

CONTENTS IN DETAIL

FOREWORD by Dr. Jared DeMott	xv
-------------------------------------	-----------

ACKNOWLEDGMENTS	xvii
------------------------	-------------

INTRODUCTION	xix
---------------------	------------

Prerequisites for the Reader	xx
A Brief Game Hacking History	xx
Why Hack Games?	xxi
How This Book Is Organized	xxii
About the Online Resources	xxiv
How to Use This Book	xxiv

PART 1 TOOLS OF THE TRADE

1 SCANNING MEMORY USING CHEAT ENGINE	3
---	----------

Why Memory Scanners Are Important	4
Basic Memory Scanning	4
Cheat Engine's Memory Scanner	5
Scan Types	6
Running Your First Scan	6
Next Scans	7
When You Can't Get a Single Result	7
Cheat Tables	7
Memory Modification in Games	8
Manual Modification with Cheat Engine	8
Trainer Generator	9
Pointer Scanning	11
Pointer Chains	11
Pointer Scanning Basics	12
Pointer Scanning with Cheat Engine	14
Pointer Rescanning	17
Lua Scripting Environment	18
Searching for Assembly Patterns	19
Searching for Strings	21
Closing Thoughts	22

2 DEBUGGING GAMES WITH OLLYDBG 23

A Brief Look at OllyDbg's User Interface	24
OllyDbg's CPU Window	26
Viewing and Navigating a Game's Assembly Code	27
Viewing and Editing Register Contents.	29
Viewing and Searching a Game's Memory	29
Viewing a Game's Call Stack	30
Creating Code Patches	31
Tracing Through Assembly Code	32
OllyDbg's Expression Engine.	33
Using Expressions in Breakpoints	34
Using Operators in the Expression Engine	34
Working with Basic Expression Elements	35
Accessing Memory Contents with Expressions	36
OllyDbg Expressions in Action.	36
Pausing Execution When a Specific Player's Name Is Printed	37
Pausing Execution When Your Character's Health Drops	39
OllyDbg Plug-ins for Game Hackers	42
Copying Assembly Code with Asm2Clipboard	42
Adding Cheat Engine to OllyDbg with Cheat Utility.	42
Controlling OllyDbg Through the Command Line.	43
Visualizing Control Flow with OllyFlow	45
Closing Thoughts	47

3 RECONNAISSANCE WITH PROCESS MONITOR AND PROCESS EXPLORER 49

Process Monitor	50
Logging In-Game Events.	50
Inspecting Events in the Process Monitor Log.	52
Debugging a Game to Collect More Data	53
Process Explorer.	55
Process Explorer's User Interface and Controls	56
Examining Process Properties	57
Handle Manipulation Options.	59
Closing Thoughts	61

PART 2 GAME DISSECTION

4 FROM CODE TO MEMORY: A GENERAL PRIMER 65

How Variables and Other Data Manifest in Memory	66
Numeric Data.	67
String Data.	69
Data Structures	71

Unions	73
Classes and VF Tables	74
x86 Assembly Crash Course	78
Command Syntax	79
Processor Registers	81
The Call Stack	86
Important x86 Instructions for Game Hacking	89
Closing Thoughts	96

5 ADVANCED MEMORY FORENSICS 97

Advanced Memory Scanning	98
Deducing Purpose	98
Finding the Player’s Health with OllyDbg	99
Determining New Addresses After Game Updates	101
Identifying Complex Structures in Game Data	105
The std::string Class	105
The std::vector Class	108
The std::list Class	110
The std::map Class	114
Closing Thoughts	118

6 READING FROM AND WRITING TO GAME MEMORY 119

Obtaining the Game’s Process Identifier	120
Obtaining Process Handles	121
Working with OpenProcess()	121
Accessing Memory	122
Working with ReadProcessMemory() and WriteProcessMemory()	122
Accessing a Value in Memory with ReadProcessMemory() and WriteProcessMemory()	123
Writing Templated Memory Access Functions	123
Memory Protection	124
Differentiating x86 Windows Memory Protection Attributes	125
Changing Memory Protection	126
Address Space Layout Randomization	128
Disabling ASLR to Simplify Bot Development	128
Bypassing ASLR in Production	128
Closing Thoughts	130

