Hardware CWE™ Special Interest Group (SIG)

Bob Heinemann, Gage Hackford, Steve Christey Coley, Alec Summers

MITRE

May 10, 2024

Agenda

REMINDER: This meeting is being recorded.

1	Search for a HW SIG Co-Chair	Bob H	10 min
2	Comparing the Hardware and Software Vulnerability Management Infrastructures	Peter Mell (NIST)	30 Min
3	HW Covert Channels Discussion	Bob H	20 Min

Housekeeping

- **Schedule:**
  - **Next Meeting: June 14**
    - **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

- **Reminder that the April meeting was cancelled.**
- **CWE Content Development Repository (CDR) pilot now on GitHub! Open to anyone by request. Public access in the next few months.**
- **CWE 4.15 release will be around June/July.**

HW SIG Co-Chair

Co-Chair Role

Looking for a partner to help advance the HW SIG

Role Expectations:

- Encouraging HW SIG members to share their thoughts and getting the community to suggest important topics to tackle
- Forming ad-hoc working groups for content creation and updates
- Leading technical discussions
- Agenda development for meetings
- Preferences: Active attendance at HW SIG meetings and contribution to CWE

Process for selection:

- Nominations close on Friday May 17
- Short discussion with nominees
- Make selection by Friday May 31

Peter Mell's Presentation

Your Hardware has Bugs -

Comparing the Hardware and Software Vulnerability Management Infrastructures

Peter Mell, Computer Security Division
Irena Bojanova, Software and Systems Division

Computer Scientists
National Institute of Standards and Technology

Historically hardware was viewed as an ‘immutable root-of-trust'

- Is this because greater care is taken in designing hardware?
- Software has never achieved this
  - Large IT company had the slogan “Unbreakable”, “Can't break it, can't break in”, (2002)
- But…
- Hardware is made with code and contains code
  - Hardware design (Verilog, VHDL), code etched on a chip
  - Hardware contains microcode and firmware
  - Worse than software, hardware often can’t be updated/patched/fixed
- Hardware is more than software; it can have physically related vulnerabilities
- Hardware is complex
  - Systems on a chip (SOCs) contain many integrated modules, designed independently
  - Continually increasing functionality in products ensures instability/insecurity

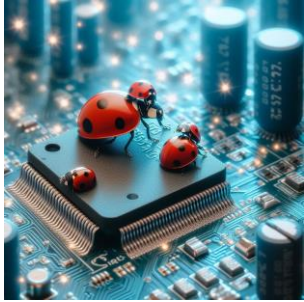
Software Vulnerability Landscape

- There are estimates for between 15 and 50 bugs per 1000 lines of **delivered** code
  - Some of those bugs will affect security
- 28,000 security vulnerabilities published in 2023

- Software has a mature ecosystem to handle the vulnerabilities
  - Common Vulnerabilities and Exposures (CVE)
  - National Vulnerability Database (NVD)
  - Common Vulnerability Scoring System (CVSS)
  - Exploit Prediction Scoring System (EPSS)
  - Known Exploited Vulnerabilities (KEV)
  - Common Weakness Enumeration (CWE)
  - NIST Bugs Framework

- 130 weakness types (i.e., CWEs) cover 94% of vulnerabilities (i.e., CVEs)
  - These are ways in which software has security failures

Hardware Vulnerability Landscape



- 108 hardware weakness types identified (i.e., CWEs)
- Only half of these have observed examples (i.e., CVEs)
  - Hardware producers don't always disclose, in part because they can't patch
  - Many may have surfaced during development and were fixed pre-production
- We know of only 131 published hardware vulnerabilities (i.e., CVEs)
- Only 3 of the hardware weaknesses (CWEs) overlap with software weaknesses
  - Is hardware security really that different from software?
  - 105 of the hardware weaknesses are hardware specific
  - Do the rest of the 127 software weaknesses apply (130 software vulns – 3 overlap)?
    - After all, hardware contains code

