

Contents

Introduction	xv
On The Book's DVD	xxiii
1 Anonymizing Your Activities	1
<i>Recipe 1-1: Anonymous Web Browsing with Tor.</i>	3
<i>Recipe 1-2: Wrapping Wget and Network Clients with Torsocks</i>	5
<i>Recipe 1-3: Multi-platform Tor-enabled Downloader in Python</i>	7
<i>Recipe 1-4: Forwarding Traffic through Open Proxies</i>	12
<i>Recipe 1-5: Using SSH Tunnels to Proxy Connections</i>	16
<i>Recipe 1-6: Privacy-enhanced Web browsing with Privoxy</i>	18
<i>Recipe 1-7: Anonymous Surfing with Anonymouse.org.</i>	20
<i>Recipe 1-8: Internet Access through Cellular Networks</i>	21
<i>Recipe 1-9: Using VPNs with Anonymizer Universal</i>	23
2 Honey pots	27
<i>Recipe 2-1: Collecting Malware Samples with Nepenthes.</i>	29
<i>Recipe 2-2: Real-Time Attack Monitoring with IRC Logging</i>	32
<i>Recipe 2-3: Accepting Nepenthes Submissions over HTTP with Python.</i>	34
<i>Recipe 2-4: Collecting Malware Samples with Dionaea</i>	37
<i>Recipe 2-5: Accepting Dionaea Submissions over HTTP with Python</i>	40
<i>Recipe 2-6: Real-time Event Notification and Binary Sharing with XMPP</i>	41
<i>Recipe 2-7: Analyzing and Replaying Attacks Logged by Dionea.</i>	43
<i>Recipe 2-8: Passive Identification of Remote Systems with p0f.</i>	44
<i>Recipe 2-9: Graphing Dionaea Attack Patterns with SQLite and Gnuplot</i>	46
3 Malware Classification	51
<i>Recipe 3-1: Examining Existing ClamAV Signatures</i>	52
<i>Recipe 3-2: Creating a Custom ClamAV Database.</i>	54
<i>Recipe 3-3: Converting ClamAV Signatures to YARA.</i>	59
<i>Recipe 3-4: Identifying Packers with YARA and PEiD.</i>	61
<i>Recipe 3-5: Detecting Malware Capabilities with YARA</i>	63
<i>Recipe 3-6: File Type Identification and Hashing in Python.</i>	68
<i>Recipe 3-7: Writing a Multiple-AV Scanner in Python</i>	70

Recipe 3-8: Detecting Malicious PE Files in Python	75
Recipe 3-9: Finding Similar Malware with ssdeep	79
Recipe 3-10: Detecting Self-modifying Code with ssdeep	82
Recipe 3-11: Comparing Binaries with IDA and BinDiff	83
4 Sandboxes and Multi-AV Scanners	89
Recipe 4-1: Scanning Files with VirusTotal	90
Recipe 4-2: Scanning Files with Jotti	92
Recipe 4-3: Scanning Files with NoVirusThanks	93
Recipe 4-4: Database-Enabled Multi-AV Uploader in Python	96
Recipe 4-5: Analyzing Malware with ThreatExpert	100
Recipe 4-6: Analyzing Malware with CWSandbox	102
Recipe 4-7: Analyzing Malware with Anubis	104
Recipe 4-8: Writing AutoIT Scripts for Joebox	105
Recipe 4-9: Defeating Path-dependent Malware with Joebox	107
Recipe 4-10: Defeating Process-dependent DLLs with Joebox	109
Recipe 4-11: Setting an Active HTTP Proxy with Joebox	111
Recipe 4-12: Scanning for Artifacts with Sandbox Results	112
5 Researching Domains and IP Addresses	119
Recipe 5-1: Researching Domains with WHOIS	120
Recipe 5-2: Resolving DNS Hostnames	125
Recipe 5-3: Obtaining IP WHOIS Records	129
Recipe 5-4: Querying Passive DNS with BFK	132
Recipe 5-5: Checking DNS Records with Robtex	133
Recipe 5-6: Performing a Reverse IP Search with DomainTools	134
Recipe 5-7: Initiating Zone Transfers with dig	135
Recipe 5-8: Brute-forcing Subdomains with dnsmap	137
Recipe 5-9: Mapping IP Addresses to ASNs via Shodan	138
Recipe 5-10: Checking IP Reputation with RBLs	140
Recipe 5-11: Detecting Fast Flux with Passive DNS and TTLs	143
Recipe 5-12: Tracking Fast Flux Domains	146
Recipe 5-13: Static Maps with Maxmind, matplotlib, and pygeoip	148
Recipe 5-14: Interactive Maps with Google Charts API	152
6 Documents, Shellcode, and URLs	155
Recipe 6-1: Analyzing JavaScript with Spidermonkey	156
Recipe 6-2: Automatically Decoding JavaScript with Jsunpack	159
Recipe 6-3: Optimizing Jsunpack-n Decodings for Speed and Completeness	162
Recipe 6-4: Triggering exploits by Emulating Browser DOM Elements	163

