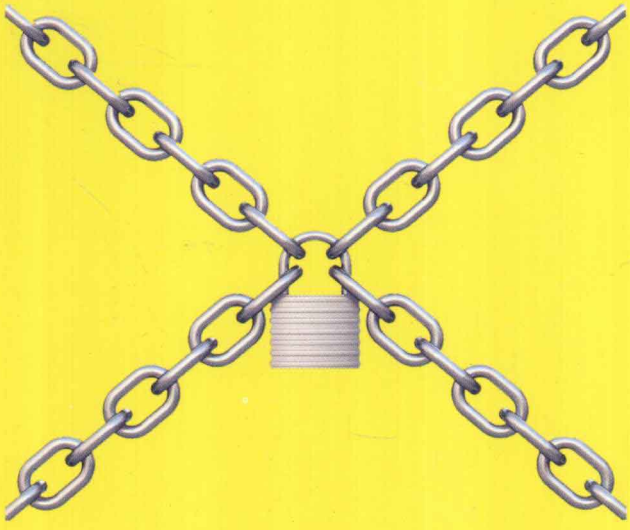
HZ BOOKS
华章科技

国际对象管理组（OMG）KDM分析部门CTO和CEO共同执笔。
美国国土安全部、国家网络安全部、全球网络安全管理组软件质量保证总监鼎力推荐！

安全技术大系
SECURITY



System Assurance: Beyond Detecting Vulnerabilities

系统安全保证

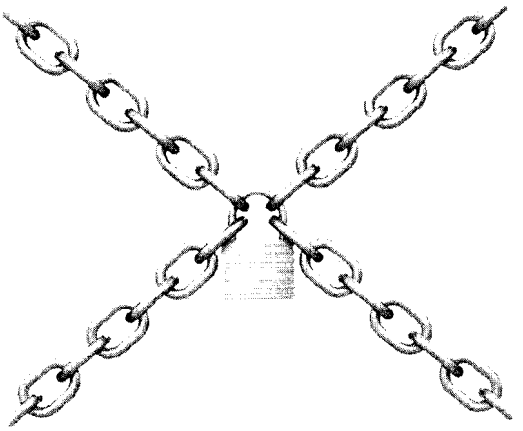
策略、方法与实践

Nikolai Mansourov　Djenana Campara　著

莫凡　赵见星　杨勇　莫非　译

机械工业出版社
China Machine Press

安全技術大系
SECURITY



System Assurance: Beyond Detecting Vulnerabilities

系统安全保证

策略、方法与实践

Nikolai Mansourov　Djenana Campara　著

莫凡　赵见星　杨勇　莫非　译

机械工业出版社
China Machine Press

本书由国际对象管理组织（OMG）KDM 分析部门 CTO 和 CEO 共同执笔，美国国土安全部、国家网络安全部、全球网络安全管理组的软件质量保证总监鼎力推荐。

全书用系统化方式描述了软件系统的安全保证方法，充分利用了对象管理组的软件安全保证体系标准，最终形成一个综合系统模型用于系统的分析和证据收集。主要内容包括：第一部分（第 1～3 章）介绍网络安全基础知识，以及对象管理组的软件安全保证体系。第二部分（第 4～7 章）介绍网络安全知识的不同方面，以建立网络安全论据，包括系统知识、与安全威胁和风险相关的知识、漏洞知识，还描述了网络安全内容的新格式，即机器可识别的漏洞模式。第三部分（第 8～11 章）介绍对象管理组的软件安全保证体系的协议，包括通用事实模型、语义模型、业务词汇和业务规则语义标准，以及知识发现元模型。第四部分（第 12 章）通过一个端到端的案例研究来阐释系统安全保证方法、综合系统模型和系统安全保证案例。

本书内容广泛，系统性强，适合信息安全领域的研究人员、技术开发人员、高校教师等参考。

System Assurance: Beyond Detecting Vulnerabilities
Nikolai Mansourov, Djenana Campara
ISBN: 978-0-12-381414-2

本书版权登记号：图字：01-2011-6598

图书在版编目（CIP）数据

系统安全保证：策略、方法与实践/（俄）曼索洛夫（Mansourov, N.）等著；莫凡等译. —北京：机械工业出版社，2012.6
书名原文：System Assurance: Beyond Detecting Vulnerabilities

ISBN 978-7-111-38860-9

Ⅰ. 系…　Ⅱ. ①曼…　②莫…　Ⅲ. 信息系统－安全技术　Ⅳ. TP309

中国版本图书馆 CIP 数据核字（2012）第 130798 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：吴　怡
三河市杨庄长鸣印刷装订厂印刷
2012 年 9 月第 1 版第 1 次印刷
186mm×240mm · 16.25 印张
标准书号：ISBN 978-7-111-38860-9
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

