HZ BOOKS
华章科技

PEARSON

C和C++安全编码

（原书第2版）

[美] Robert C. Seacord 著　　卢涛 译　　李颖 审校

Secure Coding in C and C++
Second Edition

- C/C++安全编程领域的“标准”著作，资深安全专家撰写，美国CERT主管亲自作序推荐，权威性毋庸置疑
- 详细阐述C/C++语言及其相关库固有的安全问题和陷阱，系统总结导致软件漏洞的常见编码错误，并给出了应对的解决方案
- 对C/C++软件中常见漏洞的危害、被利用方式、检测方法和应对之道进行了全方位讲解，包含大量编码练习，实践性强

机械工业出版社
China Machine Press

C和C++安全编码

（原书第2版）

Secure Coding in C and C++
Second Edition

[美] Robert C. Seacord 著　卢涛 译　李颖 审校

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

C 和 C++ 安全编码（原书第 2 版）/（美）塞克德（Seacord, R. C.）著；卢涛译；李颖审校 . —北京：机械工业出版社，2013.12
（华章程序员书库）
书名原文：Secure Coding in C and C++, Second Edition

ISBN 978-7-111-44279-0

Ⅰ. C… Ⅱ. ①塞… ②卢… ③李… Ⅲ. C 语言 – 程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2013）第 237110 号

本书是 C/C++ 安全编码领域的权威著作，被视为“标准”参考书，由国际资深软件安全专家撰写，美国 CERT 主管亲自作序推荐。本书结合国际标准 C11 和 C++11，以及 C 和 C++ 语言的最新发展，既详细阐述了 C/C++ 语言及其相关库固有的安全问题和陷阱，系统总结了导致软件漏洞的各种常见编码错误，并给出了应对错误的解决方案；又对 C/C++ 软件中常见漏洞的危害、被利用方式、检测方法和应对之道进行了全方位讲解，包含大量编码练习，实践性强。

本文从 C 和 C++ 语言的各个部分分别介绍了可能导致安全问题的软件漏洞：第 1 章介绍安全术语和概念，并指出为何 C 和 C++ 程序中存在如此多的漏洞。第 2 章描述 C 和 C++ 中的字符串操作、常见的安全缺陷以及由此导致的漏洞。第 3 章介绍任意内存写漏洞利用方式，它允许攻击者对内存中任意位置的一个地址进行写操作。第 4 章描述动态内存管理，讨论了动态分配的缓冲区溢出、写入已释放内存，以及重复释放漏洞。第 5 章讨论整数安全问题（即与整数操作相关的安全议题），包括整数溢出、符号错误以及截断错误等。第 6 章描述格式化输出函数的正确和错误的用法，对因这些函数的错误使用所导致的格式字符串和缓冲区溢出漏洞都有讨论。第 7 章重点介绍并发和可能导致死锁、竞争条件和无效的内存访问序列的漏洞。第 8 章描述和文件 I/O 相关的常见漏洞，包括竞争条件和检查时间与使用时间漏洞。第 9 章推荐一些可以整体改善 C/C++ 应用程序安全性的具体开发实践，这些建议是对每一章中用于解决特定漏洞问题的推荐做法的补充。

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：肖晓慧
三河市杨庄长鸣印刷装订厂印刷
2014 年 1 月第 1 版第 1 次印刷
186mm × 240mm • 24.75 印张
标准书号：ISBN 978-7-111-44279-0
定 价：79.00 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 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

译 者 序

随着信息技术与相关产业的发展，人类的工作、学习与生活越来越依赖于计算机软件和网络，但与此同时，安全事件层出不穷，网络攻击造成的损失也与日俱增。就在本书的翻译过程中，国际上发生了“棱镜门”事件。“信息战士”的工作使受害国家的安全都受到了威胁，更不用说窃取普通公民的网上交易信息以及个人信息了。然而，这些信息与人们的生产、生活息息相关，信息的盗用给人们造成了财产和无形的损失。这使人们比以往更加关注安全，逐渐意识到完善网络安全势在必行，各级政府以及各企事业单位在安全方面的投入也在不断提高。本书正是在这样的背景下产生的，它紧贴实际编程活动，深入浅出，帮你解决实际应用中的安全漏洞问题。

