



# Index

- %00u format token, 322–324
- 0day exploits
  - IDSs for, 240
  - undisclosed, 10
- 93C46 chips, 422
- 430TX chips, 417
- 8048 chips, 411
- 82439TX chips, 417
- 82559 chips, 422–423
- Access
  - to executable files, 162–163
  - low-level disk, 429–430
  - root, 152
- Access Control Lists (ACLs), 183
- Access requirement audits, 90
- Activating payloads, 57, 281
- Activation zones in injection vectors, 56–57
- Active armor, 356
- ActiveX
  - preloader, local filenames with, 225
  - and Web browsers, 224–225
- add\_long\_cmt function, 100
- Adding users, 180
- Address-based arithmetic, 303–304
- Address resolution protocol (ARP)
  - cache poisoning attacks in, 25
  - packet leaking in, 89–90
- Addresses
  - effective, 98
  - in injection vectors, 283–284
  - IP, 67, 274
- ADM (Association De Malfaiteurs), 20
- ADM worm, 20
- Administrator access, need for, 152
- Adoption rates of technologies, 4
- Aggregation elements, 57
- AIX/PowerPC payloads, 356–360
- Akkerman, Wichert, 226
- Alchemy Eye Network attacks, 188–189
- Allocated memory in heap overflows, 324–328
- Alternate encoding. *See* Equivalent input and requests
- Alternative IP addresses, 274
- AN/SP5-73 radar systems, 287
- Analog in-band switching signals, 205
- Analytical Engine, 2
- AND operator, 340
- Anomaly-based IDSs, 238–239
- Anti-aircraft radar systems, 409
- Apache HTTPD, overflow in, 295
- API calls
  - buffer overflow in, 297, 299
  - mapping, 264–266
  - for reverse engineering, 90–91
  - tracing, 243
- API monitors for injection points, 154–155
- APISPY tool, 91, 243
- Application security vs. software security, 44
- Application Service Provider (ASP) model of software licensing, 30
- Applications vs. operating systems, 24
- Applying attack patterns, 65–69
- Architectural flow, 39, 41–42
- Arguments
  - expansion of, buffer overflow from, 298
  - in shell command injection, 169–171
- Arithmetic
  - address-based, 303–304
  - buffer overflow from, 309–317
- ARP (address resolution protocol)
  - cache poisoning attacks in, 25
  - packet leaking in, 89–90
- ARPANET, 4
- asp dot bug, 264
- ASP pages, embedding Perl within, 164
- Association De Malfaiteurs (ADM), 20
- Assumptions
  - in attack patterns, 48–49
  - undermining, 71
- at utility, 181
- ATMEL 93C46 chips, 422
- Attaching to running processes, 158–159

- Attack examples
  - adding a user with injection, 180
  - address-based arithmetic problem, 303
  - alchemy eye network management software file system, 188
  - alternate encoding triple dot in SpoonFTP, 268
  - alternate encoding with ghost characters in FTP and Web servers, 267
  - Apache HTTPD cookie buffer overflow, 295
  - Baltimore Technologies MailSweeper, 229
  - breaking Oracle 9i with a socket attack, 182
  - buffer qverflow in \$HOME, 297
  - buffer overflow in Internet Explorer 4.0 via EMBED tag, 231
  - buffer overflow in TERM, 297
  - buffer overflow on a Cisco router running on a Motorola CPU, 289
  - building binary files using debug.exe with injection, 175
  - building text files with injection, 175
  - C5 clear forward and seize in-band attack, 205
  - Cold Fusion CFEEXECUTE argument injection, 169
  - combined encodings in CesarFTP, 275
  - defined, 55
  - dotless IP addresses in Internet Explorer274
  - EasyNews PHP script XSS, 215
  - embedded Perl script that calls system() to execute netcat, 164
  - embedded Perl scripts within ASP, 164
  - embedded script in nonscript element from GNU mailman XSS, 215
  - entrust and injection, 187
  - escaped slashes in alternate encodings, 270
  - Excel host () function, 218
  - executable fonts, 165
  - file traversal, query string, and GroupWise, 188
  - file traversal, query string, and Hsphere, 188
  - filter failure in Taylor UUCP daemon, 296
  - FTP glob(), 298
  - Horde IMP, 228
  - Hotmail Java tag filtering, 229
  - HPUX passwd, 298
  - HTTP headers in webalizer XSS, 216
  - Informix database file system, 189
  - injection and FTP, 178
  - injection and remote xterms, 179
  - injection and Tiny FTP (TFTP), 180
  - the Internet Explorer GetObject() call, 225
  - IPSwitch Imail, blind trusted mailbox name, 193
  - ixsso.query ActiveX object, 226
  - Javascript alert dialog XSS, 212
  - keyboard buffer injection, 210
  - Libc in FreeBSD, 297
  - local filenames and the ActiveX preloader, 225
  - meta-characters and the FML mailing list archive, 226
  - microsoft Outlook view control, 228
  - misclassification in NTFS file streams specifier, 264
  - mples: Internet Explorer 5, 230
  - the Outlook application object, 227
  - Outlook XP and HTML on reply or forward, 227
  - overflow binary resource file in Netscape, 294
  - overflow variables and tags in Exim, 294
  - overflow variables and tags in MidiPlug, 294
  - overflow with symbolic links in EFTP server, 294
  - PHP command injection using delimiters, 173
  - PHP global variables, 190
  - PostNuke content management system XSS, 216
  - scheduling a process with injection, 180
  - Scripting.FileSystemObject, 218
  - Scriptlet.TypeLib, 218
  - Sendmail overflow, 295
  - simple script injection, 215
  - slashes in alternate encodings, 269
  - Solaris getopt, 298
  - Syslog(), 324
  - Unicode encodings in the IIS server, 272
  - UNIX environment variable, 189
  - URL encodings in IceCast MP3 server, 273
  - URL encodings in Titan application firewall, 273
  - Wscript.Network, 218
  - Wscript.Shell, 218
  - XSS in MP3 files and spreadsheets, 217
  - Xtlib, 297
- Attack patterns, 37–38
  - alternative IP addresses, 274
  - alternate encoding of leading ghost characters, 267
  - argument injection, 169
  - API calls for buffer overflow, 297
  - applying, 65–69
  - attacker viewpoint, 48–49
  - binary resource files, 293
  - blueprints for disaster, 55–57
  - boxes, 70
  - C++ compiler example, 57–65
  - choosing, 68
  - client invisibility, 150
  - client-side injection and buffer overflow, 231

- command delimiters, 172
- configuration files
  - to run command to elevated privilege, 153
  - for search paths, 156
- content-based file system function injection, 229
- defined, 56
- direct access to executable files, 162
- embedding scripts
  - in nonscript elements, 215
  - within scripts, 164
- environment variables
  - for buffer overflow, 297
  - manipulating, 189
- escaped slashes in alternate encoding, 270
- executable code in
  - nonexecutable files, 165
- filter failure through buffer overflow, 296
- global variables, 190
- HTTP cookies, 295
- HTTP query strings, 216
- in-band switching signals, 205
- local command-line utilities
  - for buffer overflow, 297
- local filenames passed to functions expecting URLs, 225
- lock picks in, 50–51
- meta-characters in e-mail headers, 226
- MIME conversion, 295
- multiple parsers and double escapes, 173
- open-systems view, 42–47
- parameter expansion, 298
- postfix NULL terminators, 186
- programs writing to privileged OS resources, 152
- relative path traversal, 187
- session IDs, resource IDs, and blind trusts, 192
- simple script injection, 214
- slashes in alternate encoding, 268, 274
- string format overflow in syslog, 324
- symbolic link overflow
  - taxonomy of, 38–41
  - trust in, 49–50
- unicode encoding, 271
- URL encoding, 273–274
- user-controlled filenames, 217
- user-supplied variables
  - passed to file system calls, 185
- UTF-8 encoding, 273
- variable and tag overflow, 294
- web logs, 275
- web server misclassification, 263
- XSS in HTTP headers, 216
- Attack signatures, 173, 238–239
- Attacker
  - defined, 55–56
  - in attack patterns, 55–56
  - hiding identity of, 68–69
  - intention, 40
  - viewpoint, 48
- Audit logs
  - poisoning, 275–276
  - truncation of, 296
- Auditing
  - automatic and bulk, 111–121
  - for directly executable files, 162
  - humans best at, 244
- Authentication
  - multiple paths of, 198
  - session, 192
- Automated attacks, 21
- Automatic auditing, 111–121
- AUTORUN.INF file, 429–430
- Babbage, Charles, 2
- Backdoors
  - in attack patterns, 69
  - in flash ROM, 409
  - history of, 368
  - in outsourced software, 9
  - prevalence of, 5
  - on TFTP, 180
  - from worms, 20
  - on X Windows, 179
- Backslashes (\)
  - in alternate encoding, 268–270
  - for Null terminators, 186
  - in shell command injection, 174–175
- Backtracing, 245–247
- Backwash attacks, 230
- Bad software, 11–14
  - Bad Software* (Kananar and Pels), 77
- Baggage handling systems, 11
- Banks, attacks on, 22
- Batch analysis with IDA-Pro, 114–121
- be instruction, 351
- BGP (Border Gateway Protocol) handlers, 281
- Big endian byte ordering
  - vs. little endian, 284–285
  - in MIPS processors, 338
- Binary files
  - overflow in, 293–294
  - patching, 394–395
    - NT kernel, 397–408
    - peephole patches, 395–397
  - shell command injection for, 175–177
- Binary flags for state, 280
- BIND, buffer overflows in, 240
- Biological defense models, 32
- BIOS memory, 409
- Black box analysis
  - limitations of, 234
  - for reverse engineering, 80–81
  - vs. white box analysis, 82
- Black lists
  - for input, 149
  - vs. white lists, 149, 236
- BlackHat security conference, 444
- Blaster worm, 14, 17
- Blind trust, 192–193
- bltjal instruction, 339
- Blue boxes, 21, 204, 206
- Blueprints for disaster, 55–57
- bnel instruction, 356
- Boot disk attacks, 147
- Border Gateway Protocol (BGP) handlers, 281
- Boron tags
  - in buffer overflow, 300
  - checking for, 144–145
  - in code tracing, 253
  - for security, 438

