SAND2017-7655C

# The Center for Cyber Defenders

Expanding computer security knowledge

# Symbolic Execution for Vulnerability Discovery

## Evaluating and Extending the Manticore Platform

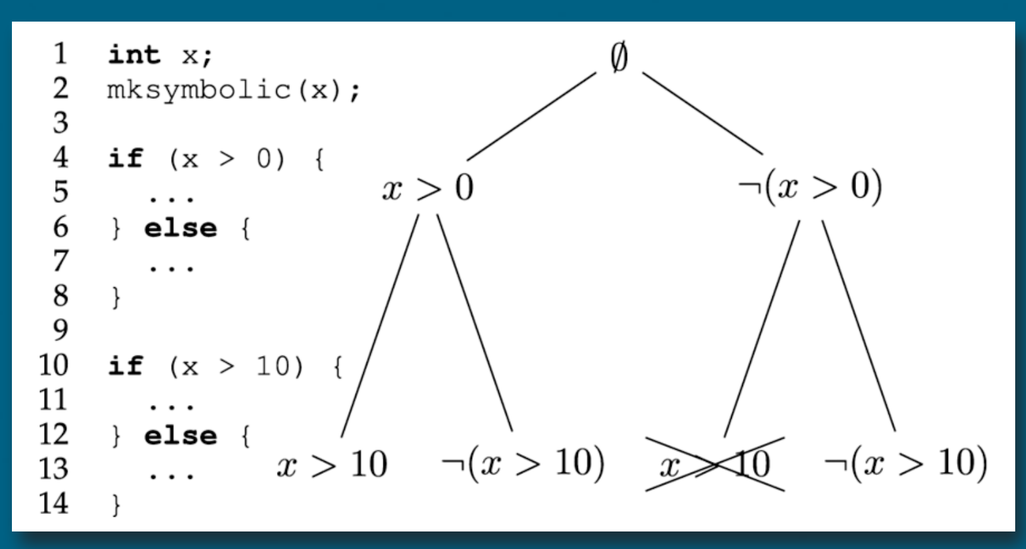
Matt Webb, Kansas State University
Tiemoko Ballo, Carnegie Mellon University

**Project Mentor: Michelle Leger, Org. 5836**

## Problem Statement:

Flaws, or "bugs", in software are constantly introduced despite developers' best efforts. A subset of these flaws are vulnerabilities – weaknesses that may be identified and exploited by attackers. There exist several tools, commercial and academic, that leverage program analysis techniques in an attempt to identify vulnerabilities in an automated fashion. This project aims to evaluate a range of such tools to understand their strengths and shortcomings, identify those that could be built upon with R&D investment, and create solutions for effectively defending critical systems given the prevalence of such vulnerabilities.

We are specifically focused on Manticore, a new and open-source symbolic execution engine that underpinned a Cyber Reasoning System used in DARPA's Cyber Grand Challenge. We aim to understand the tool's internals, test its usability given its current maturity, and explore its practicality for real-world problems.

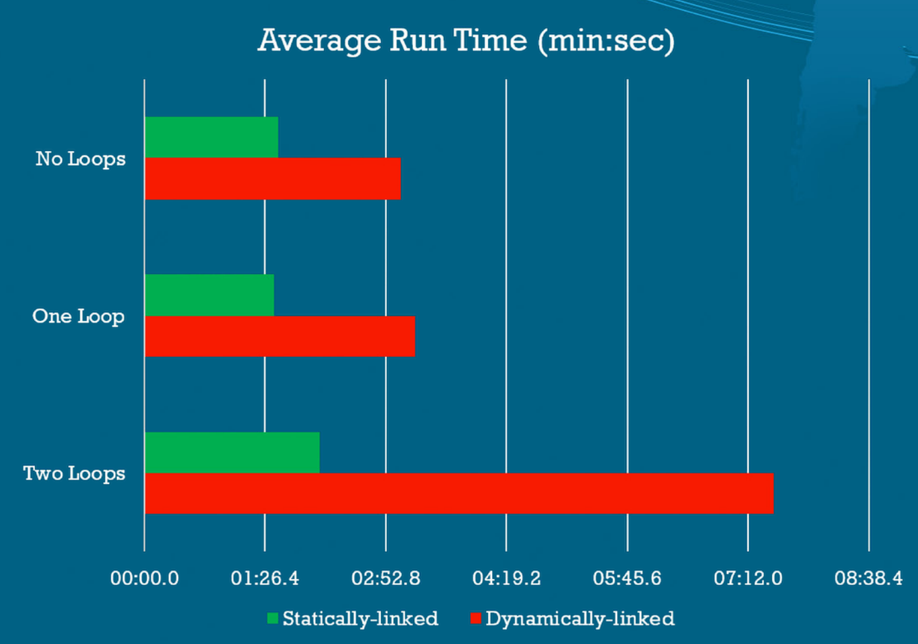



Symbolic Execution Example (Constraint Tracking)
*(Peter Collingbourne, Cristian Cadar, Paul H.J. Kelly, "Symbolic Crosschecking of Data-Parallel Floating-Point Code", August 2014)*

## Objectives and Approach:

- Understand Manticore "under the hood" (components and their functions, dependencies)
- Learn Manticore's API, concepts, and capabilities
- Evaluate Manticore's usability by solving a CTF binary representative of a time-consuming analyst task (reversing bitwise manipulation of a byte array).
- Understand where the tool excels, where it struggles, and where it fails.
- Understand if and how Manticore can interface with other tools.
- Extend the tool's functionality to perform more advanced analysis.
- Compare Manticore against similar solutions, provide a recommendation for/against adoption.

## Results:




Symbolic Solving Performance Measures

- Demonstrated both concrete and symbolic solutions for a CTF binary, documenting the scripts used.
- Identified and documented tools limitations (unsupported syscalls, difficult cases for solver, contextual shortcomings).
- Integrated Manticore with another binary analysis tool (BinaryNinja) to automatically find and prove vulnerabilities: systematically extracting stack buffers and function pointers on the stack that could be corrupted by buffer overflow, then generating concrete inputs that prove the vulnerability.
- Identified vulnerable function or root cause by having script automatically analyze trace of path in proof-of-vulnerability.
- Measured symbolic solving performance for handling more complex control flow (nested loops operating on a symbolic buffer) and linking variations (static vs. dynamic libraries).
- Will attempt to replicate a CVE for a common Linux binary using Manticore.

## Impact and Benefits:

- Gained and documented knowledge on the inner-workings of Manticore, as well as its strengths and weaknesses.
- Integrated tools in proof-of-concept to demonstrate that using them in combination allows more advanced reasoning than either is capable of alone.
- Identified and quantified performance limitations relevant to real-world binaries.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2017-XXXXC.