FuSeBMC v4

Smart Hybrid Fuzzing for C Program Security Testing

Test-Comp 2026 · Competition Contribution

Kaled Alshmrany

Institute of Public Administration, Saudi Arabia

What is FuSeBMC?

Bounded Model Checking

SMT solvers verify program paths and bypass complex mathematical guards that fuzzers cannot handle.

Evolutionary Fuzzing

Grey-box fuzzing explores program branches guided by smart seeds and shared coverage data.

Smart Seed Generation

A dedicated pre-phase produces high-impact seeds ranked by depth reached and unique labels covered.

How FuSeBMC Works

1

Instrument Code

Clang AST → inject GOAL labels into every branch, loop & function

2

Seed Generation

BMC + Fuzzer run 60 s on lightly instrumented code → rank test cases → smart seeds

3

Reachability Analysis

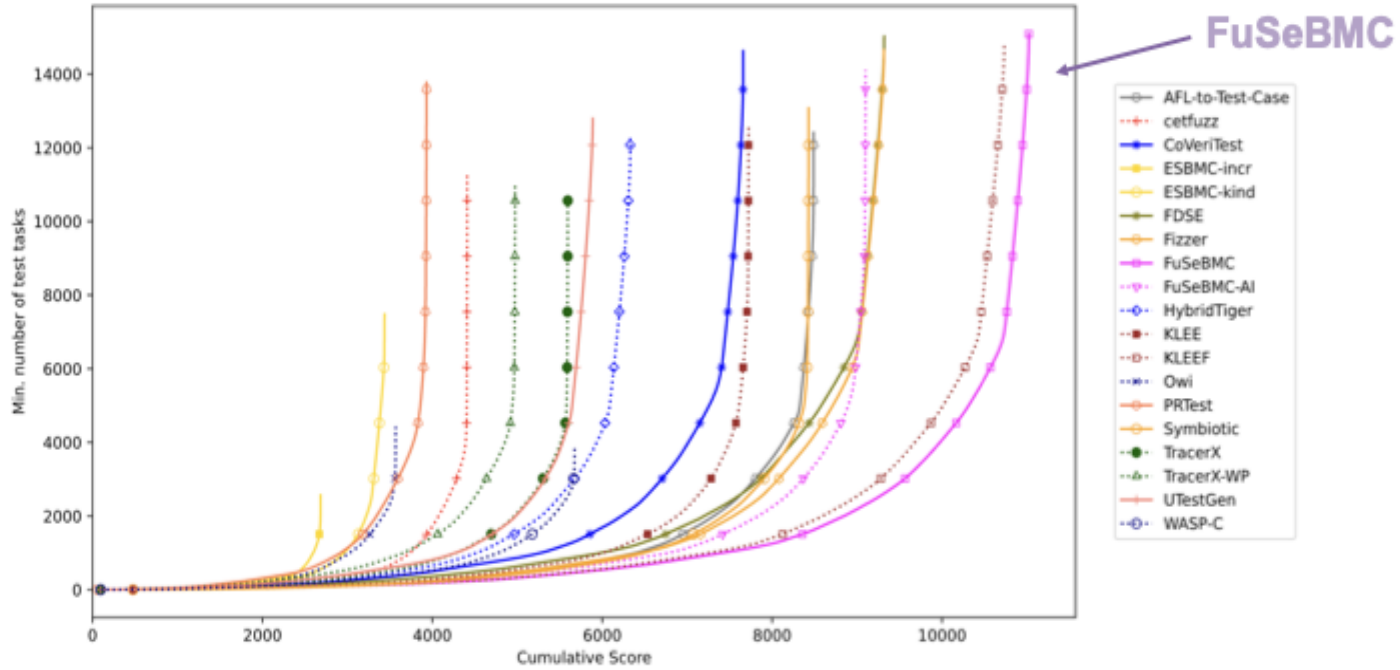
Full engines run in parallel using smart seeds; Tracer coordinates via shared memory

