SAND2021-8360D

The Center for Cyber Defenders
Expanding computer security knowledge

Characterizing Binary Uncertainty

Characterizing CFG Incompleteness Effects on Analysis Uncertainty

Jack Williams | University of Oklahoma | Computer Science BS/MS | Fall 2021

Manager: Susan Gardner | Project Mentor: Shelley Leger | CCD

- Problem Statement:
  - Analyzing programs (for example, to determine if they have vulnerabilities or could be malicious) is an NP-hard problem often performed by highly trained humans.
  - Automated analyses help in this task, but they can generate uncertain results (combining both under and over approximation).
  - Such analyses often depend on a Control Flow Graph (CFG) of the program's logic, which is often incomplete.
- Objectives and Approach:
  - Focus on static binary analysis.
  - Use Ghidra to decompile binaries with known "ground truth" information available.
  - Randomly and systematically remove instructions and control flow transfers, then analyze how such removals affected the decompilation.
  - Generalize results of the above analysis to determine which sorts of missing control flow transfers may impact automated analysis results.
  - Use various recovery strategies to try to rebuild the CFG after removing edges.
- Results:
  - No results yet. We intend to publish results internally in a SAND report and hope to consider publishing externally.
- Impact and Benefits:
  - Rapid analysis of binaries is necessary for many national security missions, but currently such analysis often takes months or years.
  - Understanding what causes this uncertainty in analysis results and which sections of the program are likely more important to model correctly could aid human analysts and improve trust in automated analyses.
  - Knowing where the CFG is likely complete or incomplete could help to focus analysis.
  - Results could validate (or counter) analyst intuition about which sections of a CFG are both important and often wrong (e.g., switch statements).