Language-based Control for Information Flow and Release

Andrei Sabelfeld
Chalmers University of Technology, Gothenburg, Sweden

Current standard security practices do not provide substantial assurance that the end-to-end behavior of a computing system satisfies important security policies such as confidentiality. An end-to-end confidentiality policy might assert that secret input data cannot be inferred by an attacker through the attacker's observations of system output; this policy regulates *information flow*.

Conventional security mechanisms such as access control and encryption do not directly address the enforcement of information-flow policies. Recently, a promising new approach has been developed: the use of programming-language techniques for specifying and enforcing information-flow policies.

In these lectures, we overview the state of the art in language-based information-flow security [2], particularly focusing on information release, or *declassification* [3], policies and on trade-offs between *static* and *dynamic* techniques to enforce information-flow policies.

The area of declassification is motivated by the need for computing systems to deliberately release (or declassify) sensitive information. For example, releasing the average salary from a secret database of salaries is sometimes needed for statistical purposes. Another example of deliberate information release is information purchase. An information purchase protocol reveals the secret information once a condition (such as "payment transferred") has been fulfilled. Yet another example is a password checking program that leaks some information about the password. Some information is released even if a log-in attempt fails: the attacker learns that the attempted sequence is *not* the same as the password.

Information release is a necessity in these scenarios. A principal security concern for systems permitting information release is whether this release is safe: is it possible that the attacker compromises the information release mechanism and extracts more secret information than intended? We provide a road map of the main directions of current research, by classifying the basic goals according to *what* information is released, *who* releases information, *where* in the system information is released, and *when* information can be released. We apply this classification in order to evaluate the security of a case study realized in a security-typed language: an implementation of a non-trivial cryptographic protocol that allows playing online poker without a trusted third party [1]. In addition, we identify some *principles* of declassification. These principles shed light on existing definitions and may also serve as useful "sanity checks" for emerging models.

References

1. A. Askarov, A. Sabelfeld. *Security-typed Languages for Implementation of Cryptographic Proto- cols: A Case Study*. Proc. "European Symp. on Research in Computer Security", LNCS 3679, pp. 197–221, Springer-Verlag, 2005.
   Available at `http://www.cs.chalmers.se/~andrei/askarov-sabelfeld-esorics05.pdf`
2. A. Sabelfeld, A. C. Myers. *Language-based Information-Flow Security*. IEEE J. Selected Areas in Communications, 21(1), pp. 5–19, 2003.
   Available at `http://www.cs.chalmers.se/~andrei/jsac.pdf`
3. A. Sabelfeld, D. Sands. *Declassification: Dimensions and Principles*. J. Computer Security, 2007.
   Available at `http://www.cs.chalmers.se/~andrei/sabelfeld-sands-jcs07.pdf`