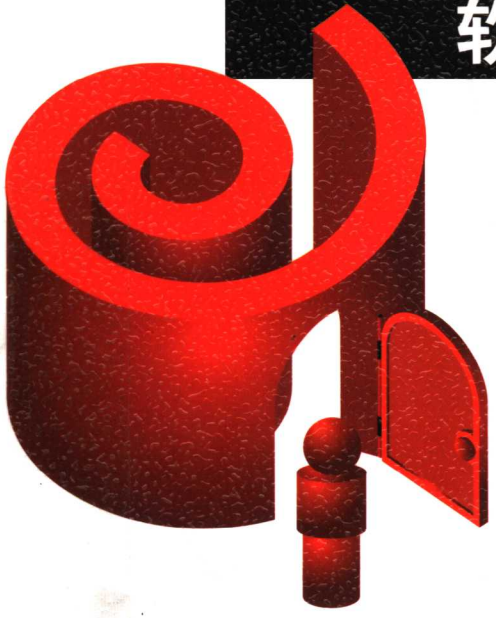
Broadview
www.broadview.com.cn

安全技术大系

Software Security：Building Security In

软件安全
——使安全成为软件开发必需的部分

[美] Gary McGraw 著
周长发 马颖华 译



- 软件安全领域的**顶级权威Gary McGraw**又一力作!
- 7种软件**安全接触点**改变软件开发的方式，让安全成为软件开发的“核心DNA”！

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

安全技术大系

软件安全——使安全成为软件开发必需的部分

[美] Gary McGraw 著
周长发 马颖华 译

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

内 容 简 介

本书是由软件安全领域的权威专家编著，讲授如何实施软件安全的专著。本书在论述软件安全理论的基础上详细讲解了如何将软件安全付诸实践。书中描述的软件安全最优方法（或者称为接触点）以优秀的软件工程方法为基础，并且在整个软件开发生命周期中都明确地仔细考量安全问题，即认识和理解普通的风险（包括实现缺陷和体系结构瑕疵）、基于安全进行设计，以及对所有的软件工件都进行彻底、客观的风险分析和测试。本书的目的是使接触点方法为你所用。采用本书的方法并不会从根本上改变你的工作方式，但是能够改善现有的软件开发生命周期，并能据此来创建自己的安全的开发生命周期。本书还介绍了知识管理、培训与认知，以及企业级的软件安全计划等方面的内容。

本书适合与软件相关的任何机构的管理人员、商业人员、软件架构人员、软件开发人员、软件测试人员以及安全管理人员阅读，可以作为大学、研究机构和培训机构的计算机安全和软件安全课程的教材和参考书。

图书在版编目（CIP）数据

软件安全：使安全成为软件开发必需的部分 /（美）麦克劳（McGraw,G.）著；周长发，马颖华译. —北京：电子工业出版社，2008.4

（安全技术大系）

书名原文：Software Security：Building Security In，1/e

ISBN 978-7-121-05889-9

Ⅰ. 软… Ⅱ. ①麦… ②周… ③马… Ⅲ. 软件开发－安全技术 Ⅳ. TP311.52

中国版本图书馆 CIP 数据核字（2008）第 011668 号

责任编辑：许 艳

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：787×980 1/16 印张：23 字数：484 千字

