Distributing and Managing Mobile Code (Abstract)

Michael Franz

University of California, Irvine

Using mobile code is fraught with risks. If an adversary deceives us into executing a malicious program, this may have catastrophic consequences and may lead to a loss of confidentiality, loss of information integrity, loss of the information itself, or a combination of these outcomes. Unfortunately, additional provisions for security frequently cause a loss of efficiency, often to the extent of making an otherwise virtuous security scheme unusable for all but "toy" programs. Moreover, current mobile-code distribution models are surprisingly primitive; for example, they assume that all constituent parts that make up a mobile program are downloaded to a single location, then verified, linked, possibly dynamically compiled, and finally executed at that same location.

Our research in the past 6 years has focused on making mobile code *practical*, to the extent that it can completely displace native code for the whole spectrum of applications from PDAs to desktop applications to scientific computing on supercomputers.

Our contributions include work on dynamic compilation, which raises the performance of such programs, often to the extent of outperforming statically compiled code.

A second major contribution is a class of representations for target-machine independent mobile programs that can provably encode only legal programs. Hence, there is no way an adversary can substitute a malicious program that can corrupt its host computer system: Every well-formed mobile program that is expressible in our encoding is guaranteed to map back to a source program that is deemed legal in the original source context, and mobile programs that are not well-formed can be rejected trivially. Further, our encoding not only guarantees referential integrity and type-safety within a single distribution module, but it also enforces these properties across compilation-unit boundaries. As a side-effect, our encoding is exceptionally dense: it outperforms the next best compression scheme for Java by almost 50

A third focus of research is on managing the mobile-code pipeline from code producer to code consumer: capturing all meaningful modes of mobile-code deployment in a model, and then creating a code management architecture that makes this model enforceable by mechanical means.