

INDEX

A

- access mask, 67–68
- AcquireFileForNtCreateSection
 - callback, 105
- address space layout randomization (ASLR), 87
- advapi ETW functions, 146–149, 211, 253
- agent design, 9–11
 - advanced, 11
 - basic, 9
 - intermediate, 10
- alertable state, 86–87, 90
- algorithmic encoding, 185
- altitude, 106
 - of popular EDRs, 108
- Alvarez, Victor, 175
- AMSI, 144, 183, 250
 - checking the trust level for, 190
 - creating a new session of, 187
 - initializing, 189
 - patching, 197–199
 - scanning the buffer of, 187–189
- AMSI_ATTRIBUTE enumeration, 194–195
- amsi!CAmsiAntimalware::Scan()
 - function, 192
- amsi.dll*, 189
- AMSI scan result values, 188
- Ancarani, Riccardo, 118
- anonymous pipes, 118
- Antimalware Scan Interface. *See* AMSI
- anti-ransomware, 11, 117
- antivirus scanning engine, 172
- AppInit_Dlls infrastructure, 22
- APT3, 247
- Arbitrary Code Guard (ACG), 91
- assembly GUID, 180
- atilk64.sys*, 235
- ATT&CK evaluations, 247
- Awesome Procedures on Cypher, 222

B

- bastion, 85
- BDCB_CALLBACK_TYPE enumeration, 204
- BDCB_CLASSIFICATION enumeration, 206
- BDCB_IMAGE_INFORMATION structure, 204, 206
- BdCbStatusUpdate events, 204
 - values of, 206–207
- Beacon
 - executing PowerShell with, 253
 - memory allocation, 234
 - named pipes, 117–118
 - postexploitation with, 249–250
- beaconing, 11, 13, 85, 125, 142, 245–246
- Beacon Object File (BOF), 59, 250
- Bifrost, 8–9
- BitLocker, 213
- Blackbone, 56
- BloodHound, 166, 222
- Blue Screen of Death, 88
- bootkits, 212–213
- Boot Manager, 213, 229
- bootmgfw.efi*, 213
- boot-start service, 210
- boundary-oriented architecture, 124
- Bratus, Sergey, 213
- breakpoint (bp), 34, 83, 167
- Bring Your Own Vulnerable Driver (BYOVD), 212
- brittle detections, 7
- Bundesamt für Sicherheit in der Informationstechnik* (BSI), 206
- bypasses, types of, 12

C

- CALLBACK_ENTRY_ITEM structure, 65
- CallTreeToJson.py*, 222
- canary file, 117

Chester, Adam, 42–43, 91, 166
choke point, 124
Christensen, Lee, 249, 266
Ciholas, Pierre, 74
ci!g_CiOptions overwriting, 101
classify callouts, 135
chr.dll, 80, 166–167, 169
Cobalt Strike, 59, 80, 104, 117–118, 151, 234, 249–250, 253
Cobalt Strike Beacon. *See* Beacon
Coburn, Ceri, 199
Code Integrity, 81–82
Code Signing ECU, 208, 230
command and control, establishing, 244–245
command line tampering, 41–45
common language runtime, 80, 164, 167
COMPlus_ETWEnabled environment variable, 165
COM server, 193, 247–248, 257
conditional jump, 23
ConsoleCtrlHandler() routine, 158
Control Flow Guard (CFG), 189–190
ConvertFrom-SddlString cmdlet, 145–146
countersignature, 202–203, 210
CREATE_SUSPENDED flag, 44
CreatingThreadId field, 37–38, 48
Cryptography API: Next Generation (CNG), 206
Cypher, 222–223, 226

D

dbghelp!MiniDumpWriteDump() function, 116, 181
debugging symbols, 256
debug registers, 199
default-username-was-already-taken, 252
DefenderCheck, 178–179
Delpy, Benjamin, 100
detections, 4
detour function, 19–22
Detrahere, 201
DigitalOcean, 245
dnSpy, 179–180, 186
download cradle, 185, 253