- Branch delay, 337
- Branches
  - in AIX/PowerPC, 356, 359
  - on PA-RISC, 349–350
- Bray, Brandon, 59
- Breakpoints
  - in code tracing, 250–251
  - for input path tracing, 156–157
  - leapfrogging, 251
  - memory page, 253
  - in reverse engineering, 123–126
  - in runtime tracing, 247–248
- Browsing directories, 186–187
- Brute forcing session IDs, 193
- BSD distribution, arithmetic problems in, 303–304
- Buffer overflow, 20, 277–279
  - from arithmetic errors, 309–317
    - audit truncation and filters with, 296
  - in BIND, 240
  - in C++, 57–65, 329
  - in client software, 230–231, 290
  - content-based, 293–295
  - in databases, 289–290
  - in domain name servers, 20
  - in embedded systems, 286–289
  - from environment variables, 296–298
  - from format strings, 317–324
  - gets for, 40
  - heap overflows, 324–328
  - in helpctr.exe, 105–111
  - injection vectors in, 280–286
  - in Java, 291–293
  - kernel, 445
  - in low-level programming languages, 16
  - from multiple operations, 298
  - from parameter expansion, 298
  - payloads in, 329–336
  - potential sources, 298–300
  - Prolog/Epilog code for, 360–366
  - security checks for, 59
  - stack overflow, 278–279, 300–308
    - state corruption in, 279–280
    - trampolining with. See Trampoline attack
    - two-stage, 40–41
- Buffers
  - keyboard, character injection in, 210
  - shared, leaking data in, 88–90
  - tracing, 251–252
- Bug instantiations, 17
- Bugs
  - defined, 39
  - in open-systems view, 45
  - reports and fixes, as vulnerability sources, 86, 106
  - per thousand lines of code, 14–18
- Bugtraq mailing list, 38
- Building Secure Software* (Viega and McGraw), 1, 41, 44, 90, 151–152
- bulk\_audit\_sprintf.idc script, 115–119
- Bulk auditing, 111–121
- Burning out hardware, 425
- Business software, 3
- bv instruction, 351
- Byte code disassemblers, 105
- Byte operations
  - with pointers, 256
  - in reversing parser code, 255–256
- C and C++ language
  - buffer overflow in, 329
  - escape codes in, 208
  - exploit example, 57–65
  - for Java, 291–292
  - string handling routines in, 279
  - vulnerabilities in, 16
  - vtables, 327
- C5 (CCITT-5) signaling system, 205–206
- Call hooking, 380
  - for backtracing, 245–247
  - for hiding processes, 380
  - IDTs, 441
  - for red pointing, 243
  - for removing process records, 383–386
  - structure of, 381–382
  - system calls, 381
- .CALL instruction, 346
- Call stacks for dead ends and runouts, 247
- Call throughs, registers for, 285
- can\_read function, 126–127
- Canary values, 361–364
  - defeating, 361
  - for buffer overflow attacks, 60, 63
- Car engine control codes, 28
- Carriage returns in shell command injection, 173
- Cascade interrupt, 439
- Categories of subversive code, 7
- CCITT-5 (C5) signaling system, 205–206
- CD-ROM images, 429–430
- Cellular phones, 22
- CesarFTP server, 275
- CFEXECUTE tag, 169–171
- CFI (Common Flash Interface)
  - for chip detection, 426–427
- cgi programs
  - misclassification with, 263
  - with Web servers, 163
- Chaos theory, 233
- Character conversions
  - in equivalent requests, 271–274
  - in reversing parser code, 255
- Character injection, 209–211
- Character sets, hostile, 245
- Chat clients for backtrace code, 246–247
- check\_boron function, 144–145
- check\_password function, 62
- check\_target\_for\_string function, 96–98
- Checked build environment, 369
- Checksums for payloads, 335–336
- Chipping cars, 28
- Chips, detecting, 426–429
- CIH virus, 409, 417–421

- Cisco routers, buffer overflow in, 281, 288
- Classification
  - in attacks, 263–264
  - of subversive code, 7
- Client software, 201–202
  - assumptions in, 48–49
  - buffer overflows on, 230–231, 290
  - content-based attacks on, 229–230
  - cross-site scripting, 212–217
  - database, 290
  - honeypots in, 203–204, 230
  - in-band signals for, 204–211
  - invisibility of, 150
  - scripts and malicious code with, 217–229
  - server control of, 202–203
- Code address targets for injection vectors, 282–283
- Code coverage
  - for reverse engineering, 87–88
  - in runtime tracing, 248–249
  - tools for, 139–145
- Code paths in FTP servers, 50–51, 73–74
- Code Red worm, 20–21
- Code-signing errors in Java, 291
- Code tracing, 73, 244–245
  - API calls, 243
  - backtracing, 245–247
  - boron tagging in, 253
  - buffers, 251–252
  - dead ends and runouts in, 247
  - leapfrogging in, 251–253
  - memory page breakpoints in, 253
  - program execution flow, 130–131
  - runtime, 247–250
  - in server software, 156–161
  - speedbreaks in, 250–251
- CodeBase property, 166
- Cold Fusion, CFEXECUTE injection in, 169–171
- Combined attacks with equivalent requests, 274–275
- Command and control activities, 7
- Command-line parameters
  - for executable files, 162
  - for setuid, 153
- Commands and command lines
  - buffer overflow from, 290, 297–298
  - delimiters in, 172–175
  - injecting. *See* Shell command injection
  - in input, 234
  - in JVMs, 291
  - separators in, 268
- Commercial systems, embedded systems in, 287
- Common Flash Interface (CFI)
  - for chip detection, 426–427
- Common Runtime Language (CRL), 79
- Communications systems
  - embedded systems in, 287
  - in software, 237
- COMP.RISKS mailing list, 21
- Compiler flaw, 57–65
- Complex computational systems, 235
- Complexity
  - of computer models, 234
  - of software, 14–17
- Component-based software
  - future of, 24
  - logically distributed systems for, 26
- Components in attack patterns, 67–68
- Computation, future of, 31–32
- Computer-Related Risks* (Neumann), 21
- Computer science theory, 233–234
- Computer Security Institute (CSI) survey, 23
- Concept virus, 218
- Conditional branches, code coverage tools for, 139–140
- Configuration files
  - for elevated privilege, 153
  - search paths in, 156
  - server software trust in, 161–166
- Connectivity of software, 21–22
- Content-based attacks
  - buffer overflow, 293–295
  - on client software, 229–230
- Contexts for threads, 127–128
- continue command, 158
- Contract software
  - backdoors in, 9
  - future of, 29–30
- Control codes
  - for client software, 202–203
  - for terminals, 207–208
- Control flow, 74, 316
- Controller chips, keyboard, 444
- Conversions
  - ASCII chart, 449–452
  - character, 255, 271–274
  - MIME, 295
- Cookies, 52
  - overflow in, 295–296
  - for session IDs, 192
- Coprocessor interrupt, 440
- Copy protection schemes
  - decompiling, 105
  - and reverse engineering, 75
- Copyright law, 76
- Copyright mechanisms, patching, 132
- Corrupting
  - log files, 266
  - states, 279–280
- Covert communication, 7
- Cowan, Crispin, 58, 360–361
- CPU registers
  - for boron tags, 144–145
  - examining, 259–260
  - in injection vectors, 285–286
  - in MIPS, 338
  - in SPARC, 341–342
- Cracking tools, 121
- Crafted input, 233–235
  - audit poisoning with, 275–276
  - code tracing for, 244–253
  - defending against, 235–236
  - equivalent requests in. *See* Equivalent input and requests
  - filters for, 236–237
  - IDSs for, 237–242

