INDEX

A
abstract classes
abstract Task class, 160–161
defined, 4
subclasing from, 5–6
abstract syntax tree (AST), 243
anonymous methods
assigning delegate to method, 9
optional arguments, 10–11
updating Firefighter class, 9–10
updating Main() method, 11–12
API (application program interface)
Arachni REST API, 224–228
Cuckoo Sandbox, 148–150
Nessus, 103–105
Nexpose
NexposeManager class, 124–125
NexposeSession class, 118–124
RPC API, 208–209
sqlmap REST API, 169–173
Arachni, 223
arachni_rpcd script, 229
arachni_rpc script, 229
installing, 223–224
Main() method, 237–239
REST API, 224–228
ArachniHTTPManager class, 226–228
ArachniHTTPSession class, 225–226, 228
RPC, 228–237
ArachniRPCManager class, 236–237
ArachniRPCSession class, 230–234
ExecuteCommand() method,
234–235
ArachniHTTPManager class, 226–228
ArachniHTTPSession class, 225–226, 228
ArachniRPCManager class, 236–237
ArachniRPCSession class, 230–234
assets (Nexpose), 118, 126–127
AST (abstract syntax tree), 243
attributes, defined, 13
Authenticate() method
MetasploitSession class, 213
NessusSession class, 105–106
NexposeSession class, 119–120
authentication
Metasploit RPC API, 208, 213–214
NessusSession class, 105–109
NexposeManager class, 124–125
NexposeSession class, 118–120
OpenVASSession class, 135–136

B
BadStore ISO
booting VM from, 17–18
fuzzing POST requests, 25–31
parameters, 29–31
writing requests, 27–29
sqlmap utility and, 182, 184–185
binding payloads, 85–86
accepting data, 86
executing commands from stream,
87–88
returning output, 87
running commands, 87
bitmasks, 194
bkhive tool, 263–264
blind SQL injection, 43–44
creating true/false responses, 44
GetValue() method, 49–50
MakeRequest() method, 47
printing values, 50–51
retrieving lengths of values, 47–49
Boolean-based blind SQL injection. See
blind SQL injection
boot key, dumping
GetBootKey() method, 259–261,
262–263
GetNodeKey() method, 261–262
GetValueKey() method, 261
StringToByteArray() method, 262
verifying boot key, 263–264
Burp Suite, 25–27
C

C# language
anonymous methods, 9–12
assigning delegate to method, 9
optional arguments, 10–11
updating Firefighter class, 9–10
updating Main() method, 11–12
classes, 4, 6–7
interfaces, 4–7
Main() method, 7–9
native libraries, 12–13
types and syntax, 2–3
child nodes
registry hives, 250, 254–257
SOAP, 58–67
CIL (Common Intermediate Language) bytecode, 245
ci_scanfile() function (ClamEngine class), 198–200
ClamAV, 191
cl_daemon, 201–206
ClamManager class, 204–205
ClamSession class, 203–204
installing, 202
starting, 202–203
testing, 205–206
installing, 192–193
native library, 193–201
accessing functions, 196–200
ClamEngine class, 197–198
classes, 195
Dispose() method, 198–200
enumerations, 194–195
scanning files, 198–200
testing, 200–201
ClamBindings class, 196
ClamDatabaseOptions enum, 194
ci_daemon, 201–202
ClamManager class, 204–205
ClamSession class, 203–204
installing, 202
starting, 202–203
testing, 205–206
ClamManager class (ci_daemon), 204–205
ClamSession class (ci_daemon), 203–204
ClamEngine class, 197–198
ClamReturnCode enum, 195
ClamScanOptions enum, 195
classes, 6–7
abstract, 4, 5–6, 160–161
ClamAV native library, 195
defined, 4
Common Intermediate Language (CIL) bytecode, 245
CONCAT() SQL function, 39–40
connect-back payloads
network stream, 82–84
running, 84–85
running commands, 84–85
constructors, 6
CreateOrUpdateSite() method (NexposeManager class), 126–127
CreateSimpleTarget() method (OpenVASManager class), 141–142
CreateSimpleTask() method (OpenVASManager class), 143
CreateTask() method (CuckooManager class), 157–158
cross-site scripting (XSS), 20–22
CsharpVulnJson web application
capturing vulnerable JSON request, 31–33
JSON fuzzer
creating, 33–37
testing, 37–38
setting up vulnerable appliance, 31
CsharpVulnSoap web application, 54, 78–79. See also SOAP endpoints
Cuckoo Sandbox, 147
creating file analysis task, 163–164
CuckooManager class, 157–162
abstract Task class, 160–161
CreateTask() method, 157–158
reporting methods, 159–160
sorting and creating different class types, 161–162
task details, 159
CuckooSession class, 151–157
creating multipart HTTP Data with GetMultipartFormData() method, 153–155
FileParameter class, 155
testing, 156–157
writing ExecuteCommand() methods to handle HTTP requests, 151–153
manually running API, 148–150
setting up, 148
testing application, 164–165
curl command line tool
testing Arachni REST API, 225
testing Cuckoo status, 149–150
testing Nexpose API, 118
testing sqlmap API, 170–173

