CWE User Experience Working Group Meeting

September 27, 2023

Agenda

This meeting is being recorded :-)

- **Purpose**
- **Housekeeping**
- **Primary topics**
  - CWE Element / Information Presentation Mockup
  - CWE Demonstrative Examples
  - CWE Graphical Depiction PDFs
  - Red Hat CWE Blog
  - Open Discussion
- **Reminders and Adjourn**

UEWG: Purpose

- **Mission:** Identifying areas where CWE content, rules, guidelines, and best practices must improve to better support stakeholder community, and work collaboratively to fix them
- **Periodic reporting of activities to CWE Board**
  - (next quarterly Board meeting TBD Q4-2023)
- **Please solicit participations from your contacts**
  - Contact: cwe@mitre.org & capec@mitre.org

Housekeeping

- **CWE UEWG Meeting frequency**
  - Meetings have been changed and are now set to occur on the last Wednesday of the month

- **The CWE Program is continuously seeking feedback on UEWG activities and priorities during these sessions or via email: cwe@mitre.org**

Topic 1

CWE Element / Information Presentation Mockup

Chris Coffin

CWE Element/Information Presentation

- When a user visits a CWE, what elements or information should be they be presented with first and in what format?
- The custom filters are a step in this direction and help users focus on the elements that are important to them
- Should a default view of CWE elements be created that focuses on just the right amount of information and ordering of elements?
- Could diagrams or graphics help understanding of weakness information and concepts?

CWE Element/Information Presentation – Current Operational View

CWE-787: Out-of-bounds Write

Weakness ID: 787
Abstraction: Base
Structure: Simple

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

Description

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

Extended Description

Typically, this can result in corruption of data, a crash, or code execution. The product may modify an index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent write operation then produces undefined or unexpected results.

Alternate Terms

Memory Corruption: Often used to describe the consequences of writing to memory outside the bounds of a buffer, or to memory that is invalid, when the root cause is something other than a sequential copy of excessive data from a fixed starting location. This may include issues such as incorrect pointer arithmetic, accessing invalid pointers due to incomplete initialization or memory release, etc.

Relationships

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

Nature	Type	ID	Name
ChildOf	C	119	Improper Restriction of Operations within the Bounds of a Memory Buffer
ParentOf	V	121	Stack-based Buffer Overflow
ParentOf	V	122	Heap-based Buffer Overflow
ParentOf	B	123	Write-what-where Condition
ParentOf	B	124	Buffer Underwrite ('Buffer Underflow')
CanFollow	B	822	Untrusted Pointer Dereference
CanFollow	B	823	Use of Out-of-range Pointer Offset