4

Output Test Suite

Highest-impact test cases exported as GraphML witness files

Test-Comp 2026 Results

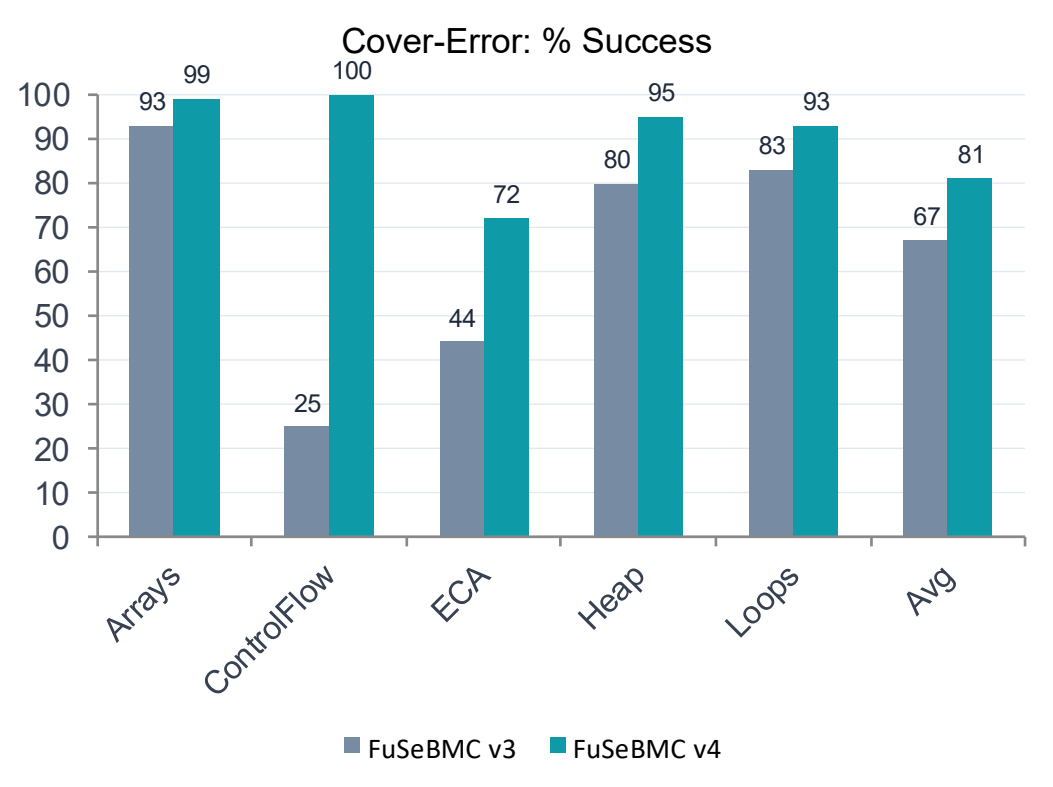


FuSeBMC achieved 3 awards: 1st place in `Cover-Error`, 2nd place in `Cover-Branches`, and 1st place in `Overall`

Alshmrany, K., Aldughaim, M., Bhayat, A., Cordeiro, L.: FuSeBMC v4: Smart Seed Generation for Hybrid Fuzzing - (Competition Contribution). FASE 2022: 336-340

https://test-comp.sosy-lab.org/2026/

Test-Comp 2026 Results vs Previous Version



+14%

avg improvement
Cover-Error

+16%

avg improvement
Cover-Branches

1st

Cover-Error
Test-Comp 2022

1st

Cover-Branches
Test-Comp 2022

FuSeBMC v4 — Test-Comp 2026

Building on our award-winning v4 architecture with targeted enhancements:

Faster Seed Phase

Reduced seed generation timeout tuning for faster warm-up on large programs.

Improved Loop Handling

Smarter loop bound estimation reduces wasted BMC executions on deep loops.

