CS 252 Project Proposal

Andrew Kalenda*[1]

[1]Department of Computer Science, San José State University

Faceted values are a cyber-security technique that provide different views of data depending on the observer, ensuring privacy where needed and protecting data from mischievous third-party programs. Currently, more compelling examples of this technique are needed; to this end, a Haskell faceted values library has been created. The intent of this project is to give a concrete implementation of the library and demonstrate faceted values' efficacy in a strongly, statically typed language.

1 Introduction

Information flow controls[4] are cyber-security mechanisms by which in-the-wild programs may confine classified data to safe channels, so that sensitive information cannot leak. (In particular, cannot leak to third-party programs.) Historically, information flow control has been a consideration of the individual software developer, but tracking the flow of information is (especially in the untamed wilds of web development) prohibitively difficult and tedious. Even a mindful developer is hard-pressed to allocate precious time and energy when other concerns are mounting. Thus the birth of programmatic controls that will relieve the burden.

Secure multi-execution[5–7] is one such control mechanism. It splits program execution into two paths: *high* and *low*. A datum's life begins on the *low* path. On *low*, data may be written to any output, public or private. However, when a datum becomes determined by classified information, it is permanently elevated to the *high* path. This elevation can either occur as the datum is determined directly (an *explicit flow* of information) or indirectly (an *implicit flow* of information). The *high* execution path has its output restricted to authorized channels, thus preserving the sanctity of data.

Another mechanism, *faceted values*[2], simulates secure multi-execution in a single process. A faceted value is a monad containing a private and public value - the facets - which express respective views of a datum depending on the observer. To an unprivileged accessor, the true (private) value is obscured; thus, *high* and *low* execution paths are largely unnecessary.

Compelling examples of faceted values in action are needed, particularly in a strongly and statically typed language. To this end, a Haskell library has been created.[3] It is the purpose of this project to put said library into action and demonstrate (un)classified data as viewed from authorized and unauthorized perspectives.

2 Information Flow Controls

Pending

*andrew.kalenda@sjsu.edu

3 Secure multi-execution

Pending

4 Faceted values

Pending

5 *Haskell-Faceted* Library

Pending

6 Tentative Work Schedule

Date	Tasks accomplished
10/12	Read all papers, include personal digests in project description
10/19	Basic Haskell use case, command line, prompts for (non)sensitive info, then prompts for (un)privileged user login, displays view of data
10/26	Get Haskell web framework serving pages
11/02	Add authentication layer
11/09	Web form equivalent of week of 10/19
11/16	Faceted values in persistence database
11/23	TBD
11/30	TBD
12/07	TBD

References

[1] Austin, T. H., & Flanagan, C. (2009). *Efficient purely-dynamic information flow analysis.* ACM Sigplan Notices, 44(8), 20-31.

[2] Austin, T. H., & Flanagan, C. (2012). *Multiple facets for dynamic information flow.* ACM SIGPLAN Notices, 47(1), 165-178.

[3] Austin T.H., Knowles K. & Flanagan C. (2014). *Typed Faceted Values for Secure Information Flow in Haskell.* ???[1].

[4] Denning, D. E., & Denning, P. J. (1977). *Certification of programs for secure information flow.* Communications of the ACM, 20(7), 504-513.

[5] Devriese, D., & Piessens, F. (2010, May). *Noninterference through secure multi-execution.* In Security and Privacy (SP), 2010 IEEE Symposium on (pp. 109-124). IEEE.

[6] Jaskelioff, M., & Russo, A. (2012). *Secure multi-execution in haskell.* In Perspectives of Systems Informatics (pp. 170-178). Springer Berlin Heidelberg.

[1] TODO: Find out what to put in here!

[7] Rafnsson, W., & Sabelfeld, A. (2013, June). *Secure multi-execution: fine-grained, declassification-aware, and transparent.* In Computer Security Foundations Symposium (CSF), 2013 IEEE 26th (pp. 33-48). IEEE.