18-month Postdoc Position

Extensive Code Security Analyses for Frama-C

Keywords: information flow security, abstract interpretation, runtime verification
Programming Languages: OCaml, C
Tool: Frama-C

Context: CEA LIST, Software Security and Reliability Lab

The Software Security and Reliability Laboratory (LSL) at CEA LIST has an ambitious goal: help designers, developers and validation experts ship high-confidence systems and software. Objects in our surroundings are getting more and more complex, and we have built a reputation for efficiently using formal reasoning to demonstrate their trustworthiness. Within the CEA LIST Institute, LSL is dedicated to inventing the best possible means to conduct formal verification. In collaboration with the most creative people in academia and the industry, we design methods and tools that leverage innovative approaches to ensure that real-world systems can comply with the highest safety and security standards

Our organizational structure is simple: those who pioneer new concepts are the ones who get to implement them. We are a forty-person team, and your work will have a direct and visible impact on the state of formal verification. CEA LIST's offices are located at the heart of Campus Paris Saclay, in the largest European cluster of public and private research.

Work Description

Our team develops Frama-C [5] (`http://frama-c.com`), a code analysis platform for C programs which provides several collaborative analyzers as plug-ins. Frama-C itself is developed in OCaml. Frama-C allows the user to annotate C programs with formal specifications written in the ACSL specification language [3]. Frama-C can then ensure that a C program satisfies its formal specification by relying on several techniques including abstract interpretation (plug-in Eva), weakest preconditions calculus (plug-in Wp), and runtime verification (plug-in E-ACSL).

In the context of the newly accepted H2020 European project ENSURESEC (2020-2022) that aims at protecting e-commerce, we plan to use Frama-C for detecting security threats in cyber interfaces such as middleware, cryptographic libraries, or communication protocol implementations. The postdoc researcher will adapt existing Frama-C plug-ins and/or design new ones for that purpose.

Among others, one important topic is information flow security in order to detect information leakage. In that respect, possible solutions include:

- abstract interpretation through the design of a new Eva's abstract domain [4];
- runtime verification by taking advantage of unused bits of the E-ACSL's shadow memory state [7];
- extending and improving the dedicated hybrid information flow plug-in SecureFlow [1, 2] that allows the user to combine static and dynamic verifications to check non-interference.

Another topic of interest is access control enforcement checking that could be performed within SecureFlow, and/or the MetACSL plug-in [6] that allows the user to specify high-level global program properties such as security policies. Designing a dedicated plug-in could also be an option.

The postdoc researcher should formalize and implement his/her solutions, prove their soundness and evaluate them on realistic use cases (e.g. provided by Ensuresec's industrial partners).

Application

Knowledge in at least one of the following fields is required:

- program development (OCaml, C)
- semantics of programming languages
- information flow security
- compilation
- static analysis
- runtime verification
- formal methods for program verification

Salary: academic competitive (vary *w.r.t.* diploma and former experience)

Availability: 2^d semester 2020; a 3-month procedure for administrative and security purposes is required

Contact: Julien Signoles (julien.signoles@cea.fr);

References

[1] M. Assaf, J. Signoles, É. Totel, and F. Tronel. Program transformation for non-interference verification on programs with pointers. In *International Information Security and Privacy Conference (SEC'13)*, July 2013.

[2] G. Barany and J. Signoles. Hybrid Information Flow Analysis for Real-World C Code. In *International Conference on Tests and Proofs (TAP'17)*, July 2017.

[3] P. Baudin, J.-C. Filliâtre, C. Marché, B. Monate, Y. Moy, and V. Prevosto. *ACSL: ANSI/ISO C Specification Language.*

[4] S. Blazy, D. Bühler, and B. Yakobowski. Structuring Abstract Interpreters through State and Value Abstractions. In *International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'17)*, January 2017.

[5] F. Kirchner, N. Kosmatov, V. Prevosto, J. Signoles, and B. Yakobowski. Frama-C: A Software Analysis Perspective. *Formal Aspects of Computing*, 2015.

[6] V. Robles, N. Kosmatov, V. Prevosto, L. Rilling, and P. Le Gall. MetAcsl: Specification and Verification of High-Level Properties. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'19)*, April 2019.

[7] K. Vorobyov, J. Signoles, and N. Kosmatov. Shadow State Encoding for Efficient Monitoring of Block-level Properties. In *International Symposium on Memory Management (ISMM'17)*, pages 47–58, June 2017.