高·等·院·校·信·息·安·全·专·业·系·列·教·材

中国计算机学会教育专业委员会与清华大学出版社联合组织编写

名誉主编：何德全　编委会主任：肖国镇

Computer Security:Art and Science

计算机安全 艺术与科学

Matt Bishop

http://www.tup.com.cn

清华大学出版社

北京市版权局著作权合同登记号　图字 01-2003-8075

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

计算机安全：艺术与科学= Computer Security: Art and Science /（美）毕晓普（Bishop, M.）著. —影印本. —北京：清华大学出版社，2004
（高等院校信息安全专业系列教材）
ISBN 7-302-08341-X

Ⅰ. 计… Ⅱ. 毕… Ⅲ. 电子计算机－安全技术－高等学校－教材－英文 Ⅳ. TP309

中国版本图书馆 CIP 数据核字（2004）第 025085 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) 6277 0175　　**客户服务**：(010) 6277 6969
策划编辑：张　民
封面设计：孟繁聪
印 刷 者：北京密云胶印厂
装 订 者：北京市密云县京文制本装订厂
发 行 者：新华书店总店北京发行所
开　　本：185×230　**印张**：70.5
版　　次：2004 年 5 月第 1 版　2004 年 5 月第 1 次印刷
书　　号：ISBN 7-302-08341-X/TP・6013
印　　数：1~5000
定　　价：*96.00* 元

高等院校信息安全专业系列教材

编审委员会

序

在社会信息化的进程中，信息已成为社会发展的重要资源，信息安全也成为21世纪国际竞争的重要战场。为了保护国家的政治利益和经济利益，各国政府都非常重视信息和网络安全，信息安全已成为一个世纪性、全球性的研究课题。

我国的信息安全事业正在蓬勃发展，国家领导高度重视，各部门通力合作、统筹规划，大大加快了我国信息安全产业发展的步伐。随着信息安全产业的快速发展，社会对信息安全人才的需求在不断增加，在高等教育领域大力推进信息安全的专业化教育，将是国家在信息安全领域掌握自主权、占领先机的重要举措。

目前，许多大学和科研院所已设立了信息安全专业或是开设了相关课程。很高兴中国计算机学会教育专业委员会和清华大学出版社在近期联合组织了一系列信息安全专业的研讨活动。他们以严谨负责的态度，认真组织全国各高校和科研院所的专家、学者，共同研讨信息安全专业的教育方法和课程体系，并在进行大量前瞻性研究工作的基础上，启动了“高等院校信息安全专业系列教材”的编写工作。这套教材将是我国信息安全专业的第一套完整、权威的教材，相信可以对全国的高等院校信息安全专业的建设起到很好的促进作用。

希望中国计算机学会教育专业委员会和清华大学出版社能够将这个研究课题一直做下去，也希望这套教材能够取得成功并不断完善，以促进各高等院校培养出更多、更好的信息安全专门人才，为我国的信息安全事业做出更大的贡献。

何德全

中国工程院院士

高等院校信息安全专业系列教材编审委员会名誉主编

2003年7月于北京

出版说明

21世纪是信息时代，信息已成为社会发展的重要战略资源，社会的信息化已成为当今世界发展的潮流和核心，而信息安全在信息社会中将扮演极为重要的角色，它直接关系到国家安全、企业经营和人们的日常生活。随着信息安全产业的快速发展，国家对信息安全人才的需求量不断增加，但目前信息安全人才极度匮乏，远远不能满足金融、商业、公安、军事和政府等部门的需求。要解决供需矛盾，必须加快信息安全人才的培养，以满足社会的需求。为此，教育部继2001年批准在武汉大学开设信息安全本科专业之后，又批准了多所高等院校设立信息安全本科专业，而且许多高校和科研院所已设立了信息安全方向的具有硕士和博士学位授予权的学科点。

