Identifying Security Critical Properties for the Dynamic Verification of a Processor

Submitted by Katie Dey on Tue, 04/11/2017 - 6:36pm. Contributor: Rui Zhang

ABSTRACT

Hardware companies conduct extensive testing and verification throughout the design phase, yet errata in the design persist to the final shipped product. Just as is the case with software, bugs in the hardware can create vulnerabilities that are exploitable by malicious software. The dynamic verification, post deployment, of security properties of a processor has been shown to be an effective way to prevent exploits of vulnerabilities that may exist within the processor [2]. However, the question of how to identify the security properties to verify remains an open one, with existing work relying on a manual process and human expertise to come up with a small set of security-critical properties [1, 2].

In this talk, I will discuss our research toward answering the question: what are the hardware properties that are critical to security? I will present a new methodology for semi-automatically identifying which security-critical properties to assert for the dynamic verification of a processor.

To start, we automatically generate a set of processor invariants using a modified version of Daikon, an invariant detection tool. This initial set of invariants will contain both invariants that matter to security and invariants that are mostly for functional correctness. We then bring a human in the loop to help with finding the security critical invariants (SCI). But rather than asking the human to go through these invariants one by one, an arduous task unlikely to produce optimal results given the 80,000+ invariants in the set and the architecture-specific details that characterize the invariants, we instead rely on the human to classify high-level processor errata as either security critical or purely functional. We observe that security-critical errata are vulnerabilities precisely because they violate some underlying security property, and we can use the errata to identify security-critical invariants: those that get violated when a security-critical erratum is triggered. This process produces a set of invariants that will defend against previously seen classes of security bugs. We then use machine learning techniques to identify additional invariants as security critical. The added invariants have the potential to defend against new classes of security bugs.

We build a tool chain that implements the approach and evaluate it for the open source OR1200 RISC processor. We find that our tool can identify 19 (86.4%) of the 22 manually crafted security-critical properties from prior work and generates 3 new security properties not covered in prior work [1, 2]. This work was published in the proceedings of the Architectural Support for Programming Languages and Operating Systems (ASPLOS) 2017 [3].

References:

Rui Zhang is a Ph.D. student at the University of North Carolina at Chapel Hill, where she is advised by Prof. Cynthia Sturton. Rui’s research focuses on hardware security and formal verification. Together with
her colleagues at UNC, she worked on developing a semi-automatic method to identify security properties for the dynamic verification of a processor. Previously, she got her bachelor's degree from Peking University in 2013, and her master's degree from Columbia University in 2015.

Rui Zhang

License: Creative Commons 2.5

Other available formats:

Identifying Security Critical Properties for the Dynamic Verification of a Processor

Switch to normal viewer
Switch to experimental viewer

HCSS 2017