D

DateTime class, 3
decompilers, 242–245
DecompressData() method
   (ArachniRPCSession class), 233
delegates, assigning to methods, 9
DeleteSite() method (NexposeManager class), 128
DeleteTask() method (SqlmapManager class), 178–179
deserialization, 175
dispatchers (RPC framework), 250
Dispose() method
   ArachniRPCSession class, 234
   ClamAV native library, 198–200
   ClamEngine class, 200
   CuckooManager class, 160
   MetasploitSession class, 216
   NessusSession class, 107–108
   NexposeSession class, 123
   SqlmapManager class, 178
   SqlmapSession class, 174
dumping boot key
   GetBootKey() method, 259–261, 262–263
   GetNodeKey() method, 261–262
   GetValueKey() method, 261
   StringToByteArray() method, 262
   verifying boot key, 263–264

E

EICAR file, 200–201
endpoints
   SOAP. See SOAP endpoints
sqlmap API, 167
enumerations (ClamAV), 194–195
ExecuteCommand() methods
   ArachniRPCSession class, 234–235
   CuckooSession class, 151–153
   NexposeSession class, 120–123
   OpenVASSession class, 136–137
   ExecuteGet() method (SqlmapSession class), 174–175
   Execute() method
      ClamSession class, 203–204
      MetasploitSession class, 213–215
      MSGPACK library, 210
   ExecuteModule() method
      (MetasploitManager class), 219
   ExecutePost() method (SqlmapSession class), 175
   ExecuteRequest() method
      (ArachniHTTPSession class), 226
exploiting SQL injections
   Boolean-based blind SQL injection, 43–51
   UNION-based, 38–43

F

FileParameter class (CuckooSession class), 155
FileTask class (Cuckoo Sandbox), 161–162
First() method (connect-back payloads), 84
for loop
   child nodes and, 256–257
   methods and, 50–51
   retrieving length of database count of user database, 45–46
   sending payloads within, 47
functions
   ClamAV native library, 196–200
   declaring, 13
   importing from libc, 98–99
   SQL, 39–40, 46
fuzzers, 15–16. See also fuzzing
cross-site scripting and, 20–22
SOAP, 185–190
FuzzHttpGetPort() method
   fuzzing SOAP service, 70–72
sqlmap utility, 189
FuzzHttpPostPort() method (fuzzing SOAP service), 69
FuzzHttpPostPort() method
   fuzzing SOAP service, 72–75
sqlmap utility, 189–190
fuzzing
   defined, 16
   GET requests with mutational fuzzer, 22–25
fuzzing (continued)

JSON, 31–38
  capturing vulnerable JSON request, 31–33
  HTTP requests, 33–34, 35–37
  iterating over key/value pairs, 34–35
  setting up vulnerable appliance, 31
  testing, 37–38
POST requests, 25–31
  parameters, 29–31
  writing requests, 27–29
SOAP endpoints for SQL injection vulnerabilities, 68–79
  HTTP POST SOAP port, 72–75
  individual SOAP services, 69–72
  running fuzzer, 78–79
  SOAP XML port, 75–78
SQL injections, 19–20, 38–51
  virtual machines, 16–18
    adding host-only virtual network, 16
    booting from BadStore ISO, 17–18
    creating, 17
FuzzService() method (SOAP service), 69
FuzzSoapPort() method
  fuzzing SOAP service, 75–78
sqlmap utility, 188–189