信息安全是计算机、通信工程、数学等领域的交叉学科，对于这一新兴学科的培养模式和课程设置，各高校普遍缺乏经验，因此中国计算机学会教育专业委员会和清华大学出版社联合主办了“信息安全专业教育教学研讨会”等一系列研讨活动，并成立了“高等院校信息安全专业系列教材”编审委员会，由我国信息安全领域著名专家何德全院士担任名誉主编，著名学者肖国镇教授担任编委会主任，共同指导“高等院校信息安全专业系列教材”的编写工作。编委会本着研究先行的指导原则，认真研讨国内外高等院校信息安全专业的教学体系和课程设置，进行了大量前瞻性的研究工作，而且这种研究工作将随着我国信息安全专业的发展不断深入。经过编委会全体委员及相关专家的推荐和审定，确定了编写教材的作者，这些作者绝大多数都是既在本专业领域有深厚的学术造诣，又在教学第一线有丰富的教学经验的学者、专家。

本系列教材是我国第一套专门针对信息安全专业的教材，其特点是：

① 体系完整，结构合理，内容先进。

② 适应面广，能够满足信息安全、计算机、通信工程等相关专业对信息安全领域课程的教材要求。

③ 立体配套，除主教材外，还配有多媒体电子教案、习题与实验指导等。

④ 版本更新及时，紧跟科学技术的新发展。

为了保证出版质量，我们坚持宁缺毋滥的原则，成熟一本，出版一本，并保持不断更新，力求将我国信息安全领域教育、科研的最新成果和成熟经验反映到教材中来。在全力做好本版教材，满足学生用书的基础上，还经由专家的推荐和审定，遴选了一批国外信息安全领域优秀的教材加入到本系列教材中，以进一步满足大家对外版书的需求。热切期望广大教师和科研工作者加入我们的队伍，同时也欢迎广大读者提出宝贵意见，以便我们对本系列教材的组织、编写与出版工作不断改进，为我国信息安全专业的教材建设与人才培养做出更大的贡献。

我们的E-mail地址是：zhangm@tup.tsinghua.edu.cn；联系人：张民。

中国计算机学会教育专业委员会

清华大学出版社

2003年7月

本书序

计算机的应用已经从20世纪80年代中期的单机模式发展到了目前遍布世界各地的Internet网络。计算机安全关注的重点也从最初单机环境下简单的文件/进程安全保护转移到了目前网络环境下系统的整体安全。计算机网络的迅猛发展已经使得计算机安全问题变得越来越复杂。计算机安全为理论研究人员和管理技术人员都提出了严峻的挑战。

计算机安全发展到今天,已不仅仅是一门科学,它同时也是一门艺术。说它是一门科学是因为它有坚实的数学基础,有严格的分析证明方法;它利用严格的归纳、推理方法揭示了各种潜在的规则,并把这些规则应用到了实际应用中。它同时也是一门艺术,它为设计者、研究者提供了自由发挥的空间。计算机安全问题的复杂性造成其不可能有标准、统一的解决办法,对于同一个目标系统,两个不同设计者根据自己对计算机安全的不同理解,可能提出完全不同的解决方案。在方案中,设计者除了实现保护系统安全这一关键目标之外,还可体现自己不同的个性特点。这也是艺术创作的基本特点之一。

《计算机安全:艺术与科学》一书把计算机系统、网络、人为因素和密码学等概念融为一体,有效地阐明了计算机安全不仅是一门科学,而且也是一门艺术;阐述了从理论到实践和从实践到理论相互促进的重要性;强调了计算机安全和密码学的差异性,着重论述了密码学在计算机安全中的作用和局限性,阐明了密码学和计算机安全的相互关系。本书还详尽地介绍了计算机安全领域最基本和最普遍的工作,包括计算机安全的本质和面临的挑战,策略与安全的关系,密码学的角色与应用,策略实现机制,保障技术和方法学,脆弱性分析和入侵检测等。论述了不同的策略模型,并提出了执行这些策略的机制。并用大量的实例从网络、系统、用户和程序几个方面说明了如何应用书中所讨论的一些原则来解决实际安全问题。

