

MICHAEL GREGG
OMAR SANTOS

Cert Guide

Learn, prepare, and practice for exam success



CEH

Certified Ethical Hacker

Version 10

PEARSON IT
CERTIFICATION

Third Edition

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



Certified Ethical Hacker (CEH) Version 10 Cert Guide

Third Edition

Michael Gregg
Omar Santos



Pearson

Copyright © 2020 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-7897-6052-4

ISBN-10: 0-7897-6052-5

Library of Congress Control Number: 2019940687

ScoutAutomatedPrintCode

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Pearson IT Certification cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided “as is” without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screenshots may be viewed in full within the software version specified.

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. Screenshots and icons reprinted with permission from the Microsoft Corporation. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact
governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact
intlcs@pearson.com.

Publisher

Mark L. Taub

Product Line Manager

Brett Bartow

Acquisitions Editor

Paul Carlstroem

Development Editor

Rick Kughen

Managing Editor

Sandra Schroeder

Senior Project Editor

Tonya Simpson

Copy Editor

MediaMix Productions,
LLC

Indexer

Tim Wright

Proofreader

Abigail Manheim

Technical Editors

Michael Angelo

Ron Taylor

Publishing Coordinator

Cindy Teeters

Cover Designer

Chuti Prasertsith

Compositor

codeMantra

Contents at a Glance

	Introduction	xxii
CHAPTER 1	An Introduction to Ethical Hacking	3
CHAPTER 2	The Technical Foundations of Hacking	45
CHAPTER 3	Footprinting and Scanning	87
CHAPTER 4	Enumeration and System Hacking	149
CHAPTER 5	Social Engineering, Malware Threats, and Vulnerability Analysis	195
CHAPTER 6	Sniffers, Session Hijacking, and Denial of Service	273
CHAPTER 7	Web Server Hacking, Web Applications, and Database Attacks	325
CHAPTER 8	Wireless Technologies, Mobile Security, and Attacks	407
CHAPTER 9	IDS, Firewalls, and Honeypots	455
CHAPTER 10	Cryptographic Attacks and Defenses	503
CHAPTER 11	Cloud Computing, IoT, and Botnets	547
CHAPTER 12	Final Preparation	573
	Glossary	577
APPENDIX A	Answers to the “Do I Know This Already?” Quizzes and Review Questions	603
	Index	639

Table of Contents

	Introduction	xxii
Chapter 1	An Introduction to Ethical Hacking	3
	“Do I Know This Already?” Quiz	3
	Security Fundamentals	6
	Goals of Security	7
	Risk, Assets, Threats, and Vulnerabilities	8
	Backing Up Data to Reduce Risk	10
	Defining an Exploit	11
	Risk Assessment	12
	Security Testing	13
	No-Knowledge Tests (Black Box)	13
	Full-Knowledge Testing (White Box)	14
	Partial-Knowledge Testing (Gray Box)	14
	Types of Security Tests	14
	Hacker and Cracker Descriptions	16
	Who Attackers Are	18
	Ethical Hackers	19
	Required Skills of an Ethical Hacker	20
	Modes of Ethical Hacking	21
	Test Plans—Keeping It Legal	24
	Test Phases	25
	Establishing Goals	26
	Getting Approval	27
	Ethical Hacking Report	28
	Vulnerability Research—Keeping Up with Changes	29
	Ethics and Legality	29
	Overview of U.S. Federal Laws	30
	Compliance Regulations	33
	Payment Card Industry Data Security Standard (PCI-DSS)	34
	Summary	35
	Review All Key Topics	36
	Define Key Terms	36

Exercises	36
1-1 Searching for Exposed Passwords	37
1-2 Examining Security Policies	37
Review Questions	37
Suggested Reading and Resources	42
Chapter 2 The Technical Foundations of Hacking	45
“Do I Know This Already?” Quiz	45
The Hacking Process	48
Performing Reconnaissance and Footprinting	48
Scanning and Enumeration	49
Gaining Access	50
Escalation of Privilege	51
Maintaining Access	51
Covering Tracks and Planting Backdoors	51
The Ethical Hacker’s Process	52
NIST SP 800-15	53
Operationally Critical Threat, Asset, and Vulnerability Evaluation	53
Open Source Security Testing Methodology Manual	54
Information Security Systems and the Stack	54
The OSI Model	55
Anatomy of TCP/IP Protocols	57
The Application Layer	59
The Transport Layer	64
Transmission Control Protocol	64
User Datagram Protocol	66
The Internet Layer	66
Traceroute	72
The Network Access Layer	74
Summary	76
Review All Key Topics	77
Define Key Terms	77
Exercises	78
2.1 Install a Sniffer and Perform Packet Captures	78
2.2 List the Protocols, Applications, and Services Found at Each Layer of the Stack	79

2.3 Using Traceroute for Network Troubleshooting	79
Review Questions	80
Suggested Reading and Resources	84
Chapter 3 Footprinting and Scanning	87
“Do I Know This Already?” Quiz	87
Overview of the Seven-Step Information-Gathering Process	90
Information Gathering	90
Documentation	91
The Organization’s Website	91
Job Boards	93
Employee and People Searches	95
EDGAR Database	98
Google Hacking	99
Usenet	103
Registrar Query	104
DNS Enumeration	107
Determining the Network Range	112
Traceroute	114
Identifying Active Machines	115
Finding Open Ports and Access Points	116
Nmap	124
SuperScan	128
THC-Amap	128
Hping	129
Port Knocking	129
War Driving	130
OS Fingerprinting	130
Active Fingerprinting Tools	132
Fingerprinting Services	134
Default Ports and Services	134
Finding Open Services	134
Mapping the Network Attack Surface	136
Manual Mapping	136
Automated Mapping	136

Summary	138
Review All Key Topics	139
Define Key Terms	139
Exercises	140
3.1 Performing Passive Reconnaissance	140
3.2 Performing Active Reconnaissance	141
Review Questions	141
Suggested Reading and Resources	146
Chapter 4 Enumeration and System Hacking	149
“Do I Know This Already?” Quiz	149
Enumeration	152
Windows Enumeration	152
Windows Security	154
NetBIOS and LDAP Enumeration	155
NetBIOS Enumeration Tools	157
SNMP Enumeration	160
Linux/UNIX Enumeration	161
NTP Enumeration	162
SMTP Enumeration	162
IPsec and VoIP Enumeration	162
DNS Enumeration	163
System Hacking	163
Nontechnical Password Attacks	164
Technical Password Attacks	165
Password Guessing	165
Automated Password Guessing	167
Password Sniffing	167
Keylogging	168
Privilege Escalation and Exploiting Vulnerabilities	169
Exploiting an Application	170
Exploiting a Buffer Overflow	171
Owning the Box	173
Windows Authentication Types	173

	Cracking Windows Passwords	175
	Linux Authentication and Passwords	178
	Cracking Linux Passwords	180
	Hiding Files and Covering Tracks	181
	Rootkits	182
	File Hiding	185
	Summary	186
	Review All Key Topics	187
	Define Key Terms	187
	Exercise	188
	4.1 NTFS File Streaming	188
	Review Questions	189
	Suggested Reading and Resources	193
Chapter 5	Social Engineering, Malware Threats, and Vulnerability Analysis	195
	“Do I Know This Already?” Quiz	195
	Social Engineering	199
	Phishing	200
	Pharming	200
	Malvertising	201
	Spear Phishing	202
	SMS Phishing	209
	Voice Phishing	210
	Whaling	210
	Elicitation, Interrogation, and Impersonation (Pretexting)	210
	Social Engineering Motivation Techniques	212
	Shoulder Surfing and USB Key Drop	212
	Malware Threats	213
	Viruses and Worms	213
	Types and Transmission Methods of Viruses and Malware	213
	Virus Payloads	215
	History of Viruses	216
	Well-Known Viruses and Worms	217
	Virus Creation Tools	219
	Trojans	220
	Trojan Types	220

Trojan Ports and Communication Methods	221
Trojan Goals	222
Trojan Infection Mechanisms	223
Effects of Trojans	224
Trojan Tools	225
Distributing Trojans	227
Wrappers	228
Packers	229
Droppers	229
Crypters	229
Ransomware	230
Covert Communication	232
Tunneling via the Internet Layer	233
Tunneling via the Transport Layer	236
Tunneling via the Application Layer	237
Port Redirection	238
Keystroke Logging and Spyware	240
Hardware Keyloggers	241
Software Keyloggers	241
Spyware	242
Malware Countermeasures	243
Detecting Malware	243
Antivirus	246
Analyzing Malware	249
Static Analysis	250
Dynamic Analysis	251
Vulnerability Analysis	253
Passive vs. Active Assessments	253
External vs. Internal Assessments	254
Vulnerability Assessment Solutions	254
Tree-based vs. Inference-based Assessments	255
Vulnerability Scoring Systems	255
Vulnerability Scanning Tools	259
Summary	260

	Review All Key Topics	262
	Define Key Terms	263
	Command Reference to Check Your Memory	263
	Exercises	264
	5.1 Finding Malicious Programs	264
	5.2 Using Process Explorer	265
	Review Questions	266
	Suggested Reading and Resources	271
Chapter 6	Sniffers, Session Hijacking, and Denial of Service	273
	“Do I Know This Already?” Quiz	273
	Sniffers	276
	Passive Sniffing	277
	Active Sniffing	277
	Address Resolution Protocol	278
	ARP Poisoning and MAC Flooding	279
	Tools for Sniffing	286
	Wireshark	286
	Other Sniffing Tools	289
	Sniffing and Spoofing Countermeasures	290
	Session Hijacking	291
	Transport Layer Hijacking	292
	<i>Identify and Find an Active Session</i>	292
	<i>Predict the Sequence Number</i>	293
	<i>Take One of the Parties Offline</i>	295
	<i>Take Control of the Session</i>	295
	Application Layer Hijacking	295
	Session Sniffing	295
	<i>Predictable Session Token ID</i>	296
	<i>Man-in-the-Middle Attacks</i>	296
	<i>Client-Side Attacks</i>	297
	<i>Man-in-the-Browser Attacks</i>	299
	<i>Session Replay Attacks</i>	299
	<i>Session Fixation Attacks</i>	299
	Session Hijacking Tools	299
	Preventing Session Hijacking	302

Denial of Service and Distributed Denial of Service	303
DoS Attack Techniques	305
Volumetric Attacks	305
SYN Flood Attacks	306
ICMP Attacks	306
Peer-to-Peer Attacks	307
Application-Level Attacks	307
Permanent DoS Attacks	309
Distributed Denial of Service	309
DDoS Tools	310
DoS and DDOS Countermeasures	312
Summary	315
Review All Key Topics	316
Define Key Terms	317
Exercises	317
6.1 Scanning for DDoS Programs	317
6.2 Using SMAC to Spoof Your MAC Address	318
6.3 Using the KnowBe4 SMAC to Spoof Your MAC Address	318
Review Questions	319
Suggested Reading and Resources	323
Chapter 7 Web Server Hacking, Web Applications, and Database Attacks	325
“Do I Know This Already?” Quiz	325
Web Server Hacking	328
The HTTP Protocol	328
Scanning Web Servers	336
Banner Grabbing and Enumeration	337
Web Server Vulnerability Identification	342
Attacking the Web Server	342
DoS/DDoS Attacks	343
DNS Server Hijacking and DNS Amplification Attacks	343
Directory Traversal	345
Man-in-the-Middle Attacks	347
Website Defacement	347
Web Server Misconfiguration	347

HTTP Response Splitting	348
Understanding Cookie Manipulation Attacks	348
Web Server Password Cracking	349
Web Server-Specific Vulnerabilities	349
Comments in Source Code	351
Lack of Error Handling and Overly Verbose Error Handling	352
Hard-Coded Credentials	352
Race Conditions	352
Unprotected APIs	353
Hidden Elements	356
Lack of Code Signing	356
Automated Exploit Tools	356
Securing Web Servers	358
<i>Harden Before Deploying</i>	358
<i>Patch Management</i>	359
<i>Disable Unneeded Services</i>	359
<i>Lock Down the File System</i>	360
<i>Log and Audit</i>	360
<i>Provide Ongoing Vulnerability Scans</i>	360
Web Application Hacking	361
Unvalidated Input	362
Parameter/Form Tampering	362
Injection Flaws	362
Understanding Cross-site Scripting (XSS) Vulnerabilities	363
Reflected XSS	364
Stored XSS	366
DOM-based XSS	367
XSS Evasion Techniques	368
XSS Mitigations	369
Understanding Cross-site Request Forgery Vulnerabilities and Related Attacks	371
Understanding Clickjacking	372
Other Web Application Attacks	372
Exploiting Web-Based Cryptographic Vulnerabilities and Insecure	

Configurations	374
Web-Based Password Cracking and Authentication Attacks	375
Understanding What Cookies Are and Their Use	377
URL Obfuscation	378
Intercepting Web Traffic	380
Securing Web Applications	381
Lack of Code Signing	383
Database Hacking	384
A Brief Introduction to SQL and SQL Injection	385
<i>SQL Injection Categories</i>	389
<i>Fingerprinting the Database</i>	391
<i>Surveying the UNION Exploitation Technique</i>	392
<i>Using Boolean in SQL Injection Attacks</i>	394
<i>Understanding Out-of-Band Exploitation</i>	394
<i>Exploring the Time-Delay SQL Injection Technique</i>	396
<i>Surveying Stored Procedure SQL Injection</i>	396
<i>Understanding SQL Injection Mitigations</i>	396
SQL Injection Hacking Tools	397
Summary	398
Review All Key Topics	399
Exercise	400
7.1 Complete the Exercises in WebGoat	400
Review Questions	400
Suggested Reading and Resources	405
Chapter 8 Wireless Technologies, Mobile Security, and Attacks	407
“Do I Know This Already?” Quiz	407
Wireless Technologies	410
Mobile Device Operation and Security	410
Mobile Device Concerns	412
Mobile Device Platforms	413
Android	414
iOS	417
Windows Mobile Operating System	417
BlackBerry	418

Mobile Device Management and Protection	418
Bluetooth	419
Radio-frequency Identification (RFID) Attacks	422
Wireless LANs	422
Wireless LAN Basics	423
Wireless LAN Frequencies and Signaling	424
Wireless LAN Security	425
<i>Installing Rogue Access Points</i>	428
<i>Evil Twin Attacks</i>	429
<i>Deauthentication Attacks</i>	429
Attacking the Preferred Network Lists	433
Jamming Wireless Signals and Causing Interference	433
War Driving	433
<i>Attacking WEP</i>	433
<i>Attacking WPA</i>	435
Wireless Networks Configured with Open Authentication	440
<i>KRACK Attacks</i>	440
<i>Attacking Wi-Fi Protected Setup (WPS)</i>	441
<i>KARMA Attack</i>	441
<i>Fragmentation Attacks</i>	441
Additional Wireless Hacking Tools	443
Performing GPS Mapping	443
Wireless Traffic Analysis	443
Launch Wireless Attacks	444
Crack and Compromise the Wi-Fi Network	444
Securing Wireless Networks	445
Site Survey	445
<i>Robust Wireless Authentication</i>	446
Misuse Detection	447
Summary	447
Review All Key Topics	448
Define Key Terms	448
Review Questions	449
Suggested Reading and Resources	452

Chapter 9	IDS, Firewalls, and Honeypots	455
	“Do I Know This Already?” Quiz	455
	Intrusion Detection and Prevention Systems	458
	IDS Types and Components	458
	Pattern Matching	461
	Protocol Analysis	463
	Heuristic-Based Analysis	463
	Anomaly-Based Analysis	464
	Global Threat Correlation Capabilities	465
	Snort	465
	IDS Evasion	470
	Flooding	470
	Insertion and Evasion	470
	Session Splicing	471
	Shellcode Attacks	471
	Other IDS Evasion Techniques	472
	IDS Evasion Tools	473
	Firewalls	474
	Firewall Types	475
	Network Address Translation	475
	Packet Filters	476
	Application and Circuit-Level Gateways	478
	Stateful Inspection	479
	Identifying Firewalls	480
	Bypassing Firewalls	484
	Honeypots	490
	Types of Honeypots	492
	Detecting Honeypots	493
	Summary	494
	Review All Key Topics	494
	Define Key Terms	495
	Review Questions	495
	Suggested Reading and Resources	500

Chapter 10 Cryptographic Attacks and Defenses 503

“Do I Know This Already?” Quiz	503
Functions of Cryptography	506
History of Cryptography	507
Algorithms	509
Symmetric Encryption	510
Data Encryption Standard (DES)	511
Advanced Encryption Standard (AES)	514
Rivest Cipher	514
Asymmetric Encryption (Public Key Encryption)	514
RSA	516
Diffie-Hellman	516
ElGamal	516
Elliptic Curve Cryptography (ECC)	516
Hashing	517
Digital Signature	518
Steganography	519
Steganography Operation	520
Steganographic Tools	521
Digital Watermark	524
Digital Certificates	524
Public Key Infrastructure	525
Trust Models	527
Single-Authority Trust	527
Hierarchical Trust	527
Web of Trust	528
Protocols, Applications, and Attacks	529
Encryption Cracking and Tools	531
Weak Encryption	534
Encryption-Cracking Tools	536
Summary	536
Review All Key Topics	537
Define Key Terms	538

Exercises	538
10.1 Examining an SSL Certificate	538
10.2 Using PGP	539
10.3 Using a Steganographic Tool to Hide a Message	540
Review Questions	540
Suggested Reading and Resources	545
Chapter 11 Cloud Computing, IoT, and Botnets	547
“Do I Know This Already?” Quiz	547
Cloud Computing	550
Cloud Computing Issues and Concerns	552
Cloud Computing Attacks	554
Cloud Computing Security	555
IoT	556
IoT Protocols	558
Hacking IoT Implementations	560
Botnets	560
Botnet Countermeasures	563
Summary	566
Review All Key Topics	566
Define Key Terms	567
Exercise	567
11.1 Scanning for DDoS Programs	567
Review Questions	568
Suggested Reading and Resources	570
Chapter 12 Final Preparation	573
Hands-on Activities	573
Suggested Plan for Final Review and Study	574
Summary	575
Glossary	577
Appendix A Answers to the “Do I Know This Already?” Quizzes and Review Questions	603
Index	639

About the Authors

Michael Gregg (CISSP, SSCP, CISA, MCSE, MCT, CTT+, A+, N+, Security+, CCNA, CASP, CISA, CISM, CEH, CHFI, and GSEC) directs the cyber security operations for a multinational organization that operates facilities worldwide. As the CISO, Michael is responsible for securing the organization's assets on a global scale. Michael is responsible for developing cost-effective and innovative technology solutions for security issues and for evaluating emerging technologies.

He has more than 20 years of experience in the IT field and holds two associate's degrees, a bachelor's degree, and a master's degree. In addition to coauthoring the first, second, and third editions of *Security Administrator Street Smarts*, Michael has written or coauthored 14 other books, including *Build Your Own Security Lab: A Field Guide for Network Testing* (Wiley, 2008); *Hack the Stack: Using Snort and Ethereal to Master the 8 Layers of an Insecure Network* (Syngress, 2006); *Certified Ethical Hacker Exam Prep 2* (Que, 2006); and *Inside Network Security Assessment: Guarding Your IT Infrastructure* (Sams, 2005).

Michael has testified before a U.S. congressional committee, has been quoted in newspapers such as the *New York Times*, and was featured on various television and radio shows, including NPR, ABC, CBS, Fox News, and others, discussing cyber security and ethical hacking. He has created more than a dozen IT security training classes. He has created and performed video instruction on many security topics, such as cyber security, CISSP, CISA, Security+, and others.

When not working, speaking at security events, or writing, Michael enjoys 1960s muscle cars and has a slot in his garage for a new project car.

You can reach Michael by email at MikeG@thesolutionfirm.com.

Omar Santos is an active member of the security community, where he leads several industry-wide initiatives and standard bodies. His active role helps businesses, academic institutions, state and local law enforcement agencies, and other participants that are dedicated to increasing the security of the critical infrastructure.

Omar is the author of more than 20 books and video courses and numerous white papers, articles, and security configuration guidelines and best practices. Omar is a principal engineer of the Cisco Product Security Incident Response Team (PSIRT), where he mentors and leads engineers and incident managers during the investigation and resolution of security vulnerabilities.

Omar has been quoted by numerous media outlets, such as The Register, Wired, ZDNet, ThreatPost, CyberScoop, TechCrunch, Fortune, Ars Technica, and more. Additional information about Omar can be obtained from h4cker.org and omarsantos.io. You can follow Omar on Twitter at [@santosomar](https://twitter.com/santosomar).

Dedications

Michael:

To my parents, Betty and Curly, who always stood behind me, encouraged me, and prayed that all my dreams would come true.

Omar:

I would like to dedicate this book to my lovely wife, Jeannette, and my two beautiful children, Hannab and Derek, who have inspired and supported me throughout the development of this book. I also dedicate this book to my father, Jose, and to the memory of my mother, Generosa. Without their knowledge, wisdom, and guidance, I would not have the goals that I strive to achieve today.

Acknowledgments

Michael:

I would like to say thanks to Grace, Michael, Paul, Tonya, and all the team at Pearson for helping make this book a reality.

Omar:

This book is a result of concerted efforts of various individuals whose help brought this book to reality. I would like to thank the technical reviewers, Ron Taylor and Michael F. Angelo, for their significant contributions and expert guidance. I would also like to express our gratitude to the team at Pearson, especially to Rick Kughen, Paul Carlstroem, Tonya Simpson, and Barbara Hacha for their help and continuous support throughout the development of this book.

We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Introduction

The EC-Council Certified Ethical Hacker (CEH) exam has become one of the leading ethical hacking and cybersecurity certifications available today. CEH is recognized by the industry as providing candidates with a solid foundation of hands-on security testing skills and knowledge. The CEH exam covers a broad range of security concepts to prepare candidates for the technologies that they are likely to be working with if they move into a role that requires hands-on security testing.

Let's talk some about what this book is. It offers you the information for what you need to know to pass the CEH exam. It's highly recommended that you spend time with the tools and software discussed in the book. You should also complete a number of practice tests to become more comfortable with the type of questions you will see on the exam and get used to completing 125 questions in four hours. Depending on your personal study habits or learning style, you might benefit from buying this book *and* taking a class.

NOTE After completing the CEH exam, candidates may elect to attempt the CEH Practical exam. Individuals who possess the CEH credential will be able to sit for the CEH Practical exam. This exam will test their limits in unearthing vulnerabilities across major operating systems, databases, and networks. The CEH Practical exam is a six-hour, hands-on exam that requires you to demonstrate the application of ethical hacking techniques, such as threat vector identification, network scanning, OS detection, vulnerability analysis, system hacking, and web app hacking.

Cert Guides are meticulously crafted to give you the best possible learning experience for the particular characteristics of the technology covered and the actual certification exam. The instructional design implemented in the Cert Guides reflects the nature of the CEH certification exam. The Cert Guides provide you with the factual knowledge base you need for the exams, and then take it to the next level with exercises and exam questions that require you to engage in the analytic thinking needed to pass the CEH exam.

EC-Council recommends that typical candidates for this exam have a minimum of 2 years of experience in IT security. In addition, EC-Council recommends that candidates have preexisting knowledge of networking, TCP/IP, and basic computer knowledge.

Now, let's briefly discuss what this book is not. It is not a book designed to teach you advanced hacking techniques or the latest hack. This book's goal is to prepare you for the CEH 312-50 exam, and it is targeted to those with some networking, OS, and systems knowledge. It provides basics to get you started in the world of ethical

hacking and prepare you for the exam. Those wanting to become experts in this field should be prepared for additional reading, training, and practical experience.

How to Use This Book

This book uses several key methodologies to help you discover the exam topics on which you need more review, to help you fully understand and remember those details, and to help you prove to yourself that you have retained your knowledge of those topics. Therefore, this book does not try to help you pass the exams only by memorization; instead, it is designed to help you truly learn and understand the topics.

