

Contents

Acknowledgments	xiii
About the Authors	xv
About the Technical Editor	xix
Introduction	xxi

1. Malware Incident Response

Volatile Data Collection and Examination on a Live Linux System

Introduction	2
Local vs. Remote Collection	4
Volatile Data Collection Methodology	5
Documenting Collection Steps	5
Volatile Data Collection Steps	5
Preservation of Volatile Data	6
Physical Memory Acquisition on a Live Linux System	7
Acquiring Physical Memory Locally	8
Documenting the Contents of the <code>/proc/meminfo</code> File	11
Remote Physical Memory Acquisition	12
Other Methods of Acquiring Physical Memory	16
Collecting Subject System Details	19
Identifying Users Logged into the System	26
Inspect Network Connections and Activity	27
Collecting Process Information	31
Preserving Process Memory on a Live Linux System	36
Examine Running Processes in Relational Context to System State and Artifacts	39
Volatile Data in <code>/proc</code> Directory	40
Correlate open Ports with Running Processes and Programs	42
Open Files and Dependencies	44
Identifying Running Services	46
Examine Loaded Modules	47
Collecting the Command History	48
Identifying Mounted and Shared Drives	49
Determine Scheduled Tasks	50
Collecting Clipboard Contents	50
Nonvolatile Data Collection from a Live Linux System	51
Forensic Duplication of Storage Media on a Live Linux System	51
Remote Acquisition of Storage Media on a Live Linux System	52
Forensic Preservation of Select Data on a Live Linux System	62
Assess Security Configuration	62
Assess Trusted Host Relationships	63

Collect Login and System Logs	64
Conclusion	65
Pitfalls to Avoid	67
Incident Tool Suites	89
Remote Collection Tools	90
Volatile Data Collection and Analysis Tools	93
Physical Memory Acquisition	93
Collecting Subject System Details	95
Identifying Users Logged into the System	98
Network Connections and Activity	100
Process Analysis	101
Loaded Modules	103
Open Files	104
Command History	104
Selected Readings	105
Books	105
Papers	105
Online Resources	105
Jurisprudence/RFCs/Technical Specifications	106

2. Linux Memory Forensics

Analyzing Physical and Process Memory Dumps for Malware Artifacts

Introduction	107
Memory Forensics Overview	109
“Old School” Memory Analysis	110
How Linux Memory Forensics Tools Work	113
Linux Memory Forensics Tools	114
Processes and Threads	116
Modules and Libraries	121
Open Files and Sockets	124
Interpreting Various Data Structures in Linux Memory	127
System Details and Logs	128
Temporary Files	129
Command History	129
Cryptographic Keys and Passwords	130
Dumping Linux Process Memory	132
Recovering Executable Files	133
Recovering Process Memory	134
Extracting Process Memory on Live Systems	135
Dissecting Linux Process Memory	137
Conclusions	141
Pitfalls to Avoid	143
Field Notes: Memory Forensics	145
Selected Readings	161
Books	161
Papers	161
Online Resources	161

3. Postmortem Forensics

Discovering and Extracting Malware and Associated Artifacts from Linux Systems

Introduction	163
Linux Forensic Analysis Overview	164
Malware Discovery and Extraction from a Linux System	168
Search for Known Malware	168
Survey Installed Programs and Potentially Suspicious Executables	173
Inspect Services, Modules, Auto-Starting Locations, and Scheduled Jobs	176
Examine Logs	177
Review User Accounts and Logon Activities	180
Examine Linux File System	182
Examine Application Traces	186
Keyword Searching	187
Forensic Reconstruction of Compromised Linux Systems	190
Advanced Malware Discovery and Extraction from a Linux System	192
Conclusion	193
Pitfalls to Avoid	195
Field Notes: Linux System Examinations	197
Forensic Tool Suites	205
Timeline Generation	210
Selected Readings	211
Books	211
Papers	211

4. Legal Considerations

Framing the Issues	214
General Considerations	214
The Legal Landscape	215
Sources of Investigative Authority	216
Jurisdictional Authority	216
Private Authority	218
Statutory/Public Authority	219
Statutory Limits on Authority	220
Stored Data	220
Real-time Data	221
Protected Data	224
Tools for Acquiring Data	229
Business Use	229
Investigative Use	229
Dual Use	230
Acquiring Data Across Borders	233
Workplace Data in Private or Civil Inquiries	233
Workplace Data in Government or Criminal Inquiries	235

