This page intentionally left blank
To Willis Ware, a hero of computer security and privacy.
Contents

Foreword xix
Preface xxv
Acknowledgments xxxi
About the Authors xxxiii

Chapter 1 Introduction 1

1.1 What Is Computer Security? 2
Values of Assets 4
The Vulnerability–Threat–Control Paradigm 5

1.2 Threats 6
Confidentiality 8
Integrity 10
Availability 11
Types of Threats 13
Types of Attackers 16

1.3 Harm 21
Risk and Common Sense 22
Method–Opportunity–Motive 26

1.4 Vulnerabilities 28
1.5 Controls 28
1.6 Conclusion 31
1.7 What’s Next? 32
1.8 Exercises 34
Chapter 2  Toolbox: Authentication, Access Control, and Cryptography 36

2.1 Authentication 38
Identification Versus Authentication 38
Authentication Based on Phrases and Facts: Something You Know 40
Authentication Based on Biometrics: Something You Are 53
Authentication Based on Tokens: Something You Have 65
Federated Identity Management 68
Multifactor Authentication 70
Secure Authentication 70

2.2 Access Control 72
Access Policies 72
Implementing Access Control 75
Procedure-Oriented Access Control 85
Role-Based Access Control 85

2.3 Cryptography 86
Problems Addressed by Encryption 87
Terminology 87
DES: The Data Encryption Standard 95
AES: Advanced Encryption System 98
Public Key Cryptography 100
Public Key Cryptography to Exchange Secret Keys 103
Error Detecting Codes 109
Trust 117
Certificates: Trustable Identities and Public Keys 121
Digital Signatures—All the Pieces 124

2.4 Exercises 127

Chapter 3  Programs and Programming 131

3.1 Unintentional (Nonmalicious) Programming Oversights 133
Buffer Overflow 134
Incomplete Mediation 152
Time-of-Check to Time-of-Use 155
Undocumented Access Point 157
Off-by-One Error 159
Integer Overflow 160
Contents

Unterminated Null-Terminated String 161
Parameter Length, Type, and Number 162
Unsafe Utility Program 162
Race Condition 163

3.2 Malicious Code—Malware 166
Malware—Viruses, Trojan Horses, and Worms 167
Technical Details: Malicious Code 176

3.3 Countermeasures 196
Countermeasures for Users 197
Countermeasures for Developers 203
Countermeasure Specifically for Security 216
Countermeasures that Don’t Work 224
Conclusion 229
Exercises 229

Chapter 4 The Web—User Side 232

4.1 Browser Attacks 234
Browser Attack Types 234
How Browser Attacks Succeed: Failed Identification and Authentication 240

4.2 Web Attacks Targeting Users 245
False or Misleading Content 246
Malicious Web Content 253
Protecting Against Malicious Web Pages 259

4.3 Obtaining User or Website Data 260
Code Within Data 261
Website Data: A User’s Problem, Too 265
Foiling Data Attacks 266

4.4 Email Attacks 267
Fake Email 267
Fake Email Messages as Spam 267
Fake (Inaccurate) Email Header Data 273
Phishing 274
Protecting Against Email Attacks 275

4.5 Conclusion 277
4.6 Exercises 278
## Chapter 5 Operating Systems

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Security in Operating Systems</td>
<td>280</td>
</tr>
<tr>
<td></td>
<td>Background: Operating System Structure</td>
<td>281</td>
</tr>
<tr>
<td></td>
<td>Security Features of Ordinary Operating Systems</td>
<td>282</td>
</tr>
<tr>
<td></td>
<td>A Bit of History</td>
<td>284</td>
</tr>
<tr>
<td></td>
<td>Protected Objects</td>
<td>286</td>
</tr>
<tr>
<td></td>
<td>Operating System Tools to Implement Security Functions</td>
<td>292</td>
</tr>
<tr>
<td>5.2</td>
<td>Security in the Design of Operating Systems</td>
<td>308</td>
</tr>
<tr>
<td></td>
<td>Simplicity of Design</td>
<td>309</td>
</tr>
<tr>
<td></td>
<td>Layered Design</td>
<td>309</td>
</tr>
<tr>
<td></td>
<td>Kernelized Design</td>
<td>312</td>
</tr>
<tr>
<td></td>
<td>Reference Monitor</td>
<td>313</td>
</tr>
<tr>
<td></td>
<td>Correctness and Completeness</td>
<td>314</td>
</tr>
<tr>
<td></td>
<td>Secure Design Principles</td>
<td>315</td>
</tr>
<tr>
<td></td>
<td>Trusted Systems</td>
<td>316</td>
</tr>
<tr>
<td></td>
<td>Trusted System Functions</td>
<td>319</td>
</tr>
<tr>
<td></td>
<td>The Results of Trusted Systems Research</td>
<td>325</td>
</tr>
<tr>
<td>5.3</td>
<td>Rootkit</td>
<td>329</td>
</tr>
<tr>
<td></td>
<td>Phone Rootkit</td>
<td>329</td>
</tr>
<tr>
<td></td>
<td>Rootkit Evades Detection</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>Rootkit Operates Unchecked</td>
<td>334</td>
</tr>
<tr>
<td></td>
<td>Sony XCP Rootkit</td>
<td>335</td>
</tr>
<tr>
<td></td>
<td>TDSS Rootkits</td>
<td>336</td>
</tr>
<tr>
<td></td>
<td>Other Rootkits</td>
<td>338</td>
</tr>
<tr>
<td>5.4</td>
<td>Conclusion</td>
<td>338</td>
</tr>
<tr>
<td>5.5</td>
<td>Exercises</td>
<td>339</td>
</tr>
</tbody>
</table>

## Chapter 6 Networks

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Network Concepts</td>
<td>342</td>
</tr>
<tr>
<td></td>
<td>Background: Network Transmission Media</td>
<td>343</td>
</tr>
<tr>
<td></td>
<td>Background: Protocol Layers</td>
<td>349</td>
</tr>
<tr>
<td></td>
<td>Background: Addressing and Routing</td>
<td>350</td>
</tr>
<tr>
<td>Part I—War on Networks: Network Security Attacks</td>
<td>353</td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>Threats to Network Communications</td>
<td>354</td>
</tr>
<tr>
<td></td>
<td>Interception: Eavesdropping and Wiretapping</td>
<td>354</td>
</tr>
<tr>
<td></td>
<td>Modification, Fabrication: Data Corruption</td>
<td>361</td>
</tr>
<tr>
<td></td>
<td>Interruption: Loss of Service</td>
<td>366</td>
</tr>
<tr>
<td></td>
<td>Port Scanning</td>
<td>369</td>
</tr>
<tr>
<td></td>
<td>Vulnerability Summary</td>
<td>374</td>
</tr>
</tbody>
</table>
6.3 Wireless Network Security 374
   WiFi Background 374
   Vulnerabilities in Wireless Networks 381
   Failed Countermeasure: WEP (Wired Equivalent Privacy) 388
   Stronger Protocol Suite: WPA (WiFi Protected Access) 390

6.4 Denial of Service 396
   Example: Massive Estonian Web Failure 396
   How Service Is Denied 398
   Flooding Attacks in Detail 402
   Network Flooding Caused by Malicious Code 403
   Network Flooding by Resource Exhaustion 407
   Denial of Service by Addressing Failures 408
   Traffic Redirection 413
   DNS Attacks 414
   Exploiting Known Vulnerabilities 419
   Physical Disconnection 420

6.5 Distributed Denial-of-Service 421
   Scripted Denial-of-Service Attacks 423
   Bots 426
   Botnets 426
   Malicious Autonomous Mobile Agents 430
   Autonomous Mobile Protective Agents 430

Part II—Strategic Defenses: Security Countermeasures 432

6.6 Cryptography in Network Security 432
   Network Encryption 433
   Browser Encryption 437
   Onion Routing 443
   IP Security Protocol Suite (IPsec) 444
   Virtual Private Networks 447
   System Architecture 450

6.7 Firewalls 451
   What Is a Firewall? 452
   Design of Firewalls 453
   Types of Firewalls 454
   Personal Firewalls 465
   Comparison of Firewall Types 467
   Example Firewall Configurations 467
   Network Address Translation (NAT) 472
   Data Loss Prevention 473
7.5 Data Mining and Big Data 535
   Data Mining 536
   Big Data 540
7.6 Conclusion 549
Exercises 549

Chapter 8   Cloud Computing 551
8.1 Cloud Computing Concepts 551
   Service Models 552
   Deployment Models 552
8.2 Moving to the Cloud 553
   Risk Analysis 553
   Cloud Provider Assessment 554
   Switching Cloud Providers 556
   Cloud as a Security Control 557
8.3 Cloud Security Tools and Techniques 560
   Data Protection in the Cloud 561
   Cloud Application Security 566
   Logging and Incident Response 567
8.4 Cloud Identity Management 568
   Security Assertion Markup Language 570
   OAuth 573
   OAuth for Authentication 577
8.5 Securing IaaS 579
   Public IaaS Versus Private Network Security 580
8.6 Conclusion 583
   Where the Field Is Headed 584
   To Learn More 584
8.7 Exercises 584

Chapter 9   Privacy 586
9.1 Privacy Concepts 587
   Aspects of Information Privacy 587
   Computer-Related Privacy Problems 590
9.2 Privacy Principles and Policies 596
   Fair Information Practices 596
   U.S. Privacy Laws 597
Chapter 10  Management and Incidents

10.1  Security Planning
   Organizations and Security Plans 648
   Contents of a Security Plan 649
   Security Planning Team Members 656
   Assuring Commitment to a Security Plan 656

10.2  Business Continuity Planning
   Assess Business Impact 660
   Develop Strategy 660
   Develop the Plan 661

10.3  Handling Incidents
   Incident Response Plans 662
   Incident Response Teams 665

10.4  Risk Analysis
   The Nature of Risk 669
   Steps of a Risk Analysis 670
   Arguments For and Against Risk Analysis 684

10.5  Dealing with Disaster
   Natural Disasters 686
   Power Loss 688
   Human Vandals 689
   Interception of Sensitive Information 692
   Contingency Planning 694
   Physical Security Recap 698

10.6  Conclusion

10.7  Exercises

Chapter 11  Legal Issues and Ethics

11.1  Protecting Programs and Data
   Copyrights 704
   Patents 711
   Trade Secrets 714
   Special Cases 716

11.2  Information and the Law
   Information as an Object 717
   Legal Issues Relating to Information 720
xvi Contents

The Legal System 721

Summary of Protection for Computer Artifacts 724

11.3 Rights of Employees and Employers 725

Ownership of Products 725

Employment Contracts 727

11.4 Redress for Software Failures 728

Selling Correct Software 729

Reporting Software Flaws 731

11.5 Computer Crime 733

Why a Separate Category for Computer Crime Is Needed 734

Why Computer Crime Is Hard to Define 736

Why Computer Crime Is Hard to Prosecute 736

Examples of Statutes 737

International Dimensions 741

Why Computer Criminals Are Hard to Catch 742

What Computer Crime Does Not Address 743

Summary of Legal Issues in Computer Security 743

11.6 Ethical Issues in Computer Security 744

Differences Between the Law and Ethics 744

Studying Ethics 746

Ethical Reasoning 747

11.7 Incident Analysis with Ethics 750

Situation I: Use of Computer Services 750

Situation II: Privacy Rights 752

Situation III: Denial of Service 753

Situation IV: Ownership of Programs 754

Situation V: Proprietary Resources 756

Situation VI: Fraud 757

Situation VII: Accuracy of Information 758

Situation VIII: Ethics of Hacking or Cracking 759

Situation IX: True Representation 762

Conclusion of Computer Ethics 764

Conclusion 765

Exercises 765

Chapter 12 Details of Cryptography 768

12.1 Cryptology 769

Cryptanalysis 769

Cryptographic Primitives 773
### Contents

- One-Time Pads 775
- Statistical Analysis 776
- What Makes a “Secure” Encryption Algorithm? 777

#### 12.2 Symmetric Encryption Algorithms 779
- DES 779
- AES 789
- RC2, RC4, RC5, and RC6 792

#### 12.3 Asymmetric Encryption with RSA 795
- The RSA Algorithm 795
- Strength of the RSA Algorithm 797

#### 12.4 Message Digests 799
- Hash Functions 799
- One-Way Hash Functions 799
- Message Digests 800

#### 12.5 Digital Signatures 802
- Elliptic Curve Cryptosystems 802
- El Gamal and Digital Signature Algorithms 803
- The NSA–Cryptography Controversy of 2012 804

#### 12.6 Quantum Cryptography 807
- Quantum Physics 807
- Photon Reception 808
- Cryptography with Photons 808
- Implementation 811

#### 12.7 Conclusion 811

### Chapter 13 Emerging Topics 813

#### 13.1 The Internet of Things 814
- Medical Devices 815
- Mobile Phones 818
- Security in the Internet of Things 820

#### 13.2 Economics 821
- Making a Business Case 821
- Quantifying Security 825
- Current Research and Future Directions 832

#### 13.3 Electronic Voting 834
- What Is Electronic Voting? 835
- What Is a Fair Election? 836
- What Are the Critical Issues? 837
13.4 Cyber Warfare 841
   What Is Cyber Warfare? 842
   Possible Examples of Cyber Warfare 843
   Critical Issues 846
13.5 Conclusion 850

Bibliography 851

Index 877
In the 1950s and 1960s, the prominent conference gathering places for practitioners and users of computer technology were the twice yearly Joint Computer Conferences (JCCs)—initially called the Eastern and Western JCCs, but later renamed the Spring and Fall JCCs and even later, the annual National (AFIPS) Computer Conference. From this milieu, the topic of computer security—later to be called information system security and currently also referred to as “protection of the national information infrastructure”—moved from the world of classified defense interests into public view.

A few people—Robert L. Patrick, John P. Haverty, and myself among others—all then at The RAND Corporation (as its name was then known) had been talking about the growing dependence of the country and its institutions on computer technology. It concerned us that the installed systems might not be able to protect themselves and their data against intrusive and destructive attacks. We decided that it was time to bring the security aspect of computer systems to the attention of the technology and user communities.

Foreword

From the authors: Willis Ware kindly wrote the foreword that we published in both the third and fourth editions of Security in Computing. In his foreword he covers some of the early days of computer security, describing concerns that are as valid today as they were in those earlier days.

Willis chose to sublimate his name and efforts to the greater good of the projects he worked on. In fact, his thoughtful analysis and persuasive leadership contributed much to the final outcome of these activities. Few people recognize Willis’s name today; more people are familiar with the European Union Data Protection Directive that is a direct descendant of the report [WAR73a] from his committee for the U.S. Department of Human Services. Willis would have wanted it that way: the emphasis on the ideas and not on his name.

Unfortunately, Willis died in November 2013 at age 93. We think the lessons he wrote about in his Foreword are still important to our readers. Thus, with both respect and gratitude, we republish his words here.
The enabling event was the development within the National Security Agency (NSA) of a remote-access time-sharing system with a full set of security access controls, running on a Univac 494 machine, and serving terminals and users not only within the headquarters building at Fort George G. Meade, Maryland, but also worldwide. Fortuitously, I knew details of the system.

Persuading two others from RAND to help—Dr. Harold Peterson and Dr. Rein Turn—plus Bernard Peters of NSA, I organized a group of papers and presented it to the SJCC conference management as a ready-made additional paper session to be chaired by me. [1] The conference accepted the offer, and the session was presented at the Atlantic City (NJ) Convention Hall in 1967.

Soon thereafter and driven by a request from a defense contractor to include both defense classified and business applications concurrently in a single mainframe machine functioning in a remote-access mode, the Department of Defense, acting through the Advanced Research Projects Agency (ARPA) and later the Defense Science Board (DSB), organized a committee, which I chaired, to study the issue of security controls for computer systems. The intent was to produce a document that could be the basis for formulating a DoD policy position on the matter.

The report of the committee was initially published as a classified document and was formally presented to the sponsor (the DSB) in January 1970. It was later declassified and republished (by The RAND Corporation) in October 1979. [2] It was widely circulated and became nicknamed “the Ware report.” The report and a historical introduction are available on the RAND website. [3]

Subsequently, the United States Air Force (USAF) sponsored another committee chaired by James P. Anderson. [4] Its report, published in 1972, recommended a 6-year R&D security program totaling some $8M. [5] The USAF responded and funded several projects, three of which were to design and implement an operating system with security controls for a specific computer.

Eventually these activities led to the “Criteria and Evaluation” program sponsored by the NSA. It culminated in the “Orange Book” [6] in 1983 and subsequently its supporting array of documents, which were nicknamed “the rainbow series.” [7] Later, in the 1980s and on into the 1990s, the subject became an international one leading to the ISO standard known as the “Common Criteria.” [8]

It is important to understand the context in which system security was studied in the early decades. The defense establishment had a long history of protecting classified information in document form. It had evolved a very elaborate scheme for compartmenting material into groups, sub-groups and super-groups, each requiring a specific personnel clearance and need-to-know as the basis for access. [9] It also had a centuries-long legacy of encryption technology and experience for protecting classified information in transit. Finally, it understood the personnel problem and the need to establish the trustworthiness of its people. And it certainly understood the physical security matter.

Thus, the computer security issue, as it was understood in the 1960s and even later, was how to create in a computer system a group of access controls that would implement or emulate the processes of the prior paper world, plus the associated issues of protecting such software against unauthorized change, subversion and illicit use, and of embedding the entire system in a secure physical environment with appropriate
management oversights and operational doctrine and procedures. The poorly under-
stood aspect of security was primarily the software issue with, however, a collateral
hardware aspect; namely, the risk that it might malfunction—or be penetrated—and
subvert the proper behavior of software. For the related aspects of communications,
personnel, and physical security, there was a plethora of rules, regulations, doctrine and
experience to cover them. It was largely a matter of merging all of it with the hardware/
software aspects to yield an overall secure system and operating environment.

However, the world has now changed and in essential ways. The desk-top computer
and workstation have appeared and proliferated widely. The Internet is flourishing
and the reality of a World Wide Web is in place. Networking has exploded and com-
unication among computer systems is the rule, not the exception. Many commercial
transactions are now web-based; many commercial communities—the financial one in
particular—have moved into a web posture. The “user” of any computer system can
literally be anyone in the world. Networking among computer systems is ubiquitous;
information-system outreach is the goal.

The net effect of all of this has been to expose the computer-based information system—
its hardware, its software, its software processes, its databases, its communications—to
an environment over which no one—not end-user, not network administrator or system
owner, not even government—has control. What must be done is to provide appropriate
technical, procedural, operational and environmental safeguards against threats as they
might appear or be imagined, embedded in a societally acceptable legal framework.

And appear threats did—from individuals and organizations, national and interna-
tional. The motivations to penetrate systems for evil purpose or to create malicious
software—generally with an offensive or damaging consequence—vary from personal
intellectual satisfaction to espionage, to financial reward, to revenge, to civil disobedi-
ence, and to other reasons. Information-system security has moved from a largely self-
contained bounded environment interacting with a generally known and disciplined user
community to one of worldwide scope with a body of users that may not be known and
are not necessarily trusted. Importantly, security controls now must deal with circum-
stances over which there is largely no control or expectation of avoiding their impact.
Computer security, as it has evolved, shares a similarity with liability insurance; they
each face a threat environment that is known in a very general way and can generate
attacks over a broad spectrum of possibilities; but the exact details or even time or cer-
tainty of an attack is unknown until an event has occurred.

On the other hand, the modern world thrives on information and its flows; the con-
temporary world, society and institutions cannot function without their computer-
communication-based information systems. Hence, these systems must be protected in
all dimensions—technical, procedural, operational, environmental. The system owner
and its staff have become responsible for protecting the organization’s information
assets.

Progress has been slow, in large part because the threat has not been perceived as real
or as damaging enough; but also in part because the perceived cost of comprehensive
information system security is seen as too high compared to the risks—especially the
financial consequences—of not doing it. Managements, whose support with appropriate
funding is essential, have been slow to be convinced.
This book addresses the broad sweep of issues above: the nature of the threat and system vulnerabilities (Chapter 1); cryptography (Chapters 2 and 12); software vulnerabilities (Chapter 3); the Common Criteria (Chapter 5); the World Wide Web and Internet (Chapters 4 and 6); managing risk (Chapter 10); and legal, ethical and privacy issues (Chapter 11). The book also describes security controls that are currently available such as encryption protocols, software development practices, firewalls, and intrusion-detection systems. Overall, this book provides a broad and sound foundation for the information-system specialist who is charged with planning and/or organizing and/or managing and/or implementing a comprehensive information-system security program.

Yet to be solved are many technical aspects of information security—R&D for hardware, software, systems, and architecture; and the corresponding products. Notwithstanding, technology per se is not the long pole in the tent of progress. Organizational and management motivation and commitment to get the security job done is. Today, the collective information infrastructure of the country and of the world is slowly moving up the learning curve; every mischievous or malicious event helps to push it along. The terrorism-based events of recent times are helping to drive it. Is it far enough up the curve to have reached an appropriate balance between system safety and threat? Almost certainly, the answer is “no, not yet; there is a long way to go.” [10]

—Willis H. Ware
RAND
Santa Monica, California
Citations


5. All of these documents are cited in the bibliography of this book. For images of these historical papers on a CDROM, see the “History of Computer Security Project, Early Papers Part 1,” Professor Matt Bishop; Department of Computer Science, University of California at Davis. http://seclab.cs.ucdavis.edu/projects/history


7. So named because the cover of each document in the series had a unique and distinctively colored cover page. For example, the “Red Book” is “Trusted Network Interpretation,” National Computer Security Center, National Security Agency, Ft. George G. Meade, Maryland; NCSC-TG-005, July 31, 1987. USGPO Stock number 008-000-00486-2.


9. This scheme is nowhere, to my knowledge, documented explicitly. However, its complexity can be inferred by a study of Appendices A and B of R-609.1 (item [2] above).

Preface

Tablets, smartphones, TV set-top boxes, GPS navigation devices, exercise monitors, home security stations, even washers and dryers come with Internet connections by which data from and about you go to places over which you have little visibility or control. At the same time, the list of retailers suffering massive losses of customer data continues to grow: Home Depot, Target, T.J. Maxx, P.F. Chang’s, Sally Beauty. On the one hand people want the convenience and benefits that added connectivity brings, while on the other hand, people are worried, and some are seriously harmed by the impact of such incidents. Computer security brings these two threads together as technology races forward with smart products whose designers omit the basic controls that can prevent or limit catastrophes.

To some extent, people sigh and expect security failures in basic products and complex systems. But these failures do not have to be. Every computer professional can learn how such problems occur and how to counter them. Computer security has been around as a field since the 1960s, and it has developed excellent research, leading to a good understanding of the threat and how to manage it.

One factor that turns off many people is the language: Complicated terms such as polymorphic virus, advanced persistent threat, distributed denial-of-service attack, inference and aggregation, multifactor authentication, key exchange protocol, and intrusion detection system do not exactly roll off the tongue. Other terms sound intriguing but opaque, such as worm, botnet, rootkit, man in the browser, honeynet, sandbox, and script kiddie. The language of advanced mathematics or microbiology is no less confusing, and the Latin terminology of medicine and law separates those who know it from those who do not. But the terms and concepts of computer security really have straightforward, easy-to-learn meaning and uses.

The premise of computer security is quite simple: Vulnerabilities are weaknesses in products, systems, protocols, algorithms, programs, interfaces, and designs. A threat is

**Vulnerability:** weakness
**Threat:** condition that exercises vulnerability
**Incident:** vulnerability + threat
**Control:** reduction of threat or vulnerability
a condition that could exercise a vulnerability. An incident occurs when a threat does exploit a vulnerability, causing harm. Finally, people add controls or countermeasures to prevent, deflect, diminish, detect, diagnose, and respond to threats. All of computer security is built from that simple framework. This book is about bad things that can happen with computers and ways to protect our computing.