The book includes many features that provide different ways to study so you can be ready for the exam. If you understand a topic when you read it but do not study it any further, you probably will not be ready to pass the exam with confidence. The features included in this book give you tools that help you determine what you know, review what you know, better learn what you don't know, and be well prepared for the exam. These tools include the following:

- **“Do I Know This Already?” Quizzes:** Each chapter begins with a quiz that helps you determine the amount of time you need to spend studying that chapter. The answers are provided in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes and Review Questions.”
- **Foundation Topics:** These are the core sections of each chapter. They explain the tools and hacking concepts, and explain the configuration of both for the topics in that chapter.
- **Exam Preparation Tasks:** This section lists a series of study activities that you should complete after reading the “Foundation Topics” section. Each chapter includes the activities that make the most sense for studying the topics in that chapter. The activities include the following:
 - **Review All Key Topics:** The Key Topic icon appears next to the most important items in the “Foundation Topics” section of the chapter. The Review All Key Topics activity lists the key topics from the chapter and their page numbers. Although the contents of the entire chapter could be on the exam, you should definitely know the information listed in each key topic. Review these topics carefully.
 - **Define Key Terms:** Although certification exams might be unlikely to ask a question such as “Define this term,” the CEH 312-50 exam requires you to learn and know a lot of tools and how they are used. This section lists some of the most important terms from the chapter, asking you to write a short definition and compare your answer to the Glossary.

- **Exercises:** One or more sample exercises at the end of many chapters list a series of tasks for you to practice, which apply the lessons from the chapter in a real-world setting.
- **Review Questions:** Each chapter includes review questions to help you confirm that you understand the content you just covered. The answers are provided in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes and Review Questions.”

Credit

Cover image © Chainarong06/Shutterstock.

Chapter opener images © Charlie Edwards/Photodisc/Getty Images.

Chapter 1, quote from the Electronic Communication Privacy Act from U.S. Code Sections 2510 and 2701.

Chapter 1, section “Payment Card Industry Data Security Standard (PCI-DSS),” list of PCI-DSS requirements from PCI-DSS.

Chapter 2, section “NIST SP 800-15,” four stages of security assessment, from “Technical Guide to Information Security Testing and Assessment,” <https://www.nist.gov/publications/technical-guide-information-security-testing-and-assessment>.

Chapter 2, section “Open Source Security Testing Methodology Manual,” OSSTMM list of key points from OSSTMM.org.

Figure 2-3, screenshot of Wireshark © Wireshark Foundation.

Figure 2-10, screenshot of Wireshark © Wireshark Foundation.

Figure 2-11, screenshot of Wireshark © Wireshark Foundation.

Figure 3-1, screenshot of Microsoft excel © Microsoft 2019.

Figure 3-2, screenshot of home page Zabasearch © 2019 Zabasearch.

Figure 3-3, screenshot of home page pipl © 2006–2019 pipl.

Figure 3-4, screenshot of Google web page © Google.

Figure 3-5, screenshot of FOCA © Telefónica Digital España.

Figure 3-7, screenshot of LoriotPro © LUTEUS SARL.

Figure 3-8, screenshot of Ping Capture © LUTEUS SARL.

Figure 3-11, screenshot of Zenmap © Nmap.

Figure 3-12, screenshot of Winfingerprint © Kirby Kuehl.

Figure 3-13, screenshot of Wireshark © Wireshark Foundation.

Figure 4-2, screenshot of DumpSec © Microsoft.

Figure 4-3, screenshot of Have I Been Pwned? © Superlative Enterprises Pty Ltd.

Figure 4-4, screenshot of Cain & Abel © Cain & Abel.

- Figure 5-3, screenshot of SET © David Kennedy.
- Figure 5-4, screenshot of SET © David Kennedy.
- Figure 5-5, screenshot of SET © David Kennedy.
- Figure 5-6, screenshot of SET © David Kennedy.
- Figure 5-7, screenshot of SET © David Kennedy.
- Figure 5-8, screenshot of SET © David Kennedy.
- Figure 5-9, screenshot of SET © David Kennedy.
- Figure 5-10, screenshot of SET © David Kennedy.
- Figure 5-14, screenshot of WannaCry Ransomware © AO Kaspersky Lab.
- Figure 5-17, screenshot of Wireshark © Wireshark Foundation.
- Figure 5-19, screenshot of Talos © Cisco Systems, Inc.
- Figure 5-20, screenshot of Process Explorer © Microsoft.
- Figure 6-3, screenshot of MAC Address © MAC flooding.
- Figure 6-5, screenshot of DNS Spoofing attack © Cain & Abel.
- Figure 6-6, screenshot of Wireshark © Wireshark Foundation.
- Figure 6-10, screenshot of Booter Sites © Google.
- Figure 7-6, screenshot of Netcraft tool Copyright © 1995–2019 Netcraft Ltd.
- Figure 7-7, screenshot of Wikto © Gareth Phillips.
- Figure 7-8, screenshot of ExploitDB.com © ExploitDB.com.
- Figure 7-11, screenshot of Damn Vulnerable Web Application © DVWA.
- Figure 7-12, screenshot of Damn Vulnerable Web Application © DVWA.
- Figure 7-13, screenshot of Damn Vulnerable Web Application © DVWA.
- Figure 7-14, screenshot of Burp Suite professional © 2019 PortSwigger Ltd.
- Figure 7-15, screenshot of OWASP ZAP © WASP.
- Figure 7-16, screenshot of SQL statement © 1999–2019 by Refsnes Data.
- Figure 7-18, screenshot of SQL injection © OWASP.
- Figure 7-19, screenshot of SQL injection © OWASP.
- Figure 7-20, screenshot of Webgoat © OWASP.
- Figure 7-21, screenshot of Blind SQL Injection © OWASP.

Chapter 7, section “iOS,” the quote “...eliminates security layers designed to protect your personal information and your iOS device and is a violation of the iOS end-user software license agreement and is grounds for Apple to deny service for the device” © Apple, Inc.

Figure 8-6, screenshot of airmon-ng © Thomas d’Otreppe de Bouvette.

Figure 8-7, screenshot of airodump-ng © Thomas d’Otreppe de Bouvette.

Figure 8-8, screenshot of aireplay-ng © Thomas d’Otreppe de Bouvette.

Figure 8-11, screenshot of airodump-ng © Thomas d’Otreppe de Bouvette.

Figure 8-12, screenshot of aireplay-ng © Thomas d’Otreppe de Bouvette.

Figure 8-13, screenshot of airodump-ng © Thomas d’Otreppe de Bouvette.

Figure 8-14, screenshot of aircrack-ng © Thomas d’Otreppe de Bouvette.

Figure 9-3, screenshot of Snort Alerts © Squert.

Figure 9-4, screenshot of Snort Alerts © Kibana.

Figure 9-7, screenshot of Router password crack © ifm Network Experts.

Figure 10-8, screenshot of S-Tools © Naman Dwivedi.

Figure 10-9, screenshot of S-Tools © Naman Dwivedi.

Figure 10-14, screenshot of Yellowpipe Internet services © Yellowpipe.com.

Figure 11-5, screenshot of Citadel © 2019 Malwarebytes.

Chapter 11, section “Cloud Computing,” list of the advantages of using a cloud-based service from Eric Simmon, “DRAFT - Evaluation of Cloud Computing Services.”

Chapter 11, section “Cloud Computing,” list of the essential characteristics of cloud computing from Eric Simmon, “DRAFT - Evaluation of Cloud Computing Services.”

Footprinting and Scanning

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz enables you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 3-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes and Review Questions.”

Table 3-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Overview of the Seven-Step Information-Gathering Process	1, 4, 6
Determining the Network Range	5
Identifying Active Machines	2, 3
Finding Open Ports and Access Points	10
Fingerprinting Services	7
Mapping the Network Attack Surface	8, 9

CAUTION The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Where should an ethical hacker start the information-gathering process?
 - a. Interview with company
 - b. Dumpster diving
 - c. Company's website
 - d. Interview employees

2. What is the common Windows and Linux tool that is used for port scanning?
 - a. Hping
 - b. Amap
 - c. Nmap
 - d. SuperScan

3. What does the Nmap **-sT** switch do?
 - a. UDP scan
 - b. ICMP scan
 - c. TCP full connect scan
 - d. TCP ACK scan

4. Which of the following would be considered outside the scope of footprinting and information gathering?
 - a. Finding physical addresses
 - b. Attacking targets
 - c. Identifying potential targets
 - d. Reviewing company website

5. During a security assessment you are asked to help with a footprinting activity. Which of the following might be used to determine network range?
 - a. ARIN
 - b. DIG
 - c. Traceroute
 - d. Ping host

6. You have been asked to gather some specific information during a penetration test. The “**intitle**” string is used for what activity?
 - a. Traceroute
 - b. Google search
 - c. Website query
 - d. Host scanning

7. During a footprinting exercise, you have been asked to gather information from APNIC and LACNIC. What are these examples of?
 - a. IPv6 options
 - b. DHCP servers
 - c. DNS servers
 - d. RIRs

8. CNAMEs are associated with which of the following?
 - a. ARP
 - b. DNS
 - c. DHCP
 - d. Google hacking

9. LorientPro is used for which of the following?
 - a. Active OS fingerprinting
 - b. Passive OS fingerprinting
 - c. Mapping
 - d. Traceroute

10. What scan is also known as a zombie scan?
 - a. IDLE scan
 - b. SYN scan
 - c. FIN scan
 - d. Stealth scan

Foundation Topics

Overview of the Seven-Step Information-Gathering Process

Footprinting is the first step of the hacking methodology, and it is all about gathering information. Most organizations share a tremendous amount of information and data through various channels, including their websites and social media pages, their employees, and even their help desks. Footprinting is about information gathering and is both passive and active. Reviewing the company's website is an example of passive footprinting, whereas calling the help desk and attempting to social engineer them out of privileged information is an example of active information gathering. Port scanning entails determining network ranges and looking for open ports on individual systems. The EC-Council divides footprinting and scanning into seven basic steps, as follows:

Key Topic

1. Information gathering
2. Determining the network range
3. Identifying active machines
4. Finding open ports and access points
5. OS fingerprinting
6. Fingerprinting services
7. Mapping the network attack surface

Many times, students ask for a step-by-step method of information gathering. Realize that these are just general steps and that ethical hacking is really the process of discovery. Although the material in this book is covered in an ordered approach, real life sometimes varies. When performing these activities, you might find that you are led in a different direction from what you originally envisioned.

Information Gathering

The information-gathering steps of footprinting and scanning are of utmost importance. Good information gathering can make the difference between a successful pen test and one that has failed to provide maximum benefit to the client. This information can be found on the organization's website, published trade papers,

Usenet, financial databases, or even from disgruntled employees. Some potential sources are discussed, but first, let's review documentation.

Documentation

One important aspect of information gathering is documentation. Most people don't like paperwork, but it's a requirement that you cannot ignore. The best way to get off to a good start is to develop a systematic method to profile a target and record the results. Create a matrix with fields to record domain name, IP address, DNS servers, employee information, email addresses, IP address range, open ports, and banner details. Figure 3-1 gives an example of what your information matrix might look like when you start the documentation process.

	A	B	C	D	E
1	Obtained Thru Search Engine	Results	Social Network Sites	Results	Website Footprinting
2	Employees		Profile		OS's
3	Login pages		News		Scripting
4	Portal URL's		Education		Job requests
5	Technologies		Family		Other
6	Email Footprinting	Results	People Search Sites	Results	Google Hacking
7	IP address		Date of birth		Files containing passwords
8	Email Address		Email		Error messages
9	Geo location		Photos		Other findings
10	Whois Footprinting	Results	Network footprinting	Results	DNS footprinting
11	Domain name		Network range		DNS servers
12	Contact details		Subnet mask		Zone transfer (Y/N)
13	Domain creation date		Traceroute findings		Types of Servers
14	Hosting company		Other data		DNSSEC (Y/N)

FIGURE 3-1 Documentation

Building this type of information early on will help in mapping the network and planning the best method of attack.

The Organization's Website

With the initial documentation out of the way, it's time to get started. The best place to begin is the organization's website. Search for the company's URL with Google, Bing, Dogpile, Shodan, or your search engine of choice. You will want to look for the following:

- **Company URL:** Domain name.
- **Internal URLs:** As an example, not only xyz.com but also support.xyz.com.
- **Restricted URLs:** Any domains not accessible to the public.
- **Internal pages:** Company news, employment opportunities, addresses, and phone numbers. Overall, you want to look for all open source information, which is information freely provided to clients, customers, or the general public.

NOTE One great tool to find internal URLs is Netcraft's "What's that site running?" tool on its home page. You can find it at <https://www.netcraft.com/>.

Let's look at an example of a local web hosting company. A quick review of its site shows it has a news and updates section. Recent news states the following:

We are proud to have just updated all of our servers to Plesk 10.0.1. Anyone logging in to these new servers as admin should use the username of the domain, for example, www.xyz.com. The passwords have been transferred from the old servers, so no password reset should be required. We used the existing domain administrator password. Our continued alliance with Extreme Networks has allowed us to complete our transition from Cisco equipment. These upgrades, along with our addition of a third connection to the Internet, give us a high degree of fault tolerance.

You might consider this good marketing information to provide potential clients. The problem is that this information is available to anyone who browses the website. This information allows attackers to know that the new systems are Linux based and that the network equipment is all Extreme Networks. If attackers were planning to launch a denial of service (DoS) attack against the organization, they now know that they must knock out three nodes to the Internet. Even a competitor would benefit from this knowledge because the company is telling the competition everything about its infrastructure.

In some cases, information may have been removed from a company website. That is when the Wayback Machine, at <https://archive.org>, is useful to browse archived web pages that date back to 1996. It's a useful tool for looking for information that no longer exists on a site.

NOTE Although the Wayback Machine is very useful for exploring old web pages, keep in mind that websites can be removed or blocked so that they are not listed.

Another big information leakage point is the company directories. These usually identify key employees or departments. By combining this information with a little social engineering, an attacker can call the help desk, pretend he works for one of these key employees, and demand that a password is reset or changed. He could also use biographical information about a key employee to perform other types of social engineering trickery. Kevin Mitnick used social engineering techniques.

During a pen test, you want to record any such findings and make sure to alert the organization as to what information is available and how it might be used in an attack.

One method to gain additional information about the organization's email server is to send an email that will bounce from the site. If the site is `www.xyz.com`, send a mail to `badaddress@xyz.com`. It will bounce back to you and give you information in its header, including the email server IP address and email server version. Another great reason for bouncing an email message is to find out whether the organization makes use of mail scrubbers. Whatever you find, you should copy the information from the headers and make a note of it as you continue to gather information.

Finally, keep in mind that it's not just logical information that you want to gather. Now is a good time to record all physical information about the targeted company. Location information is used to determine the physical location of the targeted company. Bing Maps and Google Earth are two tools that can be used to get physical layout information. Bing Maps is particularly interesting because it offers a 45-degree perspective, which gives a unique view of facilities and physical landmarks. This view enables you to identify objects such as entry points and points of ingress/egress.

In the Field: Free Speech and the Web

Although the Web has drastically enhanced the ability for people to communicate, not all countries allow free speech, and many restrict what their citizens can do or post online. Others even have plans to score citizens based on their online activity. The Chinese state is setting up one such system that will monitor the behavior of its population and rank them all based on their social credit.

This social credit system, which was first announced in 2014, is scheduled to be fully operational nationwide by 2020. However, it is already in place for millions of people across the country. The scheme will be mandatory.

The exact methodology is a secret—but examples of infractions include posting negative comments about the government, buying too many video games, and posting fake news online. You can read more at <https://www.vox.com/the-goods/2018/11/2/18057450/china-social-credit-score-spend-frivolously-video-games>.

Job Boards

If you're lucky, the company has a job posting board. Look this over carefully; you will be surprised at how much information is given here. If no job listings are posted

on the organization's website, get interactive and check out some of the major Internet job boards. Popular sites include the following:

- Careerbuilder.com
- Monster.com
- Dice.com
- Indeed.com

At the job posting site, query for the organization. Here's an example of the type of information usually found:

- Primary responsibilities for this position include management of a Windows 2019 Active Directory environment, including MS Exchange 2008, SQL 2016, and Citrix.
- Interact with the technical support supervisor to resolve issues and evaluate/maintain patch level and security updates.
- Experience necessary in Active Directory, Microsoft Clustering and Network Load Balancing, MS Exchange 2007, MS SQL 2016, Citrix MetaFrame XP, EMC CX-400 SAN-related or other enterprise-level SAN, Veritas Net Backup, BigBrother, and NetIQ Monitoring SW.
- Maintain, support, and troubleshoot a Windows 10 LAN.

Did this organization give away any information that might be valuable to an attacker? They actually have told attackers almost everything about their network. Just the knowledge that the organization is running Windows 10 is extremely valuable.

NOTE Discovering unsecured devices or infrastructure could be used to determine if a Bitcoin miner could successfully be placed on the victim's network without his knowledge.

One way to reduce the information leakage from job postings is to reduce the system-specific information in the job post or to use a company confidential job posting. Company confidential postings hide the true company's identity and make it harder for attackers to misuse this type of information.

Employee and People Searches

Security is not just about technical and physical controls. It's also about people. In many modern attacks, people are the initial target. All this really means is that an ethical hacker is also going to want to see what information is available about key personnel. Whereas websites, employee directories, and press releases may provide employee names, third-party sites have the potential to provide sensitive data an attacker might be able to leverage. We can categorize these sites as either data aggregation brokers or social networking.

A staggering number of data aggregation brokerage sites are on the Web. It is easy for an attacker to perform online searches about a person. These sites allow attackers to locate key individuals, identify home phone numbers, and even create maps to people's houses. Attackers can even see the surroundings of the company or the home they are targeting with great quality satellite pictures. Here are some of the sites:

- **Pipl:** <https://pipl.com/>
- **Spokeo:** <http://www.spokeo.com/>
- **BirthDatabase.com:** <http://www.birthdatabase.com/>
- **Whitepages:** <http://www.whitepages.com/>
- **People Search Now:** <http://www.peoplesearchnow.com/>
- **Zabasearch:** <http://www.zabasearch.com/>
- **Peoplefinders:** <http://www.peoplefinders.com/>
- **Justia email finder:** <http://virtualchase.justia.com/content/finding-email-addresses>

NOTE Keep in mind that the amount of information you gather will depend on what part of the world you are searching. Some countries have stronger laws regarding privacy than others. For example, the European Union has strict privacy laws. Citizens of the EU have the right to be forgotten.

What's interesting about these sites is that many sites promise everything from criminal background checks, to previous addresses, to marriage records, to family members. Figures 3-2 and 3-3 offer some examples of what these sites provide.

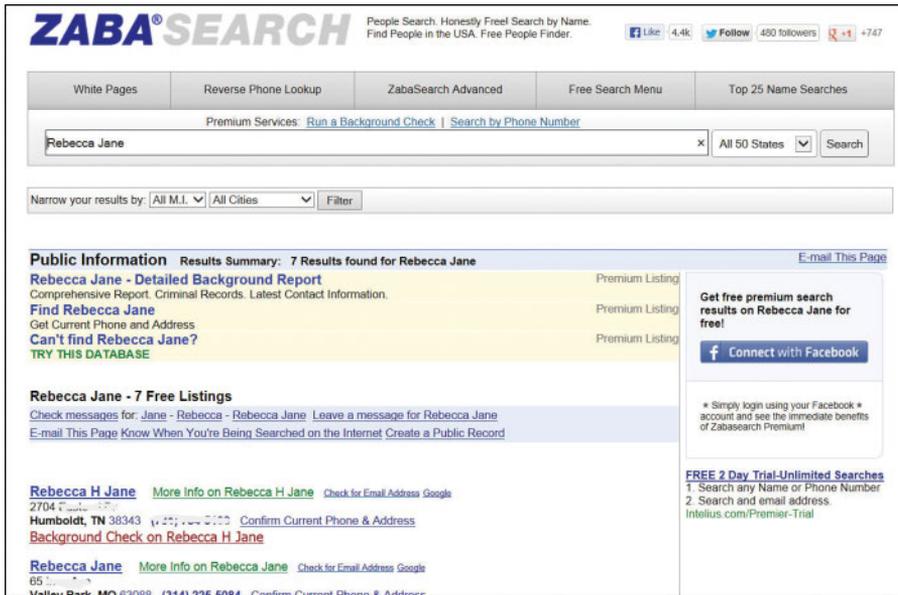


FIGURE 3-2 Zabasearch

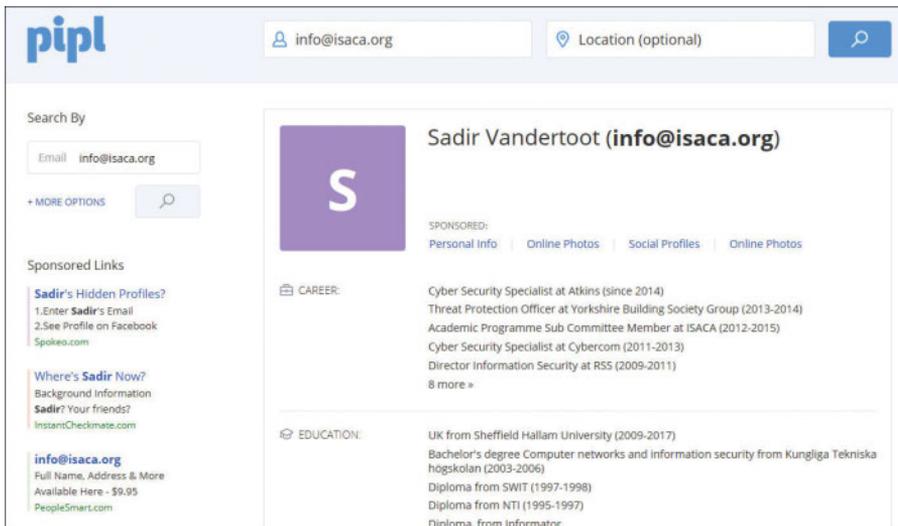


FIGURE 3-3 Pipl Email Lookup

NOTE According to the United States Federal Trade Commission, the American public has little rights over the control and dissemination of personal information except for medical records and some credit information. See <https://tcf.org/content/report/data-protection-federalism/>.

Social networks are another big target for attackers. Although social media has opened up great channels for communication and is very useful for marketers, it is fraught with potential security problems. Social networking sites are becoming one of the biggest threats to a user's security and will remain so for the foreseeable future. One reason is that users don't always think about security when using these sites. There is also the issue that these sites are designed to connect people. Security is not always the primary concern. Some sites that the ethical hacker may want to check include the following:

- Facebook
- Twitter
- LinkedIn
- Pinterest

TIP The three primary ways attackers use social networking include using social engineering to gather sensitive information, creating fake profiles, and using public information to gather information about a victim.

Although some organizations might be relatively secure, gaining the names, addresses, and locations of key employees can allow attackers to fly a drone over their homes, guess passwords, or even possibly backdoor the organization through an employee's unsecure credentials.

NOTE Ethical hackers can use tools like InSpy to perform enumeration on LinkedIn profiles and identify people based on company, job title, and email address.

TIP It's not just people that hackers are concerned with. Some attackers may scan the Web for competitive intelligence. It can be thought of as identifying, gathering, and analyzing information about a company's products or services.

The Dangers of Social Networks

Robin Sage is the name of a military exercise given to Army students before they receive their assignments to one of the Army's seven operational Special Forces groups. It is also the name that was recently given to a fictitious 25-year-old female pretending to be a cyberthreat analyst at the U.S. Navy's Network Warfare Command. The idea behind this ruse was to demonstrate the dangers of social networking. The results were startling.

Even though her fake Facebook profile was filled with inconsistencies, many people who should have known better tried to make contact, and passed potentially sensitive information. Her social network connections included senior military officers, a member from the Joint Chiefs of Staff, and someone from the National Reconnaissance Office (NRO); the NRO is responsible for launching and operating U.S. spy satellites.

The experiment was carried out by security consultant Thomas Ryan and revealed huge vulnerabilities in the use of social networking by people in the national security field. The results of this experiment were discussed by Mr. Ryan at the Black Hat security conference.

EDGAR Database

If the organization you are working for is publicly traded, you want to review the Security and Exchange Commission's EDGAR database. It's located at <https://www.sec.gov/edgar/searchedgar/companysearch.html>. A ton of information is available at this site. Hackers focus on the 10-Q and 10-K. These two documents contain yearly and quarterly reports.

NOTE The financial data found by using the EDGAR database can be used to determine whether a company should be targeted for attack or even ransomware.