CWE Element/Information Presentation – A Graphical Representation of CWE

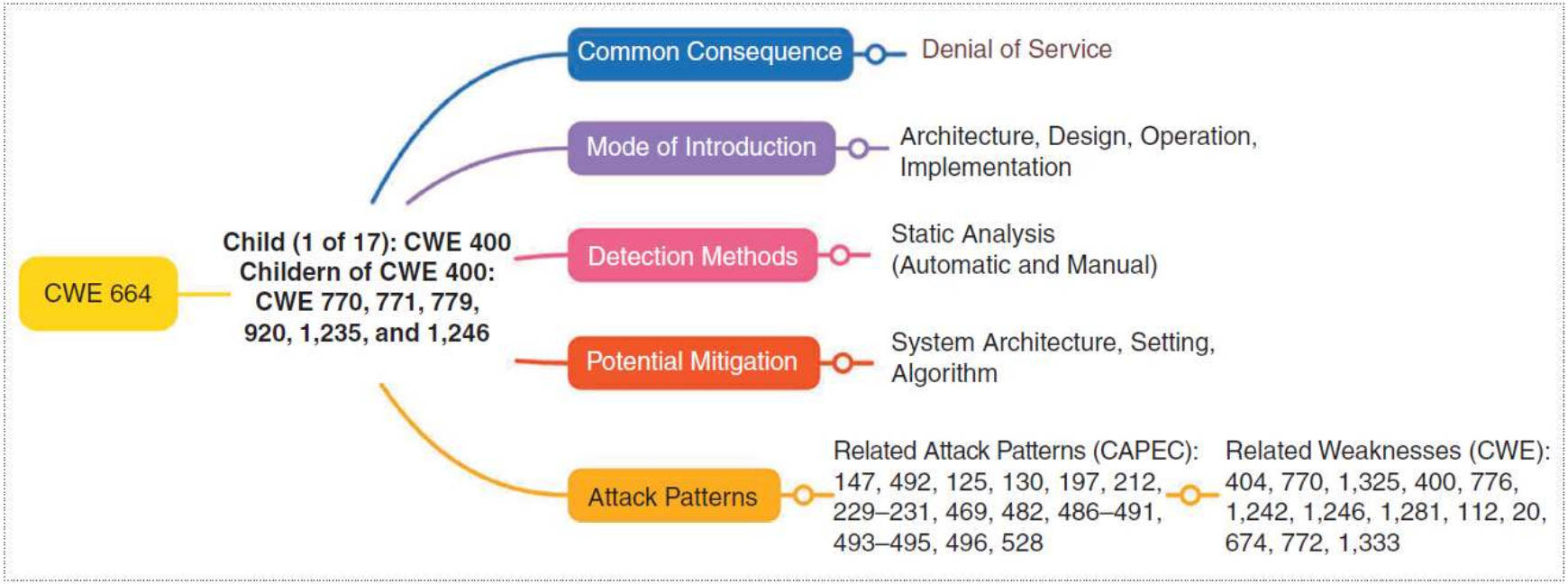


Figure 2. A partial analysis of CWE 664.

- **Example from "Toward Common Weakness Enumerations in Industrial Control Systems," Co-published by the IEEE Computer and Reliability Societies, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10194510**

CWE Element/Information Presentation – Questions and Possible Next Steps

- **Who is the audience? Are we aiming at new, experienced, or all users?**
- **What CWE elements and information are most important to these users?**
  - Should these be the first things presented on any weakness page?
- **Should a graphical representation of the CWE information be investigated and/or created?**

- **Possible activity**
  - Volunteer to create a new weakness entry page structure for an existing weakness that improves user understandability and presents the most important information
  - CWE team will propose ideas for next meeting; other volunteers?

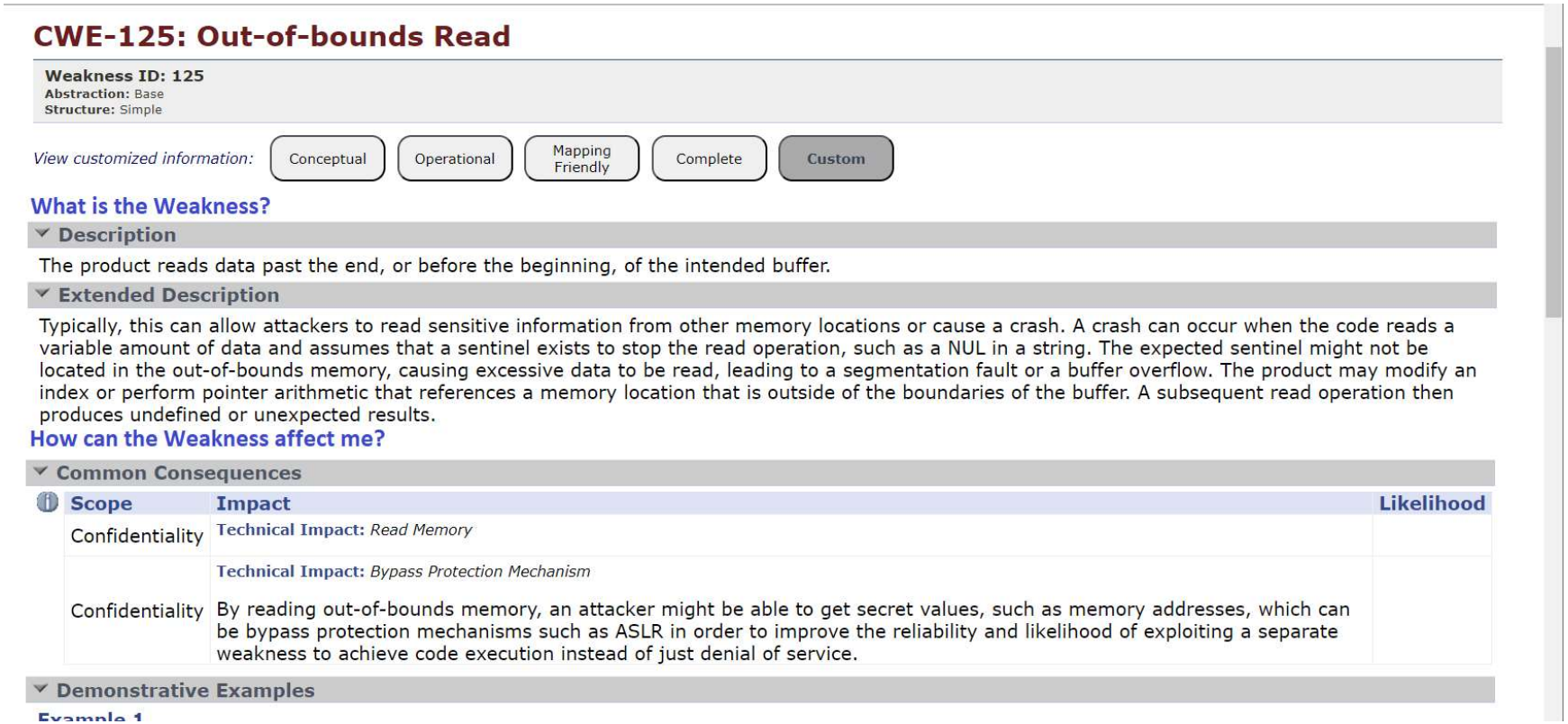
CWE Element/Information Presentation – September Update