- Crafted input (*cont.*)
  - misclassification in, 263–264
  - partition analysis in, 242–244
  - reversing parser code, 254–263
- CreateFile function, 429
- Cross-site scripting. *See* XSS (cross-site scripting)
- CRT\_INIT function, 61
- Cryptography in geographically distributed systems, 29
- Cryptotrojan attacks, 443
- CSI (Computer Security Institute) survey, 23
- CVE vulnerabilities catalog, 38
- CWD (Current Working Directory)
  - redirection with, 187
  - for servers, 163
- CyberCop tool, 66
  
- Damage potential in open-systems view, 45–46
- Data blocks in injection vectors, 286
- Data bombs, 151
- Data chains, character conversion in, 271
- Data collection by subversive code, 7
- Data encryption algorithms, publishing, 76
- Data files, buffer overflow in, 293–295
- Data leaking in shared buffers, 88–90
- Data sections for payloads, 334–335
- Database buffer overflows, 289–290
- Dead ends in code tracing, 247
- Dead listings for input path tracing, 157
- debug.exe program, 175–177
- Debug logs for helpctr.exe, 107
- Debugging and debuggers
  - for binary file building, 175–177
  - multithreading programs, 127–130
  - for reverse engineering, 77–78, 121–123
  - rootkits as, 445
  - tools for, 50
- decode function, 61–63
- Decompiler, 79
- Decompiling, 104–105
  - helpctr.exe, 105–111
  - in reverse engineering, 79
- Deferred procedure calls (DPCs), 416
- Delay slot, 337
- Delayed coordinate embedding, 193
- Delimiters, 172–175, 268
- Denial-of-service problems, 81
- Denver International Airport baggage handling system, 11
- Design-level vulnerabilities, 41–42
- Destination buffers in helpctr.exe, 108–110
- Desynchronization of packets, 240–242
- Detecting
  - chips, 426–429
  - code problems, 324
  - rootkits, 446
- Developing Windows NT Device Drivers* (Dekker and Newcomer), 372
- Device drivers. *See* Drivers
- DialogProc function, 101
- Digital Millennium Copyright Act (DMCA), 27, 76
- Digital rights management, limitations of, 9
- Digital tradecraft, 6–8
- Dir command, 112
- Direct access to executable files, 162–163
- Directories
  - browsing, 186–187
  - hiding, 392–394
  - permissions on, 184
  - redirecting, 54
- Disabling Windows system file protection, 445
- Disasm function, 139
- Disassembly, 50, 53, 138–139
  - for buffer overflow, 299–300
  - in reverse engineering, 78, 104–105
- Discriminator digits in phone systems, 206
- Disk access, rootkits for, 429–430
- Disk controller interrupt, 440
- Distributed systems, 26, 29
- Diversions by IDSs, 239
- DLL files
  - buffer overflows from, 293
  - and fonts, 165–166
- DMCA (Digital Millennium Copyright Act), 27, 76
- Domain name servers (DNS)
  - in attack patterns, 67
  - buffer overflows in, 20
  - in shell command injection, 168
- Dotless IP addresses, 274
- Double escapes in shell command injection, 173
- DPCs (deferred procedure calls), 416
- Dr. Watson log file, 105
- Dr. Watson utility, 138–139
- DrainOutputBuffer function, 413–414
- DrawGLScene function, 198
- Drip-scans, 66
- DriverEntry function, 371, 385, 416–417, 430–432
- Drivers
  - filter, 443–444
  - for kernel rootkits, 370–377
  - network support for, 430–439
  - programs using, 372–373
  - registering, 375–377
  - in reverse engineering, 88
  - structure of, 371–372
  - for Trojan executable redirection, 387–392
  - unloadable, 373–374
- Dumb terminals, 202
- dumpbin utility, 112
- Dumping memory, 259–260
- Dynamic execution for red pointing, 243
- Dynamic jump tables for payloads, 333–334
- dyninstAPI tool, 87

- E-mail attachments, 18
- E-mail injection, 226–229
- ea\_t types, 98
- EasyNews scripts, 216
- echo command, 175
- Economic threats, 22
- EEPROM chips, 409–410
  - burning out hardware, 425
  - enabling read and write from, 417
  - in Ethernet cards, 421–424
  - manufacturers of, 425
  - serial vs. parallel, 424–425
  - timing in, 421
- Effective addresses, structure for, 98
- EFTP server, overflow in, 294
- Electronic warfare, 6
- Elevated privilege problem, 151–154
- elitewrap program, 168
- EMBED tags, 231
- Embedded systems
  - buffer overflows in, 286–289
  - in cellular phones, 22
  - future of, 25, 29
- Embedded scripts, 164–165
- Emergent computation, 32
- Encapsulation
  - future of, 24–25
  - of OSs, 29
- Encryption algorithms, publishing, 76
- .END instruction, 346
- End-user license agreements (EULAs), 77
- Engine control code, 28
- .ENTER instruction, 346
- Enumerating threads and processes, 129–130
- EnumSubKeys function, 434–437
- Environment variables
  - buffer overflow in, 296–298
  - in server software, 189–192
- Environmental effects, 42–44
- Equivalent input and requests, 264
  - API layer mapping for, 264–266
  - character conversion in, 271–274
  - combined attacks in, 274–275
  - ghost characters in, 266–268
  - on IDss, 240–242
  - meta-characters in, 268–271
- Error code checking in server software, 199
- Error handling and recovery systems, 41, 58
- Escape codes
  - in alternate encoding, 270
  - in API calls, 266
  - with in-band signaling, 207–210
  - in log files, 275
  - meta-characters, 269
  - in shell command injection, 173–175
- Espionage, 6–8
- Ethernet cards, EEPROM in, 421–424
- Ethernet scrubbing problem, 89–90
- EULAs (end-user license agreements), 77
- Excel, host function in, 218
- Exception handling
  - for buffer overflow, 299
  - overwriting frames for, 308
- exec function, 168
- Executable code and files
  - direct access to, 162–163
  - in nonexecutable files, 165–166
  - single stepping for, 130–131
  - vs. source code, 10
  - in WINNT, 112–113
  - on workstations, 16
- execv function, 340
- Exim, overflow in, 294
- Existing code in injection vectors, 286
- Exploit, defined, 56
- Exploits in attack patterns, 55–56
- Exposure in open-systems, 46–47
- Expressions
  - for input path tracing, 156–157
  - in shell command injection, 173
- Extensibility of software, 18–21
- External branch instructions on PA-RISC, 350
- External input in software, 233–235
- F00F bug, 409
- Failure recovery systems, 41
- Failure Simulation Tool (FST), 78, 155
- False positives in white box analysis, 80
- Fault injection, 78, 132–133
- Fault-tolerant systems, 132
- Faults, leveraging, 68
- Feedback events, 57
- Felten, Ed, 40, 76, 230
- Fennis tool, 248, 251–252
- File handles for drivers, 372
- File streams specifier, misclassification in, 264
- File systems, 184–185
  - Alchemy Eye Network, 188–189
  - directory browsing, 186–187
  - filenames in, 185
  - Informix Database, 189
  - injection attacks on, 187–188
  - traversal in, 187–188
  - user-supplied variables passed to, 185
- File Transfer Protocol (FTP) servers
  - attacks on, 178
  - buffer overflow in, 298
  - code paths in, 50–51, 73–74
  - ghost characters with, 267
- filemon tool, 154
- Filenames, 185
  - URLs replaced by, 225–226
  - XSS in, 217
- Files
  - controllable, 153
  - hiding, 392–394
- FileSystemObject, attacks on, 218
- Filters
  - with buffer overflow, 296
  - for commands, 264
  - for input, 236–237
  - drivers for, 443–444



