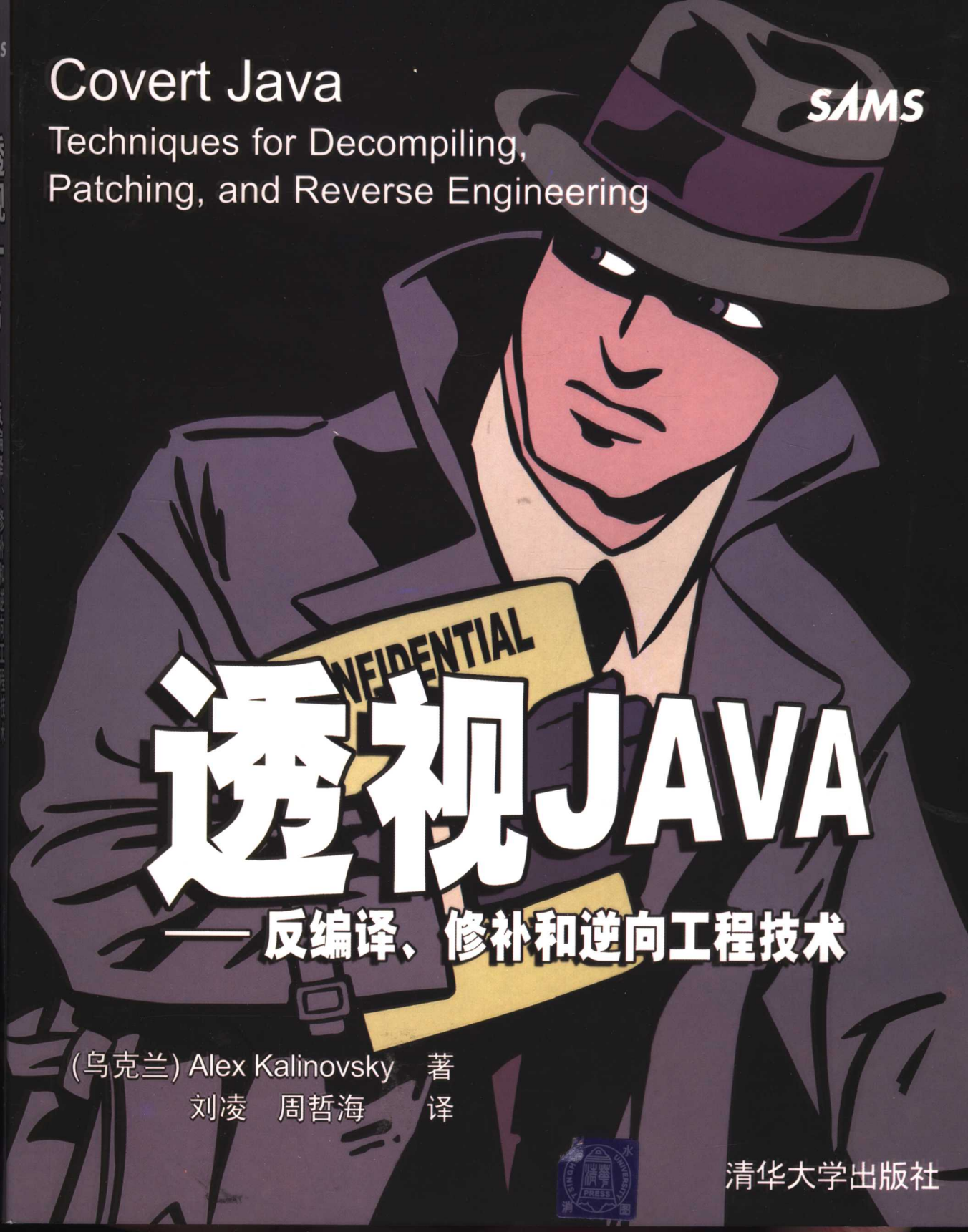
Covert Java

Techniques for Decompiling, Patching, and Reverse Engineering

SAMS



透视JAVA

——反编译、修补和逆向工程技术

(乌克兰) Alex Kalinovsky 著
刘凌 周哲海 译

清华大学出版社

透视 Java

——反编译、修补和逆向工程技术

(乌克兰) Alex Kalinovsky 著
刘凌 周哲海 译

清华大学出版社
北 京

图书在版编目(CIP)数据

透视Java——反编译、修补和逆向工程技术/(乌克兰)卡里诺维斯盖(Kalinovsky,A.)著；刘凌，周哲海 译.
—北京：清华大学出版社，　2005.10
书名原书：Covert Java：Techniques for Decompiling，patching，and Reverse Engineering
ISBN 7-302-11414-5

Ⅰ. 透… Ⅱ.①卡… ②刘… ③周… Ⅲ. JAVA 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2005)第 082175 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：曹　康
文稿编辑：王　黎
封面设计：康　博
版式设计：康　博
印 刷 者：北京鑫丰华彩印有限公司
装 订 者：三河市化甲屯小学装订二厂
发 行 者：新华书店总店北京发行所
开　　本：185×230　**印张**：17.25　**字数**：298 千字
版　　次：2005 年 10 月第 1 版　2005 年 10 月第1 次印刷
书　　号：ISBN 7-302-11414-5/TP·7499
印　　数：1～4000
定　　价：36.00 元

前言

