Symbols and Numbers
\\ (backslashes)
  vulnerability to URLs with, 280, 291
  XSS attacks with, 252–253
  10 Immutable Laws of Security, 500

A
Accept-Language header, 68
access control entries. See ACEs
access control lists. See ACLs (access control lists)
access issues
  access level identification, 15
  ACEs for setting. See ACEs (access control entries)
  ACLs of. See ACLs (access control lists)
  attacks based on. See elevation of privilege (EoP)
  DACLS of. See DACLS (Discretionary ACLs)
  SACLs of, 305
  user interaction with. See permissions
Access Violations (AVs), 128–129, 153
AccessEnum tool
table of discoverable permissions, 307
viewing permissions with, 309–310
accessibility testing, 1
ACEs (access control entries)
  access rights, 306
  container access control, 315–316
  defined, 306
denies, 317
Everyone group, 312–313
flags in, 306
information in, 306
inheritance, 306
ObjSD exe tool for viewing, 311–312
ordering within DACLS, 318
SIDs in, 306
ACK response role in handshakes, 86
ACLs (access control lists)
  ACEs of. See ACEs (access control entries)
  container access control, 315–316
  DACLS. See DACLS (Discretionary ACLs)
  indirect access to resources, 319
  locally accessible objects, 327–328
  logon rights, 314–315
  ordering of ACEs in, 318
  OS's supporting, 306
  owners of objects, 318–319
  race condition attacks, 320–322
  SACLs, 305
  securable objects, 304–305
  security check for, 304
  security descriptors, 305
types of, 305
user access token creation, 304
Windows services, 325–327
Acrobat bugs. See Adobe Acrobat action property
cross-site scripting attacks with, 227
of HTML forms, 60
active scripting setting, 449
ActiveX controls
  ActiveX Control Test Container tool, 455–456
  arbitrary code running, example, 480
  AXDetail tool, 41
  BHOs, 463–464
  binary behaviors, 466–467
  Browser Helper Objects, 463–464
  bug reports, 452
  catch statements for errors, 459–460
  causes of repurposing bugs, 444–445
  CLSIDs for, 438, 441–442
codebase attribute for trigger installs, 445–446
COM interfaces, 438, 441, 445
COMRaider tool, 456
copying to Clipboards, 470
  crashes, causing, 458
  creating in Internet Explorer, 438–440
data types indicating vulnerabilities, 463
  deceptive operations of, 458
defined, 438
determining those installed with an application, 441–442
dialog box spoofing, 477–478, 481
discovering how a control works, 451–452
  DlCanUnloadNow function, 447
  editors, silently launching, 473–478
  entry point potential of, 40
events of, 440
  exception handling, 458, 459–460
  exe files, launching, 475, 476
  expandos, 466–467
  external programs, sending data to, 479–480
  file system issues, 457
  forced connections, 74
  HTML namespaces, 466–467
  IDDispatch objects, 463
  IDL/ODL files, 455
  information disclosure issues, 458, 459–460
  initialization, 442–445, 447–448
  installation, 445–446
  instantiation, 445–447
  interfaces implemented by, determining, 450–451
  Internet Explorer security, future of, 438

531
addressing security bugs

ActiveX controls, continued
1ObjectSafety, 448, 449
IUnknown for object creation, 446
JavaScript for creation, 439–440
kill bit, 446–447, 450–451
legacy control vulnerability, 446
list of attempted actions, 457–458
member-level threat modeling, 456–458
member manipulation by HTML, 467
members, adding without security reviews, 445
members, enumeration of, 451, 452
members, multiple needed for exploits, 471
members, testing walkthrough of, 466–468
methods of, 440
multiple copies of for attacks, 485
nested objects, 461–463
Object Browser tool, 453
object calls vs. DOM calls, 462
object tags for creating, 438–439
OLE View tool, 453–455
Outlook View Control bug, 461–463
PARAMs, 440, 468, 476–477, 484–486
pasting from Clipboard, 470–472
persistent objects, 463–464
persistent properties, 447–448, 455
ProgIDs, 440
properties of, 440
RealPlayer XSS vulnerability, 257
registry, alteration by controls, 484
registry, marking safe in, 450
repurposing vulnerability of, 441, 443–444
resource use, 438
return values in, 460–461, 468–469
reverse engineering of, 452
safe, falsely marked as, 444
Safety Detailer tool, 450–451
safety testing scripts, 449
script, accessing control contents through, 470–472
scripting, 445, 449
SDK for, 452
security model overview, 445
security UIs with, 473
server redirection, 465
SPI (safe for initialization), marking as, 450–451
signing as trusted, 446
SiteLock, 224, 444
source code examination, 452
spoofing prompts example, 477–478
summary, 487
test cases, 452
Test Container tool, 455–456
testing methodology, 451
testing tips, 486–487
testing walkthroughs, overview of, 466–468
tools for testing, 451–456
TypeLib viewer, 453–455
VARIANT type, 463
VBScript for creation, 439
Web page use of, learning from, 452
Windows Media Player example in IE, 438–439
XSS attacks with, 240–241, 465–466
addressing security bugs
communicating with bug finders, 505
mitigation, 506
patch release, 506–507
related bugs, 505–506
reporting internally, 504
root cause identification, 505
testing fixes, 506
time required for, 505
version specificity, 506
Adobe Acrobat
forced connections with, 74
Reader, XML external entity bug, 270
ADODB connection vulnerability, 241
Advanced Guestbook script injection attack, 68
Affix bug, 123
algorithms
costs (CPU usage), 341–343
reverse engineering, 431–434
Anonymous group, 314
Anonymous Logon group, 312–313
anonymous pipes, 34
ANSI, Unicode expansion bugs, 170
APIs
guidelines for safe creation of, 140
memory, reading data into, 422
root cause guideline, 433
search path issues, 289–290
viewing calls to with Log Viewer, 414
AppDomains, 359–360, 382
application crashes from DoS, 335–336
Application Domain policy level, 356–357
applications
fingerprinting, 115
information disclosure issues, 109–110
approach to security testing, general, 3–4
AppVerifier tool, 312
APTCA
determining if assemblies are marked as, 379–380
functionality of, 380
Internet Explorer issues, 379
luring attacks, 375–378, 380–381, 382
manifest information showing, 379–380
purpose of, 375
reasons for using, 379
testing, 380–381
arbitrary server connections, 74, 82
array and numeric bounds errors, 121
ASP (Active Server Pages)
- encoding functions, common, 261
- functions raising XSS issues, 259, 260
- HTTP redirection, 99
- IIS data stream bug, 284
- Response.Write XSS bug example, 260–261

ASP.NET
- automatic data encoding, 261–262
- canonicalization vulnerability example, 280
- cross-site scripting vulnerability, 350–351
- CSRF attack prevention, 494
- filters for preventing XSS attacks, 255–256
- HyperLink controls, 351
- sandboxing, 359–360
- server redirection with ActiveX controls, 465
- SOAP requests and CSRF attacks, 495

assemblies
- authenticode signatures, 355
- declarative style, 356
- decompiling, 381
- defined, 354
- demands, 360–362
- evidence, 355
- FullTrust, 359
- hash values, 355
- imperative style, 356
- interactions with CAS, 353–354
- .NET permissions, 355–356
- partially trusted code, 359
- policies for, 356–357
- stack walks, 360–365
- strong names of, 354–355

assembly language
- NOPs for patching code, 419
- pushing parameters to the stack, 423
- string compares, 424
- tracing backward to an event, 417–419
- unsafe calls, determining, 429–430
- viewing with debuggers, 415

asserts
- complementing with demands, 368–369
- issues to look for, 368
- over-asserting, 369
- purpose of, 363, 364
- reducing scope of, 369–370
- testing tips for, 382
- ASX files, faking for IE, 249
- audit policies, SACL specification of, 305
- authenticated user threats, 16
- Authenticated Users group, 314
- authentication
  - passwords for, Sec passwords
  - reverse engineering example with, 432–434
- URLs including, 298
- authenticode signatures, 355
- automated code review, 166–167

B
- backslashes (\), vulnerability to URLs with, 280, 291
- XSS attacks with, 252–253
- bandwidth consumption, 346–347
- Batch group, 313
- batched transaction SQL injection attacks, 399–400
- BHOs (Browser Helper Objects), 463–464
- binary behaviors, 466–467
- binary code
  - authentication code reverse engineering example, 432–434
  - disassembling, 145
  - editors, 93, 100
  - information disclosure issues, 109
  - native vs. bytecode, 425–427
  - patching, 416–420
  - security patches, analyzing, 434–435
  - symbol files, 426, 435
  - testing tips, 435–436
  - WinHex tool, 111–113
- black box components
  - examining by observation. See observation of programs
  - examining by reverse engineering. See reverse engineering
  - examining with debuggers. See debuggers
- black box testing
  - defined, 20
  - format string attacks, for, 193
- Black Hat conference, 9, 81
- BoundsChecker, 163
- braces, buffer overflows from, 148
- breakpoints
  - client source code, 81
  - code tracing user input with, 430–431
  - setting on memory, 421–424
- Browser Helper Objects (BHOs), 463–464
- browsers. See Internet Explorer, Web browsers
- buffer overflows
  - access violations indicating, 153
  - AFFIX bug, 123
  - API creation guidelines, 140
  - areas to look for bugs, list of, 138
  - attacker mentality, assuming, 123–124
  - BoundsChecker, 163
  - braces in strings, 148
  - callbacks, 139
  - changed behavior as indicator of, 158–163
  - code hypotheses, 141
  - code, reading to determine boundaries, 142
  - Code Red worm, 122
  - common numerical limits, 142–143
  - comparison errors causing, 144
  - compound document vulnerability, 146
  - compressed data issues, 145–146
  - constructing test data, 144–148
  - copying data, 168
  - CPU registers as evidence of, 152–153
buffer overflows, continued
- crashes as symptoms of, 152–153
- data copying issues, 168
- data format expectations, 141
- data structures for testing, 150
- defined, 121
- delimiters in data, 148
- dependency vulnerabilities, 147–148
- discovery scenario, 124
- documents, finding in, 139
- duplicate data size storage, 168–170
- e-mail Web pages threat, 17
- embedded object vulnerabilities, 148
- encoded data issues, 145–146, 176–177
- encrypted data issues, 145–146
- encumbered data, 145–146
- enumerations as vulnerabilities, 147
- evaluating seriousness of, 171–172
- exception handler overwrites, 129, 153–158
- expected data templates, 141
- iterations to testing, 144
- iTunes, 123
- LBL traceroute Exploit, 137
- LCLint tool, 167
- log event overflows, 139
- Logon.exe example, 177–179
- maintaining overall data integrity, 144–148
- managed code with, 350
- memory allocation for test data, 146
- memory spikes indicators of, 158
- Microsoft Visual Studio tool, 125
- MIME for exploits, 176–177
- minimum overflow necessary for exploits, 171
- Morris Worm, 122
- network attacks, finding, 138–139
- Nimba worm, 122
- null termination failures, 171, 172
- null values causing, 148–149
- numeric bounds error, subclass of, 121
- offset reference attacks, 138
- offsets in test data, 146
- OLE DocFiles, 146
- other attack methods, 138
- overwriting data, 149–150
- parsers as sources of, 170
- path expansion bugs, 171
- Pizza example, 172–176
- pointer reset bugs, 171
- Prefast tool, 167
- primary actions, 151
- prioritizing test cases, 151–152
- privilege elevation with, 139
- process memory, 128–129
- processor-specific issues, 125
- programmable interface vulnerabilities, 140
- programming language susceptibility to, 121, 151
- queries to databases, 147
- quotation marks in strings, 148
- recognizable data for test starts, 142
- references in test data, 146–147
- replacing data, 149–150
- RPC attacks, 139
- RSS attacks with, 267
- runtime tools, 163–165
- S/MIME with Outlook Express, 45
- secondary actions, 151
- serv2 example, 158–163
- sharing between higher and lower privileged users, 139
- size field mismatches with file size, 413
- SOAP, 148
- SQL Slammer, 122
- stack overflow type, 124, 125–129
- string test length determination, 142–144, 150
- symptoms of, overview, 152
- test case prioritization, 151–152
- testing overview, 138
- testing tips, 182–183
- thoroughness of testing, 123
- try-catch blocks insufficient to stop, 155
- types of, 124
- Unicode causing, 170, 176
- URL encoding based, 171
- white box testing, 166–167
- XML, ill-formed, 147
- ZLIB bug, 123
- bugs
  - causes of, overview, 4
  - noncrucial, threats from, 21
- Bugtrap, 9, 503
- Bypass Traverse Checking right, 43–44
- bytecode, 425–427
full demands, 360–361
FullTrust, 359
inheritance demands, 362
link demands, 361–362, 370–372
.NET permissions, 355–356
overview of interactions of, 353–354
partially trusted code, 359
PermitOnly security action, 364–365
policies for assemblies, 356–357
privileges to system resources, removing, 352
sandboxing, 359–360
stack walks, 360–365
testing tips, 382–383
user security model, 353
user vs. code security, 352–353
casing of filenames, 286–287
CBV (callback verification), 6–7
CDATA, 265–267
CDO (Collaboration Data Objects) entry points, 46
CERT (Computer Emergency Response Team), 503
certificates, SSL, 76, 82
Character Entity References, 266–267
characters
ANSI to Unicode expansion bugs, 170
cntrol characters. See control characters for spoofing
cntrol characters for spoofing, 93–95
DBCS, 170
homoqraph attacks, 97–98
testing suggestion, 100–101
vulnerabilities from lack of canonicalization. See
canonicalization issues
whitelisting, 251–253
CHM (Compiled Help Module) files defined, 243
eample of XSS in, 243–244
# (hash marks) in scripts, 243–244
HTML Help Workshop tool, 243
protocol handlers, exploiting with, 244
XSS vulnerability of, 236
Cisco CallManager denial of service attack, 334–335
Class IDs (CLSID’s), 438, 441–442
classes, inheritance demands, 362
client/server interaction
arbitrary server connections, 74
client examples, 51
client source code breakpoints, 81
custom clients for malformed requests, 55
documentation, examining, 52
finding normally acceptable requests, 52
hooking programs, 55
malicious server connection causes, 73. See also server
responses
manipulating requests, 54–58
monitoring network traffic, 52–54
proxy requests, creating malformed, 55–58
request processing, 52
server address specification, 58
single request construction, 54–55
536 client-side scripts