对于管理技术人员来说,不仅需要掌握各种实用技术,还需要对理论有一定的了解,弄清理论证明了哪些是可行的,哪些是完全不可行的,本书对这两方面都做了很好的论述;对于理论研究人员来说,本书则可以帮助你进一步认识到理论在现实应用中存在什么样的限制,理论研究成果如何转化为实用工具和方法,并在现实中得到应用。因此,我认为本书对高等院校、科研院所的师生,对维护信息安全的管理人员、技术开发人员和研究人员都是一本不可多得的好书。

并且,本书还具有以下特点:

(1) 内容全面。本书全面论述了计算机安全理论和核心技术,包括基础理论、安全策略、技术实现、保障和实践等多方面的内容;在内容组织上,由浅入深,并充分考虑了各章节内容的相互依赖性;

(2) 习题丰富,每章后面作者都配有大量的习题。通过演练和回答每章后面精心安排的习题,读者可以迅速并熟练地掌握每章的基本内容和方法。这些习题不仅有利于帮助读者加深对每章内容的理解,也是对每章重点内容的再次总结;

(3) 浅显易懂,深入浅出。作者凭着自己对计算机安全的深入研究,浅显易懂地讲述了各种理

论和技术，同时还为有兴趣的读者提供了进一步深入研究的方向，推荐了一些经典的文献读物，对于一个感兴趣的研究人员来说，这是难得的指导意见。

本书无论是在内容上还是组织上，都堪称计算机安全领域的经典之作，对高校师生、研究人员和管理技术人员都有重要的指导作用。

2004 年 3 月于北京

冯登国，中国科学院软件所研究员、博士生导师，国家信息化专家咨询委员会专家，国家 863 计划信息安全技术主题专家组组长，信息安全国家重点实验室主任，国家计算机网络入侵防范中心主任。

Preface

HORTENSIO: Madam, before you touch the instrument
To learn the order of my fingering,
I must begin with rudiments of art
To teach you gamouth in a briefer sort,
More pleasant, pithy and effectual,
Than hath been taught by any of my trade;
And there it is in writing, fairly drawn.
—*The Taming of the Shrew*, III, i, 62–68.

On September 11, 2001, terrorists seized control of four airplanes. Three were flown into buildings, and a fourth crashed, with catastrophic loss of life. In the aftermath, the security and reliability of many aspects of society drew renewed scrutiny. One of these aspects was the widespread use of computers and their interconnecting networks.

The issue is not new. In 1988, approximately 5,000 computers throughout the Internet were rendered unusable within 4 hours by a program called a *worm* [432].[1] While the spread, and the effects, of this program alarmed computer scientists, most people were not worried because the worm did not affect their lives or their ability to do their jobs. In 1993, more users of computer systems were alerted to such dangers when a set of programs called *sniffers* were placed on many computers run by network service providers and recorded login names and passwords [374].

After an attack on Tsutomu Shimomura's computer system, and the fascinating way Shimomura followed the attacker's trail, which led to his arrest [914], the public's interest and apprehension were finally aroused. Computers were now vulnerable. Their once reassuring protections were now viewed as flimsy.

Several films explored these concerns. Movies such as *War Games* and *Hackers* provided images of people who can, at will, wander throughout computers and networks, maliciously or frivolously corrupting or destroying information it may have taken millions of dollars to amass. (Reality intruded on *Hackers* when the World Wide Web page set up by MGM/United Artists was quickly altered to present an irreverent commentary on the movie and to suggest that viewers see *The Net*

[1] Section 22.4 discusses computer worms.

instead. Paramount Pictures denied doing this [448].) Another film, *Sneakers*, presented a picture of those who test the security of computer (and other) systems for their owners and for the government.

