Hardware CWE™ Special Interest Group (SIG)

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

MITRE

June 14, 2024

Agenda

REMINDER: This meeting is being recorded.

1	HW SIG Co-Chair Announcement	Bob H	10 min
2	Usability Mockups	Bob	5 Min
3	Mapping Diagram Demo	Bob	5 Min
4	Covert Channel Discussion (Continued)	Manna / Bob	30 Min
5	System Verilog as a Language in Schema	Bob	If time allows

Housekeeping

- **Schedule:**
  - **Next Meeting: July 12**
    - **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

- **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 on July 16.**

HW SIG Co-Chair Announcement and Introduction

Welcome New HW SIG Co-Chair

Parbati Kumar Manna is a Principal Security Researcher working at the networking group at Intel, the company he joined in 2008 after completing his Masters and PhD in Computer Science from University of Florida, with his dissertation concentrating on network security. He obtained his Bachelor of Technology from Indian Institute of Technology, Kharagpur in 1997. He led the teams responsible for the security of various flagship Intel products including Big core, Atom core, Chipset, Graphics and Xeon Phi. He has published in leading academic and hacking conferences and journals including Black Hat, DEFCON, INFOCOMM, ICDCS, HOST, Transactions on Networking etc. He is the author of the open-source universal parsing tool ParseAndC. Dr. Manna is also an amateur standup comedian, musician, poet, actor and director. One of his stories was made into a movie that got screened at the NABC 2022 International Short Film Festival at Las Vegas.

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

Usability Mockups

Usability Task Micro Updates Initial Goals

Above the fold (before the webpage scroll point):

- **Important and concise text is above the fold so the reader can easily scan and digest**

Points to Make Above the Fold:

- **Describe just the weakness and provide a visual aid**
  - Concise summary of weakness (only in description, no extended description)
  - Describe the condition and some context about the condition
  - Concise is 3 – 4 sentences. Images will provide additional context.

Reorder Elements:

- **Alternate Terms**
- **Consequences Element** (bad outcomes)
- **Mitigations Element (**what to do about the weakness)
- Remaining Elements follow

CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

Weakness ID: 22

Vulnerability Mapping: ALLOWED

Abstraction: Base

Description

The product uses external input to construct a pathname that is intended to identify a file or directory that is located underneath a restricted parent directory, but the product does not properly neutralize special elements within the pathname that can cause the pathname to resolve to a location that is outside of the restricted directory.

Extended Description

Many file operations are intended to take place within a restricted directory. By using special elements such as ".." and "/" separators, attackers can escape outside of the restricted location to access files or directories that are elsewhere on the system. One of the most common special elements is the "../" sequence, which in most modern operating systems is interpreted as the parent directory of the current location. This is referred to as relative path traversal. Path traversal also covers the use of absolute pathnames such as "/usr/local/bin", which may also be useful in accessing unexpected files. This is referred to as absolute path traversal.

In many programming languages, the injection of a null byte (the 0 or NUL) may allow an attacker to truncate a generated

CWE-22: Improper Limitation of a Pathname to a Restricted Directory

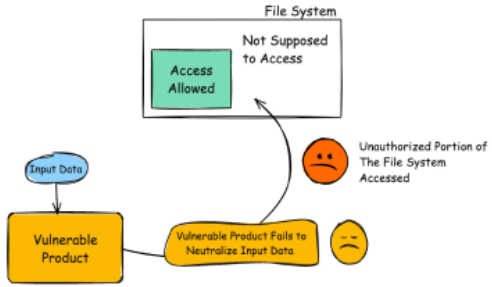
Weakness ID: 22

Vulnerability Mapping: ALLOWED

Abstraction: Base

Description

The product constructs a pathname for file or directory identification using external input, but fails to neutralize special elements within the pathname. This could lead to the pathname resolving to a location outside the restricted directory. Special elements, such as "../", can be used to escape the restricted location and access files or directories elsewhere on the system, a situation known as relative path traversal. The issue also extends to absolute path traversal, where absolute pathnames like "/usr/local/bin" are used. Additionally, the injection of a null byte in some programming languages may truncate a generated filename, potentially broadening the scope of attack.



Alternate Terms

Directory traversal

Path traversal: "Path traversal" is preferred over "directory traversal," but both terms are attack-focused.

Summary

- There is an effort to make the CWE entries easier to use and consume
- Initial effort is focused on CWE Top 25 entries
- Currently there are no plans for HW CWEs
- A current set of mockups are available for public comment
  - https://drive.google.com/drive/folders/1NqYbkZcyXE7xzlhyADFFRjHXpuKvV01Q?usp=drive_link

Mapping Diagram / Decision Tree Demo

CWE-1420:
Exposure of Sensitive Information during Transient Execution

Vulnerability Mapping Notes

Usage: ALLOWED-WITH-REVIEW *(this CWE ID could be used to map to real-world vulnerabilities in limited situations requiring careful review)*

