

# CONTENTS IN DETAIL

<b>ACKNOWLEDGMENTS</b>	<b>xvii</b>
------------------------	-------------

<b>INTRODUCTION</b>	<b>xix</b>
---------------------	------------

What Is in This Book . . . . .	xx
The Scripting Exercises . . . . .	xxi
How to Use This Book . . . . .	xxii

<b>1</b>	
<b>BASH BASICS</b>	<b>1</b>

Environmental Setup . . . . .	2
Accessing the Bash Shell . . . . .	2
Installing a Text Editor . . . . .	2
Exploring the Shell . . . . .	3
Checking Environment Variables . . . . .	3
Running Linux Commands . . . . .	4
Elements of a Bash Script . . . . .	6
The Shebang Line . . . . .	6
Comments . . . . .	7
Commands . . . . .	8
Execution . . . . .	8
Debugging . . . . .	9
Basic Syntax . . . . .	10
Variables . . . . .	10
Arithmetic Operators . . . . .	13
Arrays . . . . .	14
Streams . . . . .	15
Control Operators . . . . .	16
Redirection Operators . . . . .	18
Positional Arguments . . . . .	20
Input Prompting . . . . .	22
Exit Codes . . . . .	23
<b>Exercise 1: Recording Your Name and the Date . . . . .</b>	<b>25</b>
Summary . . . . .	25

<b>2</b>	
<b>FLOW CONTROL AND TEXT PROCESSING</b>	<b>27</b>

Test Operators . . . . .	27
if Conditions . . . . .	29
Linking Conditions . . . . .	31
Testing Command Success . . . . .	32
Checking Subsequent Conditions . . . . .	32
Functions . . . . .	33
Returning Values . . . . .	34
Accepting Arguments . . . . .	34

Loops and Loop Controls . . . . .	35
while . . . . .	35
until . . . . .	37
for . . . . .	38
break and continue . . . . .	40
case Statements . . . . .	41
Text Processing and Parsing . . . . .	42
Filtering with grep . . . . .	42
Filtering with awk . . . . .	43
Editing Streams with sed . . . . .	44
Job Control . . . . .	45
Managing the Background and Foreground . . . . .	46
Keeping Jobs Running After Logout . . . . .	46
Bash Customizations for Penetration Testers . . . . .	47
Placing Scripts in Searchable Paths . . . . .	47
Shortening Commands with Aliases . . . . .	48
Customizing the ~/.bashrc Profile . . . . .	48
Importing Custom Scripts . . . . .	49
Capturing Terminal Session Activity . . . . .	49
<b>Exercise 2: Pinging a Domain . . . . .</b>	<b>50</b>
Summary . . . . .	50

### 3

## **SETTING UP A HACKING LAB 51**

Security Lab Precautions . . . . .	52
Installing Kali . . . . .	52
The Target Environment . . . . .	54
Installing Docker and Docker Compose . . . . .	54
Cloning the Book's Repository . . . . .	55
Deploying Docker Containers . . . . .	56
Testing and Verifying the Containers . . . . .	57
The Network Architecture . . . . .	57
The Public Network . . . . .	58
The Corporate Network . . . . .	58
Kali Network Interfaces . . . . .	58
The Machines . . . . .	59
Managing the Lab . . . . .	60
Shutting Down . . . . .	60
Removing . . . . .	60
Rebuilding . . . . .	60
Accessing Individual Lab Machines . . . . .	61
Installing Additional Hacking Tools . . . . .	61
WhatWeb . . . . .	61
RustScan . . . . .	62
Nuclei . . . . .	62
dirsearch . . . . .	63
Linux Exploit Suggester 2 . . . . .	63
Gitjacker . . . . .	64
pwncat . . . . .	64
LinEnum . . . . .	65
unix-privesc-check . . . . .	66

