Guiding distributed systems synthesis with language-based security policies

Andrew Myers

Cornell University, USA

Abstract. The distributed information systems we use every day are becoming more complex and interconnected. Can we trust them with our information? Currently there is no good way to check that distributed software uses information securely, even if we have the source code. Many mechanisms are available, but are error-prone: for example, encryption, various cryptographic protocols, access control, and replication. But it is hard to know when we are using these mechanisms in a way that correctly enforces application security requirements.

This talk describes a higher-level approach to programming secure systems. Instead of using security mechanisms directly, the programming language incorporates explicit security policies specifying the confidentiality, integrity, and availability of information. The compiler then automatically transforms the source code to run securely on the available host machines, and uses a variety of security mechanisms in order to satisfy security policies. The result is systems that are secure by construction. We look at two applications of this approach: building secure web applications using partitioning beween clients and servers, and building more general secure systems by synthesizing fault-tolerance protocols for availability.

Joint work with Steve Chong, Jed Liu, Nate Nystrom, Xin Qi, K. Vikram, Steve Zdancewic, Lantian Zheng, and Xin Zheng.