Contents

Introduction xvii

I An Introduction to Memory Forensics 1

1 Systems Overview 3

Digital Environment 3
PC Architecture 4
Physical Organization. 4
CPU and MMU 4
North and Southbridge 6
Direct Memory Access 6
Volatile Memory (RAM). 6
CPU Architectures 7
Address Spaces 7
Intel IA-32 Architecture 8
Registers 8
Segmentation 8
Paging 9
Address Translation 10
Physical Address Extension. 13
Intel 64 Architecture. 14
Interrupt Descriptor Table 16
Operating Systems. 17
Privilege Separation 17
System Calls 18
Process Management 18
Threads 19
CPU Scheduling 19
System Resources 19
Memory Management 20
Virtual Memory 20
Demand Paging 21

Shared Memory 22
Stacks and Heaps. 23
File System *24*
I/O Subsystem *25*
Device Drivers 25
I/O Controls (IOCTLs) 25
Summary *26*

2 Data Structures 27

Basic Data Types *27*
C Programming Language 28
Abstract Data Types 29
Arrays 30
Bitmaps 31
Records 32
Strings 35
Linked Lists 37
Singly Linked List. 37
Doubly Linked List. 38
Circular Linked List 38
Embedded Doubly Linked Lists. 38
Lists in Physical and Virtual Memory 39
Hash Tables 39
Trees 40
Hierarchical Trees. 41
Tree Traversal 41
Analyzing Trees in Memory 42
Summary *43*

3 The Volatility Framework 45

Why Volatility? *45*
What Volatility Is Not *46*
Installation *47*
Standalone Windows Executable 47
Windows Python Module Installer 48
Source Code Packages 48
Development Branch (Code Repository) 49
Dependencies. 50

The Framework *51*
VTypes 51
Generating VTypes 52
Overlays 53
Objects and Classes. 54
Profiles 55
Address Spaces 56
Virtual/Paged Address Spaces 56
Physical Address Spaces. 56
Address Space Stacking 57
Plugin System 58
Core Plugins. 59
Using Volatility *59*
Basic Commands. 59
Displaying Help. 60
Selecting a Profile 61
Issues with Profile Selection 63
Alternatives to Command-Line Options 65
Controlling Plugin Output 66
Summary *67*

4 Memory Acquisition. 69

Preserving the Digital Environment *69*
Acquisition Overview. 70
The Risk of Acquisition 73
Atomicity 73
Device Memory 74
Cache Coherency. 76
When to Acquire Memory 76
How to Acquire Memory 77
Local Acquisition to Removable Media 77
Remote Acquisition. 78
Runtime Interrogation 79
Hardware Acquisition 79
Software Tools *79*
Tool Evaluation 80
Tool Selection. 81
Memory Acquisition with KnTDD. 83
An Example of KnTDD in Action. 84
Remote Acquisition. 88

Runtime Interrogation with F-Response 89
General Steps for Using F-Response 90
Connecting from Mac OS X and Linux 90
MoonSols Windows Memory Toolkit. 92
Local Collection. 92
Remote Collection. 94
Memory Dump Formats 95
Raw Memory Dump. 96
Windows Crash Dump. 96
Windows Hibernation File. 98
Expert Witness Format (EWF). 99
HPAK Format. 100
Virtual Machine Memory 101
VMware 101
VirtualBox 103
QEMU. 104
Xen/KVM. 105
Microsoft Hyper-V 105
Hypervisor Memory Forensics 105
Converting Memory Dumps. 106
Volatile Memory on Disk. 107
Recovering the Hibernation File 107
Querying the Registry for a Profile 108
Recovering the Page File(s). 109
Analyzing the Page File(s) 110
Crash Dump Files 113
Summary 114

II Windows Memory Forensics 115

5 Windows Objects and Pool Allocations. 117

Windows Executive Objects 117
Object Headers. 119
Optional Headers 121
Object Type Objects 122
Kernel Pool Allocations 125
Allocation APIs 126
De-allocation and Reuse 128