client/server interaction, continued
sniffing traffic, 53–54
source code, examining, 52
TCP request modification, 56–58
wget tool for custom requests, 54–55
client-side scripts, finding XSS bugs in, 244–246
clipboard operations, 470–472
CLR (Common Language Runtime)
bytecode interpretation, 425
garbage collection, 351–352
purpose of, 349
stack walks, 360–365
CLSIDs (Class IDs), 438, 441–442
CMCs (Common Messaging Calls), 46
code groups
classes of, 358
defined, 357
example of, 358–359
membership conditions, 357
multiple conditions with, 357–358
permission sets, 357
purpose of, 353–354
code library bugs, 435
Code Red worm, 122
code reviews
APTCA. See APTCA
assert issues, 368–370
automated, 166–167
exception handling issues, 372–375
finding problems with, overview of, 365–366
FxCop tool, 365, 381, 382
link demand issues, 370–372
marshaling data issues, 367–368
resources to help with, 365
SQL scripting attacks, for, 400–403
testing tips, 382–383
unsafe code, calls to, 366–368
Collaboration Data Objects (CDOs), 46
COM (Component Object Model)
ActiveX interfaces, 438, 444, 445
Browser Helper Objects, 463–464
COMRaider tool, 456
test point potential of, 40
EoP attack using, 328
IUnknown for object creation, 446
listing all installed components, 442
marshaling data for, 367–368
OLE View tool, 41, 453–455
stack corruption from, 153
.com extension checks, defeating, 282–283
command-line argument entry points
finding, 47–48
importance to attackers, 47
injection attacks, 47
low danger level of, 47
Process Explorer tool, 47–48
protocol handlers with, 47
comments in styles, XSS vulnerability, 254–255
Common Language Runtime. See CLR (Common Language Runtime)
Common Messaging Calls (CMCs), 46
comparison errors, buffer overflows from, 144
Compiled Help Modules. See CHM (Compiled Help Module) files
compilers, unnecessary code removal by, 426–427
compound documents, 146
compressed data, 145–146
Computer Emergency Response Team (CERT), 503
COMRaider tool, 456
conferences, security, 9
Confirm Open After Download flag, 26–27
connection timeouts, 340
container access control issues, 315–316
Content-Length headers, 70–71
content of files, information disclosure from, 109–113
control characters for spoofing
dialog box formatting, 93–95
table of useful characters, 95
testing, 100–101
wildcard DNS, 95–96
controls. See also HTML forms
ActiveX. See ActiveX controls
ASP.NET, 350–351
hidden input control, 62
password controls, 61
types of, 61
cookies
cross-site scripting attacks with, 225–226
defined, 65
domain property, 66
echoed script, retrieving with, 223
expires property, 65–66
HTTPOnly property, 66
issuance of, 67
name/value property, 65
path property, 66
PlaySMS bug, 68
properties of, 65–66
retrieval of, 67–68
secure property, 66
testing for tampering, 67
copy protection schemes, bypassing, 415–420, 435
copying data, buffer overflows from, 168
copyright protections, 436
CPU registers, buffer overflow effects on, 152–153
CPU resources, DoS attacks against
algorithm costs, 341–343
analyzing CPU performance, 341–343
encryption, 343
file name specification with POST, 341–343
goals of attacks, 341
recursive calls, 343–344
CPU usage reports, information disclosure from, 117
crashes, buffer overflows as symptoms of, 132–133
CreateProcess class, 289–290
credentials, using URLs for, 298
cross-site request forgery attacks. See CSRF (cross-site request forgery) attacks
cross-site scripting (XSS) attacks
  actions enabled by, list of, 223–224
  ActiveX controls allowing, 465–466
  ADODB.connection vulnerability, 241
  applications, modeling on SP2, 251
  ASP bug examples, 260–261
  ASP.NET automatic data encoding, 261–262
  ASP.NET Web control vulnerability, 350–351
  attacker data validation and encoding, 260–261
  attacker-supplied data, functions for reading, 259–260
  attribute escapes, 234
  backslashes, escaping from, 252–253
  binaries, running using local files, 240–241
  CHM files. See CHM (Compiled Help Module) files
code review to find, 259
  comments in styles, 254–255
  content returned, identifying places for, 259
data fields used in, table of, 230–231
  data in script variable fields, 233–234
defined, 220
document.location elements, 245
elements to check for XSS vulnerability, 245–246
  encoding functions, common, 261
  encoding prevention method, 232–233, 254–255
  error case handling, 232
  escaping string variables as defense, 252–253
event vulnerabilities, 234–235
  filters for preventing, 251–253
  Flash Player vulnerability, 256–257
  finding bugs in client-side scripts, 244–246
  functions returning data, table of, 259
  HTML escape code issues, 294–295
  identifying vulnerabilities, 258
  Internet zone links to My Computer zone, 256–257
  JavaScript vulnerability, 235–236, 252
  local files, reflected attacks against, 236
  local files, vulnerability example, 236–237
  location.hash elements, 245, 253
  location.search elements, 245
  non-HTML files parsed as HTML, 248–250
  NULL characters in tags, 254
  outerHTML property, 246
  parsers, internal operations of, 253–254
  persisted. See persistent cross-scripting attacks
  PHP bug examples, 260–261
  POST method for, 226–228
  query strings vulnerability, 230
  quotation marks, 252–253, 261
  reflected. See reflected cross-site scripting attacks
  res protocol vulnerability, 242
  resources, XSS bugs in, 241–243
  src property of IMG tag vulnerability, 235–236
  script disabled default for IE, 257–258
  script tag issues, 232
  scripts for reflected server attacks, 225–226
  scripts inside other scripts, 253–254
  Service Pack 2 changes, 250–251
  Shell.Application control vulnerability, 241
  SiteLock vulnerability, 224
  src attribute, 246
  SSL not protection against, 226
  style vulnerabilities, 235, 254–255
  User-Agent header vulnerability, 231
  user interface for testing, issues with, 231
  UserData vulnerability, 224
  VBScript vulnerability, 236
  Winamp vulnerability, 246–248
  Windows Media Player bug, 249–250
  zone elevation vulnerability, 224–225
cryptography reverse engineering, 432
  CSRF (cross-site request forgery) attacks
  automatic logon facilitation of, 493
defined, 492
  POST data for, 493–494
  prevention methods, 494
  query strings URLs for, 492–493
  server state defense flaws, 494
  SOAP data for, 495, 496
  testing for, 496
  validation defense against, 494
  XSS techniques for, 493