Recipe 6-5: Extracting JavaScript from PDF Files with <i>pdf.py</i>	168
Recipe 6-6: Triggering Exploits by Faking PDF Software Versions	172
Recipe 6-7: Leveraging Didier Stevens's PDF Tools	175
Recipe 6-8: Determining which Vulnerabilities a PDF File Exploits	178
Recipe 6-9: Disassembling Shellcode with <i>DiStorm</i>	185
Recipe 6-10: Emulating Shellcode with <i>Libemu</i>	190
Recipe 6-11: Analyzing Microsoft Office Files with <i>OfficeMalScanner</i>	193
Recipe 6-12: Debugging Office Shellcode with <i>DisView</i> and <i>MalHost-setup</i>	200
Recipe 6-13: Extracting HTTP Files from Packet Captures with <i>Jsunpack</i>	204
Recipe 6-14: Graphing URL Relationships with <i>Jsunpack</i>	206
7 Malware Labs	211
Recipe 7-1: Routing TCP/IP Connections in Your Lab.	215
Recipe 7-2: Capturing and Analyzing Network Traffic.	217
Recipe 7-3: Simulating the Internet with <i>INetSim</i>	221
Recipe 7-4: Manipulating HTTP/HTTPS with <i>Burp Suite</i>	225
Recipe 7-5: Using Joe Stewart's <i>Truman</i>	228
Recipe 7-6: Preserving Physical Systems with <i>Deep Freeze</i>	229
Recipe 7-7: Cloning and Imaging Disks with <i>FOG</i>	232
Recipe 7-8: Automating FOG Tasks with the MySQL Database	236
8 Automation	239
Recipe 8-1: Automated Malware Analysis with <i>VirtualBox</i>	242
Recipe 8-2: Working with <i>VirtualBox</i> Disk and Memory Images.	248
Recipe 8-3: Automated Malware Analysis with <i>VMware</i>	250
Recipe 8-4: Capturing Packets with <i>TShark</i> via <i>Python</i>	254
Recipe 8-5: Collecting Network Logs with <i>INetSim</i> via <i>Python</i>	256
Recipe 8-6: Analyzing Memory Dumps with <i>Volatility</i>	258
Recipe 8-7: Putting all the Sandbox Pieces Together.	260
Recipe 8-8: Automated Analysis with <i>ZeroWine</i> and <i>QEMU</i>	271
Recipe 8-9: Automated Analysis with <i>Sandboxie</i> and <i>Buster</i>	276
9 Dynamic Analysis	283
Recipe 9-1: Logging API calls with <i>Process Monitor</i>	286
Recipe 9-2: Change Detection with <i>Regshot</i>	288
Recipe 9-3: Receiving File System Change Notifications	290
Recipe 9-4: Receiving Registry Change Notifications.	294
Recipe 9-5: Handle Table Diffing	295
Recipe 9-6: Exploring Code Injection with <i>HandleDiff</i>	300
Recipe 9-7: Watching <i>Bankpatch.C</i> Disable Windows File Protection	301

