Abstract of the PhD research

Web applications, in support of our daily tasks, are provided with sensitive information such as banking accounts numbers, social security information, etc. Therefore, it is expected that the developers of such applications rely on adequate tools offered by JavaScript and browsers to help them develop secure applications. However, neither JavaScript nor browser security mechanisms fully address modern application security needs.

Designing a security mechanism supporting the combination of features such as portability, performance and many awkward features of JavaScript and browsers is still problematic. Furthermore, in the software development life-cycle it is important to assess the same set of access control and information flow policies during development (static) and production (dynamic). However, the current state of the art does not allow a safe and efficient combination of static and dynamic enforcement of a shared set of security policies, forcing developers to reimplement and maintain the same policies and their enforcement code in both static and dynamic environments.

This thesis explores language-based access control and information flow control policies for securing client-side web applications.

First, we present Guardia, a framework for declaratively specifying and dynamically enforcing application-level security policies for JavaScript web applications without requiring VM modifications. Second, we present Gifc, a permissive-upgrade-based inline monitoring mechanism to detect unwanted information flow in client-side web applications. Based on Guardia and Gifc, we develop a novel technique for deriving Static Application Security Testing (SAST) from an existing Runtime Application Security Protection (RASP) mechanism by means of a two-phase abstract interpretation approach. In our approach, the SAST component avoids duplicating the effort of specifying security policies and implementing their semantics. Deriving a SAST component from a RASP mechanism ensures equivalent semantics for the security policies across the static and dynamic contexts in which policies are verified during the software development lifecycle.