D
DACLs (Discretionary ACLs)
ACEs, effects of, 312
container access control, 315–316
defined, 305
DELETE permissions, 313
deny ACEs, 317
Everyone group, 312–313
FILE_ADD_FILE permissions, 313
FILE_DELETE_CHILD permissions, 313
large groups, table of, 313–314
missing, effect of, 312
NULL, 306, 317
ObjSD.exe tool for viewing, 311–312
ordering of ACEs in, 318
weak, finding, 312
WRITE_DAC permissions, 313
WRITE_OWNER permissions, 313
data copying buffer overflows, 168
Data Flow Diagrams. See DFDs (Data Flow Diagrams)
data stream canonicalization issues, 284
DBCS (Double Byte Character Set), 170
D COM (Distributed Component Object Model)
access issues, 327
Dcomcnfg.exe tool, 41
entry point potential of, 40
DDoS (distributed denial of service) attacks, 347
depth
advantages for understanding programs, 415
assembly language, advantage to viewing, 415
breakpoints, setting on memory, 421–424
buffer data, viewing, 423
code tracing steps for patches, 416–417
cumbersome data, viewing with, 145
exception-based overflows, detecting, 153–158
execution flow, modifying, 415–420
IDA Pro, integrated with disassembler, 430–431
memory, viewing with, 420–424
modifying the binary example, 419
NTSD debugger, 160
OllyDbg tool, 416
patching binaries, 416–420, 422
processes already running, attaching to, 417
registration schemes, understanding, 421–424
restrictions on debugging, bypassing, 415–416
string compares, 424
symbol files, 435
testing modifications, 419–420
testing tips, reverse engineering, 435–436
tracing backward to an event, 417–419
understanding programs with, overview of, 415
Debugging Tools for Windows, Microsoft, 414
declarative style, 356, 369
decoders, sniffer, 54
deepth
assemblers, decompiling, 381
bytecode with, 425
defined, 424–425
.NET Reflector, 425
results, using for security reviews, 426
symbols, results without, 425–426
testing tips, 435–436
decompression bombs, 269, 346
Defcon conference, 9
defense-in-depth threat model assumptions, 16
definition of security testing, 1
delimiters, buffer overflows from, 148
demands, 360–362, 368–369, 370–372
denial of service (DoS) attacks
application crashes, 335–336
bandwidth consumption, 346–347
CallManager attack, 334–335
connection timeouts for, 340
CPU resources, attacks against, 341–344
decompression bombs, 346
defined, 333
digital signatures for, 340
disk space consumption, 345–346
distributed DoS attacks, 347
DOS device name canonicalization issues, 287–288, 339
e-mail Web pages threat, 17, 20
test
encryption algorithms, 343
handle leaks from, 337
impact of, 333
implementation flaws, 334–340
log file filling, 345–346
memory consumption, 344–345
memory leaks, 336–339
Performance Monitor for detecting, 337–339
poorly designed feature flaws, 336
privileges required for, 333–334
recursive calls, 343–344
resource consumption flaws, 334, 340–347
resource leaks, 336–339
as STRIDE category, 17
summary, 348
surfaces (exposure) of applications, 334
testing tips, 348
types of, 333–334
types of vulnerable resources, 340
deny ACEs, 317
deny modifier for stack walks, 363–364
dependency vulnerabilities, 147–148
Depends.exe, 210
Detours, Microsoft, 55
device names
denial of service (DoS) attacks, 339
DOS name canonicalization issues, 287–288
DFDs (Data Flow Diagrams)
access level identification, 15
data accepted, identifying, 13
information disclosure, identifying with, 103
purpose of, 13
Web application example, 13–14
dialog box spoofing
control characters for, 93–96
default selection, 101
overview of, 91–93
reformatting with control characters, 93–95
testing, 101
Z-order spoofing, 96–97
digital certificates, 76, 82
digital signatures, 340
directories
ACLs for, 315–316
default locations, vulnerability of, 237–238
junctions, 323
predictability of as vulnerability, 249
Write access issues, 316
directory traversal attacks, 267, 281
disassemblers
code tracing strategies, 430
defined, 424–425
determining whether attackers can control data, 430–431
finding dangerous functions, overview of, 427
format string vulnerabilities, 428–431
functions calls, understanding, 428
IDA (Interactive Disassembler) tool, 428
native code, recommended for, 425
printf() example, 429–430
reverse engineering algorithms, 431–434
security patches, analyzing, 434–435
testing tips, reverse engineering, 435–436
unsafe calls, 429–430
disclosure. See information disclosure
disk space
bandwidth consumption, 346–347
decompression bombs, 346
DoS attacks consuming, 345–346
log file filling, 345–346
quotas for, 345
distributed denial of service (DDoS) attacks, 347
DLLs (Dynamic Link Libraries)
redirection of, 316
search path issues, 289–290
DNS (Domain Name System)
opoisoning scenario, 74–75
reverse lookups, spoofing, 88–89
spoofing with wildcards, 95–96
document format repurposing attacks
advantages of requesting external data, 489
dangerous external data request types, 489–490
elevation of privilege danger, 489
Excel functionality, 490
mitigation for, 490
spreadsheets, 489–490
SQL queries in, 489, 491
storing data issues, 490
testing procedures, 491
tools for testing, 491
document.location elements, 245
DOM (Document Object Model)
local file access to, 238
outerHTML property, 246
XSS attacks on, 223–224
domain name parsing issues, 296–297
DoS attacks. See denial of service (DoS) attacks
DoS devices
denial of service (DoS) attacks, 339
device name issues, 287–288
dotless IP addresses, 296–297
Double Byte Character Set (DBCS), 170
double encoded characters, 293–294
downgrade MITM attacks, 80–81
DRM (Digital Rights Management), defeating, 420

E

e-mail entry points
anonymity advantage, 45
buffer overflow in Outlook Express, 45
CDO, 46

CMC, 46
development message, 45
FileMon tool, 46
finding, 46
FormMail script, 90
header issues, 46
importance to attackers, 45–46
large number of users advantage, 45
MAIL FROM forgeries, 46
MAPI, 46
POP, 45
port monitoring, 46
preferred virus vehicle, 45
process description, 45
RCPT TO forgeries, 46
S/MIME, 45
SharePoint Services icons, 46
SMTP servers, 45
spoofing SMTP, 89–90
e-mail Web pages
code execution threats test cases, 20
denial of service test cases, 20
DFT for, 13–14
everification of threats for, 17
HTML scripts, hiding, 294
mail bombing test cases, 19
repudiation test cases, 20
spamming test cases, 19
Easter eggs, 110
EBPs (stack frame pointers), 129, 172
ECX register, 197–200
eDirectory, DOS device name bug, 287
eDoc tool, 111–112
EIPs (extended instruction pointers), 129, 172
elevation of privilege (EoP)
buffer overflow attacks with, 139
COM (Component Object Model) vulnerability, 328
document format repurposing attacks, 489
exception handling vulnerabilities, 373–375
locally accessible objects for, 327–328
multiple-stage attacks, 302
named pipes for, 34–35
permissions allowing, 302
SQL injection attacks for, 387
STRIDE category, 17
encoded data
ASP.NET automatic data encoding, 261–262
attacker data verification for XSS attacks, 260–261
buffer overflows from, 145–146, 176–177
encoding detection by IE, 255
encoding functions, common, 261
HTML encoding, table of, 232–233
method for preventing XSS attacks, 232–233
XSS attacks, preventing, 254–255
encoded URLs

double encoded characters, 293–294
examples of variations in, 290
hexadecimal escape codes, 290–291
UCS-2 encoding, 293
UTF-8 encoding, 291–292
encryption
buffer overflows from data, 145–146
DoS attacks targeting, 343
cumbered data, 145–146
Enterprise policy level, 356–357
entities, HTML, 295
entities, XML
defined, 268
external entities, 270
infinite entity reference loops, 268–269
URL references with, 268
XML bombs, 269
entry points
command-line arguments as, 47–48
common, list of, 24–25
defined, 23
e-mail See e-mail entry points
environment variables as, 48–49
files as. See files as entry points
finding, 14–15
format string attacks, identifying for, 193
guidelines for, 25
HTTP requests as, 31–33
mIRC, 38
named pipes causing, 34–38
plugable protocol handlers as possible, 38–39
points of failure associated with, 23
programmatic interfaces as potential, 40–41
rankings by risk, 24
registries as, 41–44
risk assessments of, 24
server responses as possible, 39–40
sockets as, 29–31
SQL as source of, 41
testing guidelines, 23
user interfaces as, 44
Viewpgls.exe tool, 39
enumeration of entry and exit points, 14–15
enumeration of threats
authenticated users, 16
defense-in-depth, 16
defined, 15
gain knowledge tip, 16
listing all, importance of, 18
STRIDE categories, 17
think maliciously tip, 16
tips for threat identification, 16–18
understanding related threat models, 16
Web page example, 17
enumerations, buffer overflows from, 147
environment variables
defined, 48
entry points, as, 48–49
Environment Variables dialog box, 48–49
finding entry points, 48–49
importance to attackers, 48
Process Explorer tool, 49
telnet client attack, 78–79
USERDOMAIN, 78
EoP. See elevation of privilege (EoP)
error conditions
expired trial period example, 417
information disclosure from, 115–117
resource leaks, as indicators of, 337
ESPs (extended stack pointers), 125
Ethereal tool
cookies, viewing, 67–68
HTTP request monitoring, 33
monitoring network traffic with, 31
network disclosure monitoring, 113–114
panes of, 53
request monitoring, 52–54
server responses, monitoring, 40
telnet traffic example, 57–58
cTree eDoc tool, 111–112
events, HTML, 234–235
Everyone group, 312–313
evidence
purpose of, 353–354
types of data in, 355
EvilPipe tool, 37–38
EvilServer tool, 80
Excel, Microsoft, 489–490
exceptions
ActiveX controls, problems with, 459–460
elevation of privilege attacks, 318, 373–375
filtering, 318
information disclosure bugs, 375, 382
overwrites of handlers, 129, 153–158
purpose of, 372–373
reverting permissions failures, 319–320
testing for, 382
EXEC command, SQL, 397–398, 408
evaluation flow, modifying with debuggers, 415–420
exit points, finding. See enumeration of entry and exit points
expandos, HTML, 466–467
expired software, patching, 416–420
extensions, filename. See filename extension checks,
defeating
F
Favorites list, GET method data storage by, 61
fetchmail vulnerability, 108
file API
   named pipe client security, 37–38
   SHandleInfo function, 98
file caching vulnerability, 79
File Download dialog box, 26–27
File Monitor, 106–107
file permissions, 27–28
FILE_ADD_FILE permissions, 313
FILE_DELETE_CHILD permissions, 313
FileExtInfo.exe tool, 29
FileIOPermission, 355–356
FileMon tool
   e-mail entry points, 46
   named pipes, viewing, 37
   non-HTML files parsed as HTML, 250
   purpose of, 27–28
filename extension checks, defeating
   associations with, determining, 29
   checks, defeating. See filename extension checks, defeating
   com extension, 282–283
   example code for blocking extensions, 282
   extensions that do not matter, 284–285
   GetClassFile API, 284–285
   hiding, dangers of, 98
   NTFS data stream canonicalization issues, 284
   precedence of extensions, 282–283
   trailing characters, 282, 283
   white lists recommended, 282
filenames
   associations with extensions, 29
   buffer overflows using, 149
   casing canonicalization issues, 286–287
   creating illegal characters in, 288
   DOS device name canonicalization issues, 287–288
   entry points as, 26
   extensions. See filename extension checks, defeating
   predictability of, 108–109
   short vs. long canonicalization issues, 285–286
   specification with POST, DoS attacks, 341–343
   trailing periods, removing, 341–343
   UNC shares, 288–289
files
   backups, vulnerability from, 119
   buffer overflows from parsing, 139
   content inspection, 109–113
   deleted file vulnerability, 107
   entry points. See files as entry points
   File Monitor, 106–107
   FileMon. See FileMon tool
   finding files being used, 105–107
   hidden extension issues, 98
   information disclosure overview, 104
   linked. See linked files
   metadata in, 110–111
   names of. See filenames
permission vulnerabilities, 105–106
predictability of filenames, 108–109
Process Explorer, 105–106
safe locations for, 107–108
storage issues, 107–109
Strings tool, 111–112
temporary storage of, 108
ZIP file information disclosure, 112–113
files as entry points
   access by applications criteria, 27–28
   basis of, 25
   browsing Windows, opening by, 26–27
   file names, 26
   filename extensions, 29
   image files, threat from, 26
   importance to attackers, 26
   parts usable as entry points, 25–26
   permissions issues, 27–28
   registered file types, 26–27
   unopened files, threat from, 26
   unregistered file types, 27
   filters for preventing XSS attacks, 251–256
   finding entry and exit points. See enumeration of entry and
   exit points
fingerprinting, 115
firewalls, 406
fixed width data fields, 147
Flash Player XSS vulnerability, 256–257
forceful browsing attacks, 59
format specifiers. See also format string attacks
   %d, 186–190
   defined, 186
   %n, 190–193
   output per specifier limitations, 201–203
   %s, 186–190
   table of functions using, 193
   %x, 197
format string attacks
   address formatting issues, 206–207
   analyzing exploitability, 195–197
   assembly coding for exploit payloads, 211
   black box testing for, 193
   buffer contents, writing, 186–187
   buffer references as parameters, 186–189
   C language specification basis of, 185–186
   calc.exe address example, 212–213
   defined, 124
   disassemblers to find vulnerabilities, 428–431
   draft exploit example, 211–212
   EAX assignment issue, 204–206
   entry point identification, 193
   exception generation, 217
   finding the vulnerability example, 194–195
   first parameter of printf function, injection with, 188
   format specification symbols. See format specifiers
542 FormMail script

