Information-Flow Security

Andrei Sabelfeld
Chalmers University of Technology, Gothenburg, Sweden

A vast majority of todays attacks are application-level attacks. According to the SANS (SysAdmin, Audit, Network, Security) Institute, ättacks against web applications constitute more than 60% of the total attack attempts observed on the Internetäs of September 2009. Application-level attacks are particularly dangerous because they circumvent the standard low-level protection mechanisms (such as access control). Furthermore, application-level attacks are easier to create (or simply download and launch) exactly because of their high-level nature. In addition, these attacks often exploit vulnerabilities introduced by software developers (as opposed to cryptographic protocol designers and system administrators), which is often an easier target.

It is our firm belief that attacks will continue succeeding unless a fundamental security solution, one that focuses on the security of the actual applications (code), is devised. To this end, we are convinced that application-level security can be best enforced, by construction, at the level of programming languages. Indeed, language-based approach to security gains increasing popularity because it provides natural means for specifying and enforcing application and language-level security policies. Popular highlights include Java stack inspection, to enforce stack-based access control, Java bytecode verification, to verify bytecode type safety, and web language-based mechanisms such as Caja, ADsafe, and FBJS, to enforce sandboxing and separation by program transformation and language subsets.

In these lectures, we overview the state of the art in language-based information-flow security [1], particularly focusing on information release, or *declassification* [2], policies and on trade-offs between *static* and *dynamic* techniques to enforce information-flow policies.

Inspired by the success of Caja, ADsafe, and FBJS, we turn our attention to the fundamental challenges and opportunities in web application security. Web applications are popular because they can be easily accessed from any location, as long as there is Internet access and a browser. However, the other side of the web application coin is increased security risks: sensitive information is spread between a web server and a web client, and both must be protected along with the communication link between them. The problem is exacerbated by web services such as web mashups that allow integrating content of several websites into one webpage, where code embedded from the separate websites is shared without security guarantees.

We discuss a principled approach to web application security through tracking information flow. Although the agile nature of developments in web application technology makes web application security much of a moving target, we show that there are some fundamental challenges and tradeoffs that determine possibilities and limitations of automatically securing web applications. We address challenges related to mutual distrust on the policy side (as in web mashups) and tracking information flow in dynamic web programming languages (such as JavaScript) to provide a foundation for practical web application security.

References

[1] A. Sabelfeld, A. C. Myers. *Language-based Information-flow Security.* IEEE J. Selected Areas in Communications, 21(1), pp. 5-19, 2003. `http://www.cse.chalmers.se/%7Eandrei/jsac.pdf`

[2] A. Sabelfeld, D. Sands. *Declassification: Dimensions and Principles.* J. Computer Security, 17(5), pp. 517-548, 2009.
`http://www.cse.chalmers.se/%7Eandrei/sabelfeld-sands-jcs07.pdf`