印 次：2008 年 4 月第 1 次印刷

定 价：49.80 元

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。

译 者 序

译者有幸参加了2006年3月在美国加州Santa Clara举行的SD West开发大会。会议第三天的主题演讲（keynote）是著名的安全权威Bruce Schneier的题为“What Works, What Doesn't, and Why”的演讲。但是，演讲开始时出现在讲台上的并不是大胡子的Bruce Schneier，而是一位胖胖的先生。他风趣地说，大家好，我是Bruce Schneier，引得全场大笑。这位先生做了大约20分钟的关于“软件安全”的介绍，博得全场热烈的掌声。接着，Bruce Schneier上台，他说，大家好，我是Gary McGraw，又是一次全场大笑。然后，Bruce Schneier说，感谢Gary McGraw关于软件安全的精彩演讲，他的演讲主题与此密切相关。这是译者第一次见到软件安全领域的权威Gary McGraw博士。Gary McGraw博士在演讲中阐述了解决软件安全问题的“使安全成为软件开发必需的部分”（Building Security In）的观点和名为“接触点”的一套最优方法。他的演讲给我留下了深刻的印象。因此，我修改了我的听课日程安排，选听了Gary McGraw博士的题为“Building Security In”的技术课程。正是从这个课程中，我了解到了Gary McGraw博士的新书“Software Security: Building Security In”的主要内容。我认为这本书是至今为止关于实施和保障软件安全的最全面、最实用的书籍，如果能在国内出版，肯定会给我国的软件安全带来极大的促进作用。于是，我将这本书推荐给了电子工业出版社的郭立女士。郭立女士慧眼识金，迅速与这本书的出版社Addison-Wesley Professional联系并获得了本书的中文版权，这是本书中文版得以迅速出版的基础。

SD West开发大会是美国最负盛名的软件开发技术大会之一。本次大会中，至少有四分之一的技术讲座以软件安全为主题。这有力地说明了美国软件业界目前对软件安全的重视程度。那么，什么是软件安全呢？Gary McGraw认为，软件安全就是使软件在受到恶意攻击的情形下依然能够继续正确运行的工程化软件思想。译者认为需要从如下的四个方面来理解这个定义。

首先，软件安全是计算机安全的一个分支。现代社会中，我们生活所需的一切似乎都离不开计算机系统，我们的电力、供水、交通、通讯、金融等等，都依赖于计算机系统的安全运行。但是，计算机系统并不安全，它潜伏着严重的不安全性、脆弱性和危险性。如何保障计算机系统的安全就成为我们这个时代的一个根本问题，计算机安全这门

学科也因此应运而生，并成为近二十年来最热门的学科之一。计算机安全的研究范畴包括硬件安全、软件安全、数据安全、运行安全和网络安全。

其次，软件安全是计算机安全的关键。由于病毒主要通过网络传播，而黑客主要通过网络来进行攻击，因此，多年来人们一直认为网络安全是计算机安全的主要问题。但是在网络安全上的巨大投入却没有从根本上解决计算机安全问题。经过多年的研究，人们逐步认识到软件是计算机安全的大问题：软件的不稳定导致系统崩溃和数据丢失、病毒攻击的是软件的缺陷、黑客利用的是软件的弱点、机密和隐私的泄漏是因为软件存在漏洞……总之，危害计算机安全的绝大部分因素都与软件相关。正如 Gary McGraw 博士所指出的，计算机安全中的首要和关键问题是软件问题。

第三，软件安全问题的根源是软件存在弱点，因此只有改变我们建造软件的方式，才能从根本上保障软件安全。软件安全的概念已经出现了将近十年，但是人们对软件安全的正确认识才刚刚建立起来。提到软件安全，大多数人，包括许多的软件人员，首先想到的都是反病毒程序和防火墙之类的保护程序，或者密码学之类的信息加密技术。到现在为止，我们在软件安全方面的主要精力依然花费在这些方面。但是，人们所津津乐道的这些东西能从根本上解决软件安全问题吗？回答是否定的。有大量的统计数据可以证明，我们现在所广泛采用的方法并不能解决软件安全问题。我用一个例子来说明这个问题。假设有一个程序，其密码验证是用加密算法来实现的，它安装在一个被防火墙保护的系统中，系统中安装了反病毒程序。这个程序仍然不是绝对安全的（其实，许多受到病毒感染或者黑客攻击的程序都具有这种保护）。首先，我们不知道这个程序是否健壮。如果系统中存在一个可能导致缓冲区溢出之类的弱点，一旦黑客发现了这个弱点，并利用它来使系统出现缓冲区溢出，就可能导致缓冲区或其相邻内存单元中的数据的泄漏或者改写，在最糟糕的情形中系统甚至会执行黑客传给它的任何代码！这样的话，无论是防火墙还是加密系统都不能阻止黑客的攻击，因为黑客是用“合法的”手段“诱使”系统被其利用的。其次，没有谁能保证防火墙和反病毒软件是完全可靠的，它们也可能存在缓冲区溢出之类的弱点。实际上，黑客现在正在攻击防火墙和反病毒程序之类的保护程序，完全可能通过他们内部存在的这类弱点来攻击我们的系统。我们已经有了许多的防火墙、反病毒软件之类的保护程序，但是，病毒和黑客依旧猖獗——只要打开任何一份 IT 报纸或者杂志都能找到这类报道。问题的根源是什么呢？Gary McGraw 博士指出，问题的根源就是我们所依赖的软件存在太多的安全弱点，而不断增加的软件复杂性和可扩展性更是火上浇油般地助长了这种情形。因此，解决软件安全问题的根本方法就是改善我们建造软件的方式，以建造健壮的软件，使其在遭受恶意攻击时依然能够安全可靠和正确运行。

