FOR610 – Reverse-Engineering Malware: Malware Analysis Tools and Techniques

Topics

Malware Analysis Fundamentals

- Introduction 610.1–4-22
- Malware Analysis Lab 610.1–24-36
- Static Properties Analysis 610.1–38-50
- Behavioral Analysis Essentials 610.1–52-77
- Code Analysis Essentials 610.1–79-110
- Exploring Network Interactions 610.1–112-132

Reversing Malicious Code

- Core Reversing Concepts 610.2–3-59
- Reversing Functions 610.2–61-96
- Control Flow In-Depth 610.2–98-134
- API Patterns in Malware 610.2–136-144
- 64-bit Code Analysis 610.2–146-161

Analyzing Malicious Documents and Scripts

- Malicious PDF File Analysis 610.3–3-35
- VBA Macros in MS Office Docs 610.3–37-78
- Examining Malicious RTF Files 610.3–80-100
- Deobfuscating Malicious Javascript 610.3–102-118

In-Depth Malware Analysis

- Recognizing Packed Malware 610.4–3-13
- Getting Started with Unpacking 610.4–15-26
- Using Debuggers for Dumping 610.4–28-42
- Debugging Packed Malware 610.4–44-54
- Analyzing Multi-Technology MW 610.4–56-104
- Examining .NET Malware 610.4–106-132
- Understanding Code Injection 610.4–134-152

Examining Self-Defending Malware

- Debugger Detection and Data Prot 610.5–3-41
- Unpacking Process Hollowing 610.5–43-59
- Detecting the Analysis Toolkit 610.5–61-92
- Handling Misdirection Techniques 610.5–94-142
- Unpacking by Anticipating Actions 610.5–144-167

Categories

Tools

- `accept-all-ips` 🐧 610.1–127, 129
  - 610.3–62
- AlgoVPN 610.1–19
- Any.run 610.1–13
- API Monitor 610.1–102-103
- `base64dump` 🐍 610.3–69-71, 74
  - 610.4–73
- `bbcrack` 🐍 610.5–19
- Binary Ninja 610.1–81
- `box-js` 🐧 610.3–118
- `brxor` 🐍 610.5–18
- Burp 610.3–19
- `bytehist` 🐧 610.4–11
- Capa 610.1–83, 87-89
  - 610.5–45-46
- CAPE 610.1–13
  - Sandbox 610.1–34
- CFF Explorer 610.4–19-20, 36
- ChatGPT 610.1–21
- CloneZilla 610.1–34
- `code` 🐧 610.1–76
- Cscript 🪟 610.3–106, 110, 112
- `curl` 🐧 610.3–19
- Cutter 610.1–81
- CyberChef 610.1–105-109
  - 610.3–104
- Dd 610.1–34
- de4dot 610.4–108, 112, 129
- DIE 610.1–49
  - 610.4–12
- dnSpyEx 610.4–108-109, 113, 116-126
  - Locals 610.4–126
  - Modules 610.4–120-121
- DotDumper 610.4–108, 131
- dotPeek 610.4–108
- Ether 610.4–17
- `evilclippy` 🐧 610.3–57
- Exeinfo PE 610.1–49
  - 610.4–12
- `exiftool` 🐧 610.3–54
- `fakedns` 🐧 610.1–68, 116
- `feh` 🐧 610.3–43
- Fiddler 610.1–124-125
  - 610.3–19-21, 26
- `file` 🐧 610.3–89
- FileScan.io 610.1–13
- FLOSS 610.5–25-27
- FOG 610.1–34
- `format-bytes` 🐍 610.3–88
- Ghidra 610.1–81
  - 610.2–6-59, 139, 154
  - 610.4–139
  - 610.5–117
  - CodeBrowser 610.2–10
  - Confused 610.5–119
  - Decompiler 610.4–143
  - Find Location (shortcut) 610.4–148
  - Saved Work 610.2–133
- `grep` 🐧 610.3–49, 51
- `gunzip` 🐧 610.3–71
- HashSets 610.1–13
- Hopper 610.1–81
- `httpd` 🐧 610.1–72, 116-117, 129
- Hybrid Analysis 610.1–13, 15
- Hyper-V 610.1–28
- `iconv` 🐧 610.4–71
- IDA 610.1–81
- ILSpy 🪟 610.4–108, 113, 128, 130
- `ilspycmd` 🐧 610.4–108, 111, 113
- Imports Fixer 610.4–22
- InetSim 610.1–120-123
- Intezer Analyze 610.1–13
- `iptables` 🐧 610.1–127-129