Assigning Aliases to Hacking Tools. . . . .	66
Summary . . . . .	67

**4**  
**RECONNAISSANCE** **69**

Creating Reusable Target Lists . . . . .	70
Consecutive IP Addresses. . . . .	70
Possible Subdomains. . . . .	71
Host Discovery. . . . .	73
ping . . . . .	73
Nmap . . . . .	75
arp-scan. . . . .	75
<b>Exercise 3: Receiving Alerts About New Hosts</b> . . . . .	<b>76</b>
Port Scanning . . . . .	78
Nmap . . . . .	78
RustScan . . . . .	80
Netcat . . . . .	81
<b>Exercise 4: Organizing Scan Results</b> . . . . .	<b>81</b>
Detecting New Open Ports . . . . .	83
Banner Grabbing . . . . .	85
Using Active Banner Grabbing . . . . .	86
Detecting HTTP Responses . . . . .	87
Using Nmap Scripts . . . . .	89
Detecting Operating Systems . . . . .	90
Analyzing Websites and JSON. . . . .	92
Summary . . . . .	94

**5**  
**VULNERABILITY SCANNING AND FUZZING** **95**

Scanning Websites with Nikto . . . . .	95
Building a Directory Indexing Scanner. . . . .	97
Identifying Suspicious robots.txt Entries . . . . .	98
<b>Exercise 5: Exploring Non-indexed Endpoints</b> . . . . .	<b>100</b>
Brute-Forcing Directories with dirsearch. . . . .	100
Exploring Git Repositories . . . . .	102
Cloning the Repository. . . . .	102
Viewing Commits with git log . . . . .	102
Filtering git log Information. . . . .	103
Inspecting Repository Files . . . . .	104
Vulnerability Scanning with Nuclei . . . . .	105
Understanding Templates . . . . .	105
Writing a Custom Template . . . . .	106
Applying the Template . . . . .	107
Running a Full Scan. . . . .	107
<b>Exercise 6: Parsing Nuclei's Findings</b> . . . . .	<b>111</b>
Fuzzing for Hidden Files . . . . .	112
Creating a Wordlist of Possible Filenames . . . . .	112
Fuzzing with ffuf . . . . .	113
Fuzzing with Wfuzz . . . . .	113

Assessing SSH Servers with Nmap’s Scripting Engine . . . . .	114
<b>Exercise 7: Combining Tools to Find FTP Issues . . . . .</b>	<b>115</b>
Summary . . . . .	116

## **6**

### **GAINING A WEB SHELL 117**

Arbitrary File Upload Vulnerabilities . . . . .	118
Fuzzing for Arbitrary File Uploads . . . . .	119
Bypassing File Upload Controls . . . . .	121
Uploading Files with Burp Suite . . . . .	125
Staging Web Shells . . . . .	128
Finding Directory Traversal Vulnerabilities . . . . .	129
Uploading Malicious Payloads . . . . .	130
Executing Web Shell Commands . . . . .	132
<b>Exercise 8: Building a Web Shell Interface . . . . .</b>	<b>133</b>
Limitations of Web Shells . . . . .	134
Lack of Persistence . . . . .	134
Lack of Real-Time Responses . . . . .	134
Limited Functionality . . . . .	134
OS Command Injection . . . . .	135
<b>Exercise 9: Building a Command Injection Interface . . . . .</b>	<b>138</b>
Bypassing Command Injection Restrictions . . . . .	139
Obfuscation and Encoding . . . . .	139
Globbing . . . . .	140
Summary . . . . .	141

## **7**

### **REVERSE SHELLS 143**

How Reverse Shells Work . . . . .	144
Ingress vs. Egress Controls . . . . .	144
Shell Payloads and Listeners . . . . .	144
The Communication Sequence . . . . .	145
Executing a Connection . . . . .	146
Setting Up a Netcat Listener . . . . .	146
Crafting a Payload . . . . .	146
Delivering and Initializing the Payload . . . . .	147
Executing Commands . . . . .	148
Listening with pwncat . . . . .	149
Bypassing Security Controls . . . . .	150
Encrypting and Encapsulating Traffic . . . . .	151
Alternating Between Destination Ports . . . . .	152
Spawning TTY Shells with Pseudo-terminal Devices . . . . .	154
Python’s pty Module . . . . .	154
socat . . . . .	155
Post-exploitation Binary Staging . . . . .	155
Serving Netcat . . . . .	156
Uploading Files with pwncat . . . . .	157
Downloading Binaries from Trusted Sites . . . . .	157
<b>Exercise 10: Maintaining a Continuous Reverse Shell Connection . . . . .</b>	<b>158</b>