Not only do these documents contain earnings and potential revenue, they also contain details about any acquisitions and mergers. Anytime there is a merger, or one firm acquires another, there is a rush to integrate the two networks. Having the networks integrated is more of an immediate concern than security. Therefore, you will be looking for entity names that are different from the parent organization. These findings might help you discover ways to jump from the subsidiary to the more secure parent company. You should record this information and have it

ready when you start to research the Internet Assigned Numbers Authority (IANA) and American Registry for Internet Numbers (ARIN) databases. Here are some other sites you can use to gather financial information about an organization:

- **Marketwatch:** <http://www.marketwatch.com>
- **Experian:** <http://www.experian.com>
- **Wall Street Consensus Monitor:** <http://www.wallstreetconsensusmonitor.com/>
- **Euromonitor:** <http://www.euromonitor.com>

Google Hacking

Most of us use Google or another search engine to locate information. What you might not know is that search engines, such as Google, can perform much more powerful searches than most people ever dream of. Not only can Google translate documents, perform news searches, and do image searches, but it can also be used by hackers and attackers to do something that has been termed *Google hacking*.

By using basic search techniques combined with advanced operators, Google can become a powerful vulnerability search tool. Table 3-2 describes some advanced operators.

Table 3-2 Google Search Terms

Operator	Description
Filetype	Directs Google to search only within the text of a particular type of file. Example: filetype:xls
Inurl	Directs Google to search only within the specified URL of a document. Example: inurl:search-text
Link	Directs Google to search within hyperlinks for a specific term. Example: link:www.domain.com
Intitle	Directs Google to search for a term within the title of a document. Example: intitle: "Index of.etc"

NOTE The CEH exam may ask you about specific Google search term strings.

By using the advanced operators shown in Table 3-2 in combination with key terms, Google can be used to uncover many pieces of sensitive information that shouldn't be revealed. A term even exists for the people who blindly post this information on the Internet; they are called Google dorks. To see how this works, enter the following phrase into Google:

```
intext:JSESSIONID OR intext:PHPSESSID inurl:access.log ext:log
```

This query searches in a URL for the session IDs that could be used to potentially impersonate users. The search found more than 100 sites that store sensitive session IDs in logs that were publicly accessible. If these IDs have not timed out, they could be used to gain access to restricted resources. You can use advanced operators to search for many types of data. Figure 3-4 shows a search where Social Security numbers (SSNs) were queried. Although this type of information should not be listed on the Web, it might have been placed there inadvertently or by someone who did not understand the security implications.

The screenshot shows a list of search results for the query "Social Security Number". Each result includes a document title, a URL, and a snippet of text containing a Social Security Number. The results are as follows:

- Curriculum Vitae r 2007 ewolc...** - Home
www.cer.sen.vi.verde.itale.solutions.com/.../primary.doc
CURRICULUM VITAE. NAME: NELS OWEN FRIEDLSEN. **SOCIAL SECURITY NUMBER: 307-5-7-018.** HOME ADDRESS: 8775 E. Lake St. Rockwell, IN 47872 ...
- CURRICULUM VITAE - OrthoDoc@aaos.org - American Acade...**
orthodoc.aaos.org/.../cvry.html.doc
C. U.S. **Social Security Number: 366-6-1-114.** II. Education. Year Degree Institution. 1976 B.S. University of Michigan. Ann Arbor, Michigan. 1980 M.D. University ...
- What your comprehensive resume - Maryland National Guard**
www.md.ngb.army.mil/.../200Resume/200Guidance%20and%20FAQs.htm
Additional pages of your **resume** will include your Name, **Social Security number**, and the Vacancy number on each page. John P. Smith, iv. 123 1/2 Big Street ...
- Curriculum Vitae.doc - Particle Theory**
theory.uhohio.edu/~smale/ferry_page/curriculum%20vitae.doc
Name in Full: Maria Georgios Spiropulu. U.S. **Social Security Number: 029-76-2871** ... spiropulu@physics.umd.edu. - 4 - Curriculum Vitae: Maria G. Spiropulu ...
- How to Apply to the NIH - Tutorial Script - Jobs at NIH - National ...**
www.jobs.nih.gov/jobssearch/howtoapply/wordscript.doc
There is, however, something that should be included on your federal **resume**, and that is your **Social Security Number**. Your **social security number** must go ...
- Federalizing Your Resume - Jobs at NIH - National Institutes of...**
www.jobs.nih.gov/jobssearch/federalresume/wordscript.doc
Another tip is to include your **Social Security Number** on your **résumé**.
- CURRICULUM VITAE 5/98 - Tropical and Molecular Parasitology**
http://netlib.edu/archives/TMPS/orig/2006/05-06Beverly_CV%20.doc
CURRICULUM VITAE 12/2005. 1. Personal data. Stephen M. Fryxell y ... **Social Security Number: 314-747-2820** ... 2003/05/01/ beverley@hopkins.edu. 2.

FIGURE 3-4 Google Hacking Social Security Numbers

Finally, don't forget that finding a vulnerability using Google is not unethical, but using that vulnerability can be unethical unless you have written permission from the domain owner. For example, here is a link to the Google hack for Shellshock (a Bash vulnerability introduced later in the chapter): <https://www.exploit-db.com/exploits/34895/>. Notice how it took only a few minutes for an attacker to gather this type of information. Security professionals should always be concerned about what kind of information is posted on the Web and who can access it.

Now that we have discussed some basic Google search techniques, let's look at advanced Google hacking. If you have never visited the Google Hacking Database (GHDB) repositories, I suggest that you visit <http://www.hackersforcharity.org/ghdb/> and <https://www.exploit-db.com/google-hacking-database/>. These sites have the following search categories:

- Footholds
- Files containing usernames
- Sensitive directories
- Web server detection
- Vulnerable files
- Vulnerable servers
- Error messages
- Files containing juicy info
- Files containing passwords
- Sensitive online shopping info
- Network or vulnerability data
- Pages containing login portals
- Various online devices
- Advisories and vulnerabilities

Johnny Long, Bill Gardner, and Justin Brown have written an excellent book on the subject, *Google Hacking for Penetration Testers*, Third Edition. Using these techniques, you can find all sorts of information on services, files, and even people. Figure 3-4 shows an example of some of the more unbelievable things found by Google hacking.

A tool such as the GHDB has made using Google easier, but it's not your only option. Maltego, FOCA, Recon Dog, and Shodan are others worth discussion. Maltego is an open source intelligence and forensics application. It is a tool-based approach to mining and gathering Internet data that can be compiled in an easy-to-understand format. Maltego offers plenty of data on websites and their services. FOCA is another example of an open source information-gathering tool. Figure 3-5 shows an example of FOCA being used to extract metadata from documents to determine such details as software version. Similar to FOCA is Recon Dog, which is another example of an all-in-one information-gathering tool.

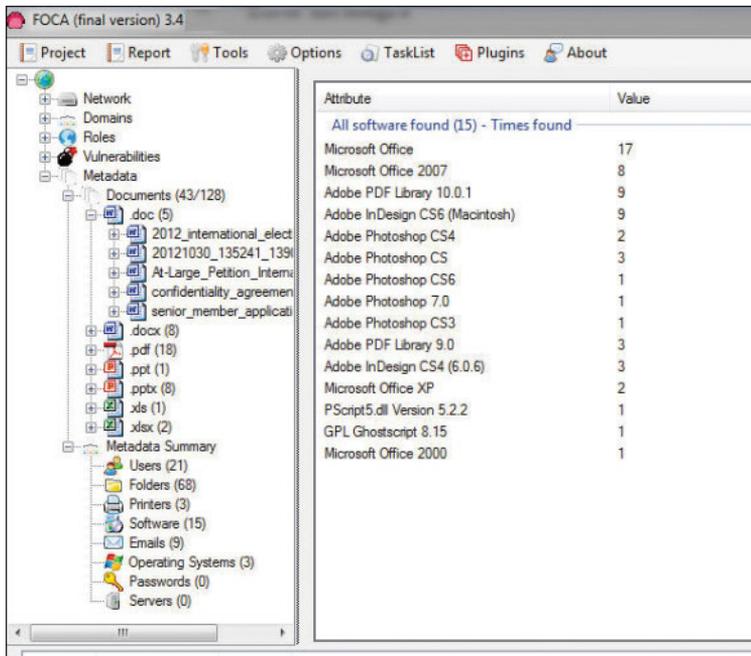


FIGURE 3-5 FOCA Extracting Metadata

Shodan offers the ability to search for the servers, webcams, printers, routers, and even SCADA devices connected to the Internet. SCADA devices are industrial controls with embedded computers that can be connected to the Internet.

Tools such as Shodan and Censys can be used to find network-connected devices, such as routers, servers, IoT devices, and even printers. Using a variety of filters, these search engines allow you to query hosts and networks for specific information.

In the Field: The Shodan Computer Search Engine

The Shodan Computer Search Engine is a powerful database of prescanned networked devices connected to the Internet. It consists of banners collected from port scans of public IP addresses, with fingerprints of services like Telnet, FTP, HTTP, and other applications.

Shodan creates risk by providing both attackers and defenders a prescanned inventory of devices connected to public IP addresses on the Internet. For example, when a new vulnerability is discovered and published, an attacker can quickly and easily search Shodan for vulnerable versions and then launch an attack. Attackers can also search the Shodan database for devices with poor configurations or other weaknesses, all without actively scanning.

Using Shodan search filters, one can really narrow down search results, by country code or CIDR netblock, for example. Shodan application programming interfaces (APIs) and some basic scripting can enable many search queries and subsequent actions (for example, a weekly query of newly discovered IPs scanned by Shodan on your CIDR netblock that runs automatically and is emailed to the security team).

Remember that public IP addresses are constantly probed and scanned already; by using Shodan, you are not scanning, because Shodan has already scanned these IPs. Shodan is a tool, and it can be used for good or evil. To mitigate risk, you can take tangible steps like registering for a free Shodan account, searching for your organization's public IPs, and informing the right network and security people of the risks of your organization's Shodan exposure. You can learn more at <https://www.shodanhq.com>.

This In the Field note was contributed by Shawn Merdinger, security researcher and founder of the MedSec LinkedIn group.

You might be wondering who is using all these web search tools. It's not just hackers. In 2013, documents made public by the National Security Agency (NSA) following a Freedom of Information Act (FOIA) request uncovered a PDF book titled *Untangling the Web: A Guide to Internet Research*. Although it is somewhat dated, its 643 pages contain many pages dedicated to showing federal agents how to "Google hack" and search directly for documents published online, such as Excel spreadsheets, Word documents, and PDFs. Although much of this document deals with manual ways to footprint, more modern tools like OSRFramework make the job much easier. OSRFramework is just a set of libraries that can be used to search for usernames, DNS data, phone numbers, and so on.

Usenet

Usenet is a user's network, which is nothing more than a collection of the thousands of discussion groups that reside on the Internet. Each discussion group contains

information and messages centered on a specific topic. Messages are posted and responded to by readers either as public or private emails. Even without direct access to Usenet, a convenient way to browse the content is by using Google Groups. Google Groups allows any Internet user a way to post and read Usenet messages. During a penetration test, you will want to review Google Groups for postings from the target company.

One way to search is to use individuals' names you might have uncovered; another is to do a simple search of the company. Searching for @company.com will work. Many times, this will reveal useful information. One company that I performed some work for had listings from the network administrator. He had been asked to set up a new router and was having trouble getting it configured properly. The administrator had not only asked the group for help but had also posted the router configuration to see whether someone could help figure out what was wrong. The problem was that the configuration file had not been sanitized and not only contained IP addresses but also the following information:

```
enable secret 5 $1$2RKf$OMOAcvzpb7j9uhfw6C5Uj1
enable password 7 583132656321654949
```

For those of you who might not be Cisco gurus, those are encrypted passwords. The first one is MD5 and the second is a type 7. According to Cisco, type 7 passwords were not designed to prevent a determined or targeted attack. Type 7 password encryption is only a basic protection mechanism based on a reversible algorithm. Because of the weak encryption algorithm, the Cisco position is that customers should treat configuration files as sensitive information. The problem is that attackers can potentially obtain these configuration files using a number of means, such as Usenet postings, help forums, or even a TFTP server. Others of you who say that “it’s only router passwords” might be right, but let’s hope that the administrator doesn’t reuse passwords (as many people do). As you can see, you can gain additional information about an organization and its technical strengths just by uncovering a few Usenet posts. With possession of the password, the attacker can then use any number of tools to quickly decode the obscured password. Well-known tools that can decode Cisco 7 passwords include Cain and Abel and the Cisco Password decoder. A quick search of the Web returns dozens of hits on such a query. This brings us to the inevitable question of how to fix this problem. Actually, it is not that hard to do. First, you should not post router or firewall configurations, and the enable password command should no longer be used. Use the **enable secret** command instead; it uses the MD5 algorithm, which is much more secure.

Registrar Query

Not long ago, searching for domain name information was much easier. There were only a few places to obtain domain names, and the activities of spammers

and hackers had yet to cause the Internet Assigned Numbers Authority (IANA) to restrict the release of this information. Today, the Internet Corporation for Assigned Names and Numbers (ICANN) is the primary body charged with management of IP address space allocation, protocol parameter assignment, and domain name system management. Its role is that of overall management, as domain name registration is handled by a number of competing firms that offer various value-added services. These include firms such as Network Solutions (<https://networksolutions.com>), Register.com (<https://www.register.com>), GoDaddy (<https://godaddy.com>), and Tucows (<http://www.tucows.com>). There is also a series of Regional Internet Registries (RIRs) that manage, distribute, and register public IP addresses within their respective regions. There are five RIRs. These are shown in Table 3-3.

Table 3-3 RIRs and Their Area of Control

RIR	Region of Control
ARIN	North and South America and sub-Saharan Africa
APNIC	Asia and Pacific
RIPE	Europe, Middle East, and parts of Africa
LACNIC	Latin America and the Caribbean
AfriNIC	Planned RIR to support Africa

TIP Know the RIR for each region of the world, because this could be something you are tested on.

The primary tool to navigate these databases is Whois. Whois is a utility that interrogates the Internet domain name administration system and returns the domain ownership, address, location, phone number, and other details about a specified domain name. Whois is the primary tool used to query Domain Name System (DNS). If you're performing this information gathering from a Linux computer, the good news is Whois is built in. From the Linux prompt, users can type **whois domainname.com** or **whois?** to get a list of various options. Windows users are not as fortunate because Windows does not have a built-in Whois client. Windows users have to use a third-party tool or website to obtain Whois information.

One tool that a Windows user can use to perform Whois lookups is SmartWhois. It can be downloaded from <http://www.tamos.com/products/smartwhois/>. SmartWhois is a useful network information utility that allows you to look up all the available information about an IP address, hostname, or domain, including country, state or province, city, name of the network provider, administrator, and technical

support contact information. You can also use a variety of other tools to obtain Whois information, including the following:

- **BetterWhois:** <http://www.betterwhois.com>
- **All NETTOOLS:** www.all-nettools.com
- **DNSstuff:** www.dnsstuff.com
- **Whois Proxy:** <http://geektools.com/whois.php>
- **Whois Lookup:** <http://www.pentest-tools.com>
- **3d Traceroute:** <http://www.d3tr.de/>
- **Path Analyzer Pro:** <https://www.pathanalyzer.com/>
- **LoriotPro:** <http://www.loriotpro.com/>

Regardless of the tool, the goal is to obtain registrar information. As an example, the following listing shows the results after www.domaintools.com/ is queried for information about www.pearson.com:

Registrant:

```
Pearson PLC
Clive Carmock
80 Strand London
London, UK WC2R 0RL
GB
Email:
```

```
Registrar Name....: CORPORATE DOMAINS, INC.
Registrar Whois...: whois.corporatedomains.com
Registrar Homepage: www.cscprotectsbrands.com
```

Domain Name: pearson.com

```
Created on.....: Mon, Nov 25, 1996
Expires on.....: Thu, Nov 23, 2023
Record last updated on..: Thu, Feb 02, 2017
```

Administrative Contact :

```
Pearson PLC
Clive Carmock
80 Strand London
```

```
., . WC2R ORL
GB
Phone: 044-2070-105580
Email:
```

```
Technical Contact:
Pearson PLC
Clive Carmock
80 Strand London
., . WC2R ORL
GB
Phone: 044-2070-105580
Email:
```

```
DNS Servers:
usrxdns1.pearsontc.com
oldtxdns2.pearsontc.com
ns.pearson.com
ns2.pearson.com
```

This information provides a contact, address, phone number, and DNS servers. A hacker skilled in the art of social engineering might use this information to call the organization and pretend to be a valid contact.

TIP A domain proxy is one way that organizations can protect their identity while still complying with laws that require domain ownership to be public information. Domain proxies work by applying anonymous contact information as well as an anonymous email address. This information is displayed when someone performs a domain Whois. The proxy then forwards any emails or contact information that might come to those addresses on to you.

DNS Enumeration

If all the previous information has been acquired, the DNS might be targeted for zone transfers. A zone transfer is the mechanism used by DNS servers to update each other by transferring the contents of their database. DNS is structured as a hierarchy so that when you request DNS information, your request is passed up the hierarchy until a DNS server is found that can resolve the domain name request. You can get a better idea of how DNS is structured by examining Figure 3-6, which shows a total of 13 DNS root servers.

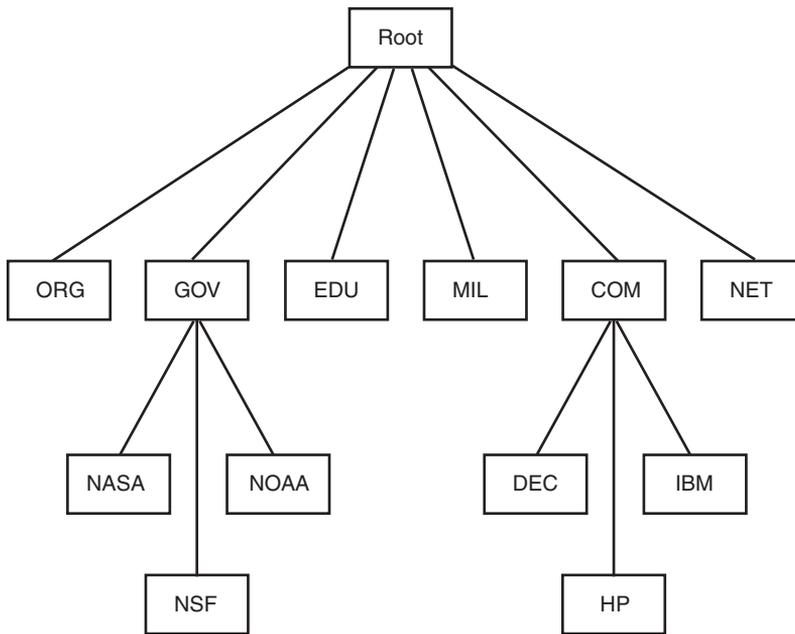


FIGURE 3-6 DNS Structure

What's left at this step is to try to gather additional information from the organization's DNS servers. The primary tool to query DNS servers is Nslookup. Nslookup provides machine name and address information. Both Linux and Windows have Nslookup clients. Nslookup is used by typing **nslookup** from the command line followed by an IP address or a machine name. Doing so causes Nslookup to return the name, all known IP addresses, and all known CNAMEs for the identified machine. Nslookup queries DNS servers for machine name and address information. Using Nslookup is rather straightforward. Let's look at an example in which Nslookup is used to find out the IP addresses of Google's web servers. If you enter **nslookup www.google.com**, the following response is obtained:

```

C:\>nslookup www.google.com
Server:dnsr1.sbcglobal.net
Address:68.94.156.1
Non-authoritative answer:
Name:www.google.com
Addresses:64.233.187.99, 64.233.187.104
Aliases:www.google.com
  
```

The first two lines of output say which DNS servers are being queried. In this case, it's dnsr1.sbcglobal.net in Texas. The nonauthoritative answer lists two IP addresses for the Google web servers. Responses from nonauthoritative servers do not contain copies of any domains. They have a cache file that is constructed from all the

DNS lookups it has performed in the past for which it has gotten an authoritative response.

Nslookup can also be used in an interactive mode by just typing **nslookup** at the command prompt in Windows or the Bash shell in Linux. In interactive mode, the user will be given a prompt of **>**; at which point the user can enter a variety of options, including attempts to perform a zone transfer. Table 3-4 shows some common DNS resource record names and types.

Table 3-4 IPv4 DNS Records and Types

Record Name	Record Type	Purpose
Host	A	Maps a domain name to an IPv4 address
Host	AAAA	Maps a domain name to an IPv6 address
Pointer	PTR	Maps an IP address to a domain name
Name Server	NS	Specifies the servers that provide DNS services
Start of Authority	SOA	Configures settings for zone transfers and record caching
Service Locator	SRV	Used to locate services in the network
Mail	MX	Used to identify SMTP servers

TIP Know the various record names and types for DNS.

TIP The SOA contains the timeout value, which can be used by a hacker to tell how long any DNS poisoning would last. The Time To Live (TTL) value is the last value within the SOA.

DNS normally moves information from one DNS server to another through the DNS zone transfer process. If a domain contains more than one name server, only one of these servers will be the primary. Any other servers in the domain will be secondary servers. Zone transfers are much like the DHCP process in that each is a four-step process. DNS zone transfers function as follows:

1. The secondary name server starts the process by requesting the SOA record from the primary name server.
2. The primary then checks the list of authorized servers, and if the secondary server's name is on that list, the SOA record is sent.
3. The secondary must then check the SOA record to see whether there is a match against the SOA it already maintains. If the SOA is a match, the process

stops here; however, if the SOA has a serial number that is higher, the secondary will need an update. The serial number indicates if changes were made since the last time the secondary server synchronized with the primary server. If an update is required, the secondary name server will send an All Zone Transfer (AXFR) request to the primary server.

4. Upon receipt of the AXFR, the primary server sends the entire zone file to the secondary name server.

NOTE In September 2012, Bash, which is widely used in Linux/UNIX systems, was discovered to be vulnerable to arbitrary command execution. This family of vulnerabilities would come to be known as Shellshock and was exploited millions of times in the days following disclosure of the vulnerabilities.

A zone transfer is unlike a normal lookup in that the user is attempting to retrieve a copy of the entire zone file for a domain from a DNS server. This can provide a hacker or pen tester with a wealth of information. This is not something that the target organization should be allowing. Unlike lookups that primarily occur on UDP 53, unless the response is greater than 512 bytes, zone transfers use TCP 53. To attempt a zone transfer, you must be connected to a DNS server that is the authoritative server for that zone. An example is shown here for your convenience:

```
Registrant:
    Technology Centre
    Domain Administrator
    200 Old Tappan Rd.
    Old Tappan, NJ 07675 USA
    Email: billing@superlibrary.com
Phone: 001-201-7846187
Registrar Name....: REGISTER.COM, INC.
Registrar Whois...: whois.register.com
Registrar Homepage: www.register.com
DNS Servers:
    usrxdns1.pearsontc.com
    oldtxdns2.pearsontc.com
```

Review the last two entries. Both usrxdns1.pearsontc.com and oldtxdns2.pearsontc.com are the DNS authoritative servers listed. These are the addresses that an attacker will target to attempt a zone transfer. The steps to try to force a zone transfer are shown here:

1. **nslookup:** Enter **nslookup** from the command line.
2. **server <ipaddress >:** Enter the IP address of the authoritative server for that zone.

3. **set type = any:** Tells Nslookup to query for any record.
4. **ls -d <domain.com >:** Domain.com is the name of the targeted domain of the final step that performs the zone transfer.