Claude Langton：我还没有听到任何关于谋杀的消息。

Hercule Poirot：你不会听到的。因为到目前为止，谋杀还没有发生。要知道，如果在谋杀发生之前就去调查，那么就可以阻止它。

——阿加莎·克里斯蒂《波洛探案集：黄蜂的巢》

系统安全保证类似于侦探，因为侦探大部分时间都在外面搜集证据：采访潜在的证人，或者搜寻一个具有重要意义的“烟头”。搜集证据是至关重要的，证据驱动着调查，调查要遵循事实，而在进行侦探时还应该花些时间来计划如何进行调查。最后，证据要展示在法官和陪审团面前。

系统安全保证需要调查系统中的安全漏洞，漏洞是指系统中允许攻击者进行网络犯罪的弱点。系统漏洞可以是有意或无意植入系统中的，也可能是因为遗漏所造成的，甚至是因为存在设计问题的协议造成的。通常来讲，漏洞是跟某一特定种类的系统相关的，即某个系统中的漏洞在其他系统中未必成为漏洞。这正如我们教育孩子时，告诫他们不要跟陌生人讲话，而对于一个销售人员来说，同陌生人讲话是再正常不过的事情了。所以，漏洞是使得网络系统易于遭受攻击的地方，也是导致不安全的地方。

系统安全保证相当于在网络遭受攻击之前执行的取证调查。然而，系统安全保证需要搜集足够的证据，以得到一个综合的结论：该系统可防御某种类型的攻击。系统安全保证是确保我们的系统更加安全、更加可靠、可抵抗网络犯罪的一种方法。网络系统的安全风险是现实存在的，在一个新系统投入运行前，必须明白和承认这些风险。最后，广大用户需要明白并承认使用该网络系统的风险。所以，研发人员和公司需要对发布新的网络系统负责，而监管机构和广大用户是“陪审团”，公司需要向他们阐明系统的安全性。系统安全保证是建立清晰的、全面的、可防御的案例，以保证系统可以安全、可靠地运行，并且不存在漏洞。发布系统安全保证案例意味着开诚布公地与“陪审团”交流，解决他们所提出的尖锐问题，以此建立信任。

安全保证案例不是单单通过声明就能得到的，其可信度必须要有证据支撑。我们可以搜集很多东西作为证据来支撑关于系统安全性和可靠性的声明。但是不同的证据有着不同的证明能力，有些证据比其他证据更具有说服力。例如：某个著名专家声明“我相信这个投票机是安全的”，这很具有说服力。但是，另一个资深的道德黑客说：“历经五天攻击，我也没能成功入侵这个系统。”这个声明可能更具有说服力。因为它是基于具体的、与所关心的系统直接相关的调查式行动得出的，而不是纯粹的专家意见。显然，第二个声明者对于所关心的系统更了解，这使得第二个声明比第一个更具有说服力。

表面上看来，防御方对于系统具有更深刻的认识，并在设计初期就考虑了安全问题以有效避免出现漏洞，从而得到一个健壮的系统。然而，防御方并不能面面俱到。有些系统由商业组件、

遗留系统组件和开源组件组成，其安全性是未知的，这样混合得到的系统更加脆弱，易于遭受攻击而被破坏。所以，尽管在系统生命周期的初始阶段就开始考虑了系统的安全性，但系统仍然存在很多未知的缺陷，并且处于开发者的控制范围之外。况且，系统开发者并不能很好地预见会造成网络攻击的漏洞，因为他们缺乏攻击者的视角，即可以设计系统攻击的犯罪潜质。最后，系统知识的生命周期很短，随着开发者不断更换项目，对于之前的系统知识只会慢慢遗忘。所以，只有代码，甚至是机器码才是唯一可信的关于系统的知识。

黑客往往比我们更了解我们的系统，这一点我们必须要正视。优秀的黑客不断地研究我们的系统，并找到新颖的方法来入侵我们的系统。他们这样做一是为了好玩，二是为了利益。网络犯罪之所以成为人们讨论的焦点，是因为攻击者在成功入侵系统后，会将攻击方法在黑客之间共享，从而形成更大的犯罪圈。这些犯罪圈中的人则更关注利用这些漏洞来获取利益，而非乐趣。网络犯罪更为严重的是攻击者的规模可以很大，而又分散于世界的各个角落，跨越了国界。黑客们不仅共享攻击技术，还武装自己，编写很容易就可以使用的脚本，使得攻击成为可重复的、可支付的。而今利用我们系统中漏洞的恶意软件已经可以养活有组织的犯罪集团，而这些犯罪集团正在形成更大的犯罪产业。