我曾经编写过 C 和 C++ 代码，对这两门编程语言都深有感触。C 和 C++ 语言是实现许多操作系统和服务器软件的开发语言，支撑着许多为数以万计的用户提供服务的重要应用程序，必须坚实可靠才能持续不断地正常运行。特别是移动互联网的发展，大大增加了应用程序的种类和数量，不安全的应用程序也会造成安全漏洞迅速地大面积传播。因为 C 和 C++ 语言产生的年代较早并且设计为与底层设备打交道，所以对效率要求很高，这使得这两种语言存在固有的安全问题，虽然语言标准的制定者意识到了这个问题，并提出了许多更加安全的解决方案，但大量的遗留代码中仍然包含不安全的实现和库函数调用，这些都会不时引发安全问题。许多安全检测工具可以帮助人们检查出代码的漏洞，但却不能帮助他们把它们修改正确，而且对 C 语言中某些功能理解上的偏差往往会导致误用。总而言之，编写安全代码完全是开发人员的责任，而开发人员若要具备这方面的技能，就得掌握相关的理论并在实践中自觉加以运用。

本书涵盖安全编码各个方面，人们普遍能意识到指针是 C 和 C++ 中最灵活的功能，也是程序员最容易犯错的地方，为强调其重要性有人曾说过“不会用指针的 C 程序员不是真正的 C 程序员”，这是因为指针直接对内存地址进行读、写访问，它的重要性不言而喻。但与此同时，这两门语言的另一些要点却常常被忽视。然而编写安全代码的从业人员，则需要全面地了解字符串、整数、内存管理、格式化输入与输出、文件操作中的安全陷阱，并学会规避之法。此外，现代应用程序绝大多数具备并发访问能力，因此本书新增了这些方面的内容，使之更符合实际需要。本书每个主题都配有实例，读者能从中了解产生安全漏洞的缘由和正确的做法。本书还总结了安全编码的实施指南，指出软件开发组织应用这些规程可以提高安全软件的开发效率并更能保证产品符合安全要求。本书引用了许多相关著作，以便于读

者进一步了解某个主题的细节。总之，本书值得每个从业人员在开始实际编程活动前阅读。

此外，本书作者 Robert C. Seacord 是业界著名的安全专家，著有多本安全编程著作。他所领导的 CERT 项目定期发布安全报告，并与软件厂商合作解决安全问题，在安全领域硕果累累。本书很多内容引用自实际的安全问题，并结合了国际标准 C11 和 C++11 展开叙述，实用且跟踪了 C 和 C++ 语言的最新发展。因此，我郑重向您推荐此书，希望它能在检测及修复安全漏洞上对您有所帮助！

感谢机械工业出版社谢晓芳编辑邀请我翻译本书并对译文细致地审查和润色。感谢本书第 1 版译者荣耀和罗翼先生做出的贡献，他们的辛勤工作保障了本书的顺利完成。

感谢我的妻子李颖一如既往的支持和儿子卢〇一的鼓励，他们是我工作的动力。

希望这本书能对读者有所帮助。由于译者经验和水平有限，译文中难免有不妥之处，恳请读者批评指正！

卢涛

序

现代社会对网络化软件系统的依赖日益加深，以这些系统为目标的攻击数量亦与日俱增。形形色色的针对政府、公司、教育机构以及个人系统的攻击，已经导致诸多严重的后果：敏感数据丢失或损坏、系统受损、生产效率降低，以及经济损失等。

尽管今天互联网上的攻击行为大多不过是恶作剧，但是越来越多的证据表明，罪犯、恐怖分子以及其他心怀叵测的人已将软件系统中的漏洞视作达成其不可告人目的的手段。近年来，每年发现的软件漏洞已逾 4000 个。这些漏洞的成因包括：在设计和实现方面对如何保护系统考虑不周，在开发实践方面对消除会导致安全缺陷的实现瑕疵关注不够。

