

CONTENTS IN DETAIL

ACKNOWLEDGMENTS	xvii
------------------------	-------------

INTRODUCTION	xix
---------------------	------------

Who This Book Is For	xx
What Is in This Book.	xx
Prerequisite Knowledge.	xxii
Setting Up.	xxiii

1	
EDR-CHITECTURE	1

The Components of an EDR.	2
The Agent	2
Telemetry	2
Sensors	3
Detections	4
The Challenges of EDR Evasion	4
Identifying Malicious Activity	5
Considering Context	6
Applying Brittle vs. Robust Detections.	7
Exploring Elastic Detection Rules	8
Agent Design.	9
Basic.	9
Intermediate	10
Advanced	11
Types of Bypasses	12
Linking Evasion Techniques: An Example Attack.	13
Conclusion	15

2	
FUNCTION-HOOKING DLLS	17

How Function Hooking Works	18
Implementing the Hooks with Microsoft Detours.	19
Injecting the DLL	22
Detecting Function Hooks	22
Evading Function Hooks	24
Making Direct Syscalls.	25
Dynamically Resolving Syscall Numbers.	27
Remapping ntdll.dll	28
Conclusion	31

3	
PROCESS- AND THREAD-CREATION NOTIFICATIONS	33

How Notification Callback Routines Work.	34
Process Notifications	34

Registering a Process Callback Routine	35
Viewing the Callback Routines Registered on a System	36
Collecting Information from Process Creation	37
Thread Notifications	39
Registering a Thread Callback Routine.	39
Detecting Remote Thread Creation	40
Evading Process- and Thread-Creation Callbacks	41
Command Line Tampering	41
Parent Process ID Spoofing	45
Process-Image Modification	49
A Process Injection Case Study: fork&run	58
Conclusion	59

4 OBJECT NOTIFICATIONS 61

How Object Notifications Work	62
Registering a New Callback	62
Monitoring New and Duplicate Process-Handle Requests	63
Detecting Objects an EDR Is Monitoring	64
Detecting a Driver's Actions Once Triggered	66
Evading Object Callbacks During an Authentication Attack	68
Performing Handle Theft.	69
Racing the Callback Routine	74
Conclusion	78

5 IMAGE-LOAD AND REGISTRY NOTIFICATIONS 79

How Image-Load Notifications Work	80
Registering a Callback Routine	80
Viewing the Callback Routines Registered on a System	80
Collecting Information from Image Loads	81
Evading Image-Load Notifications with Tunneling Tools	84
Triggering KAPC Injection with Image-Load Notifications.	86
Understanding KAPC Injection	86
Getting a Pointer to the DLL-Loading Function	87
Preparing to Inject.	87
Creating the KAPC Structure.	88
Queueing the APC	90
Preventing KAPC Injection	90
How Registry Notifications Work	91
Registering a Registry Notification.	92
Mitigating Performance Challenges.	95
Evading Registry Callbacks	96
Evading EDR Drivers with Callback Entry Overwrites	100
Conclusion	101

6 FILESYSTEM MINIFILTER DRIVERS 103

Legacy Filters and the Filter Manager	104
Minifilter Architecture	106

Writing a Minifilter	108
Beginning the Registration	108
Defining Pre-operation Callbacks	110
Defining Post-operation Callbacks	113
Defining Optional Callbacks	114
Activating the Minifilter	114
Managing a Minifilter	115
Detecting Adversary Tradecraft with Minifilters	116
File Detections	116
Named Pipe Detections	117
Evading Minifilters	118
Unloading	118
Prevention	120
Interference	121
Conclusion	122

7 NETWORK FILTER DRIVERS 123

Network-Based vs. Endpoint-Based Monitoring	124
Legacy Network Driver Interface Specification Drivers	125
The Windows Filtering Platform	126
The Filter Engine	127
Filter Arbitration	127
Callout Drivers	128
Implementing a WFP Callout Driver	128
Opening a Filter Engine Session	128
Registering Callouts	129
Adding the Callout Function to the Filter Engine	130
Adding a New Filter Object	130
Assigning Weights and Sublayers	133
Adding a Security Descriptor	134
Detecting Adversary Tradecraft with Network Filters	135
The Basic Network Data	135
The Metadata	137
The Layer Data	138
Evading Network Filters	139
Conclusion	142

8 EVENT TRACING FOR WINDOWS 143

Architecture	144
Providers	144
Controllers	149
Consumers	151
Creating a Consumer to Identify Malicious .NET Assemblies	151
Creating a Trace Session	151
Enabling Providers	153
Starting the Trace Session	155
Stopping the Trace Session	157
Processing Events	158
Testing the Consumer	164

Evading ETW-Based Detections	165
Patching	165
Configuration Modification	165
Trace-Session Tampering	166
Trace-Session Interference	166
Bypassing a .NET Consumer	166
Conclusion	170

9

SCANNERS

171

A Brief History of Antivirus Scanning	172
Scanning Models	172
On Demand	173
On Access	173
Rulesets	174
Case Study: YARA	175
Understanding YARA Rules	175
Reverse Engineering Rules	177
Evading Scanner Signatures	179
Conclusion	182

10

ANTIMALWARE SCAN INTERFACE

183

The Challenge of Script-Based Malware	184
How AMSI Works	186
Exploring PowerShell's AMSI Implementation	186
Understanding AMSI Under the Hood	189
Implementing a Custom AMSI Provider	193
Evading AMSI	196
String Obfuscation	197
AMSI Patching	197
A Patchless AMSI Bypass	199
Conclusion	199

11

EARLY LAUNCH ANTIMALWARE DRIVERS

201

How ELAM Drivers Protect the Boot Process	202
Developing ELAM Drivers	203
Registering Callback Routines	203
Applying Detection Logic	206
An Example Driver: Preventing Mimidrv from Loading	207
Loading an ELAM Driver	208
Signing the Driver	208
Setting the Load Order	210
Evading ELAM Drivers	212
The Unfortunate Reality	213
Conclusion	213

12		
MICROSOFT-WINDOWS-THREAT-INTELLIGENCE		215
Reverse Engineering the Provider		216
Checking That the Provider and Event Are Enabled		216
Determining the Events Emitted		218
Determining the Source of an Event		221
Using Neo4j to Discover the Sensor Triggers		221
Getting a Dataset to Work with Neo4j		222
Viewing the Call Trees		223
Consuming EtwTi Events		226
Understanding Protected Processes		227
Creating a Protected Process		229
Processing Events		234
Evading EtwTi		234
Coexistence		234
Trace-Handle Overwriting		235
Conclusion		237
13		
CASE STUDY: A DETECTION-AWARE ATTACK		239
The Rules of Engagement		240
Initial Access		240
Writing the Payload		240
Delivering the Payload		242
Executing the Payload		243
Establishing Command and Control		244
Evading the Memory Scanner		246
Persistence		246
Reconnaissance		249
Privilege Escalation		250
Getting a List of Frequent Users		251
Hijacking a File Handler		251
Lateral Movement		258
Finding a Target		259
Enumerating Shares		260
File Exfiltration		262
Conclusion		263
APPENDIX		
AUXILIARY SOURCES		265
Alternative Hooking Methods		265
RPC Filters		266
Hypervisors		269
How Hypervisors Work		269
Security Use Cases		270
Evading the Hypervisor		271
INDEX		273