

INDEX

A

- About text field, Trainer generator dialog, 9
- accessing memory
 - in injected DLL, 145–146
 - for writing and reading, 122–124
- Action Message Format (AMF), 169
- actor functions, 216
- actuation, 216, 223
- Address column
 - Event Properties dialog, 55
 - OllyDbg disassembler pane, 27
- addresses, memory. *See* memory addresses
- Address Space Layout Randomization (ASLR), 128
 - bypassing in injected DLL, 146–147
 - bypassing in production, 128–130
 - disabling for bot development, 128
 - in Process Explorer, 56, 57
- Adobe AIR hooking, 169
 - decode() function, 172–173, 174–175
 - encode() function, 171–172, 174–175
 - placing hooks, 173–175
 - RTMP, assessing, 169–170
- Adobe AIR.dll*, 173–175
- airlog tool, 170
- alignment
 - in numeric data, 68
 - of variables, in data structures, 70–71
- ambient light, adding, 190–192
- AMF (Action Message Format), 169
- anti-cheat software, 245–246
 - anti-cheat rootkit, defeating, 261–262
 - binary validation, defeating, 259–261
 - bot footprints, managing, 250–256
 - ESEA Anti-Cheat toolkit, 247
 - GameGuard toolkit, 248–249
 - heuristics, defeating, 262–263
 - PunkBuster toolkit, 246–247
 - screenshots, defeating, 258
 - signature-based detection, evading, 256–257
 - VAC toolkit, 247–248
 - Warden toolkit, 249–250
- anti-crowd-control hacks, 218
- anti-debugging techniques, 251, 255–256
- arithmetic instructions, 90–92
- A* search algorithm, 234
 - cost, 233
 - creating node, 234–237
 - creating path list, 239–240
 - score, 234
 - uses for, 240–241
 - writing search function, 237–239
- ASLR. *See* Address Space Layout Randomization (ASLR)
- Asm2Clipboard plug-in, 42
- assembly code
 - copying, 42
 - tracing, 32–33
 - viewing and navigating in OllyDbg, 27–29
- assembly language, 78. *See also* x86 assembly language
- assembly patterns, searching for, 19–21

- AStarNode class, 234–236
- AT&T syntax, 80
- autocombo, 219
- autododge, 219
- autokite bots, 244
- automatic healer, 218, 225–228, 230–232
- autonomous bots, 221–222. *See also*
 - control theory; state machines
 - cavebots, 241–243
 - complex hypothetical state machine, 228–230
 - error correction, 230–232
 - healer state machine, 225–228
 - pathfinding with search algorithms, 232–234
 - warbots, 243–244
- autoreload, 219
- autosnipe bots, 244
- autowall bots, 244

B

- ban waves, 246
- Bigger Than scan type, Cheat Engine, 6
- binary arithmetic instructions, 90
- binary validation, 248, 259–261
- bits, EFLAGS register, 84
- Blue Screen of Death (BSOD), 256
- bots. *See also* autonomous bots; extrasensory perception (ESP) hacks
 - anti-crowd-control hacks, 218
 - anti-debugging techniques, 251, 255–256
 - automatic healer, 218, 225–228, 230–232
 - detecting debuggers, 251–254
 - detecting visual cues, 205–206
 - disabling ASLR for
 - development, 128
 - emulating keyboard, 211–215
 - footprints, managing, 250–256
 - game updates, dealing with, 101–104

- intercepting network traffic, 206–211
 - monitoring memory, 204–205
- obfuscation, 251, 255–256
- sending packets, 215–217
- spell trainers, 219

- branching, 92–94
- breakpoints, 30, 34, 38
- Breakpoints window, OllyDbg, 26
- BSOD (Blue Screen of Death), 256
- BYTE data type, 67
- bytes, machine code, 78

