计 算 机 科 学 丛 书

HZ BOOKS 华章教育

CRC Press Taylor & Francis Group

软件安全

从源头开始

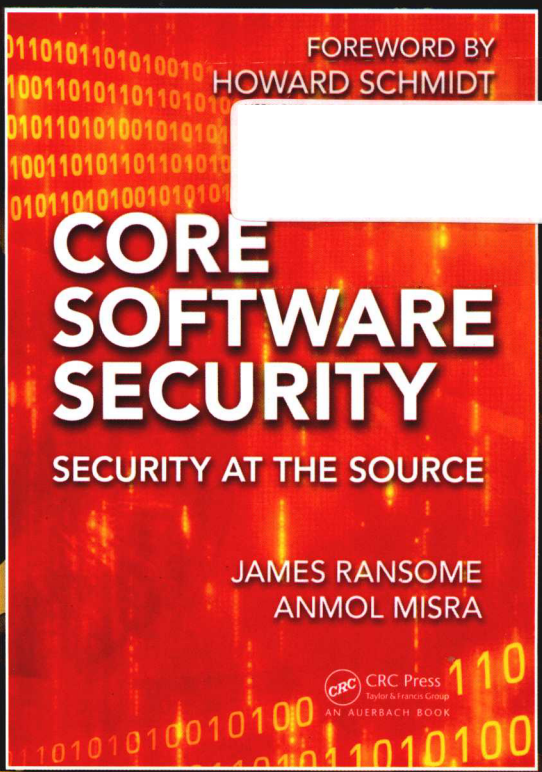
[美] 詹姆斯·兰萨姆（James Ransome）
安莫尔·米斯拉（Anmol Misra） 著

丁丽萍 卢国庆 李彦峰 穆海蓉 曹原野 译

宋宇宇 审校

Core Software Security

Security at the Source



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

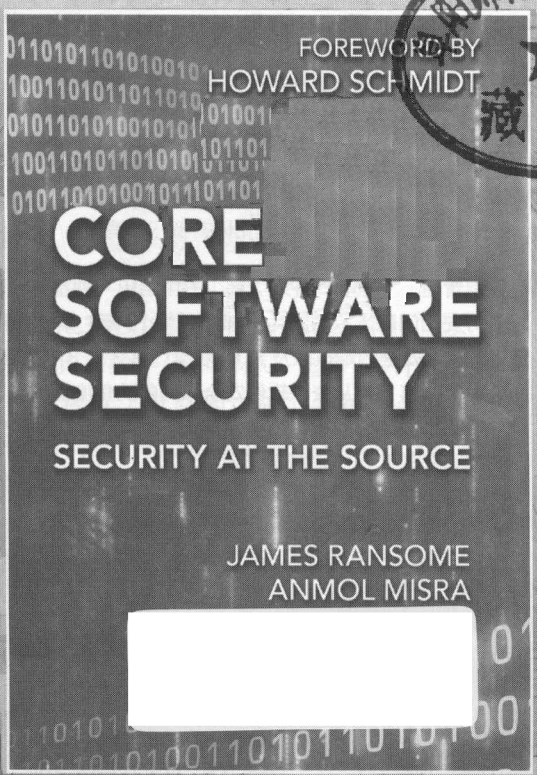
软件安全

从源头开始

[美] 詹姆斯·兰萨姆（James Ransome）
安莫尔·米斯拉（Anmol Misra） 著
丁丽萍 卢国庆 李彦峰 穆海蓉 曹原野 译
宋宇宁 审校

Core Software Security

Security at the Source



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

软件安全：从源头开始 /（美）詹姆斯·兰萨姆（James Ransome），（美）安莫尔·米斯拉（Anmol Misra）著；丁丽萍等译 . —北京：机械工业出版社，2016.7
（计算机科学丛书）
书名原文：Core Software Security: Security at the Source

ISBN 978-7-111-54023-6

Ⅰ. 软… Ⅱ. ①詹… ②安… ③丁… Ⅲ. 软件开发－安全技术 Ⅳ. TP311.52

中国版本图书馆 CIP 数据核字（2016）第 130357 号

本书版权登记号：图字：01-2016-0682

本书阐述什么是人类可控制管理的安全软件开发过程，书中给出一种基于经验的方法，来构建最好用的安全软件开发模型，以应对安全问题。本书分为三部分，共 10 章。第 1 章简要介绍软件安全领域的主题及其重要性；第 2 章讲解软件安全的难点以及 SDL 框架；第 3 ~ 8 章揭示如何将 SDL 及其最佳实践映射到一个通用的 SDLC 框架；第 9 章从资深软件安全架构师的角度给出关于成功方案的看法，并且解读在开发安全软件时针对典型挑战的一些真实方法；第 10 章结合现实世界中的安全威胁，描述如何用合理的架构设计、实现与管理的 SDL 程序来提高安全性。

出版发行：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：谢晓芳	责任校对：董纪丽
印　刷：中国电影出版社印刷厂	版　次：2016 年 8 月第 1 版第 1 次印刷
开　本：185mm × 260mm　1/16	印　张：13
书　号：ISBN 978-7-111-54023-6	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　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brian W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@hzbook.com
联系电话：（010）88379604
联系地址：北京市西城区百万庄南街 1 号
邮政编码：100037