G

get_version command (OpenVASSession class), 139
GetBootKey() method, 259–261, 262–263
GetLength() method (blind SQL injection), 47–49
GetLog() method (SqlmapLogItem class), 183–184
GetMultipartFormData() method (CuckooSession class), 153–155
GetNodeKey() method, 261–262
GetObject() method (MetasploitSession class), 216
GetOptions() method (SqlmapManager class), 179
GetPdfSiteReport() method (NexposeManager class), 128
GetProgress() method (ArachniRPCManager class), 237
GET requests
  adding sqlmap GET request support to SOAP fuzzer, 185–187
  fuzzing with mutational fuzzer, 22–25
  sqlmap REST API, 169–170
  using WebRequest method to execute, 174–175
GetScanConfigurations() method (OpenVASManager class), 141–142
GetScanStatus() method
  ArachniHTTPManager class, 227–228
  NexposeManager class, 127
  SqlmapStatus class, 181–182
GetStream() method
  ArachniRPCSession class, 233
  OpenVASSession class, 138
GetTaskDetails() method (CuckooManager class), 159, 163
GetTaskReport() method (CuckooManager class), 159, 163
GetTaskResults() method (OpenVASManager class), 143–144
GetTasks() method (OpenVASManager class), 143–144
GetValueKey() method, 261
GetValue() method (blind SQL injections), 49–50
GetVersion() method (ClamdManager class), 205
  globally unique ID (Guid), 110

H

Hello World example, 2–3
host-only virtual network, adding to VM, 16
HTTP requests
  building, 23–24
  DELETE, 167
GET requests
  adding sqlmap GET request support to SOAP fuzzer, 185–187
  fuzzing with mutational fuzzer, 22–25
sqlmap REST API, 169–170
using WebRequest method to execute, 174–175

JSON
capturing vulnerable, 31–33
Fuzz() method, 35–37
reading, 33–34
NessusSession class, 106–107
NexposeSession class, 120–121
POST
fuzzing, 25–31, 72–75
integrating sqlmap utility, 187–188
parameters, 28
sqlmap API, 167, 170–172
PUT, 167
REST APIs and, 104
writing ExecuteCommand() methods to handle, 151–153
HTTP responses (NexposeSession class), 121–123
HttpWebRequest class, 24, 36, 42

I
IDEs (integrated development environments), 1–2, 210
IL (intermediate language), 246
ILSpy decompiler, 242
instances
defined, 4
RPC framework, 230
instantiated objects, 24
integrated development environments (IDEs), 1–2, 210
interfaces, defined, 4–7
intermediate language (IL), 246
int.Parse() method, 83, 176
IsBusy() method (ArachniRPCManager class), 237

J
JavaScript Object Notation. See JSON
Join() method (connect-back payload), 84
JSON (JavaScript Object Notation).
See also sqlmap utility
fuzzing
capturing vulnerable JSON request, 31–33
HTTP requests, 33–34, 35–37
iterating over key/value pairs, 34–35
setting up vulnerable appliance, 31
testing, 37–38
Json.NET library, 34, 51
JsonConvert class, 181
SqlmapManager class, 177–179
SqlmapSession class, 176–177

K
kernel32.dll library, 96–98

L
Language-Integrated Query. See LINQ
Level property (SqlmapLogItem class), 182–183
libraries
ClamAV, 193–201
accessing functions, 196–197
ClamEngine class, 197–198
classes, 195
Dispose() method, 198–200
enumerations, 194–195
scanning files, 198–200
testing, 200–201
Json.NET, 34, 51
JsonConvert class, 181
SqlmapManager class, 177–179
SqlmapSession class, 176–177
MSGPACK, 209–210
installing, 211
NuGet package manager, 210
referencing, 211–212
Object Relational Mapping, 20, 242–244
LINQ (Language-Integrated Query)
Descendants() method, 145
LINQ to XML classes, 76
payloads and, 87
Single() method, 69, 70
StringToByteArray() method, 262
System.Linq namespace, 84
Linux
BadStore ISO, 16, 17–18, 25–31
ClamAV library, 193–201
executing native Linux payloads, 98–102
generating Metasploit payloads, 96
installing ClamAV, 192
printf() function, 13
LogOut() method
   NessusSession class, 107–108
   NexposeSession class, 121–123
long.Parse() method, 176

