A Hardware-Based Cache Side Channel Attack Detection Mechanism for RISC-V Processors

Andreas Brokalakis[1,2], Alexandros Skyvalos[1], Sotiris Ioannidis[1], Iakovos Mavroidis[1] and Ioannis Papaefstathiou[2]

[1]School of Electrical and Computer Engineering, Technical University of Crete, Greece

[2]Exascale Performance Systems PLC, Greece

Summary

- Side-channel attacks rely on information that can be gathered (or leaked) by the fundamental way a computer system operates. For CPU-based systems, a prominent number of these attacks target the caches aiming at gaining unauthorized access to sensitive data.
- Proposed solutions provide cannot reliably identify such attacks.
- Following an analysis of such attacks on RISC-V processors, we identified that they all depend on accessing specific architectural registers to successfully complete.
- We present a detection mechanism implemented at the hardware level that can detect *all* such attacks, *without producing false negative detections* and *without requiring any software assistance or modifications*.

Background & Motivation

- Cache-based Side-channel attacks exploit the principle that the access time of a particular cache line depends on whether it has been accessed by a victim process or not.
- They are particularly important on their own but they also provide the underlying layer for the success of most other side channel attacks, even if they target other parts of a processor (e.g. Meltdown and Spectre).
- Defense mechanisms that have been proposed at both hardware and software level introduce too many overheads and area/performance penalties and due to the **lack of a reliable detection mechanism**, they affect equally all process (legit and malicious).

Experimental Analysis & Observations

- We employed a **CVA6 processor** (RV64GC compliant) and implemented a fully working, Linux-based system on a Digilent Genesys 2 FPGA dev board.
- We managed to replicate a series of cache-based side channel attacks, such as **Prime+Probe** and **Evict+Reload**.
- We observed that during the attacks, an *extraordinary high amount of timing register (rdcycle) accesses*.
- The frequency of these accesses were **orders of magnitude higher** than any other legit application tested.
- We modified the code of the attacks to use other timing resources, in which event, the attacks failed, since they require cycle-level accuracy.

Proposed Solution

- At the hardware level, we added *a **counter** to measure the number of accesses to rdcycle register*. Through this counter, we constructed an attack detection mechanism.
- This mechanism can **detect 100% of these attacks, without producing false negatives**.
- *Implementation cost is negligible* and it does *not require any software* to be involved.
- As a result, we are capable to construct a complete mitigation mechanism at the hardware level, without requiring software (OS or applications) to be modified at all.

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ TECHNICAL UNIVERSITY OF CRETE

REBECCA

EXA P SYS

Co-funded by the European Union

Chips JU

ACKNOWLEDGMENT: REBECCA project is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Authorities under grant agreement n° 101097224. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the granting authority. Neither the European Union nor the granting authority can be held responsible for them.