华章科技图书出版中心

译者序

Core Software Security: Security at the Source

一直以来，无论是系统软件还是应用软件，人们在其开发过程中都很少关注安全问题。总是在软件上线后，发现或者被发现了安全问题才意识到有需要解决的安全问题。于是，一个补丁接一个补丁地弥补。用一段时间以后，软件已经补丁摞补丁，而安全问题依然层出不穷。如果开发人员不了解软件安全的基本概念和问题，不了解安全设计的基本方法，不知道安全漏洞的类型有哪些，不能设计出针对安全性的测试用例，就写不出安全可靠的代码，软件的安全就无从谈起。为此，本书作者提出了软件核心安全的根本在于把安全贯穿于软件开发过程的始终，对于安全开发生命周期（Security Development Lifecycle）的概念和模型进行了详细的分析阐述。

本书旨在让软件开发人员把安全理念贯穿于软件开发过程中，对软件的安全需求进行认真分析，把安全融入软件的概要设计和详细设计中，从而写出具有安全防护功能的代码，设计包含针对常见安全漏洞的测试用例，构建安全的运行环境。最终，从源头上关注软件安全，提高软件自身的安全性，减少安全漏洞带来的损失，降低软件安全开发的成本。

由于时间仓促，本书的译文难免有错误和不足之处，恳请广大读者批评指正。

丁丽萍

2016 年 7 月 9 日于北京

全球网络安全威胁正在持续地（即便不是每天）增加。一个一直困扰着我们的问题是如何应对当前的全球网络安全威胁。作者通过本书回答了这个问题，即要开发安全漏洞尽可能少的软件。换言之，我们应该着眼于源头安全，而不是仅仅采取如试图保护网络基础设施等阻挡入侵的方法来解决安全问题。边界安全和深度防御在安全领域中占有一席之地，但软件自身的安全是安全防护的第一关，应该是第一位的。即使在软件源头中存在较少的漏洞，这些漏洞也足以被利用，成为侵犯民族和国家经济利益的武器，或者成为有组织犯罪的网络武器储备。因此，我们不仅要把软件做得更好、更安全，同时，根据现实世界的经验，必须保证该解决方案具有较好的成本效益、操作相关性和可行性以及投资的可行性。源头安全需要软件安全，这是网络基础设施安全的核心。20 年以来，我们一直不可否认的一个事实是软件已成为我们的关键基础设施和日常生活方方面面的重要组成部分。我们注意到软件已经融入我们日常生活使用的各种各样的物品中——从家庭智能电表到汽车都含有软件。遗憾的是，软件安全并没有像软件本身那样以相同的速度发展。许多软件产品依然在这样一种环境下开发：它们在发布后解决问题而不是保证第一次做的就是正确的。这里，有两个主要的问题。

1. 第一个问题是软件漏洞普遍存在。正在寻找和利用软件漏洞的那些人有着充足的漏洞可挖掘，因此，我们必须确保漏洞管理足够好，而且，还必须将目光投向未来并扪心自问："怎样做才能让我们的工作生活越来越依赖的后续软件中不会存在这种类型的漏洞？"这个问题的答案显得尤为重要。因为对于企业来说，减少这些漏洞以及在软件开发过程中阻止它们的出现是非常有益的。使用 SDL 构建软件安全比软件发布以后的系统恢复和漏洞修复的成本更低。

2. 另一个问题是我们需要开始关注称为"0-day 漏洞"的整整一代漏洞。坚持可靠 SDL 的最佳实践，通过从一开始就不允许漏洞发生的方式，找到可能消除 0-day 漏洞的方法，将节省企业的资金投入，使得软件及其用户更安全，关键基础设施更具弹性。总之，这些将对我们所有人都更有益。

作为卓越代码软件保障论坛（Software Assurance Forum for Excellence in Code, SAFECode）的执行董事，我现在重点关注开发人员的安全培训。SAFECode 是一个非营利性的组织，专门致力于通过先进有效的软件保证方法，增加信息、通信技术产品和服务的可信性。软件工程人员安全意识和教育的匮乏，可能是努力实现软件安全计划组织的显著障碍。然而，给予软件开发人员更好的培训，使他们具备编写安全代码所需的技能，仅仅是软件安全方程式的变量之一。软件项目受限于成本和严格的时间表。在这种情况下，因为某处走捷径而牺牲安全是不可避免的。成本、时间和资源通常是软件开发支持安全的三要素，如果你牺牲三者中的任意一个，安全和质量都会深受其害。软件开发环境是围绕程序员建立的，他们受到各个方面的压力：更快地工作，走捷径，以安全和质量为代价编写更多的代码。

100% 安全的软件和系统是不存在的，但开发者和他们的管理者应始终力求最大限度地缓解风险，目的是使攻击者以未经授权的方式进行访问变得更难：

- 不得已利用可追溯的和明显的入侵手段访问，以致他们被立刻发现；

- 花费很多时间试图访问系统，以致最终被注意到；
- 放弃并转向更容易的攻击目标。

确保每一个接触到产品开发生命周期的人拥有支持组织的软件安全过程所需的知识，是致力于软件安全成功的任何组织的一个基本挑战。目标是消除组织在开发传统程序过程中面临的受限于资源约束和知识隔绝的痛苦。