`jq` 🐧 . . . 610.1–86
`js` 🐧 . . . 610.3–106-107, 114
`js-beautify` 🐧 . . . 610.3–104
. . . 610.4–71
LibreOffice . . . 610.3–44
`logman` 🪟 . . . 610.3–109
Malware Hash Registry . . . 610.1–13
MetaDefender . . . 610.1–13
`nc` 🐧 . . . 610.1–127
. . . 610.3–62
`numbers-to-string` 🐍 . . . 610.3–59
`oledump` 🐍 . . 610.3–47, 51, 58-59, 66-70, 85, 87-88
OleTools . . . 610.3–45
`olevba` 🐧 . . . 610.3–45
OllyDumpEx (x..dbg plugin) . . . 610.4–29, 36-42
. . . 610.5–108, 111
Open Threat Exchange . . . 610.1–13, 16
OpenVPN . . . 610.1–19
`pdf-parser` 🐍 . . . 610.3–9-17
`pdfid` 🐍 . . . 610.3–8, 29
`pdftk` 🐧 . . . 610.3–35
PE Tools . . . 610.4–22
`peframe` 🐧 . . . 610.1–48
`pestr` 🐧 . . . 610.1–44
PEStudio . . . 610.1–46-49, 92, 138, 142
. . . 610.4–10, 102, 109-110, 127
. . . 610.5–37, 58, 113
`pe_unmapper` . . . 610.5–163-164
Pinpoint . . . 610.3–19
PowerShell ISE 🪟 . . . 610.4–76-79
ProcDOT . . . 610.1–53, 64-66
. . . 610.4–61-64
Process Hacker . . . 610.1–53-54, 96, 119
. . . 610.3–62
. . . 610.4–58-59
Process Monitor . . . 610.1–53, 55, 62-65
. . . 610.4–58-60
`qpdf` 🐧 . . . 610.3–35
Quttera . . . 610.1–13, 20
`rar` 🐧 . . . 610.3–98-99
`re-search` 🐍 . . . 610.3–51
`reg_export` 🪟 . . . 610.4–69
Registry Editor 🪟 . . . 610.4–68
RegShot . . . 610.1–53, 56, 60-62
REM . . . 610.1–2
`rtfdump` 🐍 . . . 610.3–83-84, 89-91
`rtfobj` 🐧 . . . 610.3–84
Runsc32 . . . 610.3–97
. . . 610.4–82-86
Runsc64 . . . 610.3–97
ScDbg 🪟 . . . 610.4–76, 80-81
Scdbgc . . . 610.3–75, 95-96
Scout . . . 610.3–19
Scylla . . . 610.4–22-24, 29, 36
Scylla (x..dbg plugin) . . . 610.4–40-41
. . . 610.5–112, 164
ScyllaHide (x..dbg plugin) . . . 610.5–15, 28, 73, 76
SecurityTrails . . . 610.1–13, 16
`sed` 🐧 . . . 610.3–51, 60
`set-static-ip` 🐧 . . . 610.1–126
`SetDllCharacteritics` 🪟 . . . 610.4–19-20
`sets` 🐍 . . . 610.3–51
Shodan . . . 610.1–13
Speakeasy . . . 610.1–83-86
SpiderMonkey . . . 610.3–106-107, 114
. . . 610.4–71
`sshd` 🐧 . . . 610.4–70
`strdeob` 🐪 . . . 610.5–24-25
`strings` 🐧 . . . 610.1–44
. . . 610.3–54, 74
. . . 610.5–17
Sysmon . . . 610.2–19
TcpLogView . . . 610.1–53
ThreatFox . . . 610.1–13
Thug . . . 610.3–19
`tr` 🐧 . . . 610.4–71
`trid` 🐧 . . . 610.3–41, 84
UnpacMe 🌐 . . . 610.4–17
`unzip` 🐧 . . . 610.3–43
UrlScan.io . . . 610.1–13, 20
VBoxCloak . . . 610.5–65
ViperMonkey . . . 610.3–56
VirtualBox . . . 610.1–28
VirusTotal . . . 610.1–13-14, 20
Visual Studio Code . . . 610.1–76, 84
VMWare . . . 610.1–28
VMWareCloak . . . 610.5–65
`wget` 🐧 . . . 610.3–19
Winbindex . . . 610.1–13
Windbg . . . 610.1–81
WinSCP 🪟 . . . 610.4–70, 74
Wireshark . . . 610.1–53, 57, 67-75, 129
X32dbg . . . 610.4–83
. . . 610.5–28-36
  SEH Chain . . . 610.5–99
X64dbg . . . 610.1–81, 90-99
. . . 610.4–30-31
. . . 610.5–5, 10-11
  `cleardb` . . . 610.4–47
  Unpacking Scripts . . . 610.4–17
xAnalyzer (x..dbg plugin) . . . 610.5–152
`xmldump` 🐍 . . . 610.3–55
`xorsearch` 🐧 . . . 610.3–92-95
. . . 610.5–17
`xxd` 🐧 . . . 610.1–76
`yara-rules` 🐧 . . . 610.3–76
`zipdump` 🐍 . . . 610.3–43, 55

FOR610 – Reverse-Engineering Malware: Malware Analysis Tools and Techniques

API Calls

- BlockInput 610.5–75-76
- CallNextHookEx 610.5–70-71
- CheckRemoteDebuggerPresent 610.5–12
- CountClipboardFormats 610.5–65
- CreateFileA 610.1–85
  - 610.2–141
  - 610.3–96
- CreateMutexA 610.2–143
- CreateProcess 610.2–37
  - 610.5–48-49, 52
- CreateProcessW 610.2–86-87
- CreateRemoteThread 610.4–136, 138, 151
- CreateThread 610.4–75, 81
- CreateToolHelp32Snapshot 610.2–76
  - 610.4–136, 149-151
  - 610.5–84
- CryptDecrypt 610.1–98-100
  - 610.4–51-53
- CryptDeriveKey 610.1–47
- CryptDestroyKey 610.1–85
- EnumProcesses 610.4–136
- ExitProcess 610.5–74
- FindResourceW 610.2–140-141
- FindWindow 610.5–78
- FindWindowA 610.5–141
- FindWindowW 610.5–79
- FreeLibrary 610.2–140
- GetCursorPos 610.5–65
- GetFileSize 610.3–95-96
- GetLastError 610.1–48
  - 610.2–52
- GetLocalTime 610.5–14
- GetModuleHandle 610.4–151
- GetModuleHandleA 610.5–69
- GetModuleHandleW 610.5–77, 81
- GetProcAddress 610.3–94
  - 610.4–151
  - 610.5–51, 148
- GetSystemTime 610.5–14
- GetTempFileNameW 610.2–62-64
- GetTempPathW 610.2–121-122, 126, 128-129
- GetTickCount 610.1–47, 610.5–65
  - 610.5–14
- HttpOpenRequestA 610.2–46, 50
  - 610.3–75
- HttpSendRequestA 610.2–46, 51-54
  - 610.3–75
- InternetConnectA 610.2–46
- InternetOpenA 610.2–46-49, 55
  - 610.3–75
- InternetReadFile 610.1–89
  - 610.2–56
  - 610.3–75
- IsDebuggerPresent 610.5–6-9, 122-1124
- LoadLibrary 610.4–136, 151
  - 610.5–145
- LoadLibraryA 610.3–75, 94-96
  - 610.4–80
  - 610.5–146-151
- LockResource 610.2–140-141
- NtQueryInformationFile 610.1–21
- NtQueryInformationProcess 610.5–12, 37-40
- NtUnmapViewOfSection 610.5–51-52, 59
- OpenProcess 610.4–136, 142, 148-151
- OutputDebugString 610.5–12
- Process32First 610.4–136, 149-150
  - 610.5–84
- Process32Next 610.4–136, 149-150
- RaiseException 610.2–52
- ReadFile 610.1–92-98
  - 610.3–96
  - 610.3–45
- RegOpenKeyExA 610.2–20-24, 36, 43
  - 610.4–80, 87-91
- RegOpenKeyExW 610.5–88
- RegQueryValueA 610.4–91
- RegSetValueExA 610.1–47
- ResumeThread 610.5–49, 52
- RtlDecompressBuffer 610.5–27, 30-33, 145
- SetWindowsHookExA 610.5–66-69
- ShellExecuteA 610.2–155
- ShellExecuteW 610.2–153
- SizeOfResource 610.2–140-141
- UnhookWindowsHookEx 610.5–71
- URLDownloadToFileA 610.3–45, 52
- VirtualAlloc 610.3–73, 75
  - 610.4–75, 91-100
- VirtualAllocEx 610.4–136, 145-151
  - 610.5–50, 52, 145
- VirtualProtect 610.4–103
  - 610.5–153-155, 158-159
- WriteFile 610.1–85
  - 610.3–96
- WriteProcessMemory 610.4–136, 138, 144-151
  - 610.5–50, 52-54, 145