M
Main() method, 7–9
   Arachni, 237–239
   ClamdManager class, 205
   Cuckoo Sandbox, 156, 163
   Metasploit, 219–220
   registry hives, 259, 263
   SOAP endpoint fuzzer, 68
   SqlmapManager class, 182
   testing GetBootKey() method, 263
MakeRequest() method
   blind SQL injections, 47
   NessusSession class, 106–107
managed assemblies, 241
   ILSpy, 242
   monodis program, 245–247
   NuGet packages, 242–244
   testing decompilers, 244–245
managed code, 96
   Marshal.Copy() method (payloads), 101–102
   Marshal.GetDelegateForFunctionPointer() method (payloads), 101–102
   MessageBox() function (Windows), 13
   MessagePackToDictionary() method (MetasploitSession class), 215
   Message property (SqlmapLogItem class), 182
   Metasploit, 207
   interacting with shell, 221–222
   MSGPACK library, 209–212
   installing, 211
   NuGet package manager, 210
   referencing, 211–212
   payloads
   executing native Linux payloads, 98–102
   generating, 96
   setting up, 94–96
   unmanaged code, 96–98
   RPC API, 208–209
   running exploit, 220–221
Metasploitable 2, 209
   MetasploitManager class, 217–219
   MetasploitSession class, 212–213
   Execute() method, 213–215
   testing, 217
   transforming response data, 215–217
   method overloading, 151–152
   methods
      assigning delegates to, 9
      defined, 4
   MID() SQL function, 46
   MonoDevelop
      installing, 2
      installing MSGPACK library, 210–212
   monodis program, 245–247
   Mono framework. See managed assemblies
   msfvenom tool (Metasploit), 96, 103
   MSGPACK library, 209–210
      installing, 211
      NuGet package manager, 210
      referencing, 211–212
   mutational fuzzers
      defined, 15
      fuzzing GET requests with, 22–25

N
   Name property (SoapMessage class), 59, 61
   namespaces
      defined, 3
      SOAP XML, 76
      System.Linq namespace, 84
      XML, 56–57
   native libraries, 12–13. See also libraries
   native x86 assembly, 241. See also managed assemblies
   Nessus, 103–104
      NessusManager class, 109–110
      NessusSession class, 105–110
      HTTP requests, 106–107
      logging out, 107–108
      testing, 108–109
      performing scan, 110–113
      REST architecture and, 104–105
   .NET library. See managed assemblies
   network stream
      binding payloads, 85–88
      connect-back payloads, 82–84
   NewTask() method (SqlmapManager class), 178–179
Nexpose, 115
  automating vulnerability scan, 126–127, 130
  installing, 116–118
NexposeManager class, 124–125
NexposeSession class, 118–124
  authenticating API, 124
  Dispose() method, 123
  ExecuteCommand() method, 120–123
  finding API version, 123–124
  Logout() method, 121–123
PDF site report, 128, 130
  performing scan, 129
NodeKey class (registry hives), 250, 253–257
ORM (Object Relational Mapping) libraries, 20, 242–244
OS X
  ClamAV library, 192, 196
  .NET decompilers, 242
  Xamarin Studio, 2
P
  Packer class (Metasploit), 214
  parameters, fuzzing, 29–31
Parameters property (SoapMessage class), 59
  parent class, defined, 4
ParseChildNodes() method (NodeKey class), 256–257
ParseMessages() method (WSDL class constructor), 57–58, 62
Parse() methods
  connect-back payload, 83
  int.Parse() method, 83, 176
  long.Parse() method, 176
  ParseChildNodes() method, 256–257
  ParseMessages() method, 57–58, 62
  ParseTypes() method, 56–57
  short.Parse() method, 176
  ParseTypes() method (WSDL class constructor), 56–57
parsing
  registry hives, 252–259
  WSDL XML documents, 55–67
WSDL class constructor, 55–56
  SoapBinding class, 64–65
  SoapBindingOperation class, 65–66
  SoapMessage class, 60–61
  SoapMessagePart class, 61–62
  SoapOperation class, 63–64
  SoapPortType class, 62–63
  SoapService class, 66–67
  SoapType class, 58–60
  writing initial parsing methods, 56–58
WSDL class constructor, 55–56
payloads, 81–82
  binding, 85–88
  accepting data, 86
  executing commands from stream, 87–88
  returning output, 87
  running commands, 87

