BOSTON CYBERNETICS INSTITUTE

Reverse-Engineering with Ghidra

4-Day Course

I. Course Description

This is a majority hands-on course on using Ghidra for reverse-engineering and vulnerability research. Exercises include Windows binaries, Linux binaries, and device firmware, and will be in a variety of architectures, including ARM, PowerPC, MIPS, x86, and x64. After completing this course, students will have the practical skills to use Ghidra in their day-to-day reversing tasks.

A. Learning Objectives

- Students will have the ability to perform static analysis of real-world binaries and firmware in Ghidra
- Students will have the ability to use manual and automated techniques in Ghidra
- Students will know how to leverage Ghidra's strengths and how to complement its weaknesses

B. Student Prerequisites

Students are expected to have some experience with static and dynamic analysis, Linux, Windows, command line tools, shell scripting, C, and Python. Students should have the ability to do the following:

- Declare an array pointer in C
- Write a python script to XOR an encoded string
- Perform a function trace using a debugger
- Identify dead code using a disassembler

C. Required Hardware/Software

Students are expected to bring their own laptops. The laptops are required to run a 30GB virtual machine but will not perform any intensive computation. A recommended hardware configuration would have the following:

- 50 GB of free hard disk space
- 16 GB of RAM
- 4 Processor cores
- VMWare or Virtual Box to import an ova file

Course Schedule		
Day	**Theme**	**Topics**
1	Reversing Engineering With Ghidra	• Ghidra overview • Project management • Code navigation, manipulation • Symbols, labels, bookmarks, searching • Disassembler-decompiler interaction • Patching
2	Ghidra Expert Tools	• Decompiler deep dive • Datatype management • Memory management • P-code • Program flow • Ghidra tools • Plugin groups
3	Automation with Ghidra	• Java/Jython refresher • The Ghidra FlatAPI • Development with Eclipse and the GhidraDev plugin • Analysis in Ghidra headless mode • Java-Jython interop
4	Extending Ghidra with ExtensionPoint	• Loader, Decryptor, FileSystem • BuiltInDataType, AbstractAnalyzer

II. Bio's

Jeremy Blackthorne (@0xJeremy) is co-founder and lead instructor of the Boston Cybernetics Institute (BCI). Before BCI, he was a researcher in the Cyber System Assessments group at MIT Lincoln Laboratory. He was the co-creator and instructor for the Rensselaer Polytechnic Institute courses: Modern Binary Exploitation and Malware Analysis. Jeremy has published research at various academic conferences, including RAID, WOOT, and LatinCrypt. He has also presented or taught at hacker conferences, including INFILTRATE, REcon, and RingZer0. He served in the U.S. Marine Corps with three tours in Iraq. He is currently a PhD candidate in computer science at RPI and is a proud alumnus of RPISEC.