One of two things will happen at this point. You will receive an error message indicating that the transfer was unsuccessful, or you will be returned a wealth of information, as shown in the following:

```
C:\Windows\system32>nslookup
Default Server:dnsr1.sbcglobal.net
Address:128.112.3.12
server 172.6.1.114
set type=any
ls -d example.com
example.com. SOA      hostmaster.sbc.net (950849 21600 3600 1728000
 3600)
example.com. NS      auth100.ns.sbc.net
example.com. NS      auth110.ns.sbc.net
example.com.  A       10.14.229.23
example.com.  MX       10    dallassmtpr1.example.com
example.com.  MX       20    dallassmtpr2.example.com
example.com.  MX       30    lasmtpr1.example.com
lasmtpr1     A       192.172.243.240
dallassmtpr1 A       192.172.163.9
dallaslink2  A       192.172.161.4
spamassassin A       192.172.170.49
dallassmtpr2 A       192.172.163.7
dallasextra  A       192.172.170.17
dallasgate  A       192.172.163.22
lalink       A       172.16.208.249
dallassmtpl  A       192.172.170.49
nygate       A       192.172.3.250
www          A       10.49.229.203
dallassmtpr  MX      10    dallassmtpr1.example.com
dallassmtpr  MX      20    dallassmtpr2.example.com
dallassmtpr  MX      30    lasmtpr1.example.com
```

Dig is another tool that you can use to provide this type of information. It's built in to most all Linux distributions and can be run from Bash or run from the command prompt when installed into Windows. Dig is a powerful tool that can be used to

investigate the DNS system. There is also a range of tools that can be used to interrogate DNS servers, including the following:

- **WhereISIP:** <http://www.wherisip.net/>
- **DNSMap:** <http://code.google.com/archive/p/dnsmap/>

Internal DNS information should not be made available to just anyone. Hackers can use this to find out what other servers are running on the network, and it can help them map the network and formulate what types of attacks to launch. Notice the first line in the previous printout that has example.com listed. Observe the final value of 3600 on that line. That is the TTL value discussed previously and would inform a hacker as to how long DNS poisoning would last. 3,600 seconds is 60 minutes. Zone transfers are intended for use by secondary DNS servers to synchronize with their primary DNS server. You should make sure that only specific IP addresses are allowed to request zone transfers. Most operating systems restrict this by default. All DNS servers should be tested. It is often the case that the primary has tight security but the secondaries may allow zone transfers if misconfigured.

TIP The CEH exam expects you to understand the Nslookup and Dig functions. Be sure that you know how to get into interactive mode with Nslookup and how to extract specific information. You may be asked to verify a specific Nslookup command.

Determining the Network Range

Now that the pen test team has been able to locate names, phone numbers, addresses, some server names, and IP addresses, it's important to find out what IP addresses are available for scanning and further enumeration. If you take the IP address of a web server discovered earlier and enter it into the Whois lookup at <https://www.arin.net>, you can determine the network's range. For example, 192.17.170.17 was entered into the ARIN Whois, and the following information was received:

```

OrgName:                target network
OrgID:                  Target-2
Address:                1313 Mockingbird Road
City:                  Anytown
StateProv:             Tx
PostalCode:            72341
Country:               US
ReferralServer:        rwhois://rwhois.exodus.net:4321/
NetRange:              192.17.12.0 - 192.17.12.255

```

```

CIDR:                192.17.0.0/24
NetName:             SAVVIS
NetHandle            NET-192-17-12-0-1
Parent:              NET-192-0-0-0-0

```

This means that the target network has 254 total addresses. The attacker can now focus his efforts on the range from 192.17.12.1 to 192.17.12.254/24. If these results don't prove satisfactory, the attacker can use traceroute for additional mapping.

Subnetting's Role in Mapping Networks

Some of the items you may see on the exam but are not included in any of the official courseware include subnetting. Subnetting also allows the creation of many logical networks that exist within a single Class A, B, or C network. Subnetting is important in that it helps pen testers identify what systems are part of which specific network.

To subnet a network, you must extend the natural mask with some of the bits from the host ID portion of the address. For example, if you had a Class C network of 192.168.5.0, which has a natural mask of 255.255.255.0, you can create subnets in this manner:

```

192.168.5.0 -11001100.10101000.00000101.00000000
255.255.255.224 - 11111111.11111111.11111111.11100000
-----|subnet|-----

```

By extending the mask from 255.255.255.0 to 255.255.255.224, you have taken 3 bits from the original host portion of the address and used them to make subnets. By borrowing 3 bits, it is possible to create eight subnets. The remaining 5 bits can provide for up to 32 host addresses, 30 of which can actually be assigned to a device because host addresses with all zeros and all ones are not assigned to specific devices. Here is a breakdown of the subnets and their address ranges:

Subnet	Host Range
192.168.5.0 255.255.255.224	host address range 1 to 30
192.168.5.32 255.255.255.224	host address range 33 to 62
192.168.5.64 255.255.255.224	host address range 65 to 94
192.168.5.96 255.255.255.224	host address range 97 to 126
192.168.5.128 255.255.255.224	host address range 129 to 158
192.168.5.160 255.255.255.224	host address range 161 to 190
192.168.5.192 255.255.255.224	host address range 193 to 222
192.168.5.224 255.255.255.224	host address range 225 to 254

The more host bits you use for a subnet mask, the more subnets you have available. However, the more subnets that are available, the fewer host addresses that are available per subnet.

Traceroute

It's advisable to check out more than one version of traceroute if you don't get the required results. Some techniques can also be used to try to slip traceroute past a firewall or filtering device. When UDP and ICMP are not allowed on the remote gateway, you can use TCPTraceroute. Another unique technique was developed by Michael Schiffman, who created a patch called traceroute.diff that allows you to specify the port that traceroute will use. With this handy tool, you could easily direct traceroute to use UDP port 53. Because that port is used for DNS queries, there's a good chance that it could be used to slip past the firewall. If you're looking for a graphical user interface (GUI) program to perform traceroute with, several are available, as described here:

- LoriotPro:** LoriotPro (see Figure 3-7) is a professional and scalable SNMP manager and network monitoring solution that enables availability and performance control of your networks, systems, and smart infrastructures. The graphical display shows you the route between you and the remote site, including all intermediate nodes and their registrant information.

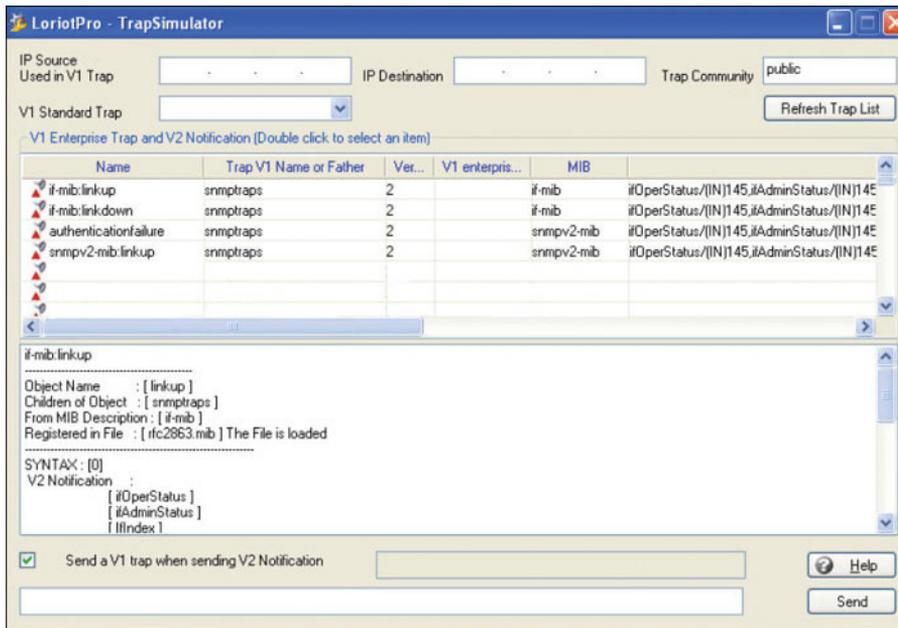


FIGURE 3-7 LoriotPro

- **Trout:** Trout is another visual traceroute and Whois program. What's great about this program is its speed. Unlike traditional traceroute programs, Trout performs parallel pinging. By sending packets with more than one TTL at a time, it can quickly determine the path to a targeted device.
- **VisualRoute:** VisualRoute is another graphical traceroute for Windows. VisualRoute not only shows a graphical world map that displays the path packets are taking, but also lists information for each hop, including IP address, node name, and geographic location. This tool is commercial and must be purchased.

TIP Traceroute and ping are useful tools for identifying active systems, mapping their location, and learning more about their location. Just keep in mind that these tools are limited by what they can see; if these services are blocked by a firewall, you may get no useful data returned.

Identifying Active Machines

Attackers will want to know whether machines are alive before they attempt to attack. One of the most basic methods of identifying active machines is to perform a ping sweep. Just because ping can be blocked does not mean it is. Although ping has been restricted by many organizations, you should still check to see if it is available. Ping uses ICMP and works by sending an echo request to a system and waiting for the target to send an echo reply back. If the target device is unreachable, a request timeout is returned. Ping is a useful tool to identify active machines and to measure the speed at which packets are moved from one host to another or to get details like the TTL. Figure 3-8 shows a ping capture from a Windows computer. If you take a moment to examine the ASCII decode in the bottom-left corner, you will notice that the data in the ping packet is composed of the alphabet, which is unlike a Linux ping, which would contain numeric values. That's because the RFC that governs ping doesn't specify what's carried in the packet as payload. Vendors fill in this padding as they see fit. Unfortunately, this can also serve hackers as a covert channel. Hackers can use a variety of programs to place their own information in place of the normal padding. Tools like Loki and icmpsend are designed for just this purpose. Then what appear to be normal pings are actually a series of messages entering and leaving the network.

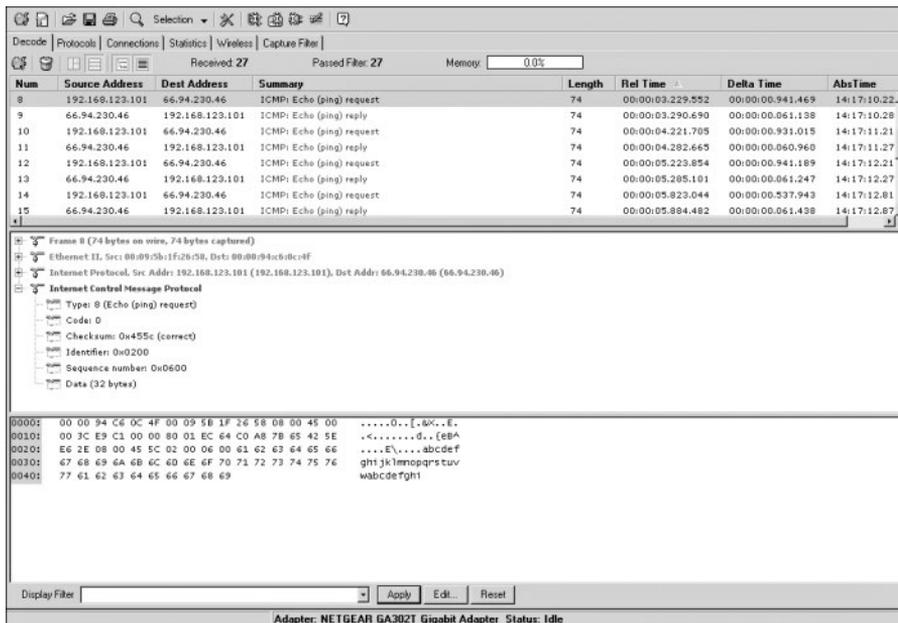


FIGURE 3-8 Ping Capture

Ping does have a couple of drawbacks: First, only one system at a time is pinged, and second, not all networks allow ping. To ping a large number of hosts, a ping sweep is usually performed. Programs that perform ping sweeps usually sweep through a range of devices to determine which ones are active. Programs that will perform ping sweeps include the following:

- **Angry IP Scanner:** <http://angryip.org/>
- **Hping:** <http://www.hping.org/>
- **WS_Ping ProPack:** <https://ws-ping-propack.en.softonic.com/>
- **SuperScan:** <http://www.mcafee.com/us/downloads/free-tools/superscan.aspx>
- **Nmap:** <https://nmap.org/>

TIP Know the positives and negatives of ping before taking the CEH exam.

Finding Open Ports and Access Points

Port scanning is the process of connecting to TCP and UDP ports for the purpose of finding what services and applications are running on the target device. After

discovering running applications, open ports, and services, the hacker can then determine the best way to attack the system.

As discussed in Chapter 2, “The Technical Foundations of Hacking,” there are a total of 65,535 TCP and UDP ports. These port numbers are used to identify a specific process that a message is coming from or going to. Table 3-5 lists some common port numbers.

Table 3-5 Common Ports and Protocols

Port	Protocol	Service/Transport
20/21	FTP	TCP
22	SSH	TCP
23	Telnet	TCP
25	SMTP	TCP
53	DNS	TCP/UDP
69	TFTP	UDP
80	HTTP	TCP
110	POP3	TCP
135	RPC	TCP
161/162	SNMP	UDP
1433/1434	MSSQL	TCP

TIP The exam may ask you about common or not so common ports, such as 514 (syslog) or even 179 (Internet Printing Protocol). If you see these on the test questions, the best approach is to first eliminate known ports and reduce down to the best answer.

As you have probably noticed, some of these applications run on TCP, others on UDP. Although it is certainly possible to scan for all 65,535 TCP and 65,535 UDP ports, many hackers will not. They will concentrate on the first 1,024 ports. These well-known ports are where we find most of the commonly used applications. You can find a list of well-known ports at <http://www.iana.org/assignments/port-numbers>. This is not to say that high-order ports should be totally ignored, because hackers might break into a system and open a high-order port, such as 31337, to use as a backdoor. So, is one protocol easier to scan for than the other? The answer to that question is yes. TCP offers more opportunity for the hacker to manipulate than UDP. Let's take a look at why.

TCP offers robust communication and is considered a connection protocol. TCP establishes a connection by using what is called a three-way handshake. Those three steps proceed as follows:

1. The client sends the server a TCP packet with the sequence number flag (SYN flag) set and an initial sequence number (ISN).
2. The server replies by sending a packet with the SYN/ACK flag set to the client. The synchronize sequence number flag informs the client that it would like to communicate with it, and the acknowledgment flag informs the client that it received its initial packet. The acknowledgment number will be one digit higher than the client's ISN. The server generates an ISN, as well, to keep track of every byte sent to the client.
3. When the client receives the server's packet, it creates an ACK packet to acknowledge that the data has been received from the server. At this point, communication can begin.

The TCP header contains a 1-byte field for the flags. Table 3-6 describes the six most common flags.

**Key
Topic**

Table 3-6 TCP Flag Types

Flag	Description
SYN	Synchronize and initial sequence number (ISN)
ACK	Acknowledgment of packets received
FIN	Final data flag used during the four-step shutdown of a session
RST	Reset bit used to close an abnormal connection
PSH	Push data bit used to signal that data in the packet should be pushed to the beginning of the queue; usually indicates an urgent message
URG	Urgent data bit used to signify that urgent control characters are present in this packet that should have priority

TIP One easy way to remember the six most commonly used flags is as follows:
Unruly Attackers Pester Real Security Folks.

At the conclusion of communication, TCP terminates the session by using a four-step shutdown:

1. The client sends the server a packet with the FIN/ACK flags set.
2. The server sends a packet ACK flag set to acknowledge the client's packet.

3. The server then generates another packet with the FIN/ACK flags set to inform the client that it also is ready to conclude the session.
4. The client sends the server a packet with the ACK flag set to conclude the session.

TIP TCP flags are considered testable topics. You should understand their use and purpose.

The TCP system of communication makes for robust communication but also allows a hacker many ways to craft packets in an attempt to coax a server to respond or to try and avoid detection of an intrusion detection system (IDS). Many of these methods are built in to Nmap and other port-scanning tools. Before we take a look at those tools, though, some of the more popular port-scanning techniques are listed here:

- **TCP Full Connect scan:** This type of scan is the most reliable, although it is also the most detectable. It is easily logged and detected because a full connection is established. Open ports reply with a SYN/ACK, and closed ports respond with an RST/ACK.
- **TCP SYN scan:** This type of scan is known as *half open* because a full TCP three-way connection is not established. This type of scan was originally developed to be stealthy and evade IDSs, although most now detect it. Open ports reply with a SYN/ACK, and closed ports respond with an RST/ACK.
- **TCP FIN scan:** Forget trying to set up a connection; this technique jumps straight to the shutdown. This type of scan sends a FIN packet to the target port. An open port should return no response. Closed ports should send back an RST/ACK. This technique is usually effective only on UNIX devices or those compliant to RFC 793.
- **TCP NULL scan:** Sure, there should be some type of flag in the packet, but a NULL scan sends a packet with no flags set. If the OS has implemented TCP per RFC 793, open ports send no reply, whereas closed ports will return an RST.
- **TCP ACK scan:** This scan attempts to determine access control list (ACL) rule sets or identify if a firewall or simply stateless inspection is being used. A stateful firewall should return no response. If an ICMP destination is unreachable, and a communication administratively prohibited message is returned, the port is considered to be filtered. If an RST is returned, no firewall is present.

- **TCP XMAS scan:** Sorry, there are no Christmas presents here, just a port scan that has toggled on the FIN, URG, and PSH flags. Open ports should provide no response. Closed ports should return an RST. Systems must be designed per RFC 793 for this scan to work, as is common for Linux. It does not work against Windows computers.

TIP You should know common scan types, such as full and stealth, to successfully pass the exam. It's suggested that you download the Nmap tool and play with it to fully understand the options. The exam may test you over any type of Nmap scan.

Certain operating systems have taken some liberties when applying the TCP/IP RFCs and do things their own way. Because of this, not all scan types work against all systems. Results will vary, but Full Connect scans and SYN scans should work against all systems.

These are not the only types of possible scans; there are other scan types. Some scanning techniques can be used to obscure attackers and help hide their identity. One such technique is the idle or zombie scan. Before we go through an example of idle scanning, let's look at some basics on how TCP/IP connections operate. IP makes use of an identification number known as an IPID. This counter helps in the reassembly of fragmented traffic. TCP offers reliable service; it must perform a handshake before communication can begin. The initializing party of the handshake sends a SYN packet to which the receiving party returns a SYN/ACK packet if the port is open. For closed ports, the receiving party returns an RST. The RST acts as a notice that something is wrong, and further attempts to communicate should be discontinued. RSTs are not replied to; if they were replied to, we might have a situation in which two systems flood each other with a stream of RSTs. This means that unsolicited RSTs are ignored. By combining these characteristics with IPID behavior, a successful idle scan is possible.

An open port idle scan works as follows: An attacker sends an IDIP probe to the idle host to solicit a response. Suppose, for example, that the response produces an IPID of 12345. Next, the attacker sends a spoofed packet to the victim. This SYN packet is sent to the victim but is addressed from the idle host. An open port on the victim's system will then generate a SYN ACK. Because the idle host was not the source of the initial SYN packet and did not at any time want to initiate communication, it responds by sending an RST to terminate communications. This increments the IPID by one to 12346. Finally, the attacker again queries the idle host and is issued an IPID response of 12347. Because the IPID count has now been incremented by

two from the initial number of 12345, the attacker can deduce that the scanned port on the victim's system is open. Figure 3-9 provides an example of this situation.

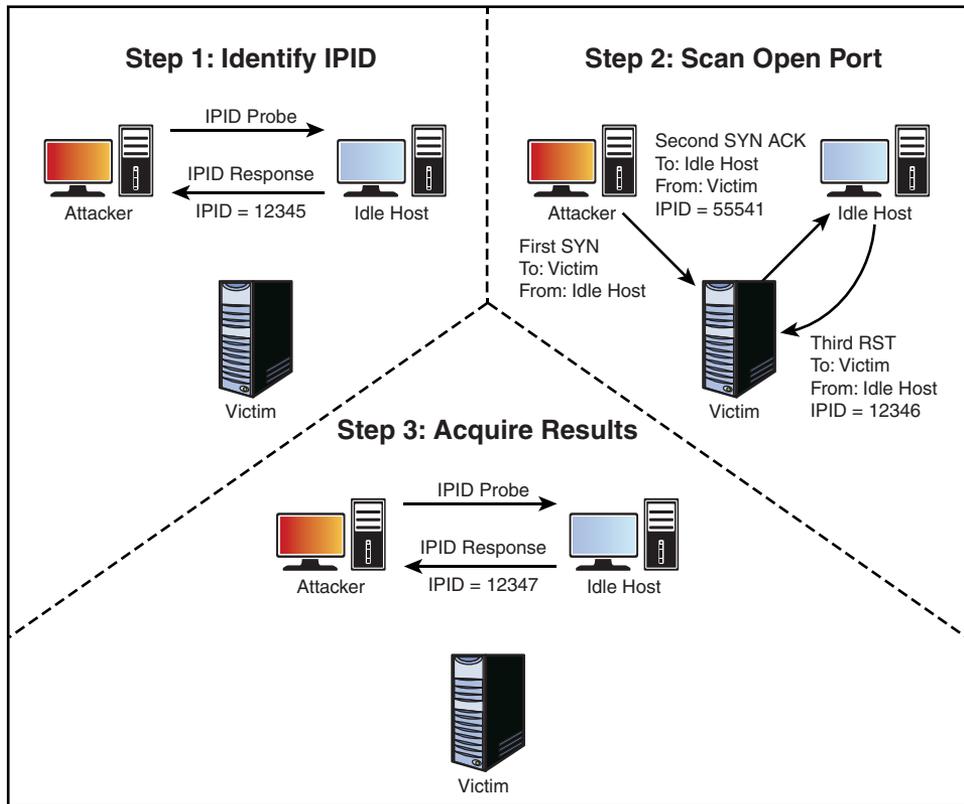


FIGURE 3-9 IPID Open Port

But what if the target system has its port closed? In that situation, the scan starts the same way as previously described. An attacker makes an initial query to determine the idle host's IPID value. Note that the value returned was 12345. In Step 2, the attacker sends a SYN packet addressed to the victim but spoofs it to appear that it originated from the idle host. Because the victim's port is closed, it responds to this query by issuing an RST. Because RSTs don't generate additional RSTs, the communication between the idle host and the victim ends here. Finally, the attacker again probes the idle host and examines the response. Because the victim's port was closed, we can see that the returned IPID was 12346. It was only incremented by one because no communication had taken place since the last IPID probe that determined the initial value. Figure 3-10 provides an example of this situation.

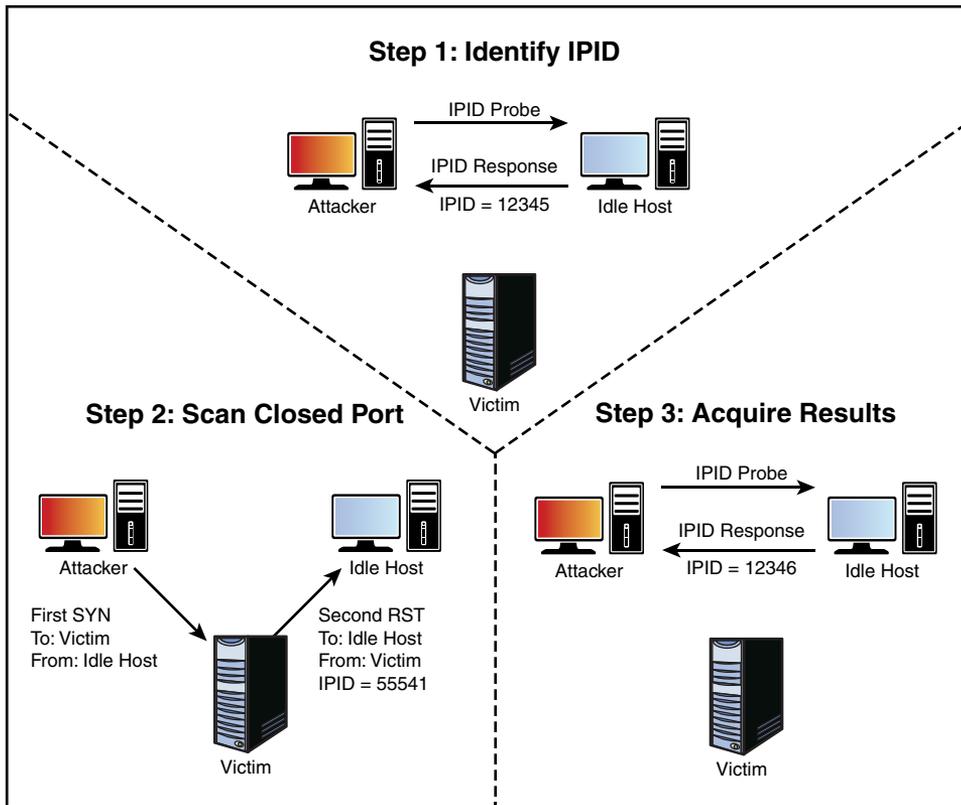


FIGURE 3-10 IPID Port Closed

Although not perfect, this scanning technique enables attackers to obscure their true address. However, limitations apply to the capability of an idle scan. First, the system designated to play the role of the idle host must truly be idle. A chatty system is of little use because the IPID will increment too much to be useful. There is also the fact that not all operating systems use an incrementing IPID. For example, some versions of Linux set the IPID to zero or generate a random IPID value. Again, these systems are of little use in such an attack. Finally, these results must be measured; by this, I mean that several passes need to be performed to validate the results and be somewhat sure that the attacker's conclusions are valid. Although the concept of idle scanning is interesting, there are a few other scan types worth briefly noting:

- **ACK scan:** Sends an ACK probe with random sequence numbers. ICMP type 3 code 13 responses may mean that stateless firewalls are being used, and an RST can mean that the port is not filtered.
- **FTP Bounce scan:** Uses an FTP server to bounce packets off of and make the scan harder to trace.

