CONTENTS IN DETAIL

FOREWORD by Rodrigo Rubira Branco xix

ACKNOWLEDGMENTS xxiii

ABBREVIATIONS xxv

INTRODUCTION xxix

Why Read This Book? xxx
What's in the Book? xxx
    Part 1: Rootkits xxxi
    Part 2: Bootkits xxxi
    Part 3: Defense and Forensic Techniques xxxiii
How to Read This Book. xxxiii

PART I: ROOTKITS

1
WHAT'S IN A ROOTKIT: THE TDL3 CASE STUDY 3

History of TDL3 Distribution in the Wild. 4
Infection Routine 5
Controlling the Flow of Data 7
    Bring Your Own Linker 7
    How TDL3's Kernel-Mode Hooks Work 8
The Hidden Filesystem 10
Conclusion: TDL3 Meets Its Nemesis. 12

2
FESTI ROOTKIT: THE MOST ADVANCED SPAM AND DDOS BOT 13

The Case of Festi Botnet 14
Dissecting the Rootkit Driver 15
    Festi Configuration Information for C&C Communication 16
    Festi's Object-Oriented Framework 17
    Plug-in Management 17
    Built-in Plug-ins 19
    Anti–Virtual Machine Techniques 20
    Antidebugging Techniques. 22
    The Method for Hiding the Malicious Driver on Disk 22
    The Method for Protecting the Festi Registry Key 25
The Festi Network Communication Protocol 26
    Initialization Phase 26
    Work Phase 26

Bypassing Security and Forensics Software 27
The Domain Generation Algorithm for C&C Failure 30
Malicious Functionality 31
The Spam Module 31
The DDoS Engine 32
Festi Proxy Plug-in 33
Conclusion 34

3
OBSERVING ROOTKIT INFECTIONS 35

Methods of Interception 36
Intercepting System Events 36
Intercepting System Calls 37
Intercepting the File Operations 40
Intercepting the Object Dispatcher 41
Restoring the System Kernel 43
The Great Rootkits Arms Race: A Nostalgic Note 44
Conclusion 46

PART II: BOOTKITS

4
EVOLUTION OF THE BOOTKIT 49

The First Bootkits 50
Boot Sector Infectors 50
Elk Cloner and Load Runner 50
The Brain Virus 51
The Evolution of Bootkits 51
The End of the BSI Era 51
The Kernel-Mode Code Signing Policy 52
The Rise of Secure Boot 53
Modern Bootkits 53
Conclusion 55

5
OPERATING SYSTEM BOOT PROCESS ESSENTIALS 57

High-Level Overview of the Windows Boot Process 58
The Legacy Boot Process 59
The Windows Boot Process 60
BIOS and the Preboot Environment 60
The Master Boot Record 60
The Volume Boot Record and Initial Program Loader 62
The bootmgr Module and Boot Configuration Data 64
Conclusion 68

6
BOOT PROCESS SECURITY 69

The Early Launch Anti-Malware Module 70
    API Callback Routines 70
    How Bootkits Bypass ELAM 72
Microsoft Kernel-Mode Code Signing Policy 73
    Kernel-Mode Drivers Subject to Integrity Checks 73
    Location of Driver Signatures 73
    The Legacy Code Integrity Weakness 74
    The ci.dll Module 76
    Defensive Changes in Windows 8 77
Secure Boot Technology 78
Virtualization-Based Security in Windows 10. 79
    Second Level Address Translation 80
    Virtual Secure Mode and Device Guard. 80
    Device Guard Limitations on Driver Development 81
Conclusion 82

7
BOOTKIT INFECTION TECHNIQUES 83

MBR Infection Techniques 84
    MBR Code Modification: The TDL4 Infection Technique 84
    MBR Partition Table Modification 90
VBR/IPL Infection Techniques 91
    IPL Modifications: Rovnix 91
    VBR Infection: Gapz 92
Conclusion 93

8
STATIC ANALYSIS OF A BOOTKIT USING IDA PRO 95

Analyzing the Bootkit MBR 96
    Loading and Decrypting the MBR 96
    Analyzing the BIOS Disk Service. 101
    Analyzing the Infected MBR's Partition Table 104
VBR Analysis Techniques. 106
    Analyzing the IPL 106
    Evaluating Other Bootkit Components 107
Advanced IDA Pro Usage: Writing a Custom MBR Loader 108
    Understanding loader.hpp 109
    Implementing accept_file 109
    Implementing load_file. 110
    Creating the Partition Table Structure. 111
Conclusion 113
Exercises. 113