最后，我们需要用工程化的方法来实施软件安全。Gary McGraw 博士在本书中全面、

详细地介绍了这种方法。本书的副标题“使安全成为软件开发必需的部分”点出了实施软件安全的工程化方法的总纲，即在整个软件开发生命周期中都要确保将安全作为软件的一个有机组成部分。支持这个总纲的三根支柱是应用风险管理、软件安全的接触点和知识。风险管理是一种战略性方法，即将追踪和减轻风险作为一种贯穿整个生命周期的指导方针。接触点，即在软件开发生命周期中保障软件安全的一套最优方法，是一种战术性方法。Gary McGraw 博士总结出了七个接触点，即代码审核、体系结构风险分析、渗透测试、基于风险的安全测试、滥用案例、安全需求和安全操作。无论你采用什么样的软件开发方法学，你都可以将这些接触点应用到你的开发生命周期中，而不需要完全改变你的软件开发生命周期。这些接触点从“黑帽子”（攻击和破解）和“白帽子”（防御和保护）两个方面综合地考察软件开发中可能出现的问题，结合了它们的开发生命周期就成为“安全的”开发生命周期。“安全的”开发生命周期能够在每一个开发阶段上尽可能地避免和消除漏洞，同时又保留了你熟悉的工作方式。软件安全的第三根支柱是知识，包括收集、压缩和共享能用于为软件安全方法提供坚实基础的安全知识。由于软件安全是一门新的学科，及时总结知识，并用知识来教育所有相关的人员，对确保软件安全是至关重要的。在整个开发生命周期中综合应用这些方法，就能从设计、编码和测试等各个层面上消除软件中的安全弱点，从制度上、方法上最大限度地保障软件安全。

本书是关于软件安全的最新、最全面和最实用的权威著作。作者提出的“使安全成为软件开发必需的部分（Build Security In，BSI）”的理念已经得到业界和政府机构的广泛认同。美国国土安全部下属的国家网络安全处（NCSD）特别建立了一个关于 BSI 的门户网站（http://buildsecurityin.us-cert.gov/portal/），并与美国国家标准与技术研究所（NIST）、国际标准化组织（ISO）以及电气电子工程师协会（IEEE）一起共同维护这个网站。

本书不仅从理论上阐述了软件安全问题的根源，更重要的是指出了一种解决软件安全问题的实用方法。正如美国国家科学基金会 ACCURATE 中心主任约翰霍普金斯大学教授 Avi Rubin 所指出的，“如果你的生意依赖软件（又有谁的生意不依赖软件呢），就应该买这本书，并将它张贴在餐厅的前面上。” 软件安全人人有责，与软件相关的任何机构的管理人员、商业人员、软件架构人员、软件开发人员、软件测试人员以及安全管理人员都应该具备软件安全的意识和知识，而阅读本书就是使他们快速装备自己的最好方法。现在越来越多的大学、研究机构和培训机构都设置了计算机安全和软件安全课程，本书也是这类课程的理想教材和参考书。

翻译这样一本里程碑式的名著，对译者是一种荣幸，也是一种挑战。作者的语言非常生动和生活化，为了帮助读者阅读，译者对一些较难理解的语句加了一些注释，在书中标记为“译注”，而原书中的注释标记为“原注”。译者虽尽力而为，但是由于水平所

