

A Bug Hunter's Diary

A Guided Tour Through the Wilds
of Software Security

TOBIAS KLEIN



no starch
press

San Francisco

CONTENTS IN DETAIL

ACKNOWLEDGMENTS

xi

INTRODUCTION

1

The Goals of This Book	1
Who Should Read the Book	1
Disclaimer	2
Resources	2

CHAPTER 1: BUG HUNTING

3

1.1 For Fun and Profit	4
1.2 Common Techniques	4
My Preferred Techniques	4
Potentially Vulnerable Code Locations	5
Fuzzing	5
Further Reading	5
1.3 Memory Errors	6
1.4 Tools of the Trade	6
Debuggers	6
Disassemblers	7
1.5 EIP = 41414141	7
1.6 Final Note	8

CHAPTER 2: BACK TO THE '90S

9

2.1 Vulnerability Discovery	10
Step 1: Generate a List of the Demuxers of VLC	10
Step 2: Identify the Input Data	11
Step 3: Trace the Input Data	11
2.2 Exploitation	12
Step 1: Find a Sample TiVo Movie File	13
Step 2: Find a Code Path to Reach the Vulnerable Code	13
Step 3: Manipulate the TiVo Movie File to Crash VLC	16
Step 4: Manipulate the TiVo Movie File to Gain Control of EIP	17
2.3 Vulnerability Remediation	18
2.4 Lessons Learned	22
2.5 Addendum	22

CHAPTER 3: ESCAPE FROM THE WWW ZONE	25
3.1 Vulnerability Discovery	25
Step 1: List the IOCTLs of the Kernel	26
Step 2: Identify the Input Data.	27
Step 3: Trace the Input Data.	28
3.2 Exploitation.	35
Step 1: Trigger the NULL Pointer Dereference for a Denial of Service.	35
Step 2: Use the Zero Page to Get Control over EIP/RIP	39
3.3 Vulnerability Remediation	48
3.4 Lessons Learned.	49
3.5 Addendum	49
CHAPTER 4: NULL POINTER FTW	51
4.1 Vulnerability Discovery	52
Step 1: List the Demuxers of FFmpeg	52
Step 2: Identify the Input Data.	52
Step 3: Trace the Input Data.	53
4.2 Exploitation.	56
Step 1: Find a Sample 4X Movie File with a Valid strk Chunk	57
Step 2: Learn About the Layout of the strk Chunk.	57
Step 3: Manipulate the strk Chunk to Crash FFmpeg	58
Step 4: Manipulate the strk Chunk to Gain Control over EIP	61
4.3 Vulnerability Remediation	66
4.4 Lessons Learned.	69
4.5 Addendum	69
CHAPTER 5: BROWSE AND YOU'RE OWNED	71
5.1 Vulnerability Discovery	71
Step 1: List the Registered WebEx Objects and Exported Methods	72
Step 2: Test the Exported Methods in the Browser	74
Step 3: Find the Object Methods in the Binary	76
Step 4: Find the User-Controlled Input Values	78
Step 5: Reverse Engineer the Object Methods.	79
5.2 Exploitation.	82
5.3 Vulnerability Remediation	84
5.4 Lessons Learned.	84
5.5 Addendum	84
CHAPTER 6: ONE KERNEL TO RULE THEM ALL	87
6.1 Vulnerability Discovery	88
Step 1: Prepare a VMware Guest for Kernel Debugging	88
Step 2: Generate a List of the Drivers and Device Objects Created by avast!	88
Step 3: Check the Device Security Settings.	90
Step 4: List the IOCTLs.	90
Step 5: Find the User-Controlled Input Values	97
Step 6: Reverse Engineer the IOCTL Handler	99

6.2	Exploitation	103
6.3	Vulnerability Remediation	110
6.4	Lessons Learned	110
6.5	Addendum	110

CHAPTER 7: A BUG OLDER THAN 4.4BSD 113

7.1	Vulnerability Discovery	114
	Step 1: List the IOCTLs of the Kernel	114
	Step 2: Identify the Input Data	114
	Step 3: Trace the Input Data	116
7.2	Exploitation	119
	Step 1: Trigger the Bug to Crash the System (Denial of Service)	119
	Step 2: Prepare a Kernel-Debugging Environment	121
	Step 3: Connect the Debugger to the Target System	121
	Step 4: Get Control over EIP	123
7.3	Vulnerability Remediation	129
7.4	Lessons Learned	130
7.5	Addendum	130

CHAPTER 8: THE RINGTONE MASSACRE 133

8.1	Vulnerability Discovery	133
	Step 1: Research the iPhone’s Audio Capabilities	134
	Step 2: Build a Simple Fuzzer and Fuzz the Phone	134
8.2	Crash Analysis and Exploitation	140
8.3	Vulnerability Remediation	147
8.4	Lessons Learned	147
8.5	Addendum	147

APPENDIX A: HINTS FOR HUNTING 149

A.1	Stack Buffer Overflows	149
	Example: Stack Buffer Overflow Under Linux	151
	Example: Stack Buffer Overflow Under Windows	152
A.2	NULL Pointer Dereferences	153
A.3	Type Conversions in C	154
A.4	GOT Overwrites	157

APPENDIX B: DEBUGGING 163

B.1	The Solaris Modular Debugger (mdb)	163
	Starting and Stopping mdb	163
	General Commands	164
	Breakpoints	164
	Running the Debuggee	164
	Examining Data	164
	Information Commands	165
	Other Commands	165