WHY READ THIS BOOK?

Admit it. You know computing entails serious risks to the privacy of your personal data, the integrity of your data, or the operation of your computer. Risk is a fact of life: Crossing the street is risky, perhaps more so in some places than others, but you still cross the street. As a child you learned to stop and look both ways before crossing. As you became older you learned to gauge the speed of oncoming traffic and determine whether you had the time to cross. At some point you developed a sense of whether an oncoming car would slow down or yield. We hope you never had to practice this, but sometimes you have to decide whether darting into the street without looking is the best means of escaping danger. The point is all these matters depend on knowledge and experience. We want to help you develop comparable knowledge and experience with respect to the risks of secure computing.

The same thing can be said about computer security in everything from personal devices to complex commercial systems: You start with a few basic terms, principles, and concepts. Then you learn the discipline by seeing those basics reappear in numerous situations, including programs, operating systems, networks, and cloud computing. You pick up a few fundamental tools, such as authentication, access control, and encryption, and you understand how they apply in defense strategies. You start to think like an attacker, predicting the weaknesses that could be exploited, and then you shift to selecting defenses to counter those attacks. This last stage of playing both offense and defense makes computer security a creative and challenging activity.

USES FOR AND USERS OF THIS BOOK

This book is intended for people who want to learn about computer security; if you have read this far you may well be such a person. This book is intended for three groups of people: college and university students, computing professionals and managers, and users of all kinds of computer-based systems. All want to know the same thing: how to control the risk of computer security. But you may differ in how much information you need about particular topics: Some readers want a broad survey, while others want to focus on particular topics, such as networks or program development.

This book should provide the breadth and depth that most readers want. The book is organized by general area of computing, so that readers with particular interests can find information easily.
ORGANIZATION OF THIS BOOK

The chapters of this book progress in an orderly manner, from general security concerns to the particular needs of specialized applications, and then to overarching management and legal issues. Thus, this book progresses through six key areas of interest:

1. Introduction: threats, vulnerabilities, and controls
2. The security practitioner’s “toolbox”: identification and authentication, access control, and encryption
3. Application areas of computer security practice: programs, user–Internet interaction, operating systems, networks, data and databases, and cloud computing
4. Cross-cutting disciplines: privacy, management, law and ethics
5. Details of cryptography
6. Emerging application domains

The first chapter begins like many other expositions: by laying groundwork. In Chapter 1 we introduce terms and definitions, and give some examples to justify how these terms are used. In Chapter 2 we begin the real depth of the field by introducing three concepts that form the basis of many defenses in computer security: identification and authentication, access control, and encryption. We describe different ways of implementing each of these, explore strengths and weaknesses, and tell of some recent advances in these technologies.

Then we advance through computing domains, from the individual user outward. In Chapter 3 we begin with individual programs, ones you might write and those you only use. Both kinds are subject to potential attacks, and we examine the nature of some of those attacks and how they could have been prevented. In Chapter 4 we move on to a type of program with which most users today are quite familiar: the browser, as a gateway to the Internet. The majority of attacks today are remote, carried from a distant attacker across a network, usually the Internet. Thus, it makes sense to study Internet-borne malicious code. But this chapter’s focus is on the harm launched remotely, not on the network infrastructure by which it travels; we defer the network concepts to Chapter 6. In Chapter 5 we consider operating systems, a strong line of defense between a user and attackers. We also consider ways to undermine the strength of the operating system itself. Chapter 6 returns to networks, but this time we do look at architecture and technology, including denial-of-service attacks that can happen only in a network. Data, their collection and protection, form the topic of Chapter 7, in which we look at database management systems and big data applications. Finally, in Chapter 8 we explore cloud computing, a relatively recent addition to the computing landscape, but one that brings its own vulnerabilities and protections.

In Chapters 9 through 11 we address what we have termed the intersecting disciplines: First, in Chapter 9 we explore privacy, a familiar topic that relates to most of the six domains from programs to clouds. Then Chapter 10 takes us to the management side of computer security: how management plans for and addresses computer security problems. Finally, Chapter 11 explores how laws and ethics help us control computer behavior.
We introduced cryptography in Chapter 2. But the field of cryptography involves entire books, courses, conferences, journals, and postgraduate programs of study. And this book needs to cover many important topics in addition to cryptography. Thus, we made two critical decisions: First, we treat cryptography as a tool, not as a field of study. An automobile mechanic does not study the design of cars, weighing such factors as aerodynamics, fuel consumption, interior appointment, and crash resistance; a mechanic accepts a car as a given and learns how to find and fix faults with the engine and other mechanical parts. Similarly, we want our readers to be able to use cryptography to quickly address security problems; hence we briefly visit popular uses of cryptography in Chapter 2. Our second critical decision was to explore the breadth of cryptography slightly more in a later chapter, Chapter 12. But as we point out, entire books have been written on cryptography, so our later chapter gives an overview of more detailed work that interested readers can find elsewhere.

Our final chapter detours to four areas having significant computer security hazards. These are rapidly advancing topics for which the computer security issues are much in progress right now. The so-called Internet of Things, the concept of connecting many devices to the Internet, raises potential security threats waiting to be explored. Economics govern many security decisions, so security professionals need to understand how economics and security relate. Convenience is raising interest in using computers to implement elections; the easy steps of collecting vote totals have been done by many jurisdictions, but the hard part of organizing fair online registration and ballot-casting have been done in only a small number of demonstration elections. And the use of computers in warfare is a growing threat. Again, a small number of modest-sized attacks on computing devices have shown the feasibility of this type of campaign, but security professionals and ordinary citizens need to understand the potential—both good and bad—of this type of attack.

HOW TO READ THIS BOOK

What background should you have to appreciate this book? The only assumption is an understanding of programming and computer systems. Someone who is an advanced undergraduate or graduate student in computing certainly has that background, as does a professional designer or developer of computer systems. A user who wants to understand more about how programs work can learn from this book, too; we provide the necessary background on concepts of operating systems or networks, for example, before we address the related security concerns.

This book can be used as a textbook in a one- or two-semester course in computer security. The book functions equally well as a reference for a computer professional or as a supplement to an intensive training course. And the index and extensive bibliography make it useful as a handbook to explain significant topics and point to key articles in the literature. The book has been used in classes throughout the world; instructors often design one-semester courses that focus on topics of particular interest to the students or that relate well to the rest of a curriculum.
WHAT IS NEW IN THIS BOOK

This is the fifth edition of *Security in Computing*, first published in 1989. Since then, the specific threats, vulnerabilities, and controls have changed, as have many of the underlying technologies to which computer security applies. However, many basic concepts have remained the same.

Most obvious to readers familiar with earlier editions will be some new chapters, specifically, on user–web interaction and cloud computing, as well as the topics we raise in the emerging topics chapter. Furthermore, pulling together the three fundamental controls in Chapter 2 is a new structure. Those are the big changes, but every chapter has had many smaller changes, as we describe new attacks or expand on points that have become more important.

One other feature some may notice is the addition of a third coauthor. Jonathan Margulies joins us as an essential member of the team that produced this revision. He is currently director of the security practice at Qmulos, a newly launched security consulting practice. He brings many years of experience with Sandia National Labs and the National Institute for Standards and Technology. His focus meshes nicely with our existing skills to extend the breadth of this book.
It is increasingly difficult to acknowledge all the people who have influenced this book. Colleagues and friends have contributed their knowledge and insight, often without knowing their impact. By arguing a point or sharing explanations of concepts, our associates have forced us to question or rethink what we know.

We thank our associates in at least two ways. First, we have tried to include references to their written works. References in the text cite specific papers relating to particular thoughts or concepts, but the bibliography also includes broader works that have played a more subtle role in shaping our approach to security. So, to all the cited authors, many of whom are friends and colleagues, we happily acknowledge your positive influence on this book.

Rather than name individuals, we thank the organizations in which we have interacted with creative, stimulating, and challenging people from whom we learned a lot. These places include Trusted Information Systems, the Contel Technology Center, the Centre for Software Reliability of the City University of London, Arca Systems, Exodus Communications, The RAND Corporation, Sandia National Lab, Cable & Wireless, the National Institute of Standards and Technology, the Institute for Information Infrastructure Protection, Qmulos, and the Editorial Board of IEEE Security & Privacy. If you worked with us at any of these locations, chances are high that your imprint can be found in this book. And for all the side conversations, debates, arguments, and light moments, we are grateful.
About the Authors

Charles P. Pfleeger is an internationally known expert on computer and communications security. He was originally a professor at the University of Tennessee, leaving there to join computer security research and consulting companies Trusted Information Systems and Arca Systems (later Exodus Communications and Cable and Wireless). With Trusted Information Systems he was Director of European Operations and Senior Consultant. With Cable and Wireless he was Director of Research and a member of the staff of the Chief Security Officer. He was chair of the IEEE Computer Society Technical Committee on Security and Privacy.

Shari Lawrence Pfleeger is widely known as a software engineering and computer security researcher, most recently as a Senior Computer Scientist with the Rand Corporation and as Research Director of the Institute for Information Infrastructure Protection. She is currently Editor-in-Chief of IEEE Security & Privacy magazine.

Jonathan Margulies is the CTO of Qmulos, a cybersecurity consulting firm. After receiving his master’s degree in Computer Science from Cornell University, Mr. Margulies spent nine years at Sandia National Labs, researching and developing solutions to protect national security and critical infrastructure systems from advanced persistent threats. He then went on to NIST’s National Cybersecurity Center of Excellence, where he worked with a variety of critical infrastructure companies to create industry-standard security architectures. In his free time, Mr. Margulies edits the “Building Security In” section of IEEE Security & Privacy magazine.
This page intentionally left blank
On 11 February 2013, residents of Great Falls, Montana received the following warning on their televisions [INF13]. The transmission displayed a message banner on the bottom of the screen (as depicted in Figure 1-1).

And the following alert was broadcast:

**FIGURE 1-1** Emergency Broadcast Warning
Chapter 1  Introduction

[Beep Beep Beep: the sound pattern of the U.S. government Emergency Alert System. The following text then scrolled across the screen:]

Civil authorities in your area have reported that the bodies of the dead are rising from their graves and attacking the living. Follow the messages on screen that will be updated as information becomes available.

Do not attempt to approach or apprehend these bodies as they are considered extremely dangerous. This warning applies to all areas receiving this broadcast.

[Beep Beep Beep]

The warning signal sounded authentic; it had the distinctive tone people recognize for warnings of serious emergencies such as hazardous weather or a natural disaster. And the text was displayed across a live broadcast television program. On the other hand, bodies rising from their graves sounds suspicious.

What would you have done?

Only four people contacted police for assurance that the warning was indeed a hoax. As you can well imagine, however, a different message could have caused thousands of people to jam the highways trying to escape. (On 30 October 1938 Orson Welles performed a radio broadcast of the H. G. Wells play War of the Worlds that did cause a minor panic of people believing that Martians had landed and were wreaking havoc in New Jersey.)

The perpetrator of this hoax was never caught, nor has it become clear exactly how it was done. Likely someone was able to access the system that feeds emergency broadcasts to local radio and television stations. In other words, a hacker probably broke into a computer system.

You encounter computers daily in countless situations, often in cases in which you are scarcely aware a computer is involved, like the emergency alert system for broadcast media. These computers move money, control airplanes, monitor health, lock doors, play music, heat buildings, regulate hearts, deploy airbags, tally votes, direct communications, regulate traffic, and do hundreds of other things that affect lives, health, finances, and well-being. Most of the time these computers work just as they should. But occasionally they do something horribly wrong, because of either a benign failure or a malicious attack.

This book is about the security of computers, their data, and the devices and objects to which they relate. In this book you will learn some of the ways computers can fail—or be made to fail—and how to protect against those failures. We begin that study in the way any good report does: by answering the basic questions of what, who, why, and how.

1.1 WHAT IS COMPUTER SECURITY?

Computer security is the protection of the items you value, called the assets of a computer or computer system. There are many types of assets, involving hardware, software, data, people, processes, or combinations of these. To determine what to protect, we must first identify what has value and to whom.
A computer device (including hardware, added components, and accessories) is certainly an asset. Because most computer hardware is pretty useless without programs, the software is also an asset. Software includes the operating system, utilities and device handlers; applications such as word processing, media players or email handlers; and even programs that you may have written yourself. Much hardware and software is off-the-shelf, meaning that it is commercially available (not custom-made for your purpose) and that you can easily get a replacement. The thing that makes your computer unique and important to you is its content: photos, tunes, papers, email messages, projects, calendar information, ebooks (with your annotations), contact information, code you created, and the like. Thus, data items on a computer are assets, too. Unlike most hardware and software, data can be hard—if not impossible—to recreate or replace. These assets are all shown in Figure 1-2.

These three things—hardware, software, and data—contain or express things like the design for your next new product, the photos from your recent vacation, the chapters of your new book, or the genome sequence resulting from your recent research. All of these things represent intellectual endeavor or property, and they have value that differs from one person or organization to another. It is that value that makes them assets worthy of protection, and they are the elements we want to protect. Other assets—such as access to data, quality of service, processes, human users, and network connectivity—deserve protection, too; they are affected or enabled by the hardware, software, and data. So in most cases, protecting hardware, software, and data covers these other assets as well.

In this book, unless we specifically distinguish between hardware, software, and data, we refer to all these assets as the computer system.

Computer systems—hardware, software, and data—have value and deserve security protection.
or sometimes as the computer. And because processors are embedded in so many devices, we also need to think about such variations as mobile phones, implanted pacemakers, heating controllers, and automobiles. Even if the primary purpose of the device is not computing, the device’s embedded computer can be involved in security incidents and represents an asset worthy of protection.

Values of Assets

After identifying the assets to protect, we next determine their value. We make value-based decisions frequently, even when we are not aware of them. For example, when you go for a swim you can leave a bottle of water and a towel on the beach, but not your wallet or cell phone. The difference relates to the value of the assets.

The value of an asset depends on the asset owner’s or user’s perspective, and it may be independent of monetary cost, as shown in Figure 1-3. Your photo of your sister, worth only a few cents in terms of paper and ink, may have high value to you and no value to your roommate. Other items’ value depends on replacement cost; some computer data are difficult or impossible to replace. For example, that photo of you and your friends at a party may have cost you nothing, but it is invaluable because there is no other copy. On the other hand, the DVD of your favorite film may have cost a significant portion of your take-home pay, but you can buy another one if the DVD is stolen or corrupted. Similarly, timing has bearing on asset

Assets’ values are personal, time dependent, and often imprecise.
value. For example, the value of the plans for a company’s new product line is very high, especially to competitors. But once the new product is released, the plans’ value drops dramatically.

The Vulnerability–Threat–Control Paradigm

The goal of computer security is protecting valuable assets. To study different ways of protection, we use a framework that describes how assets may be harmed and how to counter or mitigate that harm.

A vulnerability is a weakness in the system, for example, in procedures, design, or implementation, that might be exploited to cause loss or harm. For instance, a particular system may be vulnerable to unauthorized data manipulation because the system does not verify a user’s identity before allowing data access.

A threat to a computing system is a set of circumstances that has the potential to cause loss or harm. To see the difference between a threat and a vulnerability, consider the illustration in Figure 1-4. Here, a wall is holding water back. The water to the left of the wall is a threat to the man on the right of the wall: The water could rise, overflowing onto the man, or it could stay beneath the height of the wall, causing the wall to collapse. So the threat of harm is the potential for the man to get wet, get hurt, or be drowned. For now, the wall is intact, so the threat to the man is unrealized.
Chapter 1  Introduction

However, we can see a small crack in the wall—a vulnerability that threatens the man’s security. If the water rises to or beyond the level of the crack, it will exploit the vulnerability and harm the man.

There are many threats to a computer system, including human-initiated and computer-initiated ones. We have all experienced the results of inadvertent human errors, hardware design flaws, and software failures. But natural disasters are threats, too; they can bring a system down when the computer room is flooded or the data center collapses from an earthquake, for example.

A human who exploits a vulnerability perpetrates an **attack** on the system. An attack can also be launched by another system, as when one system sends an overwhelming flood of messages to another, virtually shutting down the second system’s ability to function. Unfortunately, we have seen this type of attack frequently, as denial-of-service attacks deluge servers with more messages than they can handle. (We take a closer look at denial of service in Chapter 6.)

How do we address these problems? We use a **control** or **countermeasure** as protection. That is, a control is an action, device, procedure, or technique that removes or reduces a vulnerability. In Figure 1-4, the man is placing his finger in the hole, controlling the threat of water leaks until he finds a more permanent solution to the problem. In general, we can describe the relationship between threats, controls, and vulnerabilities in this way:

**A threat is blocked by control of a vulnerability.**

Before we can protect assets, we need to know the kinds of harm we have to protect them against, so now we explore threats to valuable assets.

1.2 THREATS

We can consider potential harm to assets in two ways: First, we can look at what bad things can happen to assets, and second, we can look at who or what can cause or allow those bad things to happen. These two perspectives enable us to determine how to protect assets.

Think for a moment about what makes your computer valuable to you. First, you use it as a tool for sending and receiving email, searching the web, writing papers, and performing many other tasks, and you expect it to be available for use when you want it. Without your computer these tasks would be harder, if not impossible. Second, you rely heavily on your computer’s integrity. When you write a paper and save it, you trust that the paper will reload exactly as you saved it. Similarly, you expect that the photo a friend passes you on a flash drive will appear the same when you load it into your computer as when you saw it on your friend’s computer. Finally, you expect the “personal” aspect of a personal computer to stay personal, meaning you want it to protect your confidentiality. For example, you want your email messages to be just between you and
your listed recipients; you don’t want them broadcast to other people. And when you write an essay, you expect that no one can copy it without your permission.

These three aspects, confidentiality, integrity, and availability, make your computer valuable to you. But viewed from another perspective, they are three possible ways to make it less valuable, that is, to cause you harm. If someone steals your computer, scrambles data on your disk, or looks at your private data files, the value of your computer has been diminished or your computer use has been harmed. These characteristics are both basic security properties and the objects of security threats.

We can define these three properties as follows.

- **availability**: the ability of a system to ensure that an asset can be used by any authorized parties
- **integrity**: the ability of a system to ensure that an asset is modified only by authorized parties
- **confidentiality**: the ability of a system to ensure that an asset is viewed only by authorized parties

These three properties, hallmarks of solid security, appear in the literature as early as James P. Anderson’s essay on computer security [AND73] and reappear frequently in more recent computer security papers and discussions. Taken together (and rearranged), the properties are called the **C-I-A triad** or the **security triad**. ISO 7498-2 [ISO89] adds to them two more properties that are desirable, particularly in communication networks:

- **authentication**: the ability of a system to confirm the identity of a sender
- **nonrepudiation** or **accountability**: the ability of a system to confirm that a sender cannot convincingly deny having sent something

The U.S. Department of Defense [DOD85] adds auditability: the ability of a system to trace all actions related to a given asset. The C-I-A triad forms a foundation for thinking about security. Authenticity and nonrepudiation extend security notions to network communications, and auditability is important in establishing individual accountability for computer activity. In this book we generally use the C-I-A triad as our security taxonomy so that we can frame threats, vulnerabilities, and controls in terms of the C-I-A properties affected. We highlight one of these other properties when it is relevant to a particular threat we are describing. For now, we focus on just the three elements of the triad.

What can happen to harm the confidentiality, integrity, or availability of computer assets? If a thief steals your computer, you no longer have access, so you have lost availability; furthermore, if the thief looks at the pictures or documents you have stored, your confidentiality is compromised. And if the thief changes the content of your music files but then gives them back with your computer, the integrity of your data has been harmed. You can envision many scenarios based around these three properties.
The C-I-A triad can be viewed from a different perspective: the nature of the harm caused to assets. Harm can also be characterized by four acts: interception, interruption, modification, and fabrication. These four acts are depicted in Figure 1-5. From this point of view, confidentiality can suffer if someone intercepts data, availability is lost if someone or something interrupts a flow of data or access to a computer, and integrity can fail if someone or something modifies data or fabricates false data. Thinking of these four kinds of acts can help you determine what threats might exist against the computers you are trying to protect.

To analyze harm, we next refine the C-I-A triad, looking more closely at each of its elements.

Confidentiality

Some things obviously need confidentiality protection. For example, students’ grades, financial transactions, medical records, and tax returns are sensitive. A proud student may run out of a classroom screaming “I got an A!” but the student should be the one to choose whether to reveal that grade to others. Other things, such as diplomatic and military secrets, companies’ marketing and product development plans, and educators’ tests, also must be carefully controlled. Sometimes, however, it is not so obvious that something is sensitive. For example, a military food order may seem like innocuous information, but a sudden increase in the order could be a sign of incipient engagement in conflict. Purchases of food, hourly changes in location, and access to books are not
things you would ordinarily consider confidential, but they can reveal something that someone wants to be kept confidential.

The definition of confidentiality is straightforward: Only authorized people or systems can access protected data. However, as we see in later chapters, ensuring confidentiality can be difficult. For example, who determines which people or systems are authorized to access the current system? By “accessing” data, do we mean that an authorized party can access a single bit? the whole collection? pieces of data out of context? Can someone who is authorized disclose data to other parties? Sometimes there is even a question of who owns the data: If you visit a web page, do you own the fact that you clicked on a link, or does the web page owner, the Internet provider, someone else, or all of you?

In spite of these complicating examples, confidentiality is the security property we understand best because its meaning is narrower than that of the other two. We also understand confidentiality well because we can relate computing examples to those of preserving confidentiality in the real world.

Confidentiality relates most obviously to data, although we can think of the confidentiality of a piece of hardware (a novel invention) or a person (the whereabouts of a wanted criminal). Here are some properties that could mean a failure of data confidentiality:

- An unauthorized person accesses a data item.
- An unauthorized process or program accesses a data item.
- A person authorized to access certain data accesses other data not authorized (which is a specialized version of “an unauthorized person accesses a data item”).
- An unauthorized person accesses an approximate data value (for example, not knowing someone’s exact salary but knowing that the salary falls in a particular range or exceeds a particular amount).
- An unauthorized person learns the existence of a piece of data (for example, knowing that a company is developing a certain new product or that talks are underway about the merger of two companies).

Notice the general pattern of these statements: A person, process, or program is (or is not) authorized to access a data item in a particular way. We call the person, process, or program a subject, the data item an object, the kind of access (such as read, write, or execute) an access mode, and the authorization a policy, as shown in Figure 1-6. These four terms reappear throughout this book because they are fundamental aspects of computer security.

One word that captures most aspects of confidentiality is view, although you should not take that term literally. A failure of confidentiality does not necessarily mean that someone sees an object and, in fact, it is virtually impossible to look at bits in any meaningful way (although you may look at their representation as characters or pictures). The word view does connote another aspect of confidentiality in computer security, through the association with viewing a movie or a painting in a museum: look but do not touch. In computer security, confidentiality usually means obtaining but not modifying. Modification is the subject of integrity, which we consider in the next section.
Examples of integrity failures are easy to find. A number of years ago a malicious macro in a Word document inserted the word “not” after some random instances of the word “is;” you can imagine the havoc that ensued. Because the document was generally syntactically correct, people did not immediately detect the change. In another case, a model of the Pentium computer chip produced an incorrect result in certain circumstances of floating-point arithmetic. Although the circumstances of failure were rare, Intel decided to manufacture and replace the chips. Many of us receive mail that is misaddressed because someone typed something wrong when transcribing from a written list. A worse situation occurs when that inaccuracy is propagated to other mailing lists such that we can never seem to correct the root of the problem. Other times we find that a spreadsheet seems to be wrong, only to find that someone typed “space 123” in a cell, changing it from a numeric value to text, so the spreadsheet program misused that cell in computation. Suppose someone converted numeric data to roman numerals: One could argue that IV is the same as 4, but IV would not be useful in most applications, nor would it be obviously meaningful to someone expecting 4 as an answer. These cases show some of the breadth of examples of integrity failures.

