HACKAR: Helpful Advice for Code Knowledge and Attack Resilience

SIFT Smart Information Flow Technologies

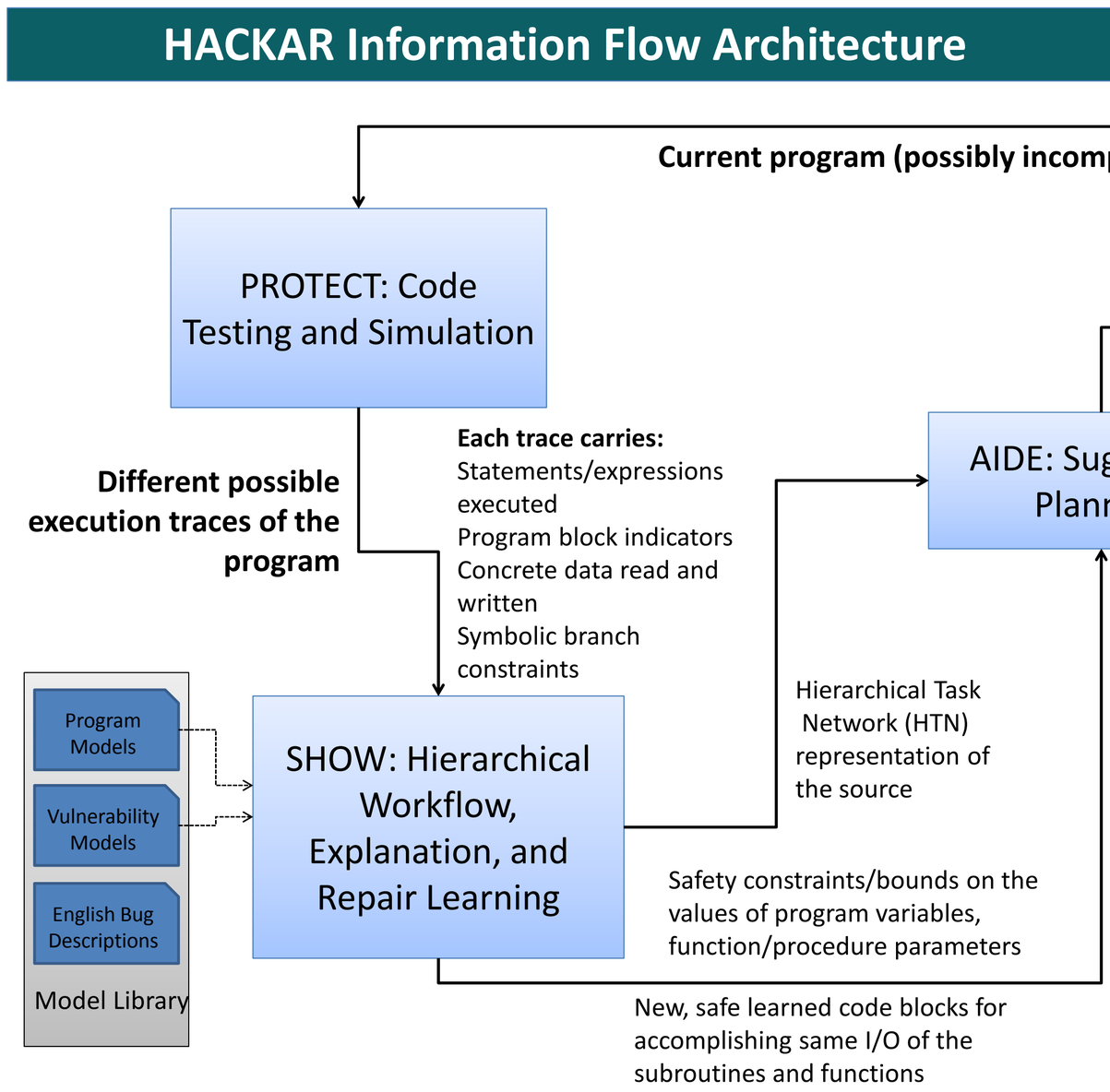
Ugur Kuter, Mark Burstein
UKuter@sift.net, Burstein@sift.net

