HW CWE SIG Meeting
Friday, February 9, 2024

Members in Attendance

Gananand G Kini	Monroe, Bruce	Hallman, John (DI SW ICS DVT SM)
Bob Heinemann	Mell, Peter M. (Fed)	Charles Timko (Red Hat)
Alec J Summers	Constable, Scott D	Evan Bryers
Steven Christey Coley	Bojanova, Irena V. (Fed)	Jason Oberg (Cycuity)
Gage Hackford	Frost, Sandy	Mohan Lal
Nicole Fern	Ahmed, Faheem	Swami, Shivam
James Pangburn	Ford, Thomas	Das, Amitabh
Mike Borza	Rich Piazza	Iyer, Priya B
Forbes, Justin	Kepa, Krzysztof	Salehi, Soheil
Sathyamurthi Sadhasivan	Masike, Takunda	
Manna, Parbati K	Milind Kulkarni	

Agenda

- Microarchitectural Weaknesses Update
- Mitigation and Detection Methods in CWE
- Observed Examples (OBEX) Working Group Formation

Housekeeping

- Next meeting: March 8, 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. Potential public release in early 2024.
- CWE 4.14 release planned for February 29. Demonstrated examples (DEMOXs) and microarchitectural weaknesses have been a priority for this release.
- CWE 4.15 release will be around June/July
- Expect a new meeting series and cancellation of the current series. Ownership of the meetings is changing. Times and frequency will be the same.
- Gana Kini is transitioning from MITRE. Will remain a SIG member and contributor.

Microarchitectural Weaknesses Update

- Working group met January 22, and approved moving forward with four new submissions (CWE-1420, CWE-1421, CWE-1422, and CWE-1423). The submissions will be included in the February 29 CWE 4.14 release. Intel, Cycuity, and MITRE plan to coordinate announcing the new microarchitectural weaknesses.
- Additional work to be done for CWE release 4.15 include: path forward for CWE-1342: Information Exposure through Microarchitectural State after Transient Execution; define what is meant by architectural and microarchitectural to clarify the distinction for readers who are not familiar; and some other minor updates.

Mitigation and Detection Methods in CWE

- A mitigation refers to a decision, action, or practice intended to reduce the impact of, or eliminate, a weakness.
- The potential elements of a mitigation are:
    - Phase (development life cycle phase in which the mitigation may be applied)
    - Description (including strengths and weaknesses of the mitigation to address the weakness)
    - Strategy (general strategy for protecting a system to which this mitigation contributes)
    - and optionally Effectiveness (summary of how effective the mitigation may be in preventing the weakness).
    - Additional notes can also be added.
- Detection elements of a weakness include:
    - Detection Method (the method used to detect the weakness)
    - Description (how the method can be applied to a specific weakness)
    - optionally Effectiveness (how effective the detection method may be in detecting associated weakness).
    - Effectiveness notes may also be added to describe strengths and shortcomings of the detection method.
- A set of Detection Method Labels is included in the meeting slides. A comment was made that identification of HW-specific weakness detection labels would be helpful.
- For CWE, testing is a detection method, not a mitigation.

Observed Examples (OBEX) Working Group Formation

- About half (61) of HW CWEs do not include an OBEX. These provide a real world example of the weakness in action, which can be helpful to new users.
- The goal is to get to 100%, but that is not practical for the CWE 4.15 release later this year. Ten to 20% is a reasonable target. Please try to contribute; it's a great opportunity to gain recognition within the community.
- CWE-1300 was presented as an example. It includes four OBEXs. That's a high bar. Let's try to get at least one for each HW CWE.

- A comment was made that the scarcity of OBEXs might be due to the fact that in the software world we have the real CVE(s) out there and they are very much evident because they can be patched later. With hardware, it's rare that it can be fixed.
- A challenge is the limited number of hardware-specific vulnerabilities in CVE. Try to use other sources other than CVE to identify OBEXs, and let us know of sources you find. Comment made to check out GitHub repositories as a source. Another comment was to look at repositories on security proceedings from conferences and journals.
- A comment was made that there may be some overlap between CVE references and a CWE OBEX.
- A GitHub page is available for HW SIG members to provide additional hardware vulnerability sources.
- Another challenge is that CVEs are not written from the CWE (root cause) perspective.
- Please try to contribute to populating more HW CWEs with an OBEX or identify sources of information for which we could generate HW OBEXs.