RUB CASA Securing the Digital Society

Chair for Software Security

# The Things That Count

## Coverage Evaluation Under the Microscope

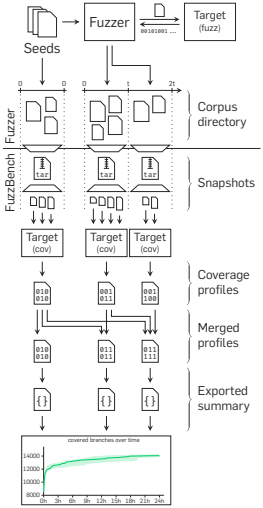
Tobias Holl[1] Leon Weiß[1] Kevin Borgolte

Ruhr University Bochum

5th International Fuzzing Workshop (FUZZING) · 27 February 2026

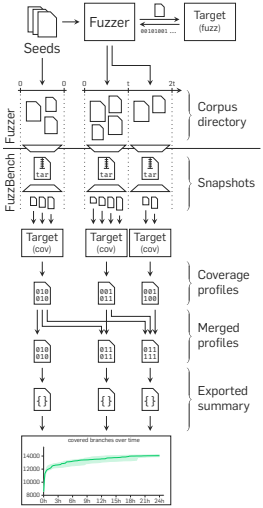
[1]Equal contribution

# FuzzBench




- Fuzzer benchmarking platform by Metzman et al. [2] at Google based on OSS-Fuzz
- Multiple trials for each fuzzer–benchmark combination
- Based on code coverage
- Uses `llvm-cov` for coverage

# FuzzBench




- Fuzzer benchmarking platform by Metzman et al. [2] at Google based on OSS-Fuzz
- Multiple trials for each fuzzer–benchmark combination
- Based on code coverage
- Uses `llvm-cov` for coverage

**Important**

- Standardized benchmarks are good
- This is **not** an attack on FuzzBench or `llvm-cov`

# Motivating Example

```
2220 int vdbePmaReaderIncrMergeInit(…) {
2229     rc = vdbeMergeEngineInit(…);
2234     if( rc==SQLITE_OK ){
             …
2256     }
         …
2280 }
```

taken: 1535 times
**not** taken: 18 446 744 073 709 551 614 times

# Motivating Example

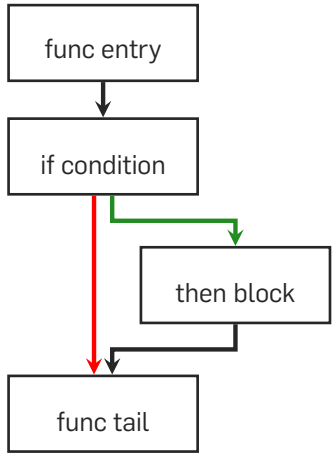
```
2220 int vdbePmaReaderIncrMergeInit(…) {
2229     rc = vdbeMergeEngineInit(…);
2234     if( rc==SQLITE_OK ){
             …
2256     }
         …
2280 }
```

taken: 1535 times
**not** taken: $2^{64} - 1$ times

# Motivating Example

```
2220 int vdbePmaReaderIncrMergeInit(…) {
2229     rc = vdbeMergeEngineInit(…);
2234     if( rc==SQLITE_OK ){
             …
2256     }
         …
2280 }
```

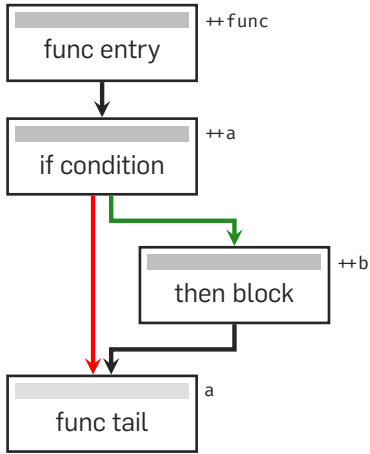
taken: 1535 times
**not** taken: $2^{64} - 1$ times