Pool-Tag Scanning *129*
Pool Tag Sources 130
Pooltag File 131
PoolMon Utility 132
Pool Tracker Tables 134
Building a Pool Scanner 136
Extending the PoolScanner 136
Extending AbstractScanCommand 137
Pool Scanner Algorithm 138
Finding Terminated Processes 139
Limitations of Pool Scanning *140*
Non-malicious Limitations 141
Malicious Limitations (Anti-Forensics) 141
Big Page Pool *142*
Big Page Track Tables 143
Bigpools Plugin 144
Exploring Big Page Pools 145
Pool-Scanning Alternatives *146*
Dispatcher Header Scans 146
Robust Signature Scans 147
Summary *148*

6 Processes, Handles, and Tokens 149

Processes *149*
Process Organization 153
Enumerating Processes in Memory 154
Critical System Processes 154
Analyzing Process Activity 156
Process Tree Visualizations 158
Detecting DKOM Attacks 160
The Case of Prolaco 160
Alternate Process Listings 162
Process Cross-View Plugin 163
Process Tokens *164*
Live Response: Accessing Tokens 167
Extracting and Translating SIDs in Memory 168
Detecting Lateral Movement 169
Privileges *170*
Commonly Exploited Privileges 171

Analyzing Explicit Privileges 172
Detecting Token Manipulation 173
Attack Simulation with Volshell 173
Revealing the Truth 175
Process Handles 176
Lifetime of a Handle 176
Reference Counts and Kernel Handles 177
Handle Table Internals 178
Enumerating Handles in Memory 181
Finding Zeus Indicators 181
Detecting Registry Persistence 183
Identifying Remote Mapped Drives 184
Summary 187

7 Process Memory Internals 189

What's in Process Memory? 189
Address Space Layout Details 190
Memory Allocation APIs 191
Permissions 192
Scope and Flexibility 193
Enumerating Process Memory 193
Process Page Tables 194
Exploring Process Memory 194
Virtual Address Descriptors 197
VAD Structures 198
VAD Tags 201
VAD Flags 202
Volatility VAD Plugins 204
Traversing the VAD in Python 208
Passwords in Browser Memory 209
Scanning Memory with Yara 210
Zeus Encryption Keys in Memory 213
Summary 217

8 Hunting Malware in Process Memory 219

Process Environment Block 219
Process Heaps 223
Finding Text on Notepad's Heap 223
Environment Variables 226

Attacks on Environment Variables 227
Coreflood Presence Marking 228
Standard Handles 230
Dynamic Link Libraries (DLLs) 233
How DLLs Are Loaded 233
Enumerating DLLs on Live Systems 234
Hiding DLLs 234
Listing DLLs in Memory 235
Detecting Unlinked DLLs 236
PE Files in Memory *238*
PE File Slack Space 239
Parsing PE Headers in Memory 240
PE Extraction Plugins 242
Caveats and Workarounds 244
Packing and Compression *245*
Unpacking Malicious Code 246
Common Unpacking Issues 248
Unpacking 64-bit DLLs 248
Code Injection *251*
Remote DLL Injection 252
DLL Injection Detection 253
Remote Code Injection 253
Code Injection Detection 253
Reflective DLL Injection 257
Reflective DLL Injection Detection 257
Hollow Process Injection 258
How to Hollow a Process 259
Detection 259
Postprocessing Dumped Code 262
Summary *263*

9 Event Logs 265

Event Logs in Memory *265*
Windows 2000, XP, and 2003 Event Logs 267
Finding Event Log Files 267
Extracting Event Logs 268
Logging Policies 269
Windows Vista, 2008, and 7 Event Logs 270
Caveats About Event Logs in Memory 272

Real Case Examples 275
The Case of the Unsuccessful Listener 275
The Case of the Unsuccessful Logon 276
The Case of the Impatient Brute Forcer 277
The Case of the Log Wiper 278
Summary 279