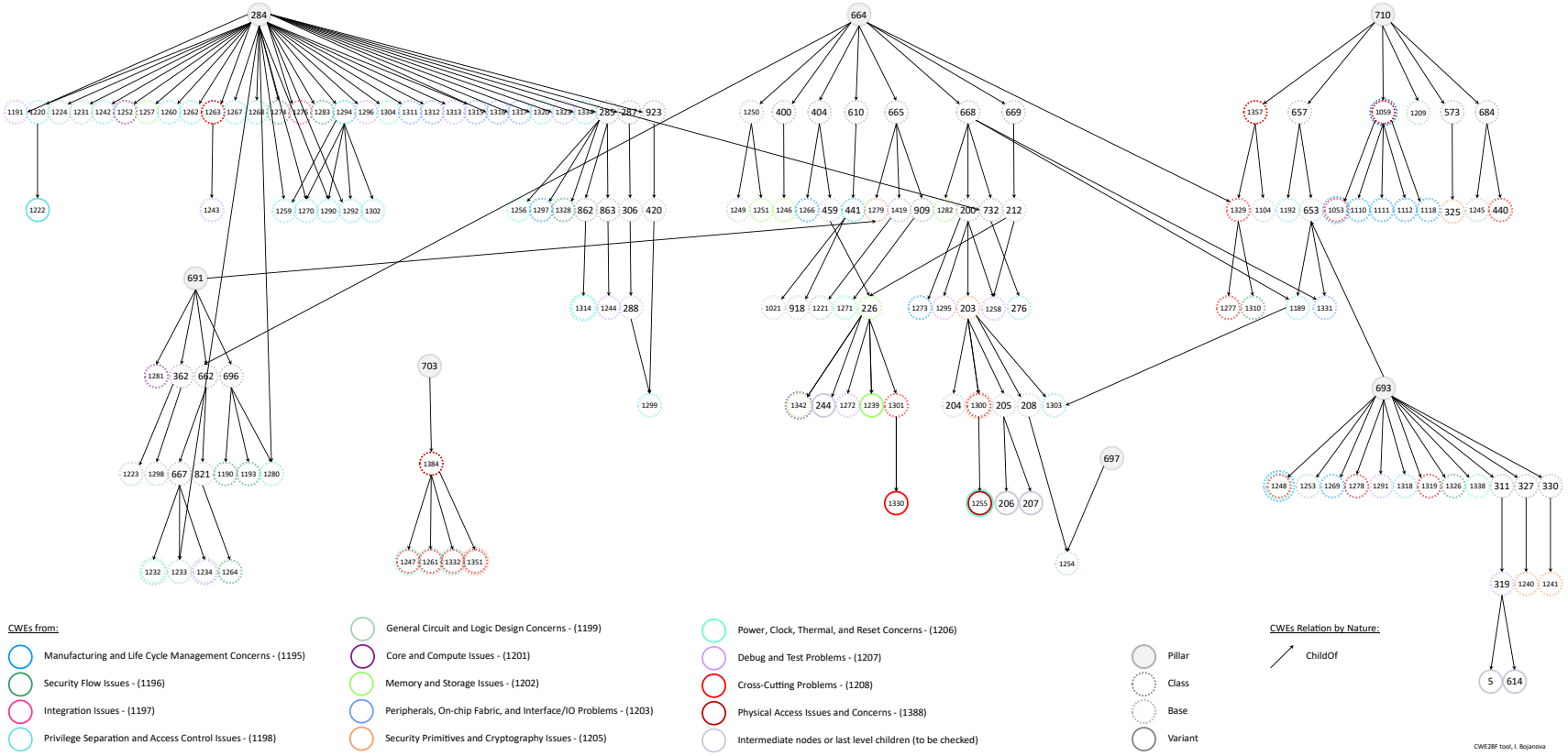
- Trust-Hub Vulnerability Database (University of Florida)
  - 38 physical attacks
  - 23 vulnerabilities (look more like weaknesses)

Hardware Weakness Categories

This is mostly ‘where’ they occur

1. Core and Compute Issues (CWE-1201)
2. Cross-Cutting Problems (CWE-1208)
3. Debug and Test Problems (CWE-1207)
4. General Circuit and Logic Design Concerns (CWE-1199)
5. Integration Issues (CWE-1197)
6. Manufacturing and Life Cycle Management Concerns (CWE-1195)
7. Memory and Storage Issues (CWE-1202)
8. Peripherals, On-chip Fabric, and Interface/IO Problems (CWE-1203)
9. Physical Access Issues and Concerns (CWE-1388)
10. Power, Clock, Thermal, and Reset Concerns (CWE-1206)
11. Privilege Separation and Access Control Issues (CWE-1198)
12. Security Flow Issues (CWE-1196)
13. Security Primitives and Cryptography Issues (CWE-1205)

