Software Engineering Seminar

Combining Fuzzing and Symbolic Execution

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). *Symbolic Execution*, on the other hand, is theoretically able to exhaustively check every program branch, but practically struggles to do so, due to the path explosion problem. Recent approaches propose to *combine fuzzing techniques and symbolic execution* to effectively tackle the problems that both techniques have on their own.

The student should examine and discuss the approaches given in the papers and compare them to each other and to similar existing techniques.

References

[1] Saahil Ognawala, Thomas Hutzelmann, Eirini Psallida, and Alexander Pretschner. Improving function coverage with munch: A hybrid fuzzing and directed symbolic execution approach. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, pages 1475–1482, New York, NY, USA, 2018. ACM.

[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