Analysis of the Automated Vulnerability Discovery Process

GTRI CIPHER Lab – Shelby Allen, Kennon Bittick, Mike Nawrocki, Noah Tobin

Problem: The demands of software analysis outpace manual analyst capabilities, and automated solutions are not yet sophisticated enough to replace manual analysts.

Background: Vulnerability Discovery

- Manual and automated techniques exist for vulnerability discovery
- Automated analyses scale well and require minimal human input
- Manual analyses are typically expert-driven and time consuming
- Human assisted analyses combine manual and automated analyses
- Effectiveness of each technique varies in complex, unknown ways, depending on the AUT

Solution: Analysis System

- Implemented multiple analyses into a single cyber reasoning system
  - Grey-Box fuzzing
  - Dynamic symbolic execution
  - Directed string construction
  - Directed symbolic execution
  - Directed backwards-slice static analysis
- Implemented Multi-Armed Bandits to choose which analyses to perform at any given time
- Performed testing to determine efficacy of different analyses and other properties

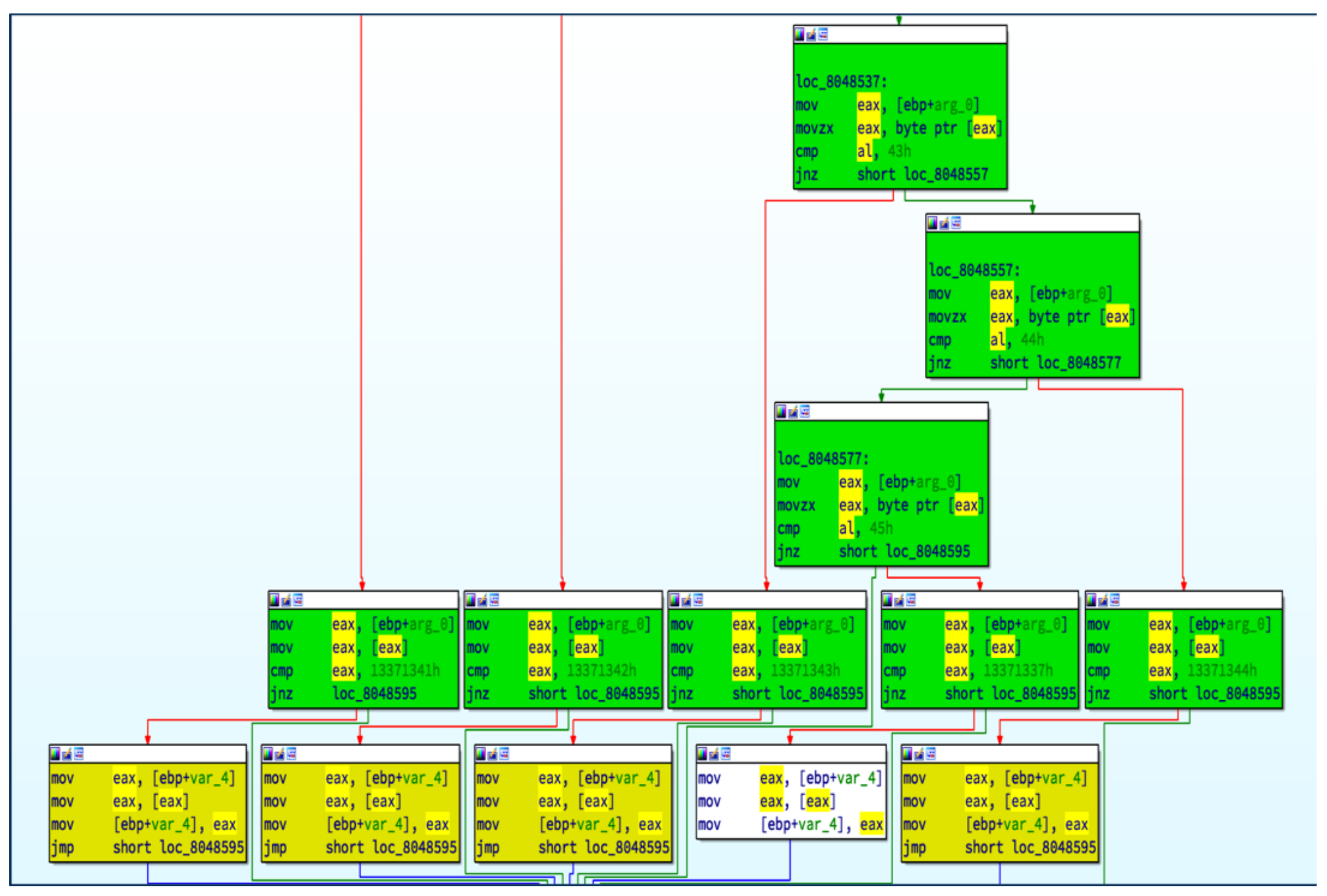
Background: Multi-Armed Bandit ML

- The goal is to minimize “regret”, i.e. the reward lost by choosing one arm over all others
- Split time between exploration (discovering properties of the analyses) and exploitation (attempting to maximize total score)
- UCB1 Algorithm
  - Balance two parts of the equation such that higher rewarding arms are favored, but less frequently chosen arms selected as well due to higher uncertainty factor
- Contextual Bandit (LinUCB)
  - Vector of context information used with linear regression to predict score for each arm
  - Choose technique with highest upper confidence bound