- **RPC scan:** Attempts to determine whether open ports are RPC ports.
- **Window scan:** Similar to an ACK scan but can sometimes determine open ports. It does so by examining the TCP window size of returned RST packets. On some systems, open ports return a positive window size and closed ones return a zero window size.

Now let's look at UDP scans. UDP is unlike TCP. TCP is built on robust connections, but UDP is based on speed. With TCP, the hacker can manipulate flags in an attempt to generate a TCP response or an error message from ICMP. UDP does not have flags, nor does UDP issue responses. It's a fire and forget protocol! The most you can hope for is a response from ICMP.

If the port is closed, ICMP attempts to send an ICMP type 3 code 3 port unreachable message to the source of the UDP scan. But, if the network is blocking ICMP, no error message is returned. Therefore, the response to the scans might simply be no response. If you are planning on doing UDP scans, plan for unreliable results.

Next, some of the programs that can be used for port scanning are discussed.

Is Port Scanning Legal?

In 2000, two contractors ended up in a U.S. district court because of a dispute over the legality of port scanning. The plaintiff believed that port scanning is a crime, whereas the defendant believed that only by port scanning was he able to determine which ports were open and closed on the span of network he was responsible for. The U.S. district court judge ruled that port scanning was not illegal because it does not cause damage. So, although port scanning is not a crime, you should still seek to obtain permission before scanning a network. Also, home users should review their service provider's terms and conditions before port scanning. Most cable companies prohibit port scanning and maintain the right to disconnect customers who perform such acts, even when they are performing such activities with permission. Time Warner's policy states the following: "Please be aware that Time Warner Road Runner has received indications of port scanning from a machine connected to the cable modem on your Road Runner Internet connection. This violates the Road Runner AUP (Acceptable Use Policy). Please be aware that further violations of the Acceptable Usage Policy may result in the suspension or termination of your Time Warner Road Runner account." See <https://latesthackingnews.com/2017/09/30/port-scanning-legal/>.



Nmap

Nmap was developed by a hacker named Fyodor Yarochkin. This popular application is available for Windows and Linux as a GUI and command-line program. It is probably the most widely used port scanner ever developed. It can do many types of scans and OS identification. It also enables you to control the speed of the scan from slow to insane. Its popularity can be seen by the fact that it's incorporated into other products and was even used in the movie *The Matrix*. Nmap with the help option is shown here so that you can review some of its many switches:

```
C:\ nmap-7.70>nmap -h
Nmap 7.70 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
* -sS TCP SYN stealth port scan (default if privileged (root))
  -sT TCP connect() port scan (default for unprivileged users)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
  -sL list scan that simply does a reverse DNS lookup without actually
    scanning
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sV Version scan probes open ports determining service and app
    names/versions
-sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan. Example range: '1-1024,1080,6666,31337'
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended. Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys
  -6 scans via IPv6 rather than IPv4
  -T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing
    policy
  -n/-R Never do DNS resolution/Always resolve [default: sometimes
    resolve]
  -oN/-oX/-oG <logfile > Output normal/XML/grepable scan logs to
    <logfile>
  -iL <inputfile > Get targets from file; Use '-' for stdin
  -sC Scripting engine
* -S <your_IP >/-e <devicename > Specify source address or network
    interface
```

```
--interactive Go into interactive mode (then press h for help)
--win_help Windows-specific features
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
```

TIP To better understand Nmap and fully prepare for the CEH exam, it's advisable to download and review Nmap's documentation. You can find it at <https://nmap.org/book/man.html>.

NOTE One example of an Nmap switch you should know is decoy. The decoy switch is used to evade an IDS or firewall. The idea is to make it appear to the target that the decoys are the source of the scan, which obscures the real source of the attacker. Decoy can be used two ways. The first is with the RND option so that nmap generates a random set of source IP addresses. The second is that the attacker can specify a specific list of spoofed source addresses.

As shown in the output of the help menu in the previous listing, Nmap can run many types of scans. Nmap is considered a required tool for all ethical hackers.

The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features. It allows users to create and use simple scripts to automate a wide variety of networking tasks. Nmap's output provides the open port's well-known service name, number, and protocol. Ports can either be open, closed, or filtered. If a port is open, it means that the target device will accept connections on that port. A closed port is not listening for connections, and a filtered port means that a firewall, filter, or other network device is guarding the port and preventing Nmap from fully probing it or determining its status. If a port is reported as unfiltered, it means that the port is closed, and no firewall or router appears to be interfering with Nmap's attempts to determine its status.

To run Nmap from the command line, type **nmap**, followed by the switch, and then enter a single IP address or a range. For the example shown here, the **-sT** option was used, which performs a TCP full three-step connection:

```
C:\ nmap-7.70>nmap -sT 192.168.1.108
Starting nmap 7.70 (https://nmap.org/) at 2015-10-05 23:42 Central
Daylight Time
Interesting ports on Server (192.168.1.108):
```

```
(The 1653 ports scanned but not shown below are in state:
filtered)
PORTSTATE SERVICE
80/tcpopenhttp
445/tcp opensmb
515/tcp openprinter
548/tcp openafpovertcp
Nmap run completed -- 1 IP address (1 host up) scanned in
420.475 seconds
```

Several interesting ports were found on this computer, including 80 and 139. A UDP scan performed with the **-sU** switch returned the following results:

```
C:\ nmap-7.70>nmap -sU 192.168.1.108
Starting nmap 7.70 (https://nmap.org/ ) at 2015-10-0523:47
Central
Daylight Time
Interesting ports on Server (192.168.1.108):
(The 1653 ports scanned but not shown below are in state:
filtered)
PORTSTATE SERVICE
69/udpopenftp
Nmap run completed -- 1 IP address (1 host up) scanned in
843.713 seconds
```

Now let's scan a second system so we can see the difference between a Windows computer and a Linux computer. One big clue is the potential for open ports such as 37, 79, 111, and 6000. Those represent programs such as Time, Finger, SunRpc, and X11.

```
[root@mg /root]# nmap -O 192.168.13.10
Starting nmap V. 7.70 (https://nmap.org// )
Interesting ports on unix1 (192.168.13.10):
(The 1529 ports scanned but not shown below are in state: closed)
Port      State      Service
21/tcp    open       ftp
23/tcp    open       telnet
25/tcp    open       smtp
37/tcp    open       time
79/tcp    open       finger
111/tcp   open       sunrpc
139/tcp   filtered   netbios-ssn
513/tcp   open       login
1103/tcp  open       xaudio
2049/tcp  open       nfs
```

```
4045/tcp    open      lockd
6000/tcp    open      X11
7100/tcp    open      font-service
32771/tcp   open      sometimes-rpc5
32772/tcp   open      sometimes-rpc7
32773/tcp   open      sometimes-rpc9
32774/tcp   open      sometimes-rpc11
32775/tcp   open      sometimes-rpc13
32776/tcp   open      sometimes-rpc15
32777/tcp   open      sometimes-rpc17

Remote operating system guess: Solaris 2.6 - 2.7
Uptime 319.638 days (since Wed Aug 09 19:38:19 2017)
Nmap run completed -- 1 IP address (1 host up) scanned in
7 seconds
```

Notice that the ports shown from this scan are much different from what was seen from Windows scans earlier in the chapter. Ports such as 37, 79, 111, and 32771 are shown as open. Also note that Nmap has identified the OS as Solaris. If you can, you also want to identify which applications are installed. Commands to find common ones include the following:

```
ls -alh /usr/bin/
ls -alh /sbin/
ls -alh /var/cache/apt/archives0
dpkg -l
rpm -qa
```

TIP Regardless of the OS, scanning an IPv6 network is much harder than scanning IPv4 network ranges in that the search space is so much larger. The amount of IP addresses that must be scanned in IPv6 make it difficult to gather valid addresses. Other techniques are typically used to gather valid addresses. IPv6 addresses must be harvested in some way, such as by network traffic, recorded logs, or address received from.

Zenmap is the official Nmap Security Scanner GUI. Most of the options in Zenmap correspond directly to the command-line version. Some people call Zenmap the Nmap tutor because it displays the command-line syntax at the bottom of the GUI interface, as shown in Figure 3-11.

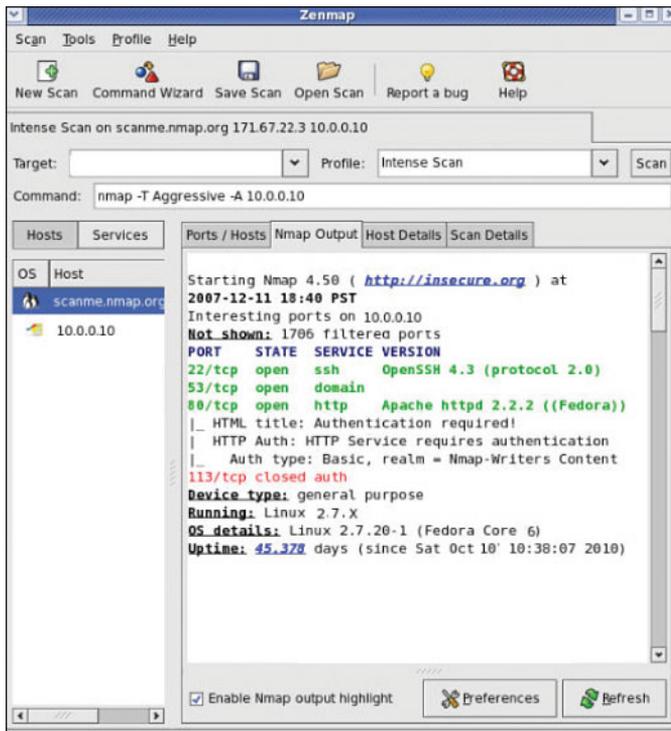


FIGURE 3-11 Zenmap

SuperScan

SuperScan is written to run on Windows machines. It's a versatile TCP/UDP port scanner, pinger, and hostname resolver. It can perform ping scans and port scans using a range of IP addresses, or it can scan a single host. It also has the capability to resolve or reverse-lookup IP addresses. It builds an easy-to-use HTML report that contains a complete breakdown of the hosts that were scanned. This includes information on each port and details about any banners that were found. It's free; therefore, it is another tool that all ethical hackers should have.

THC-Amap

THC-Amap is another example of a tool that is used for scanning and banner grabbing. One problem that traditional scanning programs have is that not all services are ready and eager to give up the appropriate banner. For example, some services, such as Secure Sockets Layer (SSL), expect a handshake. Amap handles this by storing a collection of responses that it can fire off at the port to interactively elicit it to respond. Amap was the first to perform this functionality, but it has been replaced with Nmap. One technique is to use this program by taking the greppable format of Nmap as an

input to scan for those open services. Defeating or blocking Amap is not easy, although one technique would be to use a *port-knocking* technique. Port knocking is similar to a secret handshake or combination. Only after inputting a set order of port connections can a connection be made. For example, you may have to first connect on 80, 22, and 123 before connecting to 443. Otherwise, the port will show as closed.

Hping

Hping is another very useful ethical hacking tool that can perform both ping sweeps and port scans. Hping works on Windows and Linux computers and can function as a packet builder. You can find the Hping tool at <http://www.hping.org> or download the Linux Backtrack distribution, which also contains Hping. Hping2 and 3 can be used for firewall testing, identifying honeypots, and port scanning. Here are some other Hping3 syntax examples of note:

- **Ping sweep:** `hping3 -1 IP_Address`
- **UDP scan:** `hping3 -2 IP_Address`
- **SYN scan:** `hping3 -8 IP_Address`
- **ACK scan:** `hping3 -A IP_Address`
- **IPID collection:** `IP_Address -Q -p 139 -s`
- **XMAS scan:** `hping3 -F -P -U IP_Address`

TIP Hping is a powerful tool that you can use to bypass filtering devices by injecting crafted or otherwise modified IP packets or to port scan and perform just about any type of scan that Nmap can. Hping syntax could come up on the exam.

Port Knocking

Port knocking is a method of establishing a connection to a host that does not initially indicate that it has any open ports. Port knocking works by having the remote device send a series of connection attempts to a specific series of ports. It is somewhat analogous to a secret handshake. After the proper sequence of port knocking has been detected, the required port is opened, and a connection is established. The advantage of using a port-knocking technique is that hackers cannot easily identify open ports. The disadvantages include the fact that the technique does not harden the underlying application. Also, it isn't useful for publicly accessible services. Finally, anyone who has the ability to sniff the network traffic will be in possession of the appropriate knock sequence. A good site to check out to learn more about this defensive technique is <http://www.portknocking.org>.

War Driving

War driving is named after war dialing because it is the process of looking for open access points. Many pen tests contain some type of war driving activity. The goal is to identify open or rogue access points. Even if the organization has secured its wireless access points, there is always the possibility that employees have installed their own access points without the company's permission. Unsecured wireless access points can be a danger to organizations because, much like modems, they offer the hacker a way into the network that might bypass the firewall. A whole host of security tools released for Windows and Linux is available to use for war driving and wireless cracking activities.

Key Topic

OS Fingerprinting

At this point in the information-gathering process, the hacker has made some real headway. IP addresses, active systems, and open ports have been identified. Although the hacker might not yet know the type of systems he is dealing with, he is getting close. Fingerprinting is the primary way to identify a specific system. Fingerprinting works because each vendor implements the TCP/IP stack in different ways. For example, it's much the same as when you text a specific friend who typically says something like, "Hey, what's up?" while another friend simply says, "Hi." There are two ways in which the hacker can attempt to identify the targeted devices. The hacker's first choice is passive fingerprinting. The hacker's second choice is to perform active fingerprinting, which basically sends malformed packets to the target in hope of eliciting a response that will identify it. Although active fingerprinting is more accurate, it is not as stealthy as passive fingerprinting.

Passive fingerprinting is really sniffing, because the hacker is sniffing packets as they come by. These packets are examined for certain characteristics that can be pointed out to determine the OS. The following are four commonly examined items that are used to fingerprint the OS:

- **IP TTL value:** Different operating systems set the TTL to unique values on outbound packets.
- **TCP window size:** OS vendors use different values for the initial window size.
- **IP DF option:** Not all OS vendors handle fragmentation in the same way. 1500 bytes is a common size with Ethernet.
- **IP Type of Service (TOS) option:** TOS is a 3-bit field that controls the priority of specific packets. Again, not all vendors implement this option in the same way.

These are just four of many possibilities that can be used to passively fingerprint an OS. Other items that can be examined include IP identification number (IPID), IP options, TCP options, and even ICMP. Ofir Arkin has written an excellent paper on

this, titled “ICMP Usage in Scanning.” An example of a passive fingerprinting tool is the Linux-based tool P0f. P0f attempts to passively fingerprint the source of all incoming connections after the tool is up and running. Because it’s a truly passive tool, it does so without introducing additional traffic on the network. P0fv2 is available at <http://lcamtuf.coredump.cx/p0f.tgz>.

NOTE One of the most common methods used to determine the OS is to examine the TTL. For example, the default TTL of a Linux system is 64, the default TTL of Windows is 128, and the default TTL of routers is typically 254.

Active fingerprinting is more powerful than passive fingerprint scanning because the hacker doesn’t have to wait for random packets, but as with every advantage, there is usually a disadvantage. This disadvantage is that active fingerprinting is not as stealthy as passive fingerprinting. The hacker actually injects the packets into the network. Active fingerprinting has a much higher potential for being discovered or noticed. Like passive OS fingerprinting, active fingerprinting examines the subtle differences that exist between different vendor implementations of the TCP/IP stack. Therefore, if hackers probe for these differences, the version of the OS can most likely be determined. One of the individuals who has been a pioneer in this field of research is Fyodor. He has an excellent chapter on remote OS fingerprinting at <https://nmap.org/book/osdetect.html>. Listed here are some of the basic methods used in active fingerprinting:

- **The FIN probe:** A FIN packet is sent to an open port, and the response is recorded. Although RFC 793 states that the required behavior is not to respond, many operating systems such as Windows will respond with an RST.
- **Bogus flag probe:** As you might remember from Table 3-6, the flag field is only 1 byte in the TCP header. A bogus flag probe sets one of the used flags along with the SYN flag in an initial packet. Linux will respond by setting the same flag in the subsequent packet.
- **Initial sequence number (ISN) sampling:** This fingerprinting technique works by looking for patterns in the ISN. Although some systems use truly random numbers, others, such as Windows, increment the number by a small fixed amount.
- **IPID sampling:** Many systems increment a systemwide IPID value for each packet they send. Others, such as older versions of Windows, do not put the IPID in network byte order, so they increment the number by 256 for each packet.
- **TCP initial window:** This fingerprint technique works by tracking the window size in packets returned from the target device. Many operating systems use exact sizes that can be matched against a database to uniquely identify the OS.
- **ACK value:** Again, vendors differ in the ways they have implemented the TCP/IP stack. Some operating systems send back the previous value +1, whereas others send back more random values.

- **Type of service:** This fingerprinting type tweaks ICMP port unreachable messages and examines the value in the TOS field. Whereas some use 0, others return different values.
- **TCP options:** Here again, different vendors support TCP options in different ways. By sending packets with different options set, the responses will start to reveal the server's fingerprint.
- **Fragmentation handling:** This fingerprinting technique takes advantage of the fact that different OS vendors handle fragmented packets differently. RFC 1191 specifies that the maximum transmission unit (MTU) is normally set between 68 and 65535 bytes. This technique was originally discovered by Thomas Ptacek and Tim Newsham.

Active Fingerprinting Tools

One of the first tools to be widely used for active fingerprinting back in the late 1990s was Queso. Although no longer updated, it helped move this genre of tools forward. Nmap is the tool of choice for active fingerprinting and is one of the most feature-rich free fingerprint tools in existence today. Nmap's database can fingerprint literally hundreds of different operating systems. Fingerprinting with Nmap is initiated by running the tool with the **-O** option. When started with this command switch, Nmap probes port 80 and then ports in the 20 to 23 range. Nmap needs one open and one closed port to make an accurate determination of what OS a particular system is running.

Here is an example:

```
C:\ nmap-7.70>nmap -O 192.168.123.108
Starting nmap 6.25 (https://nmap.org/) at 2005-10-0715:47
Central
Daylight Time
Interesting ports on 192.168.1.108:
(The 1653 ports scanned but not shown below are in state:
closed)
PORTSTATE SERVICE
80/tcpopenhttp
139/tcp opennetbios-ssn
515/tcp openprinter
548/tcp openafpovertcp
Device type: general purpose
Running: Linux 2.4.X|2.5.X
OS details: Linux Kernel 2.4.0 - 2.5.20
Uptime 0.282 days (since Fri Oct 07 09:01:33 2018)
Nmap run completed -- 1 IP address (1 host up) scanned in 4.927
seconds
```


Fingerprinting Services

If there is any doubt left as to what a particular system is running, this next step of information gathering should serve to answer those questions. Knowing what services are running on specific ports allows the hacker to formulate and launch application-specific attacks. Knowing the common default ports and services and using tools such as Telnet and Netcat is one way to ensure success at this pre-attack stage.

Default Ports and Services

A certain amount of default information and behavior can be gleaned from any system. For example, if a hacker discovers a Windows 2012 server with port 80 open, he can assume that the system is running IIS 8.0, just as a Linux system with port 25 open is likely to be running Sendmail. Although it's possible that the Windows 2012 machine might be running another version or type of web server, that most likely is not a common occurrence.

Keep in mind that at this point, the attacker is making assumptions. Just because a particular port is active or a known banner is returned, you cannot be certain that information is correct. Ports and banners can be changed, and assumptions by themselves can be dangerous. Additional work will need to be done to verify what services are truly being served up by any open ports.



Finding Open Services

The scanning performed earlier in the chapter might have uncovered other ports that were open. Most scanning programs, such as Nmap and SuperScan, report what common services are associated with those open ports. The easiest way to determine what services are associated with the open ports that were discovered is by banner grabbing.

Banner grabbing takes nothing more than the Telnet and FTP client built in to the Windows and Linux platforms. Banner grabbing provides important information about what type and version of software is running. Many servers can be exploited with just a few simple steps if the web server is not properly patched. Telnet is an easy way to do this banner grabbing for FTP, SMTP, HTTP, and others. The command issued to banner grab with Telnet would contain the following syntax: **telnet *IP_Address* port**. An example of this is shown here. This banner-grabbing attempt was targeted against a web server:

```
C:\>telnet 192.168.1.102 80
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/7.5
Date: Fri, 07 Oct 2012 22:22:04 GMT
```

```

Content-Type: text/html
Content-Length: 87
<html><head><title>Error</title></head><body>The parameter is
incorrect. </body>
</html>
Connection to host lost.

```

After the command was entered, **telnet 192.168.1.102 80**, the Return key was pressed a couple of times to generate a response. As noted in the Telnet response, this banner indicates that the web server is IIS 7.5.

The Microsoft IIS web server's default behavior is to return a banner after two carriage returns. This can be used to pinpoint the existence of an IIS server.

Telnet isn't your only option for grabbing banners; HTTPrint is another choice. It is available for both Windows and Linux distributions. It is not a typical banner-grabbing application, in that it can probe services to determine the version of services running. Its main fingerprinting technique has to do with the semantic differences in how web servers/applications respond to various types of probes. Here is an example of a scan:

```

./httprint -h 192.168.1.175 -s signatures.txt
httprint - web server fingerprinting tool
Finger Printing on http://192.168.1.175:80/
Finger Printing Completed on http://192.168.1.175:80/
-----
Host: 192.168.1.175
Derived Signature:
Apache/2.2.0 (Fedora RedHat)
9E431BC86ED3C295811C9DC5811C9DC5050C5D32505FCFE84276E4BB811C9DC5
0D7645B5811C9DC5811C9DC5CD37187C11DDC7D7811C9DC5811C9DC58A91CF57FCCC5
35B6ED3C295FCCC535B811C9DC5E2CE6927050C5D336ED3C2959E431BC86ED3C295
E2CE69262A200B4C6ED3C2956ED3C2956ED3C2956ED3C295E2CE6923E2CE69236ED
3C295811C9DC5E2CE6927E2CE6923
Banner Reported: Apache/2.2.0 (Fedora RedHat)
Banner Deduced: Apache/2.0.x
Score: 140
Confidence: 84.31-----

```

Netcat can also be used for banner grabbing. Netcat is shown here to introduce you to its versatility. Netcat is called the “Swiss-army knife of hacking tools” because of its many uses. To banner grab with Netcat, you issue the following command from the command line:

```
nc -v -n IP_Address Port
```

This command gives you the banner of the port you asked to check. Netcat is available for Windows and Linux. If you haven't downloaded Netcat, don't feel totally left behind; FTP is another choice for banner grabbing. Just FTP to the target server and review the returned banner.

Most all port scanners, including those discussed in this chapter, also perform banner grabbing. However, there are lots of tools for the security professional to use to analyze open ports and banners. Some of the more notable ones you may want to review include the following:

- **ID Serve:** <https://www.grc.com/id/idserve.htm>
- **NetworkMiner:** <http://www.netresec.com/?page=NetworkMiner>
- **Satori:** <http://chatteronthewire.org/>
- **Netcraft:** http://toolbar.netcraft.com/site_report

Although changing banner information is not an adequate defense by itself, it might help to slow a hacker. In the Linux environment, you can change the ServerSignature line in the httpd.conf file to ServerSignature off. In the Windows environment, you can install the UrlScan security tool. UrlScan contains the RemoveServer-Header feature, which removes or alters the identity of the server from the "Server" response header in response to the client's request.

**Key
Topic**

Mapping the Network Attack Surface

The hacker would have now gained enough information to map the network. Mapping the network provides the hacker with a blueprint of the organization. There are manual and automated ways to compile this information. Manual and automated tools are discussed in the following sections.

Manual Mapping

If you have been documenting findings, the matrix you began at the start of this chapter should be overflowing with information. This matrix should now contain domain name information, IP addresses, DNS servers, employee info, company location, phone numbers, yearly earnings, recently acquired organizations, email addresses, the publicly available IP address range, open ports, wireless access points, modem lines, and banner details.