O
  object-oriented language, 3
Object Relational Mapping (ORM) libraries, 20, 242–244
  objects, defined, 4
OMP (OpenVAS Management Protocol), 133
OpenVAS, 133
  installing, 134
  OpenVASManager class, 140–145
  automation, 144–145
  CreateSimpleTarget() method, 141–142
  CreateSimpleTask() method, 143
  GetScanConfigurations() method, 141–142
  GetTaskResults() method, 143–144
  GetTasks() method, 143–144
  StartTask() method, 143
OpenVASSession class, 134–139
  authentication, 135–136
  ExecuteCommand() method, 136–137
  get_version command, 139
  GetStream() method, 138
  ReadMessage() method, 137–138
  SSL certificate validation, 138–139
OpenVAS Management Protocol (OMP), 133
  optional arguments, 10–11
ORD() SQL function, 46
payloads (continued)
connect-back payloads, 82–85
network stream, 82–84
running, 84–85
running commands, 84–85
Metasploit, 94–102
executing native Linux payloads, 98–102
executing native Windows payloads as unmanaged code, 96–98
generating, 96
setting up, 94–96
using UDP to attack network, 88–94
attacker’s code, 92–94
code for target machine, 89–91
PDF site report (Nexpose), 128, 130
Platform Invoke (P/Invoke), 12, 193
ports (WSDL), 55
HTTP POST SOAP port, 72–75
SOAP XML port, 75–78
posix_memalign() function, 99–101
POST parameters, sending to SOAP service, 74–75
POST requests
fuzzing, 25–27
parameters, 29–31
writing requests, 27–29
integrating sqlmap utility, 187–188
sqlmap REST API, 170–172
printf() function (Linux), 13
Process class
binding payloads, 87–88
connect-back payload, 84–85
network attack via UDP, 91
ProcessStartInfo class
binding payloads, 87–88
connect-back payload, 84–85
network attack via UDP, 91
properties, defined, 4
Python
Cuckoo Sandbox and, 147, 149
sqlmap, 168, 170

ReadChildrenNodes() method (NodeKey class), 255–256
ReadChildValues() method (NodeKey class), 257
ReadInt32() method (NodeKey class), 255
ReadMessage() method
ArachniRPCSession class, 233, 235
OpenVASSession class, 137–138
ReadNodeStructure() method (NodeKey class), 254–255
Regex class (SQL injections), 42–43
RegistryHive class, 252–253
registry hives, 249–250
dumping boot key, 259–264
GetBootKey() method, 259–261, 262–263
GetNodeKey() method, 261–262
GetValueKey() method, 261
StringToByteArray() method, 262
verifying boot key, 263–264
exporting, 250–252
reading, 252–259
NodeKey class, 253–257
RegistryHive class, 252–253
ValueKey class, 258–259
structure of, 250
testing, 259
remote procedure call API. See RPC API
REST (representational state transfer) architecture. See also sqlmap utility
Arachni and, 224–228
Cuckoo Sandbox and, 148–150
Nessus and, 104–105
sqlmap, 169–170
RLIKE keyword (blind SQL injections), 43–44
calling methods, 50–51
creating true/false responses, 44
GetValue() method, 49–50
MakeRequest() method, 47
printing values, 50–51
retrieving lengths of values, 47–49
userdb table, 45–47
using to match search criteria, 44–45
root node key (registry hives), 250
RPC (remote procedure call) API
Arachni, 228–237
ArachniRPCManager class, 236–237
ArachniRPCSession class, 230–234