10 Registry in Memory 281

Windows Registry Analysis 281
Data in the Registry 283
Stable and Volatile Data 284
Finding Registry Hives 285
Address Translations 286
Printing Keys and Values 288
Detecting Malware Persistence 289
Volatility's Registry API 292
Parsing Userassist Keys 295
Detecting Malware with the Shimcache 297
Reconstructing Activities with Shellbags 298
Shellbags in Memory 299
Finding TrueCrypt Volumes with Shellbags 300
Timestomping Registry Keys 303
Dumping Password Hashes 304
Obtaining LSA Secrets 305
Summary 307

11 Networking 309

Network Artifacts 309
Windows Sockets API (Winsock) 310
Active Sockets and Connections 315
Attributing Connections to Code 318
Inactive Sockets and Connections 321
Hidden Connections 323
IP Packets and Ethernet Frames 323
DKOM Attacks 325
Raw Sockets and Sniffers 325
Creating Raw Sockets 326
Detecting Raw Sockets 326
Next Generation TCP/IP Stack 327

Working Backward from netstat.exe 328
Volatility's Netscan Plugin. 328
Partition Tables 330
Port Pools and Bitmaps. 332
Internet History. *333*
Carving IE History Records. 336
IE History in Malware Investigations 337
Brute Force URL Scans 338
DNS Cache Recovery *339*
Summary *341*

12 Windows Services 343

Service Architecture. *343*
Installing Services *345*
Tricks and Stealth. *346*
Investigating Service Activity *347*
Scanning Memory. 351
Volatility's SvcScan Plugin. 352
Recently Created Services 353
Detecting Hijacked Services 356
Registry-based Hijacks. 356
Disk-based Hijacks 358
Detecting Disk-based Hijacks 362
Revealing Hidden Services 362
Summary *366*

13 Kernel Forensics and Rootkits 367

Kernel Modules *367*
Classifying Modules. 369
How Modules Are Loaded. 370
Enumerating Modules on Live Systems. 371
Modules in Memory Dumps *372*
Ordered List of Active Modules 373
Brute Force Scanning for Modules 374
Recently Unloaded Modules 374
Extracting Kernel Modules 375
Threads in Kernel Mode *378*
Tigger's Kernel Threads 379
Detecting Orphan Threads 379

Driver Objects and IRPs 381
Scanning for Driver Objects 382
Hooking and Hook Detection 382
Stealthy Hooks 384
High Value Targets 386
Device Trees 386
Auditing Device Trees 388
Stuxnet's Malicious Devices 389
Auditing the SSDT 390
Enumerating the SSDT 391
Attacking the SSDT 393
Pointer Replacement 393
Inline Hooking 394
Table Duplication 394
SSDT Hook Disadvantages 395
Kernel Callbacks 396
Callbacks in Memory 396
Malicious Callbacks 397
Kernel Timers 399
Finding Timer Objects 400
Malware Analysis with Timers 400
Putting It All Together 402
Summary 406

14 Windows GUI Subsystem, Part I 407

The GUI Landscape 407
GUI Memory Forensics 410
The Session Space 410
Detecting Remote Logged-in Users over RDP 413
Window Stations 416
Analyzing the Frequency of Clipboard Usage 418
How to Detect Clipboard Snooping 419
Clipboard Viewers and Listeners 419
Snooping with Clipboardic.exe 420
Desktops 422
Enumerating Desktops and Associated Threads 424
Running Commands in Alternate Desktops 425
Hidden Windows versus Alternate Desktops 425
Tigger's Recipe for Alternate Desktops 426

Atoms and Atom Tables 429
Atoms and Atomscan 431
Detecting Window Class Names 432
Registered Window Messages 433
Atoms Are the New Mutexes 434
Windows 435
IEFrame: Currently Displayed IE Web Page 437
HH Parent: CHM File Exploit Titles 438
Current Local Time and Logged-in Username 438
Visualizing Parent and Child Windows 438
Taking Screen Shots from Memory Dumps 439
Malicious Window Abuse 441
Killing KAV with Rogue Window Messages 442
Secretly Dismissing Alerts/Prompts 443
Simulating Keystrokes and Mouse Movements 444
Window Procedure Callbacks 447
Malicious Window Subclassing 450
Summary 452