PART 3 PROCESS PUPPETEERING

7 CODE INJECTION 133

Injecting Code Caves with Thread Injection	134
Creating an Assembly Code Cave	134
Translating the Assembly to Shellcode	135

Writing the Code Cave to Memory	136
Using Thread Injection to Execute the Code Cave	137
Hijacking a Game’s Main Thread to Execute Code Caves	138
Building the Assembly Code Cave.	138
Generating Skeleton Shellcode and Allocating Memory.	140
Finding and Freezing the Main Thread	141
Injecting DLLs for Full Control.	142
Tricking a Process into Loading Your DLL	143
Accessing Memory in an Injected DLL	145
Bypassing ASLR in an Injected DLL.	146
Closing Thoughts	147

8 MANIPULATING CONTROL FLOW IN A GAME 149

NOPing to Remove Unwanted Code	150
When to NOP	150
How to NOP	151
Hooking to Redirect Game Execution	153
Call Hooking	153
VF Table Hooking	156
IAT Hooking	160
Jump Hooking.	165
Applying Call Hooks to Adobe AIR	169
Accessing the RTMP Goldmine	169
Hooking the RTMPS encode() Function.	171
Hooking the RTMPS decode() Function.	172
Placing the Hooks	173
Applying Jump Hooks and VF Hooks to Direct3D.	175
The Drawing Loop.	176
Finding the Direct3D Device	177
Writing a Hook for EndScene().	182
Writing a Hook for Reset()	183
What’s Next?	184
Closing Thoughts	185

PART 4 CREATING BOTS

9 USING EXTRASENSORY PERCEPTION TO WARD OFF FOG OF WAR 189

Background Knowledge	190
Revealing Hidden Details with Lighthacks	190
Adding a Central Ambient Light Source	190
Increasing the Absolute Ambient Light	191
Creating Other Types of Lighthacks	192

Revealing Sneaky Enemies with Wallhacks	192
Rendering with Z-Buffering	193
Creating a Direct3D Wallhack	194
Fingerprinting the Model You Want to Reveal.	196
Getting a Wider Field of Vision with Zoomhacks	197
Using NOPing Zoomhacks	197
Scratching the Surface of Hooking Zoomhacks	198
Displaying Hidden Data with HUDs	198
Creating an Experience HUD	199
Using Hooks to Locate Data	200
An Overview of Other ESP Hacks	201
Closing Thoughts	202

10

RESPONSIVE HACKS	203
Observing Game Events	204
Monitoring Memory	204
Detecting Visual Cues	205
Intercepting Network Traffic	206
Performing In-Game Actions	211
Emulating the Keyboard.	211
Sending Packets	215
Tying the Pieces Together	218
Making the Perfect Healer	218
Resisting Enemy Crowd-Control Attacks	218
Avoiding Wasted Mana	219
Closing Thoughts	219

11

PUTTING IT ALL TOGETHER: WRITING AUTONOMOUS BOTS	221
Control Theory and Game Hacking	222
State Machines	223
Combining Control Theory and State Machines	225
A Basic Healer State Machine	225
A Complex Hypothetical State Machine.	228
Error Correction	230
Pathfinding with Search Algorithms	232
Two Common Search Techniques	233
How Obstacles Disrupt Searches.	233
An A* Search Algorithm	234
When A* Searches Are Particularly Useful.	240
Common and Cool Automated Hacks	241
Looting with Cavebots	241
Automating Combat with Warbots	243
Closing Thoughts	244

12		
STAYING HIDDEN		245
Prominent Anti-Cheat Software		246
The PunkBuster Toolkit		246
Signature-Based Detection		246
Screenshots		247
Hash Validation		247
The ESEA Anti-Cheat Toolkit		247
The VAC Toolkit		247
DNS Cache Scans		248
Binary Validation		248
False Positives		248
The GameGuard Toolkit		248
User-Mode Rootkit		248
Kernel-Mode Rootkit		249
The Warden Toolkit		249
Carefully Managing a Bot's Footprint		250
Minimizing a Bot's Footprint		250
Masking Your Footprint		251
Teaching a Bot to Detect Debuggers		251
Anti-Debugging Techniques		255
Defeating Signature-Based Detection		256
Defeating Screenshots		258
Defeating Binary Validation		259
Defeating an Anti-Cheat Rootkit		261
Defeating Heuristics		262
Closing Thoughts		263
INDEX		265