# Motivating Example

```
2220  int vdbePmaReaderIncrMergeInit(…) {
2229      rc = vdbeMergeEngineInit(…);
2234      if( rc==SQLITE_OK ){
              …
2256      }
          …
2280  }
```

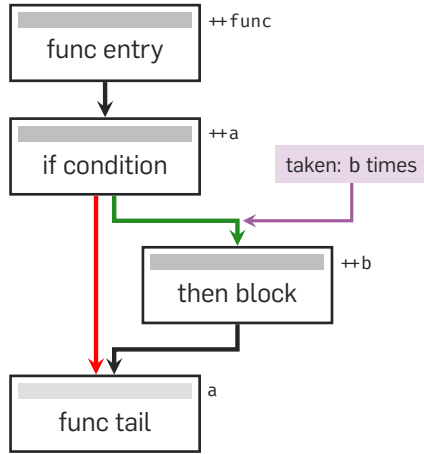
taken: 1535 times
**not** taken: $2^{64} - 1$ times



# Motivating Example

```
2220  int vdbePmaReaderIncrMergeInit(…) {
2229      rc = vdbeMergeEngineInit(…);
2234      if( rc==SQLITE_OK ){
              …
2256      }
          …
2280  }
```

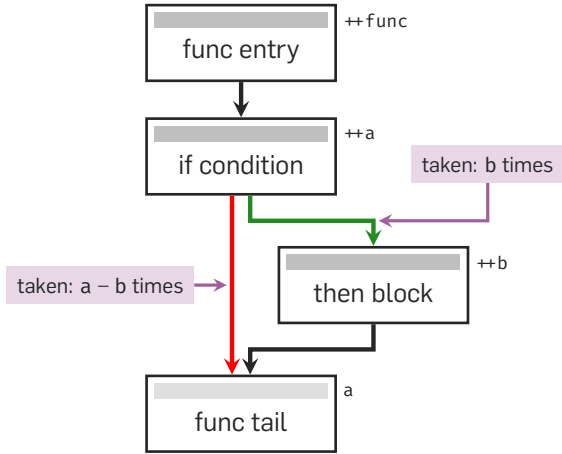
taken: 1535 times
**not** taken: $2^{64} - 1$ times



# Motivating Example

```
2220  int vdbePmaReaderIncrMergeInit(…) {
2229      rc = vdbeMergeEngineInit(…);
2234      if( rc==SQLITE_OK ){
              …
2256      }
          …
2280  }
```

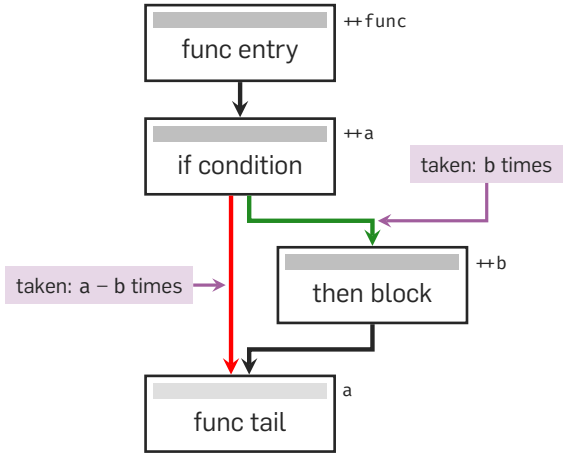
taken: 1535 times
**not** taken: $2^{64} - 1$ times



