Bachelor/Master Thesis Topic

Testing the oracle: Identify weaknesses and bugs in sanitizers

Motivation and Background:

Sanitizers like AddressSanitizer and UBSan are essential components for security testing. Amongst others, they enable the detection of memory safety violations and pointer-type errors, which malicious attackers can exploit to access confidential data or gain control of a system. Sanitizers are also very important for fuzzing, which relies on them as test oracles. A recent paper [SP'2019] provides a systematic overview of the current sanitizers. Nevertheless, detailed information about sanitizers still belongs to domain knowledge that is only accessible to security experts. This project aims to dig deeper and analyze sanitizers and identify bugs and inconsistencies, report the identified bugs, and provide documentation and recommendations for sanitizers based on the conducted experiments.

Student Task and Responsibilities:

- Make yourself familiar with fuzz testing and sanitizers. This includes to create an overview of current sanitizers and also add sanitizers published in the recent years.
- Select a subset of sanitizers with overlapping functionality.
- Curate a dataset with programs that include security vulnerabilities.
- Design/select evaluation metrics beyond functionality (e.g., performance overhead).
- Use the dataset to perform a testing campaign of the selected sanitizers (e.g., via differential fuzzing).
- Analyze the identified inconsistencies and document your findings.
- If you find bugs in sanitizers, report them.

Deliverables:

- Evaluation artifacts (dataset, testing tools, drivers, etc.)
- Documented findings of the conducted experiments. Of particular interest are: bugs in sanitizers, inconsistencies between reported documentation and actual behavior, identified usability issues, and recommendations for their usage.

Pre-Requisites: (Programming Languages, OS, Skills, Papers, etc)

Knowledge in C/C++ and/or LLVM is helpful for this project. Further, it would be beneficial if the student had some experience applying fuzz testing.

[SP'2019] D. Song *et al.*, "SoK: Sanitizing for Security," *2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2019, pp. 1275-1295, doi: 10.1109/SP.2019.00010.

[Software'2021] M. Boehme, C. Cadar and A. Roychoudhury, "Fuzzing: Challenges and Reflections" in IEEE Software, vol. 38, no. 03, pp. 79-86, 2021.

Contacts

Prof. Dr. Yannic Noller (`sq-office@rub.de`)
Software Quality group, Faculty of Computer Science, Ruhr University of Bochum