Goals

This book has three goals. The first is to show the importance of theory to practice and of practice to theory. All too often, practitioners regard theory as irrelevant and theoreticians think of practice as trivial. In reality, theory and practice are symbiotic. For example, the theory of covert channels, in which the goal is to limit the ability of processes to communicate through shared resources, provides a mechanism for evaluating the effectiveness of mechanisms that confine processes, such as sandboxes and firewalls. Similarly, business practices in the commercial world led to the development of several security policy models such as the Clark-Wilson model and the Chinese Wall model. These models in turn help the designers of security policies better understand and evaluate the mechanisms and procedures needed to secure their sites.

The second goal is to emphasize that computer security and cryptography are different. Although cryptography is an essential component of computer security, it is by no means the only component. Cryptography provides a mechanism for performing specific functions, such as preventing unauthorized people from reading and altering messages on a network. However, unless developers understand the context in which they are using cryptography, and unless the assumptions underlying the protocol and the cryptographic mechanisms apply to the context, the cryptography may not add to the security of the system. The canonical example is the use of cryptography to secure communications between two low-security systems. If only trusted users can access the two systems, cryptography protects messages in transit. But if untrusted users can access either system (through authorized accounts or, more likely, by breaking in), the cryptography is not sufficient to protect the messages. The attackers can read the messages at either endpoint.

The third goal is to demonstrate that computer security is not just a science but also an art. It is an art because no system can be considered secure without an examination of how it is to be used. The definition of a "secure computer" necessitates a statement of requirements and an expression of those requirements in the form of authorized actions and authorized users. (A computer engaged in work at a university may be considered "secure" for the purposes of the work done at the university. When moved to a military installation, that same system may not provide sufficient control to be deemed "secure" for the purposes of the work done at that installation.) How will people, as well as other computers, interact with the computer system? How clear and restrictive an interface can a designer create without rendering the system unusable while trying to prevent unauthorized use or access to the data or resources on the system?

Just as an artist paints his view of the world onto canvas, so does a designer of security features articulate his view of the world of human/machine interaction in the security policy and mechanisms of the system. Two designers may use entirely different designs to achieve the same creation, just as two artists may use different subjects to achieve the same concept.

Computer security is also a science. Its theory is based on mathematical constructions, analyses, and proofs. Its systems are built in accordance with the accepted practices of engineering. It uses inductive and deductive reasoning to examine the security of systems from key axioms and to discover underlying principles. These scientific principles can then be applied to untraditional situations and new theories, policies, and mechanisms.

Philosophy

Key to understanding the problems that exist in computer security is a recognition that the problems are not new. They are old problems, dating from the beginning of computer security (and, in fact, arising from parallel problems in the noncomputer world). But the locus has changed as the field of computing has changed. Before the mid-1980s, mainframe and mid-level computers dominated the market, and computer security problems and solutions were phrased in terms of securing files or processes on a single system. With the rise of networking and the Internet, the arena has changed. Workstations and servers, and the networking infrastructure that connects them, now dominate the market. Computer security problems and solutions now focus on a networked environment. However, if the workstations and servers, and the supporting network infrastructure, are viewed as a single system, the models, theories, and problem statements developed for systems before the mid-1980s apply equally well to current systems.

As an example, consider the issue of assurance. In the early period, assurance arose in several ways: formal methods and proofs of correctness, validation of policy to requirements, and acquisition of data and programs from trusted sources, to name a few. Those providing assurance analyzed a single system, the code on it, and the sources (vendors and users) from which the code could be acquired to ensure that either the sources could be trusted or the programs could be confined adequately to do minimal damage. In the later period, the same basic principles and techniques apply, except that the scope of some has been greatly expanded (from a single system and a small set of vendors to the world-wide Internet). The work on proof-carrying code, an exciting development in which the proof that a downloadable program module satisfies a stated policy is incorporated into the program itself,[2] is an example of this expansion. It extends the notion of a proof of consistency with a stated policy. It advances the technology of the earlier period into the later period. But in order to

[2] Section 22.7.5.1 discusses proof-carrying code.

understand it properly, one must understand the ideas underlying the concept of proof-carrying code, and these ideas lie in the earlier period.