限，翻译中肯定存在不准确甚至错误之处，恳请读者方家批评指正。

本书中文版得以出版，需要感谢许多的人。感谢郭立女士对译者的信任。感谢责任编辑许艳小姐，她认真地审阅了原稿，指出了原稿中许多的错误、细心地改正了许多译者“脑是笔非”的笔误。感谢电子工业出版社博文视点资讯有限公司所有为本书中文版的出版付出努力的人。

周长发

2007 年 1 月

周长发，北京大学理学博士，主要研究领域为图像处理、多媒体技术、软件设计与架构以及计算机安全。现在美国硅谷一家软件公司工作。编写了《多媒体计算机技术开发与应用》、《精通 Visual C++图像处理编程》、《科学与工程数值计算算法集》和《C#面向对象编程》等十多本书籍，翻译了《计算机图形学几何工具算法详解》、《黑客调试技术揭秘》和《软件安全》三本专著。

关于作者

Gary McGraw，博士，Cigital 公司的首席技术官和董事会成员。他也是软件安全领域的世界级权威，与人合著了 5 部最畅销的安全方面的著作：与 rootkit.com 的 Greg Hoglund 合著《利用软件的弱点》（Exploiting Software；Addison-Wesley 出版社，2004）；与 John Viega 合著《建造安全的软件》（Building Secure Software；Addison-Wesley 出版社，2001）；与普林斯顿大学的 Ed Felten 教授合著《Java 的安全性：有害的小程序、漏洞和解决方法》（Java Security: Hostile Applets，Holes，and Antidotes；Wiley 出版社，1996）和《保障 Java 的安全：开始认真考虑移动代码》（Securing Java: Getting Down to Business with Mobile Code；Wiley 出版社，1999），以及与 Cigital 公司的共同创始人 Jeffrey Voas 博士合著的《软件缺点注射：给程序打预防针以防范错误》（Software Fault Injection: Inoculating Programs against Errors；Wiley 出版社，1998）。McGraw 博士经常在知名的商业出版物上发表文章，他的观点也经常被全国性的媒体引用。目前他在 IT Architect 杂志上撰写一个关于安全的专栏月刊，同时也是 IEEE Security & Privacy 杂志编辑部的成员。

McGraw 博士与 Cigital Professional Services 和 Cigital Labs 一起提出软件质量管理技术策略，并指导 Cigital 的技术转让过程。他的目标是将尖端科学应用于现实世界中的实际领域，并将先进技术转化为实际领域中的具体应用。除了充当一些大型商业软件供应商的顾问之外，他还创建了 Cigital 的软件安全小组（Software Security Group），并担任 Cigital 公司技术咨询部（Cigital Corporate Technology Council）的主任。

McGraw 博士是一位研究型科学家，一直坚持不懈地进行软件安全领域的研究。他发表了 90 多篇经过专家评审的技术论文，他也是获得美国空军研究实验室（Air Force Research Labs）、DARPA、美国国家科学基金会和美国国家标准研究所（NIST）的先进技术计划资助的项目负责人。McGraw 是印地安那大学的认知科学和计算机科学的双博士，在那里曾师从 Doug Hofstadter 教授，他从弗吉尼亚大学获得哲学学士学位。

McGraw 博士是 Authentica、Counterpane 和 Fortify Software 等公司的技术顾问委员会的成员。他是加州大学戴维斯分校计算机科学系和弗吉尼亚大学计算机科学系的顾问，也是印地安那大学信息学院顾问委员会的主任。他是 IEEE Security & Privacy 杂志特别编委会的成员，最近刚刚当选为国际电子电气工程师协会计算机学会（IEEE Computer Society）理事会的理事。

序

批评软件很容易，编写软件却很难。软件越大，就越是如此。这就像说话一样——言多必失。你说得越多，听众就越容易发现批评的内容，而且他们产生误解的可能性也就越大。简短可能是智慧的灵魂，但是智慧却肯定是简短的灵魂。