15 Windows GUI Subsystem, Part II 453

Window Message Hooks 453
Message Hook Installation 454
Detecting Message Hooks for DLL Injection 456
User Handles 459
Enumerating USER Object Types 461
The Shared Info Structure 462
Algorithm for Finding Shared Info 463
Handle Table Entries 464
Enumerating a Session's USER Handles 465
Event Hooks 466
Windows Clipboard 468
Algorithm for Clipboard Extraction 470
Recovering Text from the Clipboard 471
Recovering Binary Data from the Clipboard 471
Case Study: ACCDFISA Ransomware 472
Summary 476

16 Disk Artifacts in Memory 477

Master File Table *477*

The MFTParser Plugin 479

Alternate Data Streams... 482

The Case of the Illicit File Access 484

Files and Shortcuts 484

Searching in the Trash 485

Translating the Embedded Timestamp 486

The Case of Data Exfiltration..................................... 487

Proof of Execution.. 487

The Fake “systems” Directory.................................. 489

Surveying the Network 489

WinRAR Archive Exfiltration.................................. 490

MFT-Resident Data 491

Timestomping the MFT 492

Disadvantages of MFT Scanning 492

Extracting Files *493*

Executable (Image) and Data Files 495

Shared Cached Files 496

Volatility’s Dumpfiles Plugin...................................... 497

Targeted File Extraction.. 500

Detecting Modified Code 501

Defeating TrueCrypt Disk Encryption *503*

Password Caching.. 504

Encrypted Volume Identification 505

The Cache Manager and NTFS Metadata 506

Extracting (AES) Master Keys 507

Non-Default Encryption Algorithms 508

Summary *510*

17 Event Reconstruction...511

Strings *511*

Extracting Strings 512

Windows 513

Linux... 513

Mac OS X 514

Example Output 514

Translating Strings 515

String-Based Analysis. 516
Finding Prefetch Files. 517
Spatial Proximity with IOCs 517
Strings in Free Memory 518
Detecting Shared Pages 519
Command History *523*
Windows Command Architecture 524
Console Modules and Functions 525
Data Structure Map 525
Default Settings 529
Finding Commands in Memory 530
CmdScan Plugin 531
Consoles Plugin. 532
Summary *536*

18 Timelining 537

Finding Time in Memory *537*
Timestamp Formats 538
Timestamp Sources. 538
Generating Timelines. *539*
Timeliner Plugin 539
Timeliner Output Formats. 541
Processing Timelines Using Mactime 541
Where to Begin 542
Gh0st in the Enterprise *543*
Scripting Registry Timelines. 545
Adding Packet Capture Data 545
Finding the Initial Infection Vector 546
Tracking Executed Programs 546
Phishing E-mail Artifacts 547
Examining the 6to4 Service. 548
Finding an Active Attacker 551
Mapping Remote File Shares. 554
Scheduled Jobs for Hash Dumping 555
Overlaying Attack Artifacts. 558
Decoding the Network Data 561
Correlating the Traffic with the Timeline 567
Wrapping Up the Case 572
Summary *573*

III Linux Memory Forensics. 575

19 Linux Memory Acquisition 577

Historical Methods of Acquisition 577
/dev/mem. 578
/dev/kmem 578
ptrace. 578
Modern Acquisition. 579
fmem 579
Linux Memory Extractor (LiME) 580
Compiling LiME 581
Loading LiME and Dumping Memory 582
/proc/kcore 583
Volatility Linux Profiles. 583
Software Setup. 584
Creating a Profile. 584
Creating VTypes 584
Getting Symbols 585
Making the Profile 586
Using the Profile 586
Enterprise Linux Memory Forensics 587
Caveats to Cross Compiling 587
Cross Compiling for Ubuntu 587
Summary 589

20 Linux Operating System 591

ELF Files 591
ELF Header. 592
Sections. 593
Displaying Sections with readelf 593
Packed ELF Binaries 595
Program Headers 595
Listing Program Headers. 596
UPX Effect on Program Headers 597
Shared Library Loading 598
Global Offset Table 598
Using the GOT to Locate APIs 599
Overwriting GOT Entries 599