Driver Signature Enforcement, 169, 212
Duggan, Daniel, 198

E

Early Launch Antimalware. *See* ELAM
early-launch drivers registry hive, 205
Early Launch ECU, 208
Early-Launch load-order group, 211
edges, 222
ELAM, 202, 205, 209
 callback routines, 203–206
 developing, 203
 loading a driver, 208–212
 load order, 210–212
 object identifiers, 209
 performance requirements, 205
 signatures, 205
 registration, 229
Elastic detection rules, 8
Empire, 184
encryption, 185
endpoint-based network monitoring, 124–125
Enhanced Key Usage (EKU)
 extensions, 208, 229
entropy, 253
enumerating shares, 260–262
environmental keying, 254
EPROCESS structure, 53–54, 57, 227
 process-image information of, 55
ESpecter bootkit, 212–213
ETW, 143–144, 146–147, 149, 151, 155, 157–158
 consumers, 151
 controllers, 149
 emitting events, 146
 locating event sources, 147
 processing events, 158
 providers, 144
 sensors, 221, 225
 starting a trace session, 155
 stopping a trace session, 157
ETWEnabled registry key value, 165
ETW_REG_ENTRY structure, 165, 235–236
EtwTi. *See* Microsoft-Windows-Threat-Intelligence
EtwTim sensor prefix, 221
evading function hooks, 24

- evading memory scanners, 246
- evading network filters, 139–142
- evading object callbacks, 68–69
- eventcons.h*, 159
- EVENT_DESCRIPTOR structure, 158
- EVENT_ENABLE parameters, 154
- event ID 4663, 261
- event ID 5140, 261
- event object, 158
- EVENT_RECORD structure, 158
 - members of, 158–169
- Event Tracing for Windows. *See* ETW
- Excel Add-In (XLL) files, 240, 242–243
- execute-assembly Beacon command, 59, 118
- EX_FAST_REF structures, 36
- Extended Validation (EV) certificate, 212

F

- Fast I/O, 105
- fault tolerance, 262
- file detections, 116–117
- file-digest algorithm, 210, 230
- file exfiltration, 262–263
- file handler, hijacking a, 251–258
- file signature, 262
- FileStandardInformation class, 57
- filesystem canaries, 116–117
- filesystem minifilter drivers, 103, 106, 108, 114–116, 118
 - activating, 114–115
 - altitudes of, 119
 - architecture, 106–108
 - callback routines, 106, 110, 113–114
 - detecting adversary tradecraft, 116–118
 - evading, 118–120
 - FLT_ structures, 111–114
 - load-order groups, 107
 - managing, 115–116
 - unloading, 113, 119
 - writing, 108–110
- filesystem stack, 104–106
- filter manager, 104
- FilterUnloadCallback callback, 114
- FindETWProviderImage*, 147

- firmware rootkits, 212
- Fix, Bernd, 172
- FLT_CALLBACK_DATA structure, 111, 121
 - important members of, 111–112
- FLTFL_REGISTRATION structure, 109
 - fields of, 109–115
- FltLib, 116
- fltmc.exe*, 116, 118
- fltmgr! minifilter functions, 108, 113–115, 121, 128–129
- fltmgr.sys*, 105
- fork&run, 58–59, 91, 199
- F-PROT, 172
- FRISK Software, 172
- FSFilter Activity Monitor, 107
- FSFilter Anti-Virus, 107
- function hooks
 - detecting, 22–24
 - evading, 24
- FWP_MATCH_TYPE enumeration, 131
- FWPM structures, 130–131, 134, 136
- FwpsCalloutClassifyFn callout function, 135
- FWPS structures, 129, 135–139
- fwpuclnt! filter engine functions, 130
- FWP_VALUE structure, 136

