Hands-On Ghidra – A Tutorial about the Software Reverse Engineering Framework

Roman Rohleder
roman.rohleder@thalesgroup.com
Thales Group
Munich, Germany

ABSTRACT

In this tutorial, the Ghidra software reverse engineering framework will be presented, its characteristics highlighted and its features to the hitherto industry standard in reverse engineering tools, IDA Pro - the interactive disassembler, compared against. This framework was released on March the 5th 2019, by the National Security Agency under the Apache v2 license and brought with it a powerful decompiler for many different architectures (X86 16/32/64, ARM/AARCH64, Java/DEX bytecode, ...), which will be presented and its underlying intermediate language *p-code* and the corresponding *SLEIGH*-format explained. Further, hands-on demonstrations will follow, including the aforementioned SLEIGH-format, the plugin-system and the standalone-mode, showcased on different reverse engineering tasks like binary diffing, code lifting, deobfuscation and patching.

CCS CONCEPTS

• **Security and privacy → Software reverse engineering**.

KEYWORDS

reverse engineering, framework, disassembly, decompilation, code lifting

ACM Reference Format:
Roman Rohleder. 2019. Hands-On Ghidra – A Tutorial about the Software Reverse Engineering Framework. In *3rd Software Protection Workshop (SPRO'19), November 15, 2019,London, United Kingdom.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3338503.3357725

1 INTRODUCTION

Ghidra [2] was once an internal software reverse engineering framework, developed from the NSA, for the NSA. It was until recently, March the 5th of 2019, that it was presented at the RSA conference and released to the public, open source and under the admissible Apache v2 license [3].
While until that date many reverse engineering tools and frameworks already existed [1, 5–7], Ghidra was the first to offer a stable decompiler for a variety of architectures, for free, open source & platform independent and easy to modify with an extensive plug-in system. At the core of this broad architecture support lies a compact intermediate representation and translation language, unifying and simplifying a task that was previously labor-intensive and error prone.
In this tutorial therefore, Ghidra and its internal languages will be explained and showcased on common reverse engineering tasks, providing the attendee the knowledge to use and extend Ghidra for their personal projects.

SPRO'19, November 15,2019, London, United Kingdom

ACM ISBN 978-1-4503-6835-3/19/11.
https://doi.org/10.1145/3338503.3357725

2 TUTORIAL OVERVIEW

This tutorial will be split into a more theoretical part, containing a comparison to the well known reverse engineering tool IDA Pro [4], explanations of the underlying intermediate languages *p-code* and *SLEIGH*, and an explanation of the plug-in system and ways to automate Ghidra.
In the second, more practical part, the previously detailed theoretical background will be showcased on the four common reverse engineering tasks of binary diffing, code lifting, deobfuscation and binary patching[1].

2.1 Comparison to IDA Pro

To get a feel and an initial intuitive understanding of Ghidra, it will first be compared to the well known IDA Pro disassembler, which is commonly referred to as the industry standard reverse engineering tool. Main differences in the usage, the GUI and the features, as well as supported architectures and file formats will be detailed.

2.2 p-Code and SLEIGH-format

The most distinctive feature of Ghidra is its intermediate representation *p-code*, together with the *SLEIGH* language/format. While *p-code* itself is the intermediate language used between the different architecture's assembly and the higher-level C code, *SLEIGH* specifies in a compact way how to disassemble and lift the disassembly to their semantically equivalent *p-code* snippets.
Therefore, this part of the tutorial will explain these languages, how they are used and how they can be used to i.e. extend new architectures.

2.3 Plug-In System & Automation

The second most prominent feature of Ghidra is its Java-based plug-in system. While Ghidra already comes with many different plug-ins, the extensive documentation for most of the API's allows one to quickly build new plug-ins, tailored to ones needs. Hence, this part of the tutorial will focus on what plug-ins already exist and how to write new ones.

[1] All shown examples, plug-ins and scripts will be made publicly available at https://github.com/SektorROM/SPRO2019-Ghidra-Tutorial

Further it will be shown how to use Ghidra in a GUI-less standalone mode, and use said plug-ins in an automated fashion.

2.4 Use cases demonstration

At the end of the presentation different common reverse engineering tasks will be showcased, making use of the topics presented and detailed previously, providing a practical insight into day-to-day reverse engineering jobs with Ghidra.

Patch-/Binary-diffing
This use case will demonstrate how to diff two versions of a binary, to locate the differences between the two, in order to be able to concentrate on only these changes for further analysis.

Code lifting
The code lifting use case will show how to decompile and lift certain parts of a binary for extraction and further isolated analysis, independent of the complete source binary.

Deobfuscation
This scenario will show how to parse and process obfuscated code, in order to deobfuscate it, bringing it back into a more readable form.

Binary patching
The final use case will display how to locate a given functionality in a binary program and how to permanently patch it to exhibit a different behaviour.

3 INTENDED AUDIENCE

This tutorial's main target audience are security professionals and researchers, that are familiar with the topic of reverse engineering and/or familiar with one of the previously mentioned other tools and are open to a new, free alternative.
Further, the intended audience includes practitioners and academics that are interested into the topic of reverse engineering, eager to learn about common tasks and use cases in this field and how they can be carried out in an automatic or semi-automatic fashion, using a free and extensible new framework.

REFERENCES

[1] Vector 35. 2019. Binary Ninja: A New Kind of Reversing Platform. Retrieved August 29, 2019 from https://binary.ninja/
[2] National Security Agency. 2019. Ghidra - Software Reverse Engineering Framework. Retrieved August 29, 2019 from https://www.nsa.gov/resources/everyone/ghidra/
[3] National Security Agency. 2019. Ghidra - Software Reverse Engineering Framework. Retrieved August 29, 2019 from https://github.com/NationalSecurityAgency/ghidra
[4] Chris Eagle. 2011. *The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler.* No Starch Press, San Francisco, CA, USA.
[5] Hex-Rays. 2019. IDA Pro - The Interactive Disassembler. Retrieved August 29, 2019 from https://www.hex-rays.com/products/ida/
[6] pancake aka @trufae. 2019. Unix-like reverse engineering framework and commandline tools. Retrieved August 29, 2019 from https://rada.re/r/
[7] CEA IT Security. 2019. Miasm - Reverse engineering framework in Python. Retrieved August 29, 2019 from https://github.com/cea-sec/miasm