

Table of Contents

Preface

xv

Part 1: Introduction to Ghidra

1

Getting Started with Ghidra

3

Technical requirements

3

WikiLeaks Vault 7

4

NSA release

4

Ghidra versus IDA and many other competitors

5

Ghidra overview

7

Installing Ghidra

7

Running Ghidra

10

Overview of Ghidra's features

11

Summary

20

Questions

21

2

Automating RE Tasks with Ghidra Scripts

23

Technical requirements

23

Using and adapting existing scripts

24

The script class

29

Script development

33

Summary

35

Questions

35

3

Ghidra Debug Mode

37

Technical requirements

37

Setting up the Ghidra development environment

38

Overview of the software requirements

38

Installing the JDK

39

Installing the Eclipse IDE

39

Installing PyDev	40	Explaining the Ghidra RCE vulnerability	61
Installing GhidraDev	43	Exploiting the Ghidra RCE vulnerability	61
Debugging the Ghidra code and Ghidra scripts	48	Fixing the Ghidra RCE vulnerability	63
Debugging Ghidra scripts from Eclipse	48	Looking for vulnerable computers	63
Debugging any Ghidra component from Eclipse	59	Summary	63
Ghidra RCE vulnerability	60	Questions	64
		Further reading	64

4

Using Ghidra Extensions 65

Technical requirements	66	Plugins	75
Installing existing Ghidra extensions	66	Exporters	76
Analyzing the source code of the Sample Table Provider plugin	71	Loaders	76
Understanding the Ghidra extension skeleton	74	Developing a Ghidra extension	77
Analyzers	74	Summary	82
Filesystems	75	Questions	82
		Further reading	82

Part 2: Reverse-Engineering

5

Reversing Malware Using Ghidra 85

Technical requirements	85	Analyzing the 0x00453340 function	99
Setting up the environment	86	Analyzing the 0x00453C10 function	103
Looking for malware indicators	87	Analyzing the 0x0046EA60 function	103
Looking for strings	87	Analyzing the 0x0046BEB0 function	104
Intelligence information and external sources	88	Analyzing the 0x0046E3A0 function	104
Checking import functions	90	Analyzing the 0x004559B0 function	105
Dissecting interesting malware sample parts	93	Analyzing the 0x004554E0 function	106
The entry point function	97	Analyzing the 0x0046C860 function	108
		Analyzing the 0x0046A100 function	110

Summary	113	Further reading	114
Questions	113		

6

Scripting Malware Analysis 115

Technical requirements	116	The delta offset	120
Using the Ghidra scripting API	116	Translating API hashes into addresses	123
Writing scripts using the Java programming language	118	Deobfuscating the hash table using Ghidra scripting	125
Writing scripts using the Python programming language	119	Improving the scripting results	128
Deobfuscating malware samples using scripts	120	Summary	130
		Questions	130
		Further reading	130

7

Using Ghidra's Headless Analyzer 131

Technical requirements	131	Running non-GUI scripts in a project	137
Why use headless mode?	132	Summary	144
Creating and populating projects	133	Questions	144
Analyzing imported or existing binaries	136	Further reading	144

Part 3: Binary Analysis

8

Binary Diffing 147

Technical requirements	148	Finding similar functions	160
Using Ghidra BSim	149	Querying the BSim database	165
Getting BSim up and running	149	Finding patched code – function comparison	171

Binary diffing usage in vulnerability research	176	Questions	177
Summary	177	Further reading	177

9

Auditing Program Binaries 179

Technical requirements	179	Format strings	186
Understanding memory corruption vulnerabilities	180	Finding vulnerabilities using Ghidra	187
Understanding the stack	181	Exploiting a simple stack-based buffer overflow	191
Stack-based buffer overflow	183	Summary	198
Understanding the heap	185	Questions	199
Heap-based buffer overflow	185	Further reading	199

10

Scripting Binary Audits 201

Technical requirements	201	Analyzing the caller function using P-Code	206
Looking for vulnerable functions	202	P-Code versus assembly language	208
Retrieving unsafe C/C++ functions from the symbols table	203	Retrieving P-Code and analyzing it	209
Decompiling the program using scripting	204	Using the same P-Code-based script in multiple architectures	210
Looking for sscanf callers	205	Summary	212
Enumerating caller functions	205	Questions	212
		Further reading	213

Part 4: Extending Ghidra for Advanced Reverse-Engineering

11

Developing Ghidra Plugins 217

Technical requirements	217	Overview of existing plugins	218
------------------------	-----	------------------------------	-----

Plugins included with the Ghidra distribution	218
Third-party plugins	220
The Ghidra plugin skeleton	221
The plugin documentation	221
Writing the plugin code	222
The provider for a plugin	223

Developing a Ghidra plugin	226
Documenting the plugin	226
Implementing the plugin class	226
Implementing the provider	227
Summary	232
Questions	232
Further reading	232

12

Incorporating New Binary Formats **233**

Technical requirements	233	The old-style DOS executable (MZ) loader	241
Understanding the difference between raw binaries and formatted binaries	234	Understanding filesystem loaders	249
Understanding raw binaries	234	FileSystem Resource Locator	249
Understanding formatted binaries	238	Summary	250
Developing a Ghidra loader	240	Questions	251
The old-style DOS executable (MZ) parser	240	Further reading	251

13

Analyzing Processor Modules **253**

Technical requirements	254	Developing Ghidra processors	263
Understanding the existing Ghidra processor modules	254	Documenting processors	263
Overviewing the Ghidra processor module skeleton	258	Identifying functions and code using patterns	264
Setting up the processor module development environment	258	Specifying the language and its variants	265
Creating a processor module skeleton	259	Summary	269
		Questions	270
		Further reading	270

Contributing to the Ghidra Community 271

Technical requirements	272	Suggesting new features	278
Overviewing the Ghidra project	272	Submitting questions	280
The Ghidra community	273	Submitting a pull request to the Ghidra project	282
Exploring contributions	274	Summary	287
Understanding legal aspects	274	Questions	288
Submitting a bug report	274	Further reading	288

Extending Ghidra for Advanced Reverse-Engineering 289

Technical requirements	290	Adding symbolic execution capabilities to Ghidra with AngryGhidra	295
Learning the basics of advanced reverse-engineering	290	Converting from PCode into LLVM with pcode-to-llvm	297
Learning about symbolic execution	290	Summary	298
Learning about SMT solvers	292	Questions	298
Learning about concolic execution	294	Further reading	298
Using Ghidra for Advanced reverse-engineering	295		

Part 5: Debugging and Applied Malware Analysis

Debugging 301

Technical requirements	301	Execution flow control	313
Ghidra debugger overview	302	Stepping	313
Starting the Ghidra debugger	303	Breakpoint	313
Debugger windows and toolbar	306	Debugging the simple_encoder.exe application	317
Debugger specific toolbar	311		

Remote debugging	326	Summary	336
Debugging a Windows kernel	333	Further reading	337

17

Unpacking in-the-Wild Malware 339

Technical requirements	340	Summary	364
Malware overview	340	Further reading	364
Unpacking malware	342		

18

Reverse-Engineering Ransomware 365

Technical requirements	365	Initial exploration	393
General working principles of ransomware	366	Identifying imported libraries and functions	393
Initial infection vector	366	Tracing calls to cryptographic functions	394
Installation and execution	369	Identifying custom or embedded encryption algorithms	394
Encryption	379	Using plugins to find known crypto signatures and constants	395
C2 communication and exfiltration of data	391	Summary	396
Ransom demand notification	392	Further reading	397
Identifying encryption algorithms	393		

Appendix A

Answer Key 399

Chapter 1	399	Chapter 9	407
Chapter 2	402	Chapter 10	408
Chapter 3	404	Chapter 11	408
Chapter 4	404	Chapter 12	409
Chapter 5	405	Chapter 13	409
Chapter 6	405	Chapter 14	409
Chapter 7	406	Chapter 15	410
Chapter 8	406		