Initial Access with Brute Force . . . . .	159
<b>Exercise 11: Brute-Forcing an SSH Server . . . . .</b>	<b>160</b>
Summary . . . . .	162

## **8 LOCAL INFORMATION GATHERING 163**

The Filesystem Hierarchy Standard . . . . .	164
The Shell Environment . . . . .	165
Environment Variables . . . . .	165
Sensitive Information in Bash Profiles . . . . .	165
Users and Groups . . . . .	166
Local Accounts . . . . .	166
Local Groups . . . . .	167
Home Folder Access . . . . .	168
Valid Shells . . . . .	169
Processes . . . . .	170
Viewing Process Files . . . . .	170
Running ps . . . . .	172
Examining Root Processes . . . . .	173
The Operating System . . . . .	173
<b>Exercise 12: Writing a Linux Operating System Detection Script . . . . .</b>	<b>174</b>
Login Sessions and User Activity . . . . .	174
Collecting User Sessions . . . . .	174
Investigating Executed Commands . . . . .	175
Networking . . . . .	175
Network Interfaces and Routes . . . . .	176
Connections and Neighbors . . . . .	179
Firewall Rules . . . . .	180
Network Interface Configuration Files . . . . .	181
Domain Resolvers . . . . .	181
Software Installations . . . . .	182
Storage . . . . .	183
Block Devices . . . . .	184
The Filesystem Tab File . . . . .	186
Logs . . . . .	186
System Logs . . . . .	187
Application Logs . . . . .	187
<b>Exercise 13: Recursively Searching for Readable Logfiles . . . . .</b>	<b>188</b>
Kernels and Bootloaders . . . . .	188
Configuration Files . . . . .	189
Scheduled Tasks . . . . .	191
Cron . . . . .	191
At . . . . .	193
<b>Exercise 14: Writing a Cron Job Script to Find Credentials . . . . .</b>	<b>194</b>
Hardware . . . . .	194
Virtualization . . . . .	196
Using Dedicated Tools . . . . .	196
Living Off the Land . . . . .	197
Automating Information Gathering with LinEnum . . . . .	197
<b>Exercise 15: Adding Custom Functionality to LinEnum . . . . .</b>	<b>198</b>
Summary . . . . .	199

## 9

### PRIVILEGE ESCALATION

201

What Is Privilege Escalation? . . . . .	201
Linux File and Directory Permissions . . . . .	202
Viewing Permissions . . . . .	202
Setting Permissions . . . . .	203
Creating File Access Control Lists . . . . .	204
Viewing SetUID and SetGID . . . . .	205
Setting the Sticky Bit . . . . .	206
Finding Files Based on Permissions . . . . .	207
Exploiting a SetUID Misconfiguration . . . . .	208
Scavenging for Credentials . . . . .	210
Passwords and Secrets . . . . .	210
Private Keys . . . . .	212
<b>Exercise 16: Brute-Forcing GnuPG Key Passphrases . . . . .</b>	<b>215</b>
Examining the sudo Configuration . . . . .	216
Abusing Text Editor Tricks . . . . .	218
Downloading Malicious sudoers Files . . . . .	219
Hijacking Executables via PATH Misconfigurations . . . . .	220
<b>Exercise 17: Maliciously Modifying a Cron Job . . . . .</b>	<b>222</b>
Finding Kernel Exploits . . . . .	224
SearchSploit . . . . .	225
Linux Exploit Suggester 2 . . . . .	225
Attacking Adjacent Accounts . . . . .	226
Privilege Escalation with GTFOBins . . . . .	228
<b>Exercise 18: Mapping GTFOBins Exploits to Local Binaries . . . . .</b>	<b>229</b>
Automating Privilege Escalation . . . . .	229
LinEnum . . . . .	229
unix-privesc-check . . . . .	230
MimiPenguin . . . . .	230
Linuxprivchecker . . . . .	231
Bashark . . . . .	231
Summary . . . . .	231

