
Rahul Kande†, Addison Crump†, Garrett Persyn†, Patrick Jauernig*, Ahmad-Reza Sadeghi*, Aakash Tyagi†, and Jeyavijayan Rajendran†
†Texas A&M University, USA, *Technische Universität Darmstadt, Germany
†{rahulkande, addisoncrump, gpersyn, tyagi, jv.rajendran}@tamu.edu,
*{patrick.jauernig, ahmad.sadeghi}@trust.tu-darmstadt.de

Abstract

The increasing complexity of modern processors poses many challenges to existing hardware verification tools and methodologies for detecting security-critical bugs. Recent attacks on processors have shown the fatal consequences of uncovering and exploiting hardware vulnerabilities.

Fuzzing has emerged as a promising technique for detecting software vulnerabilities. Recently, a few hardware fuzzing techniques have been proposed. However, they suffer from several limitations, including non-applicability to commonly-used hardware description languages (HDLs) like Verilog and VHDL, the need for significant human intervention, and inability to capture many intrinsic hardware behaviors, such as signal transitions and floating wires.

In this paper, we present the design and implementation of a novel hardware fuzzer, *TheHuzz*, that overcomes the aforementioned limitations and significantly improves the state of the art. We analyze the intrinsic behaviors of hardware designs in HDLs and then measure the coverage metrics that model such behaviors. *TheHuzz* generates assembly-level instructions to increase the desired coverage values, thereby finding many hardware bugs exploitable from software. We evaluate *TheHuzz* on four popular open-source processors and achieve $1.98 \times$ and $3.33 \times$ the speed compared to the industry-standard random regression approach and the state-of-the-art hardware fuzzer, DifuzzRTL, respectively. Using *TheHuzz*, we detected 11 bugs in these processors, including 8 new bugs, and we demonstrate exploits using the detected bugs. We also show that *TheHuzz* overcomes the limitations of formal verification tools from the semiconductor industry by comparing its findings to those discovered by the Cadence JasperGold tool.

I. Main Content

This research [1] is recently published in USENIX Security 2022. The original abstract and author list are shown above. We post the paper link with the conference version†.

REFERENCES


†https://www.usenix.org/conference/usenixsecurity22/presentation/kande

Rahul Kande†, Addison Crump†, Garrett Persyn†, Patrick Jauernig†, Ahmad-Reza Sadeghi†, Aakash Tyagi†, Jeyavijayan Rajendran†
†Texas A&M University, College Station, USA, ‡Technische Universität Darmstadt, Germany.
{rahul.kande, addisoncrump, pji, tyagi, jrajendran}@tamu.edu, {patrick.jauernig, ahmad.sadeghi}@trust.tu-darmstadt.de

Results

- Fuzzed 4 different real-world open-source processors from RISC-V and OpenRISC ISAs
- Faster coverage than the state-of-the-art
- Detected 11 bugs including 8 new bugs
  - All types of fence instruction
  - Permissions for control status registers
  - Implementation of overflow bit
- Developed two exploits
  - Privilege escalation using MOR1KX bugs
  - Arbitrary code execution using Ariane bugs

Our Solution: HARDWARE FUZZING

- While TheHuzz is faster than traditional verification techniques and capable of detecting new vulnerabilities, there is still scope for improvement in terms of the fuzzing speed, achieving close to 100% coverage, and fuzzing different types of hardware designs.
- Thus our future focus is on the following:
  - FPGA fuzzing: Emulate hardware on FPGA to improve the speed of fuzzing.
  - Formal fuzzing: Leverage capabilities of formal tools to improve the coverage achieved by TheHuzz.
  - Fuzzing non-processor designs: Extend TheHuzz to fuzz non-processor designs such as cryptographic accelerators.

Contact Information
RAHUL KANDE
Ph.D. in Computer Engineering
Texas A&M University
rauhulkande@tamu.edu
Ph: +1 979-739-8914
SETH research lab: https://seth.engr.tamu.edu/

Published at USENIX Security Symposium, 2022

Published at USENIX Security Symposium, 2022

References

Acknowledgements
Our research work was partially funded by the US Office of Naval Research, by Intel’s Scalable Assurance Program, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), and by the German Federal Ministry of Education and Research and the Heranian State Ministry for Higher Education, Research, and the Arts within AITBZ.

We thank Kevin Luehrs (UC Berkeley), Jiaowen Hu (Dalian University of Technology), and TAMU HiPC for their support. Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors, and do not necessarily reflect those of the US Government.