# Motivating Example

```
2220  int vdbePmaReaderIncrMergeInit(…) {
          function counter = 1533
2229      rc = vdbeMergeEngineInit(…);
          counter a = 1534
2234      if( rc==SQLITE_OK ){
              counter b = 1535
              …
2256      }
          …
2280  }
```

taken: 1535 times
**not** taken: a − b = −1 times

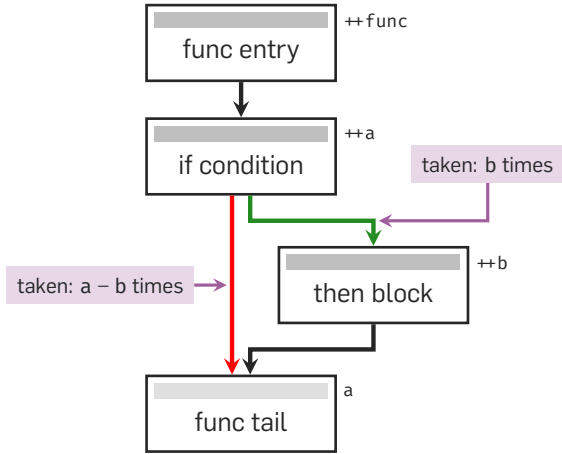


# Motivating Example

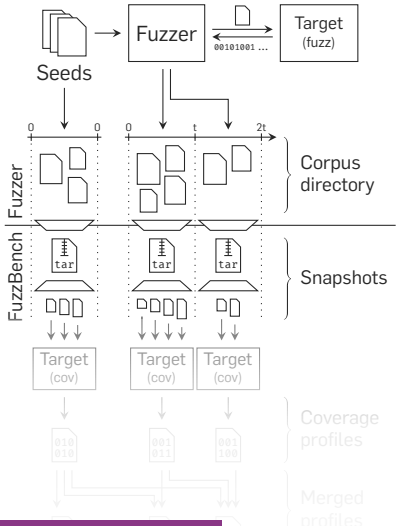
```
2220  int vdbePmaReaderIncrMergeInit(…) {
          function counter = 1533
2229      rc = vdbeMergeEngineInit(…);
          counter a = 1534
2234      if( rc==SQLITE_OK ){
              counter b = 1535
              …
2256      }
          …
2280  }
```

taken: 1535 times
**not** taken: a − b = −1 times

We counted a branch as **covered** that was **never** executed.

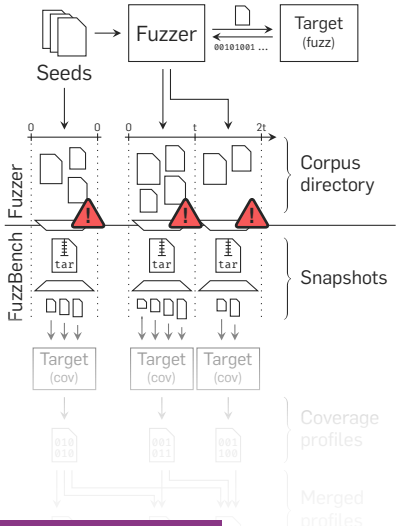


# FuzzBench Under the Microscope
## Snapshotting




- Snapshot corpus directory at fixed intervals
- Collect coverage profiles with $\text{target}_{\text{cov}}$

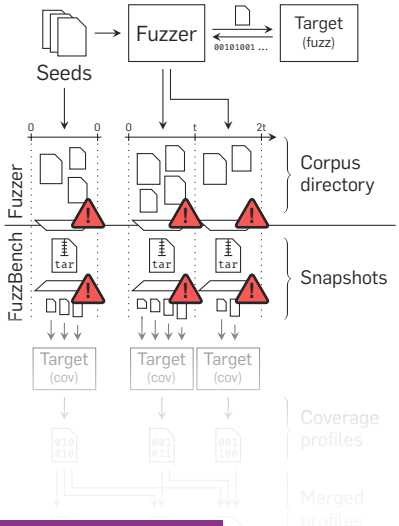
# FuzzBench Under the Microscope
## Snapshotting




- Snapshot corpus directory at fixed intervals
- Collect coverage profiles with $target_{cov}$