开发者往往需要在巨大的压力下和预算内按时提供更多的功能。很少一部分开发者能够抽出时间来审查他们的代码以发现潜在的安全漏洞。当他们真的抽出时间的时候，他们往往没有经过安全编码方面的培训，缺乏自动化工具、嵌入式流程和程序、资源等，以防止黑客使用数百个常见的漏洞利用技术来触发恶意攻击。不管你喜欢与否，开发者辛勤工作的劳动果实往往是恶意黑客攻击的受害者。那么软件厂商能够做些什么呢？答案的很大一部分是比较老套的：开发者需要获得激励、更好的工具和适当的培训。

遗憾的是，当前的商业意识倾向于不生产安全的软件产品。任何解决方案都需要将其当作一个根本的市场失败问题，而不是简单地希望它不是真的。如果安全性是一个企业追求的目标，那么它需要具有商业意义。最后，安全性要求实际上与任何商业目标都是相同的，应该被视为同等重要。雇主应该期望他们的雇员对他们的工作感到自豪并拥有一定的责任感。雇员应该期望他们的雇主提供完成工作所需的工具和培训。有了这些期望的建立和目标的达成，也许软件行业可以通过减少软件漏洞，更好地提高产品的安全性。

本书讨论了一种过程方法，该方法认为安全性在软件开发过程中发挥着至关重要的作用。本书面向企业高管、各个级别的管理人员和技术领导者，对他们而言，本书所述的这种方法是很独特的。本书同样面向学生和开发者，他们可以了解软件开发过程，树立安全意识，并找到资源帮助他们实现本书所介绍的方法。书中的信息给管理人员、项目经理和技术领导者提供了一定的基础，以改善他们创建的软件、提高应用软件的安全性和质量，并保护隐私信息。

软件开发者知道如何编写软件，以提供高水平的安全性和鲁棒性。那么，为什么软件开发者不练习这些技巧呢？本书将分两部分来回答这个问题。

1. 软件是否安全取决于人们对于程序的分析结果，这些分析包括软件如何使用、在什么条件下使用和在将要部署的环境中它必须满足的安全要求是什么。SDL 还必须延伸到产品发布之外，因为如果计划外的操作环境中软件背后的假设和它们此前隐含的需求不成立，该软件可能不再是安全的。如果需要重新设计一个完整的产品，SDL 过程可能部分或全部重新开始。从这个意义上说，作者建立了所需的准确和有意义的安全要求、管理的度量并为开发软件确立了一个实例。本书还假定不是所有的安全要求均优先于开发过程，并描述它们衍生、分析和验证的过程。

2. 软件高管、领导者和管理人员必须支持鲁棒的编码实践，以及一个业务相关的 SDL 所需的安全性增强，同时支持这种类型工作所需的人员需求、调度、预算和资源分配。作者出色地为这些管理者描述了过程、需求和度量管理，这样他们能够准确地评估一个 SDL 所需的影响和资源，使得他们在组织和环境中的工作能够相对最佳。鉴于这是一个依据真实生活、实地挑战和经验设计的方法，作者描述了如何思考问题，以制定有效的方法并遵照业务流程进行管理。

我特别喜欢第 9 章的作者 Brook Schoenfield 给本书补充的内容，带给我们一个针对企业和软件安全架构师的成熟结论。这个结论借助“第一线”的丰富经验，讲解在“现实世界”

中安全架构如何适应 SDL 过程。特别地，他提供了一个独特和重要的办法来解决 SDL 与安全架构方面的需求，而且这个方法已经过实地测试并真正使用在敏捷软件开发过程中。

我已经认识 James Ransome 博士多年，很高兴他选择信息安全作为他第 10 本书的主题。除了在政府和企业领域担任一些网络安全高级领导职务外，我最近刚刚就任总统特别助理和联邦政府网络安全协调员。我可以自信地说，这是目前信息和全球网络安全最关键的领域。业务和流程问题一直是并将继续是比技术更重要的问题。本书为目前典型的培训资源添砖加瓦，因为它使用了众所周知的 SDL 并提供了运营、业务和成本效益指标。我相信，James Ransome 和 Anmol Misra 所写的书已经一语道破这一主题，将作为专业人士的实践指南和学术教材服务社会若干年。

Hon. Howard A. Schmidt
Ridge Schmidt Cyber 合伙人
SAFECode 执行董事

Hon. Howard A. Schmidt 简介

Hon. Howard A. Schmidt 是战略咨询公司 Ridge Schmidt Cyber 的合伙人。该公司是一个行政服务公司，帮助企业和政府的领导者引领网络安全不断增长的需求。Tom Ridge 也担任该职务，他是美国国土安全部的首位部长。Schmidt 教授还担任 SAFECode 的执行董事。

Schmidt 教授通过长达 40 年杰出的职业生涯，将企业人才、国防、情报、执法、隐私、学术界和国际关系融合在一起。最近他就任总统特别助理和联邦政府网络安全协调员。在此期间，Schmidt 协调各部门网络安全政策的制定和实施，并与联邦、州、地方、国际和私营部门的网络安全合作伙伴通力合作。