format string attacks, continued
format strings defined, 186
fprintf(), 186
machine-independent exploits, 216
%n, disabling, 192–193
non-Windows versions of, 185
null bytes problem for attackers, 203–204
null terminator creation, 216
offsets for payloads, determining, 213
output per specifier limitations, 201–203
overwriting memory with, 190–191
overwriting stack return addresses, 200–201
payload creation example, 210–217
popping stack values down to a target, 190, 199
printf(), 186, 208–209
register manipulation example, 195–197
registers, setting to useful values, 197–200
return addresses, placing payloads at, 209–210
reviewing code for, 192–193
scanf(), 186
stack, effects of parameters, 187
stack interpretation at run time, 188–190
strings for testing, 217
summary, 218
table of functions using format specifiers, 193
testing for, 191–192, 213–217
unanticipated string specifiers, 188–190
Unicode characters for, 192
walkthrough of an attack, overview, 194
Weex format string vulnerability, 193
WinExec address insertion, 212
working around exploitability, overview of, 197
FormMail script, 90
forms, Web, 226–228
fprintf(), 186
frames, URL redirection with, 99
FrontPageServer Extensions, Microsoft, 57
FTP forced connections, 74
full demands, 360–361
Full Disclosure, 9, 503
FullTrust, 359
fully trusted code, 359. See also APTCA
function declaration buffer overruns, 140
functionality testing software, 1
functions
code library bugs, 435
dangerous function searches, 166, 427
parameters pushed onto stack, 423
unsafe calls, determining, 429–430
fuzz testing
buffer overflow tests, 165–166
common vulnerabilities found, 69
COMRaider tool, 456
defined, 69
dumb vs. smart fuzzing, 69
Hailstorm tool, 69
define file fuzzers, 69
Peach, 69
recommended, 72
requirements for success, 165–166
resource for, 70
SPIKE, 69
FxCop tool, 365, 381, 382

G
GAC (global assembly cache)
FullTrust requirement, 360
gacutil.exe tool, 41
garbage collection memory leaks, 351–352
general approach to security testing, 3–4
GET requests
CSRF attacks using, 492–493
HTML forms use of, 60–61
tampering with query strings, 62
Gflags tool, 163–165
global assembly cache. See GAC (global assembly cache)
groups
Anonymous group, 314
Authenticated Users group, 314
Batch group, 313
dangerous large groups, table of, 313–314
everyone group, 312–313
Guests group, 314
Interactive group, 313
Local group, 313
Network group, 313
Remote Interactive Logon group, 314
Resultant Set of Policy data, 315
Service group, 314
Users group, 314
Whoami.exe, 315
/GS compiler switch, 179–182
guestbooks XSS attack example, 228–229
guests group, 314
GUIDs vulnerabilities, 109

H
Hailstorm tool, 69
Half-Life server response entry point, 40
handle leaks from DoS, 337
hard links, 323–324
hashes, reverse engineering, 432–434
headers, HTTP, 70–71
heap overruns
allocations followed by frees, 153
defined, 124
differences in heap functionality, 137
free function exploitation, 137
Gflags tool, 163–165
heaps, purpose of, 136
JPEG COM vulnerability, 137
LBL trace route Exploit, 137
possible exploitation types, 137
resources on, 137
stacks for completing exploits, 137
helloPostDemo.asp, 226–228
help files, 236, 243–244
hexadecimal escape codes, 290–291
hexadecimal IP addresses, 297
hiberfil.sys, 107
hidden information caveat, 435
hidden input control, 62
History file, GET method data storage by, 61
hobbyists, security, 3
homograph attacks, 97–98
hooking requests, 55
hot spots. See WiF
HTA file vulnerability, 241
HTML forms. See also controls
action property, 60
controls in, 61–62
GET method, 60–61, 62
hidden input control, 62
importance of securing, 59
method property, 60–61
password controls, 61
POST method, 61, 63
proxies for modifying data. See proxy requests, HTTP,
malforming
removing fields for testing, 72
sample form, 59–60
tampering with query strings, 62
URL target specification, 60
validation, client-side, 72
HTML Help Workshop tool, 243
HTML (Hypertext Markup Language)
ActiveX control member manipulation, 467
comments in styles, 254–255
document.location elements, 245
entities, 295
escape code canonicalization issues, 294–295
events for XSS attacks, 234–235
expandos, vulnerability from creating, 466–467
exploitability of, 219
files on local drives, 236
filtering to prevent XSS, 251–253
forms. See HTML forms
# (hash marks), 236–237, 243–244, 245
Help files using, 219–220
HTA file vulnerability, 241
local file XSS vulnerability. See local HTML file XSS
vulnerability
location.hash elements, 245, 253
location search elements, 245
meta tag refresh, 99
namespaces, vulnerability from creating, 466–467
non-HTML files parsed as HTML, 248–250
outerHTML property, 246
persisted scripting attacks. See persistent cross-site
scripting attacks
playlists, vulnerability of, 246–248
proxies for modifying HTTP traffic. See proxy requests,
HTTP, malforming
query string issues. See query strings
Request.QueryString, 386reflected scripting attacks. See
reflected cross-site scripting attacks
resources, XSS bugs in, 241–243
src property of IMG tag vulnerability, 233–236
script disabled default, 257–258
script redirection, 99
scripting attacks. See cross-site scripting (XSS) attacks
Service Pack 2 Internet Explorer changes, 250–251
src attribute, 246
Styles for XSS attacks, 235
XSS attacks with. See cross-site scripting (XSS) attacks
HTML scripting attacks against RSS readers, 267
HTTP (Hypertext Transfer Protocol)
Accept-Language header, 68
compressed requests, 346
Content-Length headers, 70–71
defined, 31
form input. See HTML forms
headers, tampering with, 68
importance of, 58–59
input sources, list of, 59
length specification issues, 345
proxies for modifying traffic. See proxy requests, HTTP,
malforming
redirection attacks, 98–99
Referer headers, 62, 68, 90, 114
requests. See requests
responses. See server responses
RFC 2616, 31
security additions, 33
Set-Cookie headers, 67
splitting attacks, 225
statelessness of, 59
traffic, Ethereal tool for monitoring, 31
URL redirection attacks, 98–99
User-Agent headers, 68, 91
Web browser clients, 59
HTTP requests
Content-Length headers, 70–71
custom for testing, 70–71
entry points created by, 31–33
Ethereal tool, 33
element of, 31
finding entry points, 33

HTTP responses

HTTP requests, continued
- guidelines to determine if entry points, 32
- importance to attackers, 33
- Middleman tool, 33
- out-of-order tip, 71
- permissions, testing with lower, 72
- POST method syntax, 61
- proxies for modifying. See proxy requests, HTTP, malformed
- removing fields for testing, 72
- subparts, breaking into, 32
- testing tips, 71–72
- validation, client-side, 72
- Wftech tool for creating, 70–71

HTTP responses
- cookies in, 67
- e-mail Web pages information disclosure, 17
- URL redirection attacks, 98–99
- Human Interactive Proofs, 19
- Hyperlink controls, ASP.NET, 351
- hyperlinks, URL homograph attacks, 97–98

IDA (Interactive Disassembler) tool, 428
iDefense file fuzzers, 69
idq dll buffer overflow attack, 122
IIS (Microsoft Internet Information Server), 284 images
- opening, threat from, 26
- Web beacons, using as, 2, 114–115
- imperative style, 356, 370
- ImpersonateLoggedOnUser function, 139
- impersonation, 325, 497
- Imperva Inc., Interactive TCP Relay, 55–56
- implied disclosures, 118–119
- infinite entity reference loops, 268–269
- information disclosure
  - ActiveX controls for, 458, 459–460
  - attacker technique overview, 104
  - binary files, viewing, 111
  - data flow diagrams for identifying, 103 defined, 103
  - deleted file vulnerability, 107
  - e-mail Web pages, threat to, 17
  - Easter eggs, 110
  - eDoc tool, 111–112
  - error messages as, 115–117
  - exception handling causes, 373–375, 382
  - file backups, vulnerability from, 119
  - file content inspection, 109–113
  - file data disclosure overview, 104
  - File Monitor, 106–107
  - file storage issues, 107–109
  - filename predictability, 108–109
  - finding files being used, 105–107
- fingerprinting, 115
- functions not in use, 106
- /GS switch vulnerability, 181–182
- hidden web pages, 119
- HTTP Referer, 114
- identifying interesting data, 117
- implied disclosures, 118–119
- login error messages, 116–117
- metadata, 110–111
- monitoring network data, 113–114
- network disclosures overview, 113
- obfuscation of data, 117–118
- permission vulnerabilities, 105–106, 108
- privacy vulnerabilities from, 104
- Process Explorer, 105–106
- race condition attacks, 106
- safe file locations, 107–108
- sequential data, 118–119
- storage issue check list, 107
- STRIDE category, as a, 17
- Strings tool, 111–112
- temporary file storage, 108
- threat models for identifying, 103
- underestimation of, 103
- user name disclosure, 103–104
- Web beacons, 114–115
- WinHex tool, 111–113
- Word, Microsoft, 111
- ZIP files, 112–113
- inheritance demands, 362
- injection attacks. See LDAP injection attacks; SQL injection attacks; XPath
- input tracing review, 166
- inputs, potential maliciousness of, 23
- integer overflows
  - addition code example, 130–131
  - C++ code example, 131–136
defined, 124, 129
- memory allocation with, 131–136
- shopping cart math example, 130–131
- signed data type operations, 130
- signed short number limits table, 130
- signed vs. unsigned operations, 130
- testing guideline, 136
- intellectual property issues, 436
- Interactive group, 313
- Interactive SIDs, 314
- Interactive TCP Relay, 55–56
- interfaces
  - programmatic. See programmatic interfaces
  - spoofing. See user interface spoofing
  - user. See user interfaces
- internal access modifier, 380
- international testing, 1
- internationalization casing canonicalization, 286–287
Internet Explorer
active scripting setting, 449
ActiveX controls running in. See ActiveX controls blocking Internet zone links to My Computer zone, 256–257
Browser Helper Objects, 463–464
COMRaider tool, 456
cross-site request forgery attacks. See CSRF (cross-site request forgery) attacks
DOM vulnerability, 223–224
encoding detection feature, 255, 293
Flash Player vulnerability, 256–257
IObjectSafety, 448, 449
kill bits, 446–447, 450–451
local file XSS vulnerability, 239
MIME changes, 250
nested object vulnerability, 461–463
non-HTML files parsed as HTML, 248–250
NULL characters in tags, 254
Outlook View Control bug, 461–463
parsers, internal operations of, 253–254
pop-up blocker changes, 250
RealPlayer vulnerability, 257
repurposing attacks. See ActiveX controls, Web page repurposing attacks
resource protocol vulnerability, 242
script disabled default, 257–258
script support, 449
scripting engine vulnerability, 461–463
server redirection, 465
sniffing behavior, 249
SP2 changes for, 250–251
Trident rendering engine. See Trident
UserData feature, 224
XSS attacks with ActiveX controls, 465–466
Z-order spoofing, 96–97
zone elevation blocks, 251
zone types described, 240
zone vulnerability, 224–225
Internet zone, 240, 256–257
interoperability, 437
Intranet zone, 240
IP addresses
dotless IP addresses, 296–297
hexadecimal, 297
IPv4 format, 296–297
IPv6 canonicalization vulnerabilities, 297
reverse DNS lookup spoofing, 88–89
socket hijacking scenario, 73–76
spoofing techniques, 86–87
stealing using MAC address spoofing, 88
IPSec (Internet Protocol Security), 81
IPv6, canonicalization vulnerabilities of, 297
irc, entry points created by, 38
ISA (Microsoft Internet Security and Acceleration Server) bug, 434–435
ISNs (sequence numbers), spoofing, 86
iTunes buffer overflow vulnerability, 123
IUnknown, 446
J–K
Java format string attacks, 185
JavaScript
ActiveX control creation, 439–440
filtering to prevent XSS, 252
forced connections, 74
time property of IMG tag vulnerability, 235–236
junctions, 323
kernel32.dll, 210
kill bits, 446–447, 450–451
L
LBL traceroute Exploit, 137
LCLint tool, 167
LDAP injection attacks, 409
legal issues, 436
LIKE clause, 394–395
link demands, 361–362, 370–372, 381
LinkDialogSpoof sample program, 91–92
linked files
access issue overview, 322
canonicalization issues, 324–325
defined, 322
hard links, 323–324
junctions, 323
security concerns with, 324–325
shortcut files compared to, 322
symbolic links, 322–323
xbreaky bug, 324–325
links. See hyperlinks
Linux
AFLix bug, 123
fetchmail vulnerability, 108
format string attacks, 185
LoadResource API, 242
Local group, 313
local HTML file XSS vulnerability
ADOBD connection vulnerability, 241
binaries, running, 240–241
CHM (Compiled Help Module) files, 243–244
content, obtaining, 239
dangers from, 238–239
document.location elements, 245
document example file, 236–237
exploitation of bugs, 237–238
finding bugs in client-side scripts, 244–246
# (hash marks), 245
HTA file vulnerability, 241
Local Service account