Automated Mapping

If you prefer a more automated method of mapping the network, a variety of tools are available. Visual traceroute programs, such as SolarWinds's Network Topology

Mapper (<http://www.solarwinds.com/network-topology-mapper>), can help you map out the placement of these servers. You can even use Nmap scripts to trace a route and map the geolocation of a target. As an example, **nmap --traceroute --script traceroute-geolocation.nse -p 80 example.com** would perform a traceroute and provide geolocation data for each hop along the way. Geolocation allows you to identify information such as country, region, ISP, and the like. Examples of geolocation tools include IP Location Finder (<https://tools.keycdn.com>) and GeoIP Lookup Tool (<https://www.ultratools.com>).

Automatic mapping can be faster but might generate errors or sometimes provide erroneous results. Table 3-7 reviews some of the primary steps we have discussed.

NLog is one option to help keep track of your scanning and mapping information. NLog enables you to automate and track the results of your Nmap scans. It allows you to keep all your Nmap scan logs in a database, making it possible to easily search for specific entries. It's browser based, so you can easily view the scan logs in a highly customizable format. You can add your own extension scripts for different services, so all hosts running a certain service will have a hyperlink to the extension script. NLog is available at <http://nlog-project.org/>.

CartoReso is another network mapping option. If run from the Internet, the tool will be limited to devices that it can contact. These will most likely be devices within the demilitarized zone (DMZ). Run internally, it will diagram a large portion of the network. In the hands of a hacker, it's a powerful tool, because it uses routines taken from a variety of other tools that permit it to perform OS detection port scans for service detection and network mapping using common traceroute techniques. You can download it from <https://sourceforge.net/projects/cartoreso/>.

A final item worth discussing is that attacker the will typically attempt to hide her activity while actively probing a victim's network. This can be attempted via anonymizers and proxies. The concept is to try to obscure the true source address. Examples of tools that are available for this activity include the following:

- Proxy Switcher
- Proxy Workbench
- CyberGhost
- Tor

TIP Kali Linux, at <https://www.kali.org/>, contains many of the tools discussed in this chapter and is used for penetration testing.

Table 3-7 The Seven Steps of the Pre-Attack Phase

Step	Title	Active/Passive	Common Tools
One	Information gathering	Passive	www.domaintools.com, ARIN, IANA, Whois, Nslookup
Two	Determining network range	Passive	RIPE, APNIC, LACNIC, ARIN
Three	Identifying active machines	Active	Ping, traceroute, SuperScan, Angry IP Scanner
Four	Finding open ports and access points	Active	Nmap, Hping, Angry IP Scanner, SuperScan
Five	OS fingerprinting	Active/passive	Nmap, Winfingerprint, P0f, Xprobe2
Six	Fingerprinting services	Active	Nmap, Telnet, FTP, Netcat
Seven	Mapping the network attack surface	Active	CartoReso, traceroute, Network Topology Mapper

Summary

In this chapter, you learned the seven steps that compose the pre-attack phase: information gathering, determining the network range, identifying active machines, finding open ports and access points, OS fingerprinting, fingerprinting services, and mapping the network attack surface.

This chapter is an important step for the ethical hacker because at this point you are gathering information to launch an attack and determine the best path forward. The more information that is gathered here, the better the chance of success. You might find enough information at this point to be able to launch an attack. If not, the information gathered will serve as a foundation for subsequent steps of the attack. An important part of ethical hacking is documentation. That's why several ways to collect and document your findings are shown. There is no such thing as too much information. You may want to use a proxy or anonymizer to obscure the probes. These notes will prove useful when you prepare your report. Finally, make sure that the organization has given you written permission before beginning any work, even the reconnaissance.

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have several choices for exam preparation: the exercises here, Chapter 12, “Final Preparation,” and the exam simulation questions in the Pearson Test Prep Software Online.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 3-8 lists a reference of these key topics and the page numbers on which each is found.

Table 3-8 Key Topics for Chapter 3

Key Topic Element	Description	Page Number
List	Describes the seven-step information-gathering process	90
Table 3-6	Understand and define TCP flags	118
Section	Describes NMAP switches	124
Section	Describes how passive and active OS fingerprinting works	130
Section	Explains how to find open services: banner grabbing	134
Section	Explains tools used to map the attack surface	136



Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

active fingerprinting, CNAMEs, covert channel, demilitarized zone (DMZ), denial of service (DoS), echo reply, echo request, EDGAR database, initial sequence number (ISN), Internet Assigned Numbers Authority (IANA), intrusion detection system (IDS), Nslookup, open source, passive fingerprinting, ping sweep, port knocking, script kiddie, Simple Network Management Protocol (SNMP), social engineering, synchronize sequence number, Time To Live (TTL), traceroute, war driving, Whois, written authorization, zone transfer

Exercises

3.1 Performing Passive Reconnaissance

The best way to learn passive information gathering is to use the tools. In this exercise, you perform reconnaissance on several organizations. Acquire only the information requested.

Estimated Time: 20 minutes.

Step 1. Review Table 3-9 to determine the target of your passive information gathering.

Table 3-9 Passive Information Gathering

Domain Name	IP Address	Location	Contact Person	Address and Phone Number
Redriff.com				
Examcram.com	72.3.246.59			
Rutgers.edu				

Step 2. Start by resolving the IP address. This can be done by pinging the site.

Step 3. Next, use a tool such as <https://www.whois.net> or any of the other tools mentioned throughout the chapter. Some of these include

- <http://www.betterwhois.com>
- www.allwhois.com
- <http://geektools.com>
- www.centralops.net
- www.dnsstuff.com

Step 4. To verify the location of the organization, perform a traceroute or a ping with the **-r** option.

Step 5. Use the ARIN, RIPE, and IANA to fill in any information you have yet to acquire.

Step 6. Compare your results to those found in Appendix A. Results may vary.

3.2 Performing Active Reconnaissance

The best way to learn active information gathering is to use the tools. In this exercise, you perform reconnaissance on your own internal network. If you are not on a test network, make sure that you have permission before scanning it, or your action may be seen as the precursor of an attack.

Estimated Time: 15 minutes.

- Step 1.** Download the most current version of Nmap from <https://nmap.org/download.html>. For Windows systems, the most current version is 7.30.
- Step 2.** Open a command prompt and go to the directory in which you have installed Nmap.
- Step 3.** Run **nmap -h** from the command line to see the various options.
- Step 4.** You'll notice that Nmap has many options. Review and find the option for a full connect scan. Enter your result here: ____
- Step 5.** Review and find the option for a stealth scan. Enter your result here: ____
- Step 6.** Review and find the option for a UDP scan. Enter your result here: ____
- Step 7.** Review and find the option for a fingerprint scan. Enter your result here: ____
- Step 8.** Perform a full connect scan on one of the local devices you have identified on your network. The syntax is **nmap -sT IP_Address**.
- Step 9.** Perform a stealth scan on one of the local devices you have identified on your network. The syntax is **nmap -sS IP_Address**.
- Step 10.** Perform a UDP scan on one of the local devices you have identified on your network. The syntax is **nmap -sU IP_Address**.
- Step 11.** Perform a fingerprint scan on one of the local devices you have identified on your network. The syntax is **nmap -O IP_Address**.
- Step 12.** Observe the results of each scan. Could Nmap successfully identify the system? Were the ports it identified correct?

Review Questions

1. Your client has asked you to run an Nmap scan against the servers it has located in its DMZ. The client would like you to identify the OS. Which of the following switches would be your best option?
 - a. **nmap -P0**
 - b. **nmap -sO**
 - c. **nmap -sS**
 - d. **nmap -O**

2. During an internal pen test, you have gained access to an internal switch. You have been able to SPAN a port and are now monitoring all traffic with Wireshark. While reviewing this traffic, you are able to identify the OS of the devices that are communicating. What best describes this activity?
 - a. Vulnerability scanning
 - b. Nmap port scanning
 - c. Active OS fingerprinting
 - d. Passive OS fingerprinting
3. ICMP is a valuable tool for troubleshooting and reconnaissance. What is the correct type for a ping request and a ping response?
 - a. Ping request type 5, ping reply type 3
 - b. Ping request type 8, ping reply type 0
 - c. Ping request type 3, ping reply type 5
 - d. Ping request type 0, ping reply type 8
4. Which of the following is a vulnerability in the Bash shell that was discovered in 2014 and thereafter exploited to launch a range of attacks against Linux and UNIX systems?
 - a. Shellshock
 - b. Heartbleed
 - c. Bashshell
 - d. Poodle
5. As part of a pen test, you have port scanned a Linux system. Listed here is the scan you performed: **nmap -sX -vv -P0 192.168.1.123 -p 80**. If the system had the specific listening port open, what would be returned?
 - a. RST
 - b. No response
 - c. SYN ACK
 - d. ACK
6. Which of the following Netcat commands could be used to perform a UDP scan of the lower 1024 ports?
 - a. **Nc -sS -O target 1-1024**
 - b. **Nc -hU <host(s)>**
 - c. **Nc -sU -p 1-1024 <host(s)>**
 - d. **Nc -u -v -w2 <host> 1-1024**

7. You have been assigned a junior pen tester during a pen test. You performed the following scan:

```
nmap -sL www.example.com
Starting Nmap 6.25 ( http://nmap.org ) at 2016-10-12 18:
 46 Central Daylight Time
Host 93.184.216.34 not scanned
```

Your partner asks you to explain the results. Which of the following best describes the correct answer?

- a. The system was offline.
 - b. The technique only checks DNS and does not scan.
 - c. The syntax is incorrect.
 - d. ICMP is blocked, so no scan is performed.
8. Which of the following sets all TCP flags to zeros?
- a. **nmap -sn 192.168.1.1/24**
 - b. **nmap -null 192.168.1.1/24**
 - c. **nmap -sX 192.168.1.1/24**
 - d. **nmap -sI 192.168.1.1/24**
9. You have captured some packets from a system you would like to passively fingerprint. You noticed that the IP header length is 20 bytes and there is a datagram length of 84 bytes. What do you believe the system to be?
- a. Windows XP
 - b. Linux
 - c. Windows 7
 - d. Windows 8
10. During the network mapping phase of a pen test, you have discovered the following two IP addresses: 192.168.1.24 and 192.168.1.35. They both have a mask of 255.255.255.224. Which of the following is true?
- a. They are on the same network.
 - b. They both have a default gateway of 192.168.1.63.
 - c. They both have a default gateway of 192.168.1.254.
 - d. They are on separate subnets.

11. What type of scan is harder to perform because of the lack of response from open services and because packets could be lost due to congestion or from fire-wall blocked ports?
 - a. Stealth scanning
 - b. ACK scanning
 - c. UDP scanning
 - d. FIN scan

12. You would like to perform a scan that runs a script against SSH and attempts to extract the SSH host key. Which of the following is the correct syntax?
 - a. `nmap -sC -p21, 111, 139 -T3 www.knowthetrade.com`
 - b. `nmap -sC -p22, 111, 139 -T4 www.knowthetrade.com`
 - c. `nmap -sL -p21, 111, 139 -T3 www.knowthetrade.com`
 - d. `nmap -sI -p22, 111, 139 -T4 www.knowthetrade.com`

13. You have just performed an ACK scan and have been monitoring a sniffer while the scan was performed. The sniffer captured the result of the scan as an ICMP type 3 code 13. What does this result mean?
 - a. The firewall is only a router with an ACL.
 - b. The port is open.
 - c. Port knocking is used.
 - d. The port is closed.

14. One of the members of your security assessment team is trying to find out more information about a client's website. The Brazilian-based site has a .com extension. She has decided to use some online Whois tools and look in one of the Regional Internet Registries. Which of the following represents the logical starting point?
 - a. AfriNIC
 - b. ARIN
 - c. APNIC
 - d. RIPE

15. You have captured the Wireshark scan results shown in Figure 3-13 and are attempting to determine what type of scan was performed against the targeted system. What is your answer?
- SYN
 - IPID
 - NULL
 - XMAS

```
Internet Protocol Version 4, Src: 192.168.1.8 (192.168.1.8), Dst: 192.168.1.123 (192.168.1.123)
Transmission Control Protocol, Src Port: 33310 (33310), Dst Port: ftp (21), Seq: 1, Len: 0
  Source port: 33310 (33310)
  Destination port: ftp (21)
  [Stream index: 44]
  Sequence number: 1 (relative sequence number)
  Header length: 20 bytes
  Flags: 0x00 (<None>)
  Window size value: 2048
  [Calculated window size: 2048]
```

FIGURE 3-13 Wireshark Scan Capture

16. What is the purpose of the following Nmap scan?
- ```
Nmap -sn 192.168.123.1-254
```
- Ping only on the targets, no port scan
  - A NULL TCP scan
  - A TCP port scan
  - Port scan all targets
17. You're starting a port scan of a new network. Which of the following can be used to scan all ports on the 192.168.123.1 network?
- nmap -p 1,65536 192.168.123.1**
  - nmap -p- 192.168.123.1**
  - nmap 192.168.123.1 -ports "all"**
  - nmap -p 0-65536 192.168.123.1**
18. Which of following port-scanning techniques can be used to map out the firewall rules on a router?
- NULL scan
  - ACK scan
  - Inverse flag scan
  - Firewalk

19. What are the two ICMP codes used when performing a ping?
  - a. Type 0 and 8
  - b. Type 0 and 3
  - c. Type 3 and 5
  - d. Type 5 and 11
  
20. You have successfully scanned a system and identified the following port 80 open. What is the next step you should perform?
  - a. Attempt to go to the web page and examine the source code.
  - b. Use FTP to connect to port 80.
  - c. Telnet to the open port and grab the banner.
  - d. Attempt to connect to port 443.

## Suggested Reading and Resources

[http://www.infosecwriters.com/text\\_resources/doc/Demystifying\\_Google\\_Hacks.doc](http://www.infosecwriters.com/text_resources/doc/Demystifying_Google_Hacks.doc): Demystifying Google hacks

<http://www.domaintools.com/>: Online Whois query website

<https://nmap.org/book/man-port-scanning-techniques.html>: Port-scanning techniques

<https://www.exploit-db.com/google-hacking-database/>: The Google Hackers Guide

<https://www.greycampus.com/opencampus/ethical-hacking/scanning-methodology>: The port scanning process

<https://www.hackingloops.com/nmap-cheat-sheet-port-scanning-basics-ethical-hackers/>: Nmap Cheat Sheet

[http://www.forensicswiki.org/wiki/OS\\_fingerprinting](http://www.forensicswiki.org/wiki/OS_fingerprinting): OS fingerprinting

<http://www.utc.edu/center-information-security-assurance/pdfs/course-paper-5620-attacktcpip.pdf>: TCP/IP from a security viewpoint

<https://blog.sucuri.net/2014/09/quick-analysis-of-a-ddos-attack-using-ssdp.html>: Simple Service Discovery Protocol (SSDP) usage in scanning

*This page intentionally left blank*

# Index

## Numbers

1G technology, 410  
802.11, 424–425

## A

Absinthe, 417  
access control, to the cloud, 552  
access points, war driving, 130  
ACK scans, 119, 122  
AckCmd, 238  
ACLs (access control lists), 476–477  
active fingerprinting, 131–133  
active machines, identifying, 115  
active sniffing, 276–277  
active vulnerability assessments,  
    253–254  
activity blockers, 248  
activity profiling, 312  
Acunetix Web Vulnerability Scanner,  
    360  
AD (Active Directory), 154  
ad hoc mode, 423  
ADMutate, 474  
advantages, of cloud computing, 550  
AES (Advanced Encryption Standard),  
    511–512, 514  
airplay-ng tool, 432–433  
airmon-ng tool, 430  
airodump-ng tool, 431  
AirSnare, 447  
ALE (annual loss expectancy),  
    calculating, 12  
algorithms, 508–509  
Amitis, 226  
analyzing malware, 249  
    dynamic analysis, 251–253  
    static analysis, 250–251  
Android devices, 414–415  
    Device Administration API,  
        414–415  
    malware, 412, 416  
    rooting, 416  
    UID, 415  
    vulnerabilities, 414–415  
AndroRAT, 416  
anomaly-based analysis, 464–465  
anonymizers, 137  
antivirus software, 246–248  
APIs, unprotected, 353–356  
AppDetectivePro, 383  
appenders, 215  
Application layer, 55  
application layer (TCP/IP)  
    DHCP, 61  
    DNS, 62–63  
    FTP, 61  
    ports, 60–61  
        blocking, 61  
    session hijacking, 295  
        client-side attacks, 297–298  
        man-in-the-browser attacks, 299  
        man-in-the-middle attacks, 296  
        predictable session token ID, 296  
        session fixation attacks, 299

- session replay attacks, 299
    - session sniffing, 295–296
  - SMTP, 62
  - SNMP, 62
  - Telnet, 62
  - tunneling, 237–238, 485
  - application-level attacks, 307–308
  - application-level gateways, 478
  - applications
    - testing, 22
    - vulnerabilities in, 10
  - approval process for penetration testing, 27–28
  - APs (access points), site surveys, 445
  - ARO (annual rate of occurrence),
    - calculating, 12
  - ARP (Address Resolution Protocol), 76, 278–279
    - and TCP/IP, 278–279
  - ARP poisoning, 279–281
  - assessments, defining scope of, 24
  - assets, 8
    - EF, 12
  - asymmetric encryption algorithms, 506, 508–510, 515
    - Diffie-Hellman, 516
    - ECC, 516–517
    - ElGamal, 516
    - hashing, 517–518
    - RSA, 516
  - attack evasion techniques, 472–473
    - flooding, 470
    - insertion and evasion, 470
    - shellcode attacks, 471–472
  - attribute command, 185
  - audits, 15, 53, 360
  - Auernheimer, Andrew, 17
  - authentication, 506–507
    - basic, 374
    - certificate-based, 375
    - forms-based, 375
    - Kerberos, 175
    - Linux, 178–180
    - MD5, 375
    - passwords, 508
    - Windows, 173–175
    - wireless, 446
  - authentication system testing, 22
  - automated exploit tools
    - BeEF, 357
    - Canvas, 358
    - Core Impact, 358
    - Metasploit, 357
  - automated mapping, 136–138
  - automated password guessing, 167
  - availability, 7
  - Avatar, 183
  - Azazel, 183
- ## B
- back-ups, performing to reduce risk, 10–11
  - backdoors, 52
  - in-band SQL injection, 389
  - bandwidth, limiting, 313
  - BangleDoS, 311
  - banner grabbing, 134–136, 483
  - Base64, 535
  - basic authentication, 374
  - bastion hosts, 479
  - BeEF (Browser Exploitation Framework), 357
  - BetterCAP, 281
  - Bing Maps, 93
  - BinText, 250
  - biometrics, characteristics of, 166–167
  - BIOS infections, 214
  - BitLocker, 531
  - black box testing, 13–14
  - black hat hackers, motivations, 16
  - black hole filtering, 313

- black holes, 493
  - Blackberry, 418
  - BlackHole Rat, 225
  - BLE (Bluetooth Low Energy), 558
  - blind SQL injection, 389
  - block ciphers, 512
  - BlueBug, 421
  - Bluejacking, 420–421
  - BlueScanner, 421
  - Bluesnarfing, 421
  - Bluesniff, 421
  - Bluetooth
    - classifications of, 419
    - technologies, 419–420
    - vulnerabilities, 420–421
  - Bluetooth Smart, 558
  - bogons, 476
  - bogus flag probes, 131
  - Booleans, using in SQL injection attacks, 394
  - botnets, 560–561
    - countermeasures, 563–566
    - crimeware kits, 562
    - fast-flux, 561
    - installation, 563
    - well-known, 562
  - Brain virus, 216–217
  - broadcast MAC addresses, 75
  - Brown, Justin, *Google Hacking for Penetration Testers*, 101
  - browsers, 330–332
    - cookies, 377
    - viewing, 377–378
  - brute-force attacks, 176, 376
  - Brutus, 376, 536
  - Bryant, Darla, 487
  - BTCrack, 419
  - buffer overflows, 373–374
    - exploiting, 171–173
  - bump attacks, 413
  - Burger, Ralf, 216–217
  - Burneye, 228
  - Burp Suite, 301, 376–377
  - BYOD (bring your own device), 406, 414
  - bypassing
    - firewalls, 484–488
    - switches
      - with ARP poisoning, 279–281
      - with MAC flooding, 281–283
- ## C
- C language, vulnerabilities to buffer overflow, 172, 373–374
  - CA (Certificate authority), 526
  - Caesar’s cipher, 507–508
  - Caffrey, Aaron, 225
  - Cain and Abel, 281
  - calculating, SLE, 12
  - Canvas, 358
  - CartoReso, 137
  - CBC (Cipher Block Chaining mode), 512
  - cell phones. *See also* mobile devices
    - cloning, 410
    - regulatory laws, 410–411
    - technologies, 411
    - vulnerabilities, 410
  - cell tower “spoofing”, 413–414
  - Cellebrite, 413–414
  - CER (crossover error rate), 166
  - certificate-based authentication, 375
  - certifications, 26
    - of ethical hackers, 20–21
  - CFAA (Computer Fraud and Abuse Act), 17
  - CFB (Cipher Feedback mode), 513
  - change point detection, 312
  - China, social credit system, 93
  - chosen cipher-text attacks, 532
  - chosen plain-text attacks, 532

- Chrootkit, 184
- CIA (confidentiality, integrity, and availability), 7
  - availability, 7
  - confidentiality, 7, 506
  - integrity, 7, 507
- CIPA (Children's Internet Protection Act), 33
- cipher text, 508
- cipher-text only attacks, 532
- circuit-level gateways, 478
- Citadel, 562
- clickjacking, 372
- client-side attacks, 296
- cloning, 410
- cloud computing, 550. *See also* IoT (Internet of Things)
  - access control, 552
  - advantages of using, 550
  - attacks, 554–555
  - audits, 552
  - breaches, 553
  - characteristics of, 550
  - data classification used by provider, 553
  - deployment models, 550–551
  - disaster recovery/business continuity plan of provider, 553
  - and encryption, 553
  - and fog computing, 556
  - IaaS, 551
  - IoT, 556
  - long-term viability of the provider, 553
  - PaaS, 551
  - regulatory requirements, 552
  - SaaS, 551
  - security, 555
  - separation of data, 553
  - SLA terms, 553
  - training of provider employees, 552
- cluster viruses, 214
- code of ethics, 29–30
- Code Red virus, 218
- code signing, 383–384
- Cohen, Fred, 216–217
- commands
  - attribute, 185
  - enable secret, 104
  - enum4linux, 161
  - finger, 161
  - Linux, 179
  - nbstat, 159
  - net, 156–157
  - netstat, 244
  - nmap, 125–127
    - decoy switch, 125
    - switches, 124–125
  - nslookup, 108–109
  - ntpdate, 162
  - ntpd, 162
  - ntpq, 162
  - ntprtrace, 162
  - passwd encryption, 489
  - rcpclient, 161
  - rpinfo, 161
  - showmount, 161
  - telnet, 134–135
- comments in source code, 351
- common ports and protocols, 117
- communication system testing, 22
- community cloud, 550
- company directories, 92–93
- compliance
  - with ISO/IEC 27002, 24–25
  - with laws, 24–25
  - regulations, 33–34
- Computer Fraud and Abuse Act, 32
- Conficker worm, 218
- confidentiality, 7, 23, 506