R
Rapid7
Metasploit, 94
Nexpose, 115–116
ExecuteCommand() method, 234–235
   manually running, 229–230
Metasploit, 208–209
Ruby programming language
   Arachni web application, 223
   Metasploit, 94–96
Ruby Version Manager (RVM), 95

ScanFile() method (ClamEngine class), 198–200
Scan() method (ClamdManager class), 205
scanning
   ClamAV library, 198–200
   in Nessus, 110–113
   in Nexpose, 126–127, 129
   sqlmap scan log, 172
ScanSite() method (NexposeManager class), 127
SDLC (software development life cycle), 224
SelectNodes() method (WSDL class constructor), 57
SELinux, 100
SerializeObject() method (JsonConvert class), 181
shell, interacting with Metasploit, 221–222
short.Parse() method, 176
Simple Object Access Protocol (SOAP), 19. See also SOAP endpoints;
   SOAP fuzzer
Single() method (LINQ), 69, 70
Skip() method (connect-back payload), 84
SOAP (Simple Object Access Protocol), 19. See also SOAP endpoints;
   SOAP fuzzer
SOAPAction HTTP header (SOAP endpoint), 77–78
SoapBinding class (WSDL), 64–65
SoapBindingOperation class (WSDL), 65–66
SOAP endpoints, 53–54
   automatically fuzzing for SQL injection vulnerabilities, 68–79
   HTTP POST SOAP port, 72–75
   individual SOAP services, 69–72
running fuzzer, 78–79
SOAP XML port, 75–78
   parsing WSDL XML documents, 55–67
   class constructor, 55–56
   SoapBinding class, 64–65
   SoapBindingOperation class, 65–66
   SoapMessage class, 60–61
   SoapMessagePart class, 61–62
   SoapOperation class, 63–64
   SoapPortType class, 62–63
   SoapService class, 66–67
   SoapType class, 58–60
   writing initial parsing methods, 56–58
setting up vulnerable endpoint, 54
SOAP fuzzer
   calling new methods, 188–190
   GET requests, 185–187
   POST requests, 187–188
SoapMessage class (WSDL), 57, 60–61
SoapMessagePart class (WSDL), 61–62
SoapOperation class (WSDL), 63–64
SoapPortType class (WSDL), 62–63
SoapService class (WSDL), 66–67
SoapType class (WSDL), 58–60
SoapTypeParameter class (WSDL), 60
SOAP XML port, fuzzing, 75–78
Socket class, network attack via UDP, 89
software development life cycle (SDLC), 224
Split() method (connect-back payload), 84
SQL (Structured Query Language). See SQL injections; sqlmap utility
SQL injections, 19–20
   exploiting
      Boolean-based blind SQL injection, 43–51
      UNION-based, 38–43
   fuzzing SOAP endpoints for vulnerabilities, 68–79
HTTP POST SOAP port, 72–75
   individual SOAP services, 69–72
   running fuzzer, 78–79
   SOAP XML port, 75–78
SqlmapLogItem class, 182–183
SqlmapManager class, 177–179
  Main() method, 182
  options, 179–180
  performing scan, 180–182
SqlmapSession class, 173–174
  ExecuteGet() method, 174–175
  ExecutePost() method, 175
  testing, 176–177
SqlmapStatus class, 181–182
sqlmap utility, 167–168
  automating scan, 183–185
  integrating with SOAP fuzzer, 185–190
  calling new methods, 188–190
  GET requests, 185–187
  POST requests, 187–188
  reporting scan, 182–183
  running, 168–173
  sqlmap REST API, 169–170
  testing sqlmap API with curl, 170–173
SqlmapManager class, 177–179
  Main() method, 182
  options, 179–180
  performing scan, 180–182
SqlmapSession class, 173–174
  ExecuteGet() method, 174–175
  ExecutePost() method, 175
  testing, 176–177
SSL certificate validation
  (OpenVASSession class), 138–139
StartScan() method
  ArachniHTTPManager class, 227–228
  ArachniRPCManager class, 237
StartTask() method
  OpenVASManager class, 143
  SqlmapManager class, 180
stateful protocol, 85–88
stateless protocol, 88
static sites (Nexpose), 118
StreamReader class constructor (connect-back payload), 83
StreamReader ReadLine() method
  (connect-back payload), 83
strings-type options (monodis program), 245
StringToByteArray() method, 262
Structured Query Language. See SQL injections; sqlmap utility
subclasing, 4–6
System.Linq namespace (connect-back payload), 84