As another example, consider Saltzer and Schroeder's principles of secure design.[3] Enunciated in 1975, they promote simplicity, confinement, and understanding. When security mechanisms grow too complex, attackers can evade or bypass them. Many programmers and vendors are learning this when attackers break into their systems and servers. The argument that the principles are old, and somehow outdated, rings hollow when the result of their violation is a nonsecure system.

The work from the earlier period is sometimes cast in terms of systems that no longer exist and that differ in many ways from modern systems. This does not vitiate the ideas and concepts, which also underlie the work done today. Once these ideas and concepts are properly understood, applying them in a multiplicity of environments becomes possible. Furthermore, the current mechanisms and technologies will become obsolete and of historical interest themselves as new forms of computing arise, but the underlying principles will live on, to underlie the next generation—indeed the next era—of computing.

The philosophy of this book is that certain key concepts underlie all of computer security, and that the study of all parts of computer security enriches the understanding of all parts. Moreover, critical to an understanding of the applications of security-related technologies and methodologies is an understanding of the theory underlying those applications.

Advances in the theory of computer protection have illuminated the foundations of security systems. Issues of abstract modeling, and modeling to meet specific environments, lead to systems designed to achieve a specific and rewarding goal. Theorems about composability of policies[4] and the undecidability of the general security question[5] have indicated the limits of what can be done. Much work and effort are continuing to extend the borders of those limits.

Application of these results has improved the quality of the security of the systems being protected. However, the issue is how compatibly the assumptions of the model (and theory) conform to the environment to which the theory is applied. Although our knowledge of how to apply these abstractions is continually increasing, we still have difficulty correctly transposing the relevant information from a realistic setting to one in which analyses can then proceed. Such abstraction often eliminates vital information. The omitted data may pertain to security in nonobvious ways. Without this information, the analysis is flawed.

The practitioner needs to know both the theoretical and practical aspects of the art and science of computer security. The theory demonstrates what is possible. The practical makes known what is feasible. The theoretician needs to understand the constraints under which these theories are used, how their results are translated into practical tools and methods, and how realistic are the assumptions underlying the theories. *Computer Security: Art and Science* tries to meet these needs.

---

[3] Chapter 13 discusses these principles.

[4] See Chapter 8, "Noninterference and Policy Composition."

[5] See Section 3.2, "Basic Results."

Unfortunately, no single work can cover all aspects of computer security, so this book focuses on those parts that are, in the author's opinion, most fundamental and most pervasive. The mechanisms exemplify the applications of these principles.

Organization

The organization of this book reflects its philosophy. It begins with mathematical fundamentals and principles that provide boundaries within which security can be modeled and analyzed effectively. The mathematics provides a framework for expressing and analyzing the requirements of the security of a system. These policies constrain what is allowed and what is not allowed. Mechanisms provide the ability to implement these policies. The degree to which the mechanisms correctly implement the policies, and indeed the degree to which the policies themselves meet the requirements of the organizations using the system, are questions of assurance. Exploiting failures in policy, in implementation, and in assurance comes next, as well as mechanisms for providing information on the attack. The book concludes with the applications of both theory and policy focused on realistic situations. This natural progression emphasizes the development and application of the principles existent in computer security.

Part 1, "Introduction," describes what computer security is all about and explores the problems and challenges to be faced. It sets the context for the remainder of the book.

Part 2, "Foundations," deals with basic questions such as how "security" can be clearly and functionally defined, whether or not it is realistic, and whether or not it is decidable. If it is decidable, under what conditions is it decidable, and if not, how must the definition be bounded in order to make it decidable?

Part 3, "Policy," probes the relationship between policy and security. The definition of "security" depends on policy. In Part 3 we examine several types of policies, including the ever-present fundamental questions of trust, analysis of policies, and the use of policies to constrain operations and transitions.

Part 4, "Implementation I: Cryptography," discusses cryptography and its role in security. It focuses on applications and discusses issues such as key management and escrow, key distribution, and how cryptosystems are used in networks. A quick study of authentication completes Part 4.