Integrity is harder to pin down than confidentiality. As Stephen Welke and Terry Mayfield [WEL90, MAY91, NCS91a] point out, integrity means different things in different contexts. When we survey the way some people use the term, we find several
different meanings. For example, if we say that we have preserved the integrity of an item, we may mean that the item is

- precise
- accurate
- unmodified
- modified only in acceptable ways
- modified only by authorized people
- modified only by authorized processes
- consistent
- internally consistent
- meaningful and usable

Integrity can also mean two or more of these properties. Welke and Mayfield recognize three particular aspects of integrity—authorized actions, separation and protection of resources, and error detection and correction. Integrity can be enforced in much the same way as can confidentiality: by rigorous control of who or what can access which resources in what ways.

**Availability**

A computer user’s worst nightmare: You turn on the switch and the computer does nothing. Your data and programs are presumably still there, but you cannot get at them. Fortunately, few of us experience that failure. Many of us do experience overload, however: access gets slower and slower; the computer responds but not in a way we consider normal or acceptable.

Availability applies both to data and to services (that is, to information and to information processing), and it is similarly complex. As with the notion of confidentiality, different people expect availability to mean different things. For example, an object or service is thought to be available if the following are true:

- It is present in a usable form.
- It has enough capacity to meet the service’s needs.
- It is making clear progress, and, if in wait mode, it has a bounded waiting time.
- The service is completed in an acceptable period of time.

We can construct an overall description of availability by combining these goals. Following are some criteria to define availability.

- There is a timely response to our request.
- Resources are allocated fairly so that some requesters are not favored over others.
- Concurrency is controlled; that is, simultaneous access, deadlock management, and exclusive access are supported as required.
The service or system involved follows a philosophy of fault tolerance, whereby hardware or software faults lead to graceful cessation of service or to workarounds rather than to crashes and abrupt loss of information. (Cessation does mean end; whether it is graceful or not, ultimately the system is unavailable. However, with fair warning of the system’s stopping, the user may be able to move to another system and continue work.)

The service or system can be used easily and in the way it was intended to be used. (This is a characteristic of usability, but an unusable system may also cause an availability failure.)

As you can see, expectations of availability are far-reaching. In Figure 1-7 we depict some of the properties with which availability overlaps. Indeed, the security community is just beginning to understand what availability implies and how to ensure it.

A person or system can do three basic things with a data item: view it, modify it, or use it. Thus, viewing (confidentiality), modifying (integrity), and using (availability) are the basic modes of access that computer security seeks to preserve.

A paradigm of computer security is access control: To implement a policy, computer security controls all accesses by all subjects to all protected objects in all modes of access. A small, centralized control of access is fundamental to preserving confidentiality and integrity, but it is not clear that a single access control point can enforce availability. Indeed, experts on dependability will note that single points of control can become single points of failure, making it easy for an attacker to destroy availability by disabling the single control point. Much of computer security’s past success has focused on confidentiality and integrity; there are models of confidentiality and integrity, for

---

**Computer security seeks to prevent unauthorized viewing (confidentiality) or modification (integrity) of data while preserving access (availability).**

---

*FIGURE 1-7 Availability and Related Aspects*
example, see David Bell and Leonard La Padula [BEL73, BEL76] and Kenneth Biba [BIB77]. Availability is security’s next great challenge.

We have just described the C-I-A triad and the three fundamental security properties it represents. Our description of these properties was in the context of things that need protection. To motivate your understanding we gave some examples of harm and threats to cause harm. Our next step is to think about the nature of threats themselves.

**Types of Threats**

For some ideas of harm, look at Figure 1-8, taken from Willis Ware’s report [WAR70]. Although it was written when computers were so big, so expensive, and so difficult to operate that only large organizations like universities, major corporations, or government departments would have one, Ware’s discussion is still instructive today. Ware was concerned primarily with the protection of classified data, that is, preserving confidentiality. In the figure, he depicts humans such as programmers and maintenance staff gaining access to data, as well as radiation by which data can escape as signals. From the figure you can see some of the many kinds of threats to a computer system.

One way to analyze harm is to consider the cause or source. We call a potential cause of harm a **threat**. Harm can be caused by either nonhuman events or humans. Examples of **nonhuman threats** include natural disasters...
like fires or floods; loss of electrical power; failure of a component such as a communications cable, processor chip, or disk drive; or attack by a wild boar.

**Human threats** can be either benign (nonmalicious) or malicious. **Nonmalicious** kinds of harm include someone’s accidentally spilling a soft drink on a laptop, unintentionally deleting text, inadvertently sending an email message to the wrong person, and carelessly typing “12” instead of “21” when entering a phone number or clicking “yes” instead of “no” to overwrite a file. These inadvertent, human errors happen to most people; we just hope that the seriousness of harm is not too great, or if it is, that we will not repeat the mistake.

Most computer security activity relates to **malicious, human-caused harm**: A malicious person actually wants to cause harm, and so we often use the term *attack* for a malicious computer security event. Malicious attacks can be random or directed. In a **random attack** the attacker wants to harm any computer or user; such an attack is analogous to accosting the next pedestrian who walks down the street. An example of a random attack is malicious code posted on a website that could be visited by anybody.

In a **directed attack**, the attacker intends harm to specific computers, perhaps at one organization (think of attacks against a political organization) or belonging to a specific individual (think of trying to drain a specific person’s bank account, for example, by impersonation). Another class of directed attack is against a particular product, such as any computer running a particular browser. (We do not want to split hairs about whether such an attack is directed—at that one software product—or random, against any user of that product; the point is not semantic perfection but protecting against the attacks.) The range of possible directed attacks is practically unlimited. Different kinds of threats are shown in Figure 1-9.

Although the distinctions shown in Figure 1-9 seem clear-cut, sometimes the nature of an attack is not obvious until the attack is well underway, or perhaps even ended. A normal hardware failure can seem like a directed, malicious attack to deny access, and hackers often try to conceal their activity to look like ordinary, authorized users. As computer security experts we need to anticipate what bad things might happen, instead of waiting for the attack to happen or debating whether the attack is intentional or accidental.

Neither this book nor any checklist or method can show you *all* the kinds of harm that can happen to computer assets. There are too many ways to interfere with your use of these assets. Two retrospective lists of **known** vulnerabilities are of interest, however. The Common Vulnerabilities and Exposures (CVE) list (see http://cve.mitre.org/) is a dictionary of publicly known security vulnerabilities and exposures. CVE’s common identifiers enable data exchange between security products and provide a baseline index point for evaluating coverage of security tools and services. To measure the extent of harm, the Common Vulnerability Scoring System (CVSS) (see http://nvd.nist.gov/cvss.cfm) provides a standard measurement system that allows accurate and consistent scoring of vulnerability impact.
Advanced Persistent Threat

Security experts are becoming increasingly concerned about a type of threat called **advanced persistent threat**. A lone attacker might create a random attack that snares a few, or a few million, individuals, but the resulting impact is limited to what that single attacker can organize and manage. A collection of attackers—think, for example, of the cyber equivalent of a street gang or an organized crime squad—might work together to purloin credit card numbers or similar financial assets to fund other illegal activity. Such attackers tend to be opportunistic, picking unlucky victims’ pockets and moving on to other activities.

Advanced persistent threat attacks come from organized, well financed, patient assailants. Often affiliated with governments or quasi-governmental groups, these attackers engage in long term campaigns. They carefully select their targets, crafting attacks that appeal to specifically those targets; email messages called spear phishing (described in Chapter 4) are intended to seduce their recipients. Typically the attacks are silent, avoiding any obvious impact that would alert a victim, thereby allowing the attacker to exploit the victim’s access rights over a long time.

The motive of such attacks is sometimes unclear. One popular objective is economic espionage. A series of attacks, apparently organized and supported by the Chinese government, was used in 2012 and 2013 to obtain product designs from aerospace companies in the United States. There is evidence the stub of the attack code was loaded into victim machines long in advance of the attack; then, the attackers installed the more complex code and extracted the desired data. In May 2014 the Justice Department indicted five Chinese hackers in absentia for these attacks.
In the summer of 2014 a series of attacks against J.P. Morgan Chase bank and up to a dozen similar financial institutions allowed the assailants access to 76 million names, phone numbers, and email addresses. The attackers—and even their country of origin—remain unknown, as does the motive. Perhaps the attackers wanted more sensitive financial data, such as account numbers or passwords, but were only able to get the less valuable contact information. It is also not known if this attack was related to an attack a year earlier that disrupted service to that bank and several others.

To imagine the full landscape of possible attacks, you may find it useful to consider the kinds of people who attack computer systems. Although potentially anyone is an attacker, certain classes of people stand out because of their backgrounds or objectives. Thus, in the following sections we look at profiles of some classes of attackers.

**Types of Attackers**

Who are attackers? As we have seen, their motivations range from chance to a specific target. Putting aside attacks from natural and benign causes, we can explore who the attackers are and what motivates them.

Most studies of attackers actually analyze computer criminals, that is, people who have actually been convicted of a crime, primarily because that group is easy to identify and study. The ones who got away or who carried off an attack without being detected may have characteristics different from those of the criminals who have been caught. Worse, by studying only the criminals we have caught, we may not learn how to catch attackers who know how to abuse the system without being apprehended.

What does a cyber criminal look like? In television and films the villains wore shabby clothes, looked mean and sinister, and lived in gangs somewhere out of town. By contrast, the sheriff dressed well, stood proud and tall, was known and respected by everyone in town, and struck fear in the hearts of most criminals.

To be sure, some computer criminals are mean and sinister types. But many more wear business suits, have university degrees, and appear to be pillars of their communities. Some are high school or university students. Others are middle-aged business executives. Some are mentally deranged, overtly hostile, or extremely committed to a cause, and they attack computers as a symbol. Others are ordinary people tempted by personal profit, revenge, challenge, advancement, or job security—like perpetrators of any crime, using a computer or not. Researchers have tried to find the psychological traits that distinguish attackers, as described in Sidebar 1-1. These studies are far from conclusive, however, and the traits they identify may show correlation but not necessarily causality. To appreciate this point, suppose a study found that a disproportionate number of people convicted of computer crime were left-handed. Does that result imply that all left-handed people are computer criminals or that only left-handed people are? Certainly not. No single profile captures the characteristics of a “typical” computer attacker, and the characteristics of some notorious attackers also match many people who are not attackers. As shown in Figure 1-10, attackers look just like anybody in a crowd.

---

**No one pattern matches all attackers.**
SIDEBAR 1-1 An Attacker's Psychological Profile?

Temple Grandin, a professor of animal science at Colorado State University and a sufferer from a mental disorder called Asperger syndrome (AS), thinks that Kevin Mitnick and several other widely described hackers show classic symptoms of Asperger syndrome. Although quick to point out that no research has established a link between AS and hacking, Grandin notes similar behavior traits among Mitnick, herself, and other AS sufferers. An article in USA Today (29 March 2001) lists the following AS traits:

- poor social skills, often associated with being loners during childhood; the classic “computer nerd”
- fidgeting, restlessness, inability to make eye contact, lack of response to cues in social interaction, such as facial expressions or body language
- exceptional ability to remember long strings of numbers
- ability to focus on a technical problem intensely and for a long time, although easily distracted on other problems and unable to manage several tasks at once
- deep honesty and respect for laws

(continues)
SIDEBAR 1-1  Continued

Donn Parker [PAR98] has studied hacking and computer crime for many years. He states “hackers are characterized by an immature, excessively idealistic attitude . . . They delight in presenting themselves to the media as idealistic do-gooders, champions of the underdog.”

Consider the following excerpt from an interview [SHA00] with “Mix-ter,” the German programmer who admitted he was the author of a wide-spread piece of attack software called Tribal Flood Network (TFN) and its sequel TFN2K:

Q: Why did you write the software?
A: I first heard about Trin00 [another piece of attack software] in July ’99 and I considered it as interesting from a technical perspective, but also potentially powerful in a negative way. I knew some facts of how Trin00 worked, and since I didn’t manage to get Trin00 sources or binaries at that time, I wrote my own server-client network that was capable of performing denial of service.

Q: Were you involved . . . in any of the recent high-profile attacks?
A: No. The fact that I authored these tools does in no way mean that I condone their active use. I must admit I was quite shocked to hear about the latest attacks. It seems that the attackers are pretty clueless people who misuse powerful resources and tools for generally harmful and senseless activities just “because they can.”

Notice that from some information about denial-of-service attacks, he wrote his own server-client network and then a sophisticated attack. But he was “quite shocked” to hear they were used for harm.

More research is needed before we can define the profile of a hacker. And even more work will be needed to extend that profile to the profile of a (malicious) attacker. Not all hackers become attackers; some hackers become extremely dedicated and conscientious system administrators, developers, or security experts. But some psychologists see in AS the rudiments of a hacker’s profile.

Individuals

Originally, computer attackers were individuals, acting with motives of fun, challenge, or revenge. Early attackers acted alone. Two of the most well known among them are Robert Morris Jr., the Cornell University graduate student who brought down the Internet in 1988 [SPA89], and Kevin Mitnick, the man who broke into and stole data from dozens of computers, including the San Diego Supercomputer Center [MAR95].

Organized, Worldwide Groups

More recent attacks have involved groups of people. An attack against the government of the country of Estonia (described in more detail in Chapter 13) is believed to have been an uncoordinated outburst from a loose federation of attackers from around the world. Kevin Poulsen [POU05] quotes Tim Rosenberg, a research professor at George
Washington University, warning of “multinational groups of hackers backed by organized crime” and showing the sophistication of prohibition-era mobsters. He also reports that Christopher Painter, deputy director of the U.S. Department of Justice’s computer crime section, argues that cyber criminals and serious fraud artists are increasingly working in concert or are one and the same. According to Painter, loosely connected groups of criminals all over the world work together to break into systems and steal and sell information, such as credit card numbers. For instance, in October 2004, U.S. and Canadian authorities arrested 28 people from 6 countries involved in an international, organized cybercrime ring to buy and sell credit card information and identities.

Whereas early motives for computer attackers such as Morris and Mitnick were personal, such as prestige or accomplishment, recent attacks have been heavily influenced by financial gain. Security firm McAfee reports “Criminals have realized the huge financial gains to be made from the Internet with little risk. They bring the skills, knowledge, and connections needed for large scale, high-value criminal enterprise that, when combined with computer skills, expand the scope and risk of cybercrime.” [MCA05]

Organized Crime

Attackers’ goals include fraud, extortion, money laundering, and drug trafficking, areas in which organized crime has a well-established presence. Evidence is growing that organized crime groups are engaging in computer crime. In fact, traditional criminals are recruiting hackers to join the lucrative world of cybercrime. For example, Albert Gonzales was sentenced in March 2010 to 20 years in prison for working with a crime ring to steal 40 million credit card numbers from retailer TJMaxx and others, costing over $200 million (Reuters, 26 March 2010).

Organized crime may use computer crime (such as stealing credit card numbers or bank account details) to finance other aspects of crime. Recent attacks suggest that professional criminals have discovered just how lucrative computer crime can be. Mike Danseglio, a security project manager with Microsoft, said, “In 2006, the attackers want to pay the rent. They don’t want to write a worm that destroys your hardware. They want to assimilate your computers and use them to make money.” [NAR06a] Mikko Hyppönen, Chief Research Officer with Finnish security company f-Secure, agrees that today’s attacks often come from Russia, Asia, and Brazil; the motive is now profit, not fame [BRA06]. Ken Dunham, Director of the Rapid Response Team for VeriSign says he is “convinced that groups of well-organized mobsters have taken control of a global billion-dollar crime network powered by skillful hackers.” [NAR06b]

McAfee also describes the case of a hacker-for-hire: a businessman who hired a 16-year-old New Jersey hacker to attack the websites of his competitors. The hacker barraged the site for a five-month period and damaged not only the target companies but also their Internet service providers (ISPs) and other unrelated companies that used the same ISPs. By FBI estimates, the attacks cost all the companies over $2 million; the FBI arrested both hacker and businessman in March 2005 [MCA05].

Brian Snow [SNO05] observes that hackers want a score or some kind of evidence to give them bragging rights. Organized crime wants a resource; such criminals want to
stay under the radar to be able to extract profit from the system over time. These different objectives lead to different approaches to computer crime: The novice hacker can use a crude attack, whereas the professional attacker wants a neat, robust, and undetectable method that can deliver rewards for a long time.

**Terrorists**

The link between computer security and terrorism is quite evident. We see terrorists using computers in four ways:

- **Computer as target of attack:** Denial-of-service attacks and website defacements are popular activities for any political organization because they attract attention to the cause and bring undesired negative attention to the object of the attack. An example is the massive denial-of-service attack launched against the country of Estonia, detailed in Chapter 13.

- **Computer as method of attack:** Launching offensive attacks requires the use of computers. Stuxnet, an example of malicious computer code called a worm, is known to attack automated control systems, specifically a model of control system manufactured by Siemens. Experts say the code is designed to disable machinery used in the control of nuclear reactors in Iran [MAR10]. The persons behind the attack are unknown, but the infection is believed to have spread through USB flash drives brought in by engineers maintaining the computer controllers. (We examine the Stuxnet worm in more detail in Chapters 6 and 13.)

- **Computer as enabler of attack:** Websites, web logs, and email lists are effective, fast, and inexpensive ways to allow many people to coordinate. According to the Council on Foreign Relations, the terrorists responsible for the November 2008 attack that killed over 200 people in Mumbai used GPS systems to guide their boats, Blackberries for their communication, and Google Earth to plot their routes.

- **Computer as enhancer of attack:** The Internet has proved to be an invaluable means for terrorists to spread propaganda and recruit agents. In October 2009 the FBI arrested Colleen LaRose, also known as JihadJane, after she had spent months using email, YouTube, MySpace, and electronic message boards to recruit radicals in Europe and South Asia to “wage violent jihad,” according to a federal indictment.

We cannot accurately measure the degree to which terrorists use computers, because terrorists keep secret the nature of their activities and because our definitions and measurement tools are rather weak. Still, incidents like the one described in Sidebar 1-2 provide evidence that all four of these activities are increasing.

**SIDEBAR 1-2 The Terrorists, Inc., IT Department**

In 2001, a reporter for the *Wall Street Journal* bought a used computer in Afghanistan. Much to his surprise, he found that the hard drive contained what appeared to be files from a senior al Qaeda operative. The reporter,
Alan Cullison [CUL04], reports that he turned the computer over to the FBI. In his story published in 2004 in *The Atlantic*, he carefully avoids revealing anything he thinks might be sensitive.

The disk contained over 1,000 documents, many of them encrypted with relatively weak encryption. Cullison found draft mission plans and white papers setting forth ideological and philosophical arguments for the attacks of 11 September 2001. Also found were copies of news stories on terrorist activities. Some of the found documents indicated that al Qaeda was not originally interested in chemical, biological, or nuclear weapons, but became interested after reading public news articles accusing al Qaeda of having those capabilities.

Perhaps most unexpected were email messages of the kind one would find in a typical office: recommendations for promotions, justifications for petty cash expenditures, and arguments concerning budgets.

The computer appears to have been used by al Qaeda from 1999 to 2001. Cullison notes that Afghanistan in late 2001 was a scene of chaos, and it is likely the laptop’s owner fled quickly, leaving the computer behind, where it fell into the hands of a secondhand goods merchant who did not know its contents.

But this computer’s contents illustrate an important aspect of computer security and confidentiality: We can never predict the time at which a security disaster will strike, and thus we must always be prepared to act immediately if it suddenly happens.

If someone on television sneezes, you do not worry about the possibility of catching a cold. But if someone standing next to you sneezes, you may become concerned. In the next section we examine the harm that can come from the presence of a computer security threat on your own computer systems.

1.3 HARM

The negative consequence of an actualized threat is harm; we protect ourselves against threats in order to reduce or eliminate harm. We have already described many examples of computer harm: a stolen computer, modified or lost file, revealed private letter, or denied access to data. These events cause harm that we want to avoid.

In our earlier discussion of assets, we noted that value depends on owner or outsider perception and need. Some aspects of value are immeasurable, such as the value of the paper you need to submit to your professor tomorrow; if you lose the paper (that is, if its availability is lost), no amount of money will compensate you for it. Items on which you place little or no value might be more valuable to someone else; for example, the group photograph taken at last night’s party can reveal that your friend was not where he told his wife he would be. Even though it may be difficult to assign a specific number as the value of an asset, you can usually assign a value on a generic scale, such as moderate or minuscule or incredibly high, depending on the degree of harm that loss or damage to the object would cause. Or you can assign a value relative to other assets,
based on comparable loss: This version of the file is more valuable to you than that version.

In their 2010 global Internet threat report, security firm Symantec surveyed the kinds of goods and services offered for sale on underground web pages. The item most frequently offered in both 2009 and 2008 was credit card numbers, at prices ranging from $0.85 to $30.00 each. (Compare those prices to an individual’s effort to deal with the effect of a stolen credit card or the potential amount lost by the issuing bank.) Second most frequent was bank account credentials, at $15 to $850; these were offered for sale at 19% of websites in both years. Email accounts were next at $1 to $20, and lists of email addresses went for $1.70 to $15.00 per thousand. At position 10 in 2009 were website administration credentials, costing only $2 to $30. These black market websites demonstrate that the market price of computer assets can be dramatically different from their value to rightful owners.

The value of many assets can change over time, so the degree of harm (and therefore the severity of a threat) can change, too. With unlimited time, money, and capability, we might try to protect against all kinds of harm. But because our resources are limited, we must prioritize our protection, safeguarding only against serious threats and the ones we can control. Choosing the threats we try to mitigate involves a process called risk management, and it includes weighing the seriousness of a threat against our ability to protect.

Risk management involves choosing which threats to control and what resources to devote to protection.

Risk and Common Sense

The number and kinds of threats are practically unlimited because devising an attack requires an active imagination, determination, persistence, and time (as well as access and resources). The nature and number of threats in the computer world reflect life in general: The causes of harm are limitless and largely unpredictable. Natural disasters like volcanoes and earthquakes happen with little or no warning, as do auto accidents, heart attacks, influenza, and random acts of violence. To protect against accidents or the flu, you might decide to stay indoors, never venturing outside. But by doing so, you trade one set of risks for another; while you are inside, you are vulnerable to building collapse. There are too many possible causes of harm for us to protect ourselves—or our computers—completely against all of them.

In real life we make decisions every day about the best way to provide our security. For example, although we may choose to live in an area that is not prone to earthquakes, we cannot entirely eliminate earthquake risk. Some choices are conscious, such as deciding not to walk down a dark alley in an unsafe neighborhood; other times our subconscious guides us, from experience or expertise, to take some precaution. We evaluate the likelihood and severity of harm, and then consider ways (called countermeasures or controls) to address threats and determine the controls’ effectiveness.

Computer security is similar. Because we cannot protect against everything, we prioritize: Only so much time, energy, or money is available for protection, so we address
some risks and let others slide. Or we consider alternative courses of action, such as transferring risk by purchasing insurance or even doing nothing if the side effects of the countermeasure could be worse than the possible harm. The risk that remains uncovered by controls is called **residual risk**.

A basic model of risk management involves a user’s calculating the value of all assets, determining the amount of harm from all possible threats, computing the costs of protection, selecting safeguards (that is, controls or countermeasures) based on the degree of risk and on limited resources, and applying the safeguards to optimize harm averted. This approach to risk management is a logical and sensible approach to protection, but it has significant drawbacks. In reality, it is difficult to assess the value of each asset; as we have seen, value can change depending on context, timing, and a host of other characteristics. Even harder is determining the impact of all possible threats. The range of possible threats is effectively limitless, and it is difficult (if not impossible in some situations) to know the short- and long-term impacts of an action. For instance, Sidebar 1-3 describes a study of the impact of security breaches over time on corporate finances, showing that a threat must be evaluated over time, not just at a single instance.