目前已经出版了许多关于 Java 的书籍，但令我吃惊的是关于同一主题的书籍却是如此之多。在 www.amazon.com 上搜索关于 Enterprise JavaBeans (EJB)的书籍会返回 50 多个结果。EJB 是一项复杂的技术，今天每个 Java 开发人员都得将它写在自己的简历上，那我为什么还要在 Java 的书架上增加另一本书呢？因为我相信有一些很少公开的开发技术，在正确使用时，可以产生令人惊讶的结果。其中很多方法是用于处理核心 Java 概念和问题的，所以可以用在多种应用程序中。本书中提供的技术是针对 Java 开发中常见问题的非正式解决方案。其中一些是有争议的，应该非常小心地使用。但是所有这些技术都是达到目标的强有力的方法。掌握这些技术，您就可以在其他人还在忙于了解真正问题症结的时候就给出解决方案，从而将自己与其他大部分开发人员区别开来。您可能已经用过本书中给出的一些技术，如果是这样的话，恭喜您，不过我确信，在您仔细阅读我在此给出的建议后，您至少将会掌握一些有用的新窍门。

本书的大部分专注于通常被称作破译(hacking)的技巧。破译在媒体中使用得相当普遍，并且通常带有负面含义。黑客(hacker)经常被描绘成疯狂的讨厌鬼，他们想极度膨胀自己的自尊，从某些情况来看，这确实是真的。但是本书中的方法是提供给专业软件开发人员的，每种技术都有其实实在在的应用。

本书读者对象

Java 开发人员和架构师最有可能从本书中获得最大收益。为了真正理解本书中描述的问题和解决方案，您应该至少开发过一些重要的 Java 应用程序以及处理过第三方代码。这并不是说初级开发人员就不会从本书中获益。为了保持本书的简洁并

集中于主题，几乎没有涵盖那些众所周知或已有很好文档记录的主题。例如：在谈及破译非公共类成员时，本书没有解释每个访问限定符的具体含义。这些信息可以在 Internet 上或详细涵盖这些主题的书籍中找到。本书就是关于突破通常认为的边界的终极技术的。值得指出的是这里介绍的技术大部分是彼此相互独立的。由于介绍的材料遵循“最常见较简单的方法最先”的顺序，因此您可以随意跳过一些章节直接进入所感兴趣的章节。第 1 章“入门指南”中有一节简要介绍了每种技术以及运用每一种技术的时机。所以，我推荐您首先熟悉一下第 1 章。

