Proved-Secure Compilation for RISC-V Processor

2023-2024

VERIMAG Laboratory (Grenoble – France)
Teams: PACS (Proofs and Code Analysis for Security) and Formal Proofs
Advisors: David Monniaux & Bruno Ferres
david.monniaux@univ-grenoble-alpes.fr
bruno.ferres@univ-grenoble-alpes.fr

Keywords secured compilation; `CompCert`; hardware security; micro-architecture

Context

Fault injections (using laser beams, EM injections, or even purely software attacks) are a real threat against all embedded systems [1, 2]. Even though counter-measures exist (either hardware, software, or hybrid) for those faults, it is difficult to add them in the compilation flow of programs. Moreover, the effectiveness of those counter-measures in the generated code is a difficult question, particularly when multiple counter-measures can be composed to harden the code against several fault models.

In the context of the PEPR Arsene, a national project aiming at developping secured hardware and software solutions for RISC-V, VERIMAG laboratory is investigating secured compilation flows. In such flows, the counter-measures can be included automatically in the generated code by the compiler, with no intervention from the programer — in particular, VERIMAG is interested in adding security feature in the CompCert compiler. CompCert[1] [3] is a compiler from the C language toward the assembly language of various architectures. Its particularity lies on formal proofs that the generated object code corresponds to the source code, those proofs being provided through the usage of a proof assistant (Coq[2]).

In the long term, implementing security counter-measures in CompCert could hence provide strong guarantes on the security of the compiled code. In particular, this could be used to demonstrate the robustness of the code against various state-of-the-art attacks.

Goal of the Internship

In this context, we are investigating the implementation of low-level security counter-measures in the CompCert compiler. Those counter-measures should consider the micro-architectural details of the target processor (such as pipeline structure, memory hierarchy, ...), and may, if needed, rely on dedicated hardware circuitry. In particular, in this internship, it is expected that the student:

- identifies one or multiple low-level counter-measure(s) for program hardening. Among the investigated counter-measures, we consider protections over the control-flow of the program (**C**ontrol-**F**low **I**ntegrity) [4], or pointer hardening [5]
- implements this counter-measure in CompCert

[1] `https://compcert.org/`
[2] `https://coq.inria.fr/`

- study various impacts of this new feature, including:
  - traceability: how can we prove that the counter-measure does exist in the generated machine code?
  - correctness: does the program with the additional counter-measure respects the initial semantics of the source code?
  - efficitiveness: is the counter-measure really effective against the considered faalt model?
  - performances: what is the impact of adding the counter-measure on the performances of the program?

This internship will benefit from VERIMAG expertise in the domain of verified compilation, in particular with CompCert[3], as well as several preliminary works on the topic of embedded systems security. The internship may take different directions, depending on the student's interests.
Moreover, depending on the internship progress, a pursuit for a Ph.D. thesis may be considered, in the context of the PEPR Arsene[4].

Ideal Applicant

This internship proposal is for Computer Science students, ideally at M2 level[5] (or last year of engineering school).
The applicant must be proficient in the following knowledge/skills:

- strong knowledge of compilation and formal methods
- interest in the low-level details of computers, notably processors architectures

Moreover, any interest in topics linked to computer security (hardware security, program hardening, *etc.*) is a real plus. However, the applicant may acquire those skills during the internship.

Applications

To apply, send an email to david.monniaux@univ-grenoble-alpes.fr and bruno.ferres@univ-grenoble-alpes.fr, with your resume, a short covering letter, and any document that may support your application.

Location

The internship will take place in VERIMAG laboratory, located in the campus of Grenoble:

Laboratoire VERIMAG, Bâtiment IMAG,
150 place du Torrent,
38401 Saint-Martin-d'Hères

[3] https://gricad-gitlab.univ-grenoble-alpes.fr/certicompil/Chamois-CompCert
[4] https://www.pepr-cyber-arsene.fr/
[5] Motivated applications at M1 level will also be considered.

Bibliographie

[1] B. Colombier, A. Menu, J.-M. Dutertre, P.-A. Moëllic, J.-B. Rigaud, and J.-L. Danger, "Laser-induced single-bit faults in flash memory: Instructions corruption on a 32-bit microcontroller," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 1–10, IEEE, 2019.

[2] I. Alshaer, B. Colombier, C. Deleuze, V. Beroulle, and P. Maistri, "Variable-length instruction set: Feature or bug?," in *2022 25th Euromicro Conference on Digital System Design (DSD)*, pp. 464–471, IEEE, 2022.

[3] S. Blazy and X. Leroy, "Formal verification of a memory model for C-like imperative languages," in *International Conference on Formal Engineering Methods (ICFEM 2005)*, vol. 3785 of *Lecture Notes in Computer Science*, pp. 280–299, Springer, 2005.

[4] M. Abadi, M. Budiu, U. Erlingsson, and J. Ligatti, "Control-flow integrity principles, implementations, and applications," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 1, pp. 1–40, 2009.

[5] Inc. Qualcomm Technologies, "Pointer Authentication on ARMv8.3 - Design and Analysis of the New Software Security Instructions." https://www.qualcomm.com/media/documents/files/whitepaper-pointer-authentication-on-armv8-3.pdf.