攻击者和防御者都钟爱自动化的代码分析工具，用于检测系统漏洞。然而，事实上双方有根本的不同点：攻击者可以进行特定的或碰巧的漏洞检测，而这些方法却不适合于防御者，防御者需要系统地、精细地了解系统风险并设计安全机制。因此，我们应当把重点放在通过系统、合理地调查安全威胁来实现系统的安全保证上。

那么如何才能使得网络防御是系统的、可重复的和可支付的？解决之道在于采用综合治理方案：积累公共网络安全知识，建立自动化的工具来扩大防御者的优势，从而超越攻击者。建立这些协作性方案需要注意如下几个方面。首先，需要标准化的协议将用于处理现代复杂系统的分析工具组织起来。通常，一个公司或者安全研究人员所做的工作有一定的局限性，会使得解决方案有诸多限制、效率低下、方案折中。同时还需要一个支持互操作组件的更大市场，与使用乐高积木块类似，通过不同厂商的组件，组建强健的分析方案。其次，需要开发标准协议以支持网络安全内容的存储和信息交换。为了使得网络安全内容和漏洞知识是可重复的、可支付的，它们必须是机器可识别的，并且对于防御者是可用的。换句话说，对于安全保证来说，既要有工具组成的基本系统，还要有机器可识别的内容。

我们之所以将本书命名为《System Assurance：Beyond Detecting Vulnerabilities》，是因为系统的、可重复的、可支付的网络防御远远超出了漏洞检测方面的知识，此外本书还包括系统知识、风险和威胁知识、安全防护知识、安全保证论据知识，以及回答系统为什么安全的相应证据。换句话说，当检测到至少一个潜在漏洞并将其作为证据呈现时，就可以声明系统是不安全的。但是，如果没有检测到任何漏洞，就真的意味着系统是安全的吗？实际上并不是这样的。为了证明系统是安全的，还需要有力的证据，包括：工具被正确使用；在理解不同人编写的代码时，没有任何歧义；没有任何代码被落下，等等。系统安全保证工具远远超出了漏洞检测的范围，它还要包括提供证据以支持系统安全的声明。

我们被授权参加正在解决此问题的网络安全社区，其中，对象管理组（Object Management Group，OMG）是一个开放成员的非营利性国际计算机组织，目的是开发企业整合标准。本书介绍

了对象管理组的软件安全保证体系（Software Assurance Ecosystem，SAE），这是一个发现、整合、分析、发布现存软件系统中事实的通用框架，它的基础是系统事实交换的标准协议，是对象管理组的知识发现元模型（Knowledge Discovery Metamodel，KDM）。业务词汇和业务规则的语义（Semantics of Business Vocabularies and Business Rules，SBVR）定义了安全策略规则和安全保证模型交换的标准协议。综合使用这些标准，网络安全社区就可以积累并发布机器可识别的网络安全内容，并实现系统保护的自动化。最后，安全保证论据已由对象管理组的软件安全保证案例元模型（Software Assurance Case Metamodel，SACM）定义为机器可识别的内容。我们描述了一个特有的系统安全保证方法，它充分利用了对象管理组的软件安全保证体系标准，最终形成一个综合系统模型（Integrated System Model），该系统是一个通用表示形式，用于系统分析和证据收集。

对象管理组（OMG）的软件安全保证体系的核心叫做通用事实模型（Common Fact Model，CFM），它是一个形式化的方法，为信息交换、通用 XML 交换格式和面向事实的整合建立通用词汇库。

本书涉及内容广泛。第一部分（第 1 ~ 3 章）介绍了网络安全基础知识，用于建立系统的、可重复的、可支付的网络防御解决方案，以及对象管理组创建软件安全保证体系的动机。接着讨论了系统安全保证的本质及其与漏洞检测的区别，并简要介绍了对象管理组的软件安全保证体系的标准。对象管理组的软件安全保证体系描述了一个端到端的方法，该体系综合了风险分析、架构分析和代码分析方法来建立一个完整的安全保证方案，该体系以 KDM 的面向事实的可重复的系统安全保证方法（Fact-Oriented Repeatable System Assurance，FORSA）为基础。

第二部分（第 4 ~ 7 章）介绍了网络安全知识的不同方面，以建立网络安全论据。这些知识包括系统知识、与安全威胁和风险相关的知识以及漏洞知识。最后，描述了网络安全内容的新格式，即机器可识别的漏洞模式。在描述网络安全知识时，我们使用业务词汇和业务规则的语义（SBVR）来概述通用网络安全词汇，而这些词汇由通用事实模型和 SBVR 标准开发生成。

第三部分（第 8 ~ 11 章）介绍了对象管理组的软件安全保证体系的协议。首先，介绍了通用事实模型（CFM）方法的细节。然后，描述了业务词汇和业务规则的语义（SBVR）标准。最后，介绍了知识发现元模型（KDM）。读者可以通过阅读规范来了解更多的对象管理组标准。

