

Alexandros Ntyrkai

Dimitris Gizopoulos
University of Athens, Greece

Vasileios Karakostas

cal.di.uoa.gr

1. Motivation & Goal

Motivation

- Vulnerability assessment against cache side-channel attacks is critical when security is a must-have feature
- Prior works either target a subset of attacks, lack support for the RISC-V ISA, or rely on in-house simulators & tools

Goal

- Provide support for comprehensive evaluation of RISC-V CPUs against a wide range of cache timing attacks
 - Real systems → variety of hardware platforms
 - Simulated systems → gem5 open-source simulator

Our contributions enable

- Security evaluation, even early during the design phase
- Comparison of different ISAs against cache attacks
- Systematic assessment of mitigation mechanisms

Background: The CTVS methodology [1]

- 88 strong vulnerabilities → 1094 generated tests
- Quantifies the vulnerability of the L1 data cache
- Lower CTVS means that the CPU is more secure

2. Approach

Our Approach

- We leverage the Cache Timing Vulnerability Score (CTVS) methodology [1] and we enhance it along two axes:
 - We port CTVS to the RISC-V ISA
 - We integrate CTVS with the gem5 simulator

Challenges

- Avoid platform specific instructions
 - The base RISC-V ISA lacks instruction for flushing the data cache (like the x86 clflush instruction)
 - We transform flush operations into prime operations with CacheFlush() [2]
- Adjust and tune tests to run them correctly in gem5
 - Proper math functions to analyze results
 - Automation scripts for easy experimentation

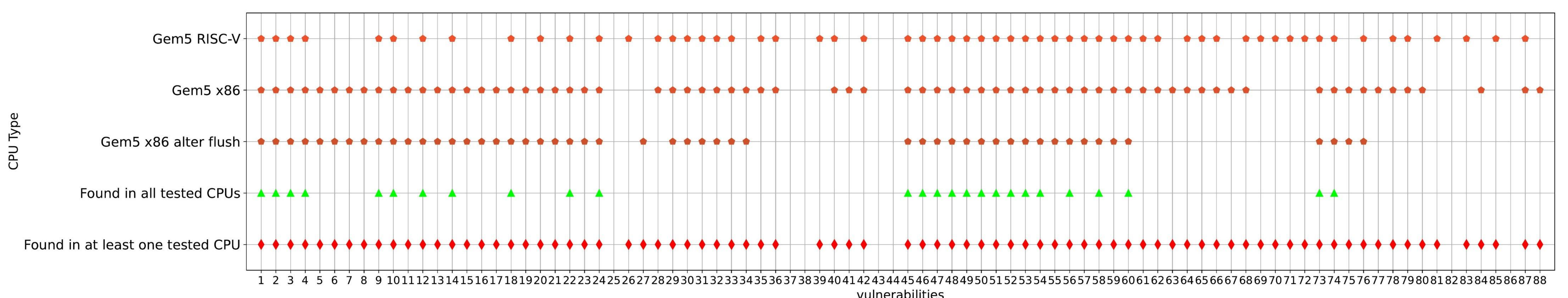


3. Evaluation

Methodology

- gem5 full-system simulation
- Timing CPU model with classic caches
- 4-core systems based on the RISC-V & x86 ISAs

	RISC-V	x86	x86-alt	all CPUs	≥1 CPU
CTVS Score	59/88	71/88	51/88	26/88	81/88



CTVS results for the 88 strong theoretical types of attacks on the simulated RISC-V & x86 systems (a dot means the system is vulnerable to that attack type).
 "RISC-V" → ported tests with CacheFlush() | "x86" → original tests with clflush | "alter flush" → modified tests with CacheFlush().

Preliminary Results

- RISC-V tests have fewer effective cases than the original x86 tests with clflush, but more effective cases than the modified x86 tests with CacheFlush()
- Most RISC-V tests indicate vulnerabilities that are also triggered by x86 tests, but not always
- CacheFlush() has fewer effective cases than clflush on x86

References

- [1] Shuwen Deng, Wenjie Xiong, and Jakub Szefer. "A Benchmark Suite for Evaluating Caches Vulnerability to Timing Attacks". In: ASPLOS '20
- [2] <https://github.com/riscv-boom/boom-attacks>
- [3] C. Austa, J. T. Mühlberg, J. M. Dricot. "Systematic Assessment of Cache Timing Vulnerabilities on RISC-V Processors". In: ESORICS '25 [Concurrent work that has also ported CTVS to RISC-V, but lacks support for gem5]