Part 5, "Implementation II: Systems," considers how to implement the requirements imposed by policies using system-oriented techniques. Certain design principles are fundamental to effective security mechanisms. Policies define who can act and how they can act, and so identity is a critical aspect of implementation. Mechanisms implementing access control and flow control enforce various aspects of policies.

Part 6, "Assurance," presents methodologies and technologies for ascertaining how well a system, or a product, meets its goals. After setting the background, to explain exactly what "assurance" is, the art of building systems to meet varying levels

of assurance is discussed. Formal verification methods play a role. Part 6 shows how the progression of standards has enhanced our understanding of assurance techniques.

Part 7, "Special Topics," discusses some miscellaneous aspects of computer security. Malicious logic thwarts many mechanisms. Despite our best efforts at high assurance, systems today are replete with vulnerabilities. Why? How can a system be analyzed to detect vulnerabilities? What models might help us improve the state of the art? Given these security holes, how can we detect attackers who exploit them? A discussion of auditing flows naturally into a discussion of intrusion detection—a detection method for such attacks.

Part 8, "Practicum," presents examples of how to apply the principles discussed throughout the book. It begins with networks and proceeds to systems, users, and programs. Each chapter states a desired policy and shows how to translate that policy into a set of mechanisms and procedures that support the policy. Part 8 tries to demonstrate that the material covered elsewhere can be, and should be, used in practice.

Each chapter in this book ends with a summary, descriptions of some research issues, and some suggestions for further reading. The summary highlights the important ideas in the chapter. The research issues are current "hot topics" or are topics that may prove to be fertile ground for advancing the state of the art and science of computer security. Interested readers who wish to pursue the topics in any chapter in more depth can go to some of the suggested readings. They expand on the material in the chapter or present other interesting avenues.

Roadmap

This book is both a reference book and a textbook. Its audience is undergraduate and graduate students as well as practitioners. This section offers some suggestions on approaching the book.

Dependencies

Chapter 1 is fundamental to the rest of the book and should be read first. After that, however, the reader need not follow the chapters in order. Some of the dependencies among chapters are as follows.

Chapter 3 depends on Chapter 2 and requires a fair degree of mathematical maturity. Chapter 2, on the other hand, does not. The material in Chapter 3 is for the most part not used elsewhere (although the existence of the first section's key result, the undecidability theorem, is mentioned repeatedly). It can be safely skipped if the interests of the reader lie elsewhere.

The chapters in Part 3 build on one another. The formalisms in Chapter 5 are called on in Chapters 19 and 20, but nowhere else. Unless the reader intends to delve into the sections on theorem proving and formal mappings, the formalisms may be

skipped. The material in Chapter 8 requires a degree of mathematical maturity, and this material is used sparingly elsewhere. Like Chapter 3, Chapter 8 can be skipped by the reader whose interests lie elsewhere.

Chapters 9, 10, and 11 also build on one another in order. A reader who has encountered basic cryptography will have an easier time with the material than one who has not, but the chapters do not demand the level of mathematical experience that Chapters 3 and 8 require. Chapter 12 does not require material from Chapter 10 or Chapter 11, but it does require material from Chapter 9.

Chapter 13 is required for all of Part 5. A reader who has studied operating systems at the undergraduate level will have no trouble with Chapter 15. Chapter 14 uses the material in Chapter 11; Chapter 16 builds on material in Chapters 5, 13, and 15; and Chapter 17 uses material in Chapters 4, 13, and 16.

Chapter 18 relies on information in Chapter 4. Chapter 19 builds on Chapters 5, 13, 15, and 18. Chapter 20 presents highly mathematical concepts and uses material from Chapters 18 and 19. Chapter 21 is based on material in Chapters 5, 18, and 19; it does not require Chapter 20. For all of Part 5, a knowledge of software engineering is very helpful.