C

- C++, 66
- callee, 94–95
- caller, 94–95
- callHook() function, 154
- call hooking, 153–156. *See also* Adobe AIR hooking
- calling conventions, 95
 - for call hooks, 155
 - __cdecl, 95, 155
 - __fastcall, 95
 - __stdcall, 95
 - __thiscall, 95, 217
 - for trampoline functions, 168
 - for VF table hooks, 156–158
- CALL instruction, 94–95
- call stack
 - overflow, 255–256
 - viewing, 30
 - x86 assembly language, 86–88
- Call stack window, OllyDbg, 26
- capacity of std::vector, 109
- casting spells. *See* spells
- cavebots, 241–243
- __cdecl convention, 95, 155
- Changed Value scan type, Cheat Engine, 7
- characters. *See also* enemies
 - health bars, monitoring with bots, 204–205
 - pausing execution when health drops, 39–42
 - player health, finding with OllyDbg, 99–101

- char data type, 67
- Cheat Engine, 3, 5–6
 - automatically locating string addresses with, 102
 - cheat tables, 7–8
 - correct address, determining, 7
 - first scan, running, 6
 - installing, 4
 - Lua scripting environment, 18–22
 - memory modification, 8–11
 - next scan, running, 7
 - pointer scanning with, 14–18
 - scan types, 6
 - std::list, determining whether data is stored in, 112–113
 - std::map, determining whether data is stored in, 117
 - trainer generator, 9–11
 - VF tables, 78
 - zoom factor, finding, 197
- cheat tables, Cheat Engine, 7–8
- Cheat Utility plug-in, 42–43
- CheckRemoteDebuggerPresent()
 - function, 251
- classes, 74–78
- class instances, 76
- CloseHandle() function, 122, 138
- closing mutexes, 59–60
- CMP instruction, 92
- code caves, 134
 - loading DLLs, 143–146
 - thread hijacking, 138–142
 - thread injection, 134–138
- code injection, 133–134
 - bypassing ASLR in production, 128–130
 - DLLs, 142–146
 - with thread hijacking, 138–142
 - with thread injection, 134–138
- code patches, creating, 31–32
- column configurations, Process Monitor, 51
- combat, automating, 243–244
- command line plug-in, OllyDbg, 43–44
- command syntax, x86 assembly language, 79–81
- Comment column, OllyDbg
 - disassembler pane, 28
- complex hypothetical state machine, 228–230
- conditional breakpoints, 34, 38
- conditional statements, 93
- constant ratio of health, adjusting for, 230–231
- control-critical routines, timing, 254
- control flow hacks, 31
- control flow manipulation, 149–150.
 - See also* Adobe AIR
 - hooking; Direct3D
 - hooking
 - call hooking, 153–156
 - IAT hooking, 160–165
 - jump hooking, 165–169
 - NOPing, 150–152
 - VF table hooking, 156–160
- control theory, 222
 - combining with state machines, 225
 - complex hypothetical state machine, 228–230
 - error correction, 230–232
 - healer state machine, 225–228
- control windows, OllyDbg, 25–26
- cooldowns, displaying enemy, 200–201
- copying assembly code, 42
- copy-on-write protection, 126
- corpses, bot behavior toward, 229, 240
- correct address, determining in Cheat Engine, 7
- CPU window, OllyDbg, 26–30, 40
- crashing debuggers, 255
- CreateRemoteThread() function, 129, 130, 134, 138
- CreateToolhelp32Snapshot() function, 120, 141
- creature data, knowing structure behind, 106–107
- critical game information, displaying, 198–201
- crowd-control attacks, 218
- cryptographic functions,
 - hooking, 170

