很强的综合性和实践性，要掌握这项技术需要耐心和毅力。建议大家从最简单的程序入手，按照本书安排的顺序逐章阅读，在学习的过程中逐步提高难度，一边看书，一边积极思考和总结。对于一些理论知识，如果你兴趣不大，在初学阶段可以跳过，待以后需要提高时再回过头来阅读，可以暂时跳过的知识我都在书中做了说明。

随着时间的积累，你会逐渐形成一套属于自己的分析代码的风格和习惯。这样一来，任何软件在你眼中都没有了神秘感。

联系作者

本书的讨论和勘误建立在看雪安全论坛（http://bbs.pediy.com/）的图书项目版块中，我们会在这里发布本书的勘误和其他对大家有用的增值服务。大家也可以在这里发表对本书的意见和建议，更重要的是，大家还能在这里结交到一些志同道合的朋友。同时，也欢迎大家直

接通过 QQ（159262378）或 E-mail（ollydbg@foxmail.com）联系我，由于平时上网较少，如果回复不及时，还请谅解。

致谢

在本书写作的过程中，我得到过很多从事逆向分析技术的同行的指点和帮助，在此表示感谢，其中一部分包括段钢、姚辉、彭国军、雷建云、林子深、印豪、单海波、王清、谭文、邵坚磊、田阗、方志强、岳磊、余坦兮、李长坤、曹剑锐、胥国银、胡晓春、吕明坤、张慧等（排名不分先后）还有很多朋友在我写作本书的过程中给予了帮助，这里一并表示感谢。

特别感谢本书的策划编辑杨福川和责任编辑姜影，他们花费了许多时间和精力来校正本书中的各类错误。正是他们的敬业和努力，才使得本书能在保证质量的前提下顺利出版。

钱林松

2011.7 于武汉

目　录

第一部分

准备工作

❑ 第 1 章　熟悉工作环境和相关工具

第 1 章　熟悉工作环境和相关工具

1.1　调试工具 Microsoft Visual C++ 6.0 和 OllyDBG

在软件的开发过程中，程序员会使用一些调试工具，以便高效地找出软件中存在的错误。而在逆向分析领域，分析者也会利用相关的调试工具来分析软件的行为并验证分析结果。由于操作系统提供了完善的调试接口，所以利用各类调试工具可以非常方便灵活地观察和控制目标软件。在使用调试工具分析程序的过程中，程序会按调试者的意愿以指令为单位执行。调试者可以随时中断目标的指令流程，以观察相关计算的结果和当前的设备情况，也可以随时继续执行程序的后续指令。像这样使用调试工具加载程序并一边运行一边分析的过程，我们称之为“动态分析”。

本书中使用了两款调试工具：Microsoft Visual C++ 6.0 和 OllyDBG。对于调试版（Debug 编译选项组），我们使用 Microsoft Visual C++ 6.0 进行调试，它可以将 C++ 源码反汇编，方便学习；对于发布版（Release 编译选项组），我们使用 OllyDBG 进行调试分析，它的调试功能十分强大。Microsoft Visual C++ 6.0 的调试功能相对简单，同时有源码做对照，故不过多讲解。OllyDBG 的默认功能界面如图 1-1 所示。

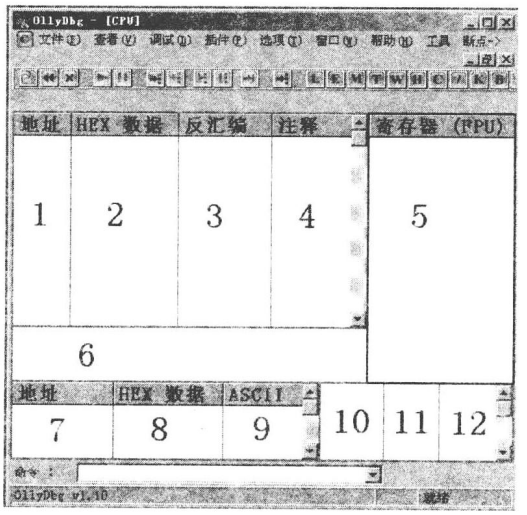


图 1-1　OllyDBG 的默认功能界面

图 1-1 中的标号说明如下：

❑ 1：汇编代码对应的地址窗口
❑ 2：汇编代码对应的十六进制机器码窗口
❑ 3：反汇编窗口
❑ 4：反汇编代码对应的注释信息窗口
❑ 5：寄存器信息窗口
❑ 6：当前执行到的反汇编代码的信息窗口
❑ 7：数据窗口——数据所在的内存地址
❑ 8：数据窗口——数据的十六进制编码信息
❑ 9：数据窗口——数据对应的 ASCII 码信息
❑ 10：栈窗口——栈地址
❑ 11：栈窗口——栈地址中存放的数据
❑ 12：栈窗口——对应的说明信息