Recipe 9-8: Building an API Monitor with Microsoft Detours	304
Recipe 9-9: Following Child Processes with Your API Monitor	311
Recipe 9-10: Capturing Process, Thread, and Image Load Events	314
Recipe 9-11: Preventing Processes from Terminating	321
Recipe 9-12: Preventing Malware from Deleting Files	324
Recipe 9-13: Preventing Drivers from Loading	325
Recipe 9-14: Using the Data Preservation Module	327
Recipe 9-15: Creating a Custom Command Shell with ReactOS	330
10 Malware Forensics	337
Recipe 10-1: Discovering Alternate Data Streams with TSK	337
Recipe 10-2: Detecting Hidden Files and Directories with TSK	341
Recipe 10-3: Finding Hidden Registry Data with Microsoft's Offline API	349
Recipe 10-4: Bypassing Poison Ivy's Locked Files	355
Recipe 10-5: Bypassing Conficker's File System ACL Restrictions	359
Recipe 10-6: Scanning for Rootkits with GMER	363
Recipe 10-7: Detecting HTML Injection by Inspecting IE's DOM	367
Recipe 10-8: Registry Forensics with RegRipper Plug-ins	377
Recipe 10-9: Detecting Rogue-Installed PKI Certificates	384
Recipe 10-10: Examining Malware that Leaks Data into the Registry	388
11 Debugging Malware	395
Recipe 11-1: Opening and Attaching to Processes	396
Recipe 11-2: Configuring a JIT Debugger for Shellcode Analysis	398
Recipe 11-3: Getting Familiar with the Debugger GUI	400
Recipe 11-4: Exploring Process Memory and Resources	407
Recipe 11-5: Controlling Program Execution	410
Recipe 11-6: Setting and Catching Breakpoints	412
Recipe 11-7: Using Conditional Log Breakpoints	415
Recipe 11-8: Debugging with Python Scripts and PyCommands	418
Recipe 11-9: Detecting Shellcode in Binary Files	421
Recipe 11-10: Investigating Silentbanker's API Hooks	426
Recipe 11-11: Manipulating Process Memory with WinAppDbg Tools	431
Recipe 11-12: Designing a Python API Monitor with WinAppDbg	433
12 De-Obfuscation	441
Recipe 12-1: Reversing XOR Algorithms in Python	441
Recipe 12-2: Detecting XOR Encoded Data with yaratize	446
Recipe 12-3: Decoding Base64 with Special Alphabets	448
Recipe 12-4: Isolating Encrypted Data in Packet Captures	452

Recipe 12-5: Finding Crypto with SnD Reverser Tool, FindCrypt, and Kanal	454
Recipe 12-6: Porting OpenSSL Symbols with Zynamics BinDiff	456
Recipe 12-7: Decrypting Data in Python with PyCrypto	458
Recipe 12-8: Finding OEP in Packed Malware	461
Recipe 12-9: Dumping Process Memory with LordPE	465
Recipe 12-10: Rebuilding Import Tables with ImpREC	467
Recipe 12-11: Cracking Domain Generation Algorithms	476
Recipe 12-12: Decoding Strings with x86emu and Python	481
13 Working with DLLs	487
Recipe 13-1: Enumerating DLL Exports	488
Recipe 13-2: Executing DLLs with rundll32.exe	491
Recipe 13-3: Bypassing Host Process Restrictions	493
Recipe 13-4: Calling DLL Exports Remotely with rundll32ex	495
Recipe 13-5: Debugging DLLs with LOADDLL.EXE	499
Recipe 13-6: Catching Breakpoints on DLL Entry Points	501
Recipe 13-7: Executing DLLs as a Windows Service	502
Recipe 13-8: Converting DLLs to Standalone Executables	507
14 Kernel Debugging	511
Recipe 14-1: Local Debugging with LiveKd	513
Recipe 14-2: Enabling the Kernel's Debug Boot Switch	514
Recipe 14-3: Debug a VMware Workstation Guest (on Windows)	517
Recipe 14-4: Debug a Parallels Guest (on Mac OS X)	519
Recipe 14-5: Introduction to WinDbg Commands And Controls	521
Recipe 14-6: Exploring Processes and Process Contexts	528
Recipe 14-7: Exploring Kernel Memory	534
Recipe 14-8: Catching Breakpoints on Driver Load	540
Recipe 14-9: Unpacking Drivers to OEP	548
Recipe 14-10: Dumping and Rebuilding Drivers	555
Recipe 14-11: Detecting Rootkits with WinDbg Scripts	561
Recipe 14-12: Kernel Debugging with IDA Pro	566
15 Memory Forensics with Volatility	571
Recipe 15-1: Dumping Memory with MoonSols Windows Memory Toolkit	572
Recipe 15-2: Remote, Read-only Memory Acquisition with F-Response	575
Recipe 15-3: Accessing Virtual Machine Memory Files	576
Recipe 15-4: Volatility in a Nutshell	578
Recipe 15-5: Investigating processes in Memory Dumps	581
Recipe 15-6: Detecting DKOM Attacks with psscan	588