## 10

### PERSISTENCE

233

The Enemies of Persistent Access . . . . .	234
Modifying Service Configurations . . . . .	234
System V . . . . .	235
systemd . . . . .	237
Hooking into Pluggable Authentication Modules . . . . .	238
<b>Exercise 19: Coding a Malicious pam_exec Bash Script . . . . .</b>	<b>238</b>
Generating Rogue SSH Keys . . . . .	239
Repurposing Default System Accounts . . . . .	240
Poisoning Bash Environment Files . . . . .	241
<b>Exercise 20: Intercepting Data via Profile Tampering . . . . .</b>	<b>243</b>
Credential Theft . . . . .	245
Hooking a Text Editor . . . . .	245
Streaming Executed Commands . . . . .	247
Forging a Not-So-Innocent sudo . . . . .	249
<b>Exercise 21: Hijacking Password Utilities . . . . .</b>	<b>251</b>

Distributing Malicious Packages . . . . .	251
Understanding DEB Packages . . . . .	252
Packaging Innocent Software . . . . .	253
Converting Package Formats with alien . . . . .	254
<b>Exercise 22: Writing a Malicious Package Installer . . . . .</b>	<b>254</b>
Summary . . . . .	256

## 11

### **NETWORK PROBING AND LATERAL MOVEMENT 257**

Probing the Corporate Network . . . . .	258
Service Mapping . . . . .	258
Port Frequencies . . . . .	260
<b>Exercise 23: Scanning Ports Based on Frequencies . . . . .</b>	<b>261</b>
Exploiting Cron Scripts on Shared Volumes . . . . .	263
Verifying Exploitability . . . . .	264
Checking the User Context . . . . .	265
<b>Exercise 24: Gaining a Reverse Shell on the Backup Server . . . . .</b>	<b>265</b>
Exploiting a Database Server . . . . .	266
Port Forwarding . . . . .	266
Brute-Forcing with Medusa . . . . .	267
Backdooring WordPress . . . . .	268
Running SQL Commands with Bash . . . . .	270
<b>Exercise 25: Executing Shell Commands via WordPress . . . . .</b>	<b>271</b>
Compromising a Redis Server . . . . .	271
Raw CLI Commands . . . . .	272
Metasploit . . . . .	273
Exposed Database Files . . . . .	275
Dumping Sensitive Information . . . . .	277
Uploading a Web Shell with SQL . . . . .	278
Summary . . . . .	279

## 12

### **DEFENSE EVASION AND EXFILTRATION 281**

Defensive Controls . . . . .	281
Endpoint Security . . . . .	282
Application and API Security . . . . .	283
Network Security . . . . .	284
Honeypots . . . . .	284
Log Collection and Aggregation . . . . .	285
<b>Exercise 26: Auditing Hosts for Landmines . . . . .</b>	<b>285</b>
Concealing Malicious Processes . . . . .	286
Library Preloading . . . . .	286
Process Hiding . . . . .	288
Process Masquerading . . . . .	289
<b>Exercise 27: Rotating Process Names . . . . .</b>	<b>290</b>
Dropping Files in Shared Memory . . . . .	292
Disabling Runtime Security Controls . . . . .	292
Manipulating History . . . . .	294
Tampering with Session Metadata . . . . .	295
Concealing Data . . . . .	296

Encoding . . . . .	297
Encryption . . . . .	298
<b>Exercise 28: Writing Substitution Cipher Functions . . . . .</b>	<b>299</b>
Exfiltration . . . . .	300
Raw TCP . . . . .	300
DNS . . . . .	301
Text Storage Sites . . . . .	302
Slack Webhooks . . . . .	303
Sharding Files . . . . .	304
Number of Lines . . . . .	304
Size . . . . .	304
Chunks . . . . .	305
<b>Exercise 29: Sharding and Scheduling Exfiltration . . . . .</b>	<b>305</b>
Summary . . . . .	306

**INDEX** **307**