熟悉了各窗口视图的功能之后，我们来更深一步了解 OllyDBG 的操作方法。首先介绍一下 OllyDBG 的快捷键，掌握各个快捷键的使用，可以提高分析效率。OllyDBG 的基本快捷键及其功能如表 1-1 所示。

表 1-1　OllyDBG 的基本快捷键及其功能

编号	快捷键	功能说明
01	F2	断点，在 OllyOBG 反汇编视图中，使用 F2 指定断点地址
02	F3	加载一个可执行程序，进行调试分析
03	F4	程序执行到光标处
04	F5	缩小、还原当前窗口
05	F7	单步步入，进入函数实现内，跟进到 CALL 地址处
06	F8	单步步过，越过函数实现，CALL 指令不会跟进函数实现
07	F9	直接运行程序，遇到断点处，程序暂停
08	Ctrl+F2	重新运行程序到起始处，用于重新调试程序
09	Ctrl+F9	执行到函数返回处，用于跳出函数实现
10	Alt+F9	执行到用户代码处，用于快速跳出系统函数
11	Ctrl+G	输入十六进制地址，在反汇编或数据窗口中快速定位到该地址处

通过实际操作演练，我们可以进一步熟悉 OllyDBG：调试一个简单的“Hello world”程序，将对话框标题“Hello world”修改为“I Like C++”，步骤如下。

（1）加载可执行程序（如图 1-2 所示）

选择一个调试程序，有多种方案：

❑ 使用快捷键 F3 选择所要调试程序的路径
❑ 在菜单选项中（文件 \ 打开）选择调试程序路径

❑ 将 OllyDBG 加入系统资源管理菜单中，右击选择“打开”

（依次选择 OllyDBG 菜单\选项\添加到浏览器\添加 OllyDBG 到系统资源管理菜单\完成，即可将 OllyDBG 加入系统资源管理菜单中。）

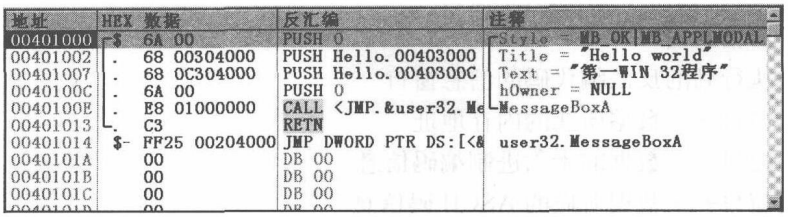


图 1-2 初识 OllyDBG

在图 1-2 中，代码运行到地址 0x00401000 处，对应反汇编指令 PUSH 0，此汇编指令对应的机器码为 6A 00（汇编指令对应的机器码可查询 Intel 的指令帮助手册）。在 OllyDBG 的注释窗口中，已经分析出此汇编指令的含义——OllyDBG 根据 CALL 指令的地址得知这个函数的首地址为 API MessageBoxA 的首地址，进而分析出对应的参数个数和参数功能。

（2）查看 API MessageBoxA 各参数的功能

查看 MSDN 文档，获取 MessageBoxA 各参数的功能，找到弹出对话框的标题参数（PUSH Hello.00403000），此参数保存了字符串“Hello world”的首地址。

（3）定位数据（如图 1-3 所示）

选中数据窗口（如图 1-1 所示），使用快捷键 Ctrl+G，弹出数据跟随窗口。输入查询地址 0x00403000，单击“确定”按钮快速定位到该地址处。

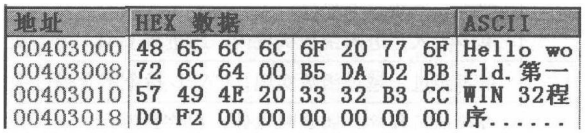


图 1-3 初识数据窗口

（4）修改数据（如图 1-4 所示）

找到要修改数据的地址所对应的 HEX 数据，在图 1-3 中，地址 0x00403000 对应的十六进制数据为 0x48。双击 HEX 数据窗口中“48”处，弹出对应的编辑数据对话框，如图 1-4 所示。去掉对“保持大小”的勾选，可向后修改数据。在 ASCII 文本编辑框中，输入“I Like C++”，由于 C\C++ 中字符串以 00 结尾，需要将字符串最末尾的数据修改为 00。选择十六进制编码文本框，在最末尾处插入 00。单击“确定”按钮，完成对字符串的修改。

（5）调试程序

使用快捷键 F8 单步调试运行，连续按 4 次 F8 键，单步运行 4 条汇编指令，观察栈窗口变化，如图 1-5 所示。函数 MessageBoxA 所需参数都已被保存在栈中。按快捷键 F7 可跟进