SIFT: A Multithreading Extension to KLEE

Tuba Yavuz, Associate Professor, ECE Department, University of Florida, tuba@ece.ufl.edu

Architecture of SIFT
SIFT is a multithreaded extension of KLEE that performs selective thread scheduling to reveal bug triggering scenarios. It uses dynamic data-flow analysis and light-weight static analysis to identify property relevant interleaving points (IPs). Explicit properties include assertions within the code and memory safety is treated as the implicit property. Check out SIFT at https://github.com/sysrel/SIFT

SIFT Exploration Steps

- **Step 1**: Conservative thread scheduling, i.e., when a thread blocks schedule another IP.
- **Step 2**: New thread scheduling scenarios due to interleaving points in IP1.
- **Step 3**: New thread scheduling scenarios due to interleaving points in IP2.

SIFT on 10 CVE & 10 SVComp Benchmarks

- Error Detection (Random+Coverage Scheduling, # of steps=3)
  - Single: > 0.07s, < 19.96s
  - Common: > 0.07s, < 52.09s
  - One: > 0.07s, < 24.62s

- Error Detection (Depth-First Search Scheduling, # of steps=3)
  - Single: > 0.07s
  - Common: < 52.09s
  - One: < 24.62s

SIFT can detect the bugs in all 10 CVE benchmarks whereas ConVul misses one.