PhD Position in Computer Science:
Combining Program Analysis and Machine Learning for Cybersecurity

Keywords: computer science, machine learning, program analysis, software security, formal methods

The CEA LIST, Software Security Lab, has an open PhD position at the crossroads of automated software security analysis, program analysis and artificial intelligence, to begin *as soon as possible* in Paris-Saclay, France. The position benefits from the collaboration between CEA, Université of Montpellier and Simula.

Context and goals

Binary-level security analysis is sometimes mandatory, e.g., malware or vulnerability detection. Yet, such tasks are extremely hard to perform manually and would greatly benefit from automation. While binary-level program analysis is highly challenging, recent progress have been recently obtained [6,7] by adapting advanced methods from program analysis and formal methods. But these logical methods also reach their limits at some point. On the other hand, machine learning also starts to be successfully applied to such problems [4,5]. Interestingly, these two families of approaches are complementary: program analysis can *prove* facts while learning can *infer* facts.

The goal of this doctoral work is to understand how deduction-based approaches (from program analysis) and learning-based approaches (from AI) can be combined together for attacking hard challenges arising from security-oriented program analysis – typically, code hardening, vulnerability detection or reverse. Results will be implemented and experimented in the binary-level analysis tool BINSEC [3].

Position and host institution

The position is 3-years long. The successful candidate will be hosted at CEA LIST (Paris area, France) where he will be supervised by Sébastien Bardin, in close collaboration with Arnaud Gotlieb (Simula) and Nadjib Lazaar (Université de Montpellier).

Requirements

We welcome curious and enthusiastic students with a solid background in Computer Science – both theoretical and practical aspects. Candidates should be familiar with at least one of the following topics: program analysis or formal verification, machine learning, logic (especially automated solvers). A good knowledge of functional programming (OCaml) is a plus. Some experience in compiling, hacking or security challenges would be great.

Application

Applicants should send an email to Sébastien Bardin `sebastien.bardin@cea.fr` - including CV, motivation letter and references. **deadline:** please contact us as soon as possible, *a first round of selection takes place at the end of May.* **more information:** email, or `http://sebastien.bardin.free.fr/`

References

1 Cristian Cad, Koushik Sen: Symbolic execution for software testing: three decades later. Commun. ACM, 2013

2 Clark Barrett, Cesare Tinelli: Satisfiability Modulo Theories. Handbook of Model Checking 2018

3 Robin David, Sébastien Bardin, Thanh Dinh Ta, Laurent Mounier, Josselin Feist, Marie-Laure Potet, Jean-Yves Marion: BINSEC/SE: A Dynamic Symbolic Execution Toolkit for Binary-Level Analysis. SANER 2016

4 Patrice Godefroid, Hila Peleg, Rishabh Singh: Learn & Fuzz: machine learning for input fuzzing. ASE 2017

5 Blazytko, t., Contag, M., Aschermann, C., Holz, T.: Syntia: Synthesizing the Semantics of Obfuscated Code. In:USENIX Security Symposium 2017. Usenix

6 Sébastien Bardin, Robin David, Jean-Yves Marion: Backward-Bounded DSE: Targeting Infeasibility Questions on Obfuscated Codes. IEEE Symposium on Security and Privacy 2017

7 Thanassis Avgerinos, David Brumley, John Davis, Ryan Goulden, Tyler Nighswander, Alexandre Rebert, Ned Williamson: The Mayhem Cyber Reasoning System. IEEE Security & Privacy 16(2): 52-60 (2018)