- CS register, 85
 - C-style operators, OllyDbg, 34–35
 - custom behaviors for cavebots, scripting, 243
- D**
- dark environments, lighting up, 190–192
 - data modification instructions, 89
 - data structures, 71–73
 - data types, 66
 - classes and VF tables, 74–78
 - numeric data, 67–69
 - OllyDbg, 36
 - string data, 69–71
 - unions, 73–74
 - DBG_RIPEXCEPTION handlers, checking for, 253
 - debugging. *See also* OllyDbg
 - anti-debugging techniques, 255–256
 - debug drivers, checking for, 254
 - debug strings, printing, 253
 - detecting debuggers, 251–254
 - Process Monitor, 52–53
 - `__declspec(naked)` convention, 168
 - `decode()` function, hooking, 172–173, 174–175
 - Decreased Value By scan type, Cheat Engine, 7
 - Decreased Value scan type, Cheat Engine, 7
 - dependencies, DLL, 145
 - dependency loading, 160
 - depositor, 242
 - destination operand, 80
 - detection, avoiding. *See* anti-cheat software
 - `device->SetRenderState()` function, 192
 - Dijkstra’s algorithm, 233–234
 - Direct3D 9, 176
 - Direct3D hooking, 175–176. *See also* extrasensory perception (ESP) hacks
 - detecting visual cues in games, 205–206
 - drawing loop, 176–177
 - finding devices, 177–181
 - optional fixes for stability, 184
 - writing hook for `EndScene()`, 182–183
 - writing hook for `Reset()`, 183–184
 - directional lighthacks, 190–191
 - disabling ASLR, 128
 - disassembler pane, OllyDbg, 27–29, 42
 - Disassembly column, OllyDbg disassembler pane, 28
 - `dispatchPacket()` function, 210
 - display base, 27
 - DLL (dynamic link library), injecting, 142–146
 - `DllMain()` entry point, 144–145
 - DLLs option, Process Explorer pane, 57
 - Domain Name System (DNS) cache scans, 248
 - DOS header, 160–161
 - `DrawIndexedPrimitive()` function, 194, 195, 196, 200
 - drawing loop, Direct3D, 176–177
 - DS register, 85
 - dump pane, OllyDbg, 29–30
 - DWORD data type, 67, 145–146
 - dynamically allocated memory, 6, 11, 12
 - dynamic link library (DLL), injecting, 142–146
 - dynamic lure, 242–243
 - dynamic structures, 105
 - `std::list` class, 110–113
 - `std::map` class, 114–118
 - `std::string` class, 105–108
 - `std::vector` class, 108–110
- E**
- EAX register, 81
 - EBP register, 83
 - EBX register, 82
 - ECX register, 82, 157
 - EDI register, 83
 - EDX register, 82
 - EFLAGS register, 84, 92
 - EIP register, 83, 139

- emulating keyboard, 211–215
- enableLightHackDirectional() function, 190–191
- encode() function, hooking, 171–172, 174–175
- EndScene() function
 - jump hooking, 178–181
 - stability of, 184
 - writing hook for, 182–183
- endSceneTrampoline() function, 181
- enemies. *See also* extrasensory perception (ESP) hacks
 - cooldowns, displaying, 200–201
 - critical game information, displaying, 198–201
 - predicting movements of, 241
 - texture, changing, 195–196
- entropy, 5, 7
- Environment tab, Process Explorer
 - Properties dialog, 58
- error correction, 230–232
- ESEA (E-Sports Entertainment Association), 247
- ESEA Anti-Cheat toolkit, 247
- ESI register, 83
- ESP hacks. *See* extrasensory perception (ESP) hacks
- ESP register, 83
- ES register, 85
- Euclidean distance heuristic, 236
- event class filters, Process Monitor, 51–52
- event log, Process Monitor, 52–53
- Event Properties dialog, 54–55
- Exact Value scan type, Cheat Engine, 6
- exception handlers, checking for, 253
- execute protection, 125–128
- Execute until return button, OllyDbg, 25
- experience-tracking HUD, 200
- exponent, float data type, 68
- expressions, OllyDbg, 36–37
 - accessing memory contents with, 36
 - elements evaluated by, 35–36
 - expression engine, 33–36

- pausing execution when health of character drops, 39–42
- pausing execution when name of player is printed, 37–38
- supported data types, 36
- extrasensory perception (ESP) hacks, 189–190
 - background knowledge, 190
 - floor spy hacks, 201–202
 - HUDs, 198–201
 - lighthacks, 190–192
 - loading-screen HUDs, 201
 - pick-phase HUDs, 201
 - range hacks, 201
 - wallhacks, 192–197
 - zoomhacks, 197–198