**SIDEBAR 1-3  Short- and Long-term Risks of Security Breaches**

It was long assumed that security breaches would be bad for business: that customers, fearful of losing their data, would veer away from insecure businesses and toward more secure ones. But empirical studies suggest that the picture is more complicated. Early studies of the effects of security breaches, such as that of Campbell [CAM03], examined the effects of breaches on stock price. They found that a breach’s impact could depend on the nature of the breach itself; the effects were higher when the breach involved unauthorized access to confidential data. Cavusoglu et al. [CAV04] discovered that a breach affects the value not only of the company experiencing the breach but also of security enterprises: On average, the breached firms lost 2.1 percent of market value within two days of the breach’s disclosure, but security developers’ market value actually increased 1.36 percent.

Myung Ko and Carlos Dorantes [KO06] looked at the longer-term financial effects of publicly announced breaches. Based on the Campbell et al. study, they examined data for four quarters following the announcement of unauthorized access to confidential data. Ko and Dorantes note many types of possible breach-related costs:

*Examples of short-term costs include cost of repairs, cost of replacement of the system, lost business due to the disruption of business operations, and lost productivity of employees. These are also considered tangible costs. On the other hand, long-term costs include the loss of existing customers due to loss of trust, failing to attract potential future customers due to negative reputation*
from the breach, loss of business partners due to loss of trust, and potential legal liabilities from the breach. Most of these costs are intangible costs that are difficult to calculate but extremely important in assessing the overall security breach costs to the organization.

Ko and Dorantes compared two groups of companies: one set (the treatment group) with data breaches, and the other (the control group) without a breach but matched for size and industry. Their findings were striking. Contrary to what you might suppose, the breached firms had no decrease in performance for the quarters following the breach, but their return on assets decreased in the third quarter. The comparison of treatment with control companies revealed that the control firms generally outperformed the breached firms. However, the breached firms outperformed the control firms in the fourth quarter.

These results are consonant with the results of other researchers who conclude that there is minimal long-term economic impact from a security breach. There are many reasons why this is so. For example, customers may think that all competing firms have the same vulnerabilities and threats, so changing to another vendor does not reduce the risk. Another possible explanation may be a perception that a breached company has better security since the breach forces the company to strengthen controls and thus reduce the likelihood of similar breaches. Yet another explanation may simply be the customers’ short attention span; as time passes, customers forget about the breach and return to business as usual.

All these studies have limitations, including small sample sizes and lack of sufficient data. But they clearly demonstrate the difficulties of quantifying and verifying the impacts of security risks, and point out a difference between short- and long-term effects.

Although we should not apply protection haphazardly, we will necessarily protect against threats we consider most likely or most damaging. For this reason, it is essential to understand how we perceive threats and evaluate their likely occurrence and impact. Sidebar 1-4 summarizes some of the relevant research in risk perception and decision-making. Such research suggests that, for relatively rare instances such as high-impact security problems, we must take into account the ways in which people focus more on the impact than on the actual likelihood of occurrence.

**SIDEBAR 1-4  Perception of the Risk of Extreme Events**

When a type of adverse event happens frequently, we can calculate its likelihood and impact by examining both frequency and nature of the collective set of events. For instance, we can calculate the likelihood that it will
rain this week and take an educated guess at the number of inches of precipitation we will receive; rain is a fairly frequent occurrence. But security problems are often extreme events: They happen infrequently and under a wide variety of circumstances, so it is difficult to look at them as a group and draw general conclusions.

Paul Slovic’s work on risk addresses the particular difficulties with extreme events. He points out that evaluating risk in such cases can be a political endeavor as much as a scientific one. He notes that we tend to let values, process, power, and trust influence our risk analysis [SLO99].

Beginning with Fischoff et al. [FIS78], researchers characterized extreme risk along two perception-based axes: the dread of the risk and the degree to which the risk is unknown. These feelings about risk, called affects by psychologists, enable researchers to discuss relative risks by placing them on a plane defined by the two perceptions as axes. A study by Loewenstein et al. [LOE01] describes how risk perceptions are influenced by association (with events already experienced) and by affect at least as much if not more than by reason. In fact, if the two influences compete, feelings usually trump reason.

This characteristic of risk analysis is reinforced by prospect theory: studies of how people make decisions by using reason and feeling. Kahneman and Tversky [KAH79] showed that people tend to overestimate the likelihood of rare, unexperienced events because their feelings of dread and the unknown usually dominate analytical reasoning about the low likelihood of occurrence. By contrast, if people experience similar outcomes and their likelihood, their feeling of dread diminishes and they can actually underestimate rare events. In other words, if the impact of a rare event is high (high dread), then people focus on the impact, regardless of the likelihood. But if the impact of a rare event is small, then they pay attention to the likelihood.

Let us look more carefully at the nature of a security threat. We have seen that one aspect—its potential harm—is the amount of damage it can cause; this aspect is the impact component of the risk. We also consider the magnitude of the threat’s likelihood. A likely threat is not just one that someone might want to pull off but rather one that could actually occur. Some people might daydream about getting rich by robbing a bank; most, however, would reject that idea because of its difficulty (if not its immorality or risk). One aspect of likelihood is feasibility: Is it even possible to accomplish the attack? If the answer is no, then the likelihood is zero, and therefore so is the risk. So a good place to start in assessing risk is to look at whether the proposed action is feasible. Three factors determine feasibility, as we describe next.
Method–Opportunity–Motive

A malicious attacker must have three things to ensure success: method, opportunity, and motive, depicted in Figure 1-11. Roughly speaking, method is the how; opportunity, the when; and motive, the why of an attack. Deny the attacker any of those three and the attack will not succeed. Let us examine these properties individually.

Method

By method we mean the skills, knowledge, tools, and other things with which to perpetrate the attack. Think of comic figures that want to do something, for example, to steal valuable jewelry, but the characters are so inept that their every move is doomed to fail. These people lack the capability or method to succeed, in part because there are no classes in jewel theft or books on burglary for dummies.

Anyone can find plenty of courses and books about computing, however. Knowledge of specific models of computer systems is widely available in bookstores and on
the Internet. Mass-market systems (such as the Microsoft or Apple or Unix operating systems) are readily available for purchase, as are common software products, such as word processors or database management systems, so potential attackers can even get hardware and software on which to experiment and perfect an attack. Some manufacturers release detailed specifications on how the system was designed or how it operates, as guides for users and integrators who want to implement other complementary products. Various attack tools—scripts, model programs, and tools to test for weaknesses—are available from hackers’ sites on the Internet, to the degree that many attacks require only the attacker’s ability to download and run a program. The term script kid
die describes someone who downloads a complete attack code package and needs only to enter a few details to identify the target and let the script perform the attack. Often, only time and inclination limit an attacker.

**Opportunity**

**Opportunity** is the time and access to execute an attack. You hear that a fabulous apartment has just become available, so you rush to the rental agent, only to find someone else rented it five minutes earlier. You missed your opportunity.

Many computer systems present ample opportunity for attack. Systems available to the public are, by definition, accessible; often their owners take special care to make them fully available so that if one hardware component fails, the owner has spares instantly ready to be pressed into service. Other people are oblivious to the need to protect their computers, so unattended laptops and unsecured network connections give ample opportunity for attack. Some systems have private or undocumented entry points for administration or maintenance, but attackers can also find and use those entry points to attack the systems.

**Motive**

Finally, an attacker must have a motive or reason to want to attack. You probably have ample opportunity and ability to throw a rock through your neighbor’s window, but you do not. Why not? Because you have no reason to want to harm your neighbor: You lack motive.

We have already described some of the motives for computer crime: money, fame, self-esteem, politics, terror. It is often difficult to determine motive for an attack. Some places are “attractive targets,” meaning they are very appealing to attackers. Popular targets include law enforcement and defense department computers, perhaps because they are presumed to be well protected against attack (so they present a challenge and a successful attack shows the attacker’s prowess). Other systems are attacked because they are easy to attack. And some systems are attacked at random simply because they are there.

---

Method, opportunity, and motive are all necessary for an attack to succeed; deny any of these and the attack will fail.
By demonstrating feasibility, the factors of method, opportunity, and motive determine whether an attack can succeed. These factors give the advantage to the attacker because they are qualities or strengths the attacker must possess. Another factor, this time giving an advantage to the defender, determines whether an attack will succeed: The attacker needs a vulnerability, an undefended place to attack. If the defender removes vulnerabilities, the attacker cannot attack.

1.4 VULNERABILITIES

As we noted earlier in this chapter, a vulnerability is a weakness in the security of the computer system, for example, in procedures, design, or implementation, that might be exploited to cause loss or harm. Think of a bank, with an armed guard at the front door, bulletproof glass protecting the tellers, and a heavy metal vault requiring multiple keys for entry. To rob a bank, you would have to think of how to exploit a weakness not covered by these defenses. For example, you might bribe a teller or pose as a maintenance worker.

Computer systems have vulnerabilities, too. In this book we consider many, such as weak authentication, lack of access control, errors in programs, finite or insufficient resources, and inadequate physical protection. Paired with a credible attack, each of these vulnerabilities can allow harm to confidentiality, integrity, or availability. Each attack vector seeks to exploit a particular vulnerability.

Security analysts speak of a system’s attack surface, which is the system’s full set of vulnerabilities—actual and potential. Thus, the attack surface includes physical hazards, malicious attacks by outsiders, stealth data theft by insiders, mistakes, and impersonations. Although such attacks range from easy to highly improbable, analysts must consider all possibilities.

Our next step is to find ways to block threats by neutralizing vulnerabilities.

1.5 CONTROLS

A control or countermeasure is a means to counter threats. Harm occurs when a threat is realized against a vulnerability. To protect against harm, then, we can neutralize the threat, close the vulnerability, or both. The possibility for harm to occur is called risk. We can deal with harm in several ways:

- **prevent** it, by blocking the attack or closing the vulnerability
- **deter** it, by making the attack harder but not impossible
- **deflect** it, by making another target more attractive (or this one less so)
- **mitigate** it, by making its impact less severe
- **detect** it, either as it happens or some time after the fact
- **recover** from its effects

---

**Vulnerabilities are weaknesses that can allow harm to occur.**
Of course, more than one of these controls can be used simultaneously. So, for example, we might try to prevent intrusions—but if we suspect we cannot prevent all of them, we might also install a detection device to warn of an imminent attack. And we should have in place incident-response procedures to help in the recovery in case an intrusion does succeed.

To consider the controls or countermeasures that attempt to prevent exploiting a computing system’s vulnerabilities, we begin by thinking about traditional ways to enhance physical security. In the Middle Ages, castles and fortresses were built to protect the people and valuable property inside. The fortress might have had one or more security characteristics, including

- a strong gate or door to repel invaders
- heavy walls to withstand objects thrown or projected against them
- a surrounding moat to control access
- arrow slits to let archers shoot at approaching enemies
- crenellations to allow inhabitants to lean out from the roof and pour hot or vile liquids on attackers
- a drawbridge to limit access to authorized people
- a portcullis to limit access beyond the drawbridge
- gatekeepers to verify that only authorized people and goods could enter

Similarly, today we use a multipronged approach to protect our homes and offices. We may combine strong locks on the doors with a burglar alarm, reinforced windows, and even a nosy neighbor to keep an eye on our valuables. In each case, we select one or more ways to deter an intruder or attacker, and we base our selection not only on the value of what we protect but also on the effort we think an attacker or intruder will expend to get inside.

Computer security has the same characteristics. We have many controls at our disposal. Some are easier than others to use or implement. Some are cheaper than others to use or implement. And some are more difficult than others for intruders to override. Figure 1-12 illustrates how we use a combination of controls to secure our valuable resources. We use one or more controls, according to what we are protecting, how the cost of protection compares with the risk of loss, and how hard we think intruders will work to get what they want.

In this section, we present an overview of the controls available to us. In the rest of this book, we examine how to use controls against specific kinds of threats.

We can group controls into three largely independent classes. The following list shows the classes and several examples of each type of control.

- **Physical controls**: stop or block an attack by using something tangible too, such as walls and fences
  - locks
– (human) guards
– sprinklers and other fire extinguishers

- **Procedural** or **administrative** controls use a command or agreement that
  – requires or advises people how to act; for example,
  – laws, regulations
  – policies, procedures, guidelines
  – copyrights, patents
  – contracts, agreements

- **Technical controls** counter threats with technology (hardware or software), including
  – passwords
  – program or operating system access controls
  – network protocols
  – firewalls, intrusion detection systems
  – encryption
  – network traffic flow regulators

(Note that the term “logical controls” is also used, but some people use it to mean administrative controls, whereas others use it to mean technical controls. To avoid confusion, we do not use that term.)

As shown in Figure 1-13, you can think in terms of the property to be protected and the kind of threat when you are choosing appropriate types of countermeasures. None of these classes is necessarily better than or preferable to the others; they work in different ways with different kinds of results. And it can be effective to use **overlapping controls** or **defense in depth**: more than one control or more than one class of control to achieve protection.
1.6 CONCLUSION

Computer security attempts to ensure the confidentiality, integrity, and availability of computing systems and their components. Three principal parts of a computing system are subject to attacks: hardware, software, and data. These three, and the communications among them, are susceptible to computer security vulnerabilities. In turn, those people and systems interested in compromising a system can devise attacks that exploit the vulnerabilities.

In this chapter we have explained the following computer security concepts:

- Security situations arise in many everyday activities, although sometimes it can be difficult to distinguish between a security attack and an ordinary human or technological breakdown. Alas, clever attackers realize this confusion, so they may make their attack seem like a simple, random failure.

- A threat is an incident that could cause harm. A vulnerability is a weakness through which harm could occur. These two problems combine: Either without the other causes no harm, but a threat exercising a vulnerability means damage. To control such a situation, we can either block or diminish the threat, or close the vulnerability (or both).

- Seldom can we achieve perfect security: no viable threats and no exercisable vulnerabilities. Sometimes we fail to recognize a threat, or other times we may be unable or unwilling to close a vulnerability. Incomplete security is not a bad situation; rather, it demonstrates a balancing act: Control certain threats and vulnerabilities, apply countermeasures that are reasonable, and accept the risk of harm from uncountered cases.
An attacker needs three things: method—the skill and knowledge to perform a successful attack; opportunity—time and access by which to attack; and motive—a reason to want to attack. Alas, none of these three is in short supply, which means attacks are inevitable.

In this chapter we have introduced the notions of threats and harm, vulnerabilities, attacks and attackers, and countermeasures. Attackers leverage threats that exploit vulnerabilities against valuable assets to cause harm, and we hope to devise countermeasures to eliminate means, opportunity, and motive. These concepts are the basis we need to study, understand, and master computer security.

Countermeasures and controls can be applied to the data, the programs, the system, the physical devices, the communications links, the environment, and the personnel. Sometimes several controls are needed to cover a single vulnerability, but sometimes one control addresses many problems at once.

1.7 WHAT'S NEXT?

The rest of this book is organized around the major aspects or pieces of computer security. As you have certainly seen in almost daily news reports, computer security incidents abound. The nature of news is that failures are often reported, but seldom successes. You almost never read a story about hackers who tried to break into the computing system of a bank but were foiled because the bank had installed strong, layered defenses. In fact, attacks repelled far outnumber those that succeed, but such good situations do not make interesting news items.

Still, we do not want to begin with examples in which security controls failed. Instead, in Chapter 2 we begin by giving you descriptions of three powerful and widely used security protection methods. We call these three our security toolkit, in part because they are effective but also because they are applicable. We refer to these techniques in probably every other chapter of this book, so we want not only to give them a prominent position up front but also to help lodge them in your brain. Our three featured tools are identification and authentication, access control, and encryption.

After presenting these three basic tools we lead into domains in which computer security applies. We begin with the simplest computer situations, individual programs, and explore the problems and protections of computer code in Chapter 3. We also consider malicious code, such as viruses and Trojan horses (defining those terms along with other types of harmful programs). As you will see in other ways, there is no magic that can make bad programs secure or turn programmers into protection gurus. We do, however, point out some vulnerabilities that show up in computer code and describe ways to counter those weaknesses, both during program development and as a program executes.

Modern computing involves networking, especially using the Internet. We focus first on how networked computing affects individuals, primarily through browsers and other basic network interactions such as email. In Chapter 4 we look at how users can be tricked by skillful writers of malicious code. These attacks tend to affect the protection of confidentiality of users’ data and integrity of their programs.
Chapter 5 covers operating systems, continuing our path of moving away from things the user can see and affect directly. We see what protections operating systems can provide to users’ programs and data, most often against attacks on confidentiality or integrity. We also see how the strength of operating systems can be undermined by attacks, called rootkits, that directly target operating systems and render them unable to protect themselves or their users.

In Chapter 6 we return to networks, this time looking at the whole network and its impact on users’ abilities to communicate data securely across the network. We also study a type of attack called denial of service, just what its name implies, that is the first major example of a failure of availability.

We consider data, databases, and data mining in Chapter 7. The interesting cases involve large databases in which confidentiality of individuals’ private data is an objective. Integrity of the data in the databases is also a significant concern.

In Chapter 8 we move even further from the individual user and study cloud computing, a technology becoming quite popular. Companies are finding it convenient and cost effective to store data “in the cloud,” and individuals are doing the same to have shared access to things such as music and photos. There are security risks involved in this movement, however.

You may have noticed our structure: We organize our presentation from the user outward through programs, browsers, operating systems, networks, and the cloud, a progression from close to distant. In Chapter 9 we return to the user for a different reason: We consider privacy, a property closely related to confidentiality. Our treatment here is independent of where the data are: on an individual computer, a network, or a database. Privacy is a property we as humans deserve, and computer security can help preserve it, as we present in that chapter.

In Chapter 10 we look at several topics of management of computing as related to security. Security incidents occur, and computing installations need to be ready to respond, whether the cause is a hacker attack, software catastrophe, or fire. Managers also have to decide what controls to employ, because countermeasures cost money that must be spent wisely. Computer security protection is hard to evaluate: When it works you do not know it does. Performing risk analysis and building a case for security are important management tasks.

Some security protections are beyond the scope an individual can address. Organized crime from foreign countries is something governments must deal with, through a legal system. In Chapter 11 we consider laws affecting computer security. We also look at ethical standards, what is “right” in computing.

In Chapter 12 we return to cryptography, which we introduced in Chapter 2. Cryptography merits courses and textbooks of its own, and the topic is detailed enough that most of the real work in the field is done at the graduate level and beyond. We use Chapter 2 to introduce the concepts enough to be able to apply them. In Chapter 12 we expand upon that introduction and peek at some of the formal and mathematical underpinnings of cryptography.

Finally, in Chapter 13 we raise four topic areas. These are domains with an important need for computer security, although the areas are evolving so rapidly that computer
security may not be addressed as fully as it should. These areas are the so-called Internet of Things (the interconnection of network-enabled devices from toasters to automobiles and insulin pumps), computer security economics, electronic voting, and computer-assisted terrorism and warfare.

We trust this organization will help you to appreciate the richness of an important field that touches many of the things we depend on.

1.8 EXERCISES

1. Distinguish between vulnerability, threat, and control.

2. Theft usually results in some kind of harm. For example, if someone steals your car, you may suffer financial loss, inconvenience (by losing your mode of transportation), and emotional upset (because of invasion of your personal property and space). List three kinds of harm a company might experience from theft of computer equipment.

3. List at least three kinds of harm a company could experience from electronic espionage or unauthorized viewing of confidential company materials.

4. List at least three kinds of damage a company could suffer when the integrity of a program or company data is compromised.

5. List at least three kinds of harm a company could encounter from loss of service, that is, failure of availability. List the product or capability to which access is lost, and explain how this loss hurts the company.

6. Describe a situation in which you have experienced harm as a consequence of a failure of computer security. Was the failure malicious or not? Did the attack target you specifically or was it general and you were the unfortunate victim?

7. Describe two examples of vulnerabilities in automobiles for which auto manufacturers have instituted controls. Tell why you think these controls are effective, somewhat effective, or ineffective.

8. One control against accidental software deletion is to save all old versions of a program. Of course, this control is prohibitively expensive in terms of cost of storage. Suggest a less costly control against accidental software deletion. Is your control effective against all possible causes of software deletion? If not, what threats does it not cover?

9. On your personal computer, who can install programs? Who can change operating system data? Who can replace portions of the operating system? Can any of these actions be performed remotely?

10. Suppose a program to print paychecks secretly leaks a list of names of employees earning more than a certain amount each month. What controls could be instituted to limit the vulnerability of this leakage?

11. Preserving confidentiality, integrity, and availability of data is a restatement of the concern over interruption, interception, modification, and fabrication. How do the first three concepts relate to the last four? That is, is any of the four equivalent to one or more of the three? Is one of the three encompassed by one or more of the four?

12. Do you think attempting to break in to (that is, obtain access to or use of) a computing system without authorization should be illegal? Why or why not?
13. Describe an example (other than the ones mentioned in this chapter) of data whose confidentiality has a short timeliness, say, a day or less. Describe an example of data whose confidentiality has a timeliness of more than a year.

14. Do you currently use any computer security control measures? If so, what? Against what attacks are you trying to protect?

15. Describe an example in which absolute denial of service to a user (that is, the user gets no response from the computer) is a serious problem to that user. Describe another example where 10 percent denial of service to a user (that is, the user’s computation progresses, but at a rate 10 percent slower than normal) is a serious problem to that user. Could access by unauthorized people to a computing system result in a 10 percent denial of service to the legitimate users? How?

16. When you say that software is of high quality, what do you mean? How does security fit in your definition of quality? For example, can an application be insecure and still be “good”?

17. Developers often think of software quality in terms of faults and failures. Faults are problems (for example, loops that never terminate or misplaced commas in statements) that developers can see by looking at the code. Failures are problems, such as a system crash or the invocation of the wrong function, that are visible to the user. Thus, faults can exist in programs but never become failures, because the conditions under which a fault becomes a failure are never reached. How do software vulnerabilities fit into this scheme of faults and failures? Is every fault a vulnerability? Is every vulnerability a fault?

18. Consider a program to display on your website your city’s current time and temperature. Who might want to attack your program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?

19. Consider a program that allows consumers to order products from the web. Who might want to attack the program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?

20. Consider a program to accept and tabulate votes in an election. Who might want to attack the program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?