FOR610 – Reverse-Engineering Malware: Malware Analysis Tools and Techniques

A

- Access Restrictions 610.2–54
- Access Violation → Memory Access Violation ....
- ActiveX Control 610.3–53
- ActiveX Object 610.3–106, 110
- Activity Diagram 610.1–65
- Addressing 610.2–31-33, 149
- AddressOfEntryPoint 610.4–7
- ADMX 610.3–39
- ADS 610.3–39
- Affiliate ID 610.1–130
  - 610.5–17
- AI Tools 610.1–21
- AlgoVPN 610.1–19
- AMSI Monitoring 610.3–109
- Analysis Lab 610.5–62
- Annot (keyword) 610.3–16-17, 33
- ANSI 610.2–20
- Anti-Debugging 610.5–4
  - Defenses 610.5–9, 15
  - Lessons Learned 610.5–41
  - Technique 610.1–48
    - 610.5–6, 12-14
- Anti-Malware Scan Interface → AMSI ....
- Antivirus Scanners 610.1–13
- Any.run 610.1–13
- API Calls 610.1–83-84, 89
  - 610.2–19
  - 610.4–7, 46, 50-51, 136
  - Examples → see **Categories** ....
  - Patterns 610.2–18
  - Risky 610.4–102
    - 610.5–26-27
    - Code Injection 610.4–150-151
    - 610.5–27
    - DLL Loading At Runtime 610.5–146-148
    - Process Hollowing 610.5–59
  - Spying 610.1–102
- API References 610.2–153
- AppData 610.1–39
- Arbitrary Code 610.3–4
- Argument 610.2–70, 76
  - 64-bit 610.2–155
  - Display 610.5–152
  - Example 610.2–81, 89
  - Location 610.2–70, 155
- Argument Register 610.2–25
- Arguments.callee 610.3–118
- Arithmetic Instructions 610.2–38
- Armadillo 610.4–8
- ASLR 610.1–54
  - 610.3–94
  - Disable 610.4–18-20
- Assemblies 610.4–107, 120-121
- Assembly 610.3–93
- Assembly (64-bit) 610.2–149
  - Comparison 32-bit 610.2–159-160
- Assembly Explorer 610.4–117, 121-122
- Assembly.Load 610.4–115-119, 123-124
  - 610.5–145
- Authoritative DNS 610.1–19
- Automated Analysis Sandbox 610.5–62, 72
- Automated Tools 610.1–22
- Automatic Action 610.3–6
- AutoOpen 610.3–45-46, 59, 67
- AVG Security Software 610.5–77
- Avghookx.dll 610.5–77

B

- Backdoor 610.1–114
  - 610.3–61, 76
- Balbuzard Toolkit 610.5–19
- Base 610.2–32-33
- Base64 .... 610.3–67-72
  - 610.4–72-73
- Basic Block 610.2–39
- BAT 610.4–65
- Beaconing 610.1–17
- Behavioral Analysis 610.1–81
  - 610.3–63
  - 610.4–46
  - Findings 610.1–77
  - Initial 610.4–57
  - Tools 610.1–35, 53
- BeingDebugged 610.5–12
- BHO 610.2–147
- Binary Content 610.3–82, 90
- Bit Manipulation Instructions 610.2–38
- Boolean Instructions 610.2–38
- Botnet 610.1–118
- Branching 610.2–39
  - Instructions 610.2–38
  - Primitive 610.2–99
- Breakpoint 610.1–91, 93
  - 610.3–57
  - 610.4–78, 84, 113
  - Defense 610.5–166
  - Hardware 610.4–52
    - 610.5–105, 122, 136, 167
    - Trigger 610.5–160
  - Set → SetBPX ....
  - Software 610.4–52
    - 610.5–122, 167
    - Failure 610.5–122, 167
- Browser Attacks 610.3–103
- Browser Field Formatter 610.2–15
- Browser Script 610.3–106, 118
- Byte-Usage Histogram 610.4–11

C

- C Representation 610.2–6
- CALL 610.2–21-22, 43
  - 610.3–93

  - Step Into/Over 610.4–33
  - Suspicious 610.5–118
  - Tricky Jump 610.5–118
- Call Operand 610.1–98
- Call Stack 610.1–97
- 610.4–89
- 610.5–32
- Calling Conventions → Functions
- Capabilities → Malware Analysis
- CAPE 610.1–13
- Carry Flag 610.2–42
- Cdecl 610.2–72, 74, 85, 92
- CDFV2 610.3–40, 65
- CFBF 610.3–40
- ChatGPT 610.1–21
- CIL 610.4–107
- Cleardb (x64dbg) 610.4–47
- Clipboard 610.5–65
- Cloaking Capabilities 610.5–15
- Cloud Policy 610.3–39
- Cloudflare 610.3–22
- CLR 610.4–107
- CMP 610.2–41, 55
- Cobalt Strike 610.3–76
- Code Analysis 610.1–81
- 610.2–161
  - Approaches 610.1–82
  - Comparison 32/64-bit 610.2–159-160
  - Findings 610.1–110
  - Key Concepts 610.2–59
  - Prior Step 610.4–13
  - Start 610.2–161
  - Tools 610.1–35, 81
  - True Beginning 610.5–115
- Code Block 610.2–127
- Code Injection 610.4–59, 114, 135-136
  - API Calls 610.4–150-151
  - 610.5–27
  - Combined With Packing 610.5–59
  - DLL 610.4–151
  - Example 610.4–142-143
  - Lessons Learned 610.4–152
  - Memory Allocation 610.4–145-147
  - Pattern 610.5–50, 52
  - Security Tools 610.5–77
  - Static Code Analysis 610.4–138
  - Steps 610.4–150
  - Target Processes 610.4–148-149
- Code Lifecycle 610.2–5
- Code Segment 610.2–26
- Code-Data Duality 610.2–35
- Command & Control (C2) 610.1–17, 88
- 610.2–45-47, 56
- Command Execution 610.2–153
- Comment 610.2–44
- Compiled Code → P-code
- Compiler 610.2–5, 131
  - JIT 610.4–107
  - SEH 610.5–98
- Compound Expression 610.2–123-126, 128
- Compressed Source Code → P-code
- Conditional Instructions 610.2–38
- Conditional Jump 610.2–14, 22, 40-44, 99, 109, 122
- Container 610.3–81
- Content-Disposition (header) 610.3–24
- Context 610.2–35
- Control Channels 610.2–45
- Control Variable 610.2–117-118
- Control Words 610.3–82
- Counter-Defenses 610.5–15
- CPU 610.2–24
- CPU Core 610.5–65
- CREATE_SUSPENDED 610.5–49
- Cross-Reference Table 610.3–5
- Cross-References 610.2–161
- CSV 🗎 610.1–64