Stronger Engine Synergy

Enhanced Tracer coordination minimises redundant goals coverage across Fuzzing and BMC.

Extended Benchmark Support

Better handling of programs redefining C library functions — addressing v4's known weakness.

Participation & Setup

Competition Categories

- **Cover-Error**
- **Cover-Branches**
- **Overall**

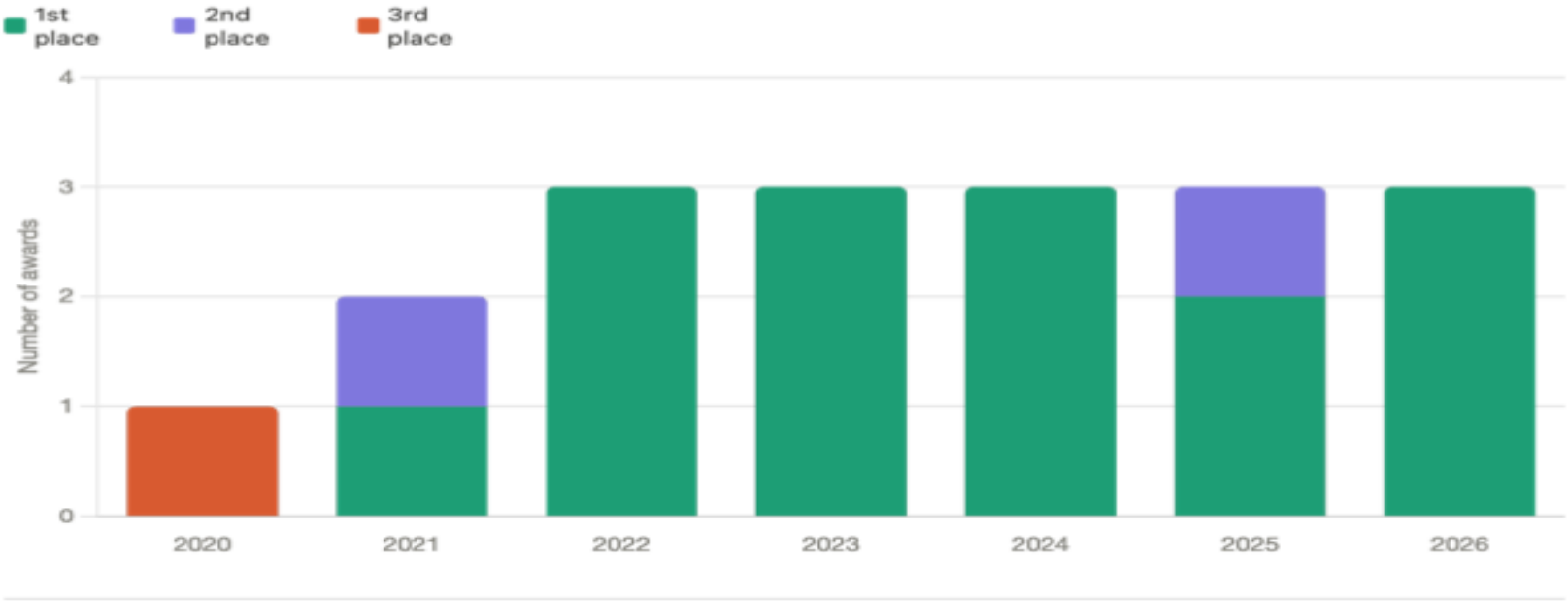
Run Command

```
fusebmc.py [-a {32,64}]
 [-p PROPERTY_FILE]

 [-s incr]

 [BENCHMARK]
```

Strategy used: incremental BMC (incr)
Max unwind: 50 bounds

MIT License · github.com/FuSeBMC

FuSeBMC Performance in Test-Comp (2020-2026)



Total awards	1st place finishes	Editions entered
20	16	7

Thank You

FuSeBMC v4 — Test-Comp 2026

Contact us for more info: