Open Challenges in the Field of Fuzzing: Pitfalls, Blind Spots, and Real-World Lessons

Nico Schiller

Summary

In recent years, fuzzing has become the most popular technique for automated vulnerability discovery in software. Yet, despite many publications and new tools, progress in practice appears to have stalled. Many fuzzers claim improvements through complex techniques, but their actual impact in real-world testing scenarios remains unclear. In this dissertation, we critically reflect upon the state of fuzzing research and identify key structural weaknesses that hinder its advancement.

The first contribution is a longitudinal study of AFL++, one of the most widely used fuzzers, known for incorporating cutting-edge academic research. By systematically benchmarking AFL++versions released over the past six years, we assess whether these integrated techniques translate into measurable progress. We find that performance has not improved consistently, and, in some cases, even regressed. A subsequent literature analysis reveals that only a small fraction of academic extensions to AFL++or its predecessor have been merged back into their baseline. Discussions with the maintainers point to a growing disconnect between research and practice, underscoring the need for stronger reproducibility standards and more realistic benchmarks.

Complementing this external perspective on adoption, the second contribution focuses on an internal algorithmic limitation of modern fuzzers such as AFL++. We analyze *input shadowing*, a subtle side effect of coverage-guided fuzzers that use novelty search-based corpus management. We propose a simple yet effective mitigation: periodically restarting the fuzzer to reset its search state. Despite its simplicity, this approach improves exploration and bug discovery, showing that small, practical interventions can outperform increasingly complex algorithms.

Finally, we illustrate the broader implications of these findings through a case study on the testing of proprietary consumer drones. This highlights how hidden assumptions in fuzzing limit applicability to real-world, hardware-integrated targets, where creative engineering and domain knowledge are essential to enable automated testing.

This dissertation argues that the future of fuzzing lies not in increasing technical complexity but in reevaluating what "progress" means: reproducibility over novelty, simplicity over sophistication, and real-world impact over artificial benchmarks.