(+86) 13220190100
Haidian, Beijing, China
sunhao.th@gmail.com

Hao Sun

M.Sc., Tsinghua, System Security

GitHub: SunHao-0
Page: haosun.info

- System security researcher focusing on operating system kernels, with an emphasis on the **Linux** kernel.
- Interested in developing practical vulnerability discovery tools, author of the kernel fuzzer **Healer**.
- Passionate about designing complex, distributed systems, familiar with source code of multiple huge systems.
- Interested in programming language theory, current language preference is **Rust**.

Education

- **M.Sc.**, *School of Software*, *Tsinghua University* 09/2020 — Current
  Software System Security Assurance Group, supervised by Prof. Yu Jiang.
  System security, operating system kernels

- **B.Sc**, *School of Software*, *Beijing University of Posts and Telecommunications (BUPT)* 09/2016 — 06/2020

Publications

- **HEALER: Relation Learning Guided Kernel Fuzzing**
  Hao Sun, Yuheng Shen, Cong Wang, Jianzhong Liu, Yu Jiang, Ting Chen, and Aiguo Cui
  *2021, ACM SIGOPS 28th Symposium on Operating Systems Principles (**SOSP '21**)*

- **Rtkaller: State-aware Task Generation for RTOS Fuzzing**
  Yuheng Shen , **Hao Sun**, Yu Jiang, Heyuan Shi, Yixiao Yang, and Wanli Chang
  *2021, ACM Transactions on Embedded Computing Systems (**EMSOFT '21**)*

- **Go-Sanitizer: Bug-Oriented Assertion Generation for Golang**
  Cong Wang , **Hao Sun**, Yiwen Xu, Yu Jiang, Huafeng Zhang, Ming Gu
  *2019 IEEE International Symposium on Software Reliability Engineering Workshops (**ISSREW**)*

Software

- **Healer, kernel fuzzer inspired by Syzkaller**
  Healer is an automated kernel vulnerability discovery tool, written in 17,000+ lines of Rust. It utilizes system call specifications that encode structures and partial semantics of the input to generate system call sequences. Assisted with various sanitizers, Healer detects kernel bugs via triggering kernel crashes with generated sequences. While the idea behind Healer is relatively straightforward, it incorporates many tricks and techniques for efficient runtime behavior, resulting in relatively high system load compared with other tools and contributing to its good performance.

  PARTIAL RESULTS: 100+ reported and fixed Linux bugs, **10+** CVEs assigned, **187** stars on github.

- **KSG, kernel syscall specification generator**
  Writing system call specifications in a domain-specific language is quite laborious, requiring significant manual efforts. KSG is designed to generate these specifications automatically for *Healer* and Google's *Syzkaller*. It incorporates probe-based tracing and symbolic execution-based analysis to extract system call information from the kernel's source code. KSG is implemented using 7000+ lines of C++ based on the Clang Static Analyzer (CSA). This work is under revision at **ATC' 22**.

- **UFUZZ, fuzzer for OSEK/VDX RTOS kernel.**
  UFUZZ is an automated bug discovery tool, designed for embedded RTOS kernels that conform to the OSEK/VDX specification. UFUZZ is implemented in pure Rust. It generates test cases with the awareness of the application model, e.g., prioritized tasks and ISRs, and transfers inputs via directly accessing the memory of guest VM, and evolves the corpus with execution feedback collected from the t32 simulator. UFUZZ has been adopted and deployed in a private organization.

Experience

- **Intern**, System Developing, *Architecture Group*, *ByteDance Ltd*. 01/2019 — 06/2019
  Designed and implemented a core data hub system with 10k+ QPS in Golang. Based on a Graphql-liked framework, the hub separated data fetching and extracting into different components as well as cached hot data into LRU buffer thus increasing system throughout rate and stability.

- **Intern**, System Developing, *Cainiao Network*, *Alibaba Group*. 05/2018 — 09/2018
  Designed and implemented a template-based database interacting code generator, which automatically generates code that interacts with the database in a uniform style and reduces manual efforts.