

CONTENTS IN DETAIL

FOREWORD BY JACOB SOO	xvii
------------------------------	-------------

FOREWORD BY SHUBHAM SHAH, AKA “SHUBS”	xix
--	------------

INTRODUCTION	xxi
---------------------	------------

Who Should Read This Book and Why	xxii
What This Book Is About	xxii
Source Code and Online Resources	xxiv
Further Reading	xxv

0	
DAY ZERO	1

What Is a Vulnerability?	2
Common Vulnerabilities and Exposures Records	3
Bugs vs. Vulnerabilities	3
What Is Zero-Day Vulnerability Research?	4
Vulnerability Research vs. Penetration Testing	5
Disciplines and Techniques	6
Selecting a Vulnerability Research Target	8
Familiarity	8
Availability	8
Impact	9
Where to Explore Projects	10
Summary	10

PART I

CODE REVIEW

1	
TAINT ANALYSIS	13

A Buffer Overflow Example	14
Triggering the Buffer Overflow	16
Applying Taint Analysis	18

Sink-to-Source Analysis	20
Choosing the Right Sinks	20
Filtering for Exploitable Scenarios	22
Confirming Exploitability	24
Identifying an Attacker-Controlled Source	26
Confirming a Reachable Attack Surface	29
Testing the Exploit	29
Building the Proof of Concept	34
Summary	37

2

MAPPING CODE TO ATTACK SURFACE 39

The Internet	40
Web Client Vulnerabilities	40
Web Server Vulnerabilities	43
Network Protocols	50
Data Structures	52
Procedures	53
Local Attack Surface	55
Files in Inter-Process Communication	56
Sockets	61
Named Pipes	63
Other IPC Methods	65
File Formats	66
Type-Length-Value	67
Directory-Based	69
Custom Fields	70
Summary	71

3

AUTOMATED VARIANT ANALYSIS 73

Abstract Syntax Trees	74
Static Code Analysis Tools	77
CodeQL	77
Simgrep	84
Variant Analysis	87
Single-Repository Variant Analysis	87
Multi-Repository Variant Analysis	101
Summary	103

PART II REVERSE ENGINEERING

4 BINARY TAXONOMY 107

Beyond Executable Binaries and Shared Libraries	108
Scripts	109
Reverse Engineering Node.js Electron Applications	109
Reverse Engineering a Python Application	122
Intermediate Representations	126
Common Language Runtime Assemblies	127
Java Bytecode	131
Machine Code	137
Statically Linked	139
Dynamically Linked	140
Stripped	141
Packed	142
Summary	143

5 SOURCE AND SINK DISCOVERY 145

Static Analysis	146
Dumping Strings	147
Disassembling and Decompiling with Ghidra	148
Dynamic Analysis	155
Tracing Library and System Calls	156
Analyzing Library Function Calls in ImageMagick	158
Instrumenting Functions with Frida	161
Monitoring Higher-Level Events	165
Evaluating Exploitability	167
Analyzing Errors	167
Using Canary Strings	168
Examining Inter-Process Communication Artifacts	169
Summary	169

6 HYBRID ANALYSIS IN REVERSE ENGINEERING 171

Code Coverage	172
Applying Code Coverage for Compiled Binary Analysis	172
Visualizing Code Coverage with Lighthouse	175
Emulation	178
Emulating Firmware with Qiling	178
Hijacking API Calls	185
Binding Virtual Paths	187

Symbolic Analysis	189
Performing Symbolic Execution	191
Solving Constraints	193
Writing SimProcedures	195
Summary	199

PART III FUZZING

7		
QUICK AND DIRTY FUZZING		203
Why Fuzzing Works	204	
Fuzzing Criteria and Approaches	204	
Target Information	205	
Generation Approach	205	
Input Type	206	
Feedback Loop	206	
Black-Box Fuzzing with boofuzz	207	
Introduction to boofuzz	207	
Exploring the MQTT Protocol	208	
Fuzzing the MQTT Protocol	209	
Fuzzing the MQTT PUBLISH Packet	212	
Fuzzing NanoMQ	214	
Zero-Setup Mutation-Based Fuzzing with Radamsa	219	
Fuzzing libxls	220	
Analyzing Fuzz Coverage with OSS-Fuzz	222	
Bootstrapped Fuzzing	223	
Summary	229	

8		
COVERAGE-GUIDED FUZZING		231
Advantages of Coverage-Guided Fuzzing	232	
Fuzzing with AFL++	234	
Fuzzing Optimizations	238	
Patching Validation Checks	238	
Minimizing the Seed Corpus	243	
Writing a Harness	246	
Fuzzing in Parallel	248	
Measuring Fuzzing Coverage with afl-cov	248	
Fuzz Introspector	250	
Identifying Fuzz Blockers	252	
Analyzing Function Complexity	253	
Summary	255	

9		
FUZZING EVERYTHING		257
Closed Source Binaries	258	
QEMU Mode	258	
Frida Mode	259	
Managed Memory Binaries	262	
Jazzer	263	
Go Fuzzing	268	
Syntactic and Semantic Targets	273	
Dictionaries	274	
Grammars	278	
Intermediate Representations	280	
Summary	281	
10		
BEYOND DAY ZERO		283
Coordinated Vulnerability Disclosure	284	
Hunting Bug Bounties	285	
Writing Vulnerability Reports	287	
Disclosing Vulnerabilities	291	
Assigning a CVE	292	
Securing Organizations with Vulnerability Research	294	
In the Software Development Life Cycle	295	
Through Product Security Assessments	296	
Summary	296	
INDEX		299