21. Consider a program that allows a surgeon in one city to assist in an operation on a patient in another city via an Internet connection. Who might want to attack the program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?
Index

1×1 GIF, 254, 628
802.11 protocols (WiFi), 376

Abstraction, in operating system, 289
Acceptance testing, 211
Access
  blocked, 399
  concurrent, 11
  controlled, 283
  exclusiveness, 11
  log, 74
  mediation, 152, 155
  mode, 72
  point, promiscuous, 386
  point, undocumented, 157
  point, wireless, 376
  rights of individuals, 603
  simultaneous, 11
  theft of, 750
  to data, 8
  to data, as asset, 3
  tracking, 546
  unauthorized physical, 689
Access control, 12, 72, 76, 78, 815
  capability, 82
  directory, 76 82
  ticket, 82
  database, 506, 508, 511
  device, 284
  failure of, 152, 155
  file, 284
  general object, 284
  granularity of, 287, 511
  list, 80, 292
  matrix, 78
  paradigm, 292
  physical, 690
  privacy, 594, 608
  policy, 9
  privilege list, 82
  procedure-oriented, 85
  propagation, 77, 83
  revocation, 76
  role-based, 85–86
  segmentation, 303
Access log, 74
Accountability
  for private data collection, 597
  of asset, 7
  versus privacy, 641
Accumulation, of data, 613
Accuracy, 55, 62, 242, 488, 531
  authentication, 56
  data, 11, 827
  data collection, 599
  elections, 840
  element, database, 513
  risk analysis, 685
ACL, see Access control list
Activation, process, 320
Ad, third-party, 622
Add subkey, in AES, 790
Add-in, browser, 232
Add-on, security as an, 315
Address
  hiding, 303
  resolution, 414
  space randomization, 210
  spoofing, 413
  translation, page, 306
  translation, segment, 303
Addressing 418, 434, 446
  advertising routing, 410
  failure of, 408
  network, 351
  stealth mode, 487
Adelman, Leonard, 103, 795
Administration, of access control, 73
Administrator, database, 502
Administrator, system, 358
Advanced Encryption Standard (AES), see AES
Advanced Persistent Threat, 15
Advertiser, 626
Advertising, and privacy, 629
Advertising, Internet, 622
Adware, 170, 629
Aerospace industry, target of attack, 15
AES (Advanced Encryption Standard), 98, 393, 439, 779, 789, 803
cycle in, 99
key length in, 109
speed of encryption, 103
tamper detection, 113
Agent, software, 474
Aggregation, 246
data, 640
database, 526
data mining, 537
personal data, 623
privacy, 618
AH (Authentication Header), see Authentication Header
Al Qaeda, 20–21
Alarm, 691
Alarm, intrusion detection system, 483, 484
Aleph One, see Levy, Elias
Algebraic inference, database, 525
Algorithm, encryption, 88
Algorithm weakness attack, against encryption, 770
Alias, email, 632
Allocation,
device, 281
resource, 286
Alteration, digital signature, 802
Alureon, 334, 336
Amazon GovCloud, 556
Amazon.com, Inc., 631
Ames, Stan, 312
Analysis, of problem, 816
Analysis, risk, see Risk analysis
Analyzer, code, 150
Anderson, James P., 7, 75, 172, 318, 733
Android, 818
Annenberg Public Policy Center, 631
Anomaly-based intrusion detection system, 476
Anonymity, 613
e-mail, 634
Internet, 620
network, 355
partial, 606
privacy, 605
Anonymization, 613, 615
data, 608
Hadoop, 545
privacy, 597
Antivirus tool, 191, 329
AOL, 527
Apache Hadoop, 542
API, 211
App store, 819
App, 819
review of, 819
signed, 819
Appended virus, 181
Apple Corp., 818
iCloud, 559
Mac OS operating system, 291, 302
SSL bug, 213
Application Programming Interface (API), 211
Application proxy firewall, 459, 468
Application whitelisting, 581
Application-based network attack, 398
Approximate value, of data, 9
Architecture
network, 470
system, 450
tagged, 301, 305
Arithmetic inference, database, 522
ARPANET, 143, 508
Asia, 19
Asperger syndrome, 17
Aspidistra, 107
Assembler language, 150
Assessment, situation, 488
Asset, 2
access as, 3
accountability of, 7
auditability of, 7
authenticity of, 7
critical, 660
data as, 3
fabrication of, 8
decision-making, 826
hardware as, 3
harm to, 5, 8
intellectual property as, 3
interception of, 8
modification of, 8
nonrepudiation of, 7
property as, 3
risk analysis of, 671
software as, 3
timeliness of, 4
use of, 7
value of, 4, 5, 6, 21
viewing of, 7
Association,
hijacking, 386
preferred, in wireless network, 386
WiFi, 380, 383
Assurance, 76, 820
Common Criteria, 327
level, 327
operating system design, 312
Asymmetric cryptography, 89, 93, 795
digital signature, 114
key exchange using, 105, 107
signing, 116
Attachment, of malicious code, 178
Attachment, virus, 188
Attack, 6
capability, 26
data driven, 148
denial of service, see Denial of service
directed, 14, 19, 423
feasibility of, 27–28
foiled, 32
malicious code, 166. See also Malicious code
man-in-the-middle, 106
method, 26
multistep, 148
of honeypot, 295
predicting, 826
random, 14
reason for, 16
source of, 828
surface, 28
targeted, 14, 19, 423
toolkit, 166, 424
type, classifying, 829
web site defacement, 20
zero-day, 172
Attacker, 18
characteristics of, 16
motivation of, 16
profile of, 16, 17
psychology of, 17
Attractive target, 27
Attribute
database, 504
database, hidden, 528
personal, for authentication, 611
Attribution, of attack source, 843, 844
Audit
Asset auditability, 7
balloting, 840
big data, 546
database, 507, 510
log, 74, 292
privacy, 608
Australia, 848
Authentication, 38, 108, 240, 569, 610, 816
attribute, 611
biometrics, 53
challenge–response, 461
center, 241
continuous, 245, 817
cookie, as basis for, 65
database, 507, 512
distributed, 357
DNA for, 61
Extensible Authentication Protocol, 393
forgery in, 58–59
Header, in IPsec, 445
human, 240
incomplete, 394
IPsec, 446
knowledge, as basis for, 40
MAC address, used for, 377
multifactor, 70
network communication, 445
nonexistent, in wireless network, 390
one-time password, 244
operating system, 283
password, 40
physical characteristic, as basis for, 40, 53
possession, as basis for, 65
privacy aspects of, 610, 612
puzzle, as basis for, 52
questions, as basis for, 39, 52
remote, 66
replay of credentials, 364
request, wireless network, 383
something known, as basis for, 40
something possessed, as basis for 40
strength of, 612, 817, 820
success of, 56
token, as basis for, 65, 66
trusted path for, 323
versus identification, 60, 61
weak, 820
WiFi, 380
wireless network, 385
Authenticity, 92, 108, 114, 115, 117, 126
digital signature, 802
asset, 7
email, 635
Author, rights of, 705
Authorization, 8, 11, 574
big data applications, 548
Authorship, 246, 705
autoexec.bat, 181
Automobile, 4
Autonomous mobile agent, 430
Autorun (automatically executing program), 181
Availability, 6, 7, 8, 11–13, 75, 398
as asset, 671
data, 11
database, 507, 512
service, 11
voting, 834
wireless network, 382
Backdoor, 158, 170, 356, 787, 790, 845
Background task, 358
Backup, 198, 421, 694
cloud, 697
complete, 694
offsite, 696
periodic, 694
revolving, 695
selective, 696
Badge, for authentication, 66
Bagle, 430
Ballot, privacy of, 641
Bandwidth, 490
Bank, attack on, 16
Barlow, John Perry, 486
Base register, 298
Base station, wireless network, 382
Bastion host, see Application proxy firewall
Battery, 817
Bauer, Kenneth, 474
Beacon, WiFi, 380, 383
Bell, David, 13
Bell–La Padula model, 13
Bellovin, Steven, 417
Bernstein, Mary, 143, 508
Best evidence rule, 735
Best practice, 824
BetCRIS, 425
Beth-Israel-Deaconess hospital, 401
Biba, Kenneth, 13
Big data, 540
access control in, 545
access granularity, 545
auditing, 546
authentication for, 548
data sharing, 543
data source provenance, 547
data validation for, 547
encryption for, 548
filtering for, 547
integrity of, 547
joining fields, 547
keys for, 547
personal data, 545
prediction using, 541
privacy, 544
proprietary data, 545
secure data storage, 546
security addition for, 548
security monitoring, 546
Biham, Eli, 788
BiN computer, 290, 302
BIND (Berkeley Internet Name Domain), 414
Biometrics, 53
disadvantages of, 55
speed of, 59, 60
template for, 59
BIOS (Basic I/O System), 292
Birthday, 615
Bitcoin, 621
Black box, airline, 640
Black hat hacking, 759
Black-box testing, 214
Blacklisting, 431, 490
Block cipher, see Block encryption
Block encryption, 93, 96, 792, 795
Blocked access, denial of service, 399
Blood vessel, for authentication, 54
Boneh, Dan, 103
Book cipher, 775
Boot sector virus, 187
Boot, operating system, 280
Bootstrap loader, 291
Bootstrap process (system initialization), 187
Bot, 168, 170, 426
Botmaster, 427, 429
Botnet, 426, 429, 430
Boundary checking, 149
Bounded disclosure, 520
Boxcryptor, 564
Branch instruction, 136
Brazil, 19, 743, 835
Breach,
data, 609
notification, 740
survey, 828
Break-in, system, 668
Breaking, encryption, 90, 91, 92
Britain, 89, 107, 318, 771, 835, 846
Broadcast mode, wireless network, 384
Browser, 232
encryption in, 437
hijacker, 170
vulnerability in, 233
Brute force attack, 791
on password, 42, 48
BSAFE, RSA cryptographic toolkit, 807
Buckingham, Earl of, 48–49
Buffer overflow, 134, 139, 140, 145, 149
Bug, program, 132
Business case, for security, 821
Business continuity plan, 659, 661
Business failure, 658
Byte substitution, in AES, 790
Bytecode verifier, Java, 295
Byzantine generals problem, 430
C (programming language), 131, 150
C++ (programming language), 131, 150
CA (certification authority), 441
Cable, network, 343
Cable, optical fiber, interception from, 346
California Breach Notification Law, 609, 740
Call, procedure, 136
CAN SPAM Act, 740
Canada, 19, 318, 741, 844
Canary, stack protection, 150
Capability, access control mechanism, 82
Capacity
availability attribute, 11
network, 398
planning, 489
CAPTCHA, 237
Caputo, Deanna, 276
CartManager, 600
CARVER, 675
Catastrophe, 659
Center for Democracy and Technology, 628, 629
Central key distribution, 124
CERT (Computer Emergency Response Team), U.S., see U.S. Computer Emergency Response Team
CERT, see Incident response team
Certifiability, reference monitor property, 76
Certificate, public key, 121, 123, 819
Certification authority, 122, 124, 441
Chain of custody, 735
Chaining, in cryptography, 113, 363, 784, 786
Challenge, motive for attack, 18
Charlotte-Mecklenburg, North Carolina, Police Department, 541
Check digit, 109
Checking,
access authorization, 156
data area boundaries, 149
Checksum, 109, 112, 113, 251, 429
Cheswick, Bill, 295
China, 15, 275, 391, 444, 464, 844
Chosen plaintext attack, 771
C–I–A triad, 7–13, 134, 432, 545
Cipher suite, in SSL, 439
Cipher, 769
Ciphertext, 88, 103
Circuit-level gateway firewall, 462
Civil law, 722
Classical probability, 676
Clear GIF, 254, 627
Clear-box testing, 214
Clickjacking, 256
Client–server network, 18
Clipper, 805
Clock, controlled access to, 283
Closed mode, wireless network, 384
Cloud computing
backup, 697
capabilities, 551-552
deployment models, 552
identity management, 568
migration, 553
platform, 579
privacy implications, 642
processing, 817
provider assessment 554
risk analysis, 553
security controls, 554
service models, 552
storage, 557, 561, 580
threats, 566
vulnerabilities 554
Code
analyzer, static, 150
breaking, see Encryption, breaking
development practices, see Program development practices
error correction, 516
error detecting, see Error detecting code
error detection, 516
hiding, 192
library, 189
modification checker, 482
modification of, 148, 819
program, 137
review, program assurance technique, 221
reviewer, 158
signed, 251
Code Red, 172, 175, 179, 182, 209, 731
Cohesion, of software, 206
Cold site, disaster recovery, 698
Cold, effect on semiconductor, 772
Collision,
in error detection codes, 110, 800
stack and heap, 148
Colorado State University, 17
Command sequence, 817
Command-and-control center, botnet, 245, 426 427, 428
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commerce, Internet, 630</td>
</tr>
<tr>
<td>Commit, two-phase update, 514</td>
</tr>
<tr>
<td>Common Criteria, 327</td>
</tr>
<tr>
<td>Common Rule, 763</td>
</tr>
<tr>
<td>Common Vulnerabilities and Exposures (CVE), 14</td>
</tr>
<tr>
<td>Common Vulnerability Scoring System (CVSS), 14</td>
</tr>
<tr>
<td>Communication, email, 632</td>
</tr>
<tr>
<td>Communication, interprocess, see Interprocess communication</td>
</tr>
<tr>
<td>Community clouds, 552, 555</td>
</tr>
<tr>
<td>Compartment, access control, 80</td>
</tr>
<tr>
<td>Competition, employment contract, 728</td>
</tr>
<tr>
<td>Compiler, 201, 209</td>
</tr>
<tr>
<td>correct code generation by, 140</td>
</tr>
<tr>
<td>role in program security, 150</td>
</tr>
<tr>
<td>Complete mediation, design principle, 217, 315</td>
</tr>
<tr>
<td>Completeness</td>
</tr>
<tr>
<td>mediation, 217, 315</td>
</tr>
<tr>
<td>operating system, 314, 320</td>
</tr>
<tr>
<td>operating system design, 314</td>
</tr>
<tr>
<td>security requirements, 653</td>
</tr>
<tr>
<td>testing, 214</td>
</tr>
<tr>
<td>Complexity</td>
</tr>
<tr>
<td>network, 358</td>
</tr>
<tr>
<td>operating system, 187</td>
</tr>
<tr>
<td>operating system design, 291</td>
</tr>
<tr>
<td>program, 149</td>
</tr>
<tr>
<td>versus security, 208</td>
</tr>
<tr>
<td>Compliance</td>
</tr>
<tr>
<td>824</td>
</tr>
<tr>
<td>Component failure, 420, 421</td>
</tr>
<tr>
<td>Compromise</td>
</tr>
<tr>
<td>74</td>
</tr>
<tr>
<td>Computability, 218, 219</td>
</tr>
<tr>
<td>Computer,</td>
</tr>
<tr>
<td>medium of crime, 736</td>
</tr>
<tr>
<td>security, 2</td>
</tr>
<tr>
<td>source of attack, 20</td>
</tr>
<tr>
<td>subject of attack, 736</td>
</tr>
<tr>
<td>system, 3</td>
</tr>
<tr>
<td>target of attack, 20</td>
</tr>
<tr>
<td>time, theft of, 750</td>
</tr>
<tr>
<td>tool for attack, 736</td>
</tr>
<tr>
<td>Computer crime, 733</td>
</tr>
<tr>
<td>complexity, 736, 743</td>
</tr>
<tr>
<td>criminal, 742</td>
</tr>
<tr>
<td>evidence, 736</td>
</tr>
<tr>
<td>evolving laws, 736. See also Computer crime laws</td>
</tr>
<tr>
<td>international aspects, 736, 741</td>
</tr>
<tr>
<td>prosecution, 736</td>
</tr>
<tr>
<td>Computer crime laws</td>
</tr>
<tr>
<td>CAN SPAM Act, 740</td>
</tr>
<tr>
<td>Council of Europe Agreement on Cybercrime, 741</td>
</tr>
<tr>
<td>E.U. Data Protection Act, 742</td>
</tr>
<tr>
<td>Freedom of Information Act, 738</td>
</tr>
<tr>
<td>U.S. Computer Fraud and Abuse Act, 738</td>
</tr>
<tr>
<td>U.S. Economic Espionage Act, 738</td>
</tr>
<tr>
<td>U.S. Electronic Communications Privacy Act, 739</td>
</tr>
<tr>
<td>U.S. Health Insurance Portability and Accountability Act (HIPAA), 739</td>
</tr>
<tr>
<td>U.S. Privacy Act, 738</td>
</tr>
<tr>
<td>U.S.A. Patriot Act, 740</td>
</tr>
<tr>
<td>Computer emergency response team (CERT), see Incident response team, Security Operations Center</td>
</tr>
<tr>
<td>Computer Emergency Response Team, U.S., see U.S. Computer Emergency Response Team</td>
</tr>
<tr>
<td>Computer forensics, 567</td>
</tr>
<tr>
<td>Computer Fraud and Abuse Act, U.S., 620</td>
</tr>
<tr>
<td>Computer security incident response team (CSIRT), see Incident response team, Security operations center</td>
</tr>
<tr>
<td>Computer security, 2</td>
</tr>
<tr>
<td>Concealment,</td>
</tr>
<tr>
<td>data, 529–535</td>
</tr>
<tr>
<td>malicious code, 178, 189</td>
</tr>
<tr>
<td>password, 46</td>
</tr>
<tr>
<td>Concurrency, 11, 286</td>
</tr>
<tr>
<td>control of, 282</td>
</tr>
<tr>
<td>database, 517</td>
</tr>
<tr>
<td>Hadoop, 543</td>
</tr>
<tr>
<td>Conficker, 174, 175, 179, 182, 428</td>
</tr>
<tr>
<td>Confidence, in trusted system, 317</td>
</tr>
<tr>
<td>Confidentiality, 6, 7, 8–10, 109, 126, 844</td>
</tr>
<tr>
<td>data, 518</td>
</tr>
<tr>
<td>database, 512, 529</td>
</tr>
<tr>
<td>database, 529</td>
</tr>
<tr>
<td>IPsec, 446</td>
</tr>
<tr>
<td>network, 441, 443</td>
</tr>
<tr>
<td>voting, 834</td>
</tr>
<tr>
<td>wireless network, 381</td>
</tr>
<tr>
<td>Configuration management, 509</td>
</tr>
<tr>
<td>Configuration, firewall, 453, 466, 472</td>
</tr>
<tr>
<td>Confinement, program development practice, 207</td>
</tr>
<tr>
<td>Confusion, in cryptography, 774, 808</td>
</tr>
<tr>
<td>Connection failure, physical, 420</td>
</tr>
<tr>
<td>Connection, rogue, 382</td>
</tr>
<tr>
<td>Connectivity, network, 371, 847, 849</td>
</tr>
<tr>
<td>Consequence, of attack, 826</td>
</tr>
<tr>
<td>Consistency</td>
</tr>
<tr>
<td>data, 11, 506, 827</td>
</tr>
<tr>
<td>security requirements, 653</td>
</tr>
<tr>
<td>Content, filtering, 464</td>
</tr>
<tr>
<td>Context, of privacy, 601</td>
</tr>
<tr>
<td>Contingency planning, 688, 694</td>
</tr>
<tr>
<td>Contract</td>
</tr>
<tr>
<td>acceptance, 723</td>
</tr>
<tr>
<td>employment, 725, 727</td>
</tr>
<tr>
<td>information, 724</td>
</tr>
<tr>
<td>law, 723</td>
</tr>
<tr>
<td>software, 724</td>
</tr>
<tr>
<td>suit involving, 725</td>
</tr>
</tbody>
</table>
validity, 723
voluntary entry, 723
Control, 6, 22, 28, 32
access, see Access control
administrative, 30
cost of, 29
ease of use, 29
logical, see Control, technical, and Control, administrative
loss of, 814
overlapping, 30
physical, 29
procedural, 30
program, 149
reducing vulnerability, 670
risk, 668
security, 653
selection, 680
technical, 30, 75
Controlled access, for privacy, 608
Controlled sharing, 287
Convention 108, of Council of Europe, 603
Cookie, 625, 627
for authentication, 65
for wireless network association, 386
third-party, 625
COPPA, see U.S. Children’s Online Privacy Protection Act
COPS (password administration tool), 43, 369
Copying, of copyrighted material, 707
Copyright, 704
backup of work, 706
computer software, 709
copying, 707
device to counter piracy, 709
digital object, 709
distribution of work, 706
employee’s work, 727
fair use, 706
first sale principle, 708
independent work, 709
infringement, 709
originality of work, 706
ownership of, 726
personal use doctrine, 708
piracy, 707
public domain, 705
registration of, 708
requirement to publish, 709
web content, 716
work for hire, 726
works subject to, 705
Cornell University, 18
Correction, of error, 11
Correctness,
data mining, 538
data, 616
operating system design, 314
operating system, 317, 320
program, 133, 219
proof of program, 219
RFID sensor, 639
security requirements, 653
software, 206
Correlation, 537, 613, 617, 622
Corruption, data, 361, 432
Cost,
data loss, 695
hidden, 679
malicious code, 179
security, 657, 824
Cost–benefit analysis, 669, 681
Council of Europe Agreement on Cybercrime, 741
Council of Europe, 603
Counterattack, 485
Countermeasure, 6, 22, 28, 32, see also Control
Coupling, of software, 206
Coverage, testing, 214
Covert redirect, 577
Cowan, Crispin, 150
Crack (password administration tool), 43
Cracking, 761
Credit card theft, 19, 22
Credit card, disposable, 621
Crime, computer, see Computer crime
Crime, organized, see Organized crime
Criminal law, 722
Criminal, 16, 19, 742
Crisis management, see Business continuity plan, Incident response
Crocker, Stephen, 143, 508
Crossover, network, 363
Cross-site scripting, 261
Cryptanalysis, 90, 769
brute force, 791
chosen plaintext attack, 771
freezing attack, 772
frequency analysis attack, 769
frequency analysis, 793
full plaintext, 770
implementation attack, 769
in AES, 99
inferring the algorithm, 774
known plaintext, 770
pattern analysis attack, 769
plaintext and ciphertext, 770
plaintext-only, 768
probable plaintext attack, 770
probable plaintext, 793
RSA, 797
 statistical analysis, 776
Cryptographic
 algorithm, 95
 checksum, 113
 side-channel attack, 566
Cryptography, 86, 90, 768
 asymmetric, 102
 authentication using, 817
 book cipher, 775
 BSAFE toolkit, 807
 chaining, 784, 786
 checksum using, 113
 confusion, 774, 808
differential cryptanalysis, 788
diffusion, 774
dual-EC-DBRG, 806
El Gamal, 803
elliptic curve cryptosystem (ECC), 802
export control, 792, 793, 794, 805
Keccak, 801
key escrow, 805
Lucifer algorithm, 780
 mathematical basis, 778
 MD4, 800
 MD5, 800
 network security control, 433
 one-time pad, 774
 product cipher, 782
 public key, 100, 102
 public scrutiny of algorithm, 779
 quantum, 807
 RC2, 792
 RC4, 792
 RC5, 794
 RC6, 795
 RSA, 795
 secret key, 96
 separation using, 296
 SHA, 800
 SHA-3, 801
 strength of, 817
 substitution, 774
 Suite B, 803
 symmetric, 96
 transposition, 774
 Vernam cipher, 775
 weakness in, 789, 792, 794, 806
Cryptolocker, 565
Cryptology, 90
 and NSA, 805. See also U.S. National Security Agency
Cryptosystem, 87

CSA STAR (Cloud Security Alliance Security, Trust and Assurance Registry), 555
CSIRT, see Incident response team
Currency, virtual, 621
CVE, see Common Vulnerabilities and Exposures
CVSS, see Common Vulnerability Scoring System
Cyber warfare, 842
Cyber weapon, 847
Cybercrime, 19
 Council of Europe Agreement on, 741
Cyberworthiness, 730
Cycle,
in AES, 98
 in SHA-3, 801
Cyclic redundancy check, 111, 516

