# HW CWE SIG Meeting

## Friday, November 3, 2023

### Members in Attendance

| | | |
|---|---|---|
| Gananand G Kini | Devraj, Keerthi | Khattri, Hareesh |
| Bob Heinemann | Ahmed, Faheem | Jason Oberg (Cycuity) |
| Steven M Christey | Kepa, Krzysztof | Salehi, Soheil - (ssalehi) |
| Alec J Summers | Nicole Fern | Hallman, John |
| Monroe, Bruce | James Pangburn | Ramesh, Sayee Santhosh |
| Das, Amitabh | Mohan Lal | Wortman, Paul |
| Constable, Scott D | Harner, Alexander | |
| Aftabjahani, Sohrab | Ford, Thomas | |
| Jiang Fu | Fung, Jason M | |
| Bormann, Joerg | Rafael dos Santos | |

### Agenda

- Housekeeping and Announcements
- CWE Nit Bits: Demonstrative Examples
- 4.13 Release Items for HW
- Microarchitectural Weaknesses Update
- Community Items

### Housekeeping

- Next meeting: December 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

- New CWE Content Development Repository (CDR) pilot now on GitHub! Currently invite only. Potential public release in early 2024.
- CWE 4.13 was released October 26.
- CWE 4.14 planning is ongoing. Let us know if you have any priority items. We are prioritizing the Microarchitectural Weaknesses for this release.

### CWE Nit Bits: Demonstrative Examples (Bob Heinemann)

- A demonstrative example (DEMOX) is an element in a CWE entry that's meant to illustrate the weakness through code, explanatory text, and/or diagram/table examples.

- Elements of the preferred structure of a DEMOX:
    - What the example is trying to do without mentioning the specific weakness potentially followed by an example “bad code” block.
    - What did the example do wrong, i.e., explain the weakness.
    - How it can be exploited (optional)
    - How to fix it, i.e., summarize weakness mitigation potentially followed by an example “good code” block.
- The majority of DEMOXs use coding blocks for their illustrations, so CWE has a style guide about what makes good code blocks. For example, blocks should be snippets (not fully functioning code), syntactically correct, indented correctly, and not contain multiple weaknesses. See meeting slides for other attributes.
- Refer to the meeting slides for an example: CWE-1233 (Security-Sensitive Hardware Controls with Missing Lock Bit Protection). It shows a ‘bad’ and ‘good’ DEMOX illustration; the good illustration includes text about how to fix the weakness.

**4.13 Release Items for HW (Bob Heinemann)**

- One new hardware related entry: CWE-1419: Incorrect Initialization of Resource. The entry is at the class level with five child weaknesses and adds a hardware focus to its extended description.
- Many new DEMOXs were added from HACK@DAC, providing illustrations of HW weaknesses for the following CWEs:
    - CWE-325: Missing Cryptographic Step
    - CWE-1191 (2): On-Chip Debug and Test Interface with Improper Access Control
    - CWE-1220: Insufficient Granularity of Access Control
    - CWE-1221: Incorrect Register Defaults or Module Parameters
    - CWE-1231: Improper Prevention of Lock Bit Modification
    - CWE-1241: Use of Predictable Algorithm in Random Number Generator
    - CWE-1243: Sensitive Non-Volatile Information Not Protected During Debug
    - CWE-1276: Hardware Child Block Incorrectly Connected to Parent System
    - CWE-1300: Improper Protection of Physical Side Channels
    - CWE-1326: Missing Immutable Root of Trust in Hardware
- There were also 13 OBEXs (see meeting slides) added from Top 25 Mapping and HW SIG Members.

**Microarchitectural Weaknesses Update (Ganu Kini)**

- We have four new submissions targeted for inclusion in the next CWE release:
    - CWE A: Exposure of Sensitive Information during Transient Execution
    - CWE B: Exposure of Sensitive Information in Shared Microarchitectural Structures during Transient Execution

- CWE C: Exposure of Sensitive Information caused by Incorrect Data Forwarding during Transient Execution
- CWE D: Exposure of Sensitive Information caused by Shared Microarchitectural Predictor State that influences Transient Execution.

- These submissions are in a GitHub content development repository (CDR) pilot for review, and other submissions are welcome from the group.
- Each submission has a description, extended description and observed examples (see meeting slides). Work continues to fill out the remaining fields in the CDR. The submissions will be accessible to the public in the future.
- Slide 20: Microarchitectural Weaknesses Proposed Hierarchy Changes slide proposes making the proposed CWE-A a Class level child under CWE-669: Incorrect Resource Transfer Between Spheres and CWE-B, CWE-C and CWE-D becoming base level children under CWE-A. This is because CWE-A is a more abstract case of a transient execution weakness whereas CWE-B, CWE-C and CWE-D are all more specific issues related to weakness conditions during transient execution.
- Program needs help to populate CWE entry elements, e.g., Modes of Introduction, Applicable Platforms, Common Consequences, Demonstrative Examples, and Observed Examples.
- To participate in the review and comment, send your GitHub username to cwe@mitre.org.
- Question on why CWE-669 was chosen for transient execution weaknesses. It was found that the weaknesses themselves did not relate to sensitive information storage/transfer/removal (CWE-212, CWE-226), rather it related to access control over resources that manage the sensitive information. So it was moved one level up from CWE-212 dealing with control spheres (access-control related).
- Would like to revisit this topic at a future SIG meeting.

**Community Items (Bob Heinemann)**

- Out of time.