而事实上，我们的软件的确非常的絮叨可厌、代码混乱、设计不当、冗长乏味并且未经仔细推敲。软件就像我们的语言一样很容易被曲解，这令人惊讶吗？我们的软件，就像我们的语言一样，会“被骗子歪曲成制造陷阱的工具（twisted by knaves to make a trap for fools[①]）”，这令人惊讶吗？不，这并不令人惊讶，但是，就像依赖语言一样，我们现在凡事都依赖软件。软件是如此重要，没有软件，甚至世界上的人口可能不会像现在这么多——软件用于交通管理，用于商业金融交易，用于记录和存储信息，用于翻译，用于运输，用于电源变压。换句话说，这些无可争辩的证据说明，我们必须让软件正确运行，同样地，无可争辩的证据也说明，让软件正确运行的目标现在不可能、将来也不可能会自然地实现。

McGraw 博士提醒我们，破坏一种东西比设计一种不能被破坏的东西要容易得多。我个人更喜欢 Sam Rayburn 的朴实说法，即“任何一匹驴子都能踢翻谷仓，但是只有好木匠才能建造谷仓。”安全的软件的精确定义是，能够抵御有知觉的对手的软件。这正是人们特别关注安全的软件的原因。以质位变换命题的形式来解析这个定义：如果一个产品不存在有知觉的对手，那么它就没有安全需求。检查这一命题的最佳方法就是研究产品失败的原因——如果你的产品因为一些愚蠢的用户（“嘿，看看这个问题！”）、阿尔法粒子或者没有充电的电池而不能正常运行，那么它就不存在安全问题。假如一个快乐的傻瓜发现，在你的产品的某些输入域中输入 5000 个小写的 a 之后，他就能够成为超级用户，并且因此而导致你的产品不能正常运行，那么，即使这个傻瓜可能不具备完全的知觉能力，但是你的产品仍然存在安全需求。

这里并不存在一条泾渭分明的界线，而只有相对的差别。根据定义，在设计安全的

① 译注：本句出自英国著名文学家 Rudyard Kipling（1865-1936）的诗《如果》（If），原句为“If you can bear to hear the truth you've spoken twisted by knaves to make a trap for fools”。

软件时应该考虑到软件可能出现的问题。即使对手全身心地期待这样的问题出现，安全的软件也能够避免。在设计安全的软件时，应充分研究失败案例与成功案例，甚至研究失败案例还应更多。安全的软件的设计者和实现者都会预想存在一个具有思考能力的对手。

正如 McGraw 博士在这本书中反复强调的，只有专心致志地对待安全问题，才能烘焙出安全的产品。小时候每当我用“我不是有意这样做的，爸爸。”作为做错事的借口时，父亲总会斥责我。他总是尖锐地反问，“可是，你是有意不这样做的吗？”他这样说是为了使我成为一个更完美的人。由于工作原因，我每天都阅读关于弱点的报告。这些报告中的每一个都说，“我不是有意这样做的，爸爸。”有时，他们甚至会说，“我没有这样做，要是我做了，也不是有意的。无论如何，你没有注意到这一点，因此，如果你不希望即将发生的事故成为你的过错，就必须安装这个小小的补丁。难道我不是一个好人吗？”我真想大声质问，“你是有意不这样做的吗？”但是，最诚实的回答也不过是“我曾想有意不这样做。”

安全专家太少，我们总是听不到足够多的意见。优秀的安全专业人才很难找到，而且对他们的需求的增长速度要快于提供的速度。当亟需安全专业人才却找不到时，你该怎么办呢？你可以将所缺乏的专业知识转化为一种处理方法，让其他的人都遵循这种方法，但是这种方法必须具备如下的特征：能够加强严谨的思维方式，避免让不怀好意的人有机可乘，并且可以充分地衡量这种方法以了解它的有效性。如果不全部采用完整的方法，只采用其中的一部分也能够从中受益，那这种方法就更好。当然，最好采用完整的方法，但是，使用任何方法所得的回报都是逐渐减少的，基于这种限制，部分努力就能得到部分价值是一件好事。McGraw 博士认为他自己并不完全是设计这种方法的人，但是他所做的工作与我上面所要求的完全一致。

