# Transient Execution Attacks and Countermeasures on RISC-V Implementations

Suzaki Laboratory, Institute of Information Security (情報セキュリティ大学院大学　須崎研究室)
Tuo Chen (陳　拓, M1, mgs234502@iisec.ac.jp) and Kunihiro Suzaki (須崎　有康, suzaki@iisec.ac.jp)

## I. Introduction

Transient execution attacks, symbolized by Spectre variants and categorized as cache timing SCA (Side Channel Attack), have been growing for a few years. This situation will be further extended by the open architecture "RISC-V ecosystem" because many implementations will appear. In this presentation, we present the research history of transient execution attacks at first. Then, we explain Spectre variants and several countermeasures on current CPUs. We also want to discuss the countermeasure ideas for the next stage.

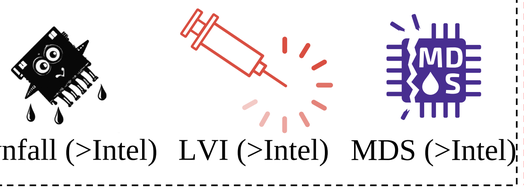
## II. Transient execution attack (TEA)

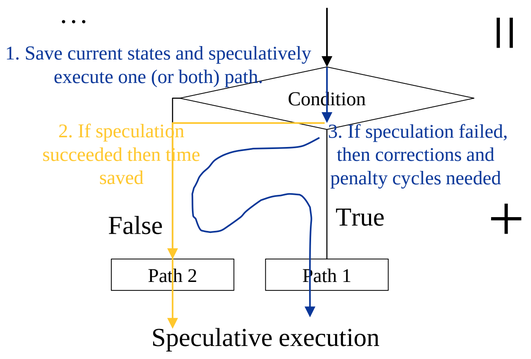
It is a broad category of microprocessor hardware security issues introduced by implementing **speculative execution.** The adversary conducts general cache timing SCAs like **Flush/Evict+Reload/Time**, **Prime+Probe**, etc. at the last stage. Until now, most of them target mainstream processors of Intel, AMD, ARM, Apple. Case studies of them are of significance for RISC-V ISA.

A. Some of them can be directly validated on existing RISC-V instances. -- We are investigating and counting these.

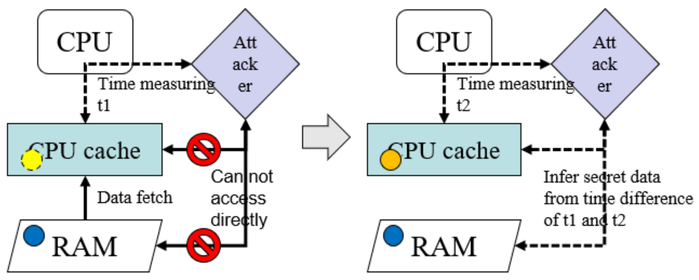
B. The remainders temporarily can not be replicated due to proprietary designs, but mechanisms behind them may get transplanted to or have their equivalences on RISC-V platforms in the future.

CROSSTalk (>Intel)
Retbleed (>AMD)
CTRAP (>AMD)
Zenbleed (>AMD)
INCEPTION (>AMD)
…




=




Timing SCA (Flush/Evict+Reload/Time, Prime+Probe, etc)

## III. Spectre vulnerabilities

Spectre is a typical group of TEAs, exploiting **branch prediction**. We have confirmed the following 3 dyed rows on RISC-V boards.

| Spectre-* | CVE | Public name | Citation |
|---|---|---|---|
| v1 | 2017-5753 | BCB: Bounds Check Bypass | Kocher et al, IEEE S&P 19 |
| v1.1 | 2018-3693 | BCBS: Bounds Check Bypass Store | V. Kiriansky et al, arXiv 18 |
| v1.2 | (Unindexed) | RPB: Read-only protection bypass | V. Kiriansky et al, arXiv 18 |
| v2 | 2017-5715 | BTI: Branch Target Injection | Kocher et al, IEEE S&P 19 |
| v3 | 2017-5754 | Meltdown or RDCL: Rogue Data Cache Load | M. Lipp et al, USENIX Sec 18 |
| v3a | 2018-3640 | RSRR: Rogue System Register Read | INTEL-SA-00115 18 |
| v4 | 2018-3639 | SSB: Speculative Store Bypass | Kocher et al, IEEE S&P 19 |
| v5 | (Unindexed) | ret2spec: Return Mispredict | Koruyeh et al, USENIX Sec 18 |
| Lazy FP | 2018-3665 | Lazy FP State Restore | Stecklina et al, arXiv 18 |
| BHI | 2022-0001<br>2022-0002<br>2022-23960 | BHI: Branch History Injection | Barberis et al, USENIX Sec 22 |
| v6 | (Unindexed) | SRV: Speculative Vectorization Exploit | S. Karuppanan et al, arXiv 23 |
| LAM | (Unindexed) | SLAM: Spectre based on Linear Address Masking | Hertogh et al, IEEE S&P 24 |

## IV. Present mitigations against Spectre

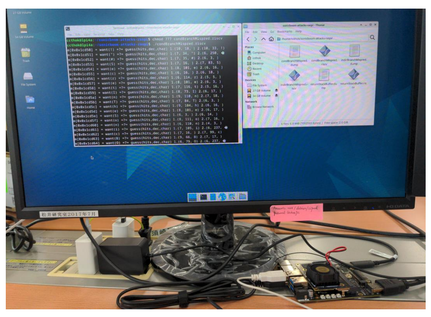
Some of the following proposals are already based on RISC-V.