F

- false positives, VAC toolkit, 248
- __fastcall convention, 95
- feedback loop, 222
- file accesses, inspecting in Process Explorer, 60
- Filesystem event class filter, 52
- FIFO (first-in-last-out), 86
- filters, event class, 51–52
- findItem() function, 116–117
- findSequence() function, 175
- first-in-last-out (FIFO), 86
- first-person shooter (FPS), xxii, 246
- first scan, running in Cheat Engine, 6
- flags, process access, 121
- float data type, 67–68
- floor spy hacks, 201–202
- fog of war, 189. *See also* extrasensory perception (ESP) hacks
- footprints, managing, 250–256
- Found intermodular calls window, OllyDbg, 40
- FPS (first-person shooter), xxii, 246
- FPU registers, 29
- Frame column, Event Properties window, 54
- frames, in Direct3D drawing loop, 176

Freeze interval, Trainer generator dialog, 9

freezing
addresses, 8
main thread, 141

frontier, 233

FS register, 85

function calls, x86 assembly
language, 94–95

function flowchart, OllyFlow, 45

function names, finding for IAT
hooking, 163

G

GameActuators class, 225

game automation state machine,
223–224

GameGuard toolkit, 248–249

game updates, determining new
addresses after, 101–104

general registers, 81–82

generic memory functions, 123–124

getAddressforNOP() function, 152

GetAsyncKeyState() function, 196

GetExitCodeThread() function, 129

GetModuleFileName() function, 144

GetModuleHandle() function, 129–130,
134, 144, 146–147

GetSystemTimeAsFileTime() function, 258

GetThreadContext() function, 139, 142

GetTickCount() function, 254

GetWindowThreadProcessId() function, 120

goal state, 238

Go To button, OllyDbg, 25

greedy best-first search algorithm,
233–234

GS register, 85

guard protection, 126

H

halting problem, 250

handle manipulation options,
Process Explorer, 59–60

handler functions, 208

handles, 56, 121, 210–211, 252

Handles option, Process Explorer
pane, 57

Handles window, OllyDbg, 26

hardware breakpoints, checking for,
252–253

hash validation, 247

heads-up display (HUD), 198–201

healer state machine, 225–228,
230–232

health of characters

health bars, monitoring with
bots, 204–205

health bars of enemies,
displaying, 150–152

pausing execution upon
drop in, 39–42

heap data, 16

heuristics, 233

defeating, 262–263

Euclidean distance, 236

Manhattan distance, 235

Hex dump column, OllyDbg

disassembler pane, 27–28

hidden data, displaying, 198–201

Hidden option, Process Explorer
pane, 57

hooking, 42, 149, 153. *See also* Adobe

AIR hooking; Direct3D

hooking; extrasensory

perception (ESP) hacks

call, 153–156

detecting visual cues in games,
205–206

IAT, 160–165

intercepting network traffic,
206–211

jump, 165–169

prewritten libraries, 169

signature-based detection,
evading, 257

VF table, 156–160

zoomhacks, 198

hotkeys

Patches window, OllyDbg, 32

Process Explorer, 57

Process Monitor, 52

for trainer, setting up, 10

- hourly experience, finding, 200
- HTTP (HyperText Transfer Protocol), 169
- HTTPS (HTTP Secure), 169
- HUD (heads-up display), 198–201

I

- IAT (import address table) hooking, 160–165
- IDIV instruction, 92
- IMAGE_DOS_HEADER structure, 161
- IMAGE_IMPORT_DESCRIPTOR structure, 162
- IMAGE_OPTIONAL_HEADER structure, 161
- Image tab, Process Explorer
 - Properties dialog, 57–58
- IMAGE_THUNK_DATA structure, 162
- immediate value, 80
- import address table (IAT) hooking, 160–165
- import descriptors, 162
- IMUL arithmetic instruction, 90–91
- Increased Value By scan type, Cheat Engine, 7
- Increased Value scan type, Cheat Engine, 7
- index registers, 83
- infinite loops, causing
 - unavoidable, 255
- in-game actions, bots for
 - anti-crowd-control hacks, 218
 - automatic healer, 218, 225–228, 230–232
 - emulating keyboard, 211–215
 - sending packets, 215–217
 - spell trainers, 219
- in-game events, logging, 50–52
- instructions, 79
 - arithmetic, 90–92
 - branching, 92–94
 - data modification, 89
 - function calls, 94–95
 - jump, 92–94
- int data type, 67
- Intel syntax, 80
- interrupt handlers, checking for, 252
- iterator, 120