第四部分（第 12 章）通过一个端到端的案例研究来讲解系统安全保证项目的某些部分，以阐释第 3 章所定义的系统安全保证方法（System Assurance Methodology，ASM）、综合系统模型和系统安全保证案例。

本书还包括一个在线附录[⊖]，详细地介绍了如何使用综合系统模型为安全保证案例收集证据。附录是针对技术人员的，并包括了 KDM Analytics 中 KDM Workbench 工具的截图。这也是没有将该部分材料作为本书主要部分的原因。

本书主要面向以下读者：希望更详尽地理解系统安全保证是什么，如何证明一个系统的安全状况，及如何进行综合安全评价；希望了解基于架构的安全评价过程、建立系统安全保证案例和搜集系统安全证据的安全专业人员。

⊖ 本书英文网站是：www. books. elsevier. com，本书英文书号为 ISBN 978-0-12-381414-2 可上网查询该附录。也可从华章网站下载：www. hzbook. com。——编辑注

安全专业人员在阅读本书后可以熟悉标准的系统安全保证方法。本书还可以引导读者了解对象管理组的知识发现元模型、通用事实模型和相关标准，这有利于建立支持互操作的解决方案，充实网络安全内容，并通过多个工具厂商来形成解决方案。对象管理组软件安全保证体系中越来越多的组件成为开源项目，这对于大学里的安全研究人员、开源软件开发人员具有相当大的吸引力。

本书对安全保证研究实验机构也很有价值，因为本书提供了将多种商业工具整合为一个功能强大的、高度自动化的评价方案的蓝图，在这个整合过程中，知识发现元模型（KDM）和通用事实模型（CFM）发挥了巨大作用。

安全工具厂商也可以从本书学习如何通过简单的导入导出以插入端到端的方案，从而利用安全体系，并扩大产品市场份额。

接受系统安全服务的用户可以从本书受益。除了获得更好、更便宜、更快、更综合的系统安全评估外，还可以了解并非由清晰的、具有说服力的论据所支持的漏洞检测的缺陷。

系统设计相关人员也可以从本书受益，本书可以帮助理解开放标准的、协作式网络安全的架构。这样就可以选择最好的工具来满足要求，并要求厂商开发额外的功能，使得工具开发厂商和安全研究人员能够高效地协作。在网络攻击面前，这对于做好安全工作至关重要。

译 者 序

这不是本轻松的书，无论是对于读者还是译者，都不轻松。最大的难点，不是它谈及的知识，不是它引用的术语，而是它站的角度。不同于其他介绍安全技术的书籍，本书的视野相当开阔，高屋建瓴地指点安全工作可能涉及的每个环节。如果作者的本意是想把安全工作从刺刀见红的战术层面提升到运筹帷幄的战略层面，从各有异同的实践中抽象出广泛适用的理论，那么，他成功地做到了这一点。但是要理解一套理论，显然比掌握一门技术需要更多的思考和精力。

我们学习安全技术，总是爱问一个问题：哪款工具最好用？我们总爱幻想掌握那么一款工具，能够一劳永逸地解决所有问题。市面上大多数安全类书籍也迎合了这一需要，对一款或者一类工具进行介绍，当然，其中也不乏优秀者，字里行间不经意就流露出丰富的实战经验，只是我们读着读着，很容易就只见树木不见森林。

安全技术领域包含了方方面面，仅凭一款工具甚至仅靠一个高手来包打天下都是强人所难，单枪匹马行侠仗义的英雄年代已经过去，分工合作才是这个时代最基本的生存法则。“安全”二字今天已经进化成一条环环相扣互相依存的产业链，可能涉及不同的工具、不同的社区，如何组织它们最快、最好地达到不同的目的，也许才是更需要安全从业人员关注的问题。

本书最大的特点，正是将日常繁琐纷乱的各类安全事务进行概括抽象，根据目的的不同分门别类，又以流程为线索串连起来，让安全人员在跳入茫茫的具体而琐屑的事务海洋之前，能够理清头绪，明确目的，时时刻刻都能回答这样三个简单而为难的问题：“我在做什么”、“我要做什么”以及“我需要做什么”，而不至于在扑面而来的大小事务中迷失了自己，空耗了时间和精力。

本书由莫凡、赵见星、杨勇、莫非共同承担了翻译工作，尽管我们在翻译时字斟句酌，但受限于时间和精力，以及自身的经验和视野，译文难免存在疏漏，恳请广大读者包容和不吝斧正。