一个好主意，就是你一听到它就会说，“对，就是这样。”在本书中，你将发现许多这样令人交口称赞的好主意——你会情不自禁地说，“对，就是这样。”例如，代码审核是可以训练并改善软件安全的最有力的工具。再如，如果不认真地研究软件在故意的滥用时可能出现的问题，就不可能知道软件在经受挑战时需要进行多少战斗。当然，只有使用这种方法，才能得到它所带来的益处。缓冲区溢出一直是最常见的攻击手段，而且多年前我们就已经知道应该如何避免它，因此，仅仅知道如何做显然是不够的。

你可能会问，“凭什么说 Cigital 的方法比 XYZ 的方法好呢？”对此，有一种非常自然的反应：这个问题值得探讨。使用现有方法仅仅得到了很有限的效果，其中肯定出了某些问题。这些“问题”要么是因为缺乏技术，要么是因为缺乏训练。如果是缺乏技术，专家们就有责任与其他人共享有效的经验。可能存在许多可行的方法，但是本书证明至少有一种方法是有效的。本书出版后，今后继续出现的任何失败的原因都必须被归结为

缺乏训练。我们很快就能知道足够多的信息。

如果读者偏好用数据说明问题（即使在一篇序中），这里就有三个数据：每 4 个小时就会出现一种新的 Windows 病毒；可能有 15%的台式机都运行着或多或少地存在问题的软件；嵌入式系统的数量要超过台式机一到两个数量级，而且它们几乎是不可升级的。本书存在的目的就是为了证明其中的内容是有用的。

我所做的研究使我相信，采用最好的软件安全方法的公司和采用最差的软件安全方法的公司之间的差距正在不断增加；我推测每 12 个月这种不对称性（用最好和最差的缺陷密度之比来表示）就会增加一倍。如果你像我和 McGraw 博士一样相信，安全是可靠性的子集，那么只需要借用有效的微积分方法就能得出这样的结论：一个包含 5 个系统组件的商业应用程序，每一个组件有 98%的正常运行时间，那么可以预期这个程序每天将有 2.5 个小时不能正常运行。

安全与软件之间的关系就像突变与自然选择一样，但是它们之间还是有一个极端重要的区别：你可以借助软件安全来控制你的生存优势。如果这种说法对你有吸引力，你就应该使用（至少）一些 McGraw/Cigital 的程序。这样做不轻松，也不好玩，但是，正如美军突击队手册所说的：

渴望舒适和态度消极是对生存危害最大的两种常见因素。

现在该是你采取行动的时候了。

Dan Geer
2005 年 9 月 17 日
于麻省剑桥市

权威专家对《软件安全》的赞誉

“二十多年来我一直试图解决安全问题——开始时采用独立的桌面系统，随着安全成为一个普遍的问题，后来过渡到网络安全系统。我曾经自己创办过公司，做过企业的高级管理人员，现在是一名投资者和公司创建者。在这些职位中，我的主要工作都是设法解决这类重要问题。在这二十多年的工作中，我认识到我们在减缓安全问题的蔓延速度方面已经做得不错，但是在解决安全问题方面，从开始到现在，我们并没有取得多大的进展。”

“我们已经花费了 20 年的投资，期望达到的目标就是‘将坏蛋挡在外面’。我们曾经想在我们公司的四周建造一堵很高很厚的围墙，让怀有恶意的人不能进入。在今天的世界中，这种方法根本行不通。我们生活在一个没有围墙的经济体系中，公司必须允许在企业的内部以及与企业的外部世界自由地进行交流。能够自由地访问信息以及自由地使用应用程序是保持竞争力的主要动力。也就是说，战场已经发生变化。因此，我们也必须相应地改变用于保护我们的资产的武器和策略。”

“我认为，解决安全这个复杂难题的唯一方法就是正视这个问题，而不是仅仅研究问题的症状表现。我们需要从一开始就在设计中考虑安全，把安全解决方案融入软件架构之中。任何没有进行过安全设计和安全弱点检查的应用程序、操作系统和中间件都不得发行。只有这样，我们才能够使用正确的武器和策略来开始战斗，并赢得胜利。”