伴随着软件漏洞的日益增加，攻击手段也与时俱进，变得越发老练、有效。入侵者在发现软件产品中的漏洞后，可以迅速开发出利用漏洞的脚本，继而使用这些脚本威胁计算机的安全。更糟糕的是，他们还将这些脚本与其他攻击者共享。这些脚本与一些“自动扫描网络以窥探脆弱系统”的程序联合起来，攻击那些脆弱系统，危害其安全，甚至利用它们进一步传播攻击。

由于每年都会发现大量的漏洞，系统管理员为了给既有系统打补丁日益疲于奔命。打补丁有时并非易事，并可能会导致意想不到的副作用。在供应商发布一个安全补丁后，可能需耗时数月乃至数年的时间，90% ~ 95% 的受影响计算机才能完成修补工作。

互联网用户已经严重依赖于这样一种解决方式：互联网社区作为一个整体，对发生的安全攻击快速做出反应，以确保将损害降至最低，并将攻击迅速击溃。然而，显而易见，如今的“反应式解决方案”已经达到其效力的极限。虽然个体响应组织都在为将漏洞处理过程的精简化和自动化而努力工作，但是今天商业软件产品中的漏洞数目，已经多至只有财力雄厚的组织才能应对漏洞修复工作，其他组织实质上已心有余而力不足。

没有任何证据表明大多数产品中的安全问题得到了改善，很多软件开发者尚未很好地理解从各种漏洞成因中获得的教训，或者尚未运用能够起作用的缓解策略，CERT/CC 不断在同一软件产品的更新版本中发现早期版本已存在的同类漏洞就是明证。

这些因素交织在一起，预示着我们完全可以相信，很多攻击甚至会在我们实际希望的最短响应时间内造成巨大的经济损失和服务崩溃。

虽然积极且协调的反应仍然不可或缺，但我们还必须构建出不易被破坏的、更安全的系统。

关于本书

本书致力于解决 C 和 C++ 中已经导致危险的、破坏性的常见软件漏洞的基本编程错误，这些漏洞自 CERT 1988 年创立以来就记录在案。针对导致这些漏洞的编程错误，本书既出色地给出了深度工程分析，又提出了缓解策略，可以富有成效地降低或消除漏洞被恶意利用的风险。

自 Robert 于 1987 年 4 月加入 SEI 以来，我就一直与他共事。Robert 是一位知识渊博、技术熟练的软件工程师，他在软件漏洞细节分析和表达洞察成果方面卓尔不凡。作为研究结果，本书细致而精确地分析了软件开发者面临的常见问题，并提供了实用的解决方案。Robert 在软件开发方面具备的宽广背景，也使得他擅长在性能、易用性以及其他在开发安全代码过程中必须考虑的质量特性之间进行权衡。除了 Robert 外，本书还凝聚了 CERT 积累和提炼的知识，以及 CERT/CC 漏洞分析小组、CERT 操作人员、SEI 编辑和工作人员的杰出工作。

Richard D. Pethia

CERT 主管

前　言

1988 年 11 月爆发的 Morris 蠕虫事件造成当时全球 10% 的互联网系统陷入瘫痪，作为对该事件的回应，当月美国国防部高级研究计划局（Defense Advanced Research Projects Agency，DARPA）成立了 CERT。CERT 位于宾夕法尼亚州匹兹堡市的软件工程研究院（Software Engineering Institute，SEI）内，这是一个由美国国防部发起的研发中心，并受联邦政府资助。

CERT 最初的工作重点是对各种网络事件做出快速响应和分析。这里所说的事件既包括得逞的攻击（如系统受损与拒绝服务等），也包括未得逞的攻击企图、探测和扫描。自 1988 年以来，CERT 共已接到逾 22 665 个报告计算机安全事件或咨询有关信息的热线电话，已处理逾 319 992 起计算机安全事件，而且每年报告的事件数目呈持续增长的态势。

