Detecting and Patching Transient Execution Side Channels in an Out-of-Order RISC-V Core

Tobias Jauch[1], Alex Wezel[1], Mohammad Rahmani Fadiheh[1], Dominik Stoffel[1] and Wolfgang Kunz[1]
[1]Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau

Transient Execution Attacks

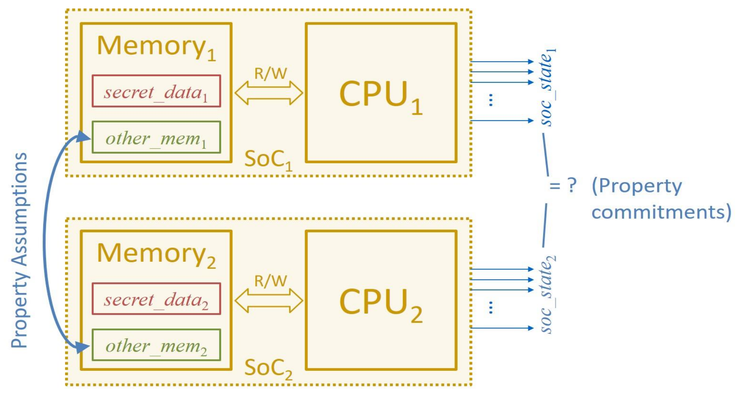
- Transient Execution Side Channel (TES) attacks exploit speculative and out-of-order execution in modern CPUs
- An attacker can trick the CPU into transiently accessing confidential data that leave a footprint in microarchitectural buffers although the architectural state is reverted
- Examples are **Spectre**, **Meltdown** or **MDS attacks**

UPEC: Formal RTL Security Verification

- Exhaustively detect Transient Execution Side Channels in RTL implementations
- No need for a-priori knowledge on attacks
- Proof method scales to out-of-order processors
- UPEC has also been extended to other security targets, such as integrity, data-oblivious execution or confidentiality in SoCs

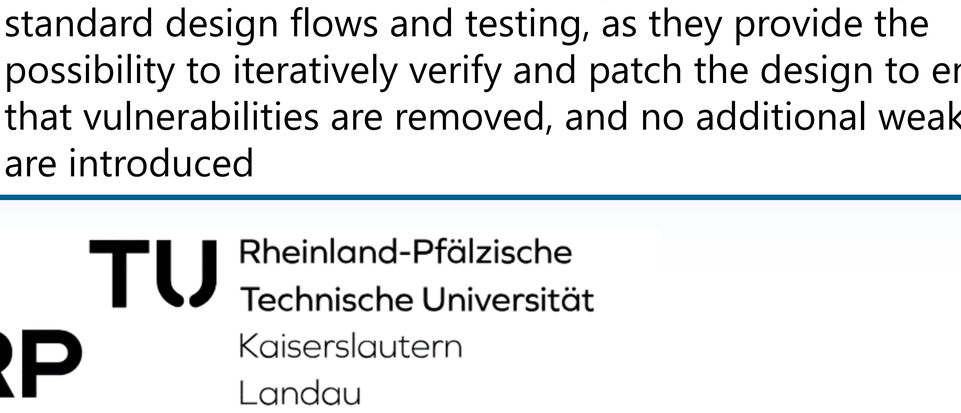
Unique Program Execution Checking

- Miter model: Two instances of the design under verification
- Initial state (memory, pipeline buffers, etc.) is constrained to be arbitrary but identical → CPU may execute any program
- Only difference: secret data in memory
- If both instances produce the same execution trace (clock-cycle accuracy), execution is independent of secret data and no transient execution side channel exists in the design



