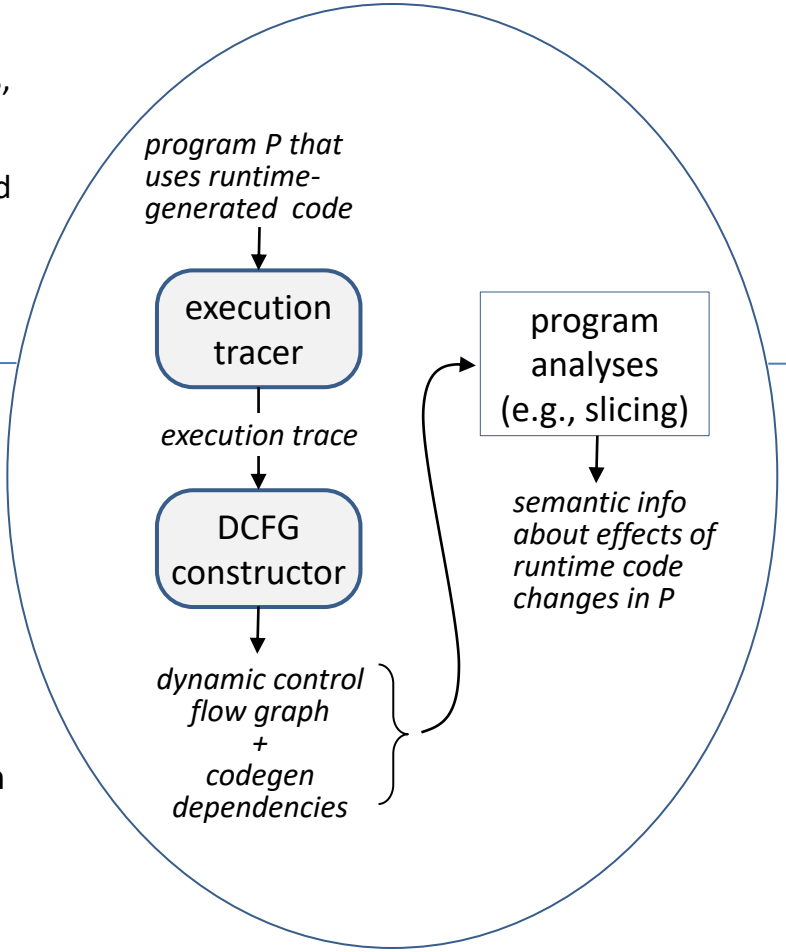
Reasoning about Dependencies and Information Flow in Dynamic Code

Challenge:

- Runtime-generated code ubiquitous (e.g., JIT compilers, packet filters, ...)
- Traditional program representations do not extend to such code
- This makes reasoning about their semantic and security properties difficult

Solution:

- Use dynamic analysis to trace the runtime behavior of the program
- *Dynamic control flow graphs* (DCFGs): capture effects of runtime code changes
- *Codegen dependencies:* capture information flow in dynamic code



Scientific Impact:

- Provides a rigorous basis for reasoning about information flow in dynamic code, e.g.:
  - analysis of obfuscated code;
  - information leakage in JIT-compiled code

Broader Impact and Broader Participation:

- Improve security and privacy analyses of widely used software, e.g., web browsers
- Undergraduates and women involved in the project
- K-12 outreach to local-area middle- and high-school CS teachers

Award no. 1908313

PI: Saumya Debray
Institution: University of Arizona