Software Engineering Seminar

Finding Bugs/Exploits with Symbolic Execution

Description

Exploits in software systems may pose big safety and security risks. Naturally, (automated) techniques to detect existing exploits are necessary to ensure the well-behaviour of software systems. In this context, symbolic execution, as, for example, used in [2], can be a powerful tool.

The student is to examine and discuss techniques using symbolic execution to detect bugs/exploits.

References

[1] Jacob Burnim, Sudeep Juvekar, and Koushik Sen. Wise: Automated test generation for worst-case complexity. In *Proceedings of the 31st International Conference on Software Engineering*, ICSE '09, pages 463–473, Washington, DC, USA, 2009. IEEE Computer Society.

[2] Sang Kil Cha, Thanassis Avgerinos, Alexandre Rebert, and David Brumley. Unleashing mayhem on binary code. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 380–394, Washington, DC, USA, 2012. IEEE Computer Society.

Contacts

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