- Filters (*cont.*)
  - in IDss, 239
  - in parsing, 254
  - for server software input, 149
  - in shell command injection, 173
- Financial threats, 22
- Firewalls
  - limitations of, 33
  - for port scans, 66
  - as reactive technology, 236
- Firewalls and Internet Security* (Cheswick, Bellovin, and Rubin), 65
- Firmware exploitation, 10
- First parallel port interrupt, 440
- First serial port interrupt, 440
- Fixed-size buffers in stack
  - overflow, 301–302
- Flash RAM, detecting, 427
- Flash ROM, 409–410
- Flaws
  - defined, 39
  - in open-systems view, 45
- Floating point unit interrupt, 440
- flog function, 324
- Floppy disk controller
  - interrupt, 440
- Fluttering windows, 168
- FML mailing list archive, 226–227
- FnDebugDispatch function, 364–365
- Fonts, executable, 165–166
- fOpenThread function, 128
- Forking processes, 183
- Format string vulnerabilities, 317–324
- Formatting poison pills, 293
- Forms, trust assumptions in, 49
- Forwards, injection with, 227
- FoundScan tool, 66
- Fragmentation of packets, 241–242
- fread function, 243
- Free build environments, 369
- free function, 328
- FreeBSD distribution
  - address-based arithmetic in, 303–304
  - buffer overflow in, 297
- freedom to tinker site, 76
- FS register, 299
- FST (Failure Simulation Tool), 78, 155
- FTP (File Transfer Protocol)
  - servers
    - attacks on, 178
    - buffer overflow in, 298
    - code paths in, 50–51, 73–74
    - ghost characters with, 267
    - speedbreaks for, 250–251
- Function call nesting, 344
- Function return addresses in
  - buffer overflow attacks, 60
- Future of software, 23
  - long-term, 30–32
  - medium-term, 28–30
  - short-term, 24–28
  - threads in, 32
- fwrite function, 243
- Gates, 441
- GDB tool, 156–161
- General problems, 38
- General registers in SPARC, 341
- Generic rules in injection
  - vectors, 56
- Geographically distributed
  - systems, 29
- Geopolitics in indirection, 69
- get\_func\_qty function, 98
- GET requests
  - in PHP, 191
  - segmented, 241–242
- get\_user\_defined\_prefix
  - function, 93–94
- getFilenameDialog function, 100–101
- getn\_func function, 98
- GetObject function, 225–226
- getopt function, 298
- GetProcAddress function, 334
- gets function, 40
- Ghost characters, 266–268
- glob function, 298
- Global offset table (GOT)
  - pointers, 60
- Global variables
  - in buffer overflow attacks, 64
  - in PHP, 190–192
- GlobalAlloc function, 333
- GNU Mailman, embedded
  - scripts in, 215
- GOT (global offset table)
  - pointers, 60
- Graphing
  - phase space analysis, 193–198
  - for reverse engineering
    - software, 50–52
- Gray box analysis
  - for input path tracing, 157
  - on Microsoft SQL Server 7, 82–84
  - for reverse engineering, 81–84
- grep tool, 160
- GroupWise, file traversal in, 188
- /GS compiler option, 59–63
- Hackers
  - defined, 34
- Hacking Exposed* (McClure, Scambray, and Kurtz), 1, 8, 148, 167, 199
- Hailstorm tool, 78, 81, 83–84
- Handle inheritance, 183
- Hard-coded function calls for
  - payloads, 332
- Hardware viruses, 408–410
  - burning out hardware, 425
  - chip detection, 426–429
  - CIH, 417–421
  - EEPROM in
    - enabling read/write from, 417
  - Ethernet cards, 421–424
  - manufacturers of, 425
  - serial vs. parallel, 424–425
  - reading and writing
    - hardware memory, 410–417
- Hash loading for payloads, 335–336
- Hayes modem protocol,
  - reflection problem with, 211
- Headers
  - e-mail, 226–227
  - for memory blocks, 324–326
- Heap overflows, 279, 324–328



- HeapFree function, 326–327
- helpctr.exe, reversing, 105–111
- Hiding
  - attacker identity, 68–69
  - files and directories, 392–394
  - processes, 380
  - rootkit programs, 69
  - storage files, 69
- High-potency attacks in open-systems view, 46
- Highland addresses, 283–284
- History of software, 2–5
- Hollingsworth, Jeff, 87
- Holodeck tool, 78
- HOME environment variable, 297
- Honeypots, 203–204, 230
- Hooking. *See* Call hooking
- Horde IMP, injection with, 228–229
- Host-based fault injectors, 78
- host function, 218
- Hostile character sets, 245
- Hostile statement sets, 245
- HOSTNAME environment variable, 191
- Hot fixes, 86
- Hotmail, injection with, 229
- House of logic, 72
- Howard, Michael, 41, 59
- HPUX
  - buffer overflow in, 298
  - self-decrypting payloads on, 353–355
- HSphere, file traversal in, 188
- HTML
  - escape codes for, 208
  - injection with, 227
  - maxsize attribute in, 49
- HTTP
  - cookies in, 295–296
  - headers in, 216
  - query strings in, 216–217
- hunt\_address function, 115–119
- I LOVE YOU virus, 12–14, 217–218
- I-Planet Server, decompiling, 156–161, 258–263
- IceCast MP3 Server, URL encoding in, 273
- ID mode for chip detection, 427–429
- IDA (Interactive Disassembler)
  - batch analysis with, 114–121
  - with coverage analysis, 249
  - for decompiling, 104–106
  - for input path tracing, 156
  - for mapping runtime memory addresses, 157
  - for partition analysis, 243
  - plugins for, 92–104
  - for signed/unsigned mismatches, 313–315
  - tracking work with, 261
  - for white box analysis, 80
- IDC scripts, 114–121
- IDE channel interrupts, 440–441
- Identity, hiding, 68–69
- IDSs (intrusion detection systems), 237
  - alternate encodings with, 240–242
  - as reactive subscription services, 239–240
  - signature-based vs. anomaly-based, 238–239
  - signatures in, 173
- IDTs (Interrupt Descriptor Tables), hooking, 441
- IDv3 tags, 293
- IIS Server
  - elevated privileges in, 154
  - unicode encodings in, 272
- ILoveYou virus, 12–14, 217–218
- Implicit trust assumption, 48
- In-band signals, 204
  - C5 attack example, 205–206
  - for character injection, 209–211
  - in cross-site scripting, 212–217
  - history of, 204–205
  - with printers, 208–209
  - reflection with, 211
  - uses of, 207–208
- In instruction, 410
- In registers in SPARC, 341
- include function, 191
- Indirection in attack patterns, 68–69
- info command, 158–159
- info reg command, 258–259
- Information warfare (IW), 5–8
- Informix Database, 189
- Inheritance, permission, 183
- init function, 94
- Injection
  - character, 209–211
  - command. *See* Shell command injection
  - command injection
    - e-mail, 226–229
    - on file systems, 187–188
  - Injection points, 154–155
  - Injection vectors, 280–281
    - in attack patterns, 56–57
    - code address targets for, 282–283
    - existing code in, 286
    - number representation in, 284–285
    - registers in, 285–286
  - Input files, finding, 155
  - Input/output request packets (IRPs), 372
  - Input tracing
    - in reverse engineering, 84–86
    - in server software, 156–161
  - Inputs
    - in black box analysis, 80
    - crafted. *See* Crafted input
    - in open-systems view, 46
    - in partition analysis, 243
- Inside-out breakpoints, 157
- Instruction pointers
  - injection vectors with, 281–283
  - in MIPS processors, 338–339
- Intel interrupt request architecture, 439–441
- Intellectual property laws, 75
- Intelligence gathering, 6–8
- Intelligent devices, 31
- Inter-space branching, 349–350
- Inter-space trampolines, 351
- Interactive Disassembler. *See* IDA (Interactive Disassembler)
- Interactive shells, 438–439
- Internal states
  - mapping, 235–236
  - in software, 233–234