J

- jumpHookCallback() function, 168
- jump hooking, 165–169, 178–181
- jump instructions, x86 assembly language, 92–94

K

- kernel-mode rootkit, GameGuard toolkit, 249
- keyboard, emulating, 211–215
- KEYEVENTF_KEYUP flag, 212
- kiting, 222, 240–241

L

- libraries, hooking, 169
- lighthacks, 190–192
- list class, 110–111
- ListItem class, 110–111
- little-endian ordering, 67
- loader lock, 144
- loading-screen HUDs, 201
- LoadLibrary() function, 143–144
- Location column, Event Properties window, 54
- logging events, Process Monitor, 50–52
- Log window, OllyDbg, 25
- long data type, 67
- long long data type, 67
- looting, 229, 241–243
- Lua scripting environment, Cheat Engine, 18–22
- lure mode, 242

M

- machine code, 78
- main loop
 - Direct3D drawing loop, 176–177
 - syncing with, 164–165
- mana, avoiding wasted, 219
- Manhattan distance heuristic, 235
- mantissa, float data type, 68

- massively multiplayer online
 - role-playing games (MMORPGs), xxi–xxii, 198, 248
- massive online battle arena (MOBA), xxii, 189, 197, 201, 206
- memcpy() function, 136
- memory, 65–66
 - classes and VF tables, 74–78
 - data structures, 71–73
 - numeric data, 67–69
 - string data, 69–71
 - unions, 73–74
- memory access
 - in injected DLL, 145–146
 - for writing and reading, 122–124
- memory addresses, 4
 - accessing with OllyDbg expressions, 36
 - correct, determining in Cheat Engine, 7
 - freezing, 8
 - new, determining after game updates, 101–104
 - rebasng at runtime, 128–129
 - static, 6
- memory-based lighthacks, 192
- memory dump
 - of class data, 76
 - of code cave, 137
 - of data structures, inspecting, 70–71
 - of numeric data, inspecting, 68–69
 - of string data, inspecting, 70
- memory forensics, 97–98
 - new addresses, determining after game updates, 101–104
 - player health, finding with OllyDbg, 99–101
 - purpose of data, deducing, 98–99
 - std::list class, 110–113
 - std::map class, 114–118
 - std::string class, 105–108
 - std::vector class, 108–110
- memory manipulation, 119
 - accessing memory, 122–124
 - address space layout randomization, 128–130
 - memory protection, 124–128
 - process identifier, obtaining, 120–122
- Memory map window, OllyDbg, 26
- memory modification, 8–11
- memory monitoring with bots, 204–205
- memory offset, 80
- memory on write breakpoint, 208
- memory pointer, 11
- memory protection, 124–128, 151
- memory scanning, 3, 98. *See also* Cheat Engine; pointer scanning
 - basic, 4–5
 - importance of, 4
 - memory modification, 8–11
 - new addresses, determining after game updates, 101–104
 - optimization of code, 22
 - player health, finding with OllyDbg, 99–101
 - purpose of data, deducing, 98–99
- MMORPGs (massively multiplayer online role-playing games), xxi–xxii, 198, 248
- mnemonics, 78
- MOBA (massive online battle arena), xxii, 189, 197, 201, 206
- modifying memory values, 8–11
- Module32First() function, 144, 174
- Module32Next() function, 144, 174
- Module column, Event Properties window, 54
- Modules window, OllyDbg, 25
- monitoring memory with bots, 204–205
- monsters, kiting, 240–241
- mouse movements, emulating, 215, 240
- MOV instruction, 89
- multiclient patching, 30
- mutexes, closing, 59–60

