CONTENTS IN DETAIL

FOREWORD by Matt Graeber xii

PREFACE xvii

Why Should I Trust Mono? xviii
Who Is This Book For? xviii
Organization of This Book xix
Acknowledgments xxi
A Final Note xxi

1
C# CRASH COURSE 1

Choosing an IDE 1
A Simple Example 2
Introducing Classes and Interfaces 4
    Creating a Class 4
    Creating an Interface 4
    Subclassing from an Abstract Class and Implementing an Interface 5
    Tying Everything Together with the Main() Method 7
    Running the Main() Method 8
Anonymous Methods 9
    Assigning a Delegate to a Method 9
    Updating the Firefighter Class 9
    Creating Optional Arguments 10
    Updating the Main() Method 11
    Running the Updated Main() Method 12
Integrating with Native Libraries 12
Conclusion 13

2
FUZZING AND EXPLOITING XSS AND SQL INJECTION 15

Setting Up the Virtual Machine 16
    Adding a Host-Only Virtual Network 16
    Creating the Virtual Machine 17
    Booting the Virtual Machine from the BadStore ISO 17
SQL Injections 19
Cross-Site Scripting 20
Fuzzing GET Requests with a Mutational Fuzzer 22
    Tainting the Parameters and Testing for Vulnerabilities 23
    Building the HTTP Requests 23
    Testing the Fuzzing Code 25
Fuzzing POST Requests 25
    Writing a POST Request Fuzzer 27
    The Fuzzing Begins 28
    Fuzzing Parameters 29

Fuzzing JSON 31
Setting Up the Vulnerable Appliance 31
Capturing a Vulnerable JSON Request 31
Creating the JSON Fuzzer 33
Testing the JSON Fuzzer 37
Exploiting SQL Injections 38
Performing a UNION-Based Exploit by Hand 38
Performing a UNION-Based Exploit Programmatically 40
Exploiting Boolean-Blind SQL Vulnerabilities 43
Conclusion 51

3
FUZZING SOAP ENDPOINTS 53

Setting Up the Vulnerable Endpoint 54
Parsing the WSDL 55
Creating a Class for the WSDL Document 55
Writing the Initial Parsing Methods 56
Writing a Class for the SOAP Type and Parameters 58
Creating the SoapMessage Class to Define Sent Data 60
Implementing a Class for Message Parts 61
Defining Port Operations with the SoapPortType Class 62
Implementing a Class for Port Operations 63
Defining Protocols Used in SOAP Bindings 64
Compiling a List of Operation Child Nodes 65
Finding the SOAP Services on Ports 66
Automatically Fuzzing the SOAP Endpoint for SQL Injection Vulnerabilities 68
Fuzzing Individual SOAP Services 69
Fuzzing the HTTP POST SOAP Port 72
Fuzzing the SOAP XML Port 75
Running the Fuzzer 78
Conclusion 79

4
WRITING CONNECT-BACK, BINDING, AND METASPLOIT PAYLOADS 81

Creating a Connect-Back Payload 82
The Network Stream 82
Running the Command 84
Running the Payload 85
Binding a Payload 85
Accepting Data, Running Commands, and Returning Output 86
Executing Commands from the Stream 87
Using UDP to Attack a Network 88
The Code for the Target's Machine 89
The Attacker's Code 92
Running x86 and x86-64 Metasploit Payloads from C# 94
Setting Up Metasploit 94
Generating Payloads 96

Executing Native Windows Payloads as Unmanaged Code 96
Executing Native Linux Payloads 98
Conclusion 102

5
AUTOMATING NESSUS 103

REST and the Nessus API 104
The NessusSession Class 105
Making the HTTP Requests 106
Logging Out and Cleaning Up 107
Testing the NessusSession Class 108
The NessusManager Class 109
Performing a Nessus Scan 110
Conclusion 113

6
AUTOMATING NEXPOSE 115

Installing Nexpose 116
Activation and Testing 117
Some Nexpose Parlance 118
The NexposeSession Class 118
The ExecuteCommand() Method 120
Logging Out and Disposing of Our Session 123
Finding the API Version 123
Driving the Nexpose API 124
The NexposeManager Class 124
Automating a Vulnerability Scan 126
Creating a Site with Assets 126
Starting a Scan 127
Creating a PDF Site Report and Deleting the Site 128
Putting It All Together 128
Starting the Scan 129
Generating a Report and Deleting the Site 129
Running the Automation 130
Conclusion 131