Daemon, 352
 name, 414
Damage control, in software design, 311
Damage, from malicious code, 179
Dark Internet, 444
Darwin (computer game), 172
Data
 access to, 8
 access to, as asset, 3
 access, rights of individuals, 603
 accuracy of, 11
 anonymization, for privacy, 597
 approximate value of, 9
 as asset, 3, 671
 bias, 759
 breach law, 609
 consistency of, 11
 correctness, 616
 corruption of, 361
 critical, 281
disclosure of, 8
 driven attack, 148
error, and privacy, 608
 existence of, 9
 exposure of, 177
 integrity of, 10–11
 irreplaceable, 696
 loss, cost of, 695
 meaningful, 11
 misleading, 759
 modification of, 11, 529, 597
 modification, for privacy, 597
 object, 9
 ownership of, 8, 596, 608
 perturbation, database, 534
 precision of, 11
 privacy, 736
private, 587
protection of, 687
protection, and privacy, 597
quality, and privacy, 596, 608
replacement, 3, 4
retention, limited, 597
retraction of, 594
sensitive, 587
separation of, 11
shared access, 287
storage, 546
storage, for privacy, 608
subject, 9
suppression, database, 529
swapping, database, 535
transfer, and privacy, 603
unchecked, 153
use, government, and privacy, 607
use, privacy of, 590
use, restricted, 608
validation, with big data, 547
value of, 736
versus instruction, 137
Data collection,
accuracy, 599
consent, 591
control of, 599
for specific purpose, 603
limit on, 596, 603
notice of, 591, 599
openness of, 597
ownership, 592
privacy of, 590, 640
security of, 599
Data mining, 246, 527, 536
aggregation, 537
correctness, 537
correlation, 537
false negative, 540
false positive, 540
inference, 537
interoperability, 540
mistakes, 538
privacy, 537, 616
privacy-preserving, 617
sensitivity, 537
Data Encryption Standard, see DES
Data Loss Prevention (DLP), 473
Data Protection Act, 742
Database Management System (DBMS), 502
Database, 502
access control, 508
administrator, 502
aggregation, 526
algebraic inference, 525
arithmetic inference, 522
auditing, 510
authentication, 512
availability, 512
bounded disclosure, 520
concurrency, 517
confidentiality, 512
data concealing, 529–535
data disclosure, 529
data perturbation, 529, 534
data swapping, 535
disclosure, 518
element 502
element accuracy, 513
element integrity, 508
element integrity, 513
exact disclosure, 519
existence disclosure, 520
field check, 508
field, 502
granularity, 512
hidden data, 527
inference, 511, 521–525
integrity, 513
integrity, two-phase update, 514
join query, 505
key, 512, 621
limited response suppression, 532
mean inference, 523
median inference, 523
negative disclosure, 520
operating system protection, 513
performance, 511
probable value disclosure, 520
project operation, 504
protecting data in, 721
query, 504
query analysis, 535
query language, 504
random sample disclosure, 534
range disclosure, 533
record, 502
recovery, 516
relation, 504
reliability, 513
rounded disclosure, 533
schema, 502
sensitive data, 518
shadow field, 516
small sample concealment, 534
SQL query language, 504
subschema, 502
table, 502
tracker inference, 524
tuple, 504
user authentication, 512
Data-driven attack, 189
Datagram, 407, 415
DataNode, in Hadoop, 543
DBMS, see Database Management System (DBMS)
DDoS attack, see Distributed Denial of Service attack
DEA, encryption algorithm, 780
DEA1, encryption algorithm, 780
Deadlock, 11, 282
DEC VAX computer, 290, 314, 326
Deception, and privacy, 600
Deception, email, 740
Deceptive practice, 630
Decidability, 190, 218, 219
Decipherment, 87
Decision–making, 25, 684, 831
Decoding, 87
Decryption, 87
Defacement, web site, see Web site defacement
Defense in depth, 30, 218, 471
Defense, Department of, see U.S. Department of Defense
Defensive programming, 222
Defibrillator, 816
Deflection, attack countermeasure, 28
Degauss, magnetic media, 693
Degradation, graceful, see Graceful degradation
Degraded service, network, 849
Delay, in access checking, 156
Deletion, data, 134, 692, 772
Delphi method, 677, 678
Demand, network, 398
Demilitarized zone (DMZ), firewall architecture, 470
Denial of service, 6, 14, 18, 20, 175, 367, 396, 753
Denial of service (DoS) attack, 6, 20, 425, 843
access, blocked, 399
address resolution, 414
address spoofing, 413
addressing failure, 408
distributed, see Distributed denial of service attack
DNS, 414
DNS cache poisoning, 418
DNS spoofing, 409
echo–chargen, 404
flooding, 398, 402, 407
hardware, 845
incident, 401
insufficient resources, 402, 407
malicious, 403
overload, 399
ping of death, 404
root name server, 414
routing, 409, 413
scripted, 423
session hijack, 415
smurf, 404
source routing, 413
SYN flood, 405
tear drop, 407
Tribal flood network, 424
volumetric, 399, 423
Denning, Dorothy, 530
Denning, Peter, 72, 292
Deontology, 749
Department of Defense, see U.S. Department of Defense
Department of Justice, see U.S. Department of Justice
Dependability, 12
DES (Data Encryption Standard), 95, 439, 779
computational limitation of, 98
cycle in, 99
decryption, 784
design secrecy, 787
differential cryptanalysis on, 788
for tamper detection, 113
key length in, 96, 98, 109
number of cycles, 788
reversibility, 784
security of, 98, 87
speed of encryption, 103
strength of, 789, 805
Design by contract, program design technique, 223
Design flaw, 6
Design principles, 216
Design, cryptographic algorithm, 779
Design, layered, 309
Design, RSA, 797
Design, simplicity of, 309
Design, TCB, 321
Detection,
attack countermeasure, 28
avoidance, by malicious code, 191
error, 11
malicious code, 189
tamper, 151
Detector, virus (program), see Virus detector
Deterrence, attack countermeasure, 28
Development
practices, see Program development practices
program, security in, 158
quality software, 816
Device
access control, 284
allocation, 281
Document virus, 180
DoD, see U.S. Department of Defense
Domain, 82
effectiveness, 286
name, 444
switching, 320
DoS (Denial of Service), see Denial of service
Dot-dot-slash attack, 264
Double DES, 96
DoubleClick, 625
Download substitution attack, 237
Download, and privacy, 629
Drive-by-download, 258
Dropbox, 561, 563
Dropper, 170
Drug trafficking, 19
DSA, see Digital Signature Algorithm
Dual-EC-DBRG cipher suite, 806
Dual-homed gateway, 450
E.U. Data Protection Act, 742
Ease of use, design principle, 217, 317
Easter egg, 158
Eavesdrop, 243, 343, 354, 432, 808
ECC, see Elliptic curve cryptosystem
Echo–chargen attack, 404, 477
Economics, of security, 821
Economy of mechanism, design principle, 217, 315
Effectiveness, of testing, 215
Egoism, 748
e-Government Act, 599
Egypt, 847, 849
El Gamal algorithm, 803, 804
El Gamal, Taher, 803
Eleanore (attack toolkit), 166
Election, fair, 836, 837
Election, margin of victory, 838
Electrical use, 817
Electronic commerce, protection of goods in, 721
Electronic Communications Privacy Act, U.S., 620, 739
Electronic publishing, compensation for, 721
Electronic voting, 835
Element integrity, database, 507, 508, 513
Element, database, 502
Elliptic curve cryptosystem (ECC), 439, 802, 804, 806
Email 607
accuracy of headers, 273
address, disposable, 634
alias with, 632
authentication, 39
deceptive, 740
disappearing, 635
exposed content of, 632
filtering, 560
forged, 267
forwarding of, 632
header data, 273
interception of, 633
monitoring of, 633
PGP, 276
phishing, 274
S/MIME, 277
security of, 632
spam, 740
spear phishing, 274
Emanation, electromagnetic, 693
Embedded device, 4
Emergency alert system, 1–2
Employee rights, 725
Employer rights, 725, 754
Employment contract, 725, 727
non-compete clause, 728
Encapsulated Security Payload, in IPsec, 445
Encapsulation, 204
by layering, 311
of software, 206
Encipherment, 87
Encoding, 87
Encrypted password, 46
Encrypted virus, 194
Encryption, 86, 87
algorithm design, 433
asymmetric, 89, 93, 795
block, 93
breaking, 90, 91
chaining, 363
desk-to-end, 435, 437, 438
exhaustive key search, 395
for continuous authentication, 245
for privacy, 597
in network, 360, 363
in the cloud, 561
in wireless network, 383
initialization vector collision, 389
key, 88, 126, 562
key length, 388
key management, 446
key, private, 126
keyed, 88
keyless, 89
link, 433, 437
non-static key, 392
protection using, 433
speed of, 126
static key, 388
stream, 93
symmetric, 88, 92, 779, 786
TKIP, 393
weak, 388
See also AES, DES, RC2, RC4, RC5, RSA
End-to-end encryption, 435, 437, 438
Enforcement, of integrity, 11
Engineering, social, see Social engineering
Enigma machine, 771, 774
Entry point, secret, 27, 158. See also Backdoor
Equipment failure, 420
Equivalence, of programs, 189, 219
Erasing, sensitive data, 692
Error
correction, 11
detection, 11
in data, 608, 617
in encryption, 778
inadvertent, 14
nonmalicious, 133
off-by-one, 159
program, 132
unintentional, 6
Error correction code, 516. See also Error detection code
Error detection code, 109, 111, 251, 516
Escrow, encryption key, 805
ESP (Encapsulated Security Payload), see Encapsulated Security Payload
Espionage, 171, 668, 738
Estimation technique, 676
Estimation, Delphi method, 678
Estonia, 18, 2, 391, 396, 641, 838, 842, 843, 846
Ethical hacking, see Penetration testing
Ethical system, 745
Ethics, 744
analysis, 746
and religion, 746
consequence-based, 748
consequence-based, 749
depth of action, 753
deoontology, 749
egoism, 748
fair sharing, 753
intrinsic goodness, 749
in the cloud, 561
privacy and, 752
overriding principles, 748
pluralism, 746
privacy and, 752
religion and, 746
rule-deontology, 749
teleology, 748
to justify a position, 748
to make a reasoned choice, 748
to make a reasoned choice, 749
utilitarianism, 749
variability, 746
versus law, 744
Euler totient function, 797
Europe, Council of, 603
European Privacy Directive, 603–604, 605, 849
European Union, 596, 603
data breach laws, 609
Even parity, 111
Evidence,
  authenticity, 735
  chain of custody, 735
  computer output as, 734
  incident, 664
  rule of, 734
Exact disclosure, 519
Exchange, cryptographic key, 104
Exclusiveness, of access, 11
Executable code, hiding, 192
Executive, operating system, 280, 285
Exfiltration, of sensitive data, 474, 845
Exhaustive attack, on password, 48
Exhaustive key search, 395
Existence,
  disclosure, 520
  of data, 9
Expected loss, 678
Experimentation, informed consent, 763
Exploitation, vulnerability, 419
Export control, of cryptography, 562, 792, 793, 794, 805
Exposure, risk, see Risk exposure
Extensible Authentication Protocol (EAP), 393
Externality, 834
Fabrication, 87, 107
  air defense signal, 844
  asset, 8
  encrypted data, 785
  network data, 361
Facebook, 526, 594, 635, 696, 762
Facial recognition, for authentication, 53, 59, 60, 62, 63, 64
  in error detection codes, 111
  of malicious code, 192, 198, 200
Finland, 641
Fire, 659, 687
Firesheep, browser extension, 386
Firewall, 448, 452, 492
  application proxy, 459
  circuit-level gateway, 462
  demilitarized zone, 470
  Great Firewall of China, 464
  guard, 463
  packet filtering gateway, 456
  personal, 464
  stateful inspection, 458
First sale, principle of, 708
Fit for use, 730
Fake email, 267
False
  acceptance, see False positive
  accusation, 608
  alarm, 824
  negative, 55, 56, 62, 64, 488, 540
  positive, 55, 56, 62, 64, 488, 540, 824
  reading, 55
  reject, see False negative
Farmer, Dan 369
Fault tolerance, 12
Fault tree analysis, 673
Fault, program, 132, 136
FBI, see U.S. Federal Bureau of Investigation
Feasibility, of attack, 27–28
Federal Bureau of Investigation, see U.S. Federal Bureau of Investigation
Federal Information Processing Standard 197 (FIPS 197), 99
Federal Trade Commission (FTC), 630, 635
Federated identity management, 68, 569
FedRAMP (Federal Risk and Automation Management Program), 555
Fence, 297
Fence register, 298
FidM, see Federated identity management
Field check, database, 508
Field, database, 502
File, 320
File access control, 284
File sharing, peer-to-peer, 629
File tag, 528
Filter, packet, see Packet filtering gateway
Filter, polarizing, 808
Filtering, 486
Filtering, in big data, 547
Fingerprint,
  for authentication 53, 59, 60, 62, 63, 64
  in error detection codes, 111
  of malicious code, 192, 198, 200
Fair election, 836, 837
Fair Information Practices, 596
Fair use, 706
Fairness, 11, 281
Flaw
   design, 6
   impact of, 134
   program, 133, 184
   reporting, 731
Floating-point error, Intel Pentium chip, 10
Flood, 686
Flooding attack, 479, 840
   in denial of service, 399, 479
FOIA, see U.S. Freedom of Information Act
Forensic analysis, 74, 202, 736
Forgery,
   digital signature, 802
   in authentication, 58–59, 65, 66
   protection against, 116
Formal methods, program assurance technique, 220
Forwarding, email, 632, 634
Frame,
   Data-link, 352
   SSID in, 384
   WiFi, 379
Framing, web page, 258
France, 318, 846
Fraud, 19, 22, 722, 757
Free public WiFi, 392
Frequency
   analysis, against encryption, 769, 793
   distribution, in cryptanalysis, 777
   probability, 676
Front end (database management system), 502
Front-end (intrusion detection system), 480
f-Secure Corp., 19
Full disclosure, of software flaws, 731, 760
Full plaintext attack, 700
Function testing, 211
Functionality, in Common Criteria, 328
Gasser, Morrie, 314
Gateway, application proxy, see Application proxy firewall
Geer, Daniel, 209, 210
Genetic diversity, program environment characteristic, 209
Geographic diversity, 558
Geography, and cyber warfare, 850
Geotag, 528
Germany, 107, 318, 431, 668, 771
Get_root exploit, 291
Ghostery, 623
Gong, Li, 295
Gonzales, Albert, 19, 391
Good, greatest for greatest number, 762
Goodness, program characteristic, 218
Google, 818, 820
   docs, 697
   Street View project, 378
GOTO Fail bug, 213
Government data use, 607
Graceful degradation, 12
Graham, Scott, 72, 292
Gramm–Leach–Bililey Act, 598, 739
Grandin, Temple, 17
Granularity,
   database, 512
   in big data, 545
   of access control, 74, 75, 287, 297
Great Falls, Montana, 1
Great Firewall of China, 464
Greece, 356
GrIDSure authentication system, 52
Group, access control, 80
Guard firewall, 463
Guard, human, 680, 690
Hachette, 631
Hacker, 15, 18, 19, 759. See also Malicious code attack
Hacking, black hat, 759
Hacking, white hat, 759
Hadoop, 542
   anonymization for, 545
   concurrency, 543
   DataNode, 543
   map–reduce, 543
   NameNode, 543
   privacy for, 544
   redundancy, 543
   secure mode, 548
   sharing, 543
   trusted users, 543
Halme, Larry, 474
Halting problem, 218, 219
Hamming code, 799
Hand geometry, for authentication, 54
Hard problems, cryptographic, 92
Hardware,
   as asset, 3, 671
   failure, 6, 420, 398, 421, 772
   interface to, 282
   loss of, 691
   malicious, 845
   modification of, 845
Harm, 6, 13, 23, 28
   causes of, 22
   from buffer overflow, 138
   from computer attack, 21–25
   from malicious code, 176, 179
   from vulnerability search, 762
   likelihood of, 22
   limitation of, 845
   malicious, 14
measurement of, 14
potential, 764
severity of, 22
stopping, 845
to asset, 5, 8
types of, 14
Hash code, 109, 112, 116, 125, 428, 799
Hash function, see Hash code
Hastiness, in faulty error analysis, 816
Hazard analysis, 672
Hazard and operability study, 673
Header, packet, 458
Healthcare data, 739
Heap, 147
Heap, memory, 136, 139
Hellman, Martin, 96, 98, 101, 791
Heuristic intrusion detection system, 476
Hiding,
address, 303
malicious code, 192
Hierarchical design, 311
Hijack attack, 242
HIPAA, see U.S. Health Insurance Portability and Accountability Act
Hoare, Anthony (C.A.R.), 149
Hoax, 2, 176
Honan, Mat, 559
Honeypot, 295, 668
Hooking, to operating system, 288, 337, 465
Hop, 413
Host scanning, 566
Host-based firewall, see Personal firewall
Host-based intrusion detection system (HIDS), 476, 480
Hostile mobile agent, 170
Hot site, disaster recovery, 698
Hot spot, wireless, 382
HTTPS (HTTP Secure) protocol, see SSL
Human error, in encryption, 771
Human subjects, 762
Human, threat from, 13–14
Human–computer interaction (HCI), 654
Hybrid clouds, 553, 555
Hypervisor, 292
Hypönen, Mikko, 19

IaaS (Infrastructure as a Service), 552, 558, 579, 580
IBM Corp. 95, 97, 290, 779, 788, 789
ICD, 816, 817
Iceland, 613
Identification versus authentication, 60, 61
Identification, 38, 243, 610, 617, 815, 816
only when necessary, 603
unique, 610

versus authentication, 38
voluntary, 638
weak, 820
Identity, 38, 122
card, for authentication, 66
documents, 612
unordered, 611
linking, 606
management, cloud, 568
management, federated, see Federated identity management
multiple, 606
non-unique, 611
records confirming, 49
theft, 609
uniqueness of, 606
IDS (Intrusion Detection System), see Intrusion Detection System
IEEE (Institute of Electrical and Electronics Engineers), 132
IETF (Internet Engineering Task Force), 444
Iframe, web page, 258
IKE, see Internet Security Association Key Management Protocol Key Exchange
ILoveYou (malicious code), 172, 175, 179
ImageShield, authentication system, 52
Immunity, from malicious code infection, 195, 200
Impact,
of attack, 831
of computer failure, 660
of security incident, 25
risk, 668
Impersonation, 107, 474
Implantable Cardioverter Defibrillator, see ICD
Implanting, of malicious code, 186
Implementation weakness, against encryption, 770
Implementation, TCB, 322
In-the-clear, message, 434
Incident cost, 828
Incident response, 567
action, 662
coordination, 666
declaring an incident, 662
national, 666
plan, 662
post-incident review, 665
taking charge, 662
team membership, 665
team, 397, 665
Incident survey, 828
Incident, security, 25
Incomplete mediation, 152
Independence, program quality, 204
Independent testing, 215
India, 743
Index

Individual, versus identity, 611, 612
Inductance, network, 343
Infection, malicious code, 186, 194, 430
Infection, spread of (malicious code), 185
Inference engine, intrusion detection, 476
Inference, database, 511, 521–525
Inference
  in data mining, 537
  in intrusion detection system, 478
Information
  as an object, 717
  commerce, 720
  cost of, 718
  depletion of, 718
  disclosure of, 739
  hiding, of software, 204, 206
  replication of, 718
  transfer of, 719
  value of, 719
Informed consent, 763
Infowar, 486
Infrastructure
  ownership of, 849, 850
  shared, 566, 580
  virtual, 581
Infringement
  copyright, 709
  patent, 712–713
Initialization vector, 786, 793
Injection attack, 839
Injection, SQL, attack, 263
Input validation, 153
Input, unchecked, 154
Insecure APIs, 566
Insertion, in network communication, 364
Insider, 474
Insider threat, 357
Installation
  malicious code, 186
  program, 237
  testing, 211
Instruction, machine, 136
Insurance, 23, 669, 688, 695
Integer overflow, 160
Integrated vulnerability assessment, 675
Integration testing, 211
Integrity, 6, 7, 8, 117, 251, 758
  big data, 547
  check, 109, 112
  code, 151, 482
  computation, 133
  contextual, 602
  data 10–11, 506
  database, 507
  enforcement by operating system, 317
  enforcement of, 11
  failure from race condition, 165
  failure of, 109
  faulty check in wireless network, 390
  inaccurate linking, 608
  incorrect form, 608
  network communications, 366
  protecting, 109
  protection in WPA, 393
  stack, 151
  voting, 834
  wireless network, 381
Intel, 10
Intellectual property, 705
  as asset, 3
Intent, two-phase update, 514
Interception, 87, 236, 808, 845
  air defense signal, 844
  asset, 8
  authentication data, 243
  cryptographic key, 105, 107
  encryption, 91
  Internet, 635
  lawful, 355
  network, 354
  network, 360
  pacemaker signal, 816
  personal data, 820
  signal, 344
  WiFi communication, 364, 391
  WiFi, 364
Interface design, 654
Interface, usability, 840
Internal intrusion detection system, 480
Internet, the
  governing, 419
  international nature of, 741
  payments using, 621
  privacy and, 619
  site registration on, 622
  user ID, 622
Internet access, free, see Public hot spot
Internet of things, 814
Internet Security Association Key Management Protocol
  Key Exchange (IKE), 446, 447
Internet Service Provider, see ISP
Internet Society, 124
Internet-enabled product, 814
Interpreted language, 189
Interpreter, 189
Interprocess communication, 281, 320
Interruption,
  access, 849
network communication, 366
of asset, 8
Intrusion Detection System (IDS), 474
anomaly-based, 476
false alarm, 824
front-end, 480
heuristic, 476
host-based (HIDS), 476, 480
inference engine, 476
internal, 480
model-based, 478
network-based (NIDS), 476, 480
response, 483
signature-based, 476
situation assessment, 488
state-based, 478
stateful protocol analysis, 479
stealth mode, 487
Intrusion Prevention System (IPS), 474, 482
Intrusion, system, 668
Invention, patent for, 711
Inverse,
of a function, 112
of encryption, 103
Investigation, security, 426
iOS (Apple operating system), 818
IP fragmentation attack, 407
IPS (Intrusion Prevention System), see Intrusion Prevention System
IPsec (IP security protocol suite), 444
authentication header, 444
security association in, 444
IPv4, 444
IPv6, 444
Iran, 368, 444, 843, 847
ISAKMP (Internet Security Association Key Management Protocol), see Internet Security Association Key Management Protocol
ISO 7489-2, 7
Isolation, 204
in operating system design, 311
malicious code countermeasure, 195
of potential malicious code, 197ISP (Internet Service Provider), 19, 425, 633
Israel, 843, 844, 845
Iteration, in DES, 96
J.P. Morgan Chase, 16
Japan, 741, 772
Jaschen, Sven, 430
Java script attack, 262
Java, sandbox in, 294
Jet Blue, 601
Jihad Jane, 20
Join query, database, 505
Join, in big data, 547
Justice, Department of, see U.S. Department of Justice
Justification, with risk analysis, 684
Kahn, David, 90, 771, 774
Kahneman, Daniel, 25
Karger, Paul, 172, 219, 314
Kaspersky Labs (security research firm), 169
Keccak, 801
Kerberos, in big data application, 548
Kerckhoffs, Auguste, 227
Kernel
operating system, 284, 312, 334
security, see Security kernel
Kernell, David, 39
Key
asymmetric, 796
change, 771
cryptographic, 96, 103, 777
database, 512, 621
deduction, against encryption, 770
derivation functions, 562
distribution, 93, 124
encryption, 88, 789
encryption, sharing, 92
escrow, encryption, 805
exchange, 104
exchange, Diffie–Hellman, 446
exchange, with asymmetric cryptography, 105, 107
for RC4, 793, 794
in big data, 547
length, 96, 97, 109, 792, 805
length, in AES, 791
management, encryption, 93, 433, 446
physical, security properties of, 184
recovery, 805
search, exhaustive, 395
Keys (cryptographic), proliferation of, 102
Keystroke logger, 236, 442, 628
Kill switch, 845, 848
Known plaintext attack, 770
Koobface, botnet, 426
Korea, victim of phishing attack, 275
Krebs, Brian, 159
l0pht computer group, 139–140
La Padula, Leonard, 13
LAN (Local Area Network), 343
Landau, Susan, 645
Language,
interpreted, 189
programming, 150
safe, 149
Laptop, loss of, 691
Lastpass 564
Law,
  as security protection 426
civil, 722
civil, 722
civil, 722
civil, 722
civil, 722
civil, 722
civil, 722
civil, 722
civil, 722
E.U. Data Protection, see European Privacy Directive
  see European Privacy Directive
security control, 426
tort, 722
  versus ethics, 744
Layered protection, 471
Layering, 310
Leakage, data, 474, 620
Least common mechanism, design principle, 217, 317
Least privilege, 73, 216, 218, 316, 358
Legal
  action, 485
countermeasure, 426
issue, incident, 664
  protection, 703
Leverage, risk, 669
Leveson, Nancy, 815
Levy, Elias, 145
License, software, 727
Likelihood,
  of event, 668
  of exploitation, 675
  of harm, 22
  of security incident, 25
Limitation on data collection, 596
Limitations of testing, 215
Limited privilege, 317. See also Least privilege
Limited response suppression, 532
Link encryption, 433, 437, 447
Linking, personal data, 627
Linux operating system, 291
List, access control, see Access control list
Litchfield, David, 134, 174, 732
LMS (Learning Management System), 570
Load balancing, 431, 489, 492
Local data, 141
Location-sensing, 820
Lock, physical, 680, 690
Log analysis see SIEM
Log data, see System log
Log, access, see Access log
Log, audit, see Audit log
Logarithm, 802
Logger, keystroke, 236, 628
Logic bomb, 170
Logical integrity, database, 507
Lookup, DNS, 409
Loss, 5
data, 695
  expected, 678
from security incident, 668
power, 688
  service, 366
Lucifer, encryption algorithm, 780, 788, 789
Lyon, Barrett, 425
MAC (Media Access Control), 343, 351, 377
MAC address, 378, 380
  changing, 385
  spoofing, 394
MAC header, WiFi, 379
Mach, 302
Macro, 189
Macro virus, 10
Mafia, 769
Magnetic media, destruction of, 692
Magnetic remanence, 325
Maintenance, software, 205
Malfunction, device, 815
Malicious code, 166–196, 167
  addressing failure, 408
  adware, 170
  Alureon, 334, 336
  antivirus detector (program), 191
  appended virus, 181
  attachment of, 178
  attack toolkit, 419, 424
  backdoor, 170
  Bagle, 430
  boot sector virus, 187
  bot, 170
  browser hijacker, 170
  Code Red, 172, 175, 179, 182, 209, 731
  concealment of, 178, 189
  Conficker, 174, 179, 182, 428
  destructive, 176
  detection avoidance by, 191
  detection of, 189
  disassembly of, 201
  DNS spoofing, 409
  dropper, 170
  echo–chargen, 404
  encrypting virus, 194
  encryption of, 194
  fingerprint of, 192, 198, 200
  flooding, 403, 407
  forensic analysis of, 202
  harm from, 176
  hijacker, 629
Index