此前，Schmidt 教授担任信息安全论坛（Information Security Forum, ISF）的总裁兼 CEO。在 ISF 任职之前，他担任 eBay 公司的副总裁、首席信息安全官和首席安全战略官。此前，他担任微软公司的首席安全官。他还担任过美国国土安全部 US-CERT 合作伙伴计划的首席安全战略官。Schmidt 还拥有超过 26 年的军旅生涯。开始是在空军服役，后来加入了美国亚利桑那州空军国民警卫队。在空军服役期间，他曾担任许多军事和民间职务，最高职务为特别调查办公室的监事会特别代理（AFOSI）。他在刑事调查部门的计算机罪案组担任陆军预备役特别代理长达 12 年，在亚利桑那州钱德勒警察署担任警察。

Schmidt 教授于凤凰城大学获得工商管理学士学位（bachelor's degree in business administration, BSBA）和组织管理硕士学位（master's degree in organizational management, MAOM）。他还拥有人文学名誉博士学位。Schmidt 是爱达荷州立大学的讲座教授，兼卡内基梅隆大学 CyLab 实验室的特聘研究员和波耐蒙隐私研究所（Ponemon Privacy Institute）的特聘研究员。

Schmidt 还是一名业余无线电操作员（W7HAS）、私人飞行员、户外运动者和狂热的哈雷 - 戴维森（Harley-Davidson）骑手。他夫人 Raemarie J. Schmidt 是一个退休的取证科学家和研究人员，是计算机取证领域的讲师。总之，他们是自豪的父母和快乐的祖父母。

前　言

Core Software Security: Security at the Source

软件驱动机器的时代在过去的几年里取得了飞跃式的发展。战斗机飞行员、股票交易所场内交易员、医生、工业生产者和发电厂运营商等专业人员的工作对武器系统、医疗系统以及国家基础设施关键要素的运行至关重要，而这些行业已经或正在迅速地被软件接管。这是革命性的一步。从前必须要人类大脑完成的复杂的、非重复性的任务，现在已经被软件驱动程序控制的机器大脑和神经系统所取代。这种变化体现在政府、军队、罪犯、活动家和其他对手可以尝试破坏、修改或影响国家基础设施的改变，也体现在社会和文化方式的转变。同样，这种变革对企业也是有一定影响的，比如，这些年来我们看到的越来越多的网络商业间谍案件。以前的大规模军队、昂贵和毁灭性的武器系统与平台、武装抢劫、信息窃取、暴力抗议活动和武装叛乱很快被所谓网络战、网络犯罪和网络行为所替代。

这些网络方法可能最终就像之前使用的技术一样产生深远的影响，而潜在的软件漏洞如果被利用，则可能会导致以下问题。

- 全部或部分基础设施被中断，包括电网、核电站、通信介质和应急系统。
- 化工厂被用于产生大规模的爆炸和剧毒云。
- 远程控制、修改或破坏关键的武器系统或平台。
- 导致监控系统被篡改或不可用。
- 经济剥削或讹诈犯罪。
- 操控金融市场和投资。
- 通过改变医疗支持系统或设备、手术计划或药物处方来实施谋杀或伤害人体健康。
- 通过网站毁损或底层 Web 应用的关闭和破坏修改自动投票软件、进行敲诈勒索或导致品牌信用降级，从而产生政治暴动或者导致一些人群的特殊利益受到影响。

网络方法的一个副作用就是提供给我们以之前想都不敢想的大规模、远程和匿名方式解决上述问题的能力。这一能力是在受到司法保护的位置通过远程探测和入侵的方式解决上述问题。同时，这也赋予了政府、犯罪集团和活动家特殊的能力，包括代理主要犯罪人用于逃避责任、避免接受检测和规避政治后果。

虽然有很多关于网络安全的宣传，但真正的致命弱点是（不安全的）软件。这些软件提供总体控制或变更如上所述的一个目标的潜在能力。软件安全的关键性在于，我们迅速步入了新的时代，从前被忽视的关于人类思维被软件驱动机器所替代的这件事也不容小觑。正是出于这个原因，我们撰写了这本书。与此相反，在可预见的将来，软件程序将继续由人类编写。这也意味着，新的软件将继续构建在遗留代码或软件之上，而这些遗留代码在编写时安全还不被重视或者是在复杂攻击盛行之前编写的。只要人类写程序，成功的关键就是保障有关计划的实施，使得软件开发项目的过程更加高效和有效。尽管本书的方法包括软件安全人员、软件安全流程和软件安全技术方法，但是我们认为在软件安全中人是最重要的组成部分，原因是软件的开发、管理、使用都是由人来完成的。以下是软件安全性的逐步实施过程，该过程与当今的技术、运营、业务和开发环境相关，并且强调人可以做什么去控制和管理流程的最佳实践与度量方式。我们将永远面临安全问题，但是本书有助于在软件最终发布或部署时