- Internet
  - adoption rate of, 4
  - connectivity with. *See* Connectivity
  - security on, 147
- Internet Explorer
  - content-based attacks on, 230
  - dotless IP addresses in, 274
  - GetObject call in, 225–226
- Internet toaster, 28
- Interrupt Descriptor Tables (IDTs), hooking, 441
- Interrupt request (IRQ)
  - architecture, 439–441
- Interrupts, 439
  - IDT hooking, 441
  - IRQ architecture, 439–441
  - Programmable Interrupt Controllers, 441–442
- Intrusion Detection Systems (IDSs), 237
  - alternate encodings with, 240–242
  - failures of, 239
  - as reactive subscription services, 239–240
  - signature-based vs. anomaly-based, 238–239
  - signatures in, 173
- Invisibility of clients, 150
- Inward operators, 206
- IP addresses
  - alternative, 274
  - in attack patterns, 67
- IPSwitch Imail, blind trusts in, 193
- IRC.DLL for backtrace code, 246–247
- IRPs (input/output request packets), 372
- IRQ (interrupt request)
  - architecture, 439–441
- ISO9660 file system, 429
- ITS4 program, 59
- IXIA tool, 81
- ixsso.query object, 226
- Java
  - buffer overflow in, 291–293
  - byte code disassemblers for, 105
  - configurable trust in, 166
  - extensibility of, 18
    - and .NET, 28–29
    - security. *See* Security, Java
- java.security.Policy class, 166
- Java Virtual Machine (JVM)
  - buffer overflows in, 291–293
  - encapsulation of, 25
  - extensibility of, 18
- Javascript, alert dialog attack in, 212
- JEDEC ID mode, 427–429
- jedec\_read\_id function, 428
- jedec\_read\_mfr function, 427
- jedec\_reset function, 428
- KeCancelTimer function, 416
- Kernel
  - buffer overflows, 445
  - infecting images of, 446
  - modifying, 69
  - patching, 397–408
  - in reverse engineering, 88
- Kernel-mode debugger, 78
- Kernel rootkits, 368–369
  - building, 370
  - checked build environment for, 369
  - drivers for, 370–377
  - files for, 370
  - writing, 369
- KERNEL32.DLL, 243
- KeSetTimerEx function, 416
- KeStallExecutionProcessor function, 412
- Key logging, 443–444
- Keyboard buffer injection, 210
- Keyboards
  - controller chips for, 444
  - interrupts for, 439, 442
  - reading and writing to, 411–417
- Keystroke monitors, 69
- KLOC (thousand lines of code)
  - in bug rates, 14–18
- Knowledge-driven models, 239
- Language-based attacks, 291
- LD\_LIBRARY\_PATH
  - environment variable, 189, 191
- ldil instruction, 347
- ldo instruction, 347–349
- Leading ghost characters, 267
- Leaf functions, 348–349
- Leaking data in shared buffers, 88–90
- Leapfrogging in code tracing, 251–253
- .LEAVE instruction, 346
- LED keyboard indicators, 411–417
- Legality of reverse engineering, 75–77
- Leveraging faults, 68
- li instruction, 338
- libc module, 297
- Licensing
  - ASP model of, 30
  - and reverse engineering, 75
- Linkage on PA-RISC, 352
- Linux
  - key loggers in, 443
  - terminal character injection in, 209–210
- Litchfield, David, 182
- Little-endian byte ordering
  - vs. big endian, 284–285
  - in MIPS processors, 338
- load\_file function, 95–96
- Loadable kernel modules (lkms), 443
- LoadLibrary function, 334
- Local branch instructions on PA-RISC, 350
- Local calls, weak, 219–224
- Local command-line utilities, 297–298
- Local filenames, URLs
  - replaced by, 225–226
- Local registers in SPARC, 341
- Local sockets, 181–182
- Location-based computation, 31–32
- Lock picks, 50–51
- Log files
  - corrupting, 266
  - manipulating, 275–276
  - overflow in, 296
  - privileges for, 151–152
  - for server software, 177
- Logging, key, 443–444
- Logic bombs, 151
- Logical program flow, 78
- Logically distributed systems, 26
- Long-term future of software, 30

- Love Bug virus, 12–14
- Lovelace, Ada, 2–3
- Low-level disk access, 429–430
- Low-level programming
  - languages, 16
- Lowland addresses, 283–284
- lr register in AIX/PowerPC, 356
- ls command, 294
- lsof command, 182
- lstrcpy function
  - finding, 121
  - in reverse engineering, 90–91
- ltrace tool, 160
- Lunt, Teresa, 238
  
- Machine code disassemblers, 78, 138–139
- Mailing lists, 10
- MailSweeper, injection with, 229
- Malicious input, crafting. *See* Crafted input
- malloc function, 327–328
- Managed-writable mechanism, 64
- Manufacturers, EEPROM, 425
- Mapping
  - API layer, 264–266
  - internal states, 235–236
  - memory, 390–392
  - network, 67
  - runtime memory addresses, 157
- Mars Lander, 11
- Master boot record (MBR), reading and writing, 429
- maxsize attribute, 49
- Measurement in reverse engineering, 87
- Medium-term future of software, 28–30
- Melissa virus, 218
- memcpy function, 126, 315
- Memory
  - buffer overflows in. *See* Buffer overflow
  - dumping, 259–260
  - hardware, reading and writing, 410–417
  - management of, 309–317
  - process snapshots for, 133–138
  - in reverse engineering, 126–127
  - writing to, 445
- Memory mapping
  - runtime, 157
  - for Trojan files, 390–392
- Memory page breakpoints, 253
- Message pumps, 247
- Meta-characters
  - in e-mail headers, 226–227
  - in equivalent requests, 268–271
  - in parsing, 254
- mflr register, 356
- Micromachines, 421
- Microsoft compiler flaw, 57–65
- Microsoft Developer Network (MSDN), 126
- Microsoft IIS Server
  - elevated privileges in, 154
  - unicode encodings in, 272
- Microsoft operating systems, lines of code in, 15–16
- Microsoft SQL Server 7, gray box analysis for, 82–84
- Microsoft Word, lines of code in, 14
- MidiPlug, overflow in, 294
- Military sites
  - honeypots in, 203–204
  - telephone system infiltration, 206
- Military systems
  - aircraft, 11–12
  - embedded, 287
- MIME conversions, 295
- MIPS-based payload
  - construction, 337
- MIPS instructions, 337–338
- Misclassification, 263–264
- Missile systems, 12
- Mitnick, Kevin, 180
- Mobile code, 18–19, 26–27, 30–31
- Modeling computers, 234
- Monitor programs for injection points, 154–155
- Motorola CPU, buffer overflow in, 289
- Mouse interrupt, 440
- MP3 files, XSS in, 217
- MSDN (Microsoft Developer Network), 126
- Multibyte number
  - representation, 284–285
- Multiplatform payloads, 358–360
- Multiple-command trick, 54
- Multiple operations, buffer overflow from, 298
- Multiple parsers in shell
  - command injection, 173
- Multiple paths of authentication, 198
- Multithreaded programs, 127–130
- Munging data, 298
- MV-22 Osprey, 11–12
- MyDialogProc function, 101–102
  
- %n format token, 321–322
- NASA Mars Lander, 11
- Navigation systems, 287
- NDIS library, 430–432
- NdisOpenAdapter function, 432
- NdisRegisterProtocol function, 432
- NdisRequest function, 432–433
- Negative values, buffer overflow from, 309–310
- Nesting function calls, 344
- .NET
  - extensibility of, 18–19
  - future of, 26
  - and Java, 28–29
- net start \_root\_ command, 377
- net stop \_root\_ command, 377
- netcat program, 150, 164–165
- Netscape, overflow in, 294
- Netscape I-Planet Application Server, decompiling, 156–161, 258–263
- netstat command, 181–182
- Netterm program, 208
- Network-based fault injectors, 78
- Network-based software, 5
- Network cards
  - EEPROM in, 421–424
  - finding, 433–438