G

- GenerateFileNameCallback function, 114
- Get-SmbShare PowerShell command, 260
- Get-WmiObject PowerShell command, 260
- Ghidra, 221
- global uniqueness, 243
- Golchikov, Andrey, 165
- Graeber, Matt, 197
- Green, Benjamin, 74
- gTunnel, 85–86

H

- Halls, Dylan, 169
- HAMSICONTEXT handle, 191–192
- HAMSISESSION handle, 191, 193
- handle duplication, 63–64, 68
- hardware breakpoint, 199
- hijacking a file handler, 251–258
- HKU hive, 254

hThemAll.cpp, 77
Hypervisor-Protected Code Integrity (HVCI), 101

I

IAntimalware interface, 189
IAntimalwareProvider::Scan(), 192
IDA, 221–222
IMAGE_INFO structure, 81
image-load notifications, 79

- collecting information, 81
- evading, 84
- registering a callback routine, 80
- viewing signature levels, 80–82

Impacket, 84
INF file, 115
InfinityHook, 169
initial access, 240–246
InlineExecute-Assembly Beacon object file, 250
interrupt request levels, 88
Interrupt Request Packets (IRPs), 105
Invoke-Expression PowerShell command, 185
I/O completion port, 75–76
iorate.sys, 208
IRQL_NOT_LESS_OR_EQUAL bug check, 88

J

jitter, 245
JMP instruction, 19
JNE instruction, 23
Johnson, Jonathan, 12

K

KAPC_ENVIRONMENT enumeration, 89
KAPC injection, 22, 79, 86–91

- mitigation of, 90
- registration functions, 89–90

Kerberoasting, 8, 13
kernel32! functions, 9

- allocating memory on the heap, 47
- creating a base service, 233–234
- creating a process, 45
- creating a remote thread, 244
- creating a transaction object, 51
- duplicating a handle, 73, 251
- installing an ELAM certificate, 232

- loading a library, 87
- locking a file, 110
- mapping a portion of a file, 83
- opening a process, 18, 47
- placing a thread in an alertable state, 86
- populating a process attribute list, 46
- reading process memory, 43
- resuming a suspended thread, 44
- rolling back a transaction, 51
- setting a process mitigation policy, 91
- writing process memory, 44

kernel address space layout randomization (KASLR), 236
kernel asynchronous procedure call (KAPC) injection, 22, 79, 86–91

Kernel Driver Utility (KDU), 169
kernel-mode driver, 5, 9–11, 33
Kernel Patch Protection (KPP), 19
key derivation, 255
known unknowns, 247–248
Kogan, Eugene, 51
Korkin, Igor, 165

L

Landau, Gabriel, 52
language emulation, 184–185, 197
lateral movement, 124, 258–262
layered network drivers, 125
legacy filters, 104–106
Leidy, Emily, 247
LetsEncrypt SSL certificate, 245
lha.sys, 235
Lieberman, Tal, 51
LIST_ENTRY structure, 65
living-off-the-land, 184
LLVM, 256
loading an ELAM driver, 208–212
logman, 149–151
lsass.exe, 34, 67–69, 71–73

M

magic bytes, 262–263
mailslots, 103–104, 116
major functions and their purposes, 110

makecert.exe, 208–210
 Malleable profile, 59, 117
 Managed Object Format (MOF), 146
 manifests, 146
 Marnerides, Angelos K., 74
 Matrosov, Alex, 213
 Measured Boot, 206–207, 213
 measurements, 213
memcpy() function, 169, 243
 memory scanner, evading, 246
 Metasploit, 84
 Michael, Duane, 186
 Microsoft Defender, 115

- AMSI provider, 186
- ELAM, 205
- filters, 141
- minifilter, 115
- object callback routines, 66
- process protection, 228
- ruleset, 177
- scanning, 173