D

- Data Manipulation Instructions 610.2–38
- Data Segment 610.2–26
- Data Structure 610.2–93-95
- Data Transfer Instructions 610.2–38
- Data Type Manager 610.2–48
- DCTDecode 610.3–13
- Debug Shell 610.4–81
- Debugging 610.1–81-82, 91
- 610.2–5-6, 50, 161
- 610.3–57, 74
- 610.4–29, 108, 113, 125
  - Concealment 610.5–15
  - Detection 610.5–5, 12-14, 37
  - Without Dumping Packed Code 610.4–45-46, 54
- Decision Point 610.2–51-55, 107
- Decoding 610.1–104
- Decoding Patterns 610.5–19
- Decompilation 610.2–6, 120, 129, 158
- 610.4–108
- Default Configuration 610.5–63
- Default Hardware 610.5–63
- Defined Strings 610.2–103-104
- Deobfuscation 610.1–105
- 610.3–56-57, 70-72
- 610.4–108
- Dereferencing 610.2–29-31
- Destination Unreachable 610.1–67
- Detection Evasion 610.3–8, 11, 18, 31, 38, 46, 93-94
- 610.4–59, 115, 135
- 610.5–92, 139
- Device Drivers 610.2–147
- Digital Signature 610.2–147
- 610.3–39
- Direct Addressing 610.2–31
- Disassembly 610.1–81-82, 90
- 610.2–6
- 610.3–74

  - Obfuscated Code 610.5–118
- Disk Cloning 610.1–34
- Disk Drive Identifier 610.5–88
- DLL 610.1–14
  610.2–16-17, 137-139, 152
  - Injection 610.4–151
  - Loading (shellcode) 610.3–94
  - Malicious 610.4–151
- DllCharacteristics 610.4–19
- DNS Emulator 610.1–120
- DNS Leakage 610.1–19
- DNS Tunneling 610.2–45
- Do Not Break 610.5–29, 121
- Do While Loop 610.2–112
  - Example 610.2–120
- Document.write 610.3–107
- Downloader 610.1–122-123
  610.3–45, 103
- Dropper 610.2–140, 144
  610.3–38, 103
- DS (segment register) 610.5–13
- Dump Memory To File 610.4–101
  610.5–36, 57, 162
- Dumping 610.4–22-26, 39-42, 96-101, 125-126
  610.5–36, 111-112, 162-163
  - Debugging Without 610.4–45-46, 54
  - Issues 610.4–26
  - Lessons Learned 610.4–42
- Dword 610.2–28
- Dynamic Code Analysis 610.1–82
  - Shellcode 610.4–87
- Dynamically Allocated Memory 610.2–33
- DynamicBase 610.4–19-20, 23
  610.5–5, 73, 120
- DynamicInvoke ~ 610.4–115
- Dyre 610.5–65

E

- E[ABCD]X 610.2–25
- EAX 610.4–95
  - Set To Zero 610.5–90
- EBP 610.2–68, 70
- EDI 610.2–25
- Effective Address (EA) 610.2–32
- EFLAGS 610.2–26, 41
- EIP 610.2–26
  610.3–93
  - OEP 610.4–37
- Embedded Image 610.3–9, 12
- Embedded Javascript 610.3–4, 7
- Embedded Links 610.3–4, 7, 11
- Embedded Objects 610.3–81-85
- Embedded Program 610.3–4
- Emulation 610.1–82-84
  610.3–74-75, 95
  610.4–80
  - Limitation 610.1–85
  - Tools 610.1–83
- Enable Content 610.3–42-43
- ENTER 610.2–69, 83
- Entropy 610.4–9-10
- Entry Breakpoint 610.5–120
- Entry Point 610.2–13, 16
  610.4–7, 22, 24-26, 29
  - Executing Code Before 610.5–115
- Epilogue 610.2–67, 69
  - Example 610.2–84, 91
- Equate 610.2–36, 49
  610.4–147
- Equation Editor 610.3–85, 87-88
- ESP 610.2–23, 25-26, 67-68, 70, 76
- ETL 🗎 610.3–109
- Eval 610.3–105-107, 112-113
  610.4–68
- Exception Handling → SEH
- EXCEPTION_REGISTRATION 610.5–99
- Exceptions 610.5–29
- Executable File 610.2–5
- Execute Flag 610.4–48
- Execute Until Return 610.4–94, 98
- Execution Flow Concealment 610.5–142
- Execution Monitoring 610.1–53
  610.4–108
- Exfiltration 610.1–17
- Expect-CT 610.3–22
- Exploit 610.3–86, 103
  - Kit 610.1–18
    610.3–19
- Explorer.exe 610.4–59
- Exports 610.2–16
- Extended Function 610.2–20
- External Parameters 610.2–10, 48

F

- False Positive 610.3–92
- Fastcall 610.2–73, 75, 149
- File Handle 610.1–85
- File Reputation 610.1–13
- File System 610.3–40
- Fileless Software 610.4–104
- FileScan.io 610.1–13
- Fill With NOP's 610.5–10
- Filter 610.3–6
- Fix Dump 610.4–24, 41
  610.5–164
- Flags 610.2–37
- FLG_HEAP_* 610.5–13
- Folder Sharing 610.1–29
- Follow Dword in Disassembler 610.5–149
- Follow In Disassembler 610.4–49, 51
  610.5–6
- Follow In Dump 610.4–95, 99
  610.5–31, 55, 155, 159
- Follow In Memory Map 610.4–101