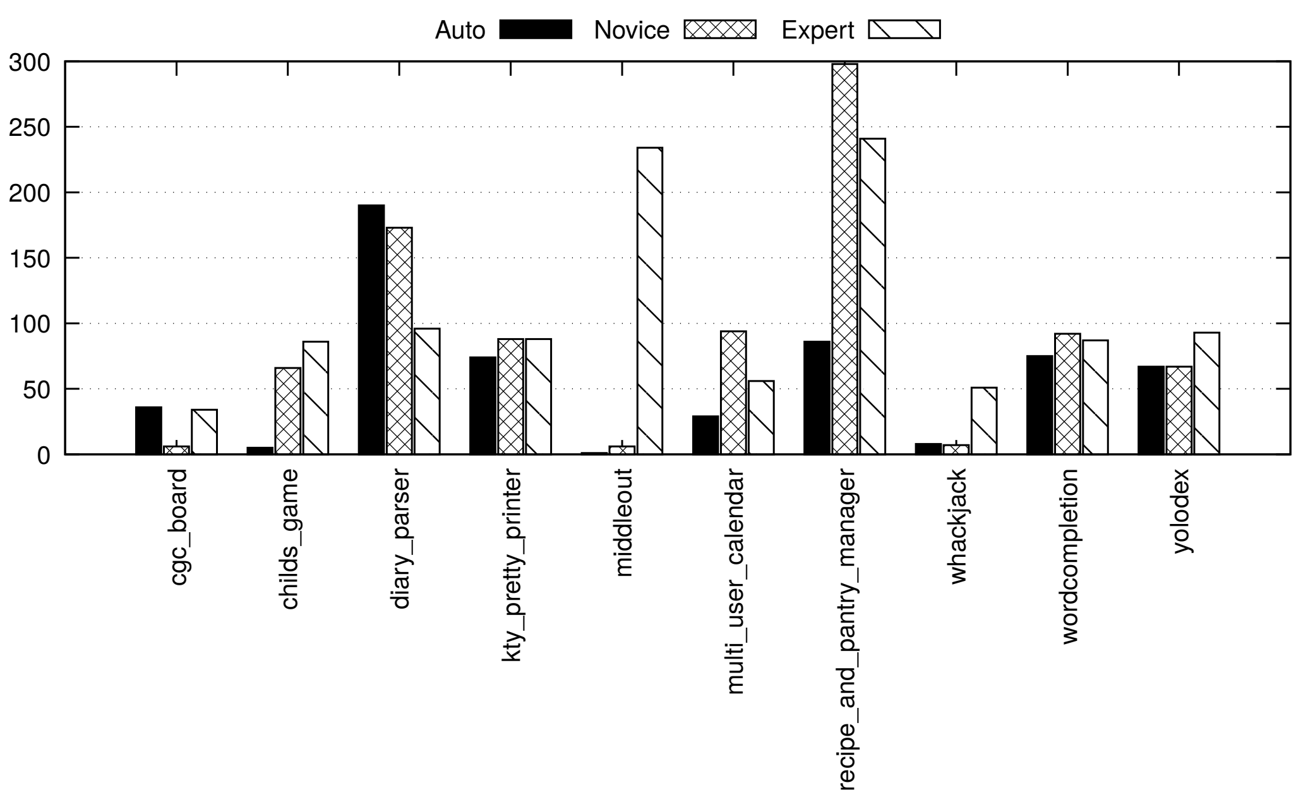
Solution: Testing

- Used Cyber Grand Challenge binaries as training set
- Employed analysts of varying skill to drive human assisted analyses, gathered results of each analysis run
- Conclusions:
  - Human assisted analysis techniques improve analysis efficacy in general
  - Novice analysts take longer than experts but have similar impacts on analysis
  - Directed backwards-slice static analysis and directed string construction tend to reveal large portions of the AUT to the analysis system
  - Symbolic execution evolves as fuzzing progresses and should be run continuously

Human-Directed Symbolic Execution



Expertise Comparison – Total Branches



Future Work:

- Improve analyses, share symbolic state where possible, deduplicate work
- Improve UI with expert feedback
- Run longer tests, evaluate effectiveness of different bandits
- Ease the burden of harnessing new binaries for analysis

C3E
Computational Cybersecurity in Compromised Environments
2018 Fall Workshop | September 17-19 | Atlanta, Georgia