Software Engineering Seminar

Combining Fuzzing and Symbolic Execution for Debugging

Description

Fuzzing is a great technique to, for example, discover and reproduce software system vulnerabilities. However, there exist problems with finding test inputs for complex checks (e.g., string equality checks). On the other hand, symbolic execution is able to find inputs for branches that are the hard to reach, but is comparably slow and uses a lot of computational power. Recent approaches propose to combine fuzzing techniques with symbolic execution to effectively tackle their individual problems [1, 2].

The student should examine and discuss different approaches that combine symbolic execution and fuzzing techniques in the context of debugging.

References

[1] Saahil Ognawala, Thomas Hutzelmann, Eirini Psallida, and Alexander Pretschner. Improving function coverage with munch: A hybrid fuzzing and directed symbolic execution approach. *CoRR*, abs/1711.09362, 2017.

[2] Nick Stephens, John Grosen, Christopher Salls, Andrew Dutcher, Ruoyu Wang, Jacopo Corbetta, Yan Shoshitaishvili, Christopher Krügel, and Giovanni Vigna. Driller: Augmenting fuzzing through selective symbolic execution. In *NDSS*, 2016.

Contacts

Simon Heiden (`heiden@informatik.hu-berlin.de`)
Software Engineering Group
Institut für Informatik
Humboldt-Universität zu Berlin