  610.5–36, 56, 162
Follow TCP Stream 610.1–74
For Loop 610.2–112-113
FOSS 610.2–6
Frame Pointer 610.2–68, 154
FS (segment register) 610.3–94
  610.5–12, 99, 128
  Confusion 610.5–128-130
FS:[0] 610.5–99, 128-130
FSG 610.4–8
FUN 610.2–13, 43
Function Call Graph 610.4–148-149
Function Call Tree 610.2–56
Function Graph 610.2–14, 107
Function Size 610.2–58
Functions 610.2–65-79, 96
  Basic Components 610.2–65
  Calling Conventions 610.2–72-75, 85, 92, 149
  Control Transfers 610.2–66, 77
  Enter 610.2–69
  Examples 610.2–79-96
  Leave 610.2–69

G

GAR 🗎 610.2–133
GDT 610.2–48
  610.5–127
General-Purpose Registers → Registers
Generation Number 610.3–6
Get EIP as OEP 610.4–37
  610.5–111
Get Imports 610.4–24, 41
Get RIP as OEP 610.4–37
GetEIP (pattern) 610.3–92-95
  610.4–81, 86
GetTypes ~ 610.4–116
Global Variables 610.2–54
GNU Compiler 610.2–73
GPR 🗎 610.2–133
Graph View 610.2–53
Group 610.3–82
Group Policy 610.3–39
GS (segment register) 610.5–99, 127-128
  Confusion 610.5–128-130
GS:[0] 610.5–128-130
GUI Interactions 610.5–72
Gzip 610.3–70

H

Hard-Coded IP Address 610.1–126, 128
Hardware Breakpoint → Breakpoint
Hardware Characteristics 610.5–92
Hardware Components 610.5–62
Hardware-Based Access Dword Breakpoint 610.5–106
HARDWARE_DEVICEMAP 610.5–88
Hash Values 610.1–47
HashSets 610.1–13
Header File 610.2–49
HKCU Software 610.4–68-69
  Classes 610.4–64
HKCU Software Classes 66-67
HKEY_CURRENT_USER 610.2–36
hModule 610.5–51
Honeypot 610.1–20
Hook 610.5–69
  Information 610.5–70
Hot Patching 610.2–67
HTA 610.4–60, 68
HTML Applications ⊞ 610.4–60, 68
HTML Components 610.3–106
HTTP Headers Spoofing 610.3–19
HTTPS Emulator 610.1–123-124
Hybrid Analysis 610.1–13, 15
Hyper-V 610.1–28
HypeVPN 610.3–26

I

IAT 610.2–17
  610.4–7, 22, 29
  610.5–59
  Absent DLL 610.5–148-149
  AutoSearch 610.4–24, 41
    610.5–112, 164
  Fix 610.4–24, 40-41
    610.5–112, 164
ICMP 610.1–67
IDA 610.2–31
IDS 610.2–152
IEX → Invoke-Expression
If-Else 610.2–99-102
ImageBase 610.4–18, 37
Immediate (Addressing) 610.2–23
Immunity Debugger 610.5–79
Import Table 610.1–47
Index 610.2–32-33
Indicators 610.1–46
Indirect Addressing 610.2–32-33
Indirect Object 610.3–6
Indirection 610.2–29-30
Infection Marker 610.1–15
Inlining 610.2–77-78
InMemory (property) 610.4–121
Input Event Blocking 610.5–73
Instructions 610.2–23-24
  Components 610.2–23
  Equivalents 610.2–41-42, 67, 74, 85, 106
  Types 610.2–38
    Branch 610.2–39
INT3 (opcode) 610.5–167
Intel Syntax (Instruction) 610.2–23
Intellectual Property 610.4–5
Intermediate Bytecode 610.4–107

- Intermodular Calls . . . 610.4–34, 50
  - . . . 610.5–5, 151
- Internal IP Addressing Scheme . . . 610.5–62
- Internet Access . . . 610.1–19
- Internet Explorer . . . 610.2–147
- Intezer Analyze . . . 610.1–13
- Invoke ~ . . . 610.4–115
- Invoke-Expression ▣ . . . 610.3–70
  - . . . 610.4–72
- InvokeMember ~ . . . 610.4–115-116
- IOC . . . 610.1–11, 17, 77
  - . . . 610.2–20, 62
  - . . . 610.3–11
  - Examples . . . 610.1–45, 47
  - Network . . . 610.1–118
- IP Investigation . . . 610.1–13
- IRC . . . 610.2–45
- ISO 639-1 . . . 610.3–55
- ISO 🗎 . . . 610.3–39
- ISO/IEC 32000 . . . 610.3–4
- Isolation Measures . . . 610.1–29

J

- Javascript . . . 610.3–35, 610.3–103
  - Beautify . . . 610.3–104
  - Deobfuscation . . . 610.3–104-111
    - . . . 610.4–69-72
  - Interpreters . . . 610.3–106
  - Object Definitions . . . 610.3–107
- JB/JC . . . 610.2–106, 116
  - Equivalence . . . 610.2–106
- Jcc-Formatted Instructions . . . 610.2–42
- JGE . . . 610.2–114
- JIT → Compiler . . .
- JMP . . . 610.4–31-33, 132
- JNL/JNG . . . 610.2–124
- JPEG . . . 610.3–13
- Jump Concealed . . . 610.5–101
- Jump Types . . . 610.2–40

K

- KdDebuggerEnabled . . . 610.5–79
- Kernel-Mode Debugger . . . 610.5–79
- Kernel-Mode Malware . . . 610.2–147
- Kernel32.dll . . . 610.3–92, 94
  - . . . 610.4–151
  - Reload . . . 610.5–148
- Key Binding . . . 610.5–22-23
- Key Differences 32/64-bit . . . 610.2–160
- Keyboard Event Blocking . . . 610.5–73
- Known Patterns . . . 610.3–76
- KUSER_SHARED_DATA . . . 610.5–79