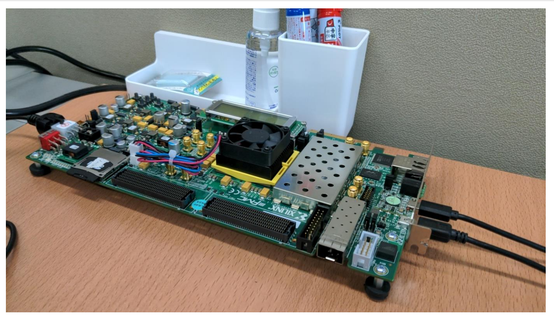
| Proposal | Description | Type(s) | against | Citation |
|---|---|---|---|---|
| (General remarks) | 1. Prevent speculative execution<br>2. Prevent access to secret data<br>3. Prevent data from entering covert channel<br>4. Limit data extraction from covert channels<br>5. Prevent branch poisoning | All | v1, v2, v4 | P. Kocher et al IEEE S&P 19 ("Spectre white paper") |
| Retpoline | Include lfence/pause instruction and utilization of user-controllable RSB (Return Stack Buffer), instead of solely relying on indirect branch predictor that is vulnerable. | SW | v2 | Google Project Zero |
| Indirect branch instructions | Porting of Retpoline from Intel x86 into RISC-V and some improvements. Use modified indirect "jump" and "call" similarly. | SW | v2. v5 | R. Bălucea and P. Irofti arXiv, Jun. 09, 2022 |
| Condition branch conversion, others | 1. Remove dependencies on branch output in secret data.<br>2. Convert conditional branch into equivalent unconditional instructions<br>3. Other HW-assisted SW defenses. | SW + HW | v1, v2, v4, v5 | D. Evtyushkin et al ACM, Mar. 2018 |
| SpecBuf | Propose certain forms of "dedicated speculation buffer". Their common feature is holding data for speculative executions. For InvisiSpec, secret data will be invisible even from the covert channel, until it is confirmed secure and is then released open. Data that breaks memory consistency will also be detected. For SpecBuf, data from failed speculation will be eliminated in the buffer, avoiding influence on caches for general purpose. | HW | v1, v2, v5 | Gonzalez et al U. C. Berkeley, 2018 |
| SafeSpec | | | | Khasawneh et al arXiv, Jun. 15, 2018 |
| InvisiSpec | | | | M. Yan, J. Choi et al IEEE, Oct. 2018 |
| SSE-RV | Along with loaded data for speculative execution, mark the destination registers with "taints". If next speculation accesses addresses of tainted registers again, it will be blocked by the LSU with fence instr. | HW | v1, v2, v5 | M. Sabbagh et al CARRV 2021 |

**Reference complements:**
[1] L. Gerlach, D. Weber, R. Zhang, and M. Schwarz, "A Security RISC: Microarchitectural Attacks on Hardware RISC-V CPUs," in 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA.
[2] Le Anh-Tien, "Research of RISC-V Out-of-order Processor Cache-Based Side-Channel Attacks - Systematic Analysis, Security Models and Countermeasures -," PhD Thesis, University of Electro-Communications, 2023.
[3] C. Canella et al., "A Systematic Evaluation of Transient Execution Attacks and Defenses," presented at the 28th USENIX Security Symposium (USENIX Security 19), 2019, pp. 249–266.

## V. Experiment environment for RISC-V TEAs

We tested Spectre-v1, v2, v5 on several LicheePi 4A SBCs. A 64-bit SonicBOOM Linux-capable single RISC-V core is also running on an AMD Xilinx Virtex 7 FPGA VC707 evaluation kit but still needs tuning. Now we are looking for more solutions with high flexibility like Chipyard, VMs, etc.

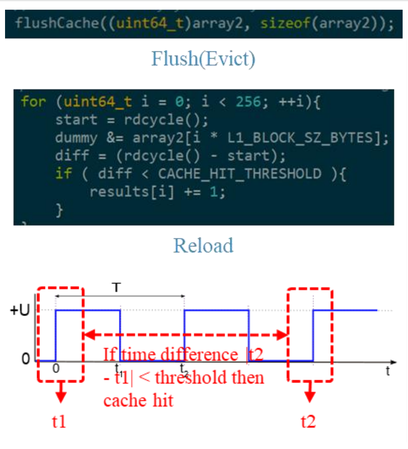


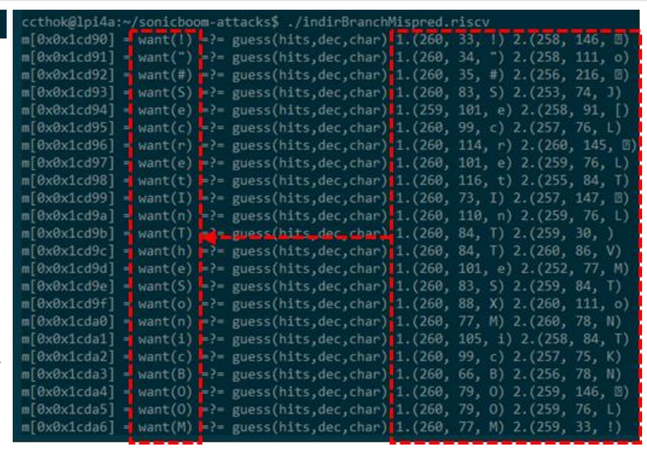


…

## VI. Spectre attack on RISC-V test results

We analyzed and modified Spectre codes from previous studies and successfully logged Spectre-v2.




## VII. Our ideas on RISC-V TEA countermeasures

1. Eliminate or disable CPU cache covert channel.
2. Detect potential TEA attempts and turn off speculation temporarily.
3. Switch flow dependency between control and data on demand.