NIST Research: Hardware Weakness Hierarchies Looking at ‘how’ they are exploited



Looking Towards the Future

- Much of the software vulnerability management infrastructure can support hardware vulnerabilities
- Progress is being made in this direction
- Hardware contains weaknesses not found in software
  - Importance of the physical dimension
- Hardware should share more than three weaknesses with software
  - but this has yet to be determined

Presentation / Speaker Information

Title: Your Hardware has Bugs - Managing Hardware Vulnerabilities

Abstract: Hardware has historically been viewed as trustworthy while software has never achieved this. However, hardware is vulnerable; it has additional complications and weaknesses not present in software. Software, while massively vulnerable, is supported by an extensive management ecosystem. Hardware can leverage this ecosystem and work is being done to promote such support.

Peter Mell Biography: Peter Mell is a computer scientist with the National Institute of Standards and Technology. He has conducted computer security research for over 25 years and has over 65 academic publications. He created and managed the National Vulnerability Database (NVD). He assisted in the development of two versions of the Common Vulnerability Scoring System (CVSS). He is currently investigating hardware weaknesses, how they can be exploited, where they occur, and what damage can be done.

Irena Bojanova Biography: Irena Bojanova is a computer scientist at the National Institute of Standards and Technology. She has conducted formal methods and computer security research for over 35 years and has over 80 academic publications. She invented and is creating the NIST Bugs Framework (BF), which goal is formalization of software security weaknesses and vulnerabilities. She is the Editor of the Cybersecurity Department of IEEE *IT Professional* magazine and a member at large of the IEEE Computer Society Publications Board Executive Committee.

Covert Channel Coverage in CWE

CWE-514: Covert Channel

https://cwe.mitre.org/data/definitions/514.html

Abstraction: Class

Description: A covert channel is a path that can be used to transfer information in a way not intended by the system's designers[1].

Extended Description: Typically, the system has not given authorization for the transmission and has no knowledge of its occurrence.

Relationships:

ChildOf	CWE-1229:Creation of Emergent Resource
ParentOf	CWE-385:Covert Timing Channel
ParentOf	CWE-515: Covert Storage Channel
CanFollow	CWE-205: Observable Behavioral Discrepancy

1. Landwehr, C. E., Bull, A. R., McDermott, J. P., & Choi, W. S. (September 1994). A Taxonomy of Computer Program Security Flaws, with Examples. Information Technology Division, Code 5542, Naval Research Laboratory. Retrieved from https://dl.acm.org/doi/10.1145/185403.185412

Vulnerability Mapping Notes:

Usage: Allowed-with-Review; **Reason:** Abstraction

Rationale: This CWE entry is a Class and might have Base-level children that would be more appropriate; **Comments:** Examine children of this entry to see if there is a better fit.

NOTE: Nothing about EM based Covert Channels, nor HW cause, e.g., SMPS

Covert Channel Discussion

Previous HW SIG Member Comments:

- Covert Channels should have coverage in the hardware view *–Jason Oberg*
- Covert Channels should be in the HW categories Security Flow Issues, General Circuit and Logic Design Concerns, or Debug and Test Problems. *–Paul Wortman*
- CWE-514 as currently written it's specific to software and would need to be tweaked *–Bruce Monroe*

Questions:

- Does CWE-514 need to be modified to address more HW centric concerns?
- Is CWE-514 really a weakness or attacker focused?
- Coverage in HW View?
- Other questions about covert channels and HW CWE?

Discussion

https://github.com/CWE-CAPEC/hw-cwe-sig/issues/108

Covert Channels vs Side Channels Notes

Covert Channel (CC) / Side Channel (SC)

- Intentional transmission (CC). Accidental transmission (SC) – *Ross Anderson* [1]
- Adversary controls input and output (CC). Adversary can only read output (SC) – *Intel* [2]
- Not an intended resource but exists due the application's behaviors. –*CWE-514 Notes* [3]

1. https://www.cl.cam.ac.uk/~rja14/Papers/SEv3-ch19-7sep.pdf
2. https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/best-practices/refined-speculative-execution-terminology.html
3. https://cwe.mitre.org/data/definitions/514.html

Intel - Covert Channels Notes

- Incidental channels are unintended communication channels
- Inevitable due to sharing of resources
- Cannot have advanced features without shared resources
- Not feasible to remove
- Covert Channels are types of Incidental Channels
- Incidental covert channels require an adversary to not only control both ends of the incidental channel but also have access to secret information.

