# HW CWE SIG Meeting
**Friday, January 12, 2024**

## Members in Attendance

| | | |
|---|---|---|
| Bob Heinemann | Alexander, Erin | Jason Oberg |
| Steven Christey Coley | Harner, Alexander | Constable, Scott D |
| Gage Hackford | Allen Krell | Ford, Thomas |
| Alec J Summers | Soheil Salehi | Aftabjahani, Sohrab |
| Bojanova, Irena V. | Jean Mousinho | Kepa, Krzysztof |
| Hallman, John | Mohan Lal | Iyer, Priya B |
| Milind Kulkarni | Abdullah Ahmad | James Pangburn |
| Mell, Peter M. | Khattri, Hareesh | Turner, Christopher A. |
| Devraj, Keerthi | Daniel DiMase | Ashrafi Gulam Mohammed |
| Monroe, Bruce | Rodriguez, David | |
| Wortman, Paul | Rafael dos Santos | |
| Forbes, Justin | Morrow, Jeremy | |

## Agenda

- Community Items
- Microarchitectural Weaknesses Update
- HW DEMOXs Added with Examples
- Composites and Chains Discussion

## Housekeeping

- Next meeting: February 9, 12:30 – 1:30 PM EST (16:30 – 17:30 UTC), MS Teams. For 2024, this meeting has been scheduled to fall on the second Friday of each month.
- 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

- Reminder: December meeting was cancelled.
- CWE Content Development Repository (CDR) pilot is now on GitHub. It is currently invite-only, with a potential public release in early 2024.
- CWE 4.14 release is planned for February 29. HACK@DAC DEMOXs and Microarchitectural Weaknesses have been a priority for this release.

**Community Items (Bob Heinemann)**

- Four hardware CWEs are missing mitigations. No change since the last time. Having a mitigation is important to the quality of a CWE.
- There are 15 HW CWEs missing a Demonstrative Example (DEMOX), a reduction of one (added by Hack@DAC for CWE-440).
- About half (61) of HW CWEs do not have an Observed Example, no change.
- In terms of an easier lift in making a contribution to CWE, especially in the Observed Examples, if you think of data that can fill these gaps, email us, or go to our GitHub page.
- We have open discussion items on GitHub, all of which need a lead:
    - Resonant frequency weakness, proposal
    - Covert Channel Coverage in HW View
    - CWE Coverage of HW Cryptography
    - Lifecycle-stage classification for HW CWEs –Dan DiMase
- Anyone interested in taking the lead on one or more of these, contact us.

**Microarchitectural Weaknesses Update (Bob Heinemann)**

- We have some microarchitectural coverage in CWE, but there are gaps.
- Took a green field approach to address the gaps, and created a working group that has identified four new submissions that are in Phase 8 (Full Submission Received status). We are temporarily referring to them as:
    - CWE A (Class): Exposure of Sensitive Information during Transient Execution.
    - CWE B (Base): Exposure of Sensitive Information in Shared Microarchitectural Structures during Transient Execution.
    - CWE C (Base): Exposure of Sensitive Information caused by Incorrect Data Forwarding during Transient Execution.
    - CWE D (Base): Exposure of Sensitive Information caused by Shared Microarchitectural Predictor State that influences Transient Execution.
- Significant progress has been made to get the submissions to Phase 8. The comment period on the submissions is open until January 19, and we are tentatively planning to meet on January 22 to finalize any comments or address any final concerns. We are working hard to get this into Release 4.14.
- A future item is CWE-1342. Need to figure out what to do with this entry when the four new submissions are in CWE. We potentially could deprecate 1342, or it may need to be modified and coexist with the existing set.

**HW DEMOXs Added with Examples (Bob Heinemann)**

- So far, we've received 22 demonstrative examples from HACK@DAC. Three were included in release 4.12, and ten were included in release 4.13. There are nine planned for release 4.14 (three staged in DEV, three are being added to DEV, and three are in editorial review).
- An example is CWE-1234 (one of the three staged in DEV):

    - The first part is the DEMOX introduction, followed by a bad code snippet. The introduction describes what the code example was trying to accomplish. The code snippet demonstrates how to lock read or write access to security-critical hardware registers (e.g., crypto keys, system integrity code, etc.).
    - The next part is text that describes what the weakness is in the code.
    - Last, there is a description of a fix or mitigation for the weakness, and example good code.
- Members are encouraged to review the DEMOXs.

**Composites and Chains Discussion (Bob Heinemann)**

- The definitions of Weakness and Vulnerability, and their relationship to one another, were reviewed. In summary, a weakness is the root cause of a vulnerability.
- There is not always a one-to-one relationship between a weakness and a vulnerability; there can also be a many-to-one or many-to-many relationship. Assuming one-to-one can lead to inconsistent mappings.
- CWE has recognized this, and developed the concepts of chains and composites to describe this behavior.
- A chain is a sequence of two or more separate weaknesses that can be closely linked together. One weakness (X) can directly create the conditions that are necessary to cause another weakness (Y) to enter a vulnerable condition. This example is a sequential relationship, but in some cases, the relationship is a tree-like structure.
    - Example: CWE-1260 is a weakness that creates the conditions for another weakness, CWE-119. In this case, we have a hardware CWE with a chain relationship with a software CWE.
- Named chains are common chains that are assigned their own CWE identifiers. There are not many. An example is CWE-680.
- A composite is a combination of weaknesses that must happen at the same time to create a vulnerability. One weakness, X, can be "broken down" into component weaknesses Y and Z. By eliminating any single component, a designer/implementer can prevent the composite from becoming exploitable.
    - Example: CWE-690.
- Understanding composite weaknesses can lead to more effective mitigations.
- SIG Comments:
    - Jason Oberg from Cycuity mentioned that prioritizing the population missing observed examples should be considered.
    - Irena Bojanova from NIST mentioned that many times, a vulnerability is a chain of weaknesses and it's good to be able to describe this chain to get the full picture of what's happening.
    - Soheil Salehi from University of Arizona spoke of a tool/framework that's been released as open source. The tool assists in trying to find missing links between CVE and CWE instances, particularly for hardware weaknesses.
        - Paper: https://arxiv.org/abs/2312.13530
        - Open-Source Code: https://gitlab.com/yuzhenglin/HW-V2W-Map