 Microsoft Defender for Endpoint (MDE), 215
 Microsoft Defender IOOfficeAntivirus, 186
 Microsoft Detours, 19
 MICROSOFTELAMCERTIFICATEINFO ELAM driver resource, 229
 Microsoft Macro Assembler (MASM), 25
 Microsoft Virus Initiative (MVI), 202
 Microsoft-Windows-DNS-Client, 245
 Microsoft-Windows-DotNETRuntime, 144, 151, 155, 162, 164, 166, 249
 Microsoft Windows Early Launch Anti-malware Publisher, 202
 Microsoft-Windows-Kernel-Process, 242, 245
 Microsoft-Windows-Security-Auditing, 261, 268
 Microsoft-Windows-Security-Mitigations sensors, 221
 Microsoft-Windows-SMBClient, 261
 Microsoft-Windows-Threat-Intelligence, 219

- consuming events, 226
- ETW provider, 216
- evasion, 234–237
- event sources, 221
- sensors, 221

 Microsoft-Windows-WebIO, 14, 145, 245
 Mimidrv, 100–101, 207–208
 Mimikatz, 7, 68–69, 72–73, 180–181, 207
 minifilter. *See* filesystem minifilter drivers
 Ministry of State Security (China), 247
 @modexpblog, 27
 mojo, 117
 MOV instruction, 236
MpClient.dll, 192–193
MpOav.dll, 186, 190–191
msfs.sys, 104
msmpeng.exe, 228
mssecflt.sys, 212
mup.sys, 208
 mutexes, 69, 71, 114

N

@n4r1b, 212
 named pipes, 103

- detections, 117–118

 NDIS, 125–126

- interaction between types of drivers, 126
- types, 125

 Neo4j, 221–223
 NET_BUFFER structure, 138
net.exe, 260
 NetFilter rootkit, 212
 netsh command, 139–140
 network-based monitoring, 124–125
 Network Driver Interface Specification. *See* NDIS
 network filter drivers, 123

- callouts, 128
- detecting adversary tradecraft, 135
- evading, 139–142
- filter arbitration, 127
- filter engine, 127
- legacy driver types, 125

 network intrusion detection system (NIDS), 124
 New-SelfSignedCertificate cmdlet, 230
 New Technology File System (NTFS), 103
 nodes, 222
 NonPagedPool memory, 88
 notification callback routines, 33–34
npfs.sys, 103

- ntddk.h* header, 82
- ntdll.dll*, 22–31, 83, 86–87
 - commonly hooked functions, 19
 - getting function pointers, 168–169
 - remapping, 28–31
- ntdll! functions
 - allocating virtual memory, 23
 - creating a file, 20
 - creating a process, 31, 35, 51
 - creating a thread, 26
 - loading a DLL, 87
 - querying an image, 57
 - querying an object, 71
 - querying a process, 43
 - querying system information, 67, 237
 - registering an ETW event, 147
 - setting a file for deletion, 53
 - writing an ETW event, 167
- nt!ETW functions
 - ETW providers
 - enabling, 216
 - registering, 217
 - non-data requests, collecting information about, 105
- ntfs.sys*, 103–104, 106
- NtObjectManager, 140–141
 - Get-FwCallout cmdlet, 141
 - Get-FwFilter cmdlet, 140
- nt!_OBJECT_TYPE structure, 65–66
- ntoskrnl.exe*, 101, 148, 222, 236

O

- obfuscation, 119, 172, 185, 197
- object callbacks, 62
 - evading, 68–69
 - structures, 62–63, 66–68, 73
- object manager, 61
- objects, 61
- ObjectType structure, 63–64, 67
 - supported values, 63
- on-access scanning, 173–174
- on-demand scanning, 173
- OperationRegistrationCount member, 62, 64
- optional callbacks, 114
- OriginalDesiredAccess member, 68

