C2Fuzzer overview

This work in one sentence

- Coverage-directed concurrency fuzzing to spot race conditions

Backgrounds

Data race vs. Race condition

- Data race: unordered accesses to a single location
  - It is a bug because it may confuse a compiler
  - It may or may not cause a real problem
- Race conditions: unintended interleaving causing a failure or a malfunction
  - It always cause a real problem, for example, memory corruption

Concurrency fuzzing

Scheduling mechanisms

- Random scheduling
  - KRace, SKI, PCT algorithm
- Single conflict-oriented scheduling
  - Snowboard, Razzer

Coverage metric in the concurrency dimension

- Single conflict-oriented coverages
  - Race candidates
  - Alias coverages
  - PMC
- MUZZ(?)

Motivation

Motivation 1. the demand of a new scheduling mechanism for race conditions

- Random scheduling
  - suffers from exposing following concurrency bugs
    - inclusive concurrency bug
    - bugs that require a small race window
  - Duplicated schedule
    - need to verify
- Single conflict-oriented scheduling
  - wastes a lot of computing power because of lots of duplicated schedule regarding a manifestation of a crash
    - Those duplicated interleavings are called “???”
- New scheduling mechanism should
  - diversify interleavings across runs
  - be able to explore very specific corner cases

Motivation 2. the demand of a new coverage to capture interesting behavior

- We need a coverage metric to distinguish how much two interleavings are different
    - To determine two interleavings are diversified enough
    - To determine an interleaving covers a specific corner case
- Single conflict-oriented coverages
    - Cannot differnetiate interesting behaviors
        - Examples

Our approach

High-level idea

- With an executed interleaving, we divide the interleaving into several interleaving segments called XXX
-

Design

Design

- TODO

Limitations and future works

Limitations

- Too many interleaving segments
  - It consumes a lot of memory
- Exhaustively searching all segments are practically impossible
  - To the best of our knowledges, all fuzzers share this problem

Future works

-