Secure Class Initialization with Dependencies

Willard Rafnsson Andrei Sabelfeld Keiko Nakata

May 15, 2011

Abstract

Language-based information-flow security is concerned with specifying and enforcing security policies for information flow via language constructs. Although much progress has been made on understanding information flow in object-oriented programs, little attention has been given to the impact of class initialization on information flow. This paper turns the spotlight on security implications of class initialization. Previously, Nakata and Sabelfeld revealed the subtleties of information propagation when classes are initialized, showed how these flows can be exploited to leak information through error recovery, and illustrated how to track these flows by means of a type-and-effect system (Nakata and Sabelfeld, IFIPTM'10). However, the enforcement mechanism provided disallows initialization of classes in secret contexts altogether, disregards class dependencies, and uses a fixed lattice of security levels, making it ill-suited for use in a full-fledged security-typed programming language like Jif. In this paper, we present a type-and-effect systems which remedies these shortcomings. The enforcement is parameterized by an arbitrary lattice of security levels. Flows through the class hierarchy and dependencies in field initializers are tracked by typing class initializers wherever they could be executed. The contexts in which each class can be initialized is tracked to prevent secret contextual information from flowing out of its scope through class initialization statuses and error recovery. We show that the type system enforces a security property which is a termination-insensitive notion of noninterference.