Research-oriented internships, Software Security, 4-6 months, CEA Paris-Saclay, France
Code-level Cybersecurity & Program Analysis: Vulnerabilities, Verification, Reverse

Keywords: software security, vulnerabilities, reverse & deobfuscation,
program analysis, formal methods, software testing

The BINary-level SECurity research group (BINSEC) at CEA List has several open internship positions at the crossroad of software security, program analysis and formal methods, to begin *as soon as possible* at Paris-Saclay, France. Positions are 4-6 month long and can *open the way to a doctoral work*. All these positions are articulated around the BINSEC open-source platform (`https://binsec.github.io`), which aims at providing automatic tools for low-level security analysis, by adapting software verification and testing methods initially developed for safety-critical systems.

Host	Commissariat à l'Énergie Atomique, Software Security Laboratory
Place	Paris-Saclay, France
Team	Binary-level security analysis
Advisor(s)	Sébastien Bardin, Matthieu Lemerre, Michaël Marcozzi (`first.last@cea.fr`)

Context. Several major classes of security analyses have to be performed on machine code, such as vulnerability analysis of mobile code or commercial off-the-shelf software, deobfuscation or malware inspection. These analyses are very challenging, and they are still relatively poorly tooled. Our long-term goal is to leverage recent advances in software verification, software testing, security analysis and artificial intelligence in order to propose efficient semantic tools for low-level security investigations.

Current topics. We are looking for curious and enthusiastic students willing to work on the following directions:

- *vulnerability detection at scale* [1, 3], with combination of symbolic execution, fuzzing and static analysis;
- *static analysis of microkernels* [6] or data-structure libraries [7] using abstract interpretation and advanced types;
- *binary-level formal verification* of crypto-primitives [2, 4] using symbolic execution;
- *advanced reverse, certified decompilation*, via the combination of program analysis and artificial intelligence [5, 8, 9].

More details on the topics will be happily provided! The list is not exhaustive, ask us if you have some project in mind.

For each topic, the goal is to start from existing published solutions (if any), to identify their strengths and weaknesses, and to propose and evaluate a new solution. Results will be integrated in the open-source BINSEC platform. All positions include theoretical research as well as prototyping (in OCaml for most topics) and experimental evaluation.

Host. The BINary-level SECurity research group (BINSEC) of CEA List is a leading group in code analysis for low-level security, with regular publications in top-tier international venues in security, formal methods and software engineering. We work in close collaboration with other French and international research teams, industrial partners and national agencies. CEA List is part of Université Paris-Saclay and based at Palaiseau (91). CEA is ranked as one of the leading global innovators by Reuters and Université Paris-Saclay as the 13th best university in the world in the Shanghai ranking.

Requirements. We welcome curious and enthusiastic students with a solid background in Computer Science, both theoretical and practical. A good knowledge of functional programming (OCaml) is appreciated for most topics. Some experience in verification, testing, security, logic or compilation would be great.

Application. Applicants should send an e-mail to `binsec-jobs@saxifrage.saclay.cea.fr` – including CV and motivation letter. **Deadline:** as soon as possible (first come, first served). Contact us for **more information**.

References

[1] Patrice Godefroid, Michael Y. Levin, and David A. Molnar. SAGE: whitebox fuzzing for security testing. *Commun. ACM*, 55(3):40–44, 2012.

[2] Lesly-Ann Daniel, Sébastien Bardin, and Tamara Rezk. Binsec/Rel: Efficient Relational Symbolic Execution for Constant-Time at Binary-Level In *2020 IEEE Symposium on Security and Privacy (S&P 2020)*.

[3] Sébastien Bardin, Manh-Dung Nguyen About Directed Fuzzing and Use-After-Free: How to Find Complex & Silent Bugs? Black Hat USA 2020

[4] Lesly-Ann Daniel, Sébastien Bardin, Tamara Rezk. Hunting the Haunter: Efficient Relational Symbolic Execution for spectre with Haunted RelSE. In the *28th Network and Distributed System Security Symposium (NDSS 2021)*

[5] Grégoire Menguy, Sébastien Bardin, Richard Bonichon, Cauim de Souza Lima. Search-based Local Blackbox Deobfuscation: Understand, Improve and Mitigate. In the *28th ACM Conference on Computer and Communications Security (CCS 2021)*

[6] Olivier Nicole, Matthieu Lemerre, Sébastien Bardin, Xavier Rival. No Crash, No Exploit: Automatic Verification of Embedded Kernels. In the *27th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2021, best paper)*

[7] Olivier Nicole, Matthieu Lemerre, Xavier Rival. Lightweight Shape Analysis based on Physical Types. In the *23rd International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2022)*

[8] Frédéric Recoules, Sébastien Bardin, Richard Bonichon, Matthieu Lemerre, Laurent Mounier, Marie-Laure Potet Interface Compliance of Inline Assembly: Automatically Check, Patch and Refine In the *43rd International Conference on Software Engineering (ICSE 2021)*

[9] Sébastien Bardin, Robin David, and Jean-Yves Marion. Backward-bounded DSE: targeting infeasibility questions on obfuscated codes. In *2017 IEEE Symposium on Security and Privacy (S&P 2017)*.