P

- PagedPool memory, 88
- page hashes, 210
- Palantir, 165
- ParentImage property, 39
- ParentProcessId field, 38, 48
- parent process spoofing, 47
- PatchGuard, 19, 169
- patching, 19, 165, 167–169
- payloads
 - delivering, 242
 - encryption, 242
 - writing, 240
- PEB, 42
 - returning the image path from, 55
- PebBaseAddress member, 44
- PerfView, 219
- persistence, 246–249
 - metrics, 246–247
- PFLT_FILTER filter pointer, 115
- pico, 56
- Plug and Play manager, 207
- post-operation callbacks, 34, 113–114
- PPL, 227
- pre-operation callbacks, 34, 110–113
 - supported values, 112–113
- privilege escalation, 250
- ProcDump, 68
- PROCESS_ALL_ACCESS right, 68
- ProcessBasicInformation information class, 44
- process callback routine, registering, 35–36
- PROCESS_CREATE_PROCESS right, 47
- process doppelganging, 51
- PROCESS_DUP_HANDLE right, 251
- process environment block (PEB), 42–44, 50, 53–58
- Process Explorer, xxii, 73, 227, 234
- process ghosting, 52–53, 57
- Process Hacker, 42, 44, 47–48
- process herpaderping, 52
- process hollowing, 50
- process-image modification, 49–58
 - detecting, 53–57
 - doppelganging, 51
 - ghosting, 52

- herpaderping, 52
- hollowing, 50
- process notifications, 34–39
 - creation events, collecting
 - information from, 37–39
 - registering, 35–36
 - viewing callbacks, 36–37
- ProcessParameters PEB field, 42–44, 55, 57
- process protections, 227–228
- PROCESS_QUERY_INFORMATION right, 72
- PROCESS_VM_READ right, 47, 68–69, 71
- ProgID, 256–257
- programmatically identifier, 256
- protected processes, 227–228
- Protected Process Light (PPL), 227
- Proxifier, 85
- Proxychains, 85
- proxying architecture, 84–85
- PsExec, 5, 258

Q

- quote, 213

R

- real-time consumer, 151
- real-time protection, 173–174
- reconnaissance, 249–250
- reflection, 197, 250
- REGHANDLE parameter, 149, 217, 235–236
- registering a boot-start callback
 - routine, 203–204
- registering an image callback routine, 80
- registering a process callback routine, 35–36
- registering a registry callback routine, 92–93
- registering a thread callback routine, 39–40
- RegistrationContext member, 62
- registry notifications, 79, 91, 95–96
 - evading, 96
 - mitigating performance
 - challenges, 95
 - registering a callback, 92–95
- REG_NOTIFY_CLASS registry class, 92–94, 96
- remapping *ntdll.dll*, 28–31

- remote thread creation, detecting, 40–41
- ResourceFileName registry key value, 147
- RFC 3161, 210
- robust detections, 7
- Rodionov, Eugene, 213
- Roedig, Utz, 74
- Rubeus, 181
- rulesets, 174–175
- rules of engagement, 240

S

- sacrificial process, 14, 58–59, 91, 118, 249, 253–254
- Saha, Upayan, 235
- scanner, 171
 - evading, 179–181
 - rulesets, 174–175
- scanning models, 172–173
- schedsvcs.dll*, 148
- scheduled tasks, 247–248
- Schroeder, Will, 172, 249
- Seatbelt, 151, 164, 249–251, 254–255, 259–261
- sechost! trace functions 146, 149, 151, 155
- Secure Boot, 22
- Secure ETW, 11, 226–227, 268
- security descriptor, 130, 134, 141, 145–146
- Security Events Component
 - Minifilter, 212
- self-describing events, 146
- SeLoadDriverPrivilege token privilege, 100, 118
- sensors, 3–4
- ServiceGroupOrder*, 211
- Set-FileAssoc.ps1*, 252
- sgrmagent.sys*, 211
- SharpHound, 166, 258
- shell preview handlers, 247–248
- shims, 126
- shutdown handlers, 201
- signtool.exe*, 209–210, 230
- SMB, 260–261
- socks command, 84
- software restriction policy, 81
- STARTUPINFO structure, 46