- Corpus selection
  - Metadata files

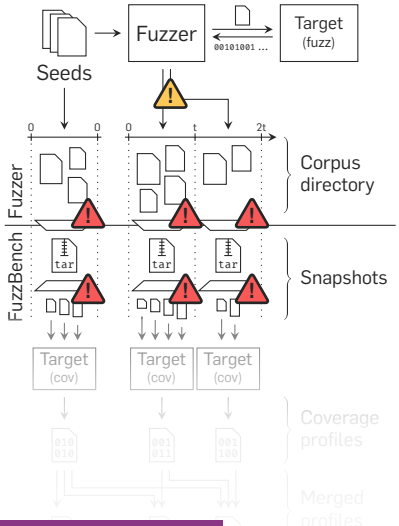
# FuzzBench Under the Microscope
## Snapshotting




- Snapshot corpus directory at fixed intervals
- Collect coverage profiles with $target_{cov}$

- Corpus selection
  - Metadata files
- Statefulness
  - Target restarts
  - Test case ordering

# FuzzBench Under the Microscope
## Snapshotting

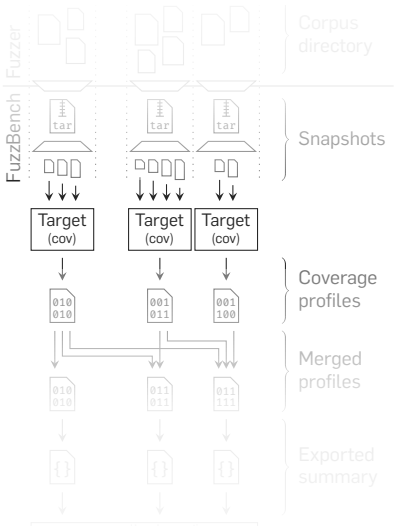



- Snapshot corpus directory at fixed intervals
- Collect coverage profiles with $\text{target}_{\text{cov}}$

- Corpus selection
  - Metadata files
- Statefulness
  - Target restarts
  - Test case ordering
  - Unsaved (non-queue) test cases ⚠ (cf. Lipp et al. [1])

# FuzzBench Under the Microscope

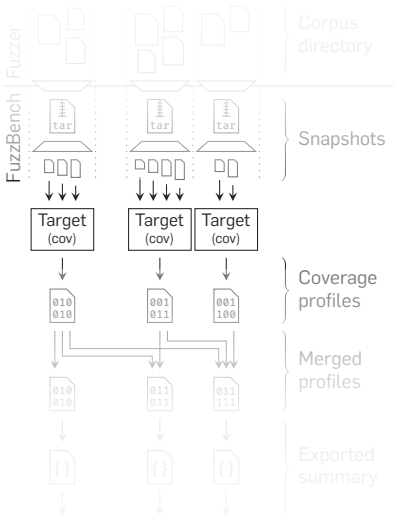
## Profile Instrumentation




- Instrument $\text{target}_{\text{cov}}$ at compile time
- Embed expressions for omitted counters
- Increment raw counters at run time

# FuzzBench Under the Microscope

## Profile Instrumentation

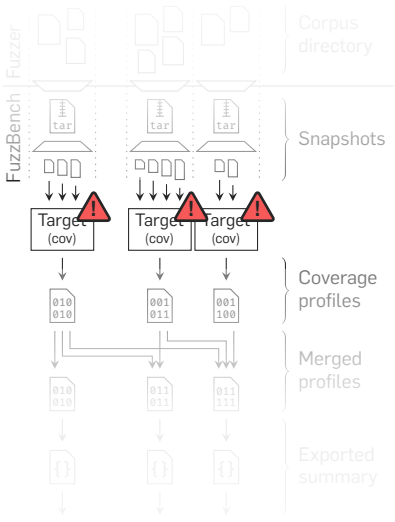



- Instrument $\text{target}_{\text{cov}}$ at compile time
- Embed expressions for omitted counters
- Increment raw counters at run time