尽量减少安全问题。希望你能喜欢我们的书，就像我们喜欢写它一样。

本书内容

本书讲述关乎当今科技、运作、商业及开发环境的软件安全过程。作者着重讲述什么是人类可控制、管理的安全软件开发过程，并用一种较好的实践方式和度量方式来表达。虽然安全问题将会始终存在，但是本书将教你如何将公司的能力最大化，同时在软件发布前将软件被攻击的概率最小化，以及如何将安全构建融入开发过程之中。作者拥有世界五百强公司的工作经历，目睹过一定数量的软件安全开发生命周期（SDL）故障。本书采用一种基于经验的方法，来构建最好用的 SDL 模型，以应对前面提到的问题，并在 SDL 安全软件开发模型中得到解决。本书开篇概要介绍 SDL，然后讲解将 SDL 实践方法映射到 SDL 模型，同时解释如何运用此模型来建立并管理成熟的 SDL 程序。虽然安全并不是近些年业界编写软件过程中的天然组成部分，但是，作者相信，将安全开发加入到软件开发中将是可能的、可行的，也是切实必要的。作者同时认为，本书中提到的软件安全实例以及模型将给所有读者留下清晰的印象，无论是经理还是主管或是从业人员都将在其中获益。

读者对象

本书面向任何对企业级软件安全感兴趣的人，包括产品安全和质量主管、软件安全架构师、安全顾问、软件开发工程师、企业 SDLC 项目经理、首席信息安全官、首席技术官和首席隐私官。如果你想了解在企业级软件开发过程中是如何实现软件安全的，本书将是一本不可错过的好书。

本书结构

本书分为三部分，共 10 章。第 1 章简要介绍软件安全领域的主题及其重要性。第 2 章讲述软件安全的难点以及 SDL 框架。第 3 ~ 8 章讲述如何将 SDL 及其最佳实践映射到一个通用的 SDLC 框架。根据第 3 ~ 8 章描述的解决方案，第 9 章叙述一个老练的软件安全架构师关于成功方案的看法，并解读在开发安全软件时针对典型挑战的一些真实方法。第 10 章结合现实世界中的安全威胁，描述我们如何用合理的架构设计、实现与管理的 SDL 程序来提高安全性。

假设

本书假设读者了解基本的软件开发过程（方法论）以及基本的安全概念。建议对 SDL、不同种类的安全测试，以及安全架构有所了解，但是不作要求。对于大部分主题，在深入探讨具体问题前，我们会循序渐进地介绍相关知识。

致谢

写书就像旅行，如果没有导师、朋友、同事还有家人的支持，它将无比困难。在写作过程中很多人给予了我帮助。首先，我们要感谢 CRC 出版社 John Wyzalek 编辑的耐心帮助、支持以及贡献。我们还要感谢 DerryField Publishing 产品组的 Theron Shreve、Lynne Lackenbach 和 Marje Pollack。

我们都要感谢 Hon. Howard A. Schmidt（Ridge Schmidt Cyber 合伙人；SAFECode 执行董事；联邦政府总统及网络安全司前特别助理），并感谢 Dena Haritos Tsamitis（网络信息研究所主管；卡耐基梅隆大学 CyLab 实验室教育、训练、外展主管）对项目的支持。我们同样要感谢 Brook Schoenfield 对项目的帮助。感谢他加入团队并证明有另一种不同于现有方法的软件安全架构、部署和管理方法，同时感谢他撰写了本书第 9 章。我们要感谢整个网络安全社区，我们属于其中的一员并以此为荣。最后我们衷心感谢这些年所有共事并合作过的人。

——James Ransome 和 Anmol Mistra

借此机会，我真心感谢我的妻子 Gail。感谢她的耐心和理解，也感谢她在初步校对中的工作。特别要感谢的是我的合作者 Anmol Misra。他作为合作者加入我们关键消息的开发已经三年多了，并且一并完成了这本书。另一位要特殊感谢的是 Howard Schmidt。感谢他为本书作序。我们彼此分享做实干者的热情。最后和各位读者分享 Walter Bagehot 的一句话："生命里最快乐的事就是做别人声称做不成的事。"

——James Ransome

多年来许多人指导并帮助过我。我想感谢所有这些人，可惜篇幅有限。你知道我说的是谁，我想感谢他们的耐心、鼓励与支持。我要感谢我的合作者 James Ransome。多年来他一直是我的良师益友。他对我的帮助无法言表。最后我想借此机会感谢我的家人：妈妈、爸爸、Sekhar、Anupam 和 Mausi。没有他们无私的支持和爱护，我是不可能完成任何事情的。他们曾经问我在写完这本书后是否能休假，是否能有一个"正常"的日程安排。我想说，我会的，现在就会。希望如此。

——Anmol Misra

James Ransome

James Ransome 博士是负责产品安全的高级总监，全面负责 McAfee 产品安全项目，该项目是一个企业范围的举措，它支持 McAfee 业务部门给客户提供一流的、安全的软件产品。在担任该职位期间，James 负责设置安全战略，管理 McAfee 业务单元相关的安全活动，保持与 McAfee 产品工程师的联系并与其他领导者合作，以帮助定义和构建产品的安全功能。

他职业生涯的特点是在私人和公共行业担任领导职务，其中包括三个首席信息安全官（Chief Information Security Officer, CISO）和四个首席安全官（Chief Security Officer, CSO）的职务。在进入企业领域前，James 有 23 年从政经历，包括美国情报界、联邦执法部门和国防部的各种职务。

James 拥有信息系统博士学位。他的博士论文包括开发 / 测试安全模型、体系结构，并提供了配合有线 / 无线网络安全方面领先的实践，其论文是信息安全保障教育程序 NSA/DHS 中心的卓越学术成果的一部分。他是多本信息安全书籍的作者，本书是第 10 本。James 是计算与信息学科国际荣誉协会 Upsilon Pi Epsilon 的成员。他是一名注册信息安全经理（Certified Information Security Manager, CISM），一名信息系统安全认证专家（Certified Information Systems Security Professional, CISSP），还是波耐蒙研究所（Ponemon Institute）的特聘研究员。