Involving Law Enforcement	237
Victim Reluctance	237
Victim Misperception	237
The Law Enforcement Perspective	238
Walking the Line	238
Improving Chances for Admissibility	239
Documentation	239
Preservation	240
Chain of Custody	241
State Private Investigator and Breach Notification Statutes	243
International Resources	245
Cross-Border Investigations	245
The Federal Rules: Evidence for Digital Investigators	246
Best Evidence	247
Expert Testimony	247
Limitations on Waiver of the Attorney–Client Privilege	247
5. File Identification and Profiling	
<i>Initial Analysis of a Suspect File on a Linux System</i>	
Introduction	249
Overview of the File Profiling Process	250
Working With Linux Executables	252
How an Executable File is Compiled	252
Static versus Dynamic Linking	253
Symbolic and Debug Information	253
Stripped Executables	254
Profiling a Suspicious File	255
Command-Line Interface MD5 Tools	261
GUI MD5 Tools	262
File Similarity Indexing	263
File Visualization	265
File Signature Identification and Classification	266
File Types	267
File Signature Identification and Classification Tools	269
Web-Based Malware Scanning Services	273
Embedded Artifact Extraction: Strings, Symbolic Information, and File Metadata	276
Strings	277
Tools for Analyzing Embedded Strings	279
Symbolic and Debug Information	285
Embedded File Metadata	293
File Obfuscation: Packing and Encryption Identification	297
Packers	298
Cryptors	299
Wrappers	300
Identifying an Obfuscated File	302
Embedded Artifact Extraction Revisited	307
Executable and Linkable Format (ELF)	307

Using the ELF Shell (<code>elfsh</code>)	309
The ELF Header (<code>Elf32_ehdr</code>)	309
The ELF Section Header Table (<code>Elf32_shdr</code>)	312
Program Header Table (<code>Elf32_phdr</code>)	317
Extracting Symbolic Information from the Symbol Table	319
Version Information	324
Notes Section Entries	324
Dynamic Section Entries	325
Version Control Information	332
Parsing a Binary Specimen with <code>Objdump</code>	333
Profiling Suspect Document Files	336
Profiling Adobe Portable Document Format (PDF) Files	338
PDF File Format	338
PDF Profiling Process: CLI Tools	341
PDF Profiling Process: GUI Tools	348
Profiling Microsoft (MS) Office Files	353
MS Office Documents: Word, PowerPoint, Excel	353
MS Office Documents: File Format	353
MS Office Documents: Vulnerabilities and Exploits	355
MS Office Document Profiling Process	355
Deeper Profiling with <code>OfficeMalScanner</code>	359
Conclusion	366
Pitfalls to Avoid	369
Selected Readings	409
Books	409
Papers	409
Online Resources	409
Technical Specifications	410
6. Analysis of a Malware Specimen	
Introduction	411
Goals	412
Guidelines for Examining a Malicious File Specimen	413
Establishing the Environment Baseline	413
System Snapshots	414
Host Integrity Monitors	414
Installation Monitors	415
Pre-Execution Preparation: System and Network Monitoring	417
Passive System and Network Monitoring	418
Active System and Network Monitoring	419
Anomaly Detection and Event-Based Monitoring with Network Intrusion Detection Systems (NIDS)	431
Execution Artifact Capture: Digital Impression and Trace Evidence	434
Impression Evidence	434
Trace Evidence	435
Digital Impression Evidence	435
Digital Trace Evidence	435

Trace and Impression Evidence in Physical Memory	436
Executing the Malicious Code Specimen	439
Execution Trajectory Analysis: Observing Network, Process, System Calls, and File System Activity	441
Network Activity: Network Trajectory, Impression and Trace Evidence	441
Environment Emulation and Adjustment: Network Trajectory Reconstruction	442
Network Trajectory Reconstruction: Chaining	444
Network Impression and Trace Evidence	445
Examining Process Activity	446
Exploring the <code>/proc/<pid></code> directory	452
Process-to-Port Correlation: Examining Network Connections and Open Ports	452
Capturing System Calls with <code>strace</code>	456
Capturing System Calls with SystemTap and Mortadelo	459
Capturing Dynamic Library Calls with <code>ltrace</code>	463
Examining a Running Process with <code>gdb</code>	466
Examining File System Activity	469
Automated Malware Analysis Frameworks	470
Embedded Artifact Extraction Revisited	471
Examining the Suspect Program in a Disassembler	472
Interacting with and Manipulating the Malware Specimen:	
Exploring and Verifying Functionality and Purpose	476
Prompting Trigger Events	476
Client Applications	477
Event Reconstruction and Artifact Review: Post-Run Data Analysis	479
Passive Monitoring Artifacts	480
Active Monitoring Artifacts	483
Analyzing Captured Network Traffic	484
Analyzing System Calls	491
Analyzing-IDS alerts	493
Physical Memory Artifacts	494
Other Considerations	495
Digital Virology: Advanced Profiling Through Malware	
Taxonomy and Phylogeny	496
Context Triggered Piecewise Hasing (CTPH)	499
Textual and Binary Indicators of Likeness	499
Function Flowgraphs	502
Process Memory Trajectory Analysis	505
Visualization	507
Behavioral Profiling and Classification	508
Conclusion	511
Pitfalls to Avoid	513
Selected Readings	563
Books	563
Papers	563