Selected topics in Low-Level Systems Verification

Hamed Nemati

March 13, 2019

1 Course Description

One of the major challenges within computer security is how to develop system software like operating system kernels, hypervisors, and microkernels that are secure, and preferably provably so, but at the same time capable of fully exploiting the performance capabilities of modern hardware. Over the last decades, formal verification has emerged as a powerful tool to provide enhanced trustworthiness to such low-level systems. Formal verification provides strong guarantees backed by mathematical proofs that all behaviors of a system meet some logical specification.

In this course, we consider the verification of system software including their modelling on different levels of abstraction and the specification of properties that are to be verified. The goal of the course is to help students to understand challenges involved in verifying low-level systems and familiarize them with fundamental concepts and advanced formal developments used for the verification of large-scale system softwares.

2 Requirements

Students are required to have basic understanding of the formal methods and verification techniques.

3 Organization

Participants obtain a dedicated topic and are expected to

- Get acquainted with the individual topic on their own
- Write an overview article: approx. one or two pages,
- Hand in the article (in PDF) 2 days before the talk
- Give a presentation: 45 min, additional 15 min are reserved for discussion
- Hand in complete presentation material (slides) 2 days before the talk
- Attend all of the presentations

4 Topics

List of available topics:

- Fundamental concepts

  - Noninterference: Joseph A. Goguen and José Meseguer. Security policies and security models.
  - Unwinding theorem: J. A. Goguen and J. Meseguer. Unwinding and inference control.
  - Refinement: Heiko Mantel. Preserving information flow properties under refinement.
  - Compositional verification: Heiko Mantel. On the Composition of Secure Systems.
  - (Bi)Simulation: Colin Stirling. Modal and Temporal Properties of Processes (Chapter 3).
  - Rely and Guarantee Reasoning (multiple references)
  - Separation Logic (multiple references)

- Binary Verification
  - Magnus O. Myreen, Michael J. C. Gordon, and Konrad Slind. Machine-code verification for multiple architectures - an application of decompilation into logic.
  - Mads Dam, Roberto Guanciale, Hamed Nemati. Machine code verification of a tiny ARM hypervisor.
  - Musard Balliu, Mads Dam, Roberto Guanciale. Automating Information Flow Analysis of Low Level Code.
  - Yuting Wang, Pierre Wilke, and Zhong Shao. An Abstract Stack Based Approach to Verified Compositional Compilation to Machine Code.

- Higher Level Verification Approaches
  - Toby Murray, et al. seL4: from General Purpose to a Proof of Information Flow Enforcement (and Noninterference for Operating System Kernels).
  - E. Alkassar, et al. Automated verification of a small hypervisor.
  - Eyad Alkassar, et al. Pervasive Verification of an OS Microkernel - Inline Assembly, Memory Consumption, Concurrent Devices.
  - Mads Dam et al. Formal verification of information flow security for a simple arm-based separation kernel.
  - Roberto Guanciale, et al. Provably secure memory isolation for Linux on ARM.
  - David Costanzo, Zhong Shao, and Ronghui Gu. End-to-End Verification of Information-Flow Security for C and Assembly Programs.

  - Hao Chen, et al. Toward Compositional Verification of Interruptible OS Kernels and Device Drivers.
  - Ronghui Gu, et al. Certified Concurrent Abstraction Layers.
  - Alexander Vaynberg and Zhong Shao. Compositional Verification of a Baby Virtual Memory Manager.

- Related Topics
  - Hamed Nemati, et al. Formal Verification of Integrity-Preserving Countermeasures Against Cache Storage Side-Channels.
  - David Costanzo and Zhong Shao. A Case for Behavior-Preserving Actions in Separation Logic.