虽然对事件做出及时响应必不可少，但是这还不足以保护互联网和互联的信息系统的安全。分析表明，大部分计算机安全事件是由特洛伊木马、社会工程学（social engineering）以及软件漏洞利用（exploitation）造成的，包括软件缺陷、设计决策、配置决策以及非预期的系统间交互等。CERT 监控漏洞信息的公共来源并且经常性地收到漏洞报告。自 1995 年以来，CERT 已经收到超过 16 726 份漏洞报告。每收到一份报告，CERT 就会分析报告所述的可能的漏洞，并与软件制造者协作，通知其产品中存在安全缺陷，促进并追踪其对问题的响应[⊖]。

和事件报告相似，漏洞报告也以惊人的速度持续增长[⊖]。虽然对漏洞的管理抑制了这一进程的发展，但是对于解决互联网和信息系统的安全问题来说，这同样远远不够。为了解决日益增加的漏洞和事件问题，必须采取相应的措施：在源头有效地控制它们，即必须在软件的开发阶段和随后的维护工作中避免引入软件漏洞。对现有漏洞的分析表明，大部分漏洞都是由少数根本原因导致。本书旨在传授开发者有关这些根本原因的知识，并介绍避免引入漏洞的措施。

本书读者对象

对任何采用 C 和 C++ 开发或维护软件的人来说，本书都是非常有价值的。

⊖ CERT 与超过 1900 名软件、硬件开发者保持互动交流。

⊖ 参见 www.cert.org/stats/cert_stats.html 以获得最新统计信息。

- 如果你是一位 C/C++ 程序员，本书将教你如何识别会导致软件漏洞的常见编程错误，理解这些错误是如何被利用的，并以安全的方式实现解决方案。
- 如果你是一位软件项目经理，本书将教你识别软件漏洞的风险及导致的后果，对投资开发安全软件给出指导。
- 如果你是一位计算机专业的学生，本书将教你有用的编程实践，帮助你避免不良的开发习惯，使你能够在职业生涯中开发出安全的程序。
- 如果你是一位安全分析师，本书对常见的漏洞提供了细致的描述，并提供了检测漏洞的方法和实用的规避措施。

本书组织和内容

本书提供了 C 和 C++ 编程中关于安全实践方面的实用指导。安全的程序需要安全的设计，然而，如果开发者不了解 C 和 C++ 编程中许多固有的安全陷阱，那么即便是最佳设计也可能最终导致不安全的程序。本书对 C 和 C++ 中常见的编程错误提供了详细的解释，并描述了这些错误是如何导致有漏洞的代码的。本书重点讨论 C 和 C++ 编程语言以及相关库固有的安全问题，而不强调和外部系统（如数据库和 Web 服务器等）的交互中牵涉的安全问题。这些方面话题丰富，应单独讨论。希望本书对于涉及开发安全的 C 和 C++ 程序的任何人都有所裨益，而不管具体开发何种应用。

本书围绕着软件工程师经常实现的、可能会导致安全问题的功能（例如，格式化输出和算术运算等）组织内容。每一章都描述了一些会导致漏洞的不安全编程实践和常见的错误、这些编程缺陷如何被利用、这种恶意利用所导致的后果，以及替代的安全做法等。对于软件漏洞的根源问题，例如，缓冲区溢出、整型范围错误和无效的格式字符串等，都有详细的解释。每一章都包含检测既有代码中漏洞的技术，以及如何安全地实现所需功能的措施。

本书包含以下各章。

- 第 1 章：概述问题，介绍安全术语和概念，并对为何 C 和 C++ 程序中存在如此多的漏洞发表看法。
- 第 2 章：描述 C 和 C++ 中的字符串操作、常见的安全缺陷以及由此导致的漏洞（包括缓冲区溢出攻击和栈溢出攻击），还介绍了代码注入（code injection）和弧注入（arc injection）两种漏洞利用方式。
- 第 3 章：介绍任意内存写（arbitrary memory write）漏洞利用方式，它允许攻击者对内存中任意位置的一个地址进行写操作。该章描述这些漏洞利用方式如何用于在受害机器上执行任意的代码。由“任意内存写”导致的漏洞在稍后的章节中讨论。
- 第 4 章：描述动态内存管理，讨论动态分配的缓冲区溢出、写入已释放内存，以及重复释放