Anmol Misra

Anmol Misra 是信息安全领域的一位作家和拥有丰富经验的安全专业人士。他的专长包括移动和应用安全、漏洞管理、应用和基础设施的安全评估，以及安全代码审校。他在思科公司信息安全组担任项目经理。在此期间，他主要负责制定和实施安全策略与计划，以把安全最佳实践纳入思科主导的产品的各个方面。就职于思科公司之前，Anmol 是安永国际会计公司的高级顾问。在这个岗位上，他给财富 500 强客户提供定义和改进信息安全计划与实践的咨询服务。他帮助企业降低 IT 安全风险，并通过改善其安全状况使其遵从法规。

Anmol 是《 Android Security: Attacks and Defenses 》的合作者，是《 Defending the Cloud: Waging War in Cyberspace 》的贡献作者。他于卡内基梅隆大学获得信息网络硕士学位和计算机工程工学学士学位。他工作于加利福尼亚州的旧金山。

目　录

Core Software Security: Security at the Source

出版者的话
序
前言
作者简介

第 1 章　引论 …… 1

1.1　软件安全的重要性和相关性 …… 1
1.2　软件安全和软件开发生命周期 …… 4
1.3　代码的质量与安全 …… 6
1.4　SDL 三个最重要的安全目标 …… 6
1.5　威胁建模和攻击面验证 …… 7
1.6　本章小结：期望从本书中学到什么 … 8
参考文献 …… 8

第 2 章　安全开发生命周期 …… 11

2.1　克服软件安全中的挑战 …… 11
2.2　软件安全成熟度模型 …… 12
2.3　ISO/IEC 27034：信息技术、安全技术、应用安全 …… 13
2.4　其他 SDL 最佳实践的资源 …… 14
2.4.1　SAFECode …… 14
2.4.2　美国国土安全软件保障计划 … 14
2.4.3　美国国家标准与技术研究院 … 15
2.4.4　MITRE 公司公共计算机漏洞和暴露 …… 16
2.4.5　SANS 研究所高级网络安全风险 …… 17
2.4.6　美国国防部网络安全与信息系统信息分析中心 …… 17
2.4.7　CERT、Bugtraq 和 SecurityFocus …… 17
2.5　关键工具和人才 …… 17
2.5.1　工具 …… 18
2.5.2　人才 …… 19
2.6　最小特权原则 …… 21
2.7　隐私 …… 22
2.8　度量标准的重要性 …… 22
2.9　把 SDL 映射到软件开发生命周期 … 24
2.10　软件开发方法 …… 28
2.10.1　瀑布开发 …… 28
2.10.2　敏捷开发 …… 29
2.11　本章小结 …… 31
参考文献 …… 31

第 3 章　安全评估（A1）：SDL 活动与最佳实践 …… 35

3.1　软件安全团队提早参与项目 …… 35
3.2　软件安全团队主持发现会议 …… 37
3.3　软件安全团队创建 SDL 项目计划 … 37
3.4　隐私影响评估计划启动 …… 38
3.5　安全评估（A1）成功的关键因素和度量标准 …… 41
3.5.1　成功的关键因素 …… 41
3.5.2　可交付成果 …… 43
3.5.3　度量标准 …… 44
3.6　本章小结 …… 44
参考文献 …… 44

第 4 章　架构（A2）：SDL 活动与最佳实践 …… 46

4.1　A2 策略一致性分析 …… 46
4.2　SDL 策略评估和范围界定 …… 48
4.3　威胁建模 / 架构安全性分析 …… 48
4.3.1　威胁建模 …… 48
4.3.2　数据流图 …… 50
4.3.3　架构威胁分析和威胁评级 …… 53
4.3.4　风险缓解 …… 65
4.4　开源选择 …… 68
4.5　隐私信息收集和分析 …… 69
4.6　成功的关键因素和度量标准 …… 69

4.6.1 成功的关键因素 …… 69
4.6.2 可交付成果 …… 70
4.6.3 度量标准 …… 70
4.7 本章小结 …… 71
参考文献 …… 71

第 5 章 设计和开发（A3）：SDL 活动与最佳实践 …… 74

5.1 A3 策略一致性分析 …… 74
5.2 安全测试计划构成 …… 74
5.3 威胁模型更新 …… 81
5.4 设计安全性分析和检查 …… 81
5.5 隐私实现评估 …… 83
5.6 成功的关键因素和度量标准 …… 85
5.6.1 成功的关键因素 …… 85
5.6.2 可交付成果 …… 86
5.6.3 度量标准 …… 87
5.7 本章小结 …… 88
参考文献 …… 88

第 6 章 设计和开发（A4）：SDL 活动与最佳实践 …… 90

6.1 A4 策略一致性分析 …… 90
6.2 安全测试用例执行 …… 92
6.3 SDLC/SDL 过程中的代码审查 …… 94
6.4 安全分析工具 …… 97
6.4.1 静态分析 …… 99
6.4.2 动态分析 …… 101
6.4.3 模糊测试 …… 103
6.4.4 人工代码审查 …… 104
6.5 成功的关键因素 …… 106
6.6 可交付成果 …… 107
6.7 度量标准 …… 107
6.8 本章小结 …… 108
参考文献 …… 108