GOT and PLT Internals 599

Procedure Linkage Table 601

Linux Data Structures *603*

Lists 603

Hash Tables 605

Trees 605

Handling Embedded Structures 606

Linux Address Translation *607*

Kernel Identity Paging 607

Finding the Kernel DTB 608

Validating the Address Space 608

procfs and sysfs *609*

Compressed Swap *610*

Summary *610*

21 Processes and Process Memory 611

Processes in Memory *611*

Enumerating Processes *613*

Active Process List 614

Linking Processes to Users 615

Parent and Child Relationships 615

PID Hash Table 616

Process Address Space *616*

Enumerating Process Mappings 618

Recovering Sections of Memory 621

Analyzing Command-line Arguments 621

Manipulating Command-line Arguments 623

Process Environment Variables *625*

Open File Handles *626*

Saved Context State *630*

Bash Memory Analysis *630*

Bash History 631

Linux Bash Plugin 631

Bash Command Hash Table 633

The Fake rm Command 633

Detecting the Fake Binary 634

Summary *635*

22 Networking Artifacts 637

Network Socket File Descriptors *637*
Network Connections *640*
TCP and UDP Connections 641
Recovering Unix Sockets 643
Queued Network Packets. *643*
Network Interfaces. *646*
Listing Interface Information 648
Interface Naming Conventions. 648
Interface Aliases 648
Finding Processes with Raw Sockets 649
The Route Cache *650*
ARP Cache *652*
Summary *655*

23 Kernel Memory Artifacts. 657

Physical Memory Maps. *657*
Hardware Resources. 658
Verifying Acquisition Tools 660
Virtual Memory Maps *661*
Kernel Debug Buffer *663*
Loaded Kernel Modules *667*
Enumerating LKMs. 669
Extracting Kernel Modules 670
Summary *673*

24 File Systems in Memory 675

Mounted File Systems *675*
Linux Mount Plugin 677
Common File System Types 678
Mount Options 679
Temporary File Systems 680
Listing Files and Directories *681*
Extracting File Metadata. *684*
Duplicating File Systems from Memory 685
Building Timelines 689
Recovering File Contents. *691*
File Extraction Algorithm 692
Find File Plugin 693
Summary *695*