Incidental Channels could be Child of or Sibling to?

CWE-1229: Creation of Emergent Resource

Weakness ID: 1229
Vulnerability Mapping: ALLOWED (with careful review of mapping notes)
Abstraction: Class

View customized information: Conceptual | Operational | Mapping Friendly | **Complete** | Custom

Description

The product manages resources or behaves in a way that indirectly creates a new, distinct resource that can be used by attackers in violation of the intended policy.

Extended Description

A product is only expected to behave in a way that was specifically intended by the developer. Resource allocation and management is expected to be performed explicitly by the associated code. However, in systems with complex behavior, the product might indirectly produce new kinds of resources that were never intended in the original design. For example, a covert channel is a resource that was never explicitly intended by the developer, but it is useful to attackers. "Parasitic computing," while not necessarily malicious in nature, effectively tricks a product into performing unintended computations on behalf of another party.

Relationships

Relevant to the view "Research Concepts" (CWE-1000)

Nature	Type	ID	Name
ChildOf	P	664	Improper Control of a Resource Through its Lifetime
ParentOf		514	Covert Channel

Applicable Platforms

Languages

Class: Not Language-Specific *(Undetermined Prevalence)*

Operating Systems

Class: Not OS-Specific *(Undetermined Prevalence)*

Architectures

Intel – Incidental Channels

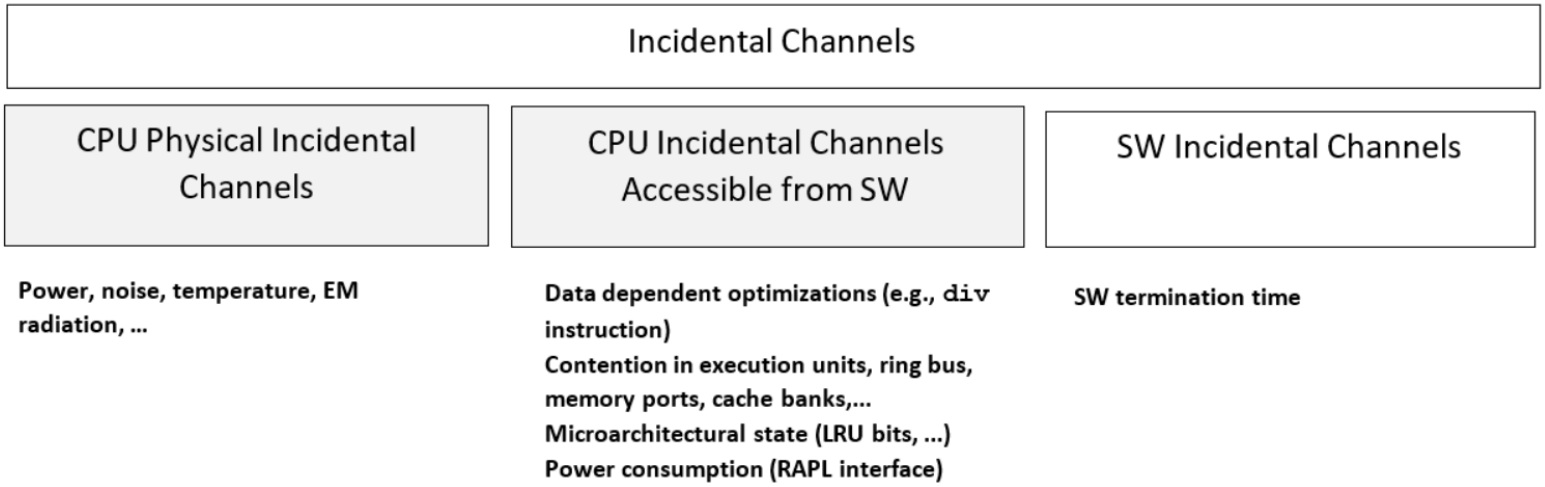


Figure 1: Example of taxonomy terminology of incidental channels on CPUs

Next Meeting (June 14)

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

Addressing Observed Example Gaps for HW CWE

OBEX Gaps Agenda

- **Current Need**
- **OBEX Challenges**
- **Participation and Organizing Work**
- **How you can help**

Observed Examples Working Group

- **Goal:** Ensure every HW CWE has at least one OBEX.
  - The OBEX element is important in that it links real-world examples to weaknesses.
  - 61 (52%) of HW CWEs lack observed examples.
  - Your contribution can reduce this gap.