“我非常相信这些观点，因此，我组建了一个名为 Fortify Software 的公司来开发、宣传和销售用于解决攻击问题的方案，并且直接地解决这类问题。我们必须主动出击，而不是被动挨打，必须在问题的萌芽阶段就解决它们。”

“Gary McGraw 是软件安全之父。我们在 Fortify 所做的许多工作都是以 Gary 的研究为基础的。它的新书无疑是实施软件安全方面的权威。你绝对应该一睹为快。”

——Ted Schlein

Kleiner Perkins Caufield & Byers 公司任职股东

“McGraw 正领导着软件安全方向的变革。他的建议简单可行。如果你的业务依赖软件（又有谁的生意不依赖软件呢？），就应该买这本书，并把它张贴在餐厅的墙上。

用书中介绍的七种软件安全接触点方法来改变你建造软件的方式。这样，我或许可以小憩一会儿，不用担心随时可能产生的安全问题了。”

——Avi Rubin
美国国家科学基金会 ACCURATE 中心主任
约翰霍普金斯大学教授
《防火墙与 Internet 安全》（Firewalls and Internet Security）的合著者

“我讨厌充满了愚蠢的安全漏洞的软件。如果你要编写一款我将来可能会使用的软件，那你就需要阅读并理解这本书。”

“Gary 的书告诉了我们早就应该知道的知识：在你开发软件时，最好使安全成为必需的组成部分，而且，他还说明了如何使安全成为必需的组成部分。”

——Marcus J. Ranum
防火墙的发明人
Tenable Security 公司首席科学家

“Gary McGraw 的书说明了如何将开发和测试结合在一起，以改善软件的质量。在这个过程中，他展示了一种非常适合开发人员、测试人员和管理人员采用的框架。McGraw 博士在他早先的著作中充分地展现了他的知识和经验，这本书继续保持了他的一贯风格，将改善软件安全技术的现状。”

——Matt Bishop
加州大学戴维斯分校计算机科学系教授
《计算机安全》（Computer Security）的作者

“保障（assurance）和评估（assessment）的方法学是所有现代工程实践的基本要素。虽然开发安全的软件是一门工程学科，但是其中却缺少严格的保障和评估方法。Gary McGraw 的《软件安全》对于这一领域发展是一个里程碑式的著述。使用书中的方法，读者不但能够完成工作，而且还能完满地完成工作。”

——George Cybenko
达特茅斯工程学院（Dartmouth College）Dorothy 和 Walter Gramm 教授

“对于软件安全来说，最难缠的就是实现细节。本书解决了其中的细节问题。”

——Bruce Schneier
Counterpane 公司的 CTO 和创建人
《超越恐惧》（Beyond Fear）和《秘密与谎言》（Secrets and Lies）的作者

“大多数人都没有综合地考量安全问题。我们必须承认，大多数人在大部分时间里根本就不考虑安全问题，其中包括软件开发人员。因此，当他们遇到糟糕的事情，比如病毒破坏了他们的磁盘时，他们的反应（与大多数人的第一反应一样）就是安装防火墙和反病毒产品。但是，防火墙和反病毒产品并不是最合适的解决方法。”

“Gary McGraw 在本书中综合地考虑了软件安全问题，并且说明了健壮和安全的软件需要事前充分的考虑和计划。这种说法合情合理，但是人们却经常不能认识到这一点。更重要的是，Gary 还描述了如何实施这种方法。当务之急就是要求所有的软件开发人员都阅读这本书。”

——Greg Rose
Qualcomm 公司主管产品安全的副总裁

“在这本最新的著作中，McGraw 继续为我们提供了权威人士的观点，说明了开发软件的公司的需求正在发生变化。要保证软件的质量和安全，以及理解它们的内涵，就必须研究和理解业务，而软件的质量和安全又决定了满足这些业务需求的技术方案。传统的方法强调通过开发方法和过程来改善软件质量，McGraw 超越了这种方法，他采用了一种更全面的观点，即着重考虑如何以系统和服务的运行为中心地将软件组件集成在一起。如果你的业务依赖于软件，你就应该阅读这本书。”