- **New users might only want to see limited information to begin with, and possibly the relationships with the rest of the corpus**
  - New users are trying to wrap their head around the corpus and could be confused by some of the CWE data
  - Try to limit the number of fields to avoid overload
  - Could limit a view of the corpus to Description, Extended Description, Demonstrative Examples, and Observed Examples
    - Would Extended Description really be needed here?
  - Could also include Alternate Terms, Common Consequences, and References
  - Last thought would be Relationships as it helps the user understand how the corpus fits together
    - Could use the existing Relationships structure, but an interactive graphical view might engage users more and help them explore the corpus
  - Might be useful to present/bucket the CWE fields at a higher level
    - e.g., What is it? How can it affect me? Third-party information?

CWE Element/Information Presentation – September Update (continued)

- **Point of this initial mockup?**
  - Limit the CWE fields to those that a new user or casual user would understand and relate to
    - Used the custom filter to change the fields to those mentioned previously
    - Would be good to allow users to rearrange some of the fields
  - Provide more context, possibly with the use of higher-level buckets that describe the included fields
    - Not sold on the mockup names by any means, but wanted to provide examples to start the process

CWE Element/Information Presentation – Mockup for New CWE Users



CWE-125: Out-of-bounds Read

Weakness ID: 125
Abstraction: Base
Structure: Simple

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

What is the Weakness?

Description

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

Extended Description

Typically, this can allow attackers to read sensitive information from other memory locations or cause a crash. A crash can occur when the code reads a variable amount of data and assumes that a sentinel exists to stop the read operation, such as a NUL in a string. The expected sentinel might not be located in the out-of-bounds memory, causing excessive data to be read, leading to a segmentation fault or a buffer overflow. The product may modify an index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent read operation then produces undefined or unexpected results.

How can the Weakness affect me?

Common Consequences

Scope	Impact	Likelihood
Confidentiality	**Technical Impact:** *Read Memory*	
Confidentiality	**Technical Impact:** *Bypass Protection Mechanism* By reading out-of-bounds memory, an attacker might be able to get secret values, such as memory addresses, which can be bypass protection mechanisms such as ASLR in order to improve the reliability and likelihood of exploiting a separate weakness to achieve code execution instead of just denial of service.	

Demonstrative Examples

Example 1

CWE Element/Information Presentation – Mockup for New CWE Users

index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent read operation then produces undefined or unexpected results.

How does this Weakness relate to others?