T

TaskFactory class (Cuckoo Sandbox), 162
TCP (Transmission Control Protocol)
  payloads, 81–82
  binding, 85–88
  connect-back payloads, 82–85
  UDP versus, 88–89
TcpClient class
  clamd daemon, 203
  connect-back payload, 82–84
TcpListener class (binding payloads), 85–86
Tenable Network Security, 103
TestGetRequestWithSqlmap() method
  (SOAP fuzzer), 185–187
testing
  ClamAV library, 200–201
  clamd daemon, 205–206
  GetBootKey() method, 263
  JSON fuzzer, 37–38
  MetasploitSession class, 217
  NessusSession class, 108–109
  Nexpose, 118
  registry hives, 259
  SqlmapSession class, 176–177
TestPostRequestWithSqlmap() method
  (SOAP fuzzer), 187–188
Time property (SqlmapLogItem class), 183
TLS (Transport Layer Security), 121
Transmission Control Protocol.
  See TCP

U

Ubuntu, 94
UDP (User Datagram Protocol)
  TCP versus, 88–89
  using to attack network, 88–94
  attacker’s code, 92–94
  code for target machine, 89–91
UdpClient class, 89
UNION-based SQL injections
  performing exploit by hand, 38–40
performing exploit programmatically, 40–43
building URL with payload, 41–42
creating markers to find user-names and passwords, 41
making HTTP request, 42–43
unmanaged code, 96–98
User Datagram Protocol. See UDP using keyword, 24

V
ValidateServerCertificate() method
(ArachniRPCSession class), 233
ValueKey class (registry hives), 250, 258–259
VirtualAlloc() function, 96–98
VirtualBox virtualization software, 16, 209. See also VMs
virtual machines. See VMs
Visual Studio IDE (Microsoft), 1–2
VMs (virtual machines), 12–13
adding host-only virtual network, 16
booting from BadStore ISO, 17–18
creating, 17
vulnerability scanners
Nessus, 103–113
NessusManager class, 109–110
NessusSession class, 105–109
performing scan, 110–113
REST architecture and, 104–105
Nexpose, 115–131
automating vulnerability scan, 126–127, 130
installing, 116–118
NexposeManager class, 124–125
NexposeSession class, 118–124
PDF site report, 128, 130
performing scan, 129
OpenVAS, 134–145
installing, 134
OpenVASManager class, 140–145
OpenVASSession class, 134–139

W
Web Service Description Language
XML documents, parsing. See WSDL XML documents, parsing
while loop
connect-back payload, 83
network attack via UDP, 89–90
Windows
ClamAV library, 192, 196
executing native Windows payloads as unmanaged code, 96–98
generating Metasploit payloads, 96
ILSpy decompiler, 242
kernel32.dll library, 96–97
MessageBox() function, 13
registry hives, 249–250
dumping boot key, 259–264
exporting, 250–252
reading, 252–259
structure of, 250
testing, 259
WSDL (Web Service Description Language) XML
documents, parsing, 55
class constructor, 55–56
SoapBinding class, 64–65
SoapBindingOperation class, 65–66
SoapMessage class, 60–61
SoapMessagePart class, 61–62
SoapOperation class, 63–64
SoapPortType class, 62–63
SoapType class, 58–60
writing initial parsing methods, 56–58

X
x86_64 assembly, 241. See also managed assemblies
Xamarin Studio IDE, 2
XElement class (SOAP XML), 76–77
XML node, 59–60
XPath query, 57–58
XSS (cross-site scripting), 20–22