L

- Lab → Malware Analysis . . .
- Language Code . . . 610.3–55
- Lateral Movement . . . 610.3–61
- Launch (keyword) . . . 610.3–4
- LEA . . . 610.2–75, 93
- LEAVE . . . 610.2–69, 84-85
- Leetspeak . . . 610.3–83
- Libraries . . . 610.2–5, 19
- License ID . . . 610.3–76
- LIFO . . . 610.2–68
- Linked List . . . 610.5–99
- Linker . . . 610.2–5
- Listener . . . 610.3–61
- Listing . . . 610.2–12
- Loader . . . 610.2–5, 149
- Local Variables . . . 610.2–54, 68, 70
  - Display . . . 610.5–152
  - Example . . . 610.2–82, 89, 93
  - Loading . . . 610.2–70, 155
- Location (header) . . . 610.3–24
- Location.href . . . 610.3–115-116
- LOLBAS . . . 610.3–53
- LOLBIN . . . 610.3–53, 87
- Long Jump . . . 610.2–67
- Loop . . . 610.2–106-119
  - Components . . . 610.2–110
  - Example . . . 610.2–111, 116-119
  - Types . . . 610.2–112
  - Usage . . . 610.2–108
- LOOP(cc) . . . 610.2–109
- Looping . . . 610.2–39
- lpProcName . . . 610.5–51

M

- MAC Address . . . 610.5–63
- Machine Code . . . 610.3–74
- Macros . . . 610.3–38-39
  - Cached Version . . . 610.3–54
  - Debugging . . . 610.3–57
  - Emulation . . . 610.3–56
  - Extraction . . . 610.3–45, 49
  - Infection Scenario . . . 610.3–77
  - Lessons Learned . . . 610.3–64, 610.3–78
  - Major Change . . . 610.3–39
  - Prevention . . . 610.3–39
  - Security Risk . . . 610.3–39
  - Support . . . 610.3–78
- MAEC . . . 610.1–11
- Magic Number . . . 610.3–83
- Malicious DLL . . . 610.4–151
- Malicious Pattern . . . 610.5–21, 39, 49-50, 52, 59, 153
- Malicious Scripts . . . 610.3–103
- Malware 32/64-bits . . . 610.2–147
- Malware Analysis . . . 610.1–6
  - Capabilities . . . 610.1–87-88
  - Experimentation . . . 610.4–104
  - Initial Look → PEStudio (see **Categories**) . . .

  Inputs/Outputs 610.1–10
  Lab 610.1–25-26
    Detection 610.5–62
  Report 610.1–11-12
  Stages 610.1–9
Malware Data Repositories 610.1–13
Malware Hash Registry 610.1–13
Malware Self-Defense Techniques 610.5–41
MalwareBazaar 610.1–13
Marshal 610.3–73
  610.4–75
MBC 610.1–11, 87
MEM_WRITE 610.4–38
Memory Access Violation 610.4–38
Memory Address 610.2–23
Memory Breakpoint 610.5–157-158
  Singleshoot 610.5–160-161
Memory Map 610.4–46, 48
Memory Region Extract 610.4–93-101, 108, 126, 131
  610.5–36, 57
Memory Segment Attributes 610.5–162
Metadata 610.1–9
  610.3–54-55
MetaDefender 610.1–13
Microsoft Compiler 610.2–73
Microsoft Excel 610.3–46
Microsoft Word 610.3–46, 81-82
Misdirection Technique 610.5–101
  Lessons Learned 610.5–114, 142
Missing Applications 610.5–62
MITRE ATT&CK 610.1–87
  610.3–39
MOTW 610.3–39
Mouse Cursor 610.5–65
Mouse Event Blocking 610.5–73
Mouse Hooking 610.5–71
MOV 610.2–76
  Equivalent 610.2–67
  Optimization 610.2–76
  Substraction 610.2–76
  Suspicious 610.5–127-128
MOV DWORD PTR 610.5–101
MS Office Documents 610.3–42
  Formats 610.3–40
  Analysis 610.3–45
MSI 610.3–87
Mutex 610.1–15
  610.2–143
MZ 610.2–12

N

NanoLocker 610.5–141
NAT 610.1–127
Native APIs 610.4–137
Native Code 610.4–107
.NET Framework 610.4–107
  Debugging 610.4–125
  Lessons Learned 610.4–132
  PowerShell 610.4–115
Net.WebClient 610.3–108
NIST SP 800-83 610.1–5
Noise Filtering 610.1–62
NOP 610.2–67
  610.5–9-11
  Sled 610.3–91
NoProfile 610.3–67
NSA 610.2–6
Nt-Prefixed APIs 610.4–137
Ntdll.dll 610.5–149
NTFS 610.3–39
NtGlobalFlag 610.5–13

O

Obfuscated Decompiled Code 610.4–111-112
Obfuscation 610.3–46
  Layers 610.3–117
Object Code 610.2–5
Object Number 610.3–6
Object Stream 610.3–29
ObjStm (keyword) 610.3–29-34
ObsidianGUI 610.5–79
OEP 610.4–22, 36-37
  610.5–105-111, 156
  EIP (32-bit) 610.4–37
  Preliminary Actions 610.5–145
  RIP (64-bit) 610.4–37
OLE Control 610.3–53
OLE1 610.3–81
OLE2 610.3–40, 47, 83-84
OllyDbg 610.5–78
OOXML 610.3–40, 47
  Examination (Without Decompressing) 610.3–55
Open File Handle Scanning 610.3–95
Open Threat Exchange 610.1–13, 16
OpenAction (keyword) 610.3–8
OpenVPN 610.1–19
Operands 610.2–23
  Addressing Modes 610.2–23
Optimization 610.2–69, 76-78, 131
Optional Header 610.4–19-20
Ordinal 610.2–138
Origin Concealment 610.1–18
OriginalFilename 610.4–117
OSINT 610.1–17-18, 22
Outgoing Calls 610.2–141
Overflow Flag 610.2–42
Overlay 610.3–89, 96

