Hardware CWE™
Special Interest Group (SIG)

Chair: Bob Heinemann (MITRE)

Co-Chair: “Manna” Parbati Kumar Manna (Intel)

MITRE Team: Gananand Kini, Steve Christey Coley, Alec Summers

MITRE

June 13, 2025

Agenda

REMINDER: This meeting is being recorded.

1	General Status of Current HW CWE Submissions	Steve C	10 min
2	Most Important Hardware Weaknesses Refresh Update	Arun K, Ganu K	10 min
3	Memory Access Related Weaknesses: Open Discussion and working exercise	Bob H to lead	30 min

Housekeeping

- **Schedule:**
  - **Next Meeting July 11:**
    - **12:30 – 1:30 PM EST (16:30 – 17:30 UTC)**
    - **Microsoft Teams**
- **Contact: cwe@mitre.org**
- **Mailing List:** *hw-cwe-special-interest-group-sig-list@mitre.org*
- **Minutes from previous meetings available on our GitHub site:**
  - **https://github.com/CWE-CAPEC/hw-cwe-sig**

Announcements

- **MIHW Poll Extended – Help Shape the Future of Hardware Security**
  - Now closing Monday, June 23rd, 2025, before 11:59 PM (Pacific Time)
  - https://forms.office.com/g/ytaKxw2bEx
- **CWE Content Development Repository (CDR) is now fully public.**
  - Enables the broader community to view, track, and contribute to entries submissions.
  - Content suggestions begin with the CWE Submission Form.
  - CDR can be accessed here:
    - https://github.com/CWE-CAPEC/CWE-Content-Development-Repository/

General Call for Topics

- **Any news items to share with the group?**
- **Any high priority topics we should address today?**
- **Future topics?**

Hardware Submissions in Progress

- **Initial Consultation (Stage 1, Phase 4 – common bottleneck)**
  - HW/SW: ES2306-c0b52346 Use of a Quantum-Vulnerable Cryptographic Algorithm (NIST)
- **Detailed Review (Stage 2, Phase 9)**
  - ES2208-9fb81a1a Speculative propagation of requests for transaction before data validation in multi-manager bus architectures (Francesco Restuccia)
- **Acknowledged Receipt (Stage 1, Phase 2 – Ack-Receipt)**
  - ES2503-821d9ec2 - CPU Control bits are used without validation of legality
  - ES2505-40961dd8 - Logic Transformation Discrepancy during RTL Synthesis
  - ES2411-411508bb - TLB Flush on Context Switch
- **2 early submissions that look like vulnerability reports**
  - ES2503-40a2092d - Improper Enforcement of SUM Bit in RISC-V mstatus CSR
  - ES2504-01bf6a89 - Clock glitch attack on RISC-V softprocessor core
- **See CDR: https://github.com/CWE-CAPEC/CWE-Content-Development-Repository/issues**

Most Important Hardware Weaknesses List Update

Arun K., Ganand K.

Most Important Hardware Weaknesses Poll Part 1

- **You can help shape the future of Hardware Security!**
- **Why Participate?**
  - **Global Impact**: Your expert insights can help shape hardware security priorities at an industry-wide scale.
  - **Recognition**: You can opt to have your name publicly acknowledged.
  - **Quick and Easy**: The poll consists of just **2 mandatory questions**.
- **So far received 8 responses. Thank you to those that contributed!**
- **We need YOUR voice to ensure community's collective expertise is fully represented: https://forms.office.com/g/ytaKxw2bEx**

Most Important Hardware Weaknesses Refresh – June 2025 Update

- **The deadline for the poll part 1 has been extended to Monday June 23, 2025 by 11:59 PM PT.**

- **Schedule for Expert Opinion survey :**
  - Conduct First Expert Opinion Poll (Jun 4, 2025 - Jun 23, 2025)
  - Conduct Second Expert Opinion Poll (Jun 26, 2025 - Jul 11, 2025)
    - Jun 26 is the MIHW working group meeting
    - Jul 10 is the MIHW working group meeting
    - Jul 11 is the HW CWE SIG meeting
  - Finalize MIHW List after applying methodology (WS: Jul 17, 2025)
  - July 25 – Ganu last day before PTO
  - Coordinate and push communications for the final list with MITRE, member organizations. (Jul 31, 2025)

