Proving Trace Properties of Programs that Execute Adversary-supplied Code

Limin Jia
Assistant Research Professor at CMU ECE & INI

Abstract:
I will present a type-theory, for proving trace properties of programs that may execute adversary-supplied code. Our type system extends Hoare Type Theory (HTT) with post conditions that specify properties of the entire execution trace, not just the initial and final heaps. We also add invariant post conditions, which hold during the execution of a computation even if evaluation gets stuck or diverges. These invariants directly represent safety properties. Our type system contains two novel rules that we find useful in proving safety properties when executed code potentially comes from an adversary. First, we include a beta-conversion rule, which ascribes a term the type of any beta-equal term. The rule is useful when we discover within a proof that a potentially adversarial program is actually equal to another program that we trust. Second, building on prior work in the first-order setting, we develop a rule that ascribes an effect type to a computation variable whose instantiation is unknown, based on knowledge of the context in which the instantiated variable is executed. This is useful when untrusted code is sandboxed prior to execution. We build a step-indexed trace-based semantic model for our types, and prove the soundness of our assertion logic. We illustrate the expressiveness of our type system by proving memory integrity of a hypervisor design. The proof leverages both new typing rules.

Biography:
Limin Jia is an Assistant Research Professor at CMU ECE&INI. She received her B.E. in Computer Science and Engineering from the University of Science and Technology of China and her Ph.D. in Computer Science from Princeton University. Her research interests include language-based security, programming languages, logic, and program verification. Limin’s research focuses on formal aspects of security. She is particularly interested in applying language-based security techniques as well as formal logic to model and verify security properties of software systems.

Tuesday, November 19, 2013
Gates Hillman Center 6501
3:30 PM – 4:30 PM