- Network sniffers
  - for IDSs, 239
  - for OS stack identification, 66–67
- Network worms, 22
- Networks, 1
  - adoption rate of, 3–4
  - code for, 48
  - for driver support, 430–439
  - mapping, 67
  - scanning, 65
- NIDES intrusion detection system, 238
- NIMDA worm, 218
- nmap port scanner, 67
- Nonexecutable files, 165–166
- Nonexecutable stacks, 364–366
- nop instructions
  - in AIX/PowerPC, 360
  - in SPARC, 342–344
- NOT operator, 340
- NTFS file streams specifier, 264
- NULL buffers in helpctr.exe, 110
- NULL characters and terminators
  - in AIX/PowerPC, 358
  - in buffer overflow, 279, 331–332
  - in MIPS opcodes, 339–340
  - in payloads, 334–335
  - postfix, 186, 188
  - in reversing parser code, 257
  - in stack overflow, 301–308
- Number representation in injection vectors, 284–285
- Object sharing, design-level vulnerabilities in, 41
- Objects, future of, 28, 31
- Observability, 37–38
- Observable effects, removing, 69
- Off-by-one NULL termination, 304–308
- Oil tankers, embedded systems in, 287
- OllyDbg tool, 251, 253
- OnOpenAdapterDone function, 433
- OnOpenAdapterOne function, 432
- OnStubDispatch function, 373–374
- OnUnload function, 373–374, 415
- Opcodes, 337
- Open dynamical systems, 233
- Open-ended systems, 233
- open function in shell
  - command injection, 168
- Open shortest path first (OSPF), buffer overflow in, 281
- Open-systems view, 42–44
  - damage potential in, 45–46
  - exposure and potency in, 46–47
  - risk in, 44–45
- OpenDataSource function, 289
- OpenThread function, 128
- Operating systems
  - encapsulation of, 29
  - extensibility of, 18
  - future of, 24, 31
  - integration of, 24
- Oracle 9i, 182
- OS stack identification
  - attack patterns, 66–67
- OSPF (open shortest path first), buffer overflow in, 281
- Osprey aircraft, software failures in, 11–12
- Out instruction, 410
- Out registers in SPARC, 341
- Outlook application, injection with, 227–228
- Outlook View Control, injection with, 228
- Output events in attack patterns, 57
- Output points in partition analysis, 243
- Outside-in breakpoints, 157
- Outsourced software
  - backdoors in, 9
  - future of, 29–30
- Overflow, buffer. *See* Buffer overflow
- Overlapping packets, 241
- Overwriting
  - exception handler frames, 308
  - memory headers, 325
- PA-RISC payloads
  - construction of, 345–346
  - inter-space branching on, 349–350
  - inter-space trampolines with, 351
  - location of, 351–353
  - stacks in, 347–349
- Packets
  - desynchronization of, 240–242
  - leaking data in, 89–90
- Parallel EEPROM, 424–425
- Parallel port interrupts, 440
- Parameters
  - expansion of, buffer overflow from, 298
  - in shell command injection, 169–171
- Parser code, reversing. *See* Reversing parser code
- Parsing, 254
  - buffer overflows from, 289
  - commands, 264
  - delimiters for, 268
- Partition analysis, 242–244
- passwd command, 298
- Password limitations, 5
- Patches
  - binary code, 394–395
  - NT kernel, 397–408
  - peephole patches, 395–397
  - finding, 92
  - in reverse engineering, 75, 86, 132
- Patents, 76
- PATH environment variable, 191
- Patterns, 56. *See also* Attack patterns
- Payloads
  - activation of, 56–57, 281
  - in buffer overflow, 329–331
  - checksum/hash loading for, 335–336

- dynamic jump tables for, 333–334
- hard-coded function calls for, 332
- size of, 332
- XOR protection for, 335
- in injection vectors, 280–285
- memory locations for, 331–332, 338–339
- on RISC architectures, 336
  - in AIX/PowerPC, 356–360
  - branch delay with, 337
  - on HPUX, 353–355
  - instruction locations in, 338–339
  - MIPS instructions, 337–338
  - in PA-RISC, 345–353
  - in SPARC, 340–344
- PCL (printer control language) codes, 209
- PDAs, embedded systems in, 25
- Peephole patches, 395–397
- Perl
  - embedding within ASP pages, 164
  - system calls in, 164–165
  - taint mode in, 85–86
- Permissions
  - with ACLs, 183
  - on directories, 184
- Person-in-the-middle attacks, 29
- Phase space analysis, 193–198
- Phone phreaks, 204
- Phone systems
  - blue boxes for, 21, 204, 206
  - in-band signals with, 204–206
- PHP
  - command injection in, 173–174
  - global variables in, 190–192
- Phrack Magazine*, 40, 444–446
- Physical memory, writing to, 445
- Physical security, 147
- PICs (Programmable Interrupt Controllers), 441–442
- PIDs (process identifications)
  - in GDB, 158
  - for threads, 130
- Ping packets, 89
- PIT tool, 130
- Pointers and pointer operations
  - buffer overflow from, 60, 309–317
  - byte operations with, 256
  - in Prolog/Epilog, 362
  - in reversing parser code, 256
- Poison pills, 293
- Policies for trust, 166
- Policy class, 166
- poll function, 160
- Polymorphism, 329
- Pop-up windows from injection, 168
- Port scans, 66–67
- Ports on controller chips, 444
- POST requests, 191
- Postfix NULL terminators, 186, 188
- PostNuke content management system, 216
- Potency in open-systems view, 46–47
- Preloader, local filenames with, 225
- Primary IDE channel interrupt, 440
- Primary opcodes, 337
- Principle of least privilege
  - limitations of, 152
  - white listing in, 236
- Printer control language (PCL) codes, 209
- Printers and printing
  - data from memory, 319–321
  - in-band signals with, 208–209
  - printf function, 324
- Privilege escalation, 151–154
- Privileged resources
  - attacking, 70
  - programs writing to, 152
  - in server software, 151–154
- .PROC instruction, 346
- .PROCEND instruction, 346
- Process identifications (PIDs)
  - in GDB, 158
  - for threads, 130
- Process injection for hiding programs, 386
- Process-permissions equal trusts, 151
- Process records, removing, 383–386
- Processes
  - attaching to, 158–159
  - enumerating, 129–130
  - hiding, 380
  - for reading from untrusted sources, 153
  - in reverse engineering, 133–138
  - scheduling on server software, 180–181
  - spawning, handle inheritance in, 183
- Program execution flow, single stepping for, 130–131
- Program structure and logic, reverse engineering for, 73–75
- Programmable Interrupt Controllers (PICs), 441–442
- Programs using drivers, 372–373
- Prolog/Epilog code, 360–361
  - canary values in, 361–364
  - nonexecutable stacks in, 364–366
- Promiscuous mode, 432–433
- Protocol clarity in packet defragmentation, 242
- PROTOS tool, 81
- PS/2 mouse interrupt, 440
- Purify tool, 81, 83–84
- Quality assurance (QA) testing
  - limitations of, 14–15
  - overlooking, 82
- Query strings
  - in file system attacks, 188
  - XSS in, 216–217
- QueryDirectoryFile function, 392
- Race conditions
  - detecting, 40
  - in geographically distributed systems, 29

- Radar systems
  - embedded systems in, 287
  - flash ROM in, 409
- Raw packet interfaces, Java
  - support for, 292
- Reactive subscription services,
  - IDSs as, 239–240
- Reactive technologies, 236
- read function, 160
- Reading
  - enabling, EEPROM for, 417
  - hardware memory, 410–417
  - master boot record, 429
  - memory in reverse
    - engineering, 126–127
  - from untrusted sources, 153
- ReadProcessMemory function, 127
- ReadRegistry function, 432, 437–438
- Real-time clock interrupt, 440
- Rebooting for removing
  - observable effects, 69
- REC program, 105
- Recovery systems, 41
- Red pointing, 243–244
- Redirection
  - with CWD, 187
  - directory, 54
  - executing, 446
  - server-side page references, 230
  - Trojan executables, 386–392
- Reference monitors, 236
- Reflection
  - with in-band signals, 211
  - against trusted sites, 213–214
- Registering
  - drivers, 375–377
  - unload routines, 373
- Registers
  - for boron tags, 144–145
  - in buffer overflow, 285–286
  - examining, 259–260
  - in MIPS, 338
  - in SPARC, 341–342
- Registry keys
  - as attack targets, 70
  - controllable, 153
- regmon tool, 154
- Regular expressions, 173
- Relative path injection, 237
- Relative path traversal, 187
- Release guards, 205
- Remote attacks, 148, 288
- Remote procedure calls (RPCs), 181
- Remote xterms with server
  - software, 179
- Removing
  - observable effects, 69
  - process records, 383–386
- Ren, Chris, 58
- Replies, injection with, 227
- report\_out.txt file, 119–121
- require function, 191
- ResetPC function, 414
- Resource files, executable code
  - in, 165
- Resource IDs, 192
- ret instruction, 344
- Return addresses
  - in buffer overflow attacks, 60
  - for injection vectors, 282
- Reverse compilers, *See*
  - Decompilers
- Reverse engineering, 71–73
  - access requirement audits in, 90
  - API resources for, 90–91
  - automatic bulk auditing in, 111–121
  - black box analysis for, 80–81
  - breakpoints for, 123–126
  - code coverage for, 87–88, 139–145
  - cracking tools for, 121–139
  - debuggers for, 77–78, 121–123
  - decompiling in, 79, 104–111
  - development of, 8
  - disassembling in, 78, 104–105, 138–139
  - fault injection in, 78, 132–133
  - graphing for, 50–52
  - gray box analysis for, 81–84
  - I-Planet Server, 258–263
  - IDA plugins for, 92–104
  - input tracing for, 84–86
  - kernel access in, 88
  - leaking buffer data in, 88–90
  - legality of, 75–77
  - multithreading programs, 127–130
  - patching in, 75, 86, 132
  - process snapshots in, 133–138
  - purpose of, 73–75
  - reading and writing memory in, 126–127
  - red pointing in, 243–244
  - reversing parser code, 254
  - single stepping in, 130–131
  - version differences for, 86
  - white box analysis for, 79–80
- RevertToSelf function, 154
- RISC architectures, payloads
  - on. *See* Payloads
- Risk and risk assessment
  - actual, 47
  - defined, 37
  - in open-systems, 44–45
  - for vulnerabilities, 37–38
- ROM, 409–410
- Root access, need for, 152
- Rootkits, 367–368
  - advanced topics, 444–446
  - call hooking for, 380–386
  - detecting, 446
  - for hardware viruses. *See*
    - Hardware viruses
  - hiding, 69
  - for hiding files and directories, 392–394
  - for interrupts, 439–442
  - key logging, 443–444
  - for low-level disk access, 429–430
  - network support for drivers, 430–439
  - patching binary code, 394–408
  - Trojan executable
    - redirection, 386–392
- Routers
  - black box analysis for, 81
  - buffer overflow in, 281, 288
- RPCs (remote procedure calls), 181
- run function, 103
- Running processes, attaching
  - to, 158–159
- Runouts in code tracing, 247

