HW CWE SIG Meeting
Friday, March 8, 2024

Members in Attendance

Bruce Monroe	Joerg Bormann
Bob Heinemann	Mell, Peter M. (Fed)
Salehi, Soheil	Constable, Scott D
Steven Christey Coley	Bojanova, Irena V. (Fed)
Gage Hackford	Frost, Sandy
Nicole Fern	Das, Amitabh
James Pangburn	Ford, Thomas
Mike Borza	Mohan Lal
Forbes, Justin	Kepa, Krzysztof
Milind Kulkarni	Jason Oberg (Cycuity)
Manna, Parbati K	Evan Bryers
Hallman, John (DI SW ICS DVT SM)	

Agenda

- Architecture Weaknesses
- Inclusive Language

Housekeeping

- Next meeting: April 12th, 12:30 – 1:30 PM EST (16:30 – 17:30 UTC), MS Teams.
- Contact: cwe@mitre.org
- Mailing list: hw-cwe-special-interest-group-sig-list@mitre.org
- Minutes from previous meetings: https://github.com/CWE-CAPEC/hw-cwe-sig

Announcements

- CWE Content Development Repository (CDR) pilot now on Github – currently invite only
- CWE 4.14 released February 29
  -4 new weakness entries related to transient execution
    - Thank you, Intel, AMD, ARM, CyCulty and Riscure

  -10 Demonstrative examples from HACK@DAC
    - Thank you Mohamadreza and team
- CWE 4.15 release will be around June/July

Transient Execution Weaknesses Media

- Chips and Salsa: Industry Collaboration for New Hardware Podcast hosted by Intel
- Intel Corporation Blog Post about Transient Weaknesses
- Dark Reading article

Overview of Recently Released Microarchitectural Weaknesses

Scott Constable (Intel)

Recap: Pre-4.14

- Observable response discrepancy in some Intel processors may allow and authorized user to potentially enable information disclosure via local access.
- This definition doesn't give much information about what the vulnerability is or what the root cause is. Why do we have this description?
- Technical Advisory states that this is an in domain transient execution methodology known as speculative code store bypass that may allow data values to be inferred during the transient execution of self-modifying code and some Intel processors.
- Why do we need to describe what the vulnerability is in a separate document and not in the CVE itself?
    - Language is supposed to be adopted from an existing CWE description
    - Fairly large corpus of CWE descriptions
    - Many for software, fewer for hardware
- This CWE describes a side channel type of behavior.
- There were some CWEs that were more specific but not fully accurate.
- CWE 1037 – developer builds a security critical protection into the software, but the processer optimizes the execution of the program such that the mechanism is removed or modified.
- Why doesn't this apply?
- JITs that run Javascript or Python are constantly generating new code and sometimes overwriting old code.
- Speculative Code Store Bypass
- CWE 1264 – The self-modifying code is not an error handling check. It is an out of order mechanism that is in place that detects when code has been modified after the code has potentially already been executed and unwinds the pipeline then starts from the point where the modifications happened and re-executes, reloads all of the code.
- CWE 1303 – Hardware resources shared across execution contexts can violate the expected architecture isolation between contexts.
- CWE 1342- The processor does not properly clear microarchitectural state after incorrect microcode assists or speculative execution resulting in transient execution.
    - Very reluctant to use this description as it prescribes a particular mitigation that most commodity hardware vendors would be reluctant to implement.

Timeline

- October 2021 – Priya and Scott first discuss HW CWE challenges
- April 2022 – Intel submits initial proposal for one new HW CWE
- September 2022 – First discussion in the CWE H SIG, decision to expand the proposal into multiple new CWEs

- October 2022 – New CWE hierarchy presented and discussed in HW SIG
- December 2022 – Intel submits draft documentation for 4 new CWEs
- January-September 2023 – Collaborative effort to refine proposal

Today: CWE 4.14

- Speculative code store bypass doesn't fall into any of 1421, 1422, or 1423. It can't leak architecturally restricted data. Does not involve microarchitectural.
- Its falls into 1420, which is very flexible.
- CWE 1420 – A processor or event or prediction may allow incorrect operations or correct operations with incorrect data, to execute transiently, potentially exposing data over a covert channel.

Microarchitectural Weakness Flowchart

- This will take anyone to the intended CWE.
- Flowchart is useful but it is not a part of the schema.

Questions:

1. Could a vulnerability have multiples of these weaknesses?

   Yes. The LVI load value injection is related to another family of vulnerabilities called microarchitectural data sampling or MSDS, and the salient difference between the two is MDDS. You can get data from another domain, so that might architecturally restrict data. That is the salient difference.

Other Content in these CWEs

- Modes of introduction
- Potential mitigations
- Detection methods
- Demonstrative examples
- Additional general information about each weakness and how certain hardware behaviors can contribute to the introduction of vulnerabilities.

Inclusive Language and HW CWEs (Inclusive Terminology)

- We talked about this and have had a history of trying to use it.
- Master and slave – we'd like to address this in next iteration.
- 9 CWEs use this terminology and is used frequently throughout.
- Can be addressed without a huge amount of effort.
- Options for changing:
  - For "bus master" – "first-party DMA"
- Must ensure that the meaning and the intention remains clear and can eliminate cognitive load on reader.