N

- named pipes, locating, 60
- name of specific player, pausing
 - execution when printed, 37–38
- Names window, OllyDbg, 29
- near calls, 153–154
- near function call, 39
- .NET processes, 59
- Network event class filter, 52
- new addresses, determining after
 - game updates, 101–104
- next scan, running in Cheat Engine, 7
- nodes, 233, 234–238
- no-operation (NOP) commands, 31, 32
- NOPing, 150–152
 - lighthacks, 192
 - zoomhacks, 197–198
- NtQueryVirtualMemory() function, 246, 257, 259
- NtWriteVirtualMemory() function, 261–262
- null terminator, 70
- numeric data types, 67–69
- numeric operators, OllyDbg, 34–35

O

- obfuscation, 251, 255–256
- observing game events
 - detecting visual cues, 205–206
 - intercepting network traffic, 206–211
 - monitoring memory, 204–205
- obstacles, searches disrupted by, 233–234
- offset, 54
- OllyDbg, 23–24
 - assembly code, 27–29, 32–33
 - call stack, viewing, 30
 - code patches, creating, 31–32
 - command line for, 43–44
 - control windows, 25–26
 - CPU window, 26–30
 - crashing debuggers, 255

- dealing with game updates, 104
- debugger buttons and
 - functions, 25
- expression engine, 33–37
- memory, viewing and searching, 29–30
- memory dump of numeric data, 68–69
- memory dump of string data, 70
- packet parser, finding, 207–208
- Patches window, 31–32
- patching if() statements, 46–47
- pausing execution when health of character drops, 39–42
- pausing execution when name of player is printed, 37–38
- plug-ins, 42–46
- register contents, viewing and editing, 29
- Run trace window, 32–33
- supported data types, 36
- translating code cave assembly to shellcode, 135–136
- user interface, 24–26
- zoom limitation code, finding, 198

OllyFlow plug-in, 45–46

opcodes, 78

OpenProcess() function, 121–122

OpenThread() function, 142

operands

- binary arithmetic instructions, 90
- IDIV instruction, 92
- MOV instruction, 89
- syntax, 80–81
- unary arithmetic instructions, 90

operations, 79

operators, using in OllyDbg

- expression engine, 34–35

optimizing memory code, 22

ordering, little-endian, 67

order of variables, in data structures, 70–71

OutputDebugString() function, 253

P

- packets
 - intercepting, 206–211
 - sending, 215–217
- packing, 251
- padding, 68
- page protection, 125–126
- pages, 124
- parsing packets, 206–211
- Patches window, OllyDbg, 26, 31–32
- patching, multiclient, 30
- patching if() statements, 46–47
- Path column, Event Properties dialog, 55
- pathfinding with search algorithms, 232–234. *See also* A* search algorithm
- path list, A* search algorithm, 239–240
- Pause button, OllyDbg, 25
- pausing execution, 37–38, 39–42
- pausing threads, 184
- PEB (process environment block)
 - structure, 146
- PeekMessage() function, 184
- PE header, 160–161
- pick-phase HUDs, 201
- PID (process identifier), 120–122
- pipes, locating named, 60
- Play button, OllyDbg, 25
- player health, finding with OllyDbg, 99–101
- player versus player (PvP) combat, 243–244
- plug-ins, OllyDbg, 42–46
- pointer chains, 11–12
- pointer path, 11
- Pointerscanner Scanoptions dialog, Cheat Engine, 14–16
- pointer scanning, 11
 - basics of, 12–14
 - with Cheat Engine, 14–18
 - pointer chains, 11–12
 - rescanning, 17–18
- Pong, 46–47
- Popup trainer on keypress field, Trainer generator dialog, 9
- predicting enemy movements, 241
- prewritten hooking libraries, 169
- printf() call, 72, 73–74, 75
- printing debug strings, 253
- Process32First() function, 120
- Process32Next() function, 120–121
- process access flags, 121
- PROCESS_ALL_ACCESS flag, 121
- Process and thread activity event class filter, 52
- PROCESS_CREATE_THREAD flag, 121
- process environment block (PEB)
 - structure, 146
- Process Explorer, 49–50, 55–56
 - configuring colors, 56
 - handle manipulation options, 59–60
 - hotkeys, 57
 - Properties dialog, 57–59
 - user interface and controls, 56–57
- process handles, obtaining, 121
- process identifier (PID), 120–122
- processInput() function, 215–216
- processKeyboardInput() function, 216
- Process Monitor, 49–50
 - configuring columns in, 51
 - debugging, 53–55
 - event class filters, 51–52
 - high-score file, finding, 55
 - hotkeys, 52
 - inspecting events in event log, 52–53
 - logging in-game events, 50–52
- Process Monitor Filter dialog, 50
- Processname field, Trainer generator dialog, 9
- processNextPacket() function, 210
- processor registers, 81–86
- Process profiling event class filter, 52
- PROCESS_VM_OPERATION flag, 121, 122
- PROCESS_VM_READ flag, 121
- PROCESS_VM_WRITE flag, 121