- STATUS_FILE_DELETED* error, 53
- STATUS_VIRUS_INFECTED* failure status, 121
- string mangling, 253, 256
- string obfuscation, 197
- SubjectUserSid field, 254
- Such, Jose Miguel, 74
- Suhanov, Maxim, 213
- syscall, 18–20
 - dynamically resolving, 27
 - making direct, 25–27
- Sysmon, 38, 40–41, 118–120
- SysmonDrv, 118–120
- system access control list (SACL), 145, 261
- System Guard Runtime Monitor, 212
- SystemHandleInformation information class, 70
- System.Management.Automation.dll*, 186
- System Service Dispatch Table (SSDT), 18–19, 218, 221
- SysWhispers, 26–27

T

- tamper sensor, 255
- tbs!Tbsi_Revoke_Attestation()
 - function, 207
- tcpip.sys*, 126
- tcpip6.sys*, 126
- tdh! ETW functions, 159, 161–163
- telemetry, 2
 - auxiliary sources of, 266–271
 - types collected, 9–12
- Teodorescu, Claudiu, 165
- TEST instruction, 23
- thread callback routine, registering, 39–40
- thread notifications, 39
- ThreatIntProviderGuid GUID, 217
- threat names, 178
- Thuraisamy, Jackson, 26
- Time-Stamp Protocol, 210
- To-Be-Signed (TBS) hash, 230–231
- Trace Data Helper (TDH) APIs, 146–147, 159
- TRACE_EVENT_INFO structure, 160
- TRACEHANDLE parameter, 153, 156, 165–166
- TraceLogging, 146–147

- trace sessions, 149–150, 165–166
- trampoline, 19
- Transactional NTFS (TxF), 51
- transport protocol stack, 125
- trap flag, 24
- Truncer, Chris, 172
- Trusted Boot, 202
- Trusted Platform Module (TPM), 206–207, 213
- tunneling, 84–86

U

- unconditional jump, 19
- Uniform Resource Identifier (URI), 244
- UserChoice hash, 252–253

V

- Vazarkar, Rohan, 222
- vectored exception handler (VEH), 24, 199
- Veil, 172
- Vienna virus, 172
- VirTool, 178
- virtual address descriptor (VAD)
 - tree, 56
- VirusTotal, 174–175

W

- WdBoot, 211
- wdboot.sys*, 206
- WdFilter, 115
- WdFilter.sys*, 37, 66
- wdm.h*, 110
- Webclient class, 185
- werfault.exe*, 41
- WerSvc, 41
- WEVT_TEMPLATE*, 147
- WFP, 123, 126–128, 134, 142, 268
 - architecture, 126–127
 - base filtering engine, 127
 - benefits, 126
 - callout drivers, 128
 - implementing, 128–134
 - default filter security descriptor, 134
 - filter arbitration, 127–128
 - filter conflict, 142
 - filter engine, 127–128
 - FWPM structures, 130–131, 134, 136

- layers and sublayers, 127
- weight, 127
- white cell, 240, 242
- whoamsi project, 186
- Win32k, 215
- Win32_SecurityDescriptorHelper WMI class, 146
- Windows bootloader, 211
- Windows Error Reporting, 41
- Windows Filtering Platform.
See WFP
- Windows firewall, 126, 134
- Windows Hardware Quality Labs (WHQL), 212
- Windows Software Trace Preprocessor (WPP), 146
- Windows Subsystem for Linux (WSL), 36
 - winload.efi*, 211
- Winter-Smith, Peter, 24
- WNODE_HEADER structure, 153

- WPP_INIT_TRACING macro, 146
- Wright, Mike, 172
- WS2_32!send() function, 126

X

- XLL files, 240, 242–243
 - functions of, 240, 242
- Xperf, 149

Y

- YARA format, 174–178
 - alternatives, 176
 - conditions, 177
 - jumps, 176
 - rules, 175–177
 - wildcards, 176–177

Z

- Zacinlo rootkit, 201
- Zhang, Jiajie, 74