Hacking(破译)的道德和法律问题

本书大多数章节介绍的均是严谨的技术，但要认识到并不是所有的技术在处理应用程序时都可以随意使用，这一点极其重要。本书给出的方法不是每种都是“黑客技术”，但如果在使用前没有复核法律后果，就会让您惹上麻烦。首先让我们给黑客技术下一个宽泛的定义，然后考虑如何对待麻烦。韦氏辞典中给黑客的定义如下：在编程和用计算机解决问题方面的专家。但随后给出了另一个解释：“非法访问计算机系统并且有时篡改计算机系统内信息的人”。成为编程方面的专家固然十分了不起，但摆弄非法材料会让您坐牢的。本书是为好孩子编写的，如果您是个坏孩子的话，现在就请停止阅读本书，在测试组找份新工作。任何信息或是发现都是双刃剑，既可以用来做好事也可以用来做坏事。不是信息本身而是如何使用信息将决定结果是正面还是负面的。到目前为止已经有一些与数字版权、逆向工程、侵犯著作权相关的立场明确的诉讼案件。已经给公司和个人造成了数百万美元的经济损失以及名誉损失。尽管法律是复杂的，而且许可证协议是律师为律师写的，但绕过法律问题并不是十分困难。以下就是两条简单而有效的基本准则：

- 如果作者希望您为其作品付费，那就一定要这样做。
- 如果您想做一些修改，一定要确认不能伤害作者的利益。

第一条很容易理解，但在使用本书中给出的方法时，第二条是非常重要的，切记。例如：如果您对某人的代码施用逆向工程，以寻找一个 bug 变通方法，作者是不太可能起诉您的。但如果您想借此制作基于完全相同原理的竞争产品，多半作者就会把您告上法庭了。

一定得记住软件都是由像您我这样的人编写的，就像您和我一样，他们也要生活。开放源代码是一种不同的现象，因为这种源代码是可以免费得到的，您不需要

使用极端的方法就可以掌握或改变产品中的某些部分。但是目前开发的大多数软件都是商用软件，很多技术革新都是由厂商来做的。对软件进行破译而不支付许可费用是破坏生产的，因为这会破坏软件市场并且间接伤害开发人员。从隔壁商店盗窃冰淇淋会导致冰淇淋的价格上涨或是商店破产。并且如果您拥有一个面包店，冰淇淋店的老板可能会开始从您这里盗窃甜面包。

上述两个准则涵盖了黑客的道德问题，但通常涵盖法律问题的是版权和知识产权法以及终端用户许可协议(EULA, end user license agreements)。法律是复杂的，很难读懂，但 EULA 是必须要懂的，因为一般而言，EULA 要比法律更为严格。编写这些 EULA 是为作者提供相应的法律不能充分承认的保护，通常，用户在使用产品之前需要明确同意协议中的条款。例如：尽管法律没有禁止逆向工程，但大多数软件产品在其 EULA 中都禁止了。所以在产品上使用本书中说明的技术前，彻底地研究 EULA 是必不可少的。为了避免重复并保持本书内容的严格技术性，后面章节中的内容不涉及与技术相关的法律问题。保证您行为的合法性就是您自己的责任了。

致谢

谨以此书献给我的双亲 Stanislav 和 Lubov Kalinovsky，是他们给了我自出生那天以来的一切。直到随着年龄的增长，我才真正开始理解和感受到家庭对一个人的影响，借此机会，我谨对他们所作的牺牲致以真挚的谢意。这本书也送给对我的人生有着重大影响的另外两个人：我的哥哥 Andrew Kalinovsky、我的继父和顾问 Sergei Boiko。感谢你们！我爱你们！