第 7 章 发布（A5）：SDL 活动与最佳实践 …… 111

7.1 A5 策略一致性分析 …… 111
7.2 漏洞扫描 …… 113
7.3 渗透测试 …… 114
7.4 开源许可审查 …… 116
7.5 最终安全性审查 …… 117
7.6 最终隐私性审查 …… 119
7.7 成功的关键因素 …… 120
7.8 可交付成果 …… 121
7.9 度量标准 …… 122
7.10 本章小结 …… 122
参考文献 …… 124

第 8 章 发布后支持（PRSA1 ~ 5）… 125

8.1 合理调整软件安全组 …… 125
8.1.1 正确的组织定位 …… 125
8.1.2 正确的人 …… 127
8.1.3 正确的过程 …… 127
8.2 PRSA1：外部漏洞披露响应 …… 130
8.2.1 发布后的 PSIRT 响应 …… 130
8.2.2 发布后的隐私响应 …… 133
8.2.3 优化发布后的第三方响应 …… 133
8.3 PRSA2：第三方审查 …… 134
8.4 PRSA3：发布后认证 …… 135
8.5 PRSA4：新产品组合或云部署的内部审查 …… 135
8.6 PRSA5：安全架构审查和基于工具评估当前、遗留以及并购的产品和解决方案 …… 136
8.6.1 遗留代码 …… 136
8.6.2 兼并和收购 …… 137
8.7 成功的关键因素 …… 138
8.8 可交付成果 …… 139
8.9 度量标准 …… 140
8.10 本章小结 …… 140
参考文献 …… 140

第 9 章 将 SDL 框架应用到现实世界中 …… 142

9.1 引言 …… 142
9.2 安全地构建软件 …… 145
9.2.1 编写安全的代码 …… 146
9.2.2 人工代码审查 …… 149

9.2.3 静态分析…… 150
9.3 确定每个项目的正确行为…… 153
9.4 架构和设计…… 161
9.5 测试…… 167
9.5.1 功能测试…… 168
9.5.2 动态测试…… 168
9.5.3 攻击和渗透测试…… 171
9.5.4 独立测试…… 172
9.6 敏捷：冲刺…… 172
9.7 成功的关键因素和度量标准…… 175
9.7.1 安全编码培训计划…… 175
9.7.2 安全编码框架（API）…… 175
9.7.3 人工代码审查…… 176
9.7.4 独立代码审查和测试（专家或第三方）…… 176
9.7.5 静态分析…… 176
9.7.6 风险评估法…… 176
9.7.7 SDL 和 SDLC 的集成…… 176
9.7.8 架构人才的发展…… 176
9.8 度量标准…… 177
9.9 本章小结…… 177
参考文献…… 178

第 10 章 集成：应用 SDL 防止现实的威胁…… 180

10.1 战略、战术和特定于用户的软件攻击…… 180
10.1.1 战略攻击…… 181
10.1.2 战术攻击…… 182
10.1.3 特定于用户的攻击…… 182
10.2 应用适当设计、管理和集中的 SDL 克服组织与业务挑战…… 182
10.3 软件安全组织的现状和影响力… 183
10.4 通过合理的政府管理克服 SDL 审计和法规挑战…… 183
10.5 软件安全的未来预测…… 184
10.5.1 坏消息…… 184
10.5.2 好消息…… 185
10.6 总结…… 185
参考文献…… 186

附录 关键的成功因素、可交付成果、SDL 模型每个阶段的指标…… 189

引　论

欢迎阅读此书，本书旨在关注未来信息安全领域最重要的话题：软件安全（software security）。下面几节将涵盖五大主题，突出说明软件安全的需求、价值和挑战。这将为本书的剩余部分定下基调。我们将描述一个软件安全模型：应用一个操作上相关且可控的安全开发生命周期（Security Development Lifecycle，SDL）来构建安全软件。SDL 适用于所有的软件开发生命周期（Software Development Lifecycle，SDLS）。五大主题以及在第 1 章引入它们的原因如下。

1. **软件安全的重要性和相关性。**软件是我们在现实世界中做任何事情的关键，同时，软件也分布在最关键的系统中。基于此，软件的安全设计是至关重要的。大多数信息技术（Information Technology，IT）相关的安全解决方案已经能够有效地降低不安全软件带来的风险。为了证明一个软件安全程序的合理性，必须知晓没有构建安全软件带来的金钱成本和其他风险的重要性与相关性，以及构建安全软件的重要性、相关性和成本。总而言之，软件安全同样是一个商业决定，因为它关注避免安全风险。

2. **软件安全和软件开发生命周期。**在这里，重要的一点是要区分在软件开发中我们熟知的*软件安全*（software security）和*应用程序安全*（application security）。尽管这两个术语经常互用，但是我们仍然需要区分它们。因为实现这两个目的的程序在管理过程中存在明显的不同。在模型中，*软件安全*表示在 SDLC 中，应用 SDL 构建安全的软件，而*应用程序安全*表示发布后运行过程中软件和系统的保护。