- Properties dialog, Process Explorer, 57–59
- protection, memory, 124–128, 151
- PunkBuster toolkit, 246–247, 257
- purpose of data, deducing, 98–99
- PvP (player versus player) combat, 243–244

R

- range hacks, 201
- reading from game memory, 119
 - accessing memory, 122–124
 - address space layout
 - randomization, 128–130
 - memory protection, 124–128
 - process identifier, obtaining, 120–122
- ReadProcessMemory() function, 122–124
- read protection, 125–128
- Real Time Messaging Protocol (RTMP)
 - assessing, 169–170
 - decode() function, hooking, 172–173, 174–175
 - encode() function, hooking, 171–172, 174–175
 - intercepting packets, 207
- real-time strategy (RTS), xxii, 197, 201, 206, 243
- rebasement addresses at runtime, 128–129
- reconnaissance, 49–50
 - Process Explorer, 55–60
 - Process Monitor, 50–55
- recv() function, 207–208
- red-black tree, 114–115
- References window, OllyDbg, 26, 28–29, 40, 100
- refiller, 242
- registers, processor, 81–86
- registers pane, OllyDbg, 29
- Registry event class filter, 51
- Rescan pointerlist window, Cheat Engine, 17–18
- responsive hacks, 203
 - anti-crowd-control hacks, 218
 - automatic healer, 218, 225–228, 230–232

- detecting visual cues, 205–206
- emulating keyboard, 211–215
- intercepting network traffic, 206–211
 - monitoring memory, 204–205
 - sending packets, 215–217
 - spell trainers, 219
- rootkits
 - defeating anti-cheat, 261–262
 - GameGuard toolkit, 248–249
- root node, 113–114
- RTMP. *See* Real Time Messaging Protocol
- RTS (real-time strategy), xxii, 197, 201, 206, 243
- runtime flexibility, 229
- Run trace window, OllyDbg, 26, 32–33

S

- SBD. *See* signature-based detection (SBD)
- scan code, 214
- scan types, Cheat Engine, 6
- scan value, 4
- score, 234
- screenshots, 247, 258
- scripting custom behaviors for cavebots, 243
- scripting engine, Cheat Engine, 18–22
- search algorithms, 232–234. *See also* A* search algorithm
- Security tab, Process Explorer Properties dialog, 58
- segment registers, 84–86
- send() function, 216–217
- sending packets, 215–217
- SendInput() function, 211–212, 215
- SendMessage() function, 213–215
- sensors, of a system, 222
- Set/Change hotkey screen, Cheat Engine, 10
- SetLight() member function, 192
- SetProcessIsCritical() function, 256
- shellcode, 134, 135–136, 138–141
- short data type, 67
- sign, float data type, 68