- Miscomputations

# FuzzBench Under the Microscope

## Profile Instrumentation

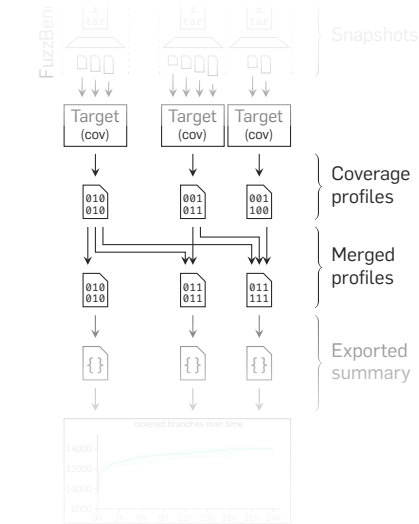



- Instrument target$_{cov}$ at compile time
- Embed expressions for omitted counters
- Increment raw counters at run time

- Miscomputations
  - Racing for counter updates
  - Crashes between counters
  - Signal handling
  - `setjmp/longjmp`

# FuzzBench Under the Microscope

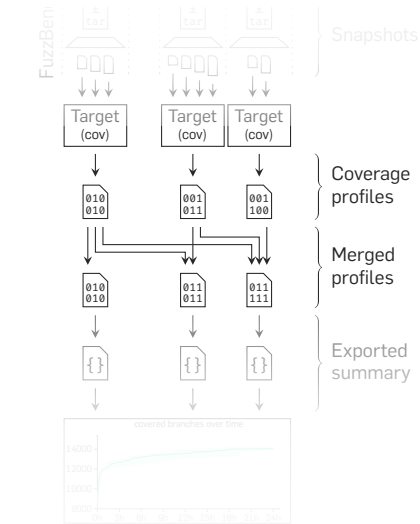
## Retention of Counters




- Write current profile to disk
- Merge with profiles of prior snapshots

# FuzzBench Under the Microscope

## Retention of Counters

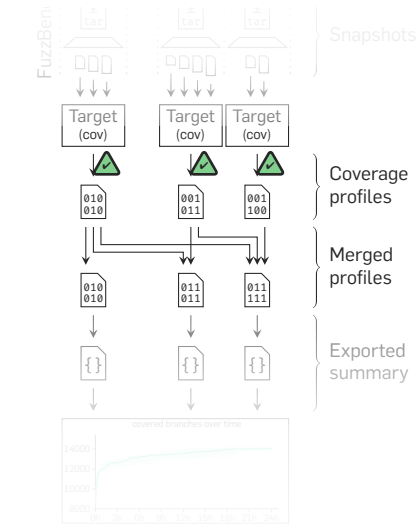



- Write current profile to disk
- Merge with profiles of prior snapshots

- Losing counters to crashes

# FuzzBench Under the Microscope

## Retention of Counters

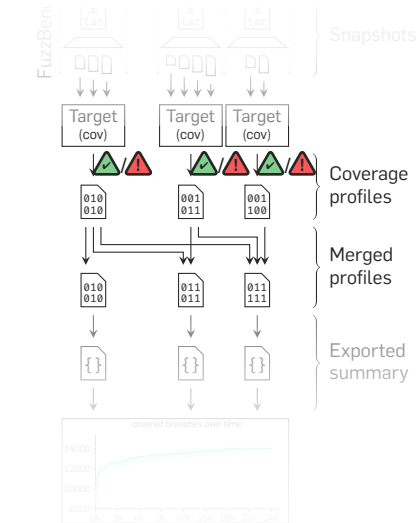



- Write current profile to disk
- Merge with profiles of prior snapshots

- Losing counters to crashes
  - No issue in FuzzBench ✅

# FuzzBench Under the Microscope

## Retention of Counters

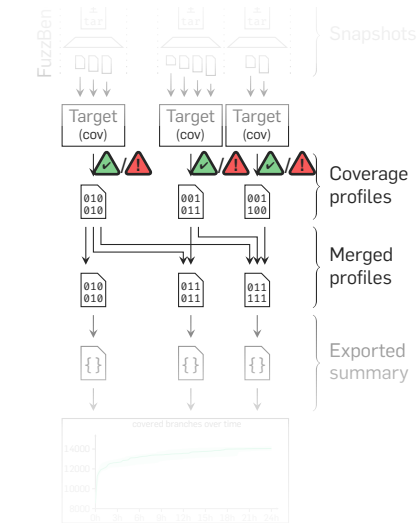



- Write current profile to disk
- Merge with profiles of prior snapshots

- Losing counters to crashes
  - No issue in FuzzBench
  - Broken in OSS-Fuzz