- **Timeframe and Commitment:** Next release cycle, approx., March through June.
  - We value your time and expertise, so would like to structure the workgroup so that participation can be flexible and asynchronous.
  - Target generating OBEXs for 10% - 20% of the HW CWEs missing OBEXs for this release.

- **Recognition:** All contributors will be acknowledged for their valuable input.
  - This is a great opportunity to gain recognition within the community and add a contribution to your professional portfolio.

CWE-1300:
Improper Protection of Physical Side Channels

Description

The device does not contain sufficient protection mechanisms to prevent physical side channels from exposing sensitive information due to patterns in physically observable phenomena such as variations in power consumption, electromagnetic emissions (EME), or acoustic emissions.

Extended Description

An adversary could monitor and measure physical phenomena to detect patterns and make inferences, even if it is not possible to extract the information in the digital domain.

Physical side channels have been well-studied for decades in the context of breaking implementations of cryptographic algorithms or other attacks against security features. These side channels may be easily observed by an adversary with physical access to the device, or using a tool that is in close proximity. If the adversary can monitor hardware operation and correlate its data processing with power, EME, and acoustic measurements, the adversary might be able to recover of secret keys and data.

Observed Examples

Reference	Description
CVE-2021-3011	electromagnetic-wave side-channel in security-related microcontrollers allows extraction of private key
CVE-2013-4576	message encryption software uses certain instruction sequences that allows RSA key extraction using a chosen-ciphertext attack and acoustic cryptanalysis
CVE-2020-28368	virtualization product allows recovery of AES keys from the guest OS using a side channel attack against a power/energy monitoring interface.
CVE-2019-18673	power consumption varies based on number of pixels being illuminated in a display, allowing reading of secrets such as the PIN by using the USB interface to measure power consumption

OBEX Challenge #1 – Specific to HW CWE

- **CVE has limited coverage for hardware specific vulnerabilities**
- **Examples:**
  - **Did not find a CVE that could be mapped to the above CWEs**
    - CWE-1351: Improper Handling of Hardware Behavior in Exceptionally Cold Environments did not turn up any results
    - CWE-1338: Improper Protections Against Hardware Overheating
    - CWE-1334: Unauthorized Error Injection Can Degrade Hardware Redundancy
  - **Found CVE that may map but not enough details to be sure**
    - CWE-1328: Security Version Number Mutable to Older Versions
  - **Found CVE that map**
    - CWE-1326:Missing Immutable Root of Trust in Hardware
    - CVE-2022-38773, CVE-2022-28383, CVE-2023-22955

How to help: HW Vulnerability Sources

- **Since CVE has limited coverage for hardware vulnerabilities, we need to consult other sources for observed examples.**
- **We like to tap into the collective knowledge and expertise of this group.**
- **Please provide sources other than CVE that we could use to pull observed examples from for HW CWEs.**
- **Preferably site that aggregate vulnerability reports.**
- **[WORKING ITEM] Sources for Hardware Vulnerability Reports**
  - https://github.com/CWE-CAPEC/hw-cwe-sig/issues/109

Observed Example Element

- Contains one or more publicly reported vulnerabilities in real-world products that exhibit the weakness.

- **Sub-Elements**
  - **Reference**: This contains ~~the CVE Identifier, e.g., CVE-2005-1951~~ a vulnerability identifier.
  - **Description**: Clear, simple, and concise summary of ~~CVE~~ vulnerability report that focuses on the link between the ~~CVE~~ report and weakness. Exclude product name, attack vectors and other irrelevant details.
  - **Link**: URL Link to ~~CVE~~ vulnerability report. ~~Preferably from https://www.cve.org/.~~

OBEX Challenge #2 – General CWE Challenge

- **Vulnerability reports are not written from a weakness aspect**
  - The original weakness is not always covered.
  - From a vulnerability management perspective, the underlying weakness may not actually be important to the organization.
  - Reports emphasize product impact, product versions affected, and how easy it is for an attacker to exploit.
  - Routinely we see that many of the vulnerability descriptions do not have enough information to determine what the underlying weakness is.
  - For OBEXs we cannot infer the weakness, the vulnerability report must specifically describe the weakness.