local HTML file XSS vulnerability, continued
Information bar warnings in IE, 239
location hash elements, 245
location search elements, 245
My Computer zone, running in, 238
outerHTML property, 246
overview of, 236
res protocol vulnerability, 242
resources, XSS bugs in, 241–243
Shell.Application control vulnerability, 241
src attribute, 246
URLs, appending victim data to, 238–239
Local Service account, 326
Local System account, 326
localization, 1
locally accessible objects, 327–328
location hash elements, 245, 253
location search elements, 245
log files
buffer overflows from events, 139
DNS spoofing, 88
DoS attacks filling log files, 345–346
Log Viewer tool, 414
Logger tool, 414
spoofing with control characters, 94–95
logins
automatic, facilitation of XCRF attacks, 493
error message information disclosure, 116–117
logon rights, 314–315
Logon.exe buffer overflow example, 177–179
long filenames, 285–286
lstrlen function, 414
luring attacks, 375–378, 380–382

M
MAC addresses
filtering, 87
Mac MakeUp tool, 87
spoofing, 87–88
Machine policy level, 356–357
MacOS format string attacks, 185
mail bombing, 17, 19
mailing list security news, 9
malicious server responses. See server responses
malicious thinking. See thinking maliciously
man-in-the-middle (MITM) attacks
defined, 29–30
downgrade MITM attacks, 80–81
malicious server connections from, 74
Man in the Middle tool, 55–56
managed code
advantages of, 349
APTCA. See APTCA
ASP.NET cross-site scripting, 350–351
assemblies. See assemblies
buffer overflow issues, 350
code reviews. See code reviews
Common Language Runtime for. See CLR (Common
Language Runtime)
entry point potential of, 40
FullTrust, 359
garbage collection, 351–352
memory leaks with garbage collection, 351–352
myths about, overview of, 350
native functions, calling from, 367
partially trusted code, 359
partially writing applications in, 350
sandboxing, 359–360
SQL injection attack vulnerability, 352
testing tips, 382–383
unmanaged code, calls to, 351, 367
unverifiable code, 350
user security model for, 353
vulnerability to attacks, 349
manual linear reviews, 166
MAPI (Messaging Application Program Interfaces), 46
mapped drives, 288–289
marshaling data for unsafe code, 367–368
memory
access of, watching in debuggers, 424
Access Violations (AVs), 128–129
addresses of, 128
APIs for reading data into, 422
breakpoints, setting on, 421–424
collection in DoS attacks, 344–345
debuggers, viewing with, 420–424
duplicate data size bugs, 168–170
integer overflows in allocation of, 131–136
managed code with memory consumption, 351–352
overwrites. See buffer overflows
process memory, 128–129
spikes as overflow indicators, 158
WinHex tool, 421
ZeroMemory function, 426–427
memory leaks
garbage collection issues, 351–352
vulnerability to DoS attacks, 337
merchandise returns malicious thinking example, 7–8
message repurposing, 496–497
metadata information disclosure, 110–111
Metasploit Project, 304
method property of HTML forms, 60–61
Microsoft Debugging Tools for Windows, 414
Microsoft Detours, 55
Microsoft Excel, 489–490
Microsoft Intermediate Language (MSIL), 354, 425
MiddleMan HTTP proxy
EvilServer tool with, 80
MiddleMan tool, 33
network disclosure monitoring, 113–114
POST data modification with, 64
MIME (Multipurpose Internet Mail Extensions)
buffer overrun exploits using, 176–177
Internet Explorer treatment of, 248
SP2 IE changes, 250
mIRC entry points, 38
MTM attacks. See man-in-the-middle (MTM) attacks
Mitnick, Kevin, 84
modular programming, 437
monitoring tools
advantages for black box component observation, 413
Ethereal. See Ethereal tool
FileMon. See FileMon tool
Logger tool, 414
NetMon, 139
RegMon, 43
monitors, vulnerabilities from, 117
Morris Worm, 122
MP3 file playlist XSS vulnerability, 246–248
MSDN (Microsoft Developer Network), 161
MSIL (Microsoft Intermediate Language), 354, 425
mutexes, accessibility of, 327–328
My Computer zone
advantages to attackers, 238
binaries, XSS attacks running, 240–241
defined, 240
Internet zone links to, 256–257
lock downs by SP2, 250
persistent cross-site scripting attacks in, 246–251
script disabled default, 257–258
Service Pack 2 Internet Explorer changes, 250–251
Shell Application control vulnerability, 241
SP2 locks on, 240
Winamp vulnerability, 246–248

N
name resolution, 54. See also DNS (Domain Name System)
named pipes
anonymous pipes compared to, 34
client security, 37–38
connection creation, 34
convention for accessing, 34
CreateNamedPipe function, 35–36
defined, 34–38
defile API access to, 37–38
FileMon for viewing, 37
finding entry points, 35
flags, 36–37
hijacking, 36–37
impersonation of clients, 37
importance to attackers, 34–35
permission checks, 36
purpose of, 34
reference paper on, 35
SQL Server 2000 vulnerability, 35
namespaces, HTML, 466–467
native code, 425–427
NCRs (Numeric Character References), 266–267
nested objects in ActiveX controls, 461–463
.NET
CLR. See CLR (Common Language Runtime)
decomposing assemblies, 381
permissions, 328, 355–356
Reflector decompiler, 425
Remoting, entry point potential of, 40
sandboxing, 359–360
security issues, 349. See also managed code
Netcat tool
socket hijacking vulnerability testing, 76
telnet client attack, 78
NetMeeting bug, 434–435
NetMon tool, dangers from, 139
Netscape browser JPEG COM vulnerability, 137
Netstat.exe tool, 30–31
network attacks
bandwidth consumption, 346–347
buffer overflow attacks, finding, 138–139
network disclosures
HTTP Referer, 114
monitoring network data, 113–114
obfuscation of data, 117–118
questions checklist, 113
Web beacon, 114–115
Network group, 313
Network Service account, 326
network shares canonicalization issues, 288–289
network traffic
finding all, 71
formatting of, 71
news sources on security flaws, 8–9
Nimba worm, 122
NTFS file system
case insensitivity of, 286
data stream canonicalization issues, 284
NTSD debugger, 160
NULL DACLs, 317
null termination failures, 171, 172, 176
null values allowing buffer overflows, 148–149
Numeric Character References (NCRs), 266–267
numeric overflows. See integer overflow

O
obfuscation of data, 117–118
obfuscation of programs, 381, 415–416, 427
Object Browser, 453
object tags for creating ActiveX controls, 438–439
objects
nested, 461–463
owners of, access rights, 318–319
race condition attacks, 320–322
objects, continued
remote accessible, 325–327
squatting attacks, 320
types of, returning at run time, 463
ObjSD.exe tool
table of discoverable permissions, 307
viewing permissions with, 311–312
observation of programs
advantages of, 412
error messages as indicators, 412
file format discovery, 412
Logger tool, 414
monitoring tools overview, 413
output, comparing based on input, 412, 413, 435
output reused as input, 412–413
purpose of, 411–412
testing tips, 435
OS (Organization for Internet Safety), 502
OLE DocFiles, 416
OLE View tool, 453–455
OleView tool, 41
OllyDbg tool
breakpoints, setting on memory, 421–424
code tracing steps for patches, 416–417
modifying the binary example, 419
purpose of, 416
tracing backward to an event, 417–419
one-click attacks. See CSRF (cross-site request forgery) attacks
ORDER BY clauses for SQL injection attacks, 394
outerHTML property, 246
Outlook Express S/MIME buffer overflow attack, 45
Outlook View Control bug, 461–463
overlong UTF-8 encoding, 291–292
overwriting memory with format specifiers. See format string attacks