# FuzzBench Under the Microscope

## Retention of Counters

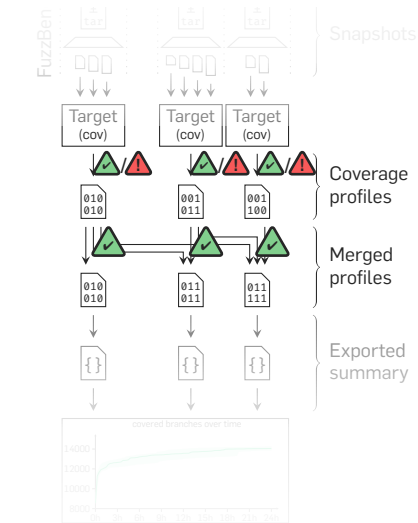



- Write current profile to disk
- Merge with profiles of prior snapshots

- Losing counters to crashes
  - No issue in FuzzBench
  - Broken in OSS-Fuzz
- Losing counters while merging

# FuzzBench Under the Microscope

## Retention of Counters

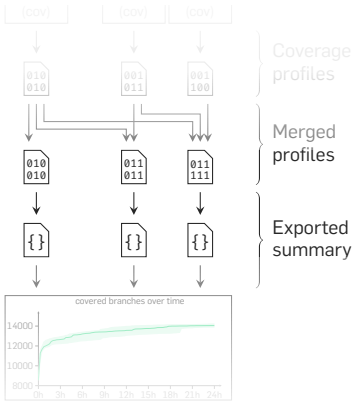



- Write current profile to disk
- Merge with profiles of prior snapshots

- Losing counters to crashes
  - No issue in FuzzBench ✅
  - Broken in OSS-Fuzz ⚠️
- Losing counters while merging
  - No issue (in practice)

# FuzzBench Under the Microscope

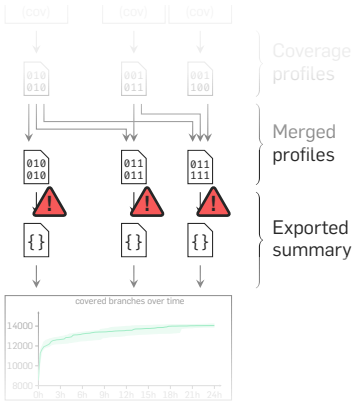
## Counters to Coverage




- Read mappings from $target_{cov}$
- Collect counters from profiles
- Map counters to lines and branches
- Export as coverage summary

# FuzzBench Under the Microscope

## Counters to Coverage

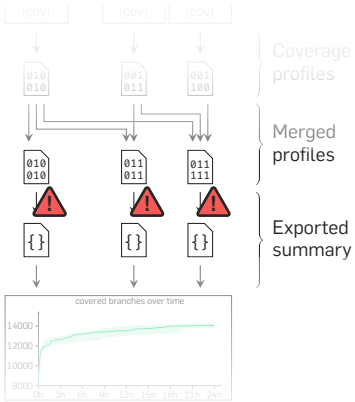



- Read mappings from $target_{cov}$
- Collect counters from profiles
- Map counters to lines and branches
- Export as coverage summary

Mismatched mappings

# FuzzBench Under the Microscope

## Counters to Coverage

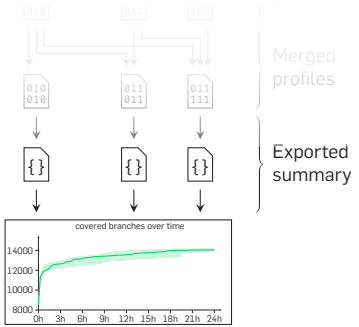



- Read mappings from $target_{cov}$
- Collect counters from profiles
- Map counters to lines and branches
- Export as coverage summary

