Week 1 Report

Zachary Porter

2022-08-31

1 Todos

- Understand checkpoint system for rr
- Demo of librr_rs
- Begin thinking about similartity metrics
- Outline usecase

2 Checkpoint system for rr

3 Project Assumptions

- WX circumvented with mprotect will guanratee that code section changes will not happen without an associated syscall.

  I am not sure if this applies with double mapped pages.

- The frontend is a good proxy for what is running in the background. In other words, visual updates are given to the user quickly after code runs in the background (quickly is subjective but probably less than 1s)
- It is not feasible to store every instruction (or even close to it). As such, it must be sufficient to scan and build paths intelligently

4 Use cases / Scenario

4.1 Date Gated Function

4.1.1 Description

Functionality of a program only exists when f(current_time)==true.
Examples of this may include malware or programs that have timing bugs.
Use case would be to identify the check and either understand or manipulate it.

4.1.2 Required Functionality

1. Replay the binary
2. Figure out screen recordings.. xlib terrifies me
3. Trace back where the instruction for manipulating the instruction is stored

4.2 Identify code injection opportunities

Record instances where

5 Data Structures

```
struct RanInstruction{
    address: usize,
    frametime: u32, // W^X Assumption,
    inputs: Vec<DataValue>,
    instruction: DataValue<Instruction>, // todo.
}

enum DataPathNode{
    SYSCALL(name: String), // todo
    FILE(name:String, offset:usize),
    RAN_INSTRUCTION(v: RanInstruction),
    UNEXPANDED,
}
enum DataValue<T> {
    value: T,
    source: DataPathNode,
}

enum Trace{
    BLACKBOX(
        inputs:??
        size:usize, // Number of instructions?
        start: usize,
        end: usize,
        subtraces:Vec<Trace>,
    ),
    EXPANDED(

    )
    LOOP(
    )
    BRANCH(

    )
    UNEXPANDED


}
```