P
packers, 427
pagefile.sys, 107
PARAMs, 440, 468, 476–477, 484–486
Paros tool, 55–56
parsers, XML
data streams, 263–264
infinite loop detection, 268–269
testing, 264
well-formed input requirement, 263
well-formed XML, rules for, 264
parsing
buffer overflows, as source of, 170
Internet Explorer, internal operations of, 253–254
non-HTML files parsed as HTML, 248–250
XML. See parsers, XML
partially trusted code, 359. See also APTCA
passwords
authentication code reverse engineering, 432–434
controls, Web form, 61
environment variables, storing in, 78
hash comparisons, 432–434
same for multiple applications, danger from, 103
URLs including, 298
viewing using debuggers, 420
patches, security
disassembling to find bugs, 434–435
quickly installing, importance of, 504
releasing simultaneously, 506–507
patching binaries without source code, 416–420, 422
paths
expansion bugs, 171
predictability of names for, 108–109
search paths, 289–290
separator-induced overflows, 148
variations on representing, 280–281
PDF files. See Adobe Acrobat
Peach fuzz tester, 69
penetration testers, 2–3
Performance Monitor, 337–339
performance testing, 1
permissions
AccessEnum tool, 307, 309–310
ACEs. See ACEs
ACLs. See ACLs (Access Control Lists)
AppVerifier tool, 312
assert affects on. See asserts
best practices, 302
Bypass Traverse Checking right, 43–44
code groups, 353–354, 357–359
container access control issues, 315–316
DACLs. See DACLs (Discretionary ACLs)
DCOM objects, 327
DELETE permissions, 313
demands, 360–362
deny ACEs, 317
directory ACLs, 315–316
elevation of privilege danger, 302
Everyone group, 312–313
exception filtering attacks, 319–320
file vulnerabilities, 105–106, 108
FILE_ADD_FILE permissions, 313
FILE_DELETE_CHILD permissions, 313
FileOpPermission, 355–356
FullTrust, 359
guidelines, 312
identifying objects and requirements, 303
impersonation, 325
importance of setting correctly, 301–302
indirect access to resources, 319
information disclosure issues, 108
large groups, table of, 313–314
linked file issues, 322–325
locally accessible objects, 327–328
multiple-stage elevation of privilege, 302
named pipes, checking for, 36
.NET permissions, 328, 355–356
NULL DACLs, 317
ObjSD.exe tool, 307, 311–312
ordering of ACEs in DACLs, 318
partially trusted code, 359
process ACLs, 315–316
Process Explorer tool, 307, 309–311
purpose of, 301
race condition attacks, 320–322
Regedit.exe, 43–44
registries, 42, 43–44
registry ACLs, 315–316
remotely accessible objects, 325–327
requests, testing with lower, 72
reversing issues, 319–320
role-based security, 331–332
SACLs, 305
sandboxing, 359–360
securable objects, 304–305
security descriptors, 305
SQL permissions, 329–331
squatting attacks, 320
stack walks, 360–365
summary, 332
symbolic links, 322–323
temporary file issues, 321
testing steps, 303
testing with user accounts, 307
tools for finding for objects, 307
viewing with Permissions dialog box, 307–309
Windows services, 325–327
WRITE_DAC permissions, 313
WRITE_OWNER permissions, 313
PermitOnly security action, 356, 364–365
persistent cross-site scripting attacks
data fields used in, table of, 230–231
data in script variable fields, 233–234
defined, 228
encoding prevention method, 232–233
event vulnerabilities, 234–235
exploiting vulnerabilities, 230
guestbook example, 228–229
identifying vulnerabilities, 238
My Computer zone, in, 246–251
non-HTML files parsed as HTML, 248–250
playlists, vulnerability of, 246–248
POST method for, 230
query strings vulnerability, 230
scr property of IMG tag vulnerability, 235–236
script tag issues, 232
style vulnerabilities, 235
User-Agent header vulnerability, 231
user interface for testing, issues with, 231
Winamp vulnerability, 246–248
Windows Media Player bug, 249–250
phishing attacks
defined, 97
reflected XSS attacks using, 225
spoofing for, 97
PHP
encoding functions, common, 261
functions raising XSS issues, 259, 260
GET XSS bug example, 260–261
PHPNuke SQL injection bug, 388
Prinvoke, 367
Pizza.exe buffer overflow example, 172–176
platform invoke, 367
playlists, vulnerability of, 246–248
PlaySMS cookie bug, 68
plugable protocol handlers
CMS files, exploiting, 244
defined, 38
entry points from, 38–39
finding entry points, 39
image loads from links, 38
importance to attackers, 38
irc/mIRC, 38
Viewplgs.exe tool, 39
pointers, failure to reset bugs, 171
points of failure, 23
policies, 356–357
pop-up windows, Z-order spoofing of, 96–97
ports
port numbers, relationship to sockets, 29
socket hijacking scenario, 73–76
POST method
applications using, 63
CSRF (cross-site request forgery) attacks with, 493–494
headers used for injection attacks, 388
helloPostDemo.asp, 226–228
local XSS vulnerability, 238–239
MiddleMan HTTP proxy, modification with, 64–65
purpose of, 61
tampering with, 63
XSS attacks using, 226–228, 230
PPTP (Point-to-Point Tunneling Protocol), 81
pre-creation attacks, 320
Prefast tool, 167
printf()
buffer contents, writing, 186–187
buffer references as parameters, 186–189
disassembling to evaluate, 429–430
first parameter of, attack basis, 188
format string attacks using, 186
overwriting memory with, 190–191
550 privileges

print(), continued
popcorn stack values down to a target, 190
stack, effects of parameters, 187
unanticipated string specifiers, 188–190 stack
interpretation at run time, 188–190
variants of, 192–193
privileges. See also access issues; permissions
best practice for granting, 302
elevation. See elevation of privilege (EoP)
Process Explorer tool
command-line argument entry points, 47–48
environment variables, finding, 49
file use, finding with, 105–106
table of discoverable permissions, 307
user interface entry points, finding, 44
viewing permissions with, 309–311
process memory, 128–129
processes, ACLs for, 315–316
ProgIDs, 440
programmable interface vulnerabilities, 140
programmatic interfaces
AXDetail tool, 41
common, list of, 40
dcomcnfg.exe tool, 41
defined, 40
finding entry points, 40–41
gcutil.exe tool, 41
importance to attackers, 40
oleview tool, 41
packet structure, 40
rcpdump tool, 41
tools, 40–41
proof-of-concept exploit code, 503–504
protectors, 427
Protocol handlers
CHM files, exploiting, 244
command-line arguments, inserting in, 47
ftp forced connections, 74
plugable. See plugable protocol handlers
telnet vulnerability, 78
proxy requests, HTTP, malformed
Accept-Language header, 68
advantage of, 64
cookies, 67
ease of proxy use, 63–64
middleman HTTP proxy, 64
POST method data modification, 64–65
Referer header, 68
User-Agent header, 68
proxy requests, TCP, malformed
data validation issues, 57–58
FrontPageServer Extensions bug, 57
Interactive TCP Relay, 55–56
Man in the Middle tool, 55–56
MiddleMan Web Proxy, 55–56
MITM proxy tool, 56
Paros tool, 55–56
purpose of, 53–56
server address specification, 58
TCP request modification, 56–58
telnet server testing, 56–57
terminal emulation value modification, 57
public disclosure of security bugs, 503–504
Punycode, 98
Q
queries, SQL
buffer overflows from, 147
injection attacks using. See SQL injection attacks
query strings
& (ampersands) in, 291
CSRF attacks using, 492–493
GET method for creating, 60–61
sources of, 62
tampering with, 62
URL redirection attacks, 98
variables, altering, 62
XSS vulnerability, 230
quotation marks
buffer overflows when missing, 148
SQL injection attack issues, 386, 391, 397, 403–404

