Lehrstuhl für Sicherheit in der Informatik
Prof. Dr. Claudia Eckert

Fraunhofer AISEC

Ausschreibung: HiWi

Fuzzing with Memory Safety

Fuzzing is a method widely used for automated testing of software. Fuzzing frameworks, such as AFL[1] and LLVM libfuzzer[2], are able to find various bugs by generating more or less random input for the program under test. Bugs are typically detected as crashes of the tested program. Consequentially, bug detection is heavily dependent on bugs causing crashes and some bugs might only be detected imprecisely or not detected at all, even if they are triggered by the fuzzing.

Approaches that make C/C++ memory safe can avoid this problem, as memory corruptions are always detected when they are triggered by the fuzzing. Different approximations of memory safety are available for C/C++. For example, Address Sanitizer (ASAN) is already widely used in conjunction with fuzzing in order to increase the bug detection rate.

In this work, different compiler-based memory safety mechanisms should be tested for their applicability in improving the bug detection rate when testing with common fuzzing frameworks, namely AFL and libfuzzer.

Requirements

- Ability to work independently and accurately
- Good C/C++ programming skills
- Basic knowledge of exploit techniques, ARM assembly and Linux
- Basic knowledge of fuzzing frameworks, e.g. AFL and LLVM libfuzzer
- Interest in fuzzing techniques and compiler development, especially LLVM

Contact

Julian Horsch
Telefon: +49 89 322-9986-155
E-Mail: julian.horsch@aisec.fraunhofer.de

Fraunhofer-Institut für Angewandte und Integrierte Sicherheit (AISEC)
Lichtenbergstr. 11, 85748 Garching (near Munich), Germany
http://www.aisec.fraunhofer.de

[1] https://github.com/google/AFL
[2] http://llvm.org/docs/LibFuzzer.html