- Mismatched mappings
- Incomplete mappings

# FuzzBench Under the Microscope

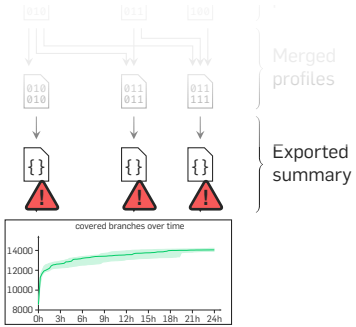
## Coverage to FuzzBench Reports




- Extract coverage from summary
- Generate report including:
  - Coverage over time
  - Unique branches information

# FuzzBench Under the Microscope

## Coverage to FuzzBench Reports

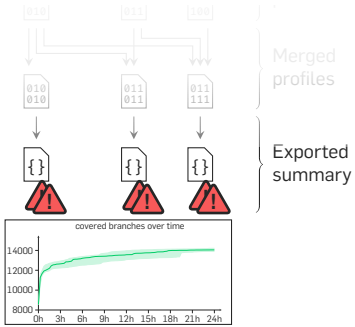



- Extract coverage from summary
- Generate report including:
  - Coverage over time
  - Unique branches information

- Instantiation summaries

# FuzzBench Under the Microscope

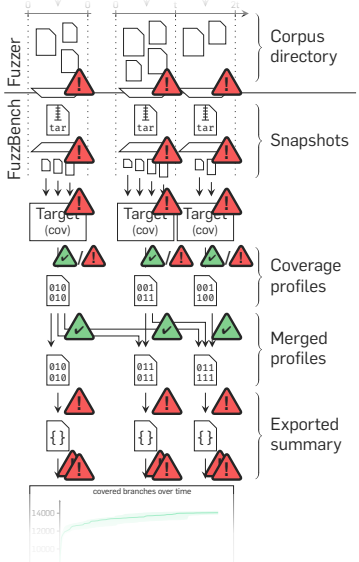
## Coverage to FuzzBench Reports




- Extract coverage from summary
- Generate report including:
  - Coverage over time
  - Unique branches information

- Instantiation summaries
- Mismatched definitions

# Sources of Inaccuracies




- Corpus Selection
- Statefulness

- Miscomputations

- Crashing test cases (OSS-Fuzz only)

- Mismatched or incomplete mappings

- Instantiation summaries
- Mismatched definitions

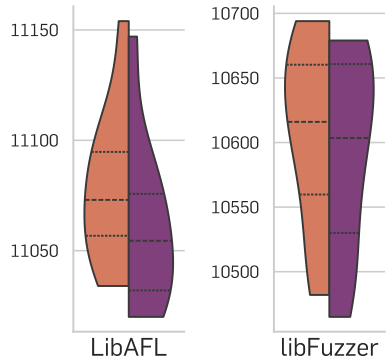
# How Large Is Their Impact?
## Preliminary Results

- Snapshotting
  - Corpus selection
  - Statefulness
- Profile Instrumentation
  - Miscomputations
- Retention of Counters
  - Crashing test cases (OSS-Fuzz only)
- Counters to Coverage
  - Mismatched or incomplete mappings
- Coverage to FuzzBench Reports
  - Instantiation summaries
  - Mismatched definitions

# How Large Is Their Impact?
## Preliminary Results

- Snapshotting
  - Corpus selection
  - Statefulness
- Profile Instrumentation
  - Miscomputations
- Retention of Counters
  - Crashing test cases (OSS-Fuzz only)
- Counters to Coverage
  - Mismatched or incomplete mappings
- Coverage to FuzzBench Reports
  - Instantiation summaries
  - Mismatched definitions

**HarfBuzz Branch Coverage**




Keeping ● and dropping ● state in the target.