Most Important Hardware Weaknesses Refresh – June 2025 Update contd…

- **MIHW Working Group looking at how to generate the list of CWEs for the second poll (Part 2):**
  - Take the top N of CWEs in **descending** order of frequency/count from data analysis list.
  - Take the top N of CWEs in **descending** order of frequency/count from CWEs to Include list.
  - Take the top N of CWEs in **ascending** order of frequency/count from CWEs to Exclude list.
- **Second poll (Part 2) will ask participants to rate (on a Likert scale) each of these for inclusion/exclusion from the MIHW List.**
- **MIHW working group to perform analysis on this final data, generate the new list and finally publish it.**

CWE is sponsored by U.S. Department of Homeland Security (DHS) Cybersecurity and Infrastructure Security Agency (CISA).

MIHW List Discussion

- **Please continue discussion on the HW CWE Mailing List (see below).**
  - You can tag email subject lines using "[MIHW]" to get more visibility.

- **Mailing List:** *hw-cwe-special-interest-group-sig-list@mitre.org*
  - ***NOTE: All mailing list items are archived publicly at:***
    - *https://www.mail-archive.com/hw-cwe-special-interest-group-sig-list@mitre.org/*

Memory Access Related Weaknesses

Open Discussion

Memory Weakness for HW Recap

- Out-of-bounds access weaknesses are well-defined in software, they are not as clearly defined in hardware.
- Differences in how out-of-bounds accesses behave between simulation and synthesized circuits may lead to security consequences.
- **Out-of-bounds Reads** may return an undefined value in simulation but in a synthesized circuit it could return data from an unexpected location.
- **Out-of-bounds Writes** in simulation are treated as no-ops, but in synthesized hardware they could modify unexpected locations.
- Existing memory weaknesses could be updated and added to the hardware view.
- It can be problematic to create HW version of existing entries and to do so would need an exceptional justification. – Steve C
- It will not be possible to distinguish CVEs referencing memory-related CWEs as HW or SW issues if existing memory CWEs are included in the HW CWE view. – Jason Oberg
- Hareesh had commented “We should also consider adding CWE-190: Integer Overflow or Wraparound”

Memory Related CWEs that are not in HW View

	CWE ID	Name	Description	HW Consequences
Jason O	CWE-125	**Out-of-bounds Read**	The product reads data past the end, or before the beginning, of the intended buffer.	• In simulation, OOB read returns 'X' and is undefined • In a synthesized circuit, they OOB read could expose sensitive information
	CWE-124	**Buffer Underwrite ('Buffer Underflow')**	The product writes to a buffer using an index or pointer that references a memory location prior to the beginning of the buffer.	• In simulation, an OOB write is essentially a "no-op" and the buffer isn't updated • In a synthesized circuit, the OOB write may compromise the integrity of the buffer
	CWE-787	**Out-of-bounds Write**	The product writes data past the end, or before the beginning, of the intended buffer.	• In simulation, an OOB write is essentially a "no-op" and the buffer isn't updated • In a synthesized circuit, the OOB write may compromise the integrity of the buffer
	CWE-786	**Access of Memory Location Before Start of Buffer**	The product reads or writes to a buffer using an index or pointer that references a memory location prior to the beginning of the buffer.	• Similar impact as above
Hareesh	CWE-190	**Integer Overflow or Wraparound**	The product performs a calculation that can produce an integer overflow or wraparound when the logic assumes that the resulting value will always be larger than the original value. This occurs when an integer value is incremented to a value that is too large to store in the associated representation. When this occurs, the value may become a very small or negative number.	

Working Exercise

- Let’s review CWE-125: Out-of-bounds Read to see what we would need to update to make it inclusive of hardware.

- https://docs.google.com/document/d/18USrLkoXb8iLAREutRPAG8I7iEpL61wt5g0QTdCn-Vs/edit?usp=sharing

Next Meeting (July 11)

CWE@MITRE.ORG

- Mailing List: *hw-cwe-special-interest-group-sig-list@mitre.org*
    - ***NOTE: All mailing list items are archived publicly at:***
        - *https://www.mail-archive.com/hw-cwe-special-interest-group-sig-list@mitre.org/*
- What would members of this body like to see for the next HW SIG agenda?
- Questions, Requests to present? Please let us know.

Backup

Memory Access Related Weaknesses

Jason Oberg

Introduction

Overview

- Several CWEs exist related to out-of-bound (OOB) buffer access that are not in the hardware view.
- In hardware, buffers manifest themselves as memories, register arrays, etc.

Impact of OOB Buffer Access in HW

- OOB access to buffers in RTL simulation results in undefined behavior (usually an ‘X’ for reads or no changes for writes)
- In real synthesized logic, data *will* be read/written and the actual result of that may be unknown to the hardware designer.

Proposal

- Move relevant CWEs to the hardware view or provide a comparable CWE in HW view

Memory Related CWEs that are not in HW View

CWE ID	Name	Description	HW Consequences
CWE-125	**Out-of-bounds Read**	The product reads data past the end, or before the beginning, of the intended buffer.	• In simulation, OOB read returns 'X' and is undefined • In a synthesized circuit, they OOB read could expose sensitive information
CWE-124	**Buffer Underwrite ('Buffer Underflow')**	The product writes to a buffer using an index or pointer that references a memory location prior to the beginning of the buffer.	• In simulation, an OOB write is essentially a "no-op" and the buffer isn't updated • In a synthesized circuit, the OOB write may compromise the integrity of the buffer
CWE-787	**Out-of-bounds Write**	The product writes data past the end, or before the beginning, of the intended buffer.	• In simulation, an OOB write is essentially a "no-op" and the buffer isn't updated • In a synthesized circuit, the OOB write may compromise the integrity of the buffer
CWE-786	**Access of Memory Location Before Start of Buffer**	The product reads or writes to a buffer using an index or pointer that references a memory location prior to the beginning of the buffer.	• Similar impact as above

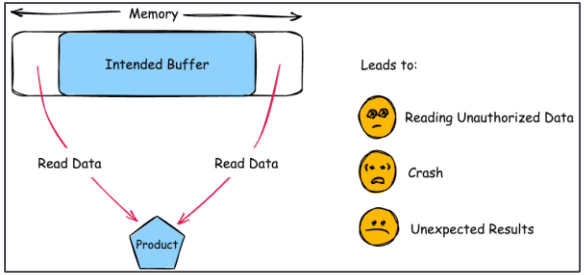
Hardware Example:
CWE-125 Out-of-bounds Read

Description:

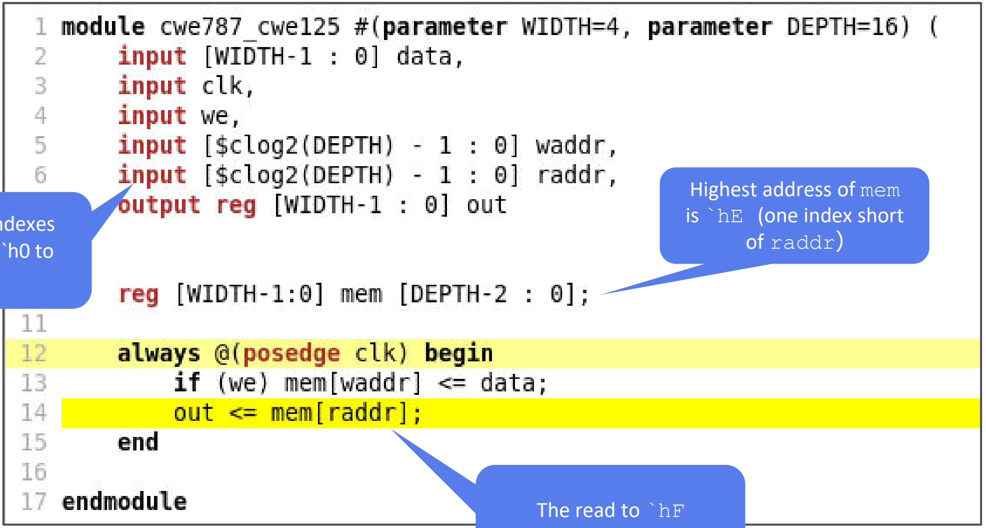
- The product reads data past the end, or before the beginning, of the intended buffer.

Hardware Consequences:

- OOB read in hardware is undefined and will return 'X' in simulation
- In a synthesized circuit, the data may be read from unexpected locations, potentially leaking data



https://cwe.mitre.org/data/definitions/125.html



```
module cwe787_cwe125 #(parameter WIDTH=4, parameter DEPTH=16) (
    input [WIDTH-1 : 0] data,
    input clk,
    input we,
    input [$clog2(DEPTH) - 1 : 0] waddr,
    input [$clog2(DEPTH) - 1 : 0] raddr,
    output reg [WIDTH-1 : 0] out


    reg [WIDTH-1:0] mem [DEPTH-2 : 0];

    always @(posedge clk) begin
        if (we) mem[waddr] <= data;
        out <= mem[raddr];
    end

endmodule
```

`raddr` indexes between `h0 to `hF

Highest address of `mem` is `` `hE `` (one index short of `raddr`)

The read to `` `hF `` results in an OOB read

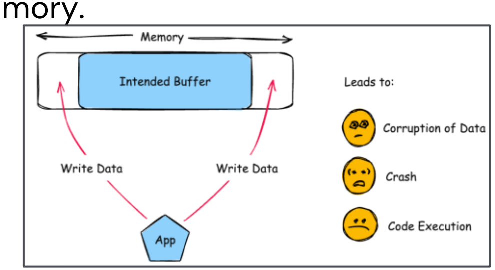
Hardware Example:
CWE-787 Out-of-bounds Write

Description:

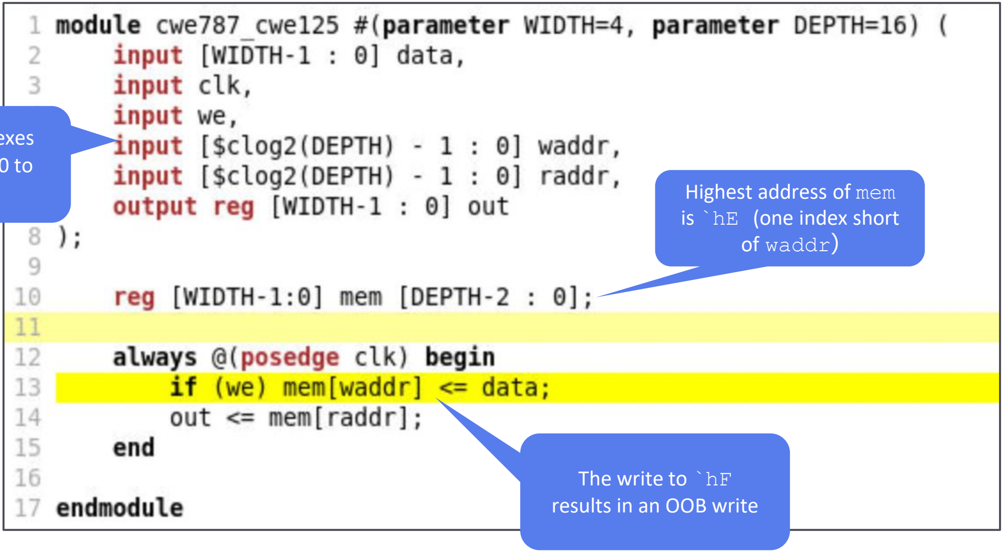
- The product writes data past the end, or before the beginning, of the intended buffer.

Hardware Consequences:

- OOB write in hardware is undefined and will basically not update the buffer/memory in *simulation*
- In real synthesized logic, a write may occur and could compromise the integrity of data in the memory.



https://cwe.mitre.org/data/definitions/787.html



```
module cwe787_cwe125 #(parameter WIDTH=4, parameter DEPTH=16) (
    input [WIDTH-1 : 0] data,
    input clk,
    input we,
    input [$clog2(DEPTH) - 1 : 0] waddr,
    input [$clog2(DEPTH) - 1 : 0] raddr,
    output reg [WIDTH-1 : 0] out
);

    reg [WIDTH-1:0] mem [DEPTH-2 : 0];

    always @(posedge clk) begin
        if (we) mem[waddr] <= data;
        out <= mem[raddr];
    end

endmodule
```

waddr indexes between `h0 to `hF

Highest address of mem is `hE (one index short of waddr)

The write to `hF results in an OOB write

Questions for Discussion

- Have others in the community encountered similar issues in hardware?
- The contents of these CWEs are software centric. Would it make sense to update them to include some hardware examples before inclusion in the HW view?