FixReverter: A Realistic Bug Injection Methodology for Benchmarking Fuzz Testing

Zenong Zhang and Zach Patterson, *University of Texas at Dallas;* Michael Hicks, *University of Maryland and Amazon;* Shiyi Wei, *University of Texas at Dallas*

https://www.usenix.org/conference/usenixsecurity22/presentation/zhang-zenong

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 31st USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

Open access to the Artifact Appendices to the Proceedings of the 31st USENIX Security Symposium is sponsored by USENIX.

A Artifact Appendix

A.1 Abstract

This artifact goes through the 3 main steps of the evaluation with FixReverter and RevBugBench: (1) FixReverter bug injection, (2) FuzzBench experiments, and (3) FixReverter bug triage. The evaluation results of 5 different fuzzers on a benchmark generated by FixReverter, namely the RevBugBench, show that FixReverter is able to generate hard-to-find bugs and differentiate the performance of fuzzers. As the full-scale experiments require a lot of time and resources, the artifact provides all the intermediate products of each step for partial reproductions. A machine with Ubuntu system, at least 24 CPU cores and 200GB RAM is recommended for the experiments.

A.2 Artifact check-list (meta-information)

- **Run-time environment:**

  Ubuntu 16.04, Docker 20.10.7 and python 3.9.

- **Hardware:**

  At least 200GB RAM and a 24-core CPU are recommended.

- **Output:**

  Performance of 5 fuzzers on RevBugBench.

- **How much disk space required (approximately)?:**

  500GB if running the full-scale evaluation. This can be reduced by running only partial experiments.

- **How much time is needed to prepare workflow (approximately)?:**

  2 hours.

- **How much time is needed to complete experiments (approximately)?:**

  One week if running the full-scale evaluation. Running partial experiments with provided intermediate products can take from 1 hour to several days.

A.3 Description

A.3.1 How to access

figshare URL: `https://figshare.com/articles/software/Supplementary_artifact_for_the_paper_FIXREVERTER_A_Realistic_Bug_Injection_Methodology_for_Benchmarking_Fuzz_Testing_/20647821`

DOI: 10.6084/m9.figshare.20647821

A.3.2 Hardware dependencies

N/A

A.3.3 Software dependencies

Ubuntu 16.04, Docker 20.10.7 and python 3.9. Other dependencies (Clang, FuzzBench and Phasar) are automatically handled in the provided docker images.

A.3.4 Data sets

N/A

A.3.5 Models

N/A

A.3.6 Security, privacy, and ethical concerns

N/A

A.4 Installation

Installation guides are included in the README of the artifact.

A.5 Experiment workflow

The workflow is described in detail with the README of the artifact. Each section comes with numbered steps explaining the workflow, and necessary actions come with highlighted commands.

A.6 Evaluation and expected results

We made 3 major claims in the evaluation of the paper.

- FixReverter injects bugs that fuzzers can actually find.
- FixReverter injects bugs that are hard to find.

- Fuzzers can find combination causes in RevBug-Bench.

First, the results show hundreds of bugs can be found by the 5 evaluated fuzzers. Second, some observations of the difference in fuzzers' performance shows the difficulty for fuzzers to find the injected bugs, as described in Section 5.2 of the paper. For example, each fuzzer detected unique bugs that other fuzzers did not find, indicating that injected bugs do not overfit a single approach in the evaluated fuzzers. Finally, there are hundreds of combined causes identified in the results. Because fuzzing is a random process, this artifact is expected to produce results that support the above 3 claims and are reasonably similar to the numbers reported in Section 5.

A.7 Experiment customization

A.8 Notes

A.9 Version

Based on the LaTeX template for Artifact Evaluation V20220119.