How to help: Submitting an OBEX

1. **Find HW CWE(s) you would like to contribute an OBEX.**
   - There is a list maintained on our GitHub.
   - There is an issue per HW CWE that is missing an OBEX.
   - NVD Data may be a help here
2. **Assign yourself to the GitHub Issue.**
3. **In the Issue comments, provide a CVE Number, URL, description, and how you would like to be cited for the contribution.**
   - Preferred Name and organization.

Issues are here:

https://github.com/CWE-CAPEC/hw-cwe-sig/labels/Missing%20OBEX

Note: We are planning to release a OBEX Style guide soon

Summary

Our ask

1. **Commit to supporting the working group.**
   - Send an email to cwe@mitre.org, subject: OBEX WG.

2. **Provide additional HW Vulnerability Sources.**
   - https://github.com/CWE-CAPEC/hw-cwe-sig/issues/109

3. **Generate 1 OBEX or many OBEXs.**
   - https://github.com/CWE-CAPEC/hw-cwe-sig/labels/Missing%20OBEX

Open *Community Items*

HW CWE's With Missing:
DEMOX's, OBEX's and Mitigations

- **Missing Mitigations**
  - 4 HW CWEs are missing mitigations (No change)
- **Missing Detection Methods**
  - How many do we have? CREATE A TRACKER
- **Missing demonstrative examples (DEMOX)**
  - 15 HW CWEs missing demonstrative examples (down 1)
    - 1 added from Hack@DAC, CWE-440
    - Note: there are other DEMOXs from Hack@DAC but now adding DEMOXs to entries that have an existing DEMOX
  - How many DEMOX's are not code based? CREATE A TRACKER

https://github.com/CWE-CAPEC/hw-cwe-sig/issues

Discussion Items on GitHub

Resonant frequency weakness, proposal (Topic Lead: OPEN)	• https://github.com/CWE-CAPEC/hw-cwe-sig/issues/105
Covert Channel Coverage in HW View (Topic Lead: OPEN)	• https://github.com/CWE-CAPEC/hw-cwe-sig/issues/108
CWE Coverage of HW Cryptography (Topic Lead: OPEN)	• https://github.com/CWE-CAPEC/hw-cwe-sig/issues/7
Lifecycle-stage classification for HW CWEs –Dan DiMase (Topic Lead: OPEN)	• https://github.com/CWE-CAPEC/hw-cwe-sig/issues/4

Most Important Hardware Weaknesses Refresh

Bob H

Most Important Hardware Weaknesses (MIHW)

- **Is this something worth revisiting?**
- **Part of CWE 4.6 Release, October 28, 2021**
- **Have there been substantial developments since the last release of MIHW?**
- **Would those affect the rankings and inclusions of the list in any meaningful way?**

Current MIHW

CWE-1189	Improper Isolation of Shared Resources on System-on-a-Chip (SoC)
CWE-1191	On-Chip Debug and Test Interface With Improper Access Control
CWE-1231	Improper Prevention of Lock Bit Modification
CWE-1233	Security-Sensitive Hardware Controls with Missing Lock Bit Protection
CWE-1240	Use of a Cryptographic Primitive with a Risky Implementation
CWE-1244	Internal Asset Exposed to Unsafe Debug Access Level or State
CWE-1256	Improper Restriction of Software Interfaces to Hardware Features
CWE-1260	Improper Handling of Overlap Between Protected Memory Ranges
CWE-1272	Sensitive Information Uncleared Before Debug/Power State Transition
CWE-1274	Improper Access Control for Volatile Memory Containing Boot Code
CWE-1277	Firmware Not Updateable
CWE-1300	Improper Protection of Physical Side Channels

New HW CWEs Since MIHW

- **CWE-1342: Information Exposure through Microarchitectural State after Transient Execution**
- **CWE-1357: Reliance on Insufficiently Trustworthy Component**
- **CWE-1384: Improper Handling of Physical or Environmental Conditions**
- **CWE-1388: Physical Access Issues and Concerns**

Discussion

- **Have there been substantial developments since the last release of MIHW?**
- **Would those affect the rankings and inclusions of the list in any meaningful way?**
- **Are there observational trends that would change the current list in any significant and meaningful way?**