（double-free）漏洞。

- 第 5 章：讨论整数安全问题（即与整数操作相关的安全主题），包括整数溢出、符号错误以及截断错误等。
- 第 6 章：描述格式化输出函数的正确和错误的用法。对因这些函数的错误使用所导致的格式字符串和缓冲区溢出漏洞都有讨论。
- 第 7 章：重点介绍并发和可能导致死锁、竞争条件和无效的内存访问序列的漏洞。
- 第 8 章：描述和文件 I/O 相关的常见漏洞，包括竞争条件（race condition）和检查时间与使用时间（Time Of Check, Time Of Use, TOCTOU）漏洞。
- 第 9 章：推荐一些可以整体改善 C/C++ 应用程序安全性的具体开发实践。这些建议是对每一章中用于解决特定漏洞问题的推荐做法的补充。

本书包含数百个安全和不安全的代码示例以及漏洞利用示例。尽管其中有一些涉及与其他语言的比较，但是几乎所有例子都是用 C 和 C++ 编写的。这些例子都在 Windows 和 Linux 操作系统上验证过。具体的示例程序通常已在一个或多个特定环境下编译和测试过，同时漏洞都被予以评估，以确定它们是否具体相关于以下因素，或者具有包含以下因素的普遍性：编译器版本、操作系统、微处理器、适用的 C 或 C++ 标准、小端表示法（即低位在前、高位在后）或大端表示法（即高位在前、低位在后）架构，以及执行栈架构等。

本书以及基于它的在线课程，重点讨论通常导致软件漏洞的常见 C 和 C++ 编程缺陷。尽管如此，限于篇幅，本书并未涵盖漏洞的每一个可能来源。与本书有关的附加与更新信息、活动时间表以及新闻参见 www.cert.org/books/secure-coding/。书中讨论的漏洞也是参照来自 US-CERT 漏洞数据库（参见 www.kb.cert.org/vuls/）的实际例子。

可以通过位于 https://oli.cmu.edu/ 的卡内基 • 梅隆大学开放学习计划（Open Learning Initiative，OLI）。输入课程的密钥 0321822137 来访问与本书配套的安全编码在线课程。

致　谢

在此我要感谢为本书问世做出贡献的每一个人。首先要感谢 Noopur Davis、Chad Dougherty、Doug Gwyn、David Keaton、Fred Long、Nancy Mead、Robert Mead、Gerhard Muenz、Rob Murawski、Daniel Plakosh、Jason Rafail、David Riley、Martin Sebor 和 David Svoboda。他们为本书做出了直接的贡献。还要感谢下列研究人员的贡献。他们是 Omar Alhazmi、Archie Andrews、Matthew Conover、Jeffrey S. Gennari、Oded Horovitz、Poul-Henning Kamp、Doug Lea、Yashwant Malaiya、John Robert 和 Tim Wilson。

我要感谢 SEI 和 CERT 的管理层，感谢他们的鼓励以及对我劳动成果的支持。他们是 Jeffrey Carpenter、Jeffrey Havrilla、Shawn Hernan、Rich Pethia 和 Bill Wilson。

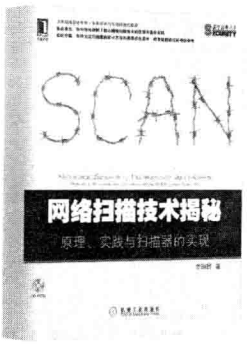
还要感谢编辑 Peter Gordon 和 Addison-Wesley 的其他一些职员，他们是 Jennifer Andrews、Kim Boedigheimer、John Fuller、Eric Garulay、Stephane Nakib、Elizabeth Ryan 和 Barbara Wood。