# How Large Is Their Impact?
## Preliminary Results

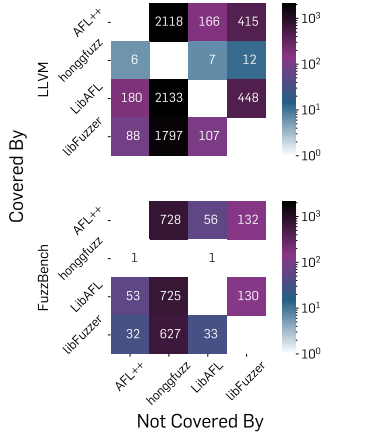
- Snapshotting
  - Corpus selection
  - Statefulness
- Profile Instrumentation
  - Miscomputations
- Retention of Counters
  - Crashing test cases (OSS-Fuzz only)
- Counters to Coverage
  - Mismatched or incomplete mappings
- Coverage to FuzzBench Reports
  - Instantiation summaries
  - Mismatched definitions

| Fuzzer | HarfBuzz | | | |
|---|---|---|---|---|
| | Summary 17 228 total | | All 32 658 total | |
| AFL++ | 10 920 | (63.39%) | 21 044 | (64.44%) |
| honggfuzz | 9 192.5 | (53.36%) | 17 652 | (54.05%) |
| LibAFL | 11 073 | (64.27%) | 21 042.5 | (64.43%) |
| libFuzzer | 10 616 | (61.62%) | 20 101.5 | (61.55%) |

# How Large Is Their Impact?
## Preliminary Results

- Snapshotting
  - Corpus selection
  - Statefulness
- Profile Instrumentation
  - Miscomputations
- Retention of Counters
  - Crashing test cases (OSS-Fuzz only)
- Counters to Coverage
  - Mismatched or incomplete mappings
- Coverage to FuzzBench Reports
  - Instantiation summaries
  - Mismatched definitions

### HarfBuzz Unique Branches



# How Large Is Their Impact?
## Experimental Plan

### Setup

- Standard experiment (10 trials $\times$ 24 hours)
- All 23 FuzzBench benchmarks
- 4 fuzzers: AFL++, honggfuzz, LibAFL, libFuzzer

# How Large Is Their Impact?
## Experimental Plan

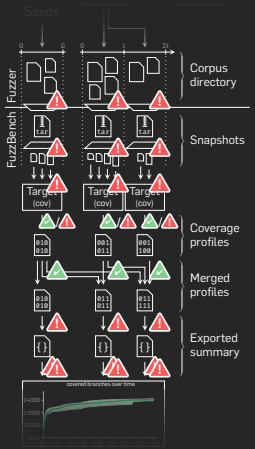
### Setup

- Standard experiment (10 trials $\times$ 24 hours)
- All 23 FuzzBench benchmarks
- 4 fuzzers: AFL++, honggfuzz, LibAFL, libFuzzer

### Evaluation

- Address each of the six sources of inaccuracies in isolation
- Reevaluate coverage on the fixed version
- Investigate coverage differences to baseline

# Takeaways




- Systematic analysis of coverage in FuzzBench
- Discovered 6 sources of inaccuracies
- Preliminary results show notable differences

- How large is the impact of each of the issues?
- Can the inaccuracies flip rankings?

Paper and slides:

softsec.link/fz26.cov

Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

CASA
Securing the Digital Society

WWTF
Vienna Science and Technology Fund

# References


[1] S. Lipp, D. Elsner, T. Hutzelmann, S. Banescu, A. Pretschner, and M. Böhme. “FuzzTastic: a fine-grained, fuzzer-agnostic coverage analyzer”. In: *Proceedings of the 2022 44th IEEE/ACM International Conference on Software Engineering (ICSE)*. May 2022. DOI: 10.1145/3510454.3516847.

[2] J. Metzman, L. Szekeres, L. M. R. Simon, R. T. Sprabery, and A. Arya. “FuzzBench: An Open Fuzzer Benchmarking Platform and Service”. In: *Proceedings of the 29th Joint Meeting on Foundations of Software Engineering (ESEC/FSE)*. Aug. 2021. DOI: 10.1145/3468264.3473932.