- signature-based detection (SBD)
 - ESEA Anti-Cheat toolkit, 247
 - evading, 256–257
 - PunkBuster toolkit, 246–247
- signatures, 246
- single-instance limitation, 59–60
- skillshots, 232
- Sleep() function, 164–165, 227
- Smaller Than scan type, Cheat Engine, 6
- source operand, 80
- Source window, OllyDbg, 26
- spawning threads, 129
- spells
 - anti-crowd-control hacks, 218
 - complex hypothetical state machine, 228–230
 - spell trainers, 219
- SS register, 85
- stack frame, 87–89
- stack overflow, 255–256
- stack pane, OllyDbg, 30
- stack trace, Process Monitor, 54–55
- state machines, 223–224
 - automated healer, 225–228
 - combining with control theory, 225
 - complex hypothetical, 228–230
 - error correction, 230–232
 - Lua functions, adding, 229–230
 - runtime flexibility, 229
- static addresses, 6
- __stdcall convention, 95
- std::list class, 110–113
- std::map class, 114–118
- std::string class, 105–108
- std::vector class, 108–110
- Step into button, OllyDbg, 25
- Step over button, OllyDbg, 25
- stochastic systems, 230
- string data, 21, 69–71, 100–101
- string operators, OllyDbg, 35
- Strings tab, Process Explorer
 - Properties dialog, 58
- struct member alignment, 71
- structures, data, 71–73

- subregisters, 83
- SuspendThread() function, 142, 184
- syncing with game threads, 164–165
- systems, controlling behavior of, 222

T

- targets, selecting, 240
- TCP/IP tab, Process Explorer
 - Properties dialog, 58
- TEB (thread environment block), 146
- templates
 - for changing memory protection, 127
 - memory access functions, 123–124, 145–146
- TEST instruction, 92
- text strings, 21, 69–71, 100–101
- texture of enemies, changing, 195–196
- __thiscall convention, 95, 156–158, 217
- Thread32First() function, 141
- Thread32Next() function, 141
- thread environment block (TEB), 146
- threads
 - hijacking, 138–142
 - injection, 134–138
 - spawning, 129
- Threads tab, Process Explorer
 - Properties dialog, 58
- Threads window, OllyDbg, 26
- thunks, 162–163
- timing control-critical routines, 254
- Title field, Trainer generator
 - dialog, 9
- toggling z-buffering, 195
- Trace into button, OllyDbg, 25
- Trace over button, OllyDbg, 25
- tracing with OllyDbg, 32–33, 39–42
- trainer generator, Cheat Engine, 9–11
- trampoline functions, 165–168, 181
- traversals
 - IAT hooking, 162
 - VF tables, 156

U

- unary arithmetic instructions, 90
- unavoidable infinite loops,
 - causing, 255
- Unchanged Value scan type, Cheat Engine, 7
- unions, 73–74
- Unix syntax, 80
- Unknown Initial Value scan type,
 - Cheat Engine, 6
- updates, determining new addresses
 - after, 101–104
- user interface, Process Explorer, 56–57
- user-mode rootkit, GameGuard toolkit, 248–249

V

- VAC toolkit, 247–248
- Value Between scan type, Cheat Engine, 6
- Value Type directive, Cheat Engine, 6
- VF (virtual function) tables
 - class instances and, 76–78
 - finding Direct3D devices, 177–181
 - hooking, 156–160, 182–183
 - traversals, 156
- VirtualAllocEx() function, 136–137, 138
- virtual functions, classes with, 75–76
- VirtualProtectEx() function, 126–128
- VirtualProtect() function, 127

W

- WaitForSingleObject() function, 129, 138
- wallhacks, 192
 - creating for Direct3D, 194–197
 - rendering with z-buffering, 193–194
- warbots, 243–244
- Warden toolkit, 249–250
- waypoints, 222, 229
- wchar_t data type, 67
- window handle, fetching, 120

- Windows window, OllyDbg, 26
- WM_CHAR messages, 213–214
- WORD data type, 67
- WriteProcessMemory() function,
 - 122–124, 136–137, 138
- write protection, 125–128
- writing to game memory, 119
 - accessing memory, 122–124
 - address space layout
 - randomization, 128–130
 - code caves, 136–137
 - memory protection, 124–128
 - process identifier, obtaining, 120–122

X

- x86 assembly language, 78–79
 - arithmetic instructions, 90–92
 - branching instructions, 92–94
 - call stack, 86–88
 - command syntax, 79–81
 - data modification
 - instructions, 89
 - function calls, 94–95
 - jump instructions, 92–94
 - NOPing, 150–152
 - processor registers, 81–86
- x86 Windows memory protection
 - attributes, 125–126

Z

- z-buffering, 192–195
- zoom factor, 197
- zoomhacks, 197–198