- Runtime memory addresses, mapping, 157
- Runtime tracing, 247–250
- %s format string, 300–301
- SAM files, 171
- sample\_callback function, 93
- Satellites, exploitation of, 10
- save instruction, 343
- SCADA software weaknesses, 10
- Scancodes, 443
- scanf function, 301
- Scheduling processes, 180–181
- Scientific method in reverse engineering, 87
- Script kiddies, 35
- Scripting.FileSystemObject, 218
- Scriptlet.TypeLib, 218
- Scripts
  - buffer overflows from, 289
  - with client software, 217–229
  - cross-site. *See* XSS (cross-site scripting)
  - embedding
    - in nonscript elements, 215
    - in scripts, 164–165
    - misclassification with, 263
  - scrrun.dll file, 219–224
  - Scrubbing problem in Ethernet, 89–90
  - SeAccessCheck function, 398–400
  - Search paths in configuration files, 156
  - seccinit.c file, 61
  - seccook.c file, 61
  - secfail.c file, 61
  - Second serial port interrupt, 439
  - Secondary IDE channel interrupt, 441
  - Secret variables, 190
  - Secrets and Lies* (Schneier), 1
  - Securing Java* (McGraw and Felten), 18, 25, 27
- Security, 33
  - boron tags for, 438
  - for buffer overflows, 59
  - on Internet, 147
  - in network-based software, 5
  - through obscurity, 10
  - software vs. application, 44
  - \_\_security\_check\_cookie function, 61
  - Security Engineering* (Anderson), 1
  - \_\_security\_error\_handler function, 61
  - Security error handlers, 58
  - Security flaws, reverse engineering for, 76
  - Security testing, 82
  - Segmented GET requests, 241–242
  - Self-decrypting payloads, 353–355
  - Self-organizing systems, 32
  - SendKeyboardCommand function, 414–415
  - Sendmail, overflow in, 295
  - Serial EEPROM, 424–425
  - Serial port interrupts, 439–440
  - Server control of client software, 202–203
  - Server-side page reference redirects, 230
  - Server software, 147–148
    - adding users, 180
    - authentication in, 192, 198
    - blind trust in, 192–193
    - configure trust in, 161–166
    - environment variables in, 189–192
    - error code checking in, 199
    - exploring file systems, 184–189
    - FTP, 178
    - injection points in, 154–155
    - input path tracing in, 156–161
    - with local sockets, 181–182
    - permissions inheritance in, 183
    - phase space analysis in, 193–198
    - privilege escalation problem in, 151–154
    - process spawning in, 183
    - remote xterms with, 179
    - scheduling processes on, 180–181
    - session IDs in, 193
    - shell command injection in. *See* Shell command injection
    - TFTP, 180
    - trusted input problem in, 149–151
  - Service outages from worms, 22
  - ServiceName value, 433
  - Session authentication, 192
  - Session IDs
    - cookies for, 192
    - in server software, 193
    - \_\_set\_security\_error\_handler function, 61
  - SetBreakpoint function, 125
  - SetEIP function, 127
  - SetLEDS function, 415
  - SetSingleStep function, 131
  - setsnap function, 137–138
  - SetSystemInformation function, 377
  - setuid utility, 153
  - sfc.dll file, 445
  - sfcfiles.dll file, 445
  - Shared buffers, leaking data in, 88–90
  - Shell code in embedded systems, 288
  - Shell command injection, 167–168
    - through arguments from other programs, 169–171
    - for binary file building, 175–177
    - delimiters in, 172–175
    - fluttering windows from, 168
    - for text file building, 175
  - Short-term future of software, 24–28
  - Signature-based IDSs, 173, 238–239
  - Signed/unsigned mismatches, 310–315
  - Signed values in memory management, 315–317
  - SignedBy property, 166
  - Simple script injection, XSS in, 214–215
  - Single-step flag, 131



- Single stepping
  - in reverse engineering, 130–131
  - in runtime tracing, 247–248
- Size
  - buffer, 110, 279
  - payload, 332
- Slashes (/) in alternate encoding, 268–270, 274–275
- Sliding registers in SPARC, 341
- slti instruction, 339–340
- Smart objects, 28
- SmartBits tool, 81
- Smashing the stack, 300
- Snapshots, process, 133–138
- Sniffers
  - for IDSs, 239
  - for OS stack identification, 66–67
- Social engineering in C5 attacks, 206
- Sockets, server software with, 181–182
- SoftIce debugger, 78, 251
- Software
  - bad, 11–14
  - defined, 2
  - essential, 3
  - future of, 23–32
  - vulnerabilities, 33
- Software copy protection limitations, 9
- Software distribution, future of, 30
- Software Fault Injection* (Voas and McGraw), 37, 78
- Software licensing, ASP model of, 30
- Software security vs. application security, 44
- Software testing, difficulties in, 234
- Solaris systems
  - buffer overflow in, 298
  - target models for, 159–160
- Sound card interrupt, 440
- Source code
  - decompilers for, 79
  - vs. executable, 10
  - in white box analysis, 79–80
- SOURCES file, 370
- SourceScope tool, 90, 301
  - for buffer overflow attacks, 59
  - for white box analysis, 80
- Space characters in equivalent requests, 268
- SPARC systems
  - function call nesting in, 344
  - payload construction in, 340–344
  - register windows in, 341–342
  - stacks on, 342–344
- Special characters in parsing, 254
- Special-purpose computational units, 28
- Special-purpose OSs, 29
- Speedbreaks, 250–251
- Spike tool, 81
- Spoofing, 230
- SpoonFTP, triple-dot vulnerability in, 267–268
- Spreadsheets, XSS in, 217
- sprintf function, 301
- Spying, 6–8
- SQL Server 7, gray box analysis for, 82–84
- SQL statements, buffer overflows from, 289–290
- Stack traces for helpctr.exe, 107
- Stacked applications, conceptual view of, 43
- StackGuard tool, 58–61, 361–364
- Stacks and stack overflow, 300–301
  - attack patterns, 66–67
  - in buffer overflow, 278–279
  - in C++, 58–60
  - exception handler frames overwriting in, 308
  - fixed-size buffers in, 301–302
  - injection vectors for, 283–284
  - NULL termination in, 302–308
  - on PA-RISC, 347–349
  - on SPARC, 342–344
- StackShield tool, 60
- Statement sets, hostile, 245
- States
  - in buffer overflow, 279–280
  - mapping, 235–236
  - in open-ended systems, 233
  - in software, 233–234
- Static analysis tools for buffer overflow attacks, 59. See also SourceScope
- Static strings in buffer overflow, 300
- Statistical windows in anomaly-based IDSs, 238
- Stealth activities, 7
- Steganography, 7
- stepti command, 352
- Storage files, hiding, 69
- Stored procedures
  - buffer overflows from, 290, 293
  - in Oracle 9i, 182
- strcat function, 301
- strcpy function
  - buffer overflow from, 58, 63, 290, 301
  - in reverse engineering, 90
- Stress testing, 48
- String functions in buffer overflow attacks, 58, 279, 300
- strlen function, 303, 310
- strncat function, 304–305, 310
- strncpy function, 300, 303
- Subopcodes, 337
- Subscription services, 27
- Subversive programs
  - classification of, 7
  - defined, 367–368
- SWIFT network, 22
- Switches, black box analysis for, 81
- Symbolic Links, overflow in, 294
- Synchronization of packets, 240–242
- syscall function, 340
- syslog function, 324
- System calls
  - hooking, 381
  - for reverse engineering, 90
  - user-supplied variables passed to, 185
- System directories as attack targets, 70