Relationships

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

Nature	Type	ID	Name
ChildOf	C	119	Improper Restriction of Operations within the Bounds of a Memory Buffer
ParentOf	V	126	Buffer Over-read
ParentOf	V	127	Buffer Under-read
CanFollow	B	822	Untrusted Pointer Dereference
CanFollow	B	823	Use of Out-of-range Pointer Offset
CanFollow	B	824	Access of Uninitialized Pointer
CanFollow	B	825	Expired Pointer Dereference

Relevant to the view "Software Development" (CWE-699)

Nature	Type	ID	Name
MemberOf	C	1218	Memory Buffer Errors

Relevant to the view "Weaknesses for Simplified Mapping of Published Vulnerabilities" (CWE-1003)
Relevant to the view "CISQ Quality Measures (2020)" (CWE-1305)
Relevant to the view "CISQ Data Protection Measures" (CWE-1340)

Where can I get more information?

References

[REF-1034] Raoul Strackx, Yves Younan, Pieter Philippaerts, Frank Piessens, Sven Lachmund and Thomas Walter. "Breaking the memory secrecy assumption". ACM. 2009-03-31. <https://dl.acm.org/doi/10.1145/1519144.1519145>. *URL validated: 2023-04-07.*

[REF-1035] Fermin J. Serna. "The info leak era on software exploitation". 2012-07-25. <https://media.blackhat.com/bh-us-

Topic 2

CWE Demonstrative Examples

Chris Coffin

Demonstrative Examples (aka “Demox”)

- **Previous UEWG: Are there demonstrative examples for all CWEs?**
  - 521 CWEs include Demonstrative Examples data
    - CWE-787: Out-of-bounds Write – includes 7 examples (6 “inherited” by children)
  - 412 CWEs do not include Demonstrative Examples data
    - CWE-863: Incorrect Authorization – 24th on the 2023 Top 25 list (2 of 6 children have examples)
    - CWE-357: Insufficient UI Warning of Dangerous Operations
- **Why do CWEs not include it?**
  - May not have good examples in the case of newer hardware CWEs
  - Priorities were on getting new (albeit incomplete) weakness entries included in the corpus
  - Require significant labor and expertise to create a good example
- **Should Demonstrative Examples be required for every CWE, and should there be an effort to add one in every case it’s currently missing?**
  - Note: ~160 demox are “shared” in multiple CWEs
  - Current content goal: each high-level CWE should “inherit” 1 basic/simple demox from each child

Demonstrative Examples – Example 1

CWE-787: Out-of-bounds Write

Weakness ID: 787
Abstraction: Base
Structure: Simple

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

Description

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

Extended Description

Typically, this can result in corruption of data, a crash, or code execution. The product may modify an index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent write operation then produces undefined or unexpected results.

Demonstrative Examples

Example 1

The following code attempts to save four different identification numbers into an array.

Example Language: **C** *(bad code)*

```
int id_sequence[3];

/* Populate the id array. */

id_sequence[0] = 123;
id_sequence[1] = 234;
id_sequence[2] = 345;
id_sequence[3] = 456;
```

Since the array is only allocated to hold three elements, the valid indices are 0 to 2; so, the assignment to id_sequence[3] is out of bounds.

Demonstrative Examples – Example 2

CWE-1189: Improper Isolation of Shared Resources on System-on-a-Chip (SoC)

Weakness ID: 1189
Abstraction: Base
Structure: Simple

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

Description

The System-On-a-Chip (SoC) does not properly isolate shared resources between trusted and untrusted agents.

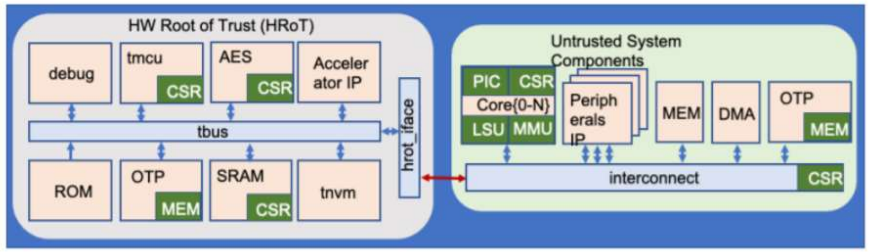
Extended Description

A System-On-a-Chip (SoC) has a lot of functionality, but it may have a limited number of pins or pads. A pin can only perform one function at a time. However, it can be configured to perform multiple different functions. This technique is called pin multiplexing. Similarly, several resources on the chip may be shared to multiplex and support different features or functions. When such resources are shared between trusted and untrusted agents, untrusted agents may be able to access the assets intended to be accessed only by the trusted agents.