7
AUTOMATING OPENVAS 133

Installing OpenVAS 134
Building the Classes 134
The OpenVASSession Class 134
Authenticating with the OpenVAS Server 135
Creating a Method to Execute OpenVAS Commands 136
Reading the Server Message 137
Setting Up the TCP Stream to Send and Receive Commands 138
Certificate Validation and Garbage Collection 138
Getting the OpenVAS Version 139

The OpenVASManager Class 140
Getting Scan Configurations and Creating Targets 141
Wrapping Up the Automation. 144
Running the Automation 145
Conclusion 146

8 AUTOMATING CUCKOO SANDBOX 147

Setting Up Cuckoo Sandbox 148
Manually Running the Cuckoo Sandbox API 148
Starting the API. 148
Checking Cuckoo's Status 149
Creating the CuckooSession Class 151
Writing the ExecuteCommand() Methods to Handle HTTP Requests 151
Creating Multipart HTTP Data with the GetMultipartFormData() Method 153
Processing File Data with the FileParameter Class 155
Testing the CuckooSession and Supporting Classes 156
Writing the CuckooManager Class 157
Writing the CreateTask() Method 157
The Task Details and Reporting Methods 159
Creating the Task Abstract Class. 160
Sorting and Creating Different Class Types. 161
Putting It Together 163
Testing the Application 164
Conclusion 165

9 AUTOMATING SQLMAP 167

Running sqlmap 168
The sqlmap REST API 169
Testing the sqlmap API with curl 170
Creating a Session for sqlmap 173
Creating a Method to Execute a GET Request. 174
Executing a POST Request 175
Testing the Session Class 176
The SqlmapManager Class 177
Listing sqlmap Options. 179
Making a Method to Perform Scans. 180
The New Main() Method 182
Reporting on a Scan. 182
Automating a Full sqlmap Scan 183
Integrating sqlmap with the SOAP Fuzzer 185
Adding sqlmap GET Request Support to the SOAP Fuzzer 185
Adding sqlmap POST Request Support. 187
Calling the New Methods 188
Conclusion 190

10
AUTOMATING CLAMAV 191

Installing ClamAV 192
The ClamAV Native Library vs. the clamd Network Daemon 193
Automating with ClamAV's Native Library. 193
    Setting Up the Supporting Enumerations and Classes. 194
    Accessing ClamAV's Native Library Functions. 196
    Compiling the ClamAV Engine 197
    Scanning Files 198
    Cleaning Up. 200
    Testing the Program by Scanning the EICAR File 200
Automating with clamd 201
    Installing the clamd Daemon. 202
    Starting the clamd Daemon 202
    Creating a Session Class for clamd 203
    Creating a clamd Manager Class 204
    Testing with clamd 205
Conclusion 206

11
AUTOMATING METASPLOIT 207

Running the RPC Server 208
Installing Metasploitable 209
Getting the MSGPACK Library. 209
    Installing the NuGet Package Manager for MonoDevelop 210
    Installing the MSGPACK Library 211
    Referencing the MSGPACK Library 211
Writing the MetasploitSession Class 212
    Creating the Execute() Method for HTTP Requests and
        Interacting with MSGPACK 213
    Transforming Response Data from MSGPACK. 215
Testing the session Class 217
Writing the MetasploitManager Class 217
Putting It All Together 219
    Running the Exploit 220
    Interacting with the Shell 221
    Popping Shells 221
Conclusion 222

12
AUTOMATING ARACHNI 223

Installing Arachni 223
The Arachni REST API 224
    Creating the ArachniHTTPSession Class 225
    Creating the ArachniHTTPManager Class. 226
Putting the Session and Manager Classes Together 227

The Arachni RPC 228
Manually Running the RPC 229
The ArachniRPCSession Class 230
The Supporting Methods for ExecuteCommand() 232
The ExecuteCommand() Method 234
The ArachniRPCManager Class 236
Putting It All Together 237
Conclusion 239

13
DECOMPILING AND REVERSING MANAGED ASSEMBLIES 241

Decompiling Managed Assemblies. 242
Testing the Decompiler 244
Using monodis to Analyze an Assembly 245
Conclusion 247

14
READING OFFLINE REGISTRY HIVES 249

The Registry Hive Structure 250
Getting the Registry Hives 250
Reading the Registry Hive 252
Creating a Class to Parse a Registry Hive File. 252
Creating a Class for Node Keys 253
Making a Class to Store Value Keys 258
Testing the Library 259
Dumping the Boot Key 259
The GetBootKey() Method 259
The GetValueKey() Method. 261
The GetNodeKey() Method 261
The StringToByteArray() Method 262
Getting the Boot Key 262
Verifying the Boot Key 263
Conclusion 264

INDEX 265