- Cookie Cadger, 301
  - cookie manipulation attacks, 348–349
  - cookies, 377
    - viewing, 377–378
  - Core Impact, 358
  - countermeasures
    - for botnets, 563–566
    - for malware, 243
    - for sniffing, 290–291
  - covering tracks, 51–52
  - crackers, 16, 18–19
  - crimeware kits, 562
  - CRL (Certificate revocation list), 526
  - Cross-Site Request Forgery attacks, 554
  - crypters, 229–230
  - cryptographic attacks, 531–532
  - cryptography, 7, 506
    - algorithms, 509
    - asymmetric encryption algorithms, 508–510, 515
      - Diffie-Hellman, 516
      - ECC, 516–517
      - ElGamal, 516
      - hashing, 517–518
      - RSA, 516
    - authentication, 506–507
    - cipher text, 508
    - encryption, 508
    - history of
      - Caesar’s cipher, 507–508
      - Navajo code talkers, 509
    - integrity, 507
    - plain text, 508
    - Scytale, 507
    - substitution cipher, 508
    - symmetric encryption algorithms, 508–511
      - AES, 514
      - DES, 511–513
      - Rivest Cipher, 514
  - CryptoTool, 536
  - CSMA/CD (carrier sense multiple access with collision detection), 424
  - CSRF (cross-site request forgery) attacks, 371–372
  - Cuckoo, 250–251
  - CurrPorts, 244
  - CVSS (Common Vulnerability Scoring System), 255–259
  - Cyber Security Enhancement Act, 32
  - cyberattacks, 9
  - cyberterrorists, 19
  - Cydia, 417
  - CypherX Crypter, 230
- D**
- DAI (Dynamic ARP Inspection), 290
  - Dark Reading, website, 29
  - data exfiltration, 412
  - data hiding Trojans, 221
  - data link layer, 56–57
    - sniffing, 276
  - databases. *See also* SQL (Structured Query Language)
    - hacking, 384–385
    - SQL, fingerprinting, 389–392
    - testing, 22
  - datagrams, fragmentation, 68–69
    - overlapping fragmentation attacks, 70
  - DDoS (distributed denial of service) attacks, 19, 309–310, 343. *See also* DoS (denial of service) attacks
    - countermeasures, 312–314
    - options, 303–304
    - tools, 310–312
  - deauthentication attacks, 429–430, 432–433
  - decoy switch, 125
  - default ports and services, 134
  - Dendroid, 416

- deny all* concept, 50
- DES (Data Encryption Standard), 511–513
  - modes of, 512–513
- detecting
  - honeypots, 493
  - malware, 249
  - sniffers, 291
- determining the network range, 112–113
  - malware, 243–246
  - traceroute, 114–115
- Device Administration API, 414–415
- DHCP (Dynamic Host Configuration Protocol), 61
- DHCP snooping, 283–284
- dictionary attacks, 176, 375
- differential backups, 11
- Diffie-Hellman, 516
- Dig, 111–112
- digital certificates, 524–525
  - PKI, 525–526
- digital signatures, 518
  - lack of code signing, 383–384
  - S/MIME, 529
  - steganography, 519–524
- digital watermark, 524
- directory traversal, 345–347
- disabling
  - SMI, 488–489
  - unnneeded services, 359
- disclosure of confidential information, 9
- disgruntled employees, 18
- distributed intelligence, 556
  - fog computing, 556
- distributing malware
  - crypters, 229–230
  - droppers, 229
  - packers, 229
  - wrappers, 228
- DMZ (demilitarized zone), 479
- DNS (Domain Name System), 62–63, 278
  - amplification attacks, 344
  - cache poisoning, 285
  - enumeration, 163
  - hijacking, 343–344
  - record types, 109
  - spoofing attacks, 285
  - structure of, 108
  - zone transfers, 110–111
- DNSSEC (DNS Security Extensions), 290
- DNSSEC (Domain Name System Security Extensions), 63
- domain names, registrar query, 104–107
- domain proxies, 107
- DOM-based XSS attacks, 367–368
- DoS (denial of service) attacks, 7, 9, 19, 343
  - application-level, 307–308
  - countermeasures, 312–314
  - DDoS, 309–310
  - ICMP, 306–307
  - options, 303–304
  - peer-to-peer, 307
  - permanent, 309
  - smurf, 307
  - SYN flood, 306
  - testing, 21
  - volumetric, 305–306
- DroidSheep, 416
- droppers, 229
- Dsniff, 289–290
- DSSS (Direct-sequence spread spectrum), 425
- due diligence, as reason for penetration testing, 25
- DumpSec, 157–158
- dumpster diving, 164
- dynamic analysis, 251–253
- dynamic ports, 60

**E**

- eavesdropping, 410
- e-banking Trojans, 221
- ECB (Electronic Code Book mode), 512
- ECC (Elliptic Curve Cryptography), 516–517
- Economic Espionage Act, 33
- EDGAR database, 98–99
- EF (exposure factor), 12
- EFS (Encrypted File System), 531
- egress filtering, 315, 564
- Electronic Communication Privacy Act, 32
- ElGamal, 516
- elicitation, 210–211
- email servers, gathering information about, 93
- employee and people searches, 95–98
  - websites, 95
- enable secret command, 104
- encoded binary IP addresses, 486–487
- encrypted passwords, 104
- encryption, 506, 508. *See also*
  - cryptography
    - asymmetric encryption algorithms
      - Diffie-Hellman, 516
      - ECC, 516–517
      - ElGamal, 516
      - hashing, 517–518
      - RSA, 516
    - basic, 374–375
    - in the cloud, 553
    - cracking tools, 536
    - digital certificates, 524–525
    - digital signatures, 518
      - steganography, 519–524
      - digital watermark, 524
      - tools, 521–524
    - successful cracks, 533
    - symmetric encryption algorithms
      - AES, 514
      - DES, 511–513
      - Rivest Cipher, 514
    - weak
      - Base64, 535
      - Uuencode, 535
      - XOR, 534–535
  - enum4linux command, 161
  - enumeration, 49–50, 152
    - DNS, 163
    - firewalls, 480–484
    - IPsec, 162–163
    - LDAP, 156–157
    - Linux/UNIX, 161
    - NetBIOS, 155
      - DumpSec, 157–158
      - Hyena, 158
    - NTP, 162
    - SMTP, 162
    - SNMP, 160
    - VoIP, 162–163
    - web servers, 337–341
    - Windows, 152
  - error checking, 171
  - error handling, improper, 352
  - escalation of privilege, 51
  - establishing, security testing goals, 26–27
  - ethical hacking, 16, 19–20
    - Andrew Auernheimer, 17
    - final reports, 28–29
    - modes of, 21–23
    - process, 52
    - required skills, 20–21
    - rules for, 22–23
    - scope of assessment, defining, 24
    - securing an organization, 52–53
    - test plans, 24–25
    - testing
      - approval process, 27–28
      - reasons for, 24–25
  - ethics, 29–30

Ettercap, 281, 300  
 plug-ins, 300–301  
 Evan's Debugger, 250  
 evasion tools, 473–474  
 evil twin attacks, 429  
 Exploit Database, website, 29, 49–50  
 exploits, 11, 169–170  
 of buffer overflows, 171–173  
 of C language, 172  
 of Java, 172  
 StickyKeys, 171  
 external penetration testing, 21  
 external vulnerability assessments, 254

## F

Facebook, 98  
 FaceNiff, 416  
 FakeToken, 416  
 Fall, Kevin, *TCP/IP Illustrated, Volume 1: The Protocols*, Second Edition, 69  
 false negatives, 462  
 false positives, 461, 472  
 FAR (false acceptance rate), 166  
 fast-flux botnets, 561  
 Federal Information Security Management Act, 32  
 federal laws, 30–33  
 18 USC 1029, 411  
 compliance with, 24–25  
 Electronic Communication Privacy Act, 32  
 relating to hacking, 31–33  
 Federal Sentencing Guidelines, 33  
 FHSS (Frequency-hopping spread spectrum), 425  
 file hiding, 185–186  
 file infections, 214  
 filters, Wireshark, 288–289  
 FIN scans, 119, 131  
 final preparation  
 hands-on activities, 573–574  
 suggested review and study plans, 574–575  
 final reports, 28–29  
 financial information, gathering, 98–99  
 finding open ports and access points  
 Hping, 129  
 nmap, 124–127  
 NSE, 125  
 port knocking, 129  
 SuperScan, 128  
 THC-Amap, 128–129  
 war driving, 130  
 finger command, 161  
 fingerprinting services, default ports and services, 134  
 Firesheep, 301  
 firewalking, 481–483  
 firewalls, 474–475  
 application-level gateways, 478  
 bypassing, 484–488  
 circuit-level gateways, 478  
 identifying, 480–484  
 NAT, 475–476  
 packet filters, 476–478  
 stateful inspection, 479–480  
 using Netcat to tunnel out through, 489–490  
 vulnerabilities, 479–480, 485–486  
 FISMA (Federal Information Security Management Act), 32  
 flag probes, 131  
 flags, TCP, 65  
 Flawfinder, 382  
 flooding, 470  
 FOCA, 102  
 fog computing, 556  
 footprinting and scanning, 48–49, 86, 90

- determining the network range,
    - 112–113
    - traceroute, 114–115
  - finding open ports and access points
    - Hping, 129
    - nmap, 124–127
    - port knocking, 129
    - SuperScan, 128
    - THC-Amap, 128–129
    - war driving, 130
  - identifying active machines, 115
  - information gathering
    - DNS enumeration, 107–112
    - documentation, 91
    - EDGAR database, 98–99
    - employee and people searches, 95–98
    - Google hacking, 99–103
    - job boards, 93–94
    - location information, 93
    - organization's website information, 91–93
    - registrar query, 104–107
    - Usenet, 103–104
  - mapping the network attack surface
    - automated mapping, 136–138
    - manual mapping, 136
  - OS fingerprinting, 130
    - active fingerprinting, 131–133
    - passive fingerprinting, 130–131
  - port scanning, 116–117
    - common ports and protocols, 117
    - TCP, 118–120
  - form grabber, 562
  - forms-based authentication, 375
  - FPipe, 240
  - FQDNs (fully qualified domain names), 62
  - fragmentation, 68–69
    - overlapping fragmentation attacks, 70
  - fraud
    - federal laws relating to, 31–33
    - sections of the U.S. Code relating to, 30–31
  - freeware, 224
  - FRR (false rejection rate), 166
  - FTP (File Transfer Protocol), 60–61
    - Trojans, 221
  - FTP bounce scans, 123
  - full backups, 11
  - Full Connect scans, 119
  - full-knowledge testing, 14
  - fuzzing, 383
- ## G
- gaining access, 50–51
  - Gardner, Bill, *Google Hacking for Penetration Testers*, 101
  - GDPR (General Data Protection Regulation), 24–25, 33
  - geolocation, 412–413
  - GFI LanGuard, 361
  - GHDB (Google Hacking Database), 101–102
  - Ghost Rat Trojan, 226
  - Gilmore, John, 533
  - GLBA (Gramm-Leach-Bliley Act), 24–25, 33
  - global threat correlation capabilities, 465
  - goals
    - of security, 7
    - of Trojans, 222–223
  - Google Earth, 93
  - Google hacking
    - search terms, 99
    - social security numbers, 100–103
  - GPS mapping, 443
  - gray box testing, 14
  - gray hat hackers, motivations, 17
  - GrayFish, 183
  - Green, Julian, 225

**H**

- hacker attacks, 9
- HackerStorm, website, 29
- HackerWatch, website, 29
- hacking, 16
  - black hat, motivations, 16
  - covering tracks, 51–52
  - cyberterrorists, 19
  - disgruntled employees, 18
  - escalation of privilege, 51
  - ethical, 19–20
    - required skills, 20–21
  - ethical hackers, process, 52
  - footprinting, 48–49
  - gaining access, 50–51
  - gray hat, motivations, 17
  - maintaining access, 51
  - methodology of, 17
  - motivations, 16–17
  - phreakers, 18
  - planting backdoors, 51–52
  - reconnaissance, 48–49
  - scanning and enumeration, 49–50
  - script kiddies, 18
  - social engineering, 49
  - suicide, motivations, 17
  - system, 19
    - and usability, 6
    - white hat, motivations, 16
- Hacking Web Applications (The Art of Hacking Series) LiveLessons*, 573–574
- hacktivists, 30, 305
- Hamster, 301
- hands-on activities, 573–574
- hard-coded credentials, 352
- hardening web servers, 358
- hardware, in DDoS attacks, 310
- hardware keyloggers, 241
- Hashcat, 536
- hashing, 517–518
- Heartbleed, 530
- heuristic scanning, 247
- heuristic-based analysis, 463
- hiding files and covering tracks, 185–186
- hierarchical database management system, 384
- hierarchical trust, 527–528
- high-interaction honeypots, 492
- hijacking
  - application layer, 295
    - client-side attacks, 296
    - man-in-the-browser attacks, 299
    - man-in-the-middle attacks, 296
    - predictable session token ID, 296
    - session fixation attacks, 299
    - session replay attacks, 299
    - session sniffing, 295–296
  - DNS, 343–344
  - preventing, 302–303
  - tools, 299–301
  - transport layer, 292–295
- Hikit, 226
- HIPAA (Health Insurance Portability and Accountability Act), 24–25, 33
- history
  - of cryptography
    - Caesar's cipher, 507–508
    - Navajo code talkers, 509
  - of viruses, 216–217
- HOIC, 311
- honeypots, 490–491
  - detecting, 493
  - placement, 491–492
  - types of, 492–493
- Horse Pill, 183
- hping, 74, 129, 480–481
- HTML (HyperText Markup Language), analyzing, 341

- HTTP (HyperText Transfer Protocol),
  - 60, 328–330
  - browsers, 330–332
  - clients, 328
  - cookies, 377
    - viewing, 377–378
  - proxies, 335
  - response splitting, 348
  - status code messages, 332
  - tunneling, 485
  - URLs, 332–333
- Hunt, 301
- hybrid attacks, 176, 376
- hybrid cloud, 550
- Hyena, 158
- I**
- IaaS (Infrastructure as a Service), 551
- IANA (Internet Assigned Numbers Authority), 104–105
- IBM AppScan, 361
- ICANN (Internet Corporation for Assigned Names and Numbers), 104–105
- IceSword, 244
- ICMP (Internet Control Message Protocol), 57, 66
  - embedded payloads, 234
  - header, 233–234
  - source routing, 72
  - traceroute, 72–74
    - example of in Windows, 73–74
  - tunneling, 233–235
  - Type 3 codes, 71
  - types and codes, 70
- ICMPSend, 238
- IDA Pro, 250
- identifying
  - active machines, 115
  - firewalls, 480–484
- IDP (intrusion detection prevention), 474
- IDS (intrusion detection systems), 49–50, 312, 458
  - anomaly-based analysis, 464–465
  - attack evasion techniques, 472–473
    - flooding, 470
    - insertion and evasion, 470
    - shellcode attacks, 471–472
  - components, 458
  - evasion tools, 473–474
  - false negatives, 462
  - false positives, 461, 472
  - heuristic-based analysis, 463
  - NIDS, 463
  - pattern matching, 461–464
  - protocol analysis, 463
  - protocol decoding, 462
  - signatures, 461, 463–464
  - Snort, 465–466
    - alerts, 468–470
    - keywords, 467
    - rules, 466–468
  - stateful pattern-matching recognition, 461
  - true/false matrix, 459
- ImageHide, 521
- impersonation, 210–211
- improper error handling, 352
- incident response plans, 15–16
- incremental backups, 11
- inference attacks, 531–532
- inference-based vulnerability assessments, 255
- information gathering, 21
  - DNS enumeration, 107–112
    - record types, 109
    - zone transfers, 110–111
  - documentation, 91
  - EDGAR database, 98–99
  - employee and people searches, 95–98
  - social networks, 97–98
  - websites, 95

- Google hacking
    - GHDB, 101
    - search terms, 101
  - job boards, 93–94
  - organization's website information, 91–93
  - registrar query, 104–107
  - Usenet, 103–104
  - infrastructure mode, 423–424
  - injection flaws, 362–363
  - insertion and evasion, 470
  - inSSIDer, 443
  - installing
    - botnets, 563
    - rogue access points, 428–429
  - INSTEON, 559
  - integrity, 7, 507
  - integrity checking, 247
  - intercepting web traffic, 380–381
  - internal penetration testing, 21
  - internal vulnerability assessments, 254
  - Internet layer
    - bypassing firewalls, 484
  - ICMP
    - embedded payloads, 234
    - header, 233–234
    - source routing, 72
    - traceroute, 72–74
    - tunneling via, 233–235
    - Type 3 codes, 71
    - types and codes, 70
  - IP, 67–70
  - interrogation, 210–211
  - iOS, 417
    - jailbreaking applications, 417
  - IoT (Internet of Things), 556
    - distributed intelligence, 556
    - hacking tools, 560
    - protocols, 558–559
    - security challenges, 556–557
  - IP (Internet Protocol), 66
    - IP forwarding, 280
    - iPhone, 417. *See also* iOS
    - IPID closed port, 122
    - IPID open port, 121
    - IPS (intrusion prevention systems), 458
      - anomaly-based analysis, 465
      - global threat correlation capabilities, 465
    - IPsec, 531
      - enumeration, 162–163
    - IPv4 addressing, 67–68
    - ISECOM (Institute for Security and Open Methodologies), OSSTMM, 23–24
    - ISN sampling, 131
    - ISO/IEC 27001:2013, 33
    - ISO/IEC 27002, compliance with, 24–25
- J**
- Jacobson, Van, 72
  - JAD (Java Application Descriptor) files, 418
  - jailbreaking, 413, 417
  - jamming wireless signals, 433
  - Java, exploits, 172
  - job boards, gathering information from, 93–94
  - Joe Sandbox, 250–251
  - John the Ripper, 177, 180–181, 536
  - Jumper, 226
- K**
- Kali Linux, 573
  - Kalman, Steve, *Web Security Field Guide*, 315
  - KARMA attacks, 441
  - KerbCrack, 168
  - Kerberos, 168, 175
  - KeyGhost Ltd, 169

- keylogging, 168–169
- keystroke loggers, 240–241
  - hardware, 241
  - software, 241
- Kimset, 447
- known plain-text attacks, 532
- Kocher, Paul, 533
- KoreK, 427
- KRACK attacks, 440–441
  
- L**
- LaBrea Tarpit, 493
- LAN Turtle, 529
- launching wireless attacks, 444
- laws, 30–33
  - compliance with, 24–25
  - and ethics, 29–30
  - federal, 31–33
  - regulatory, 33–34
    - PCI-DSS, 34
  - sections of the U.S. Code relating to
    - fraud, 30–31
- LDAP enumeration, 156–157
- legality of port scanning, 123
- Let Me Rule, 226
- Linux
  - /etc/passwd file, 178–180
  - authentication, 178–180
  - commands, 179
  - enumeration, 161
  - hiding files and covering tracks,
    - 181–182
  - nmap, 124
  - password cracking, 180–181
  - ping, 115
  - rootkits, 182–184
  - traceroute, 72
  - Whois, 105
- LLC (logical link control) layer,
  - 56–57
- LLMNR (Link-Local Multicast Name Resolution) protocol, 163
- LM authentication, 174–175
- load balancing, 312
- location information, gathering, 93
- location-based services, 412–413
- logging, 379
- LOIC, 311
- Loki, 237
- Long, Johnny, *Google Hacking for Penetration Testers*, 101
- LoRaWAN (Long Range Wide Area Network), 559
- LoriotPro, 114
- LoWPAN (IPv6 over Low Power Wireless Personal Area Networks), 559
- LRWPAN (Low Rate Wireless Personal Area Networks), 559
- LSASS (Local Security Authority Server Service), 155
  
- M**
- MAC (media access control) addresses,
  - 75
- MAC (media access control) layer, 56–57
- MAC flooding, 281–283
- macro infections, 214
- maintaining access, 51
- malvertising, 201–202
- malware, 9
  - analyzing, 249
    - dynamic analysis, 251–253
    - static analysis, 250–251
  - countermeasures, 243
  - detecting, 243–246, 249
  - distributing
    - crypters, 229–230
    - droppers, 229
    - packers, 229
    - wrappers, 228
  - keystroke loggers, 240–241
  - on mobile devices, 412
    - Android, 416

- ransomware, 230–231
  - WannaCry, 231
- spyware, 242
- Trojans, 220
  - covert communication, 232
  - distributing, 227–228
  - effects of, 224–225
  - goals of, 222–223
  - infection mechanism, 223–224
  - ports and communication methods, 221–222
  - RATs, 225–227
  - tools, 225–227
- viruses, 213
  - creation tools, 219–220
  - history of, 216–217
  - infection routine, 215
  - payloads, 215–216
  - search routine, 215
  - transmission methods, 213–215
  - well-known, 217–219
- worms, 213
  - transmission methods, 213–215
  - well-known, 217–219
- man-in-the-browser attacks, 299
- man-in-the-middle attacks, 280, 296, 347, 532
  - interceptions, 302
  - KARMA attacks, 441
  - Stingray device, 413–414
- mapping networks
  - attack surface
    - automated mapping, 136–138
    - manual mapping, 136
  - subnetting, 113
- master boot record infections, 214
- McAfee Rootkit Device, 184
- MD5, 375, 517–518
- Melissa virus, 217–218
- Mendax, 474
- Merdinger, Shawn, 103
- Metamorfo Banking Trojan, 562
- Metasploit, 357
- methodology
  - of hackers, 17
  - OSSTMM, 23–24
- Michael, 427
- Microsoft /GS, 382
- mirroring, 276. *See also* spanning
- misconfiguration, vulnerabilities in, 10
- misconfiguration attacks, 347–348
- mitigating, DDoS and DoS attacks, 312–314
- Mitnick, Kevin, 92–93
- mobile devices. *See also* wireless communication
  - Android, malware, 416
  - Blackberry, 418
  - bump attacks, 413
  - BYOD, 414
  - cell tower “spoofing”, 413–414
  - Cellebrite, 413–414
  - controls, 418–419
  - data exfiltration, 412
  - eavesdropping, 410
  - geolocation and location-based services, 412–413
  - iOS, 417
    - jailbreaking applications, 417
  - jailbreaking, 413
  - malware, 412
  - platforms, 413–414
    - Android, 414–415
  - security, 410, 412–413
  - Stingray device, 413–414
  - Windows Mobile Operating System, 417–418
- modes of ethical hackers, 21–23
- Mognet, 443
- Morphine, 229
- MoSucker, 227
- motivations, of hackers, 16–17

MTU (maximum transmission unit),  
datagram fragmentation, 68–69  
multicast MAC addresses, 75  
multipartite viruses, 214  
MyDoom virus, 218

## N

NAT (Network Address Translation), 67,  
475–476  
National Vulnerability Database, website,  
29  
natural disasters, as security threat, 9  
Navajo code talkers, 509  
NBS (National Bureau of Standards),  
511–512  
nbsat command, 159  
NDP (Network Discovery Protocol),  
67  
NeBIOS, enumeration, 155  
Necurs, 183  
Nessus, 260, 360, 474  
NetBIOS, enumeration  
DumpSec, 157–158  
Hyena, 158  
NetBus, 226  
Netcat, 74  
banner grabbing, 135–136  
for port redirection, 238–240  
using to tunnel out through a firewall,  
489–490  
Netcraft, 337–338  
Netsparker, 361  
netstat command, 244  
NetStumbler, 443  
network access layer, 74–75  
MAC addresses, 75  
network evaluations, 15  
network gear testing, 21  
Network layer, 56  
Network Performance Monitor, 160  
Nexpose, 260

NIDS (network-based intrusion detection  
systems), 463  
NIDSbench, 474  
Night Dragon Operation, 9  
Nikto, 383  
Nimda worm, 218  
NIST (National Institute of Standards  
and Technology), 511–512  
Special Publication 800–115, 53  
Special Publication (SP) 800–145,  
“The NIST Definition of Cloud  
Computing”, 550  
NLog, 137  
nmap, 133  
decoy switch, 125  
switches, 124–125  
no-knowledge testing, 13–14  
nonrepudiation, 507  
nontechnical password attacks, 164–165  
NRO (National Reconnaissance Office),  
98  
NSE (Nmap Scripting Engine), 125,  
339–340  
nslookup command, 108–109  
N-Stalker, 382  
NTLM authentication, 175  
NTP (Network Time Protocol),  
enumeration, 162  
ntpdate command, 162  
ntpd command, 162  
ntpq command, 162  
ntptrace command, 162  
NULL scans, 119

## O

Obad, 416  
obfuscated attacks, 378–379, 463–464,  
472  
OCTAVE (Operationally Critical  
Threat, Asset, and Vulnerability  
Evaluation), 53–54

