KU LEUVEN

GEDISTRIBUEERDE EN VEILIGE SOFTWARE (DISTRINET)
CELESTIJNENLAAN 200A, BOX 2402
3001 LEUVEN, BELGIUM

PhD/Researcher position:
Mechanized Systems-Level Security

Do you love building rock-solid code with uncompromising correctness? Do you enjoy hacking in Agda or Coq for hours, squeezing the most out of (dependent) types, and pushing proof assistants to their limits? Does it give you a warm, fuzzy feeling to know that what you are building comes with high-assurance machine-checkable guarantees? If so, we have the perfect PhD position for you! We are looking for an excellent PhD researcher to strengthen our research group, which is part of the DistriNet section of the department of Computer Science of KU Leuven. The position will be co-supervised by Dominique Devriese and Steven Keuchel. This is an opportunity to work at the exciting intersection of functional programming, formal methods and system security.

Research Context

The proposed position is in the context of the Katamaran project where we are developing a deductive separation logic-based program verifier for instruction set architecture (ISA) specifications written in the Sail language. An ISA specifies amongst other things the behavior of machine code and serves as a contract between hardware and software. For instance, the official formal specification of the RISC-V ISA is written in Sail. This verifier uses symbolic execution to generate verification conditions (VC) that are sufficient for the verification of the guarantees. It is implemented in Coq using the Iris separation logic framework and comes with a mechanized soundness proof of the full stack, including the symbolic executor, against an operational semantics of Sail.

ISAs provide security primitives implemented in hardware that can be used to protect components of a system. We use Katamaran to formalize and prove security guarantees of security primitives expressed as universal contracts. A universal contract is one that does not just apply to some specific code, like it is usually done in program verification, but a contract that applies to arbitrary code. The goal of ISA security is to reason about the composition of a full system that includes trusted code with untrusted code. In this setting, universal contracts can be used to express bounds on the authority of arbitrary untrusted code when proving properties about trusted code that interacts with untrusted code.

We have demonstrated the applicability of universal contracts to reason about full systems by mechanically verifying in Coq the integrity of a minimalistic proof of concept “femtokernel” that consists of a private data field and two basic blocks for initialization code and an interrupt handler. The integrity property expresses that no (in model) adversarial code can overwrite the interrupt handler code or the private data field during execution.

This provides more comprehensive end-to-end system verification results than similar, albeit much larger, verification efforts. For example, SeKVM does not connect the verification of C and Assembly code. seL4’s compilation is translation validated, but security guarantees are not end-to-end connected to the ISA level. Also, both SeKVM and seL4 do not work with authoritative semantics of ISAs.

Project

We want to scale up the verification of trusted low-level code (like firmware, security monitors, watchdogs, etc.) interacting with untrusted code through universal contracts for authoritative semantics of ISAs. For this project we want to focus on the methods to scale such verifications, but do not have a particular target code base in mind. Our intention is to work towards verification of, for instance, critical parts of the CherIoT platform or OpenSBI.

We envision developing, implementing, and mechanizing low-level intermediate representations of program code and make them particularly suited for (semi-)automated verification by, for instance, avoiding the need for higher-order features as done in similar work. A particularly appealing approach is to reuse functional variants of intermediate representations of compilers. These have not gained as much traction in practical implementations of compilers compared to traditional imperative approaches. However, compared to the compilation setting, we expect the benefits of functional representations to be particularly significant in the context of verification, and especially in mechanized verification.

We plan to extend the same ideas to intermediate verification languages (IVLs) such as Boogie. Our goal is to investigate the explicit modeling of join points and develop techniques to propagate verification information across them, introducing new alternatives to fight exponential path explosion.

Additionally, we aim to explore how verified security guarantees at the ISA level can be lifted to higher abstractions, for example as universal contracts on intermediate representations. By raising the abstraction level of verification, we seek to facilitate larger-scale, end-to-end system verifications. This involves verifying linked code that combines low-level assembly with intermediate representations, e.g., through multi-language semantics.

Requirements

- Academic Master's degree (or equivalent) in computer science or a closely related field (e.g., mathematics) obtained before the start date of the position.
- Familiarity with aspects of some of the following subjects and a strong willingness to learn more: statically-typed functional programming (e.g. in Haskell or OCaml) and monadic programming, (monadic) constraint generation (e.g. for type inference, analysis, verification, synthesis ...), compiler implementation, programming language semantics, machine-checked formal proving using proof assistants (like Agda or Coq), program logics like Hoare logics or separation logic in or outside of proof assistants.
- Strong interest in formal methods, deep scientific curiosity, and a creative mindset for developing novel ideas and solutions.
- Fluency in English, both written and spoken, with a willingness to communicate research findings through writing and presentations at international venues.

Offer

We offer a full-time research position with a PhD trajectory in computer science, with expected duration of four years with sufficient funding to attend international conferences, summer schools etc. The position starts as soon as possible, but no later than November 1st, 2025. It is co-supervised by Dominique Devriese and Steven Keuchel. You will join a small, supportive, and collaborative team in which you can acquire state-of-the-art know-how and expertise in mechanized verification and build up research and innovation skills for a future career in academia or in industry.

You will be part of the larger division of DistriNet which has a strong and broad focus on cybersecurity, covering in particular subjects such as secure languages, secure compilation and software and hardware verification. This naturally exposes you to related fields and potential applications of your research. DistriNet offers an international and multicultural working environment in the vibrant city of Leuven. (More information on working conditions.)

Interested?

You can apply for this job no later than March 16th, 2025, by sending email to both Dominique Devriese and Steven Keuchel (firstname.lastname@kuleuven.be). Please submit (1) a concise motivation letter, (2) an academic curriculum vitae, (3) a transcript of records of your previous degrees, and (4) an academic reference (name, function, and e-mail address). If you have any questions, please do not hesitate to contact us by email as well.

Steven Keuchel

Dominique Devriese