Program Synthesis for Capture-the-flags

Keywords: Binary analysis, deobfuscation, artificial intelligence, program synthesis

Main Supervisor
Grégoire MENGUY
CEA LIST
gregoire.menguy@cea.fr

Senior Supervisor
Sébastien BARDIN
CEA LIST
sebastien.bardin@cea.fr

Obfuscation [2, 3] aims to protect software from reverse engineering. It translates a program P into a functionally equivalent program P_o, harder to analyze. While obfuscation is used to protect *Intellectual Property* and other valuable software assets, it is also used to protect malware. Thus, automated *deobfuscation* methods [8, 1, 4, 6, 7] have been proposed to cope with the quick advances in obfuscation. Given an obfuscated program P_o, the goal is to simplify it into a simpler yet functionally equivalent program P^* – ideally, P^* should be as simple as the original unprotected code P.

A wide variety of deobfuscation methods have been proposed. In particular, black-box deobfuscation [6, 1] relies on program synthesis [5] to simplify highly obfuscated code snippets. Relying on input-output observations only, black-box deobfuscation is immune to standard obfuscation. However, it cannot handle code snippets too semantically complex.

In this project, we aim to study whether black-box deobfuscation and program synthesis are powerful enough to help reverse-engineer Capture-the-Flag (CTF) binaries. We especially aim to recover from CTFs a dataset of expressions that need to be understood to solve the challenges and study if synthesis can recover them. The project will proceed as follows:

- Get familiar with the state-of-the-art of black-box deobfuscation;
- Recover binaries from usual CTFs platforms (Hackropole, rootme) within the "reverse engineering" category, solve the challenges, and build a dataset of interesting expressions for it;
- Evaluate Xyntia (the open source black-box deobfuscator from the CEA — https://github.com/binsec/xyntia) over this new dataset;
- Propose an extension in Xyntia to handle the expressions currently not handled and compare this extension with Xyntia.

1 Expected deliverable

- A summary of the bibliography made by the student and a dataset of expressions found in CTFs;
- A documented implementation of the proposed extensions;
- A final report (slides) with the first two deliverables and a summary of the research and results.

2 Organization

Regular meetings will be organized with the supervisor (online or in person at CEA Nano-Innov, Saclay).

3 How to apply

To apply, students must send an email to the main supervisor (Grégoire Menguy) with the senior supervisor (Sébastien Bardin) in copy. **Please state in the email if you play CTFs, and your score on reverse engineering challenges.**

References

[1] Tim Blazytko et al. "Syntia: Synthesizing the Semantics of Obfuscated Code". In: USENIX *Security*. 2017.

[2] Christian Collberg and Jasvir Nagra. *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection, 2009.

[3] Christian Collberg, Clark Thomborson, and Douglas Low. *A taxonomy of obfuscating transformations*. 1997.

[4] Robin David, Luigi Coniglio, and Mariano Ceccato. "QSynth-A Program Synthesis based Approach for Binary Code Deobfuscation". In: *BAR 2020 Workshop*. Internet Society, 2020.

[5] Sumit Gulwani, Oleksandr Polozov, Rishabh Singh, et al. "Program synthesis". In: *Foundations and Trends® in Programming Languages* (2017).

[6] Grégoire Menguy, Sébastien Bardin, and Bonichon et al. "Search-Based Local Black-Box Deobfuscation: Understand, Improve and Mitigate". In: *Conference on Computer and Communications Security*. 2021.

[7] Rolf Rolles. "Unpacking Virtualization Obfuscators". In: *USENIX Conference on Offensive Technologies*. WOOT'09. 2009.

[8] Jonathan Salwan, Sébastien Bardin, and Marie-Laure Potet. "Symbolic deobfuscation: from virtualized code back to the original". In: *DIMVA*. 2018.