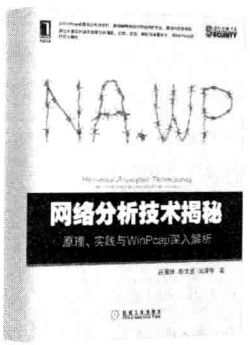
我也想感谢帮助开发开放式学习倡议课程的每个人，包括帮助设计课程的学习科学家 Marsha Lovett，以及帮助实现该课程的人们，包括 Norman Bier 和 Alexandra Drozd。

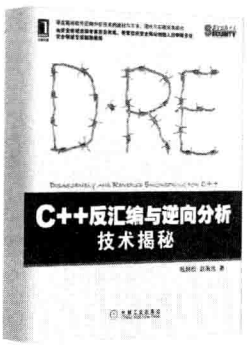
我还要感谢以下审校者给予的深思熟虑的评论和富有洞察力的见解，他们是 Tad Anderson、John Benito、William Bulley、Corey Cohen、Will Dormann、William Fithen、Robin Eric Fredericksen、Michael Howard、Michael Kaelbling、Amit Kalani、John Lambert、Jeffrey Lanza、David LeBlanc、Ken MacInnis、Gary McGraw、Randy Meyers、Philip Miller、Patrick Mueller、Dave Mundie、Craig Partridge、Brad Rubbo、Tim Shimeall、Michael Wang 和 Katie Washok。

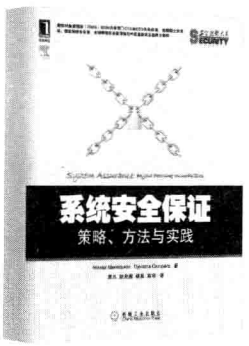
我要感谢 CERT 小组其他成员的支持和协助，没有他们我永远都不可能完成本书。最后一点也很重要，感谢内部编辑和图书管理员，他们的帮助使得本书的最终问世成为可能，他们是 Rachel Callison、Pamela Curtis、Len Estrin、Eric Hayes、Carol J. Lallier、Karen Riley、Sheila Rosenthal、Pennie Walters 和 Barbara White。

