CONTENTS IN DETAIL

FOREWORD by Dave Aitel xiii

ACKNOWLEDGMENTS xvii

INTRODUCTION xix

1
SETTING UP YOUR DEVELOPMENT ENVIRONMENT 1

1.1 Operating System Requirements 2
1.2 Obtaining and Installing Python 2.5 2
    1.2.1 Installing Python on Windows 2
    1.2.2 Installing Python for Linux 3
1.3 Setting Up Eclipse and PyDev 4
    1.3.1 The Hacker's Best Friend: ctypes 5
    1.3.2 Using Dynamic Libraries 6
    1.3.3 Constructing C Datatypes 8
    1.3.4 Passing Parameters by Reference 9
    1.3.5 Defining Structures and Unions 9

2
DEBUGGERS AND DEBUGGER DESIGN 13

2.1 General-Purpose CPU Registers 14
2.2 The Stack 16
2.3 Debug Events 18
2.4 Breakpoints 18
    2.4.1 Soft Breakpoints 19
    2.4.2 Hardware Breakpoints 21
    2.4.3 Memory Breakpoints 23

3
BUILDING A WINDOWS DEBUGGER 25

3.1 Debuggee, Where Art Thou? 25
3.2 Obtaining CPU Register State 33
    3.2.1 Thread Enumeration 33
    3.2.2 Putting It All Together 35
3.3 Implementing Debug Event Handlers 39
3.4 The Almighty Breakpoint 43
    3.4.1 Soft Breakpoints 43
    3.4.2 Hardware Breakpoints 47
    3.4.3 Memory Breakpoints 52
3.5 Conclusion 55

Gray Hat Python
(C) 2009 by Justin Seitz

4
PYDBG—A PURE PYTHON WINDOWS DEBUGGER 57

4.1 Extending Breakpoint Handlers 58
4.2 Access Violation Handlers 60
4.3 Process Snapshots 63
  4.3.1 Obtaining Process Snapshots 63
  4.3.2 Putting It All Together 65

5
IMMUNITY DEBUGGER—THE BEST OF BOTH WORLDS 69

5.1 Installing Immunity Debugger 70
5.2 Immunity Debugger 101 70
  5.2.1 PyCommands 71
  5.2.2 PyHooks 71
5.3 Exploit Development 73
  5.3.1 Finding Exploit-Friendly Instructions 73
  5.3.2 Bad-Character Filtering 75
  5.3.3 Bypassing DEP on Windows 77
5.4 Defeating Anti-Debugging Routines in Malware 81
  5.4.1 IsDebuggerPresent 81
  5.4.2 Defeating Process Iteration 82

6
HOOKING 85

6.1 Soft Hooking with PyDbg 86
6.2 Hard Hooking with Immunity Debugger 90

7
DLL AND CODE INJECTION 97

7.1 Remote Thread Creation 98
  7.1.1 DLL Injection 99
  7.1.2 Code Injection 101
7.2 Getting Evil 104
  7.2.1 File Hiding 104
  7.2.2 Coding the Backdoor 105
  7.2.3 Compiling with py2exe 108

8
FUZZING 111

8.1 Bug Classes 112
  8.1.1 Buffer Overflows 112
  8.1.2 Integer Overflows 113
  8.1.3 Format String Attacks 114
8.2 File Fuzzer 115
8.3 Future Considerations 122
  8.3.1 Code Coverage 122
  8.3.2 Automated Static Analysis 122

Gray Hat Python
(C) 2009 by Justin Seitz

9
SULLEY 123

9.1 Sulley Installation 124
9.2 Sulley Primitives 125
 9.2.1 Strings 125
 9.2.2 Delimiters 125
 9.2.3 Static and Random Primitives 126
 9.2.4 Binary Data 126
 9.2.5 Integers 126
 9.2.6 Blocks and Groups 127
9.3 Slaying WarFTPD with Sulley 129
 9.3.1 FTP 101 129
 9.3.2 Creating the FTP Protocol Skeleton 130
 9.3.3 Sulley Sessions 131
 9.3.4 Network and Process Monitoring 132
 9.3.5 Fuzzing and the Sulley Web Interface 133

10
FUZZING WINDOWS DRIVERS 137

10.1 Driver Communication 138
10.2 Driver Fuzzing with Immunity Debugger 139
10.3 Driverlib—The Static Analysis Tool for Drivers 142
 10.3.1 Discovering Device Names 143
 10.3.2 Finding the IOCTL Dispatch Routine 144
 10.3.3 Determining Supported IOCTL Codes 145
10.4 Building a Driver Fuzzer 147

11
IDAPYTHON—SCRIPTING IDA PRO 153

11.1 IDAPython Installation 154
11.2 IDAPython Functions 155
 11.2.1 Utility Functions 155
 11.2.2 Segments 155
 11.2.3 Functions 156
 11.2.4 Cross-References 156
 11.2.5 Debugger Hooks 157
11.3 Example Scripts 158
 11.3.1 Finding Dangerous Function Cross-References 158
 11.3.2 Function Code Coverage 160
 11.3.3 Calculating Stack Size 161

12
PYEMU—THE SCRIPTABLE EMULATOR 163

12.1 Installing PyEmu 164
12.2 PyEmu Overview 164
 12.2.1 PyCPU 164
 12.2.2 PyMemory 165
 12.2.3 PyEmu 165

Gray Hat Python
(C) 2009 by Justin Seitz

12.2.4 Execution 165
12.2.5 Memory and Register Modifiers 165
12.2.6 Handlers 166
12.3 IDAPyEmu 171
12.3.1 Function Emulation 172
12.3.2 PEPyEmu 175
12.3.3 Executable Packers 176
12.3.4 UPX Packer 176
12.3.5 Unpacking UPX with PEPyEmu 177

INDEX **183**

Gray Hat Python
(C) 2009 by Justin Seitz