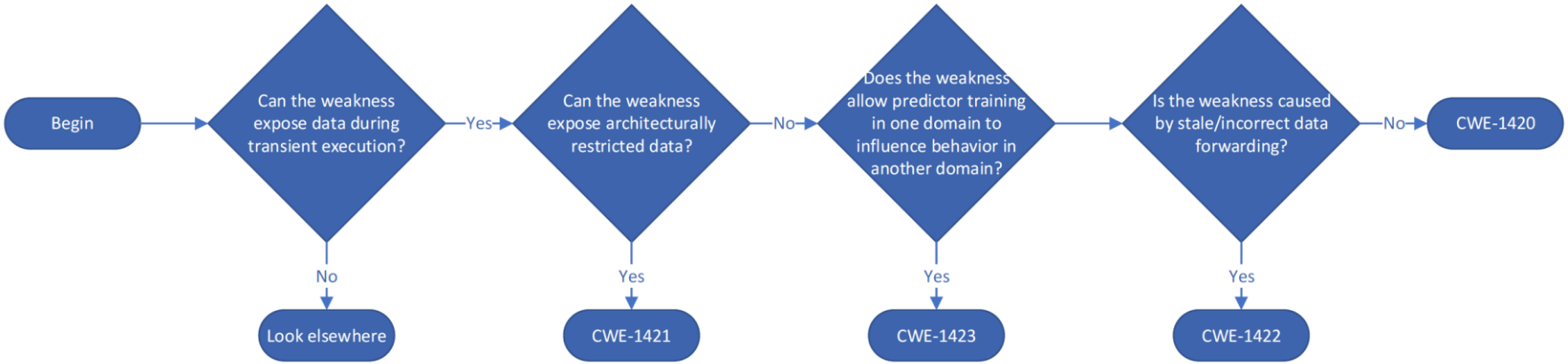
Reason: Acceptable-Use

Rationale:

This CWE entry is at the Base level of abstraction, which is a preferred level of abstraction for mapping to the root causes of vulnerabilities.

Comments:

A vulnerability should only map to CWE-1420 if it cannot map to any of CWE-1420's child weaknesses. Follow this diagram:



Covert Channel Coverage in CWE

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?
- Are covert channels weakness focused 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

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

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.**
- **This would infer a chaining relationship**

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

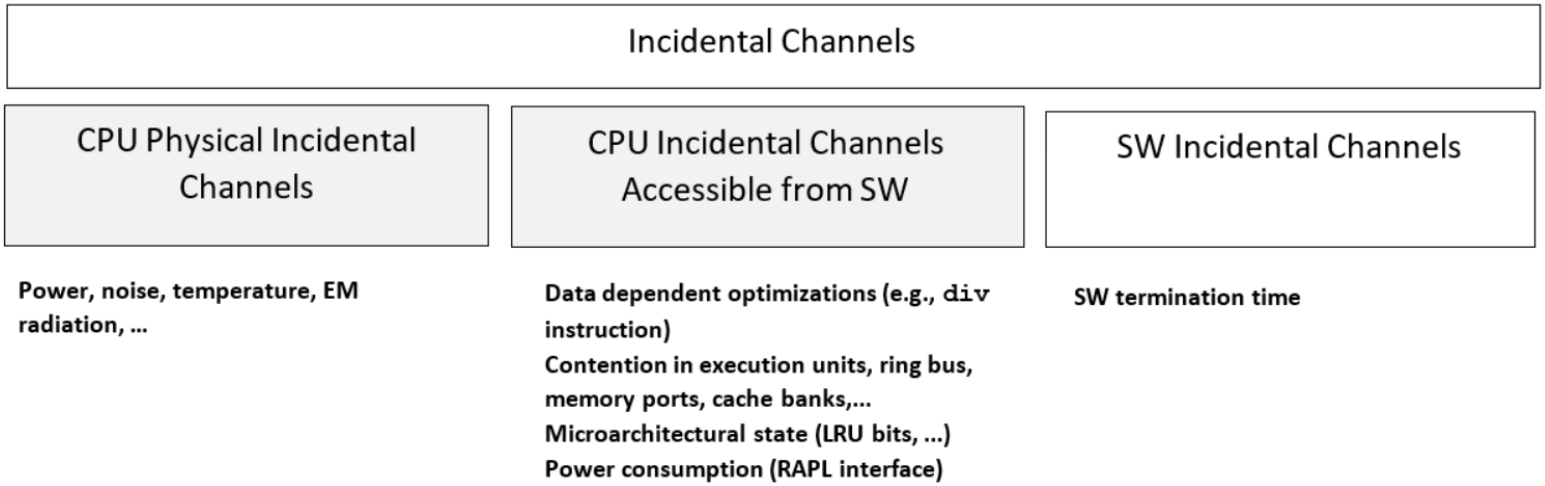


Figure 1: Example of taxonomy terminology of incidental channels on CPUs

https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/best-practices/securing-workloads-against-side-channel-methods.html

Discussion Questions

- **Does CWE-514 need to be modified to address more HW centric concerns?**
- **Should there be a completely new covert channel like weakness created for hardware?**
- **Are covert channels weakness focused or attacker focused?**
- **Coverage in HW View?**
- **Other questions about covert channels and HW CWE?**

System Verilog

System Verilog and Verilog

- Hack@DAC DEMOXs had indicated code snippets were System Verilog.
- The language enumeration in the current schema does not contain System Verilog, just Verilog.

Questions to SIG:

- What is the difference between Verilog and System Verilog?
- Is that difference significant enough that we should add System Verilog to the schema enumeration?
  - For example, we distinguish between C and C++ but not between variants of C (e.g., C89 vs C99).

Next Meeting (July 12)

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