R
race condition attacks
access rights issues, 320–322
fetchmail, 108
heap allocations followed by frees, 153
information disclosure threats from, 106
read AVs, 153
RealPlayer
forced connections with, 74
XSS vulnerability, 237
recursive calls, 343–344
recv function, 81
redirection attacks, URL, 98–99
references, buffer overflows from invalid, 146–147
Referer headers
defined, 68
spoofing, 90
vulnerability of, 62
XSS vulnerability of, 231
reflected cross-site scripting attacks
Action properties for, 227
actions enabled by, list of, 223–224
ADO DB connection vulnerability, 241
binaries, running using local files, 240–241
CHM (Compiled Help Module) files, 243–244
cookies, accessing, 223
data fields used in, table of, 230–231
data in script variable fields, 233–234
defined, 220
echoing prevention method, 232–233
event vulnerabilities, 234–235
exploitation of, 225–226
goal of attackers, 225
hashes, setting variables to, 237
helloPostDemo.asp, 226–228
HTA file vulnerability, 241
identifying vulnerabilities, 258
Information bar warnings in IE, 239
local file attack effects, 238–239
local file bug exploitation, 237–238
local file example, 236–237
local file XSS URL example, 239
local files, understanding attacks against, 236
phishing method for, 225
POST method for, 226–228, 230
query string vulnerability, 230
res protocol vulnerability, 242
resources, XSS bugs in, 241–243
scr property of IMG tag vulnerability, 235–236
script tag issues, 232
scripts for, 225–226
search engine example, 220–223
Shell Application control vulnerability, 241
SiteLock vulnerability, 224
SSL not protection against, 226
style vulnerabilities, 235
Submit method, 227–228
user interface for testing, issues with, 231
UserData vulnerability, 224
vulnerability of, understanding, 223
zone elevation vulnerability, 224–225
Regedit.exe permissions, determining, 43–44
registered file types, entry points created by, 26–27
registrars, CPU
attacking with format specifiers. See format string attacks
buffer overflow effects on, 152–153
ECX register, 197–200
registration schemes, understanding, 421–424
registries
ACLs for, 315–316
ActiveX controls access to, 484
Bypass Traverse Checking right, 43–44
defined, 41–42
encryption recommended, 42
trace points from, 41–44
finding entry points, 43–44
importance to attackers, 42–43
permissions, 42, 43–44, 307–309
predefined keys, 42
RegMon tool, 43
Remote Registry service, 42
Service Pack 2 opt-in settings, 250
RegMon tool, 43
Remote Interactive Logon group, 314
remote procedure calls. See RPCs (remote procedure calls)
Remote Registry service, 42–43
remotely accessible objects
DCOM objects, 327
types of, 325
Windows services, 325–327
reporting security bugs
Bugtraq, 503
contact information for vendors, 501
Full Disclosure, 503
immediate public disclosure argument, 499–500
importance of, 499
information to provide to vendors, 500
internal reporting, 501–502
level of detail to provide to public, 503–504
non-reportable issues, 500
parties to notify, 499
proof-of-concept exploit code, 503–504
public disclosure of security bugs, 503–504
responsible disclosure process, 500
RFPolicy, 502
selling vulnerability information, 503
time allowed before public disclosure, 502, 504
unresponsive vendors, 502–503
vendor responses, 502
repudiation
as STRIDE category, 17
test cases for e-mail Web pages, 20
repurposing attacks
ActiveX. See ActiveX controls
COM. See COM (Component Object Model)
defined, 437
document formats for. See document format repurposing
attacks
message repurposing, 496–497
shatter attacks, 497
summary, 497
requests. See also server request processing
client/server interaction, 52
custom clients for malformed requests, 55
data validation, 57–58
documentation, examining, 52
finding normally acceptable requests, 52
FrontPageServer Extensions bug, 57
GET. See GET requests
hooking programs, 55
HTTP. See HTTP requests
manipulating, 54–58
Microsoft Detours, 55
monitoring network traffic, 52–54
out-of-order tip, 71
permissions, testing with lower, 72
proxy requests, creating malformed, 55–58
requests, continued
  removing fields for testing, 72
server address specification, 58
single, constructing, 54–55, 71
sniffing traffic, 53–54
TCP, proxy modification of, 56–58
terminal emulation value modification, 57
testing tips, 71–72
validation, client-side, 72
wget tool for custom requests, 54–55
WSASend function, 55
res protocol, 242
resource consumption vulnerability
  CPU resources, attacks against, 341–344
defined, 334
leaks from DoS attacks, 336–339
nature of, 340
types of vulnerable resources, 340
resources, HTML
  LoadResource API, 242
tools for examining, 242–243
XSS bugs in, 241–243
responses, server. See server responses
responsible disclosure process, 500
Restore Files And Directories privilege, 318
Restricted Sites zone, 240
Resultant Set of Policy data, 315
return values in ActiveX controls, 460–461, 468–469
reverse engineering
  ActiveX controls, of, 452
  algorithms, 431–434
  authentication code example, 432–434
  code tracing strategies, 430
decomposers, 424–425
defined, 424–425
determining whether attackers can control data, 430–431
disassemblers, 424–425
finding dangerous functions, overview of, 427
format string vulnerabilities, 428–431
legal considerations, 436
native vs. bytecode, 425–427
obfuscation of programs, 427
Open Reverse Code Engineering Web site, 431
packers, 427
protectors, 427
results of a sample decompile, 425–426
security patches, analyzing, 434–435
summary, 436
symbol files, 426, 435
testing tips, 435–436
reverting permissions, 319–320
RevertToSelf function, 139
RFPolicy, 502
role-based security, 331–332
root cause guideline, 435, 505
RPCs (remote procedure calls)
  buffer overflows with, 139
  entry point potential of, 40
  RpcDump tool, 41
RSS (Really Simple Syndication)
  buffer overflow attacks, 267
defined, 267
directory traversal attacks, 267
enclosures, 267
format string attacks, 267–268
HTML scripting attacks, 267
user interface spoofing, 267
runtime tools
  BoundsChecker, 163
  Gflags tool, 163–165
S
S/MIME (Secure/Multipurpose Internet Mail Extensions), 45
SACLs (System ACLs), 305
sandboxing, 359–360, 382
scanf(), 186
Scapy network disclosure monitoring, 114
schema, SQL database, 406–407
schema, XML, 264–265
script injection attacks. See persistent cross-site scripting
attacks
scripts
  elements to check for XSS vulnerability, 245–246
  finding XSS bugs in client-side, 244–246
  hashes, setting variables to, 237
  inside other scripts, 253–254
  Internet Explorer, 449
  nested objects, 461–463
  playlist XSS vulnerability, 246–248
  reflected XSS attack scripts, 225–226
  resources, HTML for running, 243
search engines
  malicious sites indexed by, 225
  Web site back doors for, 91
  XSS attack example, 220–223
search path canonicalization issues, 289–290
securable objects
  DACls of, 305–306
defined, 304–305
  SACLs of, 305
security conferences, 9
security descriptors
  DACls, 305–306
defined, 305
  NULL DACls in, 317
  ObjSD.exe tool for viewing, 311–312
SACLs, 305
types of ACLs contained in, 305
security hobbyists, 3
security identifiers. See SIDs
security mailing lists, 9
security patches. See patches, security
security testers
expectations for, 4
job of, 2
malicious thought guidance for, 6–8
as part of general tester role, 2
penetration testers as, 2
security testing
alternative approaches to, 4
definition of, 1
general approach to, 3–4
malicious thought guidance, 6–8
purpose of, 1–2
security functionality testing compared to, 2–3
taking applications apart, 5–6
understanding target applications, 4
semicolons for SQL injection attacks, 391, 404–405
sender repudiation, 17
serial codes for registering software, 421–424
serv2, 159
server redirection with ActiveX controls, 465
server request processing
constructing single requests, 54–55
cookies, issuance of, 67
custom clients for malformed requests, 55
documentation, examining, 52
finding normally acceptable requests, 52
malicious requests, generating, 70–72
manipulating requests, 54–58
monitoring network traffic, 52–54
out-of-order tip, 71
overview of, 51–52
permissions, testing requests with lower, 72
proxy requests, creating malformed, 55–58
server address specification, 38
sniffing traffic, 53–54
source code, examining, 52
validation, client-side, 72
server responses
arbitrary server connections, 74
bugs found with malicious responses, 77
common vulnerabilities list, 77
debugging with malicious, 81
DNS poisoning scenario, 74–75
domain/zone elevation, 81
downgrade MITM attacks, 80–81
ease of creating malicious responses, 80
entry points from, 39–40
EvilServer tool, 80
file caching vulnerability, 79
fuzzing recommended, 82
Half-Life example, 40
importance of understanding, 73
malicious connections, causes of, 73
malicious, overview of, 73
man-in-the-middle connections, 74
myth re difficulty of creation, 80
socket hijacking scenario, 75–76
SSL to prevent maliciousness, 76, 82
telnet client environment variable attack, 78–79
testing tips, 81–82
URL replacement tests, 81
server timeouts, DoS attacks with, 340
Service group, 314
Service Pack 2, 240, 250–251
services, Windows. See Windows services
SFI (safe for initialization), 440, 450–451
SharePoint Services e-mail icon headers, 46
shatter attacks, 497
Shell Application control vulnerability, 241
SHGetFileInfo function, 98
shopping cart integer overflow example, 130–131
short filenames, 285–286
shrmul-wrap exploit, 8
SIDs (security identifiers)
defined, 306
granting correctly, example of difficulties, 315
Interactive, 314
signatures, XML issues, 185
SiteLock
ActiveX with, 444
cross-site scripting (XSS) attacks against, 224
size field mismatches with file size, 413
SMTP spoofing, 89–90
sniffing traffic
best practices for, 54
decoders, 54
name resolution, 54
network cards, disabling extra, 54
network traffic formatting, understanding, 71
promiscuous mode, 54
request monitoring, 52–54
SOAP (Simple Object Access Protocol)
buffer overflows, 148
CSRF (cross-site request forgery) attacks with, 495, 496
GET support for, 495
POST support for, 495
security dialog box for requests outside domain, 495
social engineering attacks
Motorola attack, 84
spooxing, as aspect of, 84
user interfaces for, 44
sockets
attack types based on, 29
entry points, as, 29–31
enumerating open sockets, 30–31
finding entry points created by, 30
importance to attackers, 29–30
man-in-the-middle attacks, 29–30
sockets, continued
  monitoring network traffic, 31
  Netstat exe for finding, 30–31
  port numbers of, 29
  purpose of, 29
  server hijacking scenario, 75–76
  soft links, 322–323
  software returns malicious thinking example, 8
  source file information disclosure issues, 109–110
  SP2 See Service Pack 2
  spamming, 17, 19
  specification variance from implementation, 21
  SPIKE fuzz tester, 69
  spoofing
    attackers, thinking like, 85
    authentication values, 87
    binary editors for, 93, 100
    Caller ID spoofing, 83–84
    control characters for, 93–96
    data presented to users, identifying, 85
    defined, 83
    dialog box rewording, 91–93
    e-mail, 89–90
    finding vulnerabilities, approach for, 85
    FormMail script, 90
    homograph attack URL spoofing, 97–98
    HTTP Referer headers, 90
    IP address spoofing, 86–87
    ISNs, 86
    items commonly spoofed, list of, 85
    links, dialog boxes with, 91–93
    log files, using control characters, 94–95
    MAC addresses, 87–88
    phishing attacks, using for, 97
    Referer headers, 90
    reformattting with control characters, 93–95
    reverse DNS lookups, 88–89
    SMTP, 89–90
    social engineering aspect of, 84
    as STRIDE category, 17
    TCP IP addresses, 86–87
    testing tips for, 100–101
    trusting basis of, 83
    UDP packets, 87
    URL homograph attacks, 97–98
    URL redirection attacks, 98–99
    User-Agent header spoofing, 91
    user interfaces, overview, 91
    username@ syntax, 93, 100
    voice mail, 84–85
    wildcard DNS, 93, 95–96
    Z-order spoofing, 96–97
  spreadsheet format repurposing attack vulnerability, 489–490
  sprintf(), 186
  Spy++, 44
  SQL injection attacks
    ampersand (&) delimiters, 408
    assessing vulnerability, 388
    attacker goals, 385, 386
    back-end server access, 387
    backslashes (\) for escapes, 404
    backup stored procedure example, 407–408
    batched transaction vulnerability, 399–400
    black box testing overview, 389
    brackets for, 396
    breakout techniques overview, 390–391
    Bugtraq, list of exploits, 388
    classes, table of, 401
    commands, list of dangerous, 402
    commening input, 393–394
    common mistakes, list of, 403
    data truncation for, 398–399
    document format repurposing attacks for, 489
    double quotation marks for, 397, 410
    elevation of privilege, 387
    EXEC command, 397–398, 408
    executed statements, determining, 389–390, 401–402
    firewalls for protection from, 406
    functions, built-in, 395–396
    hyphens for, 393–394
    importance of, 387
    information disclosure, 387
    known exploits, 388
    LIKE clause for, 394–395
    line breaks for breakouts, 394
    logging to SQL, vulnerability from, 388
    managed code vulnerability, 352
    mechanisms of injection, 386
    mechanics of queries, 385–386
    nested queries, 389
    NULL characters for breakouts, 394
    number field breakout techniques, 392–393, 403–404
    OR statements in queries, 392
    ORDER BY clauses for, 394
    parameterized queries, 402–403
    password attacks, 392
    permissions issues, 391–392
    PHPNuke vulnerability, 388
    PlaySMS cookie bug, 68
    POST headers as part of, 388
    QUOTENAME phrases, 398–399
    removing unwanted stored procedures, 405–406
    REPLACE phrases, 398–399
    repurposing of stored procedures, 407–409
    Request.QueryString vulnerability, 386
    restricting user permissions, 329
    sanitizing user input, 402–403
    schema determination, 406–407
    search terms table, 401
semicolon separators, 391, 404–405
server commands, ability to run, 387
single quotation mark breakouts, 386, 391, 403–404
sort specification vulnerability, 394, 410
square brackets for, 396
stored procedures for, 397–399, 405, 407–409
string field breakout techniques, 391–392
summary, 410
tampering with data, 387
technologies and search terms table, 401
testing tips, 409–410
tracing events, 389–390
user input, reviewing entry points for, 402–403
user-supplied data, locating, 390
USER_NAME SQL function, 396
wild card characters, 394–395
SQL scripting attacks
code reviews, 400–403
searching for points of statement construction, 401–402
white box testing, 400–403
SQL Server, Microsoft
account types for permissions, 329
global temporary stored procedures, 331
global views, 331
named pipes vulnerability, 35
permissions overview, 329
Profiler tool, 389–390
restricting user permissions, 329
security functions, 330
SQL Slammer attack, 122
triggers, 330
SQL Slammer attack, 122
SQL (structured query language)
entry points created by, 41
modification of data, trying to disallow, 491
square brackets for SQL injection, 396
squatting attacks, 320
src attribute, 235–236, 246
SSH (Secure Shell) downgrade MITM attacks, 80–81
SSL (Secure Sockets Layer)
arbitrary server attacks, 82
attacks using certificates, 82
certificates, importance of checking, 82
certificates obtained by attackers, 76
downgrade attacks, 81
malicious server response mitigation, 76
URL canonicalization issues, 295–296
vulnerability of, 62
XSS attacks, not protection against, 226
stack overflows
architecture of, 175–176
defined, 124
EBPs (stack frame pointers), 129, 172
EIPs (extended instruction pointers), 129, 172
ESP(s) (extended stack pointers), 125
evidence of, 153
exception handler overwrites, 129
/G5 compiler switch, 179–182
input to buffers, 126
malicious code, running, 129
memory address space, 128–129
overwriting of return addresses, 126–128, 129
Pizza.exe example, 172–176
pointers, 125
popping variables off stacks, 126
process memory, 128–129
pushing data onto stack, 125–126
return addresses, 125–129
serv2 example, 161–162
stack operation, 125
stack walks
asserts, 363, 364, 368–370, 382
defined, 360
deny, 363–364
full demands, 360–361
inheritance demands, 362
link demands, 361–362, 370–372
modifiers, 362–365
PermitOnly security action, 364–365
stacks
format string attack mechanics, 187–190
function parameters pushed onto, 423
overflows. See stack overflows
overwriting memory with format specifiers in, 190–191
return addresses, finding for attacks, 200–201
stateless protocols, 59
stored procedures
backup example, 407–408
dangerous functions, identifying, 409
data truncation for injection attacks, 398–399
global temporary, 331
injection attacks with, 397–398, 405, 407–409
local temporary, 331
nested, 389
removing unwanted, 405–406
repurposing of, 407–409
user input, identifying, 408
strcpy function, 414
STRIDE categories
denial of service, 17
elevation of privilege, 17
information disclosure, 17
repudiation, 17
spoofing, 17
tampering with data, 17
strings
null termination failures, 171
Strings tool by SysInternals, 111–112
vulnerabilities of. See canonicalization issues
strong names, 354–355
styles, 235, 254–255
Submit method, XSS attacks with, 227–228
Super Password Spy++, 247
symbol files, 426, 435
symbolic links, 322–323
SYN request role in handshakes, 86
SysInternals
File Monitor, 27–28, 106–107
Process Explorer for finding files used, 105–106
Strings tool, 111–112