P

P-code 610.3–40, 47
P2P 610.2–45
Package Object Server 610.3–81

- Packing 610.1–87
  - 610.2–35
  - 610.4–5
  - Bypass 610.4–21
  - Byte Values 610.4–11
  - Combined With Code Injection 610.5–59
  - Manipulations 610.4–7
  - Options 610.4–8
  - Symptoms 610.4–9
  - Verification 610.4–13
- PAGE_EXECUTE_READ 610.5–158
- PAGE_EXECUTE_READWRITE 610.4–75, 147
  - 610.5–50
- Parameter 610.2–68, 70
  - 64-bit 610.2–157
  - Example 610.2–81, 89
  - Information 610.2–154
- Patching 610.5–9-11, 65, 90, 129-130, 167
- Payload 610.3–81, 86
- PCAP 🗎 610.1–65
- PDF 610.3–4-5
  - Analysis 610.3–35
  - Keywords 610.3–4
  - Lessons Learned 610.3–28, 35
  - Objects 610.3–6
    - Dump 610.3–14, 35
    - Extraction 610.3–14
    - Reference 610.3–15-16, 32-33
    - Unique Identifier 610.3–6
  - Trailer 610.3–5
  - Triage 610.3–8-17
  - Version 1.5 610.3–29
- PE Header 610.4–7, 22, 26, 29
  - 610.5–98
- PEB 610.3–94
  - 610.5–12-13
- Pending Rename 610.1–61
- Persistence 610.1–45, 61, 66, 77
  - 610.2–18, 20
  - Fileless 610.4–104
- Phishing 610.3–8-9, 12
  - Webites 610.1–18
- PIC 610.2–149
- PMF 🗎 610.1–62
- Pointer 610.2–29-30
- POP 610.2–67
  - 610.3–93
- Popup 610.3–16
- Port Unreachable 610.1–67
- Post-Infection 610.3–76-77
- PowerShell 610.3–60-63, 67-73, 108-111
  - Debugging 610.4–76
  - ISE 610.4–76-79
  - .NET 610.4–115
- Pre-OEP Actions 610.5–145
- Predictable Memory Locations 610.4–18
- Process Activities 610.1–62
- Process Attributes 610.5–92
- Process Hollowing 610.5–52-53, 59
  - API Calls 610.5–59
- Process Replacement 610.5–52
- PROCESS_INFORMATION 610.2–94
- ProcessInformationClass 610.5–40
- Program Tree 610.2–12
- Prologue 610.2–67-69, 154
  - Example 610.2–83, 90
- Propagate Ext Params → External Parameters
- Protocol Emulator 610.1–120
- Public Cloud 610.1–19
- PUSH 610.2–22, 29, 76, 157
  - Substraction 610.2–76
- PXE Booting 610.1–34

Q

- QEMU 610.5–64
- Quttera 610.1–13, 20
- Qword 610.2–28, 149

R

- RAR 610.3–98-99
- Raw Alignment 610.5–163
- Raw Size 610.4–10
- RDTSC 610.5–14, 167
- Readable Strings 610.4–10
- Rebasing 610.4–19
- Recipe 610.1–105, 107
  - 610.3–104
- Redirect 610.3–22-24, 103
- Reference 610.2–32, 105
  - Example 610.2–80, 88
- Referenced Strings 610.4–34, 50
- Reflective Code Loading 610.4–114-116
- Register Machine 610.2–24
- Register-Based CPU 610.2–24
- Registers 610.2–23-27
  - 32-bit, 16-bit, 8-bit 610.2–27
  - 64-bit 610.2–150-151
  - Edit 610.5–9
  - General-Purpose 610.2–24-25, 149
  - Segment 610.2–26
    - 610.5–127
  - Special-Use 610.2–26, 149
- Registry Content Extraction 610.4–69
- Registry Key 610.1–44
  - 610.2–20
  - Obfuscated Software 610.4–104
- Registry Redirection 610.2–148
- Regsrv32 ⊞ 610.3–53
- Relocation 610.4–18
- REM 610.1–2, 36
- Remainder 610.3–88-89
- REP 🗎 610.2–133
- Resource Extraction 610.2–142

- Resource Section 610.2–140
- Resource Starvation 610.1–71
- ResumeAllThreads 610.4–85
- RET 610.2–84-85
  - Equivalent 610.2–85
- Return Path 610.1–97
- Return Register 610.2–25
- Reverse Shell 610.3–61
- ReZer0v4 610.4–130
- RFC 1918 610.3–61
- RIP 610.1–94
  - 610.2–149
  - OEP 610.4–37
- RIP-Relative Addressing 610.2–149
- Risky Capabilities 610.1–29, 88
  - 610.3–45
- Rock Debugger 610.5–79
- ROL/ROR 610.1–104
  - 610.5–19
- Root Object 610.3–5
- Rootkits 610.2–147
  - 610.4–135
  - Kernel-Mode 610.4–135
  - User-Mode 610.4–135
- ROT 610.1–104
- RTF 610.3–81-82
  - Lessons Learned 610.3–100
  - Threat 610.3–81
- Rtl-Prefixed APIs 610.4–137
- Run to User Code 610.1–97
  - 610.4–89
- Run Until Selection 610.4–52, 90
- Rundll32.exe 610.2–137
- RunPE 610.5–52
- Runtime Environment 610.4–107

S

- Sandbox 610.1–13
- Satana (ransomware) 610.5–41
- SAZ 🗎 610.3–20
- Scale 610.2–32-33
- SCASB 610.2–23
- Schtasks (command) 610.2–157
- SCISI Disk Identifier 610.5–64
- Scope 610.2–54
- Scrambling 610.4–8
- Scripting 610.4–57
- Sections 610.1–47
  - 610.2–12
  - 610.4–7
- Security Tools Presence 610.5–77-81, 83
- SecurityTrails 610.1–13, 16
- Segment Registers → Registers
- SEH 610.5–98, 128-136
  - 32/64-bit 610.5–98
  - Chain 610.5–99
  - First Record 610.5–99
  - Frame-Based 610.5–99-100
  - Handler 610.5–101, 103, 132-136
  - Structure 610.5–98, 131, 133, 136
  - Table-Based 610.5–98
- Selector 610.5–127
  - FS 610.5–128
- Self 610.2–75
- Self-Defending Capabilities 610.1–31
  - Countermeasures 610.1–32
  - Examples 610.1–47
- Self-Modifying Code 610.2–35
- Self-Referencing Code 610.2–35
- SEO 610.3–27
- Serialization 610.3–82
- SetBPX 610.1–93
  - 610.4–46, 84, 87, 93
  - 610.5–147, 154
  - DLL Name 610.4–87
- Settings.xml 610.3–55
- SFP 610.2–68, 70
- Shellcode 610.3–4, 73-75, 81, 86, 88, 91-97
  - 610.4–75, 80-81
  - Address 610.4–82, 84
  - API Calls 610.3–94
  - Attach 610.4–83
  - Debugging 610.4–86
    - Extraction 610.4–96-101
  - Dynamic Code Analysis 610.4–87
  - Emulation 610.3–96
    - 610.4–80
  - Execution 610.3–97
  - Preliminary 610.3–93-94
  - Presence 610.3–91
- ShellExecute 610.3–108
- Shodan 610.1–13
- Sign Flag 610.2–42
- Signatures 610.3–76, 93-94
  - 610.4–9
- Single Stepping 610.4–113
  - 610.5–34-35
- SMS 610.2–45
- Snapshot 610.1–28, 33-34
  - 610.2–11
  - Software 610.1–34
- Social Engineering 610.3–100
- Software Breakpoint → Breakpoint
- Software Hive 610.2–148
- Source Code 610.2–5
- Source IP 610.1–18
- Special-Use Registers → Registers
- Spyware 610.4–135
- SRE 610.2–6
- SRP Streams 610.3–54
- SS 610.3–40
- SSH Daemon 610.4–70
- SSL Keys 610.3–19
- Stack 610.2–68-70
  - Breakpoint 610.5–106-107, 114