- System file protection,
  - disabling, 445
- system function, 52–53
  - buffer overflows from, 290
  - Perl calls to, 164–165
  - in Prolog/Epilog, 365–366
  - in shell command injection, 168
- System timer interrupt, 439
- SystemLoadAndCallImage function, 377
- Systems
  - privileges for, 151–152
  - software as, 237
- T-SQL (transact SQL)
  - protocol, 289–290
- Tags
  - boron. *See* Boron tags
  - overflow in, 294
- Taint mode in Perl, 85–86
- takesnap function, 135–137
- Tankers, embedded systems in, 287
- Target components in attack
  - patterns, 67–68
- Target software, 37
- TARGETPATH environment
  - variable, 370
- Taxonomy of attack patterns, 38–41
- Taylor UUCP daemon, 296
- TCP/IP
  - packet defragmentation in, 242
  - ports as entry points, 72
- Technology adoption rates, 4
- TELNET environment
  - variables, 189
- Temporary files, 184
- TERM environment variable, 297
- term function, 95
- Terminals
  - character injection in, 209–211
  - escape codes for, 207–208, 210
- Testing methodologies, fault
  - injection for, 78, 132–133
- Text files, shell command
  - injection for, 175
- TFTP (Tiny FTP), 180
- The PIT tool, 130
- Thousand lines of code
  - (KLOC) in bug rates, 14–18
- Threads, enumerating, 129–130
- Three-dimensional phase space
  - plot of points, 195–196
- Time to market pressures, 5
- timerDPC function, 416
- Timing attacks, detecting, 40
- Timing issues
  - in EEPROM, 421
  - in geographically distributed systems, 29
- Tiny FTP (TFTP), 180
- Titan application firewall, 273
- traceroute tool, 67
- Tracing code. *See* Code tracing
- Tracing input. *See* Input
  - tracing
- Trade secrets, 10
- Tradecraft, digital, 6–8
- Trampoline attack, 65, 350–353, 362–363
- Transact SQL (T-SQL)
  - protocol, 289–290
- Transaction-based systems, 41
- Transport-level security, 29
- TRAP FLAG, 130
- Traversal, file system, 187–188
- Trigger filters, 239
- Trillian chat client, 246–247
- Trinity of trouble, 14
- Triple-dot vulnerability, 267–268
- Tripwire, redirection with, 386–387
- Trojan executable redirection, 386
  - drivers for, 387–392
  - with Tripwire, 386–387
- Trunk lines, controlling, 205–206
- Truss tool, 159–160
- Trust issues
  - in buffer overflows, 293
  - design-level, 41
  - input-based, 149
  - in Java, 291
  - in server software, 149–151, 161–166
  - with users, 49
- Trusted sites, reflection
  - against, 213–214
- Turing machines, 233
- Two-stage buffer overflow
  - attacks, 40–41. *See also* Trampoline attack
- Type confusion attacks in Java, 291
- Type safe languages, 63, 277
- TypeLib, attacks on, 218
- Undisclosed exploits, 10
- Unicode encoding in
  - equivalent requests, 271–272
- Uniform Computer
  - Information Transactions Act (UCITA), 77
- Uniform resource identifier
  - (URI) data, locating
  - routines for, 159–160
- Uniform resource locators
  - (URLs)
  - equivalent requests, 273–275
  - passing local filenames in place of, 225–226
  - trust assumptions in, 49
- Universal Turing machines, 233
- UNIX environment variables, 189
- UNIX-to-UNIX copy program
  - (UUCP), 153
- Unloadable drivers, 373–374
- Unsafe languages, 277
- Unsigned/signed mismatches, 310–315
- Untrusted sources, reading
  - from, 153
- URI (uniform resource
  - identifier) data, locating
  - routines for, 159–160
- URLs (uniform resource
  - locators)
  - in equivalent requests, 273–275
  - passing local filenames in place of, 225–226
  - trust assumptions in, 49
- US Vicennes software failures, 12–13
- User-controlled filenames, XSS
  - in, 217
- User interfaces for server
  - software, 150

- User-mode debuggers, 78
- User-supplied configuration files for elevated privilege, 153
- User-supplied variables, passed to file system calls, 185
- U usernames in attacks, 52–53
- Users, adding, 180
- UTF-8 encoding, 273
- UUCP (UNIX-to-UNIX copy program), 153
  
- Valgrind debugger, 81
- Variables
  - in buffer overflow attacks, 64, 294, 296–298
  - in PHP, 190–192
  - in server software, 189–192
  - user-supplied, 185
- Version differences for reverse engineering, 86
- Vessel Traffic Management Information System (VTMIS), 287
- Virtual machines (VMs)
  - buffer overflows in, 291–293
  - encapsulation of, 25
  - extensibility of, 18
- VirtualQuery function, 250
- VirtualQueryEx function, 137
  - breakpoints for, 124–126
  - for memory querying, 126–127
- Virus checkers as reactive technology, 236
- Viruses
  - in client scripts, 217–218
  - development of, 20
  - hardware. *See* Hardware viruses
  - poison pills for, 293
- Visibility of faults, 68
- Vitek, Ian, 175–176
- von Bertalanffy, Ludwig, 42
- Voyager spacecraft, 278
- vsprintf function, 301
- VT terminal escape codes, 207
- Vtables, 329
- VTMIS (Vessel Traffic Management Information System), 287
  
- Vulnerabilities
  - backtracing from, 245–247
  - collections of, 38
  - defined, 39–40
  - design-level, 41–42
  - increases in, 33
  - risk assessment for, 37–38
- VxWorks OS
  - in embedded systems, 287
  - flash ROM in, 409
  
- WaitForKeyboard function, 412–413
- wcsnecat function
  - in helpctr.exe, 107–111
  - in WINNT, 112–113
- WDASM disassembler, 105
- Weak local calls, finding, 219–224
- Web browsers and ActiveX, 224–225
- Web code and XML, 27
- Web logs, 275
- Web servers
  - command-line parameters with, 162
  - ghost characters with, 267
  - misclassification of, 263
- Web spoofing, 230
- Webalizer program, 216
- WEP (wired equivalent privacy) encryption algorithm, 25
- White box analysis, 79–80
  - vs. black box analysis, 82
- White lists
  - vs. black lists, 236, 149
- White space in equivalent requests, 268
- Whitehat Security Arsenal* (Rubin), 1, 171
- Whittaker, James, 78
- Winamp program, 293
- Windows operating systems
  - disabling system file protection for, 445
  - heap headers in, 325
  - kernel patching in, 397–408
  - key loggers in, 443–444
  - lines of code in, 15–16
  - message pumps in, 247
  - wcsnecat function in, 112–113
- Wired equivalent privacy (WEP) encryption algorithm, 25
- Wireless systems
  - future of, 25, 29
  - hiding attacker identity in, 68–69
- Word, lines of code in, 14
- Workstation vulnerabilities, 16
- Worms
  - operation of, 20–21
  - service outages from, 22
  - write\_eeprom function, 424
- WriteProcessMemory function, 127
- Writing
  - enabling, EEPROM for, 417
  - hardware memory, 410–417
  - kernel rootkits, 369
  - master boot record, 429
  - memory in reverse engineering, 126–127
  - to physical memory, 445
  - to privileged resources, 152
- Writing Secure Code* (Howard and LeBlanc), 44, 151, 217
- WSARecv function
  - backtracing to, 246
  - in partition analysis, 243
  - in reverse engineering, 88
- WSARecvFrom function, 84
- WSASend function, 243
- Wscript.network, attacks on, 218
- WScript.Shell, attacks on, 218
- wsprintf function, 248–249
  
- x command, 259–260
- X Windows, backdoors on, 179
- x86 debuggers, 121–123
- x86 feature set, 121
- XML markup language, 27
- XOR protection, 335
- XSS (cross-site scripting), 212
  - in HTTP headers, 216

- 
- in HTTP query strings, 216–217
  - in Javascript alert dialog attacks, 212
  - in reflection against trusted sites, 213–214
  - in simple script injection, 214–215
  - in user-controlled filenames, 217
  - xterms with server software, 179
  - Xtlib, buffer overflow in, 297
  - Zalewski, Michael, 196
  - Zone transfers, 67
  - Zuse, lines of code in, 14
  - ZwCreateProcess function, 389
  - ZwCreateSection function, 387, 389
  - ZwOpenFile function, 387–389