在我编写这本书的漫长过程中，有很多人都热情地支持我。我要感谢我最亲密的朋友 LaWanda Tetteh 和 Gleb Tulukin，是他们给了我所需要的支持和鼓励。特别要归功于 Amie Koker 的耐心和理解以及 Tricia Riviere 的幽默和睿智。Troy Davis 和 Yves Noel 在审阅本书时提供了他们的技术和个人见解。如果没有 Todd Green、Sean Dixon 和 Sams 出版公司的其他小组成员的技术和专业精神，也就不会有这本书问世。我要向所有我提及和没有提及的、帮我完成这一目标的每个人深表谢意！

我们想聆听您的高见

在阅读完本书后，您就将是我们最重要的批评者和评论员。我们非常重视您的见解，并且希望了解我们何处做得尚可，何处仍需改进，您希望我们出版哪些领域的作品，以及您想要转告我们的至理名言。

作为 Sams 出版公司的助理出版人，我真诚希望得到您的意见和建议。您可以发电子邮件或直接写信给我，让我了解您是否喜欢这本书，以及如何使我们的作品精益求精。

请注意我不能帮您解决与本书主题相关的技术问题。不过我们有一个用户服务小组，我会将与本书相关的专业技术问题转交给他们解决。

请您在写信时务必署名本书书名和作者，以及您的姓名、电子邮件地址和电话号码。我会仔细研究您的意见和建议并与本书作者和编辑分享。

Email： feedback@samspublishing.com

地址： Michael Stephens
Associate Publisher
Sams Publishing
800 East 96th Street
Indianapolis, IN 46240 USA

要了解关于本书、作者或 Sams 出版公司的更多信息，请访问我们的网站：www.samspublishing.com。请在 search 栏中输入书的标题或 ISBN(连字符除外)，搜索要寻找的页面。

目 录

第1章

入门指南

本章主要内容

- 技术综述——使用各种方法的时间和目的
- 利用文件管理器提高程序开发效率
- 示例应用程序的功能和结构
- 快速测试
- 小结

1.1 技术综述——使用各种方法的时间和目的

表 1-1 给出了相应章节将要详细论述的技术的概述。

表 1-1 技术综述

章 节	技 术	作 用
2	反编译类	• 恢复丢失的源代码 • 了解特性和窍门的实现 • 排除无文档说明的代码中的故障 • 修复产品或第三方代码中的紧急程序错误 • 评估自己的代码可能如何被破译

(续表)

章　节	技　术	作　用
3	混淆类	● 保护字节码不被反编译 ● 保护字节码内的知识产权 ● 防止应用程序被破译
4	破译类的非公用方法和变量	● 访问存在但没有暴露的功能 ● 改变内部变量的值
5	替换和修补应用类	● 不必重构整个库文件而改变类的实现 ● 改变第三方应用程序或框架的功能
6	使用有效的跟踪技术	● 创建易于维护和排除故障的应用程序 ● 了解应用程序的内部运作 ● 在现有应用程序中插入调试信息，了解实现的细节
7	管理Java安全措施	● 在关键系统资源的访问上添加或删除约束条件
8	窥探运行时环境	● 确定系统属性的值 ● 确定系统信息，例如：处理器的数量和内存容量局限 ● 确定网络配置
9	用非正式调试程序破译编码	● 对不具有良好跟踪技术的程序进行破译 ● 分析多线程应用程序的控制流 ● 破译混淆的应用程序
10	运用性能分析工具(profiler)进行应用程序运行时分析	● 运用性能分析工具研究堆的使用和垃圾回收频率以提高性能 ● 浏览对象的分配和引用，以发现并修复内存泄漏 ● 研究线程分配和同步，找到死锁和数据竞争问题以提高性能 ● 在运行时研究应用程序以更好地理解其内部结构
11	运用负载测试发现并调整可伸缩性问题	● 创建仿真系统负载的自动测试脚本 ● 分析应用程序如何满足服务级别需要，例如：可伸缩性、可用性和容错性
12	逆向工程的应用	● 对用户界面要素，如：信息、警告、提示、图像、图标菜单和颜色进行破译