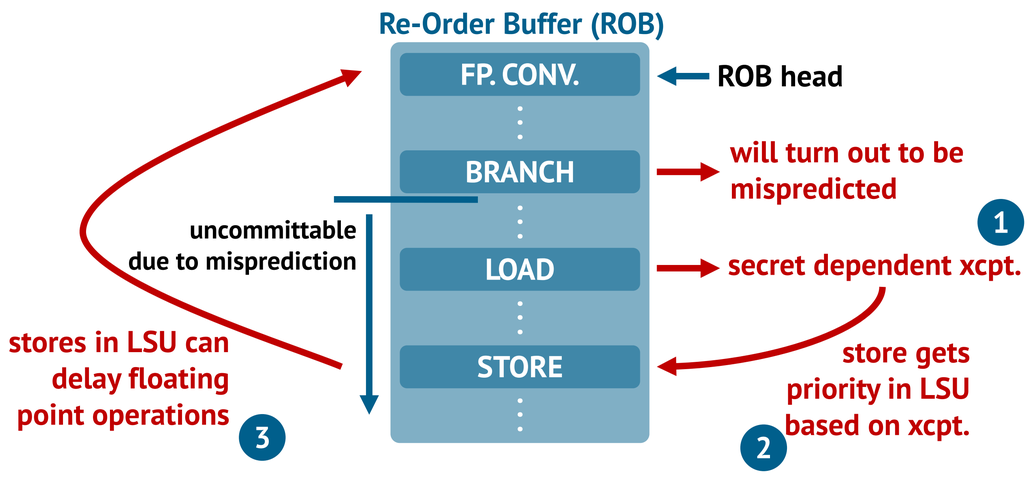
Meltdown Attack on BOOMv3

- Formal UPEC analysis discovered Meltdown vulnerability in BOOMv3 that had been introduced by minor design updates
- Misaligned loads and page faults raise exceptions as intended, but only send them to the core (1) while the faulting load itself can still propagate to the data cache
- The time between address translation and handling of the exception at ROB head can be used to read confidential data and create footprints in the microarchitecture
- Our implemented patch introduces an exception flag to the load queue and uses this flag to prevent faulting loads from being sent to the data cache (2)
- Further UPEC analysis showed that this fix introduced a new possibility for a Spectre attack. Sending a different instruction to the data cache in case a faulting load was blocked created secret-dependent timing. Killing faulty loads in the S1 stage makes the scheduling independent from transient data (3)



Spectre-Type Vulnerabilities in BOOMv3

- Besides classic Spectre attacks, UPEC analysis discovered a previously unknown contention-based Spectre variant
- Speculative secret-dependent loads can delay non-transient floating-point operations at ROB head
- The vulnerability was introduced by a design bug in BOOM's floating-point to integer unit (FPIU) that is responsible for conversions and floating-point store instructions.
- The unit can only process new requests if both operations could be performed, i.e., the FPIU is not busy, and LSU is ready to process a store operation
- Secret-dependent execution of a store instruction in LSU can therefore block an older, non-transient float. conversion



Conclusion

- Implementing secure mitigations to transient execution attacks in modern out-of-order processors is still a challenge
- Leakage paths can easily be missed in complex designs, and it is possible to even introduce new vulnerabilities when fixing the detected ones
- Being aware of all possible consequences of a design decision is a very demanding task for designer
- Formal tools like UPEC provide a significant advantage over standard design flows and testing, as they provide the possibility to iteratively verify and patch the design to ensure that vulnerabilities are removed, and no additional weaknesses are introduced

Related Publications

M. R. Fadiheh, D. Stoffel, C. Barrett, S. Mitra, W. Kunz: "Processor Hardware Security Vulnerabilities and their Detection by Unique Program Execution Checking",
Design Automation and Test in Europe (DATE), 2019

M. R. Fadiheh, J. Müller, R. Brinkmann, S. Mitra, D. Stoffel, W. Kunz: "A formal approach for detecting vulnerabilities to transient execution attacks in out-of-order processors",
57th IEEE/ACM Design Automation Conference (DAC'20), 2020

J. Müller, M. R. Fadiheh, A. Duque Anton, T. Eisenbarth, D. Stoffel, W. Kunz: „A Formal Approach to Confidentiality Verification in SoCs at the Register Transfer Level",
58th IEEE/ACM Design Automation Conference (DAC'21), 2021

L. Deutschmann, J. Müller; M. R. Fadiheh; D. Stoffel and W. Kunz: „Towards a Formally Verified Hardware Root-of-Trust for Data-Oblivious Computing ",
59th IEEE/ACM Design Automation Conference (DAC'22), 2022.

M. R. Fadiheh, A. Wezel, J. Müller, J. Bormann, S. Ray, J. M. Fung, S. Mitra, D. Stoffel, W. Kunz: „An Exhaustive Approach to Detecting Transient Execution Side Channels in RTL Designs of Processors,
IEEE Transactions on Computers, January 2023

D. Mehmedagic, M. R. Fadiheh, J. Müller, A. L. Duque Anton, D. Stoffel and W. Kunz, "Design of Access Control Mechanisms in Systems-on-Chip with Formal Integrity Guarantees,"
32nd USENIX Security Symposium, 2023 (To Appear).

Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

RISC-V Summit Europe, Barcelona, 5-9th June 2023