序　言

公元20世纪末，《Time-Life》杂志将Johannes Gutenberg发明的印刷机评为第二个千年里最为重要的发明，这说明印刷机比其他发明更深刻地影响了人们的生活。麻醉剂和种痘技术的发明革新了医疗领域，汽车和飞机的发明增加了人们的活动范围，电灯的发明也触发了众多的社会变革，但这些发明都不如印刷机的发明对人类的影响大，因为印刷机的发明促进了文化更广范围地传播。

印刷机中可交换部件这一概念使得印刷出版具有了灵活性，并成为一场具有如此影响力的变革。很久以前，书都是手工抄写得来的，抄写一本书很可能花费一人一年的时间，显然，只有极少的书可以抄写出来并流传下来。Gutenberg并没有发明印刷术或活字印刷，印刷术早在几千年之前就已经出现了，活字印刷也在一千年前就出现了。尽管是他把活字印刷引入西方并为活字印刷进行了机械化生产的改造，但是，他最重要的贡献在于将这些技术组合使用，并对社会的变革产生了深刻的影响。

本书的读者多为计算机从业人员，而计算机在《Time-Life》杂志列表上的排名远低于Gutenberg的印刷机，这一点儿都不奇怪，因为计算机只是在上个千年快结束时才出现的，那时它的影响才刚开始显现。那时我们只经历了计算机对人类产生影响的很小一部分，现在每天对计算机的使用都在迅速地改变着世界。但是，随着计算机快速的发展和革命性的改变，也出现了不少问题，比如：计算机可能被黑客入侵，我们所依赖的、以获得安全和隐私的代码也可能存在缺陷，使得代码并不是按照我们所预期的那样运行。

一些安全专家也在尝试使计算机更为安全。本书作者Nikolai Mansourov博士和Djenana Campara女士对OMG的软件质量保证体系所做的贡献，以及编写本书以便提供支持工作，为一场革命性的变革提供了基础，这场变革会改变软件如何决定和展示。基于这些知识，可以对软件获得更深刻的认识，并标识出潜在的漏洞。

OMG的软件质量保证体系的主要优势是：以很多人开发的、并被广泛认可的标准为基础。这种基于标准的方法使得软件质量保证体系的组件可以与其他组件进行交换，这些交换可能是为了更高效地处理另一种编程语言。这种方法还使得组件之间可以进行交换，这与Gutenberg所发明的印刷机非常类似。人们正是使用这些组件构建了我们当今所使用的大多数产品。同理，由这个软件质量保证体系提供的可交换性，还能够从本质上改变如何在软件系统内标识缺陷并将其作为证据的方法。

软件质量保证（SwA）所基于的标准毫无疑问是至关重要的。贯穿于全书，本书作者详细地阐释了这些根本的标准，以及如何将它们组合起来以形成软件质量保证体系。对他们使计算变得更安全、更可靠方面所做的贡献致以崇高的敬意。

Larry Wagoner博士

目　录

第 1 章

为什么黑客更了解我们的系统

我们生活在充满规则和风险的世界里。

——Clifton A. Ericson Ⅱ，“Hazard Analysis Techniques for System Safety”

纵观历史，每一个技术的进步，必然成为攻击者的目标。

——David Icove，“Computer Crime”

1.1 网络操作的风险

为实现网络空间中的高效操作，在提供灵活的面向服务的用户体验的同时，组织机构需要保持敏捷、可移动和健壮性。这一目标的实现严重依赖于基于 Web 和 Internet 的服务技术，这些技术能达到系统间自动化端到端的无缝信息交换和全自动化、24/7 无人值守操作，使得组织机构和客户的协同工作成为可能。然而，随着信息交换能力的增强，安全、隐私和监管等方面出现了新的问题和挑战。

随着全球范围内的网络互联和服务日趋普及，例如目前大量金融和商业交易都是基于网络的，网络犯罪也发生了显著转变［Icove 1995］。开始由好奇心驱使的、以信息自由为目标的单个黑客行为，向高级的、跨国的、有组织的犯罪转变，这些罪犯通常为了经济利益而进行大规模的在线犯罪活动。在过去的三十年中，黑客已经积累了大量网络攻击的方法。根据澳大利亚议会发起的网络犯罪调查报告［Australia 2010］表明，网络犯罪“已经形成行业规模，成为一个日益重要的全球性社会问题”。

此外，网络战也成为 21 世纪战争不可忽视的战场。新世纪的战场不仅包括玉米田、沙漠、山路和松树林等物理空间，还包括由信息高速公路组成的网络空间，以及背后支持的计算机、移动电话、光纤、双绞线、各种网络设备和电磁频谱［Carr 2010］。这些设备涵盖了国家关键基础设施和企业信息系统，以及所有商业和家用桌面台式机和笔记本电脑。关键基础设施由国家核心产业部门和日常民众所依赖的基础服务设施组成。国家核心产业部门包括化工、电信、银行、金融、能源、农业和食品等；基础服务设施包括水、邮政和运输、电力、公共健康和紧急服务及运输。每个组成部分都极其复杂，并在一定程度上依赖于其他组成部分。