	<i>Recipe 15-7: Exploring csrss.exe's Alternate Process Listings</i>	591
	<i>Recipe 15-8: Recognizing Process Context Tricks</i>	593
16	Memory Forensics: Code Injection and Extraction	601
	<i>Recipe 16-1: Hunting Suspicious Loaded DLLs</i>	603
	<i>Recipe 16-2: Detecting Unlinked DLLs with ldr_modules</i>	605
	<i>Recipe 16-3: Exploring Virtual Address Descriptors (VAD)</i>	610
	<i>Recipe 16-4: Translating Page Protections</i>	614
	<i>Recipe 16-5: Finding Artifacts in Process Memory</i>	617
	<i>Recipe 16-6: Identifying Injected Code with Malfind and YARA</i>	619
	<i>Recipe 16-7: Rebuilding Executable Images from Memory</i>	627
	<i>Recipe 16-8: Scanning for Imported Functions with impscan</i>	629
	<i>Recipe 16-9: Dumping Suspicious Kernel Modules</i>	633
17	Memory Forensics: Rootkits	637
	<i>Recipe 17-1: Detecting IAT Hooks</i>	637
	<i>Recipe 17-2: Detecting EAT Hooks</i>	639
	<i>Recipe 17-3: Detecting Inline API Hooks</i>	641
	<i>Recipe 17-4: Detecting Interrupt Descriptor Table (IDT) Hooks</i>	644
	<i>Recipe 17-5: Detecting Driver IRP Hooks</i>	646
	<i>Recipe 17-6: Detecting SSDT Hooks</i>	650
	<i>Recipe 17-7: Automating Damn Near Everything with ssdt_ex</i>	654
	<i>Recipe 17-8: Finding Rootkits with Detached Kernel Threads</i>	655
	<i>Recipe 17-9: Identifying System-Wide Notification Routines</i>	658
	<i>Recipe 17-10: Locating Rogue Service Processes with svcsan</i>	661
	<i>Recipe 17-11: Scanning for Mutex Objects with mutantscan</i>	669
18	Memory Forensics: Network and Registry	673
	<i>Recipe 18-1: Exploring Socket and Connection Objects</i>	673
	<i>Recipe 18-2: Analyzing Network Artifacts Left by Zeus</i>	678
	<i>Recipe 18-3: Detecting Attempts to Hide TCP/IP Activity</i>	680
	<i>Recipe 18-4: Detecting Raw Sockets and Promiscuous NICs</i>	682
	<i>Recipe 18-5: Analyzing Registry Artifacts with Memory Registry Tools</i>	685
	<i>Recipe 18-6: Sorting Keys by Last Written Timestamp</i>	689
	<i>Recipe 18-7: Using Volatility with RegRipper</i>	692
	Index	695