history of, 172
hostile mobile agent, 170
ILoveYou, 172, 175, 179
immunity from, 195, 200
implant of, 186, 760
infection by, 194
isolation, as countermeasure, 195
keystroke logger, 236
logic bomb, 170
malware detector (program), 191
man-in-the-browser, 234
Melissa, 172, 175
memory-resident virus, 188
mobile agent, 430
mobile agent, hostile, 170
Morris worm, 172, 175, 209
multipartite virus, 178
MyDoom, 430
NIMDA, 172
pattern matching to detect, 192, 198, 200
pattern recognition, 192
ping of death, 404
polymorphic, 193
prevention of, 197
propagation of, 180
rabbit, 170
Remote access Trojan horse (RAT), 170
replacement virus, 182
rootkit, 170, 329, 334, 336
Sasser, 431
scareware, 170, 195
script kiddie, 196
separation as countermeasure, 195
session hijack, 415
signature, recognition by, 192, 198, 200
Slammer, 172, 175
smurf, 404
SoBig, 172, 175
source of, 845
spyware, 170, 628
stealth, 189, 190
steganography and, 192
Stuxnet, 174, 175, 368, 843
SYN flood, 405
TDL-3, 334
teadrop, 407
TFN, 424
TFN2K, 424
time bomb, 170
tool, 170
toolkit, 170, 196, 336
transmission of, 180
trapdoor, 170
Tribal flood network, 424
Trin00, 424
Trojan horse, 169
user-in-the-middle, 237
virus, 167
volume of, 196
Waladec, 429
worm, 168
zero-day attack, 172, 419
Zeus, 245
zombie, 170
Malicious threat, 14
Malicious web activity,
clickjacking, 256
cross-site scripting, 261
dot-dot-slash, 264
drive-by-download, 258
server-side include, 265
SQL injection attack, 263
web site defacement, 246
Malware, 10, 166–196, 167.
see also Malicious code
Android phone, 819
detector (program), 191
scanner, 465
smartphone, 818
toolkit, 196
Man in the middle, 460
Management, 75
encryption key, 93
network, 489
risk, see, Risk management
security, 657
Manager, 657
Man-in-the-browser attack, 234, 442
Man-in-the-middle attack, 106, 409, 394, 840
Man-in-the-mobile attack, 245
Many-to-one function, 110
Map–reduce, in Hadoop, 543
Mash, in encryption, 792
Masquerading, 432
Matrix, access control, see Access control matrix
Mayfield, Terry, 10–11
McAfee (security firm), 19
MD4, 428
MD5, 800
MD6, 429
Mean inference, database, 523
Measurement
harm, 14
security, 825
Mechanism,
access control, 76
security, 75
Median inference, database, 523
Mediation, complete, 154
Mediation, incomplete, 152, 155
Medical device, 815, 820
Medical record, 613, 638, 758
Medium Access Code (MAC), see MAC
Melissa, 172, 175
Memory allocation, 136
Memory management,
  paging, 306
  segmentation, 303
  virtual memory, 303
Memory protection, 284, 297, 320, 321
  base/bounds, 298
  paging, 306, 307
  segmentation, 303
  tagged, 301
  virtual memory, 303
Memory, system space, 138
Memory-resident virus, 188
Merchants, Internet, 630
Merkle, Ralph, 96, 121
Message digest, 109, 112, 125, 799
  MD4, 428, 800
  MD5, 800
  MD6, 429, 800
  SHA, 800
Message, protection of, 434
Metadata, 529
Method, of attack, 26
Method–opportunity–motive, 26–28, 837
Microkernel, 289
Microscope and tweezers, 202, 735
Microsoft, 222, 818
Microsoft Trustworthy Computing Initiative, 326
Microwave signal, 346
Minimality, 212
Minimization, data, for privacy, 608
Misidentification, 815
Missile attack, 844
Mission creep, 608
Mistakes, in data mining, 538
Misuse, system, 74
Mitigation, attack countermeasure, 28
Mitnick, Kevin, 17, 18
Mitre Corp, 14
Mix, in encryption, 792
  Mix columns, in AES, 790
  Mixmaster remailer, 634
  Mixter (German hacker), 18
Mobile agent, hostile, 170, 430
Mobile phone, 4, 818
Mode, of access, 9, 72
Model-based intrusion detection system, 478
Modification, 87
  asset, 7, 8
  code, 819
  data, 11, 529
  network, 361
  program file, 177
  protection against, 110, 113
  sensitive data, 820
Modularity,
  in program design, 203, 204
  in operating system design, 312
  in operating system implementation, 322
Money laundering, 19
Monitor,
  operating system, 285
  reference, see Reference monitor
  virtual machine, 292
Monitoring, 474, 483, 484
  and privacy, 620
  implanted medical device, 816
  real-time, 546
  security, 475
Moore’s Law, 92
Morris Worm, 172, 175, 179, 209
Morris, Robert T., Jr., 18, 172, 179
Morris, Robert T., Sr., 43, 172, 417
Motivation, of attacker, 16, 19, 773, 837
Motive, 18, 19, 26, 816
Mudge, see Zatko, Peter
Mulligan, Deirdre, 605
Multics, 80, 85, 216, 219, 290, 307, 326
Multifactor authentication, 70
Multiparty virus, 178
Multiple remailer, 634
Multiplexing, network, 345, 363
Multiprogramming, 285
Multistep attack, 148
Multitasking, 283
Murder, by computer, 816
Music-sharing service, 629
Mutual authentication, 561
Mutual suspicion, software characteristic, 207
MyDoom, 430
Name server, 414, 419
Name, domain, 444
named (name daemon), 414
NameNode, in Hadoop, 543
Napster, 707
NAT, see Network Address Translation
National Bureau of Standards, see U.S. National Bureau of Standards
National Institute of Standards and Technology, see U.S. National Institute of Standards and Technology
National Security Agency (NSA), see U.S. National Security Agency
NATO, 845, 849
Natural disaster, 6, 13, 22, 686
building collapse, 687
fire, 687
flood, 686
water, 686
weather event, 687
NBS (National Bureau of Standards), see U.S. National Bureau of Standards
Need to know, 739
Negative disclosure, 520
Netherlands, 318, 641
NetSky, 430
Network, 342
client–server, 18
communication, confidential, 443
data loss prevention, 474
design, 401
interception in, 343–346
monitoring, 560
penetration of, 844
port scan, 456
traffic flow, 401
transmission media, 343
Network Address Translation (NAT), 472
Network attack,
component failure, 368
denial-of-service, 367
insertion, 364
interception, 354, 355
loss of service, 366
port scan, 369
replay, 364
routing, 367
sequencing, 363
substitution, 363
Network Interface Card (NIC), 351, 376, 380
Network management, 489
addressing, 490
bandwidth allocation, 490
blacklisting, 490
capacity planning, 489
load balancing, 489
rate limiting, 490
shunning, 490
sinkholing, 490
tuning, 490
Network-based intrusion detection system (NIDS), 476, 480
Networked storage, for backup, 697
Neumann, Peter, 311
NIC (Network Interface Card), 351, 376, 380
NIMDA, 172
Nissenbaum, Helen, 601
NIST (National Institute of Standards and Technology), see U.S. National Institute of Standards and Technology
Nixon, Richard, 596
Nmap, scanning tool, 369
Noise,
for privacy, 545
in communication, 109
Nonce, 108, 278, 793
Non-compete clause, employment contract, 728
Nonmalicious threat, 14, 420
Nonrandom attack, see Targeted attack
Nonrepudiation, 7, 115
Nothing up my sleeve numbers, 806
Notice, privacy, 600
Notification, of data breach, 609
Novelty, patent requirement, 712, 713
NSA (National Security Agency), see U.S. National Security Agency
Numbering, sequence, 419
OAuth 573
Access token, 574
Authorization server, 574
Client secret, 575
Client, 574
Refresh token, 576
Request token, 575
Resource owner, 574
Resource server, 574
Token, see OAuth access token
Object, 72
access control, 284
data as, 9
name, 77
reuse, 325
Obscurity, security through (by), 185, 226, 356, 836
Odd parity, 111
Off-by-one error, 159
Offset, page, 306
Off-the-shelf, 3
OIDC (OpenID Connect), 577
One-time pad, 774, 807
One-time password, 52, 67, 244
One-way hash function, 799
Onion Routing, 443, 635
Online profile, 627
Open design, design principle, 217, 315
Open mode, wireless network, 384
Open System Interconnection (OSI) model, see OSI
Openness, of private data collection, 597
Operating system, 136, 279–340, 513
abstraction in, 289
Apple iOS, 302
Apple Mac OS, 291
Audit in, 292
authentication by, 283
boot, 280
complexity of design, 291
complexity of, 187, 309
concurrency in, 286
correctness, 314
DEC VAX, 314
design of, 308, 820
device control, 283
device driver, 288
domain, 286
DOS, 302
for smartphone, 818
hierarchical design in, 311
history, 284
hypervisor, 292
kernel, 284, 312
layered design, 309
Linux, 291
loading in stages, 291
Mach, 302
monitor, 285
multiprogramming, 285
multitasking, 283
process, 286
resource allocation, 286
rootkit, 329
Scomp, 323
security kernel, 312
self-protection of, 290
simplicity of design, 309
single-user, 284
smartphone, 818
startup, 280
task, 286
thread of execution, 286
trusted system, 316
Unix, 291
virtualization, 292
Windows, 291, 302
Opportunity, of attack, 26
Optical fiber cable, interception from, 346
Opt-in botnet, 430
Orange Book, see Trusted Computer System Evaluation Criteria
Organized crime, 15, 19, 177
Original work of authorship, 705
Originality, and copyright, 706
OSI (Open System Interconnection) model, 350, 433, 435, 439, 455, 462
Out-of-band communication, 244
Overflow,
buffer, see Buffer overflow
data, 149
integer, 160
parameter, 140
segment, 306
table, 143
Overload, denial of service, 11, 399
Oversight, of data collection, 603
Ownership, of data, 8, 594, 596
PaaS (Platform as a Service), 552, 557
Pacemaker, 4, 816
Packet, 351, 415, 458, 477
Packet filtering gateway, 456, 461, 467
Packet sniffer, 343
Page, offset in, 306
Page, size of, 307
Page-in-the-middle attack, 237
Paging, 306
combined with segmentation, 307
memory protection in, 307
Palin, Sarah, 39, 52
Parameter overflow, 140
Parameter, mismatch of, 162
Parity, 110
check, 109
even, 111
odd, 111
Parker, Donn, 18
Partial plaintext attack, 770
Passenger Name Record (PNR), 604
Password(s), 40–51, 266, 568, 610, 657, 738, 794, 844
attack on 42
choosing, 48
common, 45
concealment of, 46
database, 266
dictionary attacks on, 43
disclosure of, 41
forgotten, 42
guessing attack, 42–45
guessing, 761
inference of, 41, 43
loss of, 42
manager, 564
one-time, see One-time password
reuse of, 266
revocation of, 42
selection, 48
shared, 569
table, 46
use of, 41
variations of, 50
weak, 568
written 50, 51
Patch, 419
  hasty, 816
  penetrate and, See Penetrate-and-patch program, 172, 184, 731, 733
timeliness, 732
Patent, 711
  employment and, 727
  enforcement, 713
  for software, 713
  infringement, 712–713
  invalidation of, 713
  license of technology, 712
  nonobviousness requirement, 712
  novelty requirement, 712
  of employee’s invention, 727
  of RSA, 802
  ownership of, 726
  prior invention, 713
  registration of, 712
  RSA algorithm, 802
  search of previous inventions, 712
  software, 713
  tangible invention, 711
Path, network, 359
Pattern analysis, against encryption, 769
Pattern, malicious code, 192, 198, 200
Pattern-matching
  for intrusion detection, 477, 479
  for malicious code, 192, 198, 200
PayPal, 621
PBKDF2, 562, 564
P-box, in DES, 787
PCI DSS (Payment Card Industry Data Security Standard), 555
Peer-to-peer file sharing, 629, 707
Penetrate-and-patch, program assurance myth, 224, 336, 733, 816
Penetration testing, 218
Pentium chip, Intel, floating-point error, 10
People, as asset, 671
Performance testing, 211
Performance, database, 511
Perimeter,
  network, 359
  security, 354, 471
Permission, for data access, 596
Permission-based, design principle, 217, 218, 316
Permutation step, in DES, 96
Permutation, in cryptography, 782
Persistent cross-site scripting attack, 262
Persistent virus, 168
Personal computer, backup, 696
Personal data, 820
Personal firewall, 464
Perturbation, data, 529, 534, 617
Petmail, 240
PGP, 276, 633
Phishing, 274, 635
Phishing attack,
  G20 summit partners, 275
  Korean diplomats, 275
  RSA Corp., 275
  White House staffers, 275
Phone, mobile, 4
Photon
  gun, 811
  orientation of, 807
  reception of, 808
Physical access, 773
Physical access, unauthorized, 689
Physical connection failure, 420
Physical integrity, database, 507
Physical protection, of computer, 284
Physical security, 447
Physical security, for separation, 296
PIN, in authentication, 40, 67, 244
Ping of death, 404
Ping, 477
Piracy, of intellectual property, 707
Plaintext, 88, 96, 103, 434
Plaintext and ciphertext attack, 770
Plaintext-only attack, 770
Plan, incident response, see Incident response plan
Plan, security, see Security plan
Planning, contingency, 694
Point-to-point communication, 633
Poisoning, DNS cache, 418
Polarizing filter, 808
Policy, 72
  access control, 9, 12
  privacy, 600, 601, 609, 626
  security, 453, 466, 649
Politics, and cyberwarfare, 850
Polymorphic virus, 193
Poor programming practice, 158
POP (Post Office Protocol), 353, 370
server, 633
Pop-up ad, 630
Porras, Phil, 428
Port, 353, 370, 472
Port scan, network, 369, 450, 456, 476
Post Office Protocol (POP), see POP
Power
  consumption, 817
  loss, 688
  spike, 688
supply, uninterruptible, 688
surge, 688
Precaution, 22
Precision, of data, 11, 530
Precision, with risk analysis, 684
Predictability, in Dual-EC, 806
Prediction, from RFID sensor, 639
Predictive value, in authentication, 57
Preferred association, in wireless network, 386
Pretty Good Privacy, see PGP
Prevalence, in authentication, 56
Prevention, attack countermeasure, 28
Price
  of computer objects, 833
  on the Internet, 631
*Primary Colors*, 246
Privacy Act (U.S.), see U.S., Privacy Act
Privacy officer, 739
Privacy, 586
  access control for, 594
  accuracy of data, 596, 599, 603, 608
  adware, 629
  affected parties, 589
  anonymity and, 605
  anonymization, 597, 613
  breach notification law, 740
  children’s web access, 598
  cloud computing, 642
  collection limitation, 596
  commerce and, 604
  context of data use, 588, 601
  controlled disclosure, 587
  cookie, 627
  correctness, data, 596, 599, 608
  data accuracy, 596, 599, 603, 608
  data collection by government, 738
  data mining, 537
  data modification, 597
  data ownership, 592
  data quality, 596, 599, 603, 608
  data retraction, 594
  deception prohibited, 600
  determining sensitivity, 587
  disappearing email, 635
  disclosure, controlled, 587
  E.U. Data Protection Act, 742
  economics and, 832
  email monitoring and, 632, 633
  email remailer, 634
  encryption and, 597
  erroneous data, 608
  ethical aspect of, 752
  Europe, 603
  expectation of, 633
  fair market, 632
  financial service organization, 598
government data misuse and, 607
  government surveillance and, 645
  Gramm–Leach–Bliley Act, 739
  Hadoop, 544
  Internet user ID, 622
  laws, 597, 736
  limited data collection, 603
  limited data retention, 597
  limited use, 597
  linking of identities, 613
  loss of, 814
  medical data, 598, 739
  new technology and, 643
  notice of collection, 599, 600
  online profile, 627
  ownership, data, 592
  permission for access, 596
  policy statement of, 598
  RFID tag, 638
  safeguards for, 597
  security of collected data, 599
  specific purpose for use, 596, 603
  spyware, 629
  student records, 598
  telephony, 642
  U.S. e-Government Act, 599
  U.S. government websites, 599
  U.S. Privacy Act, 738
  versus confidentiality, 589
  voting, 641
  web bug, 628
Privacy-preserving data mining, 617
Private cloud, 552, 555
Private key, in cryptography, 101, 102, 109, 126
Privilege, 73, 85, 158
  escalation, 139, 145, 165
  least, see Least privilege
  list, in access control, 82
  limited, 75
  limited, in operating system, 317
  operating system, 139
  root, 329
  rootkit, 333
  separation of, design principle, 217, 317
Probability,
  classical, 676
  frequency, 676
  subjective, 676
Probable plaintext attack, against encryption, 770, 793
Probable value disclosure, 520
Procedure call, 136
Procedure oriented access control, 85
Process, 286, 320
Process activation, in operating system, 320
Processor, controlled access to, 283
Product cipher, encryption, 782
Product failure, redress for, 728
Profile,
online, 627
  protection, see Protection profile
  user, 68
Profit, motive for attack, 19
Program assurance myth,
penetrate-and-patch, 224
penetration testing, 218
security through (by) obscurity, 185, 226, 356, 836
Program assurance technique,
code review, 221
formal methods, 220
penetration testing, 218
proof of correctness, 219
testing, 211
validation, 221
Program complexity, 149
Program counter,
  modification of, 136, 148, 149
  protection of, 150
  vulnerability of, 147
Program design
  complete mediation, 217, 316
defense in depth, 218
defensive programming, 222
Design by contract, 223
ease of use, 217, 317
economy of mechanism, 217, 316
least common mechanism, 217, 317
least privilege, 216, 218, 316
open design, 217, 316
permission-based, 217, 218, 317
separation of privilege, 217, 317
simplicity, 217
validate input, 217
Program development practices, 216
cohesion, 206
encapsulation, 204, 206
information hiding, 204, 206
isolation, 203
modularity, 203, 204
mutual suspicion, 207
Program equivalence, 189, 218, 219
Program file, modification of, 177
Program flaw, 184
Program implementation, 150. See also Program
development practices
Program use, responsibility for use, 758
Program verification, 219
Program, resident, 188
Program, shared access to, 287
Program, terminate-and-stay-resident, 188
Programmer, responsibility for program use, 758
Programming error,
  buffer overflow, 134, 139, 145
  faulty serialization, 163
  input validation failure, 152
  off-by-one, 159
  race condition, 163
  synchronization, 163
  time-of-check to time-of-use, 159
  unchecked data, 153
Programming language, 150
Programming practice, poor 158
Project, database operation, 504
Promiscuous access point, 386
Proof of correctness, program assurance technique, 219
Propagation,
  access right, 77, 83
  encryption error, 778
  malicious code, 180
Property, as asset, 3
Property, legal rules of, 734
Proprietary software, 756
Prosecution, 426
Protected speech, 595
Protection, 3, 6, 75, 87
  consumer financial, 621
  cookie data, 625
  copyright, 704
  critical data, 281
  data, 11
  differentiated, 305
  for computer objects, 716, 717, 721
  inadequate, 608
  layered, 471
  memory, 284, 321
  mobile agent, 430
  of critical data, 281
  of data, 11
  of implanted medical device, 817
Protection profile, 328
Protocol, 351
Protocol, cryptographic key exchange, 105, 107
Protocol, WiFi, 376
Protocol analysis, stateful, 479
Proxy, application, see Application proxy firewall
Pseudonym
  and privacy, 606, 613
  for email, 634
  of an object, 77
PSOS (Provably Secure Operating System), 311, 326
Psychology, of attacker, 16–17
Public cloud, 552, 555, 561
Public domain, 705, 755
ECC in the, 802
Public hot spot, wireless, 382, 383
Public key cryptography, 89, 93, 100, 101, 102, 109, 118, 795, 802
for digital signature, 114, 116, 118
Public scrutiny, of cryptographic algorithm, 779
Pull, command-and-control update, 428
Purpose for data use, and privacy, 597
Push, command-and-control update, 428

Qualitative risk analysis, 677
Quality
of data, and privacy, 608
of service, as asset, 3
of software, 733, 816
Quantification, of security, 825
Quantitative risk analysis, 677
Quantum cryptography, 807
Quantum physics, 807
Query analysis, database, 535
Query language, database, 504
Query, database, 504

Rabbit, 170
Race condition, 163, 815
RACS (Redundant Array of Cloud Storage), 557
Radar, jamming, 844
Radiation therapy machine, 815
Radiation, for network interception, 343
Radio frequency identification, see RFID
Rainbow table, 47
Randell, Brian, 296
Random attack, 14
Random number generator, 775, 786, 792, 806
Random sample disclosure, database, 534
Randomization, of address space, 210
Range disclosure, database, 533
Ransom, 400, 425
Rate limiting, 490
Rationality, 831
RC2, 792
RC4, 389, 393, 792
RC5, 794
RC6, 795
Realm discovery, 571
Rearrangement, encrypted data, 786
Reasoning, ethical, 747
Record, database, 502
Record-keeping, incident, 664
Recovery, 198
attack countermeasure, 28
database, 516

from malicious code attack, 179
system, 74
Redirection
browser, 237
traffic, 413
Redundancy, 421, 428
backup and, 697
database, 506
Hadoop and, 543
network design, 367
testing, 109
Reference monitor, 76, 155, 313, 454
Reflections on Trusting Trust, 172
Register,
base, 298
bounds, 298
fence, 298
program counter, 136
stack pointer, 136, 146
Registration,
copyright, 708
patent, 712
Regression testing, 213
Regularity, in cryptography, 774
Regulation, 834
Relation, database, 504
Reliability, 421
data, 827
database, 513
software, 185
Relocation, program, 301
Remailer,
email, 634
multiple, 634
Remanence, magnetic, 325
Remote access Trojan horse (RAT), 170
Remote wipe, 559
Rent-a-bot, 429
Repetition, in cryptanalysis, 776
Replacement, of data, 3, 4
Replacement virus, 182
Replay attack, 432
authentication credentials, 365
network communication, 364
Replication, data, 697
Reputation, as asset, 671
Requirements, security, 212, 649, 651
Resident routine, 188
Resident virus, 168
Residual risk, 23
Resilience, network, 847
Resolution, addressing, 414
Response team, 663
Response, timeliness of, 11
Index