网络空间和物理空间越来越交织在一起，并均由软件控制。每个系统都对安全和保障措施施加影响，并均有其独特设计和独特系列部件组成。此外，它们还包含内在固有的灾害风险。因此，我们经常需要在可能出现的风险和得到的系统效益之间进行权衡。当开发和构建系统时，我们就需要关注如何消除和降低灾害风险。安全服务需要无缝集成到新环境，以协助民用企业管理者和

军事指挥官认识到由 Web 和 Internet 服务活动带来的信息安全威胁、计算残余风险，并采取适当的安全应对措施以维持秩序和控制。一些较小风险可以很容易地接受，而巨大风险需要立即处理。目前，安全问题已经引起世界足够的重视，但随着对 Web 和 Internet 服务的依赖加深，依靠传统的认证过程依然不能缓和所产生的安全问题。虽然，判断信息是否可信仍然很重要，但验证该信息是否存在与威胁相关的活动，已成为必要的检验措施之一。

开发有效的方法，用以验证系统是否如预期运作、信息是否足够可信、是否没有与威胁相关的活动是实现系统安全保证、防止当前和未来攻击的关键。

OCED/APEC 在2008 年报告中提到，OECD 国家和 APEC 经济体中政府、企业和个人对于恶意软件威胁越来越重视。由于政府依靠互联网为公民提供服务，在信息系统安全及恶意人员的网络攻击和渗透方面面临更为复杂的挑战。公众也需要政府能够阻止和保护消费者免受在线威胁的侵害，如身份盗取等。过去五年中，使用恶意软件攻击信息系统的案例急剧增加，如收集信息、窃取金钱和身份、甚至拒绝用户获得基本的电子资源等。值得注意的是，这些案例表明，恶意软件具有干扰大型信息系统运作、修改数据完整性、攻击某些信息系统（主要用于监控和操纵关键基础设施的主要系统）的能力［OECD 2008］。

1.2 黑客屡次攻击成功的原因

黑客似乎更了解我们的系统。这听起来十分奇怪。我们的设计人员、开发人员、实施人员、管理人员和维护人员，不具有“主场优势”吗？但黑客依然能不断发现新的、能够控制我们系统的方法。每周有新的安全事故被报告，而软件厂商则为自己的产品发布补丁来解决安全事故。整个安全行业看起来像是为了赶上黑客，希望“好人”会比“坏人”更快发现漏洞，以便软件开发商能在发生安全事故之前修补系统。

现在我们假设“漏洞”是由特定系统的错误知识组成，这些错误可以使得黑客未经授权访问系统，甚至以更加恶意的方式访问系统。这些错误主要由以下原因造成：人为错误、糟糕的需求规格说明和开发过程、快速变化的技术和对威胁认知的不足，也有些错误是通过供应链故意引入的，并由于不当的开发和获取过程而进入了已交付的系统。相对于传统系统安全工程，业内人士认为在系统生命周期内，一定成本和时间限制下无差错，无故障和无风险的操作通常是不可实现的［ISO15443］。

那么，为什么攻击者比系统开发人员和防御者更了解我们系统呢？因为他们能高效地发现我们的系统信息，并通过其社区进行分发。黑客怎样发现这些信息？黑客坚持不懈地学习我们的系统，并找出新的攻击方法来攻击系统。一些黑客对所攻击的系统具有访问整个系统开发过程的优势，甚至有实际使用系统的信息。黑客研究以非法或合法手段获取的源代码，特别是对基于网络的关键专有系统。同时，黑客也学习机器代码，或者通过交互（该过程无需了解源代码）了解该系统。黑客的优势在于：

- 现实中系统往往基于现有商业系统组件，包括一小部分基本的硬件和软件平台；
- 时间的灵活性。黑客可以在不受时间限制的情况下分析我们的系统，即使该分析过程可能极其耗时；