3. **高质量和安全代码。**尽管安全代码未必是高质量代码，同时高质量代码也未必是安全代码，但是软件的开发过程是基于高质量和安全代码原则的。你不能拥有不安全的高质量代码，更不能拥有劣质的安全代码，它们相辅相成。至少，质量和软件安全程序应该在开发过程中紧密结合；理想情况下，它们应该是同一组织的组成部分，以及软件开发工程部门的一部分。这将在本书中后面的章节中从组织和操作角度进一步讨论。

4. **三个最重要的 SDL 安全目标。**所有软件安全分析和构建的核心是三个最重要的安全因素：*保密性*（Confidentiality）、*完整性*（Integrity）和*可用性*（Availability），也称为 C.I.A. 模型。为了确保软件开发是安全的，上述三个特性必须一直作为整个 SDL 过程中的主要组成部分。

5. **威胁建模和攻击界面验证。**威胁建模和攻击界面验证是 SDL 中最耗时和难以理解的部分。在当今的敏捷软件开发中，必须正确处理这个问题，否则无法保证软件安全。SDL 中的威胁建模和攻击界面验证，将最大限度地避免软件产品发布后发现安全漏洞。我们认为这个功能非常重要，因此本书将专门用一节和一章来关注这个主题。

1.1　软件安全的重要性和相关性

2005 年美国总统信息技术顾问委员会（PITAC）报告指出："通用的软件工程实践蕴含有危险的错误，比如处理不当的缓冲区溢出，这导致每年存在成百上千的攻击程序危害成千上

万的计算机。”[1] 发生这种情况的主要原因是“当今商业软件工程缺乏在合理成本内构建高质量、安全产品的科学基础和严格管理。”[2]

高德纳咨询公司（Gartner Group）报道超过 70% 的通用商业安全漏洞出现在软件应用中，而不是网络边界内[3]。因此关注应用安全，旨在降低糟糕的软件开发、集成和部署带来的风险。结果是，软件保障迅速成为金融、政府、制造业等部门信息保障（Information Assurance, IA）关注的领域，并且用于降低不安全代码带来的风险：将安全构建到软件开发生命周期中是一种良好的商业意识。

美国国土安全部 2006 年草案“软件生命周期中的安全”描述如下：

> 安全的软件和不安全的软件最关键的区别在于描述、设计和开发软件等过程与实践中的性质……主要采用安全增强的过程和实践，尽可能早地修复软件开发生命周期中潜在的安全漏洞，比当前普遍采用的频繁开发和发布运行中软件补丁的方法更加经济实用。[4]

在 2011 年美国 RSA 会议上，大家着重强调了云安全问题，但几乎没有讨论如何解决该问题；然而，在 2012 年 RSA 会议上，几乎全是关于解决一年前确定的云安全问题的研究。相同的事情也发生在 2012 年，开始于一些重要的学术会议，随后在 2013 年关于软件安全解决方案的讨论继续成为一个主要关注点。例如，在 2012 年年初，《信息周刊》（Information Week）提出“外部代码审查”是 2012 年十大安全趋势之一[5]，并且指出“业务职责是明确的：开发者必须花时间清楚地编写代码，并在代码投入生产之前根除任何可能的安全缺陷。”同样 2012 年 3 月 1 日也有一篇发表的热门安全文章，题目为“To Get Help with Secure Software Development Issues, Find Your Own Flaws”，它成为 2012 年旧金山举办的 RSA 专题讨论会上最精彩的部分。[6] 这次专题讨论会做了大量的工作以识别一些关键问题，但并没有解决识别出的软件安全挑战。然而，事情开始在 2012 年年中有所改变：2012 年 5 月召开的微软就职安全发展会议议程[7]，不是关于微软的，更多的是带来安全软件开发思想的领导力标准，并分为三个独立的模块，包括“安全工程”“安全开发生命周期（SDL）& 商业”和“过程管理”来讨论工业和安全软件开发中最重要的安全问题的解决方案。这种趋势在 2012 年美国黑帽会议[8]、2013 年 RSA 会议[9] 和 2013 年微软安全开发会议[10] 上得到延续。

想想看：是什么真正引发了当今世界大多数的信息安全问题？黑客和网络犯罪分子的主要目标又是什么？那就是不安全的代码。是什么已经迅速成为软件开发中投入最高的不必要的成本？那就是已投入市场的软件产品中不安全代码带来的缺陷。当这些缺陷被发现和（或者）被利用后，它们会导致当今产品开发周期中断以修复一些本应该在包含缺陷的产品开发过程中修复的错误；它们会导致产品发布日期推迟，因为参与当前项目的个人或者团队脱离该周期去修复以前发布产品的问题；同时它们会导致脆弱性范围渐变，因为在某一产品中发现的脆弱性可能影响网络、软件即服务（Software as a Service，SaaS）和云应用中其他产品的安全性。它们也会产生法律问题，如声誉下降和诸如在过去的几年里索尼（Sony）、赛门铁克（Symantec）和 RSA 经历的公共关系的噩梦。它们同样能够导致公司重大的责任。在有广泛的法规监管用户隐私和数据泄露的时代，企业应该承担的责任将迅速增加，即使对于大公司也一样。需要指出的是，即便是在高科技领域，消费者、客户、监管部门和媒体都已经开始意识到不仅要迫切解决软件安全问题，实际上更需要以一种结构化的安全软件开发生命周期或