Responsibility, for security implementation, 650, 653
Retraction, data, 594
Return address
  spoofing (network), 406
  subprocedure, 139
Reuse,
  authentication data, 243
  digital signature, 802
  object, 325
  serial, 287
  software, 206
Reuter, 19
Reverse engineering, 714, 817
Review, program, 158, 819
Revocation, of access, 76
Revolving backup, 695
RFID, 636
  device, 817
  reader, 638
  tag, 529, 636
Right versus wrong, 747
Rights of individuals to privacy, 597
Rijndael, 98, 790. See also AES
Ripple effect, 827
Risk, 824
  analysis, see Risk analysis
  assumption of, 669
  avoidance of, 669
  communication of, 831
  data access and, 607
  exposure, 669
  extreme events, 24–25
  leverage, 669
  management, 22
  perception of, 25, 831
  residual, see Residual risk
  transfer of, 669
Risk analysis, 23, 650, 668
  accuracy of, 685
  benefits, 684
  control selection, 680
  difficulty to perform, 685
  disadvantages, 684
  exposure, 681
  lack of accuracy, 685
  likelihood estimation, 676
  qualitative, 677
  quantitative, 677
Rivest, Ronald, 103, 107, 792, 795, 800
Rivest–Shamir–Adelman cryptographic algorithm (RSA), see RSA
Rogues, 170, 465, 474, 612
  Alureon, 334, 336
  detection of, 334
  eradication of, 334
  in operating system, 329
  in phone, 329
  integration of, 332
  mobile phone, 329
  operating system, 329
  Sony XCP, 335
  stealth, 333, 335
  TDL-3, 334
Round, in AES, 98
Rounded disclosure, 533
Router, 351, 352, 401, 492. See also Routing
  screening, see Packet filtering gateway
Routing, 352, 355, 359, 367, 410, 413, 434, 436
RSA Corp., 275, 439, 779, 795, 804
RSA encryption
  algorithm, 102
  cryptanalysis of, 103
  key selection in, 798
  speed of encryption, 103
Rule of engagement, 848
Rule of evidence, 734
Rushby, John, 296
Russia, 19, 391, 397, 743, 843, 845
S/MIME, 277, 633
SaaS (Software as a Service), 552, 557
Safe harbor, 604, 742
Safe language, 149
Safeguards, privacy, 597
Safety, 815
Salt, password table, 47
Saltzer, Jerome, 75, 202, 216, 315, 735
SAML (Security Assertion Markup Language), 570
  Asserting Party, see IdP
  Assertion, 572
  Authentication Request, 572
  Authentication Response, 572
  IdP (Identity Provider), 571
  Relying Party, see SP
  SP (Service Provider), 571
  Subject, 571
  Token, see Authentication Response
Sample size concealment, database, 534
Sampling, 55
  statistical, 532
San Diego Supercomputer Center, 18
Sandbox, 294
Sanitization, object, 325
Sasser, 431
SATAN (password administration tool), 43, 369
Satellite communication, network, 346
S-box, in DES, 787, 789
Scan, port, 369, 456
Scareware, 170, 195
Schaefer, Marv, 221
Schell, Roger, 172, 219, 225
Schema, database, 502
Schneier, Bruce, 801
Schroeder, Michael, 75, 216, 315
Scamp, 323, 326
Scope, incident, 667
Screening router, see Packet filtering gateway
Script(ed) attack, 261, 423, 839
Script kiddies, 196
Seal, tamper detecting, 108, 112, 113
Secrecy,
assurance myth, 227
code, 158, 185, 846
communication, 116
encryption, 777
programming, 184
security weakness of, 158, 185, 836
voting, 837
Secret key encryption, see Symmetric encryption
Secret, shared, in authentication, 243
Secure Hash Standard (SHS), see SHS
Secure programming, see also Program development practices
Secure Socket Layer (SSL), see SSL
SecurID authentication token, see SSL
Security
add-on, 364
association, in IPsec, 444
computer, 2
cost of, 171
designing for, 212
kernel, 287, 312, 322
operations center (SOC), 397, 492, 666
perimeter, 354
physical, 447
policy, 72, 466
program development, 158
quantifying, 825
software design and, 310
success of, 32
through (by) obscurity, 185, 226, 356, 836
Security Essentials, Microsoft security tool, 250
Security Information and Event Management (SIEM), 492, 493, 560, 568
Security plan, 648, 668
acceptance of, 656
controls, 653
extensibility, 655
maintenance of, 655
requirements, 653
responsibility for implementation, 653
risk analysis, 668
team members, 656
timetable, 655
Segment, offset in, 304
Segment, size of, 305, 307
Segmentation, combined with paging, 307
Segmentation, memory, 303
Selective backup, 696
Self-protection, of operating system, 290
Sensitive data, 587
access from smartphone, 818
control of, 814
database, 518, 529
disposal of, 692, 772
exposure of, 177, 818
interception of, 236, 692, 772
protection of, 603
RFID tags and, 638
timeliness of, 844
Sensitive information, subject to Freedom of Information act, 738
Sensitivity, in authentication, 56
Sensitivity, in data mining, 537
Sensor, 640, 815
Separation, 72, 259, 296, 305
code from data, 150
controlled, 296
cryptographic, 296
data, 11
layering, 310
logical, 296
malicious code countermeasure, 195
physical 296, 688
potential malicious code, 197
privilege, design principle, 217, 316
security kernel, 312
TCB from non-TCB code, 320
temporal, 296
using firewall, 452
virtualization, 292
Sequencing attack, 363
Sequencing, TCP, 415
Serial reuse, 287
Serialization flaw, 163
Server, 286
Server-side include, 265
Service, degradation of, 849
Service, denial of, see Denial of service
Service, theft of, 750
Session hijack attack, 386, 394, 415
Session, wireless communication, 393
Severity, of harm, 22
SHA, 113, 800
SHA-2, 800
SHA-256, 564
SHA-3, 801
Shadow field, database, 516
Shakespeare, 246
Shamir, Adi, 103, 107, 788, 795
Shannon, Claude, 90, 777
Shared data space, 141
Shared infrastructure, 566, 580
Shared key, encryption, 92
Shared passwords, 569
Shared secret, in authentication, 243
Shared use, operating system, 285
Sharing, 74
controlled, 287, 296
data, 287
enforced, 281
fair, 753
incident response information, 667
programs, 287
resource, 358
total, 296
Shielding, blocking electronic emanation, 693
Shift row, in AES, 790
Shneiderman, Ben, 654
Shock, electrical, 816, 817
Shopping, on the Internet, 630
Shredding, paper, 692
SHS ((Secure Hash Standard), see SHA
Shunning, 431, 490
SIEM, see Security Information and Event Management
Signature, digital, see Digital signature
Signature, malicious code, 192, 198, 200
Signature-based intrusion detection system, 476, 494
Signed code, 251
Silent Banker, 234
Simple Mail Transfer Protocol (SMTP), see SMTP
Simplicity,
encryption process, 778
program design principle, 217
program quality, 205
Simultaneous access, 11
Simultaneous execution, 286
Single point of failure, 55, 557
Single sign-on, 68, 461, 569
Single-key encryption, see Symmetric encryption
Single-user computer, 284
Sinkholing, 490
Situation assessment, by intrusion detection system, 488
Size, of ciphertext, 778
Skimmer, 324
Skimming, of authentication tokens, 67
Skype, 642
SLA (Service Level Agreement), 555, 567
Slammer, 172, 175
Small sample concealment, database, 534
Smart device, 814
Smartphone, 817
SMTP (Simple Mail Transfer Protocol), 273
SMTP server, 633
Smurf attack, 404
Snapchat, 635
Sniffer, 343, 345
Snow, Brian, 19
SoBig, 172, 175
SOC, see Security Operations Center
Social engineering, 50, 844
Software,
as asset, 3, 671
as asset, 671
cohesion of, 206
correctness of, 206, 728, 729
coupling, 206
encapsulation of, 206
failure, 6, 730
failure, 728
flaw reporting, 731
information hiding in, 206
license of, 727, 756
maintenance of, 205
ownership of, 725, 754
patching, 731, 733
proprietary, 756
quality of, 733
quality, 210, 221
reliability, 815
return of defective, 730
reuse of, 206
shrink-wrapped, 729
usability, 728
Software as a Service (SaaS), 552, 557
Software design, 310
damage control in, 311
hierarchical, 311
security kernel, 312
Software development practices, see Program development practices
Sony XCP rootkit, 335
Source address spoofing, 404
Source, in big data, 547
Spafford, Eugene, 761
Spam, 431, 633, 635, 740
advertising with, 270
fee to send, 273
laws, 271
links to malicious code sites, 271
outside legal jurisdictions, 271
pattern recognition, 272
pharmaceuticals, 270
pump-and-dump, 270
stocks, 270
U.S. Can Spam Act, 272
unreliable source address, 272
volume limitation, 272
volume of, 268
Spear phishing, 274, 844
Special operations, 842
Specificity, in authentication, 56
Speech. Protected, 595
Speed, of encryption, 126
Splicing,
cable, 344, 363
code modification, 337
Spoof(ing), 844
address, 413, 490
DNS, 409
e-mail, 635
source address, 404
Spying, 92, 845
Spyware, 170, 628, 630
SQL, 504
SQL injection attack, 263
Square, payment scheme, 621
SSH (Secure shell) encryption, 438
SSID (Security Set Identifier), 378, 381, 383
cloaking, 384
automatic connection to, 387
SSL (Secure Socket Layer) encryption, 235, 387, 438, 444, 561, 794
Apple bug, 213
big data applications, 548
lack of diversity in implementation of, 210
session in, 439
STaaS (Storage as a Service), see Cloud storage
Stack, 146
Stack frame, 146
Stack frame, protection of, 150
Stack memory, 136, 139
Stack pointer, 136, 146
Stack pointer, protection of, 150
Stack smashing, 145, 148
StackGuard (stack protection software), 150
Stalker, 820
Startup (automatically executing program), 181, 189
Startup, operating system 280, 323
State machine, 479
State-based intrusion detection system, 478
Stateful inspection firewall, 458
Stateful protocol analysis, 458
Statistical analysis, 477
in cryptanalysis, 776
Statistical sampling, 532
Statistics, web use, 626
Statute, see Law
Stealth, 487
mode, wireless network, 384
malicious code, 189, 190, 428
Steganography, 192
Stoll, Cliff, 295, 667
Storage, networked, for backup, 697
Strategy, business continuity, 660
strcpy, string copy utility, 162
Stream cipher, 793
Stream encryption, 93
Street View, Google, 378
Strength, of encryption, 97, 777
String copy program, 162
String
length, 161
null-terminated, 161
termination, 161
strncpy, string copy utility, 162
STU-III secure telephone, 244
Stuxnet, 20, 174, 175, 368, 843, 847
Subject, 38, 72
Subject, data, 9
Subjective probability, 676
Subnet, 450
Subprocedure, 139
Subschema, database, 502
Substitution, 363
attack, 363
encrypted data, 786
in AES, 98
in cryptography, 95, 103, 774
step, in DES, 96
Subtask, 204
Subversion, 815
Suit, contract law, 725
Suite B, cryptographic algorithms, 803
Supervisor, operating system, 136, 280
Suppression, data, 529
Suppression, limited response, 532
Surface, attack, see Attack surface
Surfing, and privacy, 624
Surge suppressor, 688
Surveillance, government, and privacy, 645
Survey results, comparability of, 830
Swapping, 303
Index

database, 535
value, 618
Sweeney, Latanya, 527, 615
Switching cloud providers, 556
Symmetric cipher, 789
Symmetric encryption, 88, 92, 96, 786
SYN flood attack, 405
SYN packet, 406
SYN–ACK, 406
Synchronization, 281
program, 163
TCP, 416
Syria, 844, 845
System log, 567, 582
System, computer, 3
System, trusted, see Trusted system
System, usability of, 12
Syverson, Paul, 443
Tablet computer, 818
Tag, RFID, 636
Tagged architecture, 301, 305
Tamper detection, 151
Tampering, data, 109
Tampering, protection against, 113
Tamperproof, reference monitor property, 76
Tamper-resistant seal, 840
Target, attractive, 27
Target Corp., 609, 616
Targeted attack, 14, 19
Targeting, behavioral, 626
Task, 286
background, 358
TCP connection, 415
TCP/IP, 439
TCSEC (Trusted Computer System Evaluation Criteria), see Trusted Computer System Evaluation Criteria
TDL-3 (malicious code), 334
TDSS rootkit, 336
Teardrop attack, 407
Telecommuter, 449
Teleology, 748
Telephony, privacy and, 642
Television, 1
Temperature, effect on semiconductor, 772
Tempest, 693
Template, for biometric authentication, 59, 62
Temporal Key Integrity Program (TKIP), 393
Terminate-and-stay-resident routine, 188
Terms of service, 643
Terms of use, 592, 763
Terrorism, 20–21
and privacy, 607
Testing, 210, 221
acceptance, 211
black-box, 214
clear-box, 214
completeness of, 214
coverage of, 214
effectiveness of, 215
function, 211
independent, 215
installation, 211
integration, 211
limitations of, 215
penetration, 218
performance, 211
regression, 213
unit, 211
TFN, see Tribal flood network
TFN2K, see Tribal flood network 2000,
The Cuckoo’s Egg, 668
Theft, 689, 692, 734
credit card, 19, 22
deterring, 692
identity, 609
Therac 25, 815
Third-party ad, 622
Third-party cookie, 625
Thompson, Ken, 43, 172
Thread, 163, 286
Threat, 5, 6, 8
Advanced Persistent, see Advanced Persistent Threat
for decision-making, 826
malicious, 14
network disconnection, 849
nonmalicious, 14, 420
Threat surface, 820
Threshold, 55
Ticket, access control mechanism, 82
Tiger team analysis, see Penetration testing
Time bomb, 170
Time
theft of, 750
response see Response time
value of, 824
wait, see Wait time
Timeliness, 777
data, 827
response, 11
sensitive data, 844
value of asset, 4
Time-of-check to time-of-use (TOCTTOU) error, 155
TJMaxx, data theft, 19, 391
TKIP (Temporal Key Integrity Program), 393
TLS (Transport Layer Security) encryption, see SSL
TNO (Trust No One), 562, 564
Index

TOCTTOU error, see Time-of-check to time-of-use error

Token,
  active, 66
  dynamic, 67
  for authentication, 65, 66
  passive, 66
  RFID, 636
  static, 66

Tolerance, fault, see Fault tolerance

Toolkit, 166, 170, 196

Top level domain, 414

Topology, network, 849

TOR (The Onion Router), see Onion routing

Tort law, 722

Totient function, Euler, 797

Tracker
  inference in database, 524
  web page, 623

Tracking
  active, 628
  Internet, 254, 620, 622, 623, 627
  passive, 627
  RFID tag, 638

Tracking bug, 254

Trade secret, 714, 720, 734
  enforcement, 714
  improper access to, 714
  ownership of, 727
  reverse engineering, 714
  secrecy of, 714

Trademark, 717

Traffic redirection, 413

Transfer, of risk, 669

Transient virus, 168

Translation, address, see Address translation

Transmission
  error, 361
  failure, 420
  of malicious code, 180

Transparency, and privacy, 629

Transparent image, Internet, 629

Transport mode, in IPsec, 446

Transposition, in cryptography, 95, 103, 774

Trapdoor, 158, 170, 356, 787, 790, 845

Treaty, 848

Trespassing, 761

Triad, C-I-A, see C-I-A triad

Triage, incident response, 664

Tribal flood network (TFN), 18, 424

Tribal flood network year 2000 edition (TFN2K), 18, 424

Trin00 (malicious software), 18, 424

Triple DES, 96–97, 98

Tripwire (modification detection program), 112, 165, 251, 481

Trojan horse, 169, 170, 423. See also Malicious code

Trove, Roland, 730

Trust, 76, 117, 172, 288, 310, 316, 409, 412, 454, 818, 838

 Trusted code, 289

Trusted Computer System Evaluation Criteria, 318, 323, 327, 651

Trusted Computing Base (TCB), 318, 319

Trusted path, 323

Trusted system, 316

Trustworthy Computing Initiative, Microsoft, 222, 326

Truth, 747, 762

Tuning, network, 431, 489

Tunnel mode, in IPsec, 446

Tunnel, encrypted, 448

Tuple, database, 504

Turn, Rein, 597

Tversky, Amos, 25

Twitter, 595

Two-factor authentication, 70

Two-phase update, database, 514

U.S. Children’s Online Privacy Protection Act (COPPA), 598

U.S. Computer Emergency Response Team (CERT), 424

U.S. Computer Fraud and Abuse Act, 738

U.S. Department of Defense, 7, 608, 694, 842

U.S. Department of Health, Education and Welfare, 596

U.S. Department of Justice, 15, 19, 610

U.S. Economic Espionage Act, 738

U.S. Federal Bureau of Investigation (FBI), 19, 20, 21, 61, 64

U.S. Federal Educational Rights and Privacy Act, 598

U.S. Federal Trade Commission, 599, 601, 610

U.S. Freedom of Information Act (FOIA), 738

U.S. Health Insurance Portability and Accountability Act (HIPAA), 598, 739, 753

U.S. National Bureau of Standards (NBS) 95, 97, 779, 788.
  See also U.S. National Institute of Standards and Technology

U.S. National Institute of Standards and Technology (NIST) 14, 95, 98, 429, 789, 800, 801, 806, 811.
  See also U.S. National Bureau of Standards

U.S. National Security Agency (NSA), 97, 781, 787, 788, 794, 801, 803, 805, 806

U.S. Privacy Act, 597, 738

U.S. Uniform Commercial Code (UCC), 729

U.S.A. Patriot Act, 740

UCC, see U.S. Uniform Commercial Code

Ukraine, 166, 845

Unchecked data, 153

Undecidability, see Decidability

Undocumented access point, 27, 157. See also Backdoor
Index

Unintentional error, 6, 420
Uninterruptible power supply, 688
Unique identity, 606
Unit testing, 211
United States, 15, 19, 211, 743, 772, 843, 846
Unix, 81, 291, 329
Unsafe code, 150
Unterminated string, 161
Usability, 51, 75, 242
in the large, 51, 52
in the small, 51, 52
system 12
voting system, 841
Use,
asset, 7
data, 11, 608
User, 72
User interface, 815, 840
User-in-the-middle, 237
Utilitarianism, 749
Utility program, 284
Validation, program assurance technique, 221
Value swapping, 618
Value, of asset, 4, 6, 21
Value, of data, 736
Vandalism, 689
Variability, in biometric authentication, 55, 59, 64
VAX, DEC computer, 290
Vendor lock-in, 556
Venema, Wietse, 369
Verifiability, reference monitor property, 76
Verification, program, see Program verification
Verizon Breach Report, 171
Vernam cipher, 775
Viewing, data, 9
Viewing, asset, 7
Virginia, 421
Virtual infrastructure, 581
Virtual machine, 292, 579
Virtual memory, 303
Virtual private network (VPN), 447, 492, 633
Virtualization, in operating system, 292
Virus, 167, 329
appended, 181
attachment of, 188
boot sector, 187
destructive, 176
detector, 198–199, 295
document, 180
encrypting, 194
hoax, 176
memory-resident, 188
multipartite, 178
persistent, 168
polymorphic, 193
resident, 168
transient, 168
See also Malicious code
VM (Virtual Machine), 558, 567, 579
Voice over IP, see VOIP
VoIP, 642
Volume-based attack, denial of service, see Volumetric attack
Volumetric attack, 398, 399, 423
Voting, electronic,
casting a ballot, 834
counting ballots, 836
privacy, 641
VPN, see Virtual private network
Vulnerability, 5, 6, 28
backdoor, 158
disclosure of, 731–733, 760, 833
disclosure, full, 760
disclosure, partial, 760
electronic voting, 834
exploitation, 419
finding, 760
for decision-making, 826
paper-based election, 834
race condition, 163
reporting, responsible, 732
risk analysis, 672
scanning, 431, 482
search for, 761
toolkit, 166, 419
trapdoor, 158
undocumented entry, 158
zero-day, 172
Vulnerability–threat–control paradigm, 5
Wait time, 11
Waladec, spam network, 269, 429
War driving, network, 382
War of the Worlds, 2
Ware, Willis, 13, 172, 318, 596, 597
Warfare, conventional, 842, 846
Warfare, cyber, see Cyber warfare
Watergate burglary, 596
Watermark, digital, 710
Weak encryption, 388
Weak passwords, 568
Weakness, 5
in cryptography, 806
Weapon,
cyber, 847
kinetic, 847
Web [the], see Internet
Web
  bug, 254, 627
  content, false, 246
  hosting, 566
  page, fake, 117
  site defacement, 20, 246
  site, fake, 249
  site, privacy of, 599, 600
Welke, Stephen, 10–11
Welles, Orson, 2
Wells, H. G., 2
WEP (Wired Equivalent Privacy), 379, 388, 398, 794
  weaknesses in, 389–390
Whistle blower, 613
White hat hacking, 759
White House, victim of phishing attack, 275
Whitelisting, application, 581
Whittaker, James, 210, 211, 214
WiFi
  communication, 364, see also Wireless network
  frame, 379
  signal interception, 391
WikiLeaks, 473, 486, 595, 620
Wild card, in access control, 81
Windows operating system, 291, 302, 339, 818
Wireless client, 364
Wireless communication, 364, 376, 816. See also WiFi
  communication
Wireless network
  association, 380
  authentication, 380
  authentication, 385
  availability in, 382
  base station, 382
  broadcast mode, 384
  closed mode, 384
  confidentiality, 381
  encryption in, 383
  integrity in, 381
  open mode, 384
  rogue access point, 383
  rogue host in, 384
  stealth mode, 384
Wireless network vulnerability,
  association hijacking, 386
  authentication, nonexistent, 390
availability, 382
confidentiality, 381
encryption initialization vector collision, 389
faulty integrity check, 390
incomplete authentication, 394
integrity, 381
integrity check, 390
integrity failure, 395
MAC address spoofing, 394
man-in-the-middle, 394
no authentication, 390
promiscuous access point, 386
rogue host, 384
session hijack, 394
short encryption key, 388
static encryption key, 388
weak encryption, 388
Wiretap attack, 242, 343, 344, 354, 355, 360, 628, 739, 770, 771
Word macro virus, 10
Work factor, 227
Work factor, cryptographic, 91, 97
Work for hire, 726
World War II, 89, 107, 771, 772
Worm, 168. See also Malicious code
Worm, Morris, see Morris worm
WPA (WiFi Protected Access), 390, 794
WPA attack,
  MAC address spoofing, 394
  man-in-the-middle, 394
XCP, Sony rootkit, 335
XMLDSig (XML digital signature), 572
x-ray, 815
Yes-or-no test, in authentication, 56, 62
Yoran, Amit, 209
Zatko, Peter (Mudge), 139–140
Zero-day exploit, 172 419
Zeus, vulnerability toolkit, 245, 419
Zip code, U.S. 615
Zombie, 170, 423, 426