Pointer 610.2–25, 68
Segment 610.2–26
Strings 610.5–20-25
Stack Cleanup 610.5–145
Stack Reuse 610.5–145
Start (command) 610.4–65
STARTUPINFOW 610.2–95
Static Analysis 610.1–81
.... 610.4–26
Findings 610.1–50
Tools 610.1–35
Static Code Analysis 610.1–82
.... 610.2–6
Code Injection 610.4–138
Static Variables 610.2–54
Stdcall 610.2–72, 74
Stopping Condition 610.2–116
Storage 610.3–40
Stream 610.3–6, 40, 47
Bytes 610.3–47
Strings 610.1–44
.... 610.2–34, 103-104
References 610.4–34, 50
.... 610.5–151
Strlen 610.2–78, 111
Structured Storage 610.3–40
Suspended State 610.4–84
Suspicious Keywords 610.3–8
Switch Statement 610.2–130-132
Symbol References 610.2–21, 47
Symbol Table 610.2–21, 46-47
Symbol Tree 610.2–16
Symbolic Constants 610.2–36, 49
Sysinternals 610.2–19
SYSTEM 610.2–156
System Anomalies 610.5–92
System Breakpoint 610.5–120
System Calls 610.2–19
Syswow64 .... 610.2–148

T

Tampared File 610.4–12
Task Manager 610.1–53
Terminated Unicode 610.2–63
TEST 610.2–41, 43, 52
ThemeFontLang 610.3–55
Themida 610.4–8
Thiscall 610.2–73, 75
Threat Intelligence 610.1–13
ThreatFox 610.1–13
TIB 610.3–94
.... 610.5–99
Tight Strings 610.5–25
Timba 610.5–65
Timing Execution 610.5–14
Tiny PE 610.2–5
TLS Callback Functions 610.5–115-125
Static Analysis 610.5–117
TLS Keys 610.3–19
Toolkit Detection 610.5–62
Lessons Learned 610.5–92
TOR 610.1–19
Triage 610.1–50
Trial and Error 610.5–105, 107, 145
TrickBot 610.5–163, 165
Tricky Jump 610.5–118
Trusted Certificate 610.2–147
Trusted Document 610.3–39
Trusted Location 610.3–39
Trusted Publisher 610.3–39
Try-except 610.5–98

U

UAC 610.1–41
Unconditional Jump 610.2–40
Unexpected DLL 610.5–148
Unicode String 610.2–63, 610.4–71
Unique Identifier 610.5–63
Unpacking 610.4–5-6
Breakpoint 610.4–31
Check 610.5–108
Code Extraction 610.5–156
Code Location 610.4–31, 48
Debugging 610.4–29
Dump 610.4–22
Manually 610.4–17
Memory Allocation 610.4–92
Techniques 610.5–145
Upatre 610.5–65
UPX 610.4–8, 12
Unpacking 610.4–14, 42
End 610.4–31
URI (keyword) 610.3–8, 10-11
URL Pattern 610.5–19
UrlScan.io 610.1–13, 20
User Code 610.1–97
User Space Code 610.2–19
User-Agent 610.1–123
.... 610.3–75
User-Interaction Events 610.5–66
UTF-16 .... 610.2–20
.... 610.3–67, 69-70
.... 610.4–71

V

Valak 610.3–56
VBA → Macros ....
Vectored Exception Handling 610.5–98
Verb 610.2–50
Version Number 610.3–6
Video Controller 610.5–63
Virtual Alignment 610.5–163

FOR610 – Reverse-Engineering Malware: Malware Analysis Tools and Techniques

Virtual Network 610.1–30
  Custom 610.1–30
  Host-Only 610.1–30
Virtual Size 610.4–10
VirtualAlloc 610.4–75, 92
  Argument 0x40 .... 610.4–75
  EAX 610.4–95
VirtualBox 610.1–28
VirtualBox Guest Additions 610.5–63
Virtualization 610.1–27
  Artifacts 610.5–62-63
  Detection 610.5–63-64
  Software 610.5–62
VirusTotal 610.1–13-14, 20
VM Escape 610.1–29
Vmcheck.dll 610.5–81
VMWare 610.1–28-30
  OUI 610.5–63
  Tools 610.1–29
    .... 610.5–62-63
VPN 610.1–19
Vulnerability 610.3–86

W

Wcsicmp 610.2–106, 116
Web Debugger 610.3–20
Website Investigation 610.1–13
WH_MOUSE_LL 610.5–69
While Loop 610.2–112, 114
Wide Character 610.2–20
Winbindex 610.1–13
WinDbgFrameClass 610.5–79
Windows Internals 610.4–57
Windows Registry 610.3–53
Windows Vista/Server 2008 .... 610.2–147
WindowStyle 610.3–67
Wine 610.5–64
WireGuard 610.1–19
WM_LBUTTON* 610.5–71
WM_MOUSEMOVE 610.5–70-71
WMI 610.4–60
    .... 610.5–64
Word 610.2–28
Workbook_Open 610.3–46, 59
Worm 610.1–126
WriteAllBytes 610.4–79
Wscript.Echo 610.3–112-113
Wscript.Shell 610.3–108
    .... 610.4–68
Wsprintf 610.2–154-157

X

x86 Architecture 610.2–24-25
XML Source Code 610.3–54
XObject (keyword) 610.3–9, 12-15
XOR 610.1–104
    .... 610.3–72, 74
    .... 610.5–16-19
Xref 610.3–5

Y

Yara 610.1–21, 48
    .... 610.3–76

Z

Zero Flag 610.2–42-43
Zeta Debugger 610.5–79
ZIP 🗎 610.3–40
Zone.Identifier 610.3–39
Zw-Prefixed APIs 610.4–137