——Ron Moritz
Computer Associates 公司高级副总裁和首席安全策略专家

“根据摩尔定律，可以集成到芯片中每平方毫米内的晶体管数量每 18 个月就会增加一倍。其结果是，微处理器运行得越来越快。RAM 芯片的容量变得越来越大。硬件能力的这种指数级的增长相应地不断促使软件复杂度的增长。”

“这种增长带来了一种灾祸：无意中造成的交互和安全瑕疵。对于包括我自己在内的每一个从事数据安全工作的人来说，我们的主要挑战就是找出一种性价比合理的方法，能够以最低的风险交付最多的功能。过分地瞻前顾后会使公司或者开发队伍丧失效率。同时，灾难是很常见的；我们公司的客户就因为可以克服的软件缺点而损失了数十亿美元。实现恰当的平衡并不容易。”

“密码学（我的专业领域）经常被人寄予厚望，看成一种可能的救星。粗略地看，这似乎很有道理：现代加密算法所提供的数学强度远远超出了任何攻击者现在（或者可能永远）所能破解的程度。令人遗憾的是，这基本是一种幻想——密码系统的强度与作为其基础的代码实现的强度是一致的。我为设计 SSL 3.0 所做的工作就非常明确地说明了这一点。即使协议本身被认为是可靠的，当运行在被间谍软件感染的计算机中含有代码缺陷的浏览器在与被危害的 Web 服务器对话时，即使它显示了一个“锁形”的图标，也没有什么意义。换句话说，不论在建造系统时使用了什么工具，建造健壮软件的能力仍将限制软件的安全性。”

“显然，有些方法不能奏效。工程技术能够有效地实现功能，但是当它们被误用于解决安全问题时，经常会导致最糟糕的问题。传统的软件开发是一种编写代码的迭代周期，在周期内不断地找出和改正问题。其结果是一种进化的过程，在其中实现想要的功能并删除可见的缺陷。令人遗憾的是，对于常规的测试来说，大多数的安全瑕疵都是不可见的。由此导致的结果是，即使系统存在问题，许多工程师的直觉仍然认为系统运行正常。”

“总的来说，解决软件安全问题是一件说起来容易做起来难的事情。你不可能找到（也不存在）任何的魔术子弹，但是本书给出了我所见过的最清晰的处理复杂问题的策略之一。”

——Paul Kocher
Cryptography Research 公司总裁和首席科学家

“软件安全是一个连续的过程，首先需要理解其中的问题。要实现有效的软件安全，还必须将这种理解和知识结合到包括设计、编码、测试和配置在内的软件开发生命周期的各个阶段中。几年前，我帮助建造了一个名为 NtSpectre，用于 Windows NT 的安全分析工具。我们建造这个工具的目的是为了分析服务器的安全设置，这个服务器是为一个付费的在线游戏而设计的。游戏的内容其实很简单，但是我们的工具拥有了一群挑剔而忠实的用户，而且我对安全层及其复杂性的理解也有了显著的增强。从这次开发经历中我学到了软件开发和软件安全的一种重要的哲学和实践方法，特别是测试方法，即既不要假设，也不要猜测。”

“本书恰如其分地描述了软件安全，并将它结合到你的软件开发过程中。无论你采用的方法是敏捷（agile）、极限（extreme）、瑞利（rational），或者可能对是否采用瀑布（waterfall）而犹豫不决，本书都能指导你在所采用的方法中考虑安全。McGraw 博士具体地描述了他在这个领域内所遇到的真实而相当普遍的安全弱点，而没有过多地纠缠理论和抽象概念。在此基础上，他继续说明了安全问题在本质上是与开发过程的漏洞相关

的，并且巧妙地指导你对开发过程进行改进。”

——Erik Hatcher
eHatcher Solutions 公司开发人员
《Lucene in Action》的合著者

“能够长期有效地解决信息安全问题的最重要的方法之一，就是使安全成为软件开发的‘核心 DNA’的一部分。McGraw 的书告诉我们如何使‘安全文化’成为你的软件开发生命周期的一部分。”

——Howard A. Schmidt
R & H Security Consulting LLC 公司总裁和 CEO
前白宫计算机安全顾问