- OFB (Output Feedback mode), 513
  - OFDM (Orthogonal frequency-division multiplexing), 425
  - OllyDBG, 250
  - OmniPeek, 289, 443
  - online pwned databases, 164
  - open authentication, 434–440
  - open port idle scans, 120–121
  - open services, finding, 134–136
  - OpenPuff, 522
  - OpenVAS, 260
  - operating systems, vulnerabilities in, 10
  - OphCrack, 177–178
  - OS fingerprinting
    - active fingerprinting, 131–133
    - finding open services, 134–136
    - fingerprinting services, default ports and services, 134
    - passive fingerprinting, 130–131
    - Winfingerprint, 133
  - Osborn, Mark, 461
  - OSI (Open Systems Interconnection)
    - model, 55–57
  - OSSTMM (Open Source Security Testing Methodology Manual), 23–24, 54
  - out-of-band SQL injection, 389, 394–395
  - overlapping fragmentation attacks, 70
  - owning the box, 173
- P**
- PaaS (Platform as a Service), 551
  - packers, 229
  - packet filters, 476–478
  - packets
    - TCP, 65
    - UDP, 66
  - Pandora, 311
  - parameter/form tampering, 362
  - partial-knowledge testing, 14
  - passing the hash, 168
  - passive fingerprinting, 130–131
  - passive sniffing, 276–277
  - passive vulnerability assessments, 253–254
  - passwd encryption command, 489
  - password cracking
    - John the Ripper, 177
    - Linux, 178–181
    - RainbowCrack technique, 177–178
    - types of attacks, 176
    - web server, 349
    - web-based, 375–377
    - Windows, 175–176
  - password guessing, 165–167
  - password sniffing, 167–168
  - patch management, 359
  - pattern matching, 461–464
    - stateful pattern-matching recognition, 461
  - PCI-DSS (Payment Card Industry Data Security Standard), 33–34
  - peer-to-peer attacks, 307
  - penetration testing, 15, 19
    - approval process, 27–28
    - due diligence as reason for, 25
    - external, 21
    - final reports, 28–29
    - internal, 21
    - test phases, establishing goals, 26–27
  - permanent DoS attacks, 309
  - PewDiePie printer hack, 11
  - PGMP (Pretty Good Malware Protection), 230
  - PGP (Pretty Good Privacy), 529
  - pharming, 200–201
  - Phatbot, 226
  - phishing attacks, 18, 200
  - phreakers, 18
  - physical layer, 57
  - physical security testing, 22
  - PII (personally identifiable information), 7

- Ping of Death, 307
  - ping sweeps, 115–116
  - PKI (Public Key Infrastructure), 525–526
  - placement of honeypots, 491–492
  - plain text, 508
  - planting backdoors, 51–52
  - poison apple attacks, 222
  - policies, developing, 52
  - Poodlebleed, 533
  - port knocking, 129
  - port redirection
    - FPipe, 240
    - Netcat, 238–240
  - port scanning
    - ACK scans, 122
    - common ports and protocols, 117
    - FTP bounce scans, 123
    - Hping, 129
    - legality of, 123
    - nmap, 124–127
    - RPC scans, 123
    - SuperScan, 128
    - TCP, 118–120
      - shutdown, 118–119
      - three-way handshake, 118
    - window scans, 123
  - port security, 283
  - ports, 60–61. *See also* scanning
    - blocking, 61
    - spanning, 276
    - Trojans, 221–222
    - well-known, 117
    - Windows, 155
  - PPTP (Point-to-Point Tunneling Protocol), 531
  - predictable session token ID, 296
  - preferred network lists, attacking, 433
  - PremiumSMS, 416
  - prependers, 215
  - Presentation layer, 56
  - pretexting, 211
  - preventing, session hijacking, 302–303
  - principle of least privilege, 61
  - private cloud, 550
  - privilege escalation, 51, 169
  - Process Explorer, 252
  - Process Monitor, 244
  - Process Viewer, 244
  - protocol analysis, 302, 463
  - protocol-decoding IDS, 462
  - protocols
    - EFS, 531
    - IoT, 558–559
    - IPsec, 531
    - PGP, 529
    - PPTP, 531
    - S/MIME, 529
    - SSH, 530
    - SSL, 530
  - proxies, 137, 335
  - proxy trojans, 221
  - public cloud, 550
  - PwnageTool, 417
- ## Q
- qualitative risk assessment, 12
  - Qualys, 260
  - quantitative risk assessment, 12
  - Queso, 132
- ## R
- RA (Registration authority), 526
  - race conditions, 352–353
  - RainbowCrack, 177–178
  - ransomware, 219, 230–231
    - WannaCry, 231
  - Rapid7, 260
  - RATs (remote-access Trojans), 225–227
  - RATS (Rough Auditing Tool for Security), 382

rcpcclient command, 161  
 Recon Dog, 102  
 reconnaissance, 48–49  
 records, DNS, 63, 109  
 RedSn0w, 417  
 redundant array of inexpensive disks  
   (RAID), 7  
 reflected XSS, 364–366  
 registered ports, 60  
 registrar query, 104–107  
 Regshot, 244  
 regulatory laws, 33–34  
   for cell phones, 410–411  
   for cloud computing, 552  
   compliance with, 24–25  
   PCI-DSS, 34  
 remote-access Trojans, 220–221  
 replay attacks, 532  
 required skills for ethical hackers, 20–21  
 researching, vulnerabilities, 29  
 RESTful APIs, 557–558  
 Restorator, 230  
 Retina CS, 361  
 Reverse WWW Tunneling Shell, 238  
 RFID (radio-frequency identification)  
   attacks, 422  
 RIDs (relative identifiers), 153–154  
 RIRs (Regional Internet Registries),  
   104–105  
 risk, 8. *See also* risk assessment  
   backing up data to reduce, 10–11  
 risk assessment, 12–13  
 Rivest Cipher, 514  
 RMF (Risk Management Framework), 8  
 robust wireless authentication, 446  
 Roesch, Martin, 465  
 rogue access points, installing, 428–429  
 rooting, 416  
 RootKitRevealer, 184  
 rootkits, 51, 182–184  
 RPC (Remote Procedure Call), 161

RPC scans, 123  
 rpinfocmd command, 161  
 RSA (Rivest, Shamir, Adelman), 516  
 RSA NetWitness, 289  
 rules, for ethical hackers, 22–23  
 rusers, 161  
 rwho, 161  
 Ryan, Thomas, 98

## S

SaaS (Software as a Service), 551  
 Sage, Robin, 98  
 SAM (Security Accounts Manager), 154  
 sandbox, 413  
 SANS  
   policy templates, 16, 37  
   Reading Room, website, 29  
 Sarbanes-Oxley (SOX), 33  
 Sasser worm, 218  
 scanning, 49–50, 86  
   heuristic, 463  
   web servers, 336  
 scoring systems, for vulnerability  
   assessments, 255–259  
 script kiddies, 18  
 Scytale, 507  
 search terms, Google, 99  
 security. *See also* security policies; security  
   testing  
   assets, 8  
   challenges for IoT, 556–557  
   CIA triad, 7  
     availability, 7  
     confidentiality, 7  
     integrity, 7  
   for cloud computing, 555  
   confidentiality, 7  
   crackers, 16  
   exploits, 11  
   goals of, 7  
   hackers, 16

- cyberterrorists, 19
- disgruntled employees, 18
- ethical, 19–20
- methodology of, 17
- phreakers, 18
- script kiddies, 18
- software crackers, 18–19
- system hackers/crackers, 19
- mobile devices, 410, 412–413
  - controls, 418–419
- risk, 8
  - backing up data to reduce, 10–11
- risk assessment, 12–13
- threats, 8–9
- vulnerabilities, 9–10
  - researching, 29
- Windows, 154–155
- world’s biggest data breaches as of December 2018, 7
- Security and Exchange Commission,
  - EDGAR database, 98–99
- security policies, 15–16
  - incident response plans, 15–16
- security testing, 13
  - final reports, 28–29
  - full-knowledge testing, 14
  - high-level assessments, 15
  - network evaluations, 15
  - no-knowledge testing, 13–14
  - partial-knowledge testing, 14
  - penetration testing, 15
    - external, 21
    - internal, 21
  - physical, 22
  - resources, 53
  - test phases, 25–26
    - establishing goals, 26–27
  - types of tests, 14–16
- Security Tracker, website, 29
- SecurityFocus, website, 29
- security-software disablers, 221
- session hijacking, 56
  - application layer, 295
    - client-side attacks, 296
    - man-in-the-browser attacks, 299
    - man-in-the-middle attacks, 296
    - predictable session token ID, 296
    - session fixation attacks, 299
    - session replay attacks, 299
    - session sniffing, 295–296
  - preventing, 302–303
  - tools, 299–301
  - transport layer, 292–295
- Session layer, 56
- session replay attacks, 299
- session sniffing, 295–296
- session riding, 554
- Session Thief, 301
- SET (Social Engineering Toolkit),
  - 204–209
- SHA-1, 518
- shellcode attacks, 471–472
- Shellshock, 101
- Shodan, 102–103
- shoulder surfing, 165, 212–213
- showmount command, 161
- shrinkwrap software, vulnerabilities in, 10
- side-channel attacks, 532, 554
- SIDs (security identifiers), 153
- signatures, 461, 463–464
  - heuristic-based, 463
- signature-scanning antivirus programs,
  - 247
- single-authority trust, 527
- site surveys, 445
- skills, of ethical hackers, 20–21
- Slammer virus, 218
- SLAs (service-level agreements), for
  - cloud computing, 553
- SLE (single loss expectancy),
  - determining, 12
- Slowloris, 308

- SMAC, 284
- SMB (Server Message Block), 155
- SMI (Smart Install) protocol, disabling, 488–489
- S/MIME (Secure/Multipurpose Internet Mail Extensions), 529
- SMS phishing, 209
- SMTP (Simple Mail Transfer Protocol), 60, 62
  - enumeration, 162
- smurf attacks, 307
- Sn0wbreeze, 417
- sniffing, 51, 58–59, 276
  - active, 276–277
  - countermeasures, 290–291
  - detecting, 291
  - FaceNiff, 416
  - passive, 276–277
  - session hijacking, 291–292
  - Snort, 465–466
    - alerts, 468–470
    - keywords, 467
    - rules, 466–468
  - tools, 289–290
  - Wireshark, 286–289
    - filters, 288–289
- SNMP (Simple Network Monitoring Protocol), 62
  - enumeration, 160
  - traps, 160
- snmpwalk, 160
- Snort, 465–466
  - alerts, 468–470
  - keywords, 467
  - rules, 466–468
- SNScan, 160
- SOA records, 109
- social activism, hacktivists, 30
- social credit system, 93
- social engineering, 22, 49, 165, 199
  - elicitation, 210–211
  - impersonation, 210–211
  - interrogation, 210–211
  - malvertising, 201–202
  - motivation techniques, 212
  - pharming, 200–201
  - phishing, 200
  - shoulder surfing, 212–213
  - SMS phishing, 209
  - spear phishing, 202–203
  - techniques, 199–200
  - USB key drop, 212–213
  - voice phishing, 210
  - whaling, 210
- social networks
  - dangers of, 98
  - gathering information from, 97
- social security numbers, gathering, 100–103
- software
  - antivirus, 246–248
  - crackers, 18–19
  - in DDoS attacks, 310
  - vulnerabilities in, 10
- source code, comments in, 351
- source routing, 72
- SOX (Sarbanes-Oxley), 24–25
- spanning, 276
- Special Publication 800–115, 53
- spoofing, 56
- spoofing attacks, countermeasures, 290–291
- spread-spectrum technology, 425
- spyware, 213, 240–242
- SQL (Structured Query Language)
  - databases, fingerprinting, 389–392
  - injection attacks, 387–391
    - mitigations, 396–397
  - out-of-band exploitation, 394–395
  - stored procedure, 396
  - time-delay technique, 396
  - use of Booleans, 394

- injection hacking tools, 397–398
  - statements, 385–387
    - UNION exploitation attack, 392–393
  - SQL injection, 554
  - Squert, 468
  - SRI (Sub-resource Integrity), 384
  - SSH (Secure Shell), 530
  - SSID (service set ID), 424
  - SSL (Secure Sockets Layer), 530
  - SSLstrip, 301
  - StackGuard, 382
  - state laws, compliance with, 24–25
  - stateful inspection, 479–480
  - stateful pattern-matching recognition, 461
  - static analysis, 250–251
  - status code messages, HTTP, 332
  - steganography, 519–524
    - digital watermark, 524
    - tools, 521–524
  - Stevens, Richard, *TCP/IP Illustrated, Volume 1: The Protocols*, Second Edition, 69
  - StickyKeys, 171
  - Stingray device, 413–414
  - stolen equipment attacks, 22
  - S-Tools, 521
  - Stored DOM-based attacks, 348–349
  - stored procedure SQL injection, 396
  - stored XSS attacks, 366–367
  - Storm worm, 218
  - stream ciphers, 512
  - subnetting, 113
  - substitution cipher, 508
  - suggested review and study plans, 574–575
  - suicide hackers, motivations, 17
  - Super Bluetooth Hack, 421
  - switches
    - bypassing, 277
    - with ARP poisoning, 279–281
    - with MAC flooding, 281–283
  - nmap, 124–125
  - symmetric encryption algorithms, 506, 508–511
    - AES, 514
    - DES, 511–513
    - Rivest Cipher, 514
  - SYN flood attacks, 306, 565
  - SYN scans, 119
  - system hacking, 19
    - cracking Windows passwords, 175–176
    - exploiting vulnerabilities, 169–170
      - applications, 170–171
      - buffer overflows, 171–173
    - file hiding, 185–186
    - nontechnical password attacks, 164–165
    - owning the box, 173
    - privilege escalation, 169
    - technical password attacks, 165
      - password guessing, 165–167
- ## T
- Talos File Reputation Online Tool, 248–249
  - Tamper IE, 301
  - TAN grabber, 562
  - Task Manager, 244
  - TCP (Transmission Control Protocol), 56, 64–66
    - flags, 65, 118
    - open port idle scans, 120–121
    - shutdown, 118–119
    - three-way handshake, 118
    - tunneling, 236–237
  - TCPdump, 290
  - TCP/IP (Transmission Control Protocol/Internet Protocol), 57
    - application layer, 59–60
    - DHCP, 61

- DNS, 62–63
- FTP, 61
- ports, 60–61
- SMTP, 62
- SNMP, 62
- Telnet, 62
- and ARP, 278–279
- Internet layer
  - ICMP, 70–74
  - IP, 67–70
- network access layer, 74–75
  - ARP, 76
  - MAC addresses, 75
- port-scanning techniques, 119–120
- Transport layer
  - TCP, 64–66
  - UDP, 66
- TCPView, 244
- TCSEC (Trusted Computer System Evaluation Criteria), 232
- Teardrop attacks, 307
- technical password attacks
  - automated password guessing, 167
  - keylogging, 228–230
  - password guessing, 165–167
  - password sniffing, 167–168
- Teflon Oil Patch, 230
- Telnet, 58, 62, 483
- telnet command, 134–135
- Tenable, 260
- testing
  - penetration testing, due diligence as
    - reason for, 25
    - reasons for, 24–25
- TFN (Tribal Flood Network), 311
- TFTP (Trivial FTP), bypassing firewalls, 487–488
- THC-Amap, 129
- THC-Hydra, 376, 536
- THC-Wardrive, 443
- TheHackerGiraffe, 11
- ThreatExpert, 250–251
- threats, 8–9
- three-way handshake, 118
- throttling, 313
- time-delay SQL injection technique, 396
- Tini, 225
- TKIP (Temporal Key Integrity Protocol), 427
- ToE (target of evaluation), 13
- traceback, 565
- traceroute, 114–115, 480
  - example of in Windows, 73–74
- traffic-cleaning, 565
- training, 53
  - of cloud provider employees, 552
- Tramp.A, 416
- transmission methods, of viruses and worms, 213–215
- Transport layer, 56
  - TCP, 64–66
  - tunneling via, 236–237
  - UDP, 66
- transport layer
  - bypassing firewalls, 484–485
  - hijacking, 292–295
- traps, 160
- tree-based vulnerability assessments, 255
- Trend Micro RootkitBuster, 184
- Trinoo, 311
- Trojan Man, 230
- Trojans, 220
  - covert communication, 232
  - distributing, 227–228
  - effects of, 224–225
  - goals of, 222–223
  - infection mechanism, 223–224
  - Obad, 416
  - ports and communication methods, 221–222
  - RATs, 225–227
  - tools, 225–227

- types of, 220–221
- trust models
  - hierarchical trust, 527–528
  - single-authority trust, 527
  - web of trust, 528–529
- TShark, 289
- tumbling, 410
- tunneling
  - ICMP, 233–235
  - TCP, 236–237
  - via application layer, 237–238
- Type 3 codes, 71

## U

- UDP (User Datagram Protocol), 56, 57, 66
- UEFI (Unified Extensible Firmware Interface), 417
- Ufasoft Snif, 281
- UI redress attacks, 372
- UID (user identifier), 415
- unicast MAC addresses, 75
- UNIX, enumeration, 161
- unprotected APIs, 353–356
- unvalidated input, 362
- UPX, 250
- URLs, 332–333
  - obfuscating, 378–379
- U.S. Code, sections relating to fraud, 30–31
- USA PATRIOT Act, 32
- usability, and security, 6
- USB key drop, 212–213
- Usenet, 103–104
- user mode, 152–153
- Uencode, 535

## V

- viruses, 9, 213
  - creation tools, 219–220
  - history of, 216–217

- infection routine, 215
- payloads, 215–216
- search routine, 215
- transmission methods, 213–215
- well-known, 217–219
- VisualRoute, 115
- voice phishing, 210
- VoIP (Voice over IP), enumeration, 162–163
- volumetric attacks, 305–306
- VPNs (virtual private networks), 507
- vulnerabilities, 9–10
  - researching, 29
  - scanning for, 259–260
  - of web servers, 342, 349–351
- XSS, 363–364
- vulnerability assessments
  - CVSS, 255–259
  - external *vs.* internal, 254
  - inference-based, 255
  - passive *vs.* active, 253–254
  - scoring systems, 255–259
  - tree-based, 255
- vulnerability scanners, 50

## W

- W3AF, 382
- WannaCry, 231
- war driving, 130
- watering holes, 224
- WaveStumbler, 443
- Wayback Machine, 92
- weak encryption
  - Base64, 535
  - Uencode, 535
  - XOR, 534–535
- web application hacking, 361
  - buffer overflows, 373–374
  - clickjacking, 372
  - cookies, 377
    - viewing, 377–378
  - CSRF attacks, 371–372

- DOM-based XSS attacks, 367–368
- injection flaws, 362–363
- intercepting web traffic, 380–381
- logging, 379
- parameter/form tampering, 362
- password cracking, 375–377
- reflected XSS, 364–366
- securing web applications, 381–383
- stored XSS, 366–367
- unvalidated input, 362
- URL obfuscation, 378–379
- XSS evasion techniques, 368–369
- XSS mitigations, 369–370
- XSS vulnerabilities, 363–364
- web of trust, 528–529
- web server hacking, 328
  - attacks, 335–336, 342–343
    - cookie manipulation, 348–349
    - directory traversal, 345–347
    - DoS/DDoS, 343
    - HTTP response splitting, 348
    - man-in-the-middle, 347
    - misconfiguration, 347–348
    - website defacement, 347
  - audits, 360
  - automated exploit tools
    - BeEF, 357
    - Canvas, 358
    - Core Impact, 358
    - Metasploit, 357
  - disabling unneeded services, 359
  - DNS hijacking, 343–344
  - enumeration, 337–341
  - HTML, analyzing, 341
  - HTTP, 328–330
    - browsers, 330–332
    - clients, 328
    - proxies, 335
    - status code messages, 332
    - URLs, 332–333
  - locking down the file system, 360
  - password cracking, 349
  - patch management, 359
  - scanning web servers, 336
  - vulnerabilities, 349–351
    - comments in source code, 351
    - hard-coded credentials, 352
    - hidden elements, 356
    - improper error handling, 352
    - lack of code signing, 356
    - race conditions, 352–353
    - unprotected APIs, 353–356
  - vulnerability scans, 360–361
  - web server vulnerability identification, 342
- WebCracker, 376
- WebInspect, 383
- websites
  - codes of ethics, 29–30
  - defacement, 347
  - for employee and people searches, 95
  - Exploit Database, 49–50
  - ISECOM, 23
  - for researching vulnerabilities, 29
- well-known ports, 60, 117
- well-known viruses and worms, 217–219
- WEP (Wired Equivalent Privacy), 407, 425–427
  - attacking, 433–435
- whaling, 210
- white box testing, 14
- white hat hackers, 16
  - motivations, 16
- Whois, 105–107
- Wikto, 340
- WinARPAAttacker, 281
- WinDNSSpoofer, 285
- window scans, 123
- Windows
  - architecture, 153–154
  - authentication types, 173–175
  - cracking passwords, 175–176
  - enumeration, 152
  - LDAP enumeration, 156–157

- NeBIOS enumeration, 155
- nmap, 124
- owning the box, 173
- ports, 155
- RIDs, 153–154
- security, 154–155
- SIDs, 153
- SmartWhois, 105–106
- traceroute, 72–74
- user mode, 152–153
- Windows Mobile Operating System, 417–418
- WinDump, 290
- Winfingerprint, 133
- wireless communication, 406–407
  - Bluetooth, 419
    - classifications of, 419
    - technologies, 419–420
    - vulnerabilities, 420–421
  - cell phone technologies, 411–412
  - GPS mapping, 443
  - launching wireless attacks, 444
  - mobile devices, security concerns, 412–413
  - RFID attacks, 422
  - spread-spectrum technology, 425
  - SSID, 424
  - wireless hacking tools, 443
  - wireless traffic analysis, 443–444
- WLANs, 422
  - ad hoc mode, 423
  - airmon-ng tool, 430
  - airodump-ng tool, 431
  - attacking preferred network lists, 433
  - compromising the Wi-Fi network, 444–445
  - deauthentication attacks, 429–430, 432–433
  - evil twin attacks, 429
  - fragmentation attacks, 441–442
  - infrastructure mode, 423–424
  - installing rogue access points, 428–429
  - jamming wireless signals, 433
  - KARMA attacks, 441
  - KRACK attacks, 440–441
  - misuse detection, 447
  - open authentication, 434–440
  - robust wireless authentication, 446
  - security, WEP, 425–427
  - site surveys, 445
  - spread-spectrum technology, 425
  - war driving, 433
  - WEP, attacking, 433–435
  - installing rogue access points, 428–429
  - jamming wireless signals, 433
  - KARMA attacks, 441
  - KRACK attacks, 440–441
  - misuse detection, 447
  - open authentication, 434–440
  - robust wireless authentication, 446
  - site surveys, 445
  - war driving, 433
  - WEP, 425–427
  - WEP, attacking, 433–435
  - WPA, attacking, 435–440
  - WPS, attacking, 441
- wireless networks, testing, 21
- Wireshark, 59, 286–289, 564
  - filters, 288–289
  - ping capture, 235
- WLANs (wireless LANs), 422
  - ad hoc mode, 423
  - airmon-ng tool, 430
  - airodump-ng tool, 431
  - attacking preferred network lists, 433
  - compromising the Wi-Fi network, 444–445
  - deauthentication attacks, 429–430, 432–433
  - evil twin attacks, 429
  - fragmentation attacks, 441–442
  - infrastructure mode, 423–424
  - installing rogue access points, 428–429
  - jamming wireless signals, 433
  - KARMA attacks, 441
  - KRACK attacks, 440–441
  - misuse detection, 447
  - open authentication, 434–440
  - robust wireless authentication, 446
  - security, WEP, 425–427
  - site surveys, 445
  - spread-spectrum technology, 425
  - war driving, 433
  - WEP, attacking, 433–435

- wireless hacking tools, 443
- WPA, 427
  - attacking, 435–440
- WPS, attacking, 441
- world's biggest data breaches as of  
December 2018, 7
- worms, 213
  - transmission methods, 213–215
  - well-known, 217–219
- WPA (Wi-Fi Protected Access), 427
  - attacking, 435–440
- WPA3, 428
- WPS (Wi-Fi Protected Setup), attacking,  
441
- wrappers, 228
- wrapping attacks, 555
- WRP (Windows Resource Protection), 7

**X**

- X.507, 525
- XMAS scans, 120
- XOR, 426, 534–535
  - basic authentication, 374
- Xprobe, 133

- XSS (cross-site scripting), 554
  - CSRF attacks, 371–372
  - DOM-based attacks, 367–368
  - evasion techniques, 368–369
  - mitigations, 369–370
  - reflected, 364–366
  - stored, 366–367
  - vulnerabilities, 363–364

## **Y**

- Yahoo Boys, 18
- Yarochkin, Fyodor, 124
- Yoda's Crypter, 229

## **Z**

- Zabasearch, 96
- Zenmap, 127
- Zeroaccess, 183
- Zigbee, 558
- Zombam.B, 227
- zombie computers, 560–561
- zone files, 63
- Z-Wave, 558–559