- 遗留系统的脆弱性。绝大多数系统依然采用的是原始的、缺乏安全预防措施的遗留系统。

然而，使得攻击者更高效的主要原因是广泛的知识共享。这也是防御者重点关注的方面。现在，我们来看看黑客知识共享是如何完成的。

攻击者知识和能力各不相同，每个攻击者都比防御者和系统开发人员更了解我们系统，是比较夸张的说法。在攻击者社区，每个人拥有不同的技能，并扮演不同的角色。这些人包括少数资深的安全研究人员（称为“精英黑客”），以及大量熟练的攻击者（称为“脚本小子”）。然而，攻击者社区作为一个志同道合的群体，在通过计算机交流和社交网络共享知识方面是十分高效的。事实上，早期黑客往往开始于对计算机新技术的热情和网络交流。借此，攻击者能够迅速积累足够的所攻击系统的知识。此外，也有专门人士将攻击理论知识转化为攻击脚本或工具，这使得攻击知识更具实用性，并且这些工具在攻击网络系统的过程中，确实起到了重要作用。而理论知识转化为自动化攻击武器并不需要复杂的技术技能。攻击者不仅分享他们的知识，也分享其工具和“武器”，这些对想发动攻击的个人来说都是可以获知的。这导致一个高效攻击系统的诞生，它放大了少数资深黑客的能力，并培养了大量虽缺少专业技能，但有足够犯罪动机的攻击者。黑客所做的事情，可能并不具备系统性，但他们成功地将黑客知识产业化。

大部分现代攻击武器都是恶意软件。援引早些时候的 OECD 报告［OECD 2008］，恶意软件是插入信息系统的软件，用于影响本系统或其他系统，或使系统无法按使用者的意图来使用。恶意软件可以获得信息系统远程访问权限、在用户没有察觉的状态下记录并向第三方发送数据、隐藏信息系统被入侵事实、取消某些安全措施、破坏信息系统，甚至影响数据和系统的完整性。恶意软件依据不同类型，通常分为病毒、蠕虫、木马、后门程序、按键记录程序、Rootkits 或间谍软件。恶意软件通过缩小发现软件产品漏洞和利用该漏洞的时间间隔，提高了网络攻击的可重复性，降低了当前安全技术和其他防御措施的有效性。

防御者社区成员所掌握的技能也有很大的差别，从精英安全研究人员（有时很难将其与精英黑客之间进行区别）到普通家用电脑使用者对安全技能知识的掌握有明显差距。然而，防御者社区由于为保持竞争优势、扩张市场空间、增强产品功能等原因设置了各种障碍，无法有效进行知识共享。

1.3 网络系统防御的挑战

网络系统安全防御涉及安全风险分析、漏洞管理、增加安全防御措施和事故响应。理解这些内容的基础涉及以下知识：1）什么是安全防御；2）安全防御对象是什么；3）需要防护的漏洞是什么；4）需要采取怎样的安全防御措施。安全防御贯穿系统的整个生命周期。采用更好的安全工程以开发更加安全的系统是长期战略，而网络防御社区也需要对已经存在的系统进行防御，增加一些防御措施，包括对当前系统打补丁、增加安全组件、完善安全规程、改善配置策略和安全培训等。

1.3.1 理解和评估安全风险的难点

多年来行业和防御机构已经开发出评估和度量系统安全态势的方法，这些方法可以分为内部

方法和第三方方法。虽然对评估内容和期望结果的理解和记录已经取得重要进展（例如通用标准[ISO15408]），但缺乏有效的方式将这些评估方法组合起来应用到实际工作中。目前，希望（或有希望）认证其软件系统安全性的厂商，受到“越少越好”口号影响，即以较少的安全评估需求，增加其通过评估的机会，并且整个评估过程也更快、成本更低。这也是某些高健壮性需求的系统，没有被评估的一个关键原因。

理解和评估网络中风险是一件非常有挑战性的任务。曾经整个行业都在试图破解这一挑战，也正是由于这个原因，通过检查整个生命周期中系统和开发管理，以及系统的复杂性，来理解这一挑战的影响是十分重要的。同时，检查关键开发趋势及其管理手段，可以帮助理解安全评估方法所必须覆盖的领域。

随着新技术和新特性的引入，导致软件快速演化，对复杂网络系统和自治软件组件引入更高层次的复杂性，都使得系统日趋复杂。

1.3.2 复杂的供应链

目前，大多数软件开发的发展趋势，都使得对系统安全态势的评估更加困难，使得用于开发和获取软件的过程和软件存在弱点。一些主要软件开发趋势包括：

1）由于严重依赖 COTS 和开源产品，软件开发更受全球化影响。这主要是由于现代软件开发和供应链支持逐渐向世界范围扩展。快速生产低成本系统的趋势最初看起来很好，对于部署了这些系统的商业应用来说，很快将面临一场噩梦。由于使用外包，来自国外的开发人员和未经评估的国内供应商，或者使用一些廉价的、仅关注功能而不考虑抗攻击性的供应链组件，这些都增加了暴露漏洞和遭受攻击的风险。由于这些原因，评估系统安全态势比评估软件应用更难，这需要评估软件供应链、开发过程、开发团队的资质，以解决针对颠覆软件供应链和内部人士攻击的不断增长的关注。