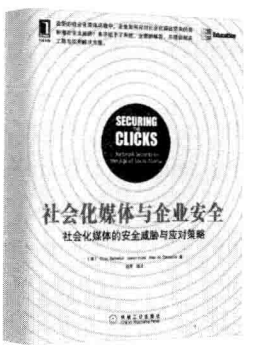
推荐阅读

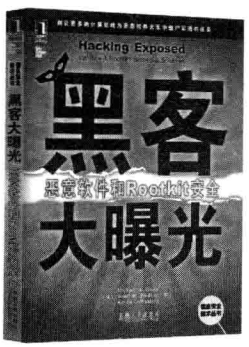

SCAN
网络扫描技术揭秘
原理、实践与扫描器的实现

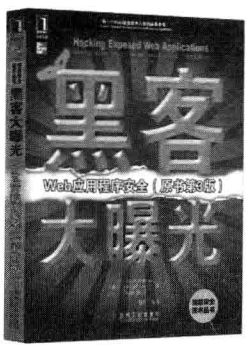

NA.WP
网络分析技术揭秘
原理、实践与WinPcap深入解析

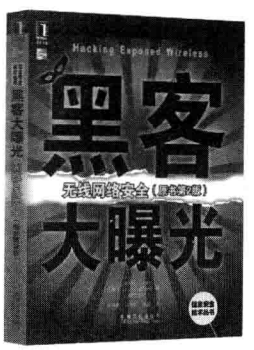

D·RE
C++反汇编与逆向分析
技术揭秘


系统安全保证
策略、方法与实践

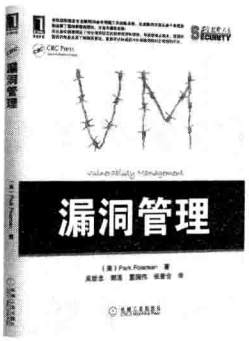

SECURING
CLICKS
社会化媒体与企业安全
社会化媒体的安全威胁与应对策略


Hacking Exposed
黑客大曝光
恶意软件和Rootkit安全

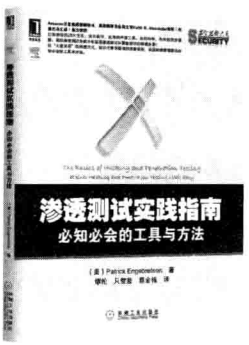

Hacking Exposed Web Applications
黑客大曝光
Web应用程序安全（原书第3版）

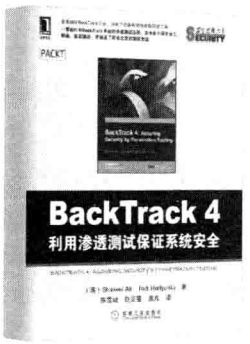

Hacking Exposed Wireless
黑客大曝光
无线网络安全（原书第2版）

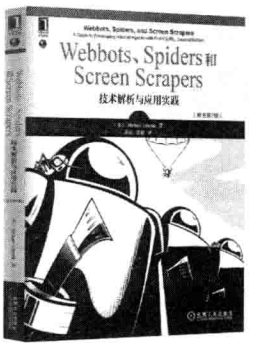
推荐阅读

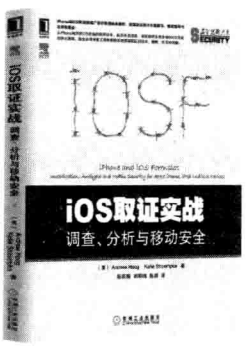

漏洞管理

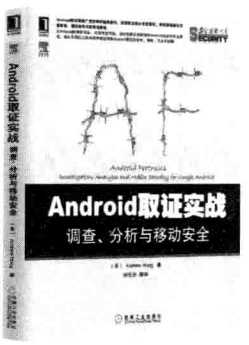

内核漏洞的
利用与防范


渗透测试实践指南
必知必会的工具与方法


BackTrack 4
利用渗透测试保证系统安全

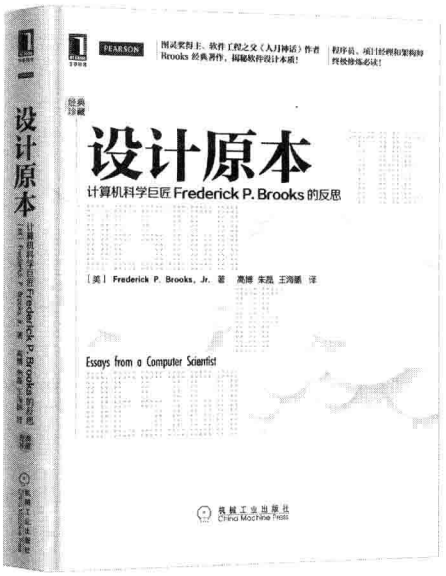

Webbots、Spiders和
Screen Scrapers
技术解析与应用实践


iOS取证实战
调查、分析与移动安全


Android取证实战
调查、分析与移动安全


Android安全机制解析
与应用实践

推荐阅读



设计原本（精装本）

如果说《人月神话》是近40年来所有软件开发工程师和项目经理们必读的一本书，那么本书将会是未来数十年内从事软件行业的程序员、项目经理和架构师必读的一本书。它是《人月神话》作者、著名计算机科学家、软件工程教父、美国两院院士、图灵奖和IEEE计算机先驱奖得主Brooks在计算机软硬件架构与设计、建筑和组织机构的架构与设计等领域毕生经验的结晶，是计算机图书领域的又一史诗级著作。

领域特定语言

本书是DSL领域的丰碑之作，由世界级软件开发大师和软件开发“教父”Martin Fowler历时多年写作而成。全面详尽地讲解了各种DSL及其构造方式，揭示了与编程语言无关的通用原则和模式，阐释了如何通过DSL有效提高开发人员的生产力以及增进与领域专家的有效沟通，能为开发人员选择和使用DSL提供有效的决策依据和指导方法。

目　录