

**Keywords:** software security, vulnerabilities, code hardening, reverse & deobfuscation, program analysis, formal methods, machine learning

The CEA LIST, Software Security Lab (LSL), has several open internship positions in the area of low-level software verification and security, 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 methods initially developed for safety-critical systems.

**Topic** Security [also: Logic and Verification, Compilation, Machine Learning]  
**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.name@cea.fr](mailto:first.name@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, security analysis and artificial intelligence in order to propose efficient semantic tools for low-level security investigations.

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

- vulnerability detection at scale, with combination of cutting edges techniques such as symbolic execution, fuzzing and static analysis;
- binary-level formal verification of crypto-primitives or microkernels, with combination of abstract interpretation and symbolic execution;
- combination of program analysis and artificial intelligence for reverse or vulnerability analysis;
- advanced methods for code hardening, with a focus on either anti-reverse (obfuscation) or anti-exploitation (control-flow integrity) protections;

*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 (preferably in OCaml) and experimental evaluation.

**Host Institution.** Within CEA LIST, LSL is a twenty-person team dedicated to software verification, with a strong focus on real-world applicability and industrial transfer. We design methods and tools that leverage innovative approaches to ensure that real-world systems can comply with the highest safety and security standards. CEA LIST is located in Campus Paris Saclay, the largest European cluster of public and private research <https://www.universite-paris-saclay.fr/en>.

**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. Some experience in verification, security, logic or compilation would be great.

**Application.** Applicants should send an e-mail to Sébastien Bardin ( [sebastien.bardin@cea.fr](mailto:sebastien.bardin@cea.fr) ) – including CV and motivation letter. **Deadline:** as soon as possible. Contact us for **more information**.

## References

- [1] Gogul Balakrishnan and Thomas W. Reps. WYSINWYX: What You See Is Not What You Execute. *ACM Trans. Program. Lang. Syst.*, 2010.
- [2] Patrice Godefroid, Michael Y. Levin, and David A. Molnar. SAGE: whitebox fuzzing for security testing. *Commun. ACM*, 55(3):40–44, 2012.
- [3] Cristian Cadar and Koushik Sen. Symbolic execution for software testing: three decades later. *Commun. ACM*, 56(2):82–90, 2013.
- [4] 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, SP 2020*.
- [5] Sébastien Bardin, Manh-Dung Nguyen About Directed Fuzzing and Use-After-Free: How to Find Complex & Silent Bugs? Black Hat USA 2020
- [6] 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, SP 2017*.
- [7] Benjamin Farinier, Sébastien Bardin, Richard Bonichon, Marie-Laure Potet. Model Generation for Quantified Formulas: A Taint-Based Approach. In *Proceedings of the 30th International Conference on Computer Aided Verification. CAV 2018*