Chapter 22 draws on ideas and information in Chapters 5, 6, 9, 13, 15, and 17 (and for Section 22.6, the reader should read Section 3.1). Chapter 23 is self-contained, although it implicitly uses many ideas from assurance. It also assumes a good working knowledge of compilers, operating systems, and in some cases networks. Many of the flaws are drawn from versions of the UNIX operating system, or from Windows systems, and so a working knowledge of either or both systems will make some of the material easier to understand. Chapter 24 uses information from Chapter 4, and Chapter 25 uses material from Chapter 24.

The practicum chapters are self-contained and do not require any material beyond Chapter 1. However, they point out relevant material in other sections that augments the information and (we hope) the reader's understanding of that information.

Background

The material in this book is at the advanced undergraduate level. Throughout, we assume that the reader is familiar with the basics of compilers and computer architecture (such as the use of the program stack) and operating systems. The reader should also be comfortable with modular arithmetic (for the material on cryptography). Some material, such as that on formal methods (Chapter 20) and the mathematical theory of computer security (Chapter 3 and the formal presentation of policy models), requires considerable mathematical maturity. Other specific recommended background is presented in the preceding section. Part 9, "End Matter," contains material that will be helpful to readers with backgrounds that lack some of the recommended material.

Examples are drawn from many systems. Many come from the UNIX operating system or variations of it (such as Linux). Others come from the Windows family of systems. Familiarity with these systems will help the reader understand many examples easily and quickly.

Undergraduate Level

An undergraduate class typically focuses on applications of theory and how students can use the material. The specific arrangement and selection of material depends on the focus of the class, but all classes should cover some basic material—notably that in Chapters 1, 9, and 13, as well as the notion of an access control matrix, which is discussed in Sections 2.1 and 2.2.

Presentation of real problems and solutions often engages undergraduate students more effecively than presentation of abstractions. The special topics and the practicum provide a wealth of practical problems and ways to deal with them. This leads naturally to the deeper issues of policy, cryptography, noncryptographic mechanisms, and assurance. The following are sections appropriate for nonmathematical undergraduate courses in these topics.

- *Policy:* Sections 4.1 through 4.4 describe the notion of policy. The instructor should select one or two examples from Sections 5.1, 5.2.1, 6.2, 6.4, 7.1.1, and 7.2, which describe several policy models informally. Section 7.4 discusses role-based access control.
- *Cryptography:* Key distribution is discussed in Sections 10.1 and 10.2, and a common form of public key infrastructures (called PKIs) is discussed in Section 10.4.2. Section 11.1 points out common errors in using cryptography. Section 11.3 shows how cryptography is used in networks, and the instructor should use one of the protocols in Section 11.4 as an example. Chapter 12 offers a look at various forms of authentication, including noncryptographic methods.
- *Noncryptographic mechanisms:* Identity is the basis for many access control mechanisms. Sections 14.1 through 14.4 discuss identity on a system, and Section 14.6 discusses identity and anonymity on the Web. Sections 15.1 and 15.2 explore two mechanisms for controlling access to files, and Section 15.4 discusses the ring-based mechanism underlying the notion of multiple levels of privilege. If desired, the instructor can cover sandboxes by using Sections 17.1 and 17.2, but because Section 17.2 uses material from Sections 4.5 and 4.5.1, the instructor will need to go over those sections as well.
- *Assurance:* Chapter 18 provides a basic introduction to the often overlooked topic of assurance.

Graduate Level

A typical introductory graduate class can focus more deeply on the subject than can an undergraduate class. Like an undergraduate class, a graduate class should cover Chapters 1, 9, and 13. Also important are the undecidability results in Sections 3.1 and 3.2, which require that Chapter 2 be covered. Beyond that, the instructor can

choose from a variety of topics and present them to whatever depth is appropriate. The following are sections suitable for graduate study.

