Milestone 7

Changes

There have been no changes since my last update.

Accomplishments Since Last Update

Since my last update, Jean and I have written an abstract for meeting of the minds that will also serve as a building block for the abstract of our final paper for Binah. Here is that abstract:

Time has proven that programmers are not good at thinking about information leaks. This is not a surprising fact: in modern applications, information flows through potentially many different pieces of functionality, each with their own security permissions, before finally being displayed to a user. Database interactions make this problem even harder, as programmers now have to reason about policy enforcement across the different semantics of application code and database queries. It has become unrealistic to assume that a programmer will be able to keep track of all the policies whenever the program interacts with sensitive information. At a basic level, we want machine-checkable proofs that our programs do not leak information. To make it easier for programmers to write these programs in the first place, we propose going one step further and *repairing* programs to prevent leaks.

In this poster, we introduce Binah, a web framework that takes responsibility for managing information flow policies across the application and database. Binah supports a policy-agnostic programming model, where policies are attached directly to sensitive values instead of littered throughout the code. Not only can Binah statically verify that there are no information leaks across the semantics of both the application and the database, but it can also automatically and statically repair programs too, with provable soundness guarantees. This allows the programmer to centralize all of the security policies for their application into one place, making it easier for them to focus on the core application logic, and making it easier to audit and modify their security policies. We have built Binah in Haskell, and we use Liquid types-- refinement types with expressive predicates over program values that can be decidably checked-- to allow its users to express policies. We have built a Haskell conference management system to run SIGBOVIK that we are now verifying in Binah with respect to expressive properties about the database interactions.

Meeting Milestones

N/A, Milestones reached. Jean was out of town this week so we will make new ones next week.

Looking Ahead

Over the next two weeks, I will be focusing on squeezing whatever inference we can get out of Liquid Haskell. Hopefully I'll be able to contribute to policy inference on the Liquid Haskell side, too!

Resources Needed

I require no additional resources to continue working on my project.