Demonstrative Examples

Example 1

Consider the following SoC design. The Hardware Root of Trust (HRoT) local SRAM is memory mapped in the core{0-N} address space. The HRoT allows or disallows access to private memory ranges, thus allowing the sram to function as a mailbox for communication between untrusted and trusted HRoT partitions.



We assume that the threat is from malicious software in the untrusted domain. We assume this software has access to the core{0-N} memory map and can be running at any privilege level on the untrusted cores. The capability of this threat in this example is communication to and from the mailbox region of SRAM modulated by the hrot_iface. To address this threat, information must not enter or exit the shared region of SRAM through hrot_iface when in secure or privileged mode.

Topic 3

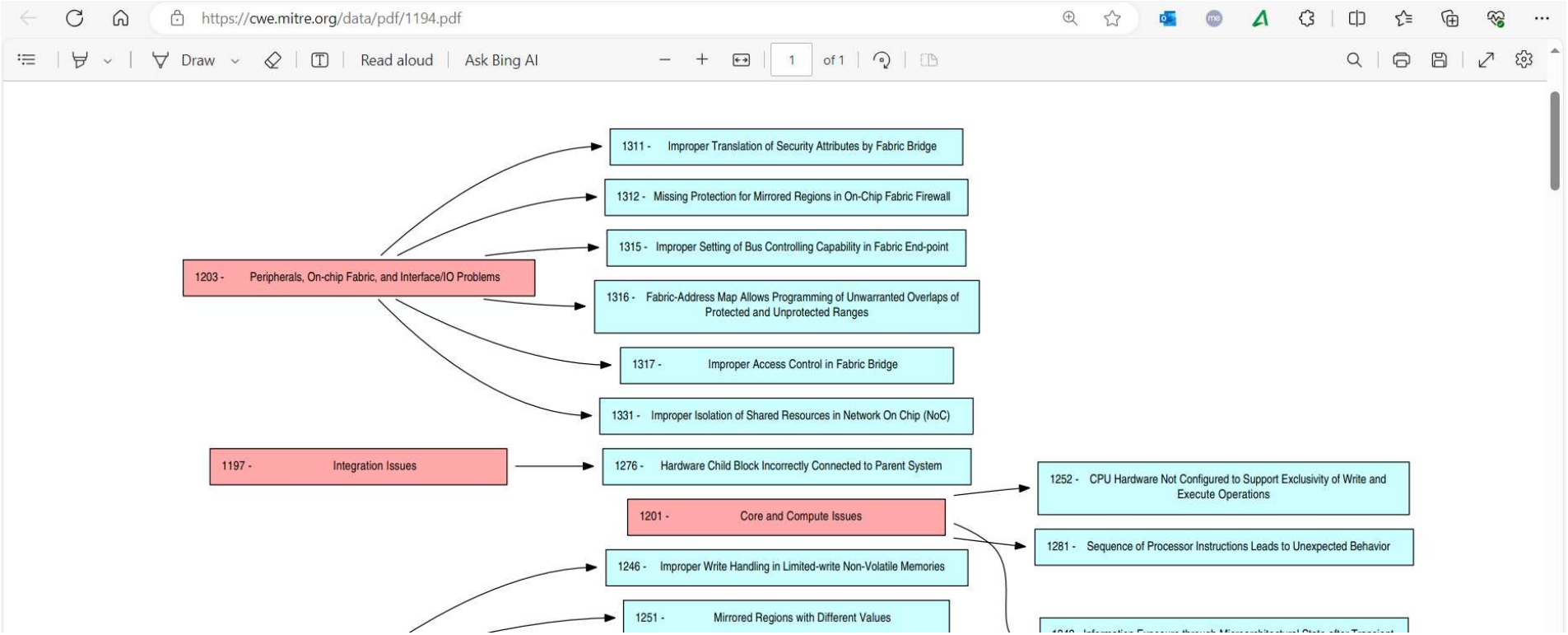
CWE Graphical Depiction PDFs

Steve Christey Coley