- *Policy models:* Part 3 covers many common policy models both informally and formally. The formal description is much easier to understand once the informal description is understood, so in all cases both should be covered. The controversy in Section 5.4 is particularly illuminating to students who have not considered the role of policy and the nature of a policy. Chapter 8 is a highly formal discussion of the foundations of policy and is appropriate for students with experience in formal mathematics. Students without such a background will find it quite difficult.
- *Cryptography:* Part 4 focuses on the applications of cryptography, not on cryptography's mathematical underpinnings.[6] It discusses areas of interest critical to the use of cryptography, such as key management and some basic cryptographic protocols used in networking.
- *Noncryptographic mechanisms:* Issues of identity and certification are complex and generally poorly understood. Section 14.5 covers these problems. Combining this with the discussion of identity on the Web (Section 14.6) raises issues of trust and naming. Chapters 16 and 17 explore issues of information flow and confining that flow.
- *Assurance:* Traditionally, assurance is taught as formal methods, and Chapter 20 serves this purpose. In practice, however, assurance is more often accomplished by using structured processes and techniques and informal but rigorous arguments of justification, mappings, and analysis. Chapter 19 emphasizes these topics. Chapter 21 discusses evaluation standards and relies heavily on the material in Chapters 18 and 19 and some of the ideas in Chapter 20.
- *Miscellaneous Topics:* Section 22.6 presents a proof that the generic problem of determining if a generic program is a computer virus is in fact undecidable. The theory of penetration studies in Section 23.2, and the more formal approach in Section 23.5, illuminate the analysis of systems for vulnerabilities. If the instructor chooses to cover intrusion detection (Chapter 25) in depth, it should be understood that this discussion draws heavily on the material on auditing (Chapter 24).
- *Practicum:* The practicum (Part 8) ties the material in the earlier part of the book to real-world examples and emphasizes the applications of the theory and methodologies discussed earlier.

[6] The interested reader will find a number of books covering aspects of this subject [240, 590, 695, 702, 888, 897, 998].

Practitioners

Practitioners in the field of computer security will find much to interest them. The table of contents and the index will help them locate specific topics. A more general approach is to start with Chapter 1 and then proceed to Part 8, the practicum. Each chapter has references to other sections of the text that explain the underpinnings of the material. This will lead the reader to a deeper understanding of the reasons for the policies, settings, configurations, and advice in the practicum. This approach also allows readers to focus on those topics that are of most interest to them.

Special Acknowledgment

Elisabeth Sullivan contributed the assurance part of this book. She wrote several drafts, all of which reflect her extensive knowledge and experience in that aspect of computer security. I am particularly grateful to her for contributing her real-world knowledge of how assurance is managed. Too often, books recount the mathematics of assurance without recognizing that other aspects are equally important and more widely used. These other aspects shine through in the assurance section, thanks to Liz. As if that were not enough, she made several suggestions that improved the policy part of this book. I will always be grateful for her contribution, her humor, and especially her friendship.

Acknowledgments

Many people have contributed to this book. Peter Salus' suggestion first got me thinking about writing it, and Peter put me in touch with Addison-Wesley. Midway through the writing, Blaine Burnham reviewed the completed portions and the proposed topics, and suggested that they be reorganized in several ways. The current organization of the book grew from his suggestions. Marvin Schaefer reviewed parts of the book with a keen eye, made suggestions that improved many parts, and encouraged me at the end. I thank these three for their contributions.

Many others contributed to this book in various ways. Special thanks to Steven Alexander, Jim Alves-Foss, Bill Arbaugh, Andrew Arcilla, Alex Aris, Rebecca Bace, Belinda Bashore, Ziad El Bizri, Logan Browne, Terry Brugger, Raymond Centeno, Michael Clifford, Christopher Clifton, Dan Coming, Crispin Cowan, Dimitri DeFigueiredo, Joseph-Patrick Dib, Jeremy Frank, Robert Fourney, Martin Gagne, Ron Gove, James Hinde, Jesper Johansson, Mark Jones, Calvin Ko, Mark-Neil Ledesma, Ken Levine, Karl Levitt, Gary McGraw, Alexander Meau, Nasir Memon, Mark Morrissey, Ather Nawaz, Iulian Neamtiu, Kimberly Nico, Stephen Northcutt, Rafael Obelheiro, Holly Pang, Ryan Poling, Sung Park, Ashwini Raina, Jorge