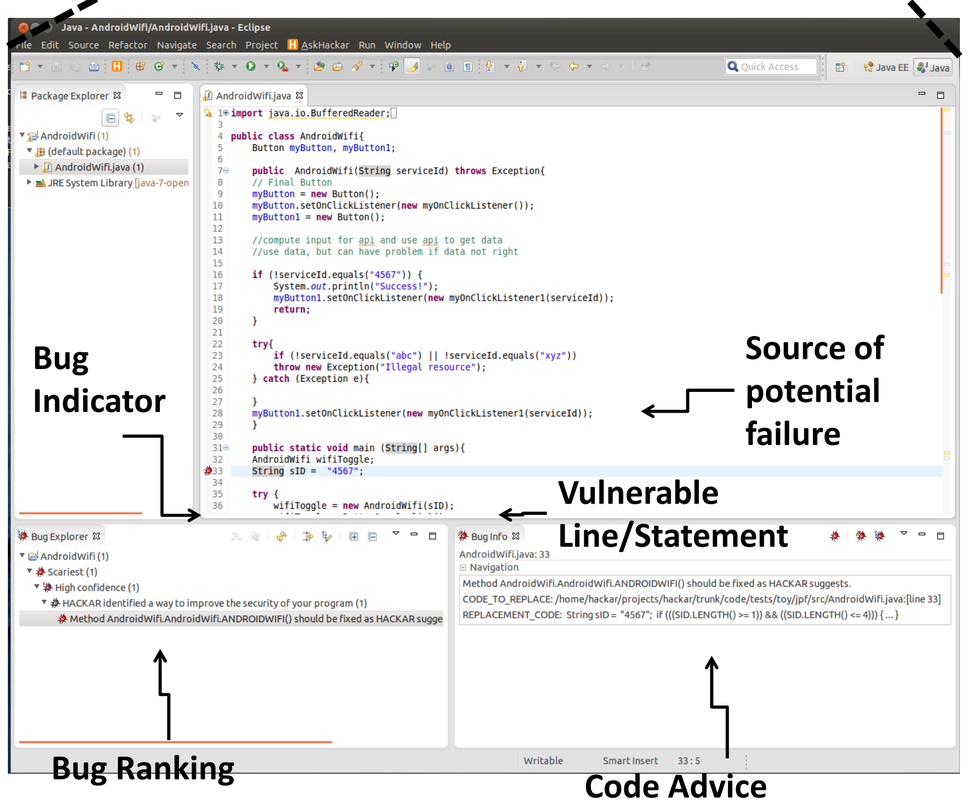
Proactive code-analysis to find program inputs that lead to security vulnerabilities and techniques for learning program workflows and generating suggestions to repair vulnerabilities as code is developed.

HACKAR Technical Capabilities and Innovations

- Identifies vulnerabilities early in development, allowing the code to be repaired most easily.
- Provides Just-In-Time suggested fixes for improving code and training software engineers
- Provides safeguards against vulnerabilities due to dynamically-changing, ill-documented or untrusted sources of input to remote functions even if remote functions are opaque or their requirements are unavailable.
- New test input generation techniques based on concolic testing and grammar-based reasoning, sidestep computational problems of path-explosion in symbolic execution, where branching and looping could explode the space of symbolic tests that need to be done.
- Structural learning and knowledge capture of program data flows, advancing state-of-the-art AI learning, works to learn arbitrary recursive, looping, and branching structures.
- Multi-suggestion generation over program hierarchies provides real-time analysis experience to the programmer.
- HACKAR and **Eclipse and Emacs integration** provides IDEs on two of the most popular code development environments

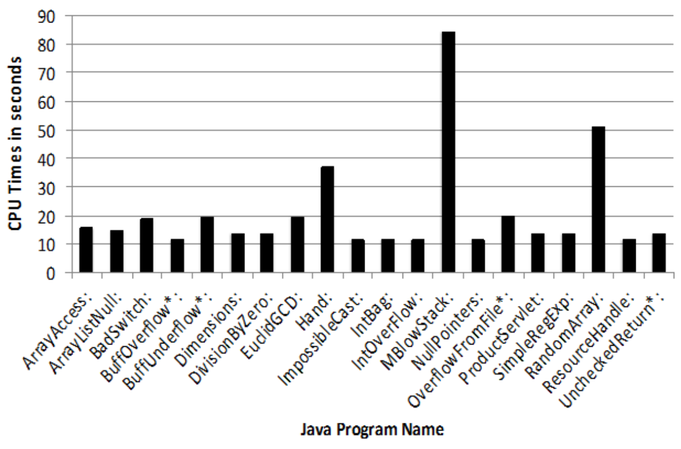
HACKAR Information Flow Architecture

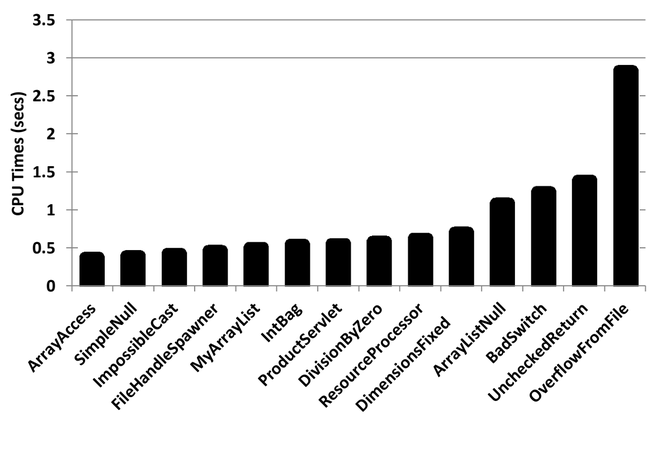




Evaluation and Validation

- Generate correct suggestions for 95% of programs, with each vulnerability analyzed, learned, and a fix is proposed under 2 seconds
- **Coverage 100%** in our experimental suites
- For our **scalability,** we can generate a suggestion **1 to 1.5 seconds on average,** for programs that range from 25-30 lines of code to over 100 LoC
- More experiments and results can be found in **Kuter et al. 2015. Proceedings of the Innovative Applications of Artificial Intelligence (IAAI-15).**





Next Steps

- Ensuring program operationality and well-being not just as a stand-alone artifact but In complex execution ecosystems
- Vulnerability analysis mobile (e.g., Android) applications and services
- HACKAR for binary code transformation and analysis

HACKAR is funded by the Office of Naval Research