25 Userland Rootkits 697

Shellcode Injection *698*

Step 1: Attaching to a Process with PTrace 699

Step 2: Finding/Allocating Memory..... 699

Overwriting Existing Code 699

Foreign Memory Allocation 700

Step 3: Writing Shellcode into a Process 701

Step 4: Controlling Foreign Process Execution..... 701

Detecting Shellcode Injection 701

Process Hollowing *703*

The Detection Algorithm..... 703

An Example of Detection..... 704

Shared Library Injection *705*

Injecting a Library from Disk 706

Detecting Disk–Based Shared Library Injections..... 707

Listing Libraries in Userland 708

Cross-Referencing Mappings 708

Injecting a Library from Memory..... 709

Extracting Executables from Memory 710

LD_PRELOAD Rootkits..... *712*

Hooking Functions with LD_PRELOAD..... 712

Detecting LD_PRELOAD Rootkits 713

Extracting Preload Files 714

Analyzing Preload Variables..... 714

Checking the GOT/PLT 715

Comparative Symbol Analysis 715

GOT/PLT Overwrites *716*

PLT Tampering 716

Detecting GOT Overwrites 717

Inline Hooking *718*

Summary *719*

26 Kernel Mode Rootkits.....721

Accessing Kernel Mode *721*

Hidden Kernel Modules..... *722*

Hiding from sysfs 723

Detecting KBeast..... 723

Examining the Disassembly 724

Carving for Hidden Modules 725
Finding Modules by File Handles. 726
Kernel-Mode Code Injection 727
Hidden Processes *728*
Sources of Process Listings 728
Cross View Exceptions 730
Elevating Privileges *730*
Hijacked Credentials. 731
Average Coder Rootkit 732
System Call Handler Hooks. *734*
Keyboard Notifiers *735*
Malicious Keyboard Notifiers 736
Resolving Malicious Functions 738
TTY Handlers *739*
Network Protocol Structures *742*
Network Sequence Operations 743
Finding Network Protocol Hooks. 744
Netfilter Hooks. *745*
Netfilter Subsystem. 746
Abusing Netfilter. 747
Enumerating Netfilter Hooks 748
File Operations *748*
Sources of File Operations 749
Frustrating Live Forensics 750
Hiding Logged-in Users. 751
Inline Code Hooks *752*
Summary *754*

27 Case Study: Phalanx2. 755

Phalanx2 . . . . *755*
Analysis Setup. 756
Installation 756
Phalanx2 Memory Analysis *757*
Kernel Tampering 757
Baseline Analysis 759
Inspecting the Xnest Process. 760
Revealing the Fake Kernel Thread 760
Determining the True Binary Path. 761
Socket File Descriptors. 761

Reverse Engineering Phalanx2 763
Kernel Module 763
Extracting Shell Commands 763
Bypassing the /dev/mem Restriction 764
Detecting Existing Infections 766
Dynamic Analysis with strace 767
Symlink Hook 770
Final Thoughts on Phalanx2 772
Summary 772

IV Mac Memory Forensics 773

28 Mac Acquisition and Internals 775

Mac Design 775
Mach and BSD Layers. 776
Kernel/Userland Virtual Address Split 777
Kernel ASLR. 777
Process Address Spaces 778
Memory Acquisition 780
Locating RAM Regions 780
Mac Memory Reader (MMR). 781
Mac Memoryze 783
OSXPmem 783
Mac Volatility Profiles 784
Downloading Profiles. 785
Building Profiles 786
Mach-O Executable Format. 787
Mach-O Header 787
Command Structures 788
Segment Commands and Sections 788
Mach-O Address Space 790
Summary 791

29 Mac Memory Overview 793

Mac versus Linux Analysis. 793
Process Analysis 794
Enumerating Processes. 796
Process Relationships 798
Address Space Mappings. 799

Listing and Recovering Mappings 801
Dynamic Loader Shared Cache. 802
Networking Artifacts 804
Process File Descriptors 806
Networking Subsystem 807
Cross-Reference Advantage. 807
Classifying Network Connections 807
SLAB Allocator 808
Recovering File Systems from Memory. 811
Loaded Kernel Extensions 815
Enumerating Kernel Modules 816
Recovering Modules from Memory 818
Other Mac Plugins 818
Mac Live Forensics 819
Summary 821

30 Malicious Code and Rootkits 823

Userland Rootkit Analysis. 823
Code Injection Techniques. 823
Macterpreter – Bundle Files 824
Detecting Code Injections 824
Crisis Injected Bundles 825
Extracting Executables 826
Detecting API Hooks 826
Inline Hooking 827
Hooking Relocation Tables 827
Library Search Order Hijacking 827
API Hook Detection 827
Kernel Rootkit Analysis. 828
Hidden Kernel Extensions 828
Kernel Extension Map 829
Hidden Processes 829
Sysctl Handlers 830
System Call and Trap Table Hooks 831
Shadow System Call Tables 832
IOKit Notifiers 832
Detecting LogKext 833
TrustedBSD. 834
IPC Handlers 835

Network Kernel Extensions 836
IP Filters 837
Socket Filters 837
Common Mac Malware in Memory 838
OSX.GetShell 839
OSX.FkCodec 840
Detecting Persistence 842
Summary 844

31 Tracking User Activity 845

Keychain Recovery 845
Volafox and Chainbreaker 846
Breaking Keychains in Memory 846
Mac Application Analysis 849
Apple Mail and GPG. 850
Finding Plaintext Messages. 850
Locating E-mail Attachments 852
Mail Account Passwords 853
Apple Contacts. 853
Apple Calendar 855
Apple Notes 856
Adium Chat Messages 856
Summary 858

Index 859