T
Take Ownership privilege, 318
tampering with data STRIDE category, 17
TCP requests
FrontPageServer Extensions bug, 57
local port configuration, 56
MITM proxy tool, 56
proxy modification of, 56–58
remote server configuration, 56
server address specification, 58
telnet server testing, 56–57
terminal emulation value modification, 57
TCP (Transmission Control Protocol)
ACK responses, 86
operating system mitigation of spoofing, 87
sequence numbers of packets, 86
spoofing techniques, 86–87
SYN requests, 86
telnet
three-way handshakes, 86
client environment variable attack, 78–79
requests, malformed using proxies, 56–57
tempest technique, 117
temporary file storage, 108
Ten Immutable Laws of Security, 500
test plans
code execution threats test cases, 20
denial of service test cases, 20
erroneous dismissal of threats, 18
mail bombing test cases, 19
repudiation test cases, 20
spamming test cases, 19
threat models, using, 18–20
testers, security. See security testers
testing
finding all network traffic, 71
malicious requests to servers, tips for, 71–72 fuzz. See fuzz testing
requests, generating malicious, 70–72
security. See security testing
types of, 1
textboxes, functions for retrieving text from, 422
thinking maliciously, 6–8
threat models
access level identification, 15
analysis compared to, 474
bug tracking systems with, 18
denial of service test cases, 20
designer assumptions, 18
denial of service test cases, 20
DFDs. See DFDs (Data Flow Diagrams)
enumeration of entry and exit points, 14–15
erroreous dismissal of threats, 18
evolution of, 11–12
implementation issues, 21
information disclosure bugs, finding, 103
key parts of, 12
mail bombing test cases, 19
prioritization of features, 15
purpose of, 11
repudiation test cases, 20
spamming test cases, 19
summary, 22
team for creating, 12
tester use of, 18–20
time required for creation, 12
tips for threat identification, 16–18
updating on design changes, 21
verification, importance of, 18
threats, enumeration of. See enumeration of threats
time of Check Time of Use attacks, 321
TOCTOU attacks, 321
trailing characters to file extensions, 283
transmissions, injection attacks using, 399–400
Trident
applications hosting, partial list of, 258
evading script disabling in IE, 258
non-HTML file parsing by IE, 248
non-IE use of, 250
SP2, opting in, 251
Winamp vulnerability, 246–248
Trusted Sites zone, 240

U
%u (UCS-2 encoding), 293
UAC (User Account Control), 302
UCS-2 encoding, 293
UDP
spoofing, 87
SQL Slammer attack, 122
UI spoofing. See user interface spoofing
UNC shares, canonicalization issues, 288–289
understanding target applications, 4
Unicode
ANSI expansion bugs, 170
buffer overflows caused by, 176
file manipulation functions, paths for, 288–289
format string attacks with, 192
functions, string termination for, 192
UNIX
  - case sensitivity of, 286
  - device name vulnerabilities, 287
  - Morris Worm, 122
  - socket hijacking scenario, 75
  - symbolic links, 322–323
unmanaged code
  - PInvoke, 367
  - user security model for, 353
  - Win32 APIs, 367
unregistered file types, 27
unsafe code
  - declaration, effect of, 366
  - marking code as, 366
  - marshaling data for, 367–368
  - PInvoke, 367
  - pointers, 367
  - unmanaged Win32 APIs, 367
  /unsafe compiler option, 366
unsafe keyword, C#, 350
unverifiable code, 350
updates, certificate verification strategy, 76
upgrade testing, 1
URLs
  - & (ampersands) in, 291
  - Action properties for XSS attacks, 227
  - backslash replacements, 291
  - buffer overflows from encoding, 171
  - canonicalization overview, 290
  - canonicalization vulnerability example, 280
  - credential handling in, 298
  - CSRF attacks using query strings, 492–493
  - domain name parsing issues, 296–297
  - dot checks, 296
  - dotless IP addresses, 296–297
  - double encoded characters, 293–294
  - evidence, as, 355
  - examples of variations in encoding, 290
  - form data, insecure, 60–61
  - fully qualified domain names, 468
  - GET method, 60–61
  - hexadecimal escape codes in, 290–291
  - hexadecimal IP addresses, 297
  - homograph attack spoofing, 97–98
  - HTTP Referer disclosure of, 114
  - improper handling issues, 295
  - IPv6 canonicalization vulnerabilities, 297
  - JavaScript IMG tag vulnerability, 235–236
  - local file XSS example, 239
  - maximum length of, 61
  - protocol handlers. See pluggable protocol handlers
  - Punycode, displaying in, 98
  - query strings in. See query strings
question marks in, 62
  - redirection attacks, 98–99
  - replacement tests for server responses, 81
  - SHGetFileInfo function, 98
  - spoofing Referer headers, 90
  - SSL canonicalization issues, 295–296
  - User Account Control (UAC), 302
  - User-Agent header
    - spoofing, 91
  - vulnerability of, 68
  - XSS vulnerability of, 231
user interface spoofing
  - binary editors for, 93, 100
  - defined, 83
  - dialog box rewording, 91–93
  - homograph attack spoofing, 97–98
  - links, dialog boxes with, 91–93
  - log files, using control characters, 94–95
  - overview, 91
  - phishing attacks, using for, 97
  - preventative measures, 96
  - reformatting with control characters, 93–95
  - RSS attacks with, 267
  - testing tips for, 100–101
  - URL redirection attacks, 98–99
  - username@ syntax, 93, 100
  - wildcard DNS, 93, 95–96
  - Z-order spoofing, 96–97
user interfaces
  - defined, 44
  - entry points, providing, 44
  - importance to attackers, 44
  - Process Explorer tool, 44
  - social engineering attacks, 44
  - spoofing. See user interface spoofing
  - Spy++, 44
  - vulnerability exploitation, 44
  - Winspector tool for finding, 44
user names
  - common, vulnerability from, 119
  - information disclosure bugs giving, 103–104
  - URLs including, 298
User policy level, 356–357
USER_NAME SQL function, 396
username@URL syntax attacks, 100
users, data storage specific to, 107–108
Users group, 314
UTF-8 encoding, 291–292

V
  - ValidateRequest feature of ASP.NET, 255–256
  - van Eck phreaking, 117
VBScript

VBScript
ActiveX control creation, 439
XSS attack vulnerability, 236
vendors
attitudes towards bug reporting, 499
contact information for reporting bugs, 501
information to provide to for bug discoveries, 500
non-reportable issues, 500
responses to security bug reporting, 502
responsible disclosure process, 500
unresponsive to bug reports, 502–503
version information, disclosure of, 115
ViewpIgs.exe tool, 39
voice mail spoofing, 84–85
Voice over IP spoofing, 83–84

W
weak data encoding, 117–118
Web beacons, 2, 114–115
Web browsers
DOM vulnerability, 223–224
functions returning data to, table of, 259
HTTP traffic from. See HTTP (Hypertext Transfer Protocol)
input sources, list of, 59
Internet Explorer. See Internet Explorer
parsers, internal operations of, 253–254
proxy port settings, 64
query strings, tampering with, 62
UserData vulnerability, 224
vulnerability overview, 59
Web controls, ASP.NET, 350–351. See also controls
Web e-mail, copying page to feature. See e-mail Web pages
Web page repurposing attacks
attacker goals, 492
CSRFs. See CSRF (cross-site request forgery) attacks
dependant data access, 492
server security responsibility, 492
Web proxy cache poisoning, 74–75
Web Service Security, 185
Web Text Converter, 295
Weex format string vulnerability, 193
Welch tool, 54–55, 70–71
white box testing
automated code review, 166–167
buffer overflows, finding with, 166–167
dangerous function searches, 166
defined, 20
following input method, 166
LCLint tool, 167
manual linear reviews, 166
Prefast tool, 167
whitelisting characters, 251–253
Whoami.exe, 315
WiFi faked access points, 74
wildcard DNS spoofing, 95–96
Winamp vulnerability, 246–248
Windows Explorer file properties, displaying, 111
Windows Media Player
ActiveX controls, displaying in IE, 438–439
forced connections with, 74
XSS bug, 249–250
Windows Server issues, 76
Windows services
account types, 326
guidelines for security, 325–327
Local Service account, 326
Local System account, 326
Network Service account, 326
rights granted when accessing, 326–327
windows Z-order spoofing, 96–97
WinExec:
address insertion for format string attacks, 212
creating and compiling calls, 211
eing point offset for, 210–211
putting parameters onto stacks, 214–215
WinHex tool, 111–113, 421
WINS tool, 44, 497
WMD files, 249
Word, Microsoft, metadata in files, 111
Write access, executables in directories with, 316
write AVs, 153
WRITE_DAC permissions, 313
WRITE_OWNER permissions, 313, 318
WSASend function, 55

X
xbreaky bug, 324–325
XML (Extensible Markup Language)
attributes, syntax for, 264
bombs, 269
buffer overflows from, 147
CDATA, 265–267
character entity references, 266–267
elements, syntax for, 264
ties, 268–270
external entities, 270
HTML scripting attacks on RSS, 267
infinitesimal reference loops, 268–269
input parsing issues. See XML input files
non-XML bugs in input, 267–268
numeric character references, 266–267
RSS attacks, 267–268
schema for, 264–265
scripting attacks, 258
signatures, 185
SOAP. See SOAP (Simple Object Access Protocol)
Web Service Security, 185
well-formed, rules for, 264
XML bombs, 269
XML injection attacks, types of, 270
XML input files
CDATA, 265–267
character references, 266–267
complex XML input, 269
data streams, 263–264
external entities, 270
image tags inside CDATA, 265–266
non-alphanumeric data in, 265–267
overview of non-XML security issues, 263–264
parsers, 263–264
validation with schema, 264–265
vulnerability from data streams, 264
well-formed input requirement, 263
well-formed XML, rules for, 264
XML bombs, 269
XmlReader
vulnerability of, 264
well-formed input issues, 263–264
XPath, injection attacks, 409
XSD (XML Schema Definition), 264–265
XSS attacks. See cross-site scripting (XSS) attacks

Z
Z-order spoofing, 96–97
zero-click attacks. See CSRF (cross-site request forgery) attacks
ZeroMemory function, 426–427
ZIP file information disclosure issues, 112–113
ZLIB buffer overflow bug, 123
zone evidence, 355
zones, security
Internet Explorer types described, 240
Internet zone links to My Computer zone, 256–257
local HTML file bug zone, 238
My Computer zone treatment, 238
persistent XSS attacks in My Computer zone, 246–251
script disabled default, 257–258
SP2 zone elevation blocks, 251
XSS attack vulnerability, 224–225