2）升级遗留系统。大量有用的、已部署的、可运行的遗留系统是在较低安全需求下开发完成的，但这些旧的软件系统依然有巨大的商业价值。因此，通过维护来延长其使用寿命是很有必要的，同时还有必要提高其适应新市场需求和政府法规的能力。现有系统变得更加庞大和复杂，并侵蚀原有的设计原则，这影响了对系统的理解，破坏了系统的完整性，降低了产品的可维护性。随着时间推移，问题也更加突出，系统更容易发生缺陷，升级也更加困难。在这种情况下，任何试图增加安全代码来解决安全问题的操作，都将导致修改大量数据，并可能触发组织业务新的不可预见风险。但这些昂贵的高风险方法从未付诸实施，反而是经常见到引入很多快速修复程序，其中包含了很多捷径，并且很少检查其中可能存在的薄弱环节。为回应对安全的恐慌，新功能通常“塞进”到现有架构，这也使得系统安全大打折扣。所以最大的问题是：“怎样评估这样一个系统的安全态势?”

3）另一个主要趋势是加速迁移到网络环境和面向服务的架构。因此，评估必须保证软件组件在没有监管情况下的交互是可信的。通常系统安全保证涉及以下几点：

- 不遵守开放标准和协议。复杂的协议被认为是过分的和多余的，因此，改变一些步骤在当时看起来很好，虽然可能不影响正常运作，但肯定会危及协议提供的安全保障。
- Web 服务配置文件导致的漏洞。Web 服务设计目的是为应用平台提供更多的灵活性。但

是，由于这也可能由于复杂的服务文件配置，导致服务中存在漏洞，给系统安全带来新问题。

- 软件漏洞。大多数情况下，原先不为网络环境设计的遗留软件，迁移到以网络为中心的系统时，很容易出现大量的代码漏洞，这使得整个系统更加脆弱。

1.3.3 复杂的系统集成

“特定系统的复杂性基本上和根据部分系统性质预测整个系统性质一样艰难。”［Weaver 1948］。这极大地影响了评估方法，并应该作为风险评估和管理中的重要组成部分。

我们都认为，当前软件系统将变得更加庞大和复杂。软件开发项目很少从零开始启动，现今的大多数代码都基于现有功能的维护和增强。通常，在现有代码库上增加的新特性或功能，本身就很复杂庞大，它不可避免地与原设计有冲突，引入错误信息。此外，市场整合过程中，对于网络系统中组件的合并和调用，也是一种挑战。因为这使得原有系统更加庞大、复杂且更难以理解。系统结构包括相互连接的软件组件，这些组件通常用不同的方法、技术，并在不同的制约因素和假设条件下开发。同时，这类系统的文档没有及时更新，而且可用信息往往基于判断且难以获得。唯一可信且及时更新的内容是源代码本身。然而，理解系统源代码也十分困难，因为系统经常不加控制地使用多种编程语言，不同的开发人员编码风格也不一样，这也会降低对软件架构控制的力度，导致对初始系统架构理念的侵蚀，更增加了系统的复杂性。这是评估工作应当揭露的重大潜在安全缺陷的根源。

1.3.4 系统评估方法的局限性

大多数系统评估主要侧重于评估开发过程和产品文档，而较少关注正式构件。并且存在侥幸心理，很少进行正式的安全分析。开发过程提供了开发系统的结构化方法，而正因为此，极大地影响系统安全态势。这也为证实系统安全提供了切入点。以下介绍一些主要评估方法：

- **获取非正式评估信息**：评估信息通常通过访谈和文件样本收集来获得，这些资料提供的结果是比较客观的，故不可能重复。
- **非最新文档**：产品文档，即使保持最新，通常是手动生成并进行人工审核。因此，这类资料带有主观性，并不能完全反映已实现系统构件的属性。
- **获取的信息和系统构件之间缺乏可追踪性**：获得的信息通常不能很好地与系统构件进行“关联”。因而，不能找到系统中存在的漏洞，也不能为安全态势的改善提供建议。

系统开发通常遵循以下开发模型：瀑布模型、迭代模型（例如敏捷、快速应用开发（RAD）、能力成熟度模型（CMM）、模型驱动架构（MDA））和一些自定义过程。这些模型除了提供结构化方式来开发产品外，也提供建立多阶段可追踪性的框架，该框架是通过将高层次策略与目标，需求、设计规格说明（文档或原型形式）以及系统构件实现相结合来实现的。一旦产品开发完成，该框架就以在需求和相应实现之间建立跟踪痕迹的形式融入项目。但正因为其可追踪性，阻碍了自动化评估方法的开展。因此对系统构件的正式分析，既不能整个忽略，也不能在特定场合下用代价高昂的手工开发应用来代替。这两种方法都涉及软件供应链和开发团队构成。

最近，随着白盒漏洞测试和黑盒漏洞测试领域新技术的发展，使得对软件和网络系统进行自