API Usability and Security

Our goal is to develop and empirically test *concrete* and *actionable* API design principles that lead to more secure code.

APIs are the boundaries between system components, defining how they interact. Programmers failing to commonly understand how an API should be used causes failures.

Project Principles:

- Secure development practices should be empirically tested and validated.
- Programmers and designers are people too.

Why APIs?

- Large impact on system security
- Long-lasting
- Designed by a small number of more-experienced people

Initial Focus:

- State management: how state of objects defined by API can be changed

Methodology

1. Semi-structured Interviews with developers
2. Prototypes addressing issues
3. Evaluation with user studies

Co-funded by NSF award 1423054

SEI participants:
Forrest Shull, Robert Seacord, David Keaton

CMU participants:
Dr. Brad Myers, Dr. Jonathan Aldrich, Dr Joshua Sunshine, Michael Coblenz

Summer students:
Sophie Gairo, Paul Peng

Programmers use APIs to get things done, but must understand each other's roles and responsibilities.

Semi-structured Interviews

Interviewed experienced programmers: at least seven and a mean of 15 years of experience, most with DoD-relevant projects

Results include:

- Controlling where/when state is changed and by whom is a serious problem
- Programmers do use concepts like immutability (data structures that cannot be changed after being created)
- Language features, like `const`, don't satisfy programmer's needs.

Issues:

- Object vs class immutability
- Abstract vs concrete state
- Viral nature of C++ `const`, ...

Question: What fraction of bugs are the result of [unexpected] state changing?

Answer:

"Oh, gosh, like, most of them!"

Prototypes

Designed three Java language extensions to address common use cases raised by developers.

- Support for transitively immutable objects, non-transitive immutable objects, and objects that are mutably only during construction phase
- Two currently implemented, one in progress

Evaluation with user studies

Use participatory design techniques. Give developers tasks and elicit how they would solve them before introducing extensions we are testing. Have developers then use features (or standard Java), eliciting their thoughts. Evaluation based upon bugs, developer speed, effectiveness, and developer feedback.

- Pilot studies in progress

Publications:

- *"Empirical Evaluation of API Usability and Security,"* LAW workshop, associated with ACSAC '14
- *"Comparing Transitive to Intransitive Object Immutability"* accepted at PLATEAU workshop, associated with SPLASH '15
- *"A Course-Based Usability Analysis of Cilk Plus and OpenMP"* accepted at VL/HCC '15 conference
- *"Exploring Language Support for Immutability"* submitted to ICSE '16

Poor state management in API design is a serious problem for developers. Language features to assist designers and programmers will improve both system security and usability.

Contact: Sam Weber samweber@cert.org

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0002941