9
BOOTKIT DYNAMIC ANALYSIS: EMULATION AND VIRTUALIZATION 115

Emulation with Bochs 116
  Installing Bochs 117
  Creating a Bochs Environment 117
  Infecting the Disk Image 119
  Using the Bochs Internal Debugger 121
  Combining Bochs with IDA. 123
Virtualization with VMware Workstation 124
  Configuring the VMware Workstation 125
  Combining VMware GDB with IDA 126
Microsoft Hyper-V and Oracle VirtualBox 130
Conclusion 130
Exercises 130

10
AN EVOLUTION OF MBR AND VBR INFECTION TECHNIQUES: OLMASCO 133

The Dropper 134
  Dropper Resources 134
  Tracing Functionality for Future Development 136
  Antidebugging and Antiemulation Tricks 137
The Bootkit Functionality 138
  Bootkit Infection Technique 138
  Boot Process of the Infected System 140
The Rootkit Functionality 141
  Hooking the Hard Drive Device Object and Injecting the Payload 141
  Maintaining the Hidden Filesystem 141
  Implementing the Transport Driver Interface to Redirect Network Communication 144
Conclusion 145

11
IPL BOOTKITS: ROVNIX AND CARBERP 147

Rovnix's Evolution 148
The Bootkit Architecture 149
Infecting the System 150
Post-Infection Boot Process and IPL 152
  Implementing the Polymorphic Decryptor 152
  Decrypting the Rovnix Bootloader with VMware and IDA Pro 153
  Taking Control by Patching the Windows Bootloader 159
  Loading the Malicious Kernel-Mode Driver 163
Kernel-Mode Driver Functionality 164
  Injecting the Payload Module 164
  Stealth Self-Defense Mechanisms. 166

The Hidden Filesystem 167
Formatting the Partition as a Virtual FAT System 168
Encrypting the Hidden Filesystem 168
Accessing the Hidden Filesystem 168
The Hidden Communication Channel 169
Case History: The Carberp Connection 171
Development of Carberp 171
Dropper Enhancements 173
Leaked Source Code 174
Conclusion 175

12
GAPZ: ADVANCED VBR INFECTION 177

The Gapz Dropper 178
Dropper Algorithm 180
Dropper Analysis 180
Bypassing HIPS 181
Infecting the System with the Gapz Bootkit 186
Reviewing the BIOS Parameter Block 186
Infecting the VBR 188
Loading the Malicious Kernel-Mode Driver 189
Gapz Rootkit Functionality 191
Hidden Storage 193
Self-Defense Against Antimalware Software 194
Payload Injection 196
Payload Communication Interface 201
Custom Network Protocol Stack 204
Conclusion 206

13
THE RISE OF MBR RANSOMWARE 207

A Brief History of Modern Ransomware 208
Ransomware with Bootkit Functionality 209
The Ransomware Modus Operandi 210
Analyzing the Petya Ransomware 212
Acquiring Administrator Privileges 212
Infecting the Hard Drive (Step 1) 213
Encrypting with the Malicious Bootloader Configuration Data 215
Crashing the System 219
Encrypting the MFT (Step 2) 220
Wrapping Up: Final Thoughts on Petya 224
Analyzing the Satana Ransomware 225
The Satana Dropper 225
The MBR Infection 226
Dropper Debug Information 227
The Satana Malicious MBR 228
Wrapping Up: Final Thoughts on Satana 230
Conclusion 231

14
UEFI BOOT VS. THE MBR/VBR BOOT PROCESS 233

The Unified Extensible Firmware Interface 234
Differences Between the Legacy BIOS and UEFI Boot Processes 235
The Boot Process Flow 235
Disk Partitioning: MBR vs. GPT 235
Other Differences 237
GUID Partition Table Specifics 238
How UEFI Firmware Works 242
The UEFI Specification 243
Inside the Operating System Loader 245
The Windows Boot Loader 250
Security Benefits of UEFI Firmware. 253
Conclusion 253

15
CONTEMPORARY UEFI BOOTKITS 255

Overview of Historical BIOS Threats 256
WinCIH, the First Malware to Target BIOS 256
Mebromi 257
An Overview of Other Threats and Counters 258
All Hardware Has Firmware 261
UEFI Firmware Vulnerabilities 263
(In)Effectiveness of Memory Protection Bits 263
Checks for Protection Bits 264
Ways to Infect the BIOS 265
Modifying an Unsigned UEFI Option ROM. 267
Adding or Modifying a DXE Driver 269
Understanding Rootkit Injection 269
UEFI Rootkits in the Wild 275
Hacking Team’s Vector-EDK Rootkit 275
Conclusion 283

16
UEFI FIRMWARE VULNERABILITIES 285

What Makes Firmware Vulnerable? 286
Classifying UEFI Firmware Vulnerabilities 289
Post-Exploitation Vulnerabilities 290
Compromised Supply Chain Vulnerabilities 291
Supply Chain Vulnerability Mitigation 292
A History of UEFI Firmware Protections 293
How BIOS Protections Work. 294
SPI Flash Protections and Their Vulnerabilities 294
Risks Posed by an Unauthenticated BIOS Update 297
BIOS Protection with Secure Boot 297
Intel Boot Guard 299
Intel Boot Guard Technology 299
Vulnerabilities in Boot Guard 300

Vulnerabilities in the SMM Modules 302
Understanding SMM 302
Exploiting SMI Handlers 302
Vulnerabilities in the S3 Boot Script 306
Understanding the S3 Boot Script 306
Targeting Weaknesses of the S3 Boot Script 307
Exploiting the S3 Boot Script Vulnerability 308
Fixing the S3 Boot Script Vulnerability 311
Vulnerabilities in the Intel Management Engine 311
A History of ME Vulnerabilities 311
ME Code Attacks 312
Case Studies: Attacks on Intel AMT and BMC 312
Conclusion 315

PART III: DEFENSE AND FORENSIC TECHNIQUES

17
HOW UEFI SECURE BOOT WORKS **319**

What Is Secure Boot? 320
UEFI Secure Boot Implementation Details 320
The Boot Sequence 321
Executable Authentication with Digital Signatures 322
The db Database 323
The dbx Database 326
Time-Based Authentication 328
Secure Boot Keys 328
UEFI Secure Boot: The Complete Picture 330
Secure Boot Policy 332
Protection Against Bootkits Using Secure Boot 334
Attacking Secure Boot 335
Patching PI Firmware to Disable Secure Boot 335
Modifying the UEFI Variables to Bypass Security Checks 337
Protecting Secure Boot with Verified and Measured Boot 338
Verified Boot 339
Measured Boot 339
Intel BootGuard 339
Finding the ACM 340
Exploring FIT 342
Configuring Intel BootGuard 343
ARM Trusted Boot Board 346
ARM Trust Zone 346
ARM Boot Loaders 347
Trusted Boot Flow 348
Verified Boot vs. Firmware Rootkits 350
Conclusion 350

18
APPROACHES TO ANALYZING HIDDEN FILESYSTEMS 351

Overview of Hidden Filesystems 352
Retrieving Bootkit Data from a Hidden Filesystem 353
    Retrieving Data from an Offline System 353
    Reading Data on a Live System 353
    Hooking the Miniport Storage Driver 354
Parsing the Hidden Filesystem Image 360
The HiddenFsReader Tool 360
Conclusion 362

19
BIOS/UEFI FORENSICS:
FIRMWARE ACQUISITION AND ANALYSIS APPROACHES 363

Limitations of Our Forensic Techniques 364
Why Firmware Forensics Matter 364
    Attacking the Supply Chain 364
    Compromising BIOS Through Firmware Vulnerability 365
Understanding Firmware Acquisition 365
The Software Approach to Firmware Acquisition 367
    Locating PCI Configuration Space Registers 368
    Calculating SPI Configuration Register Addresses 369
    Using the SPI Registers 369
    Reading Data from the SPI Flash 372
    Considering the Drawbacks of the Software Approach 373
The Hardware Approach to Firmware Acquisition 374
    Reviewing a Lenovo ThinkPad T540p Case Study 375
    Locating the SPI Flash Memory Chip 376
    Reading the SPI Flash with the FT2232 Mini Module 377
Analyzing the Firmware Image with UEFITool 380
    Getting to Know the SPI Flash Regions 380
    Viewing SPI Flash Regions with UEFITool 381
    Analyzing the BIOS Region 383
Analyzing the Firmware Image with Chipsec 386
    Getting to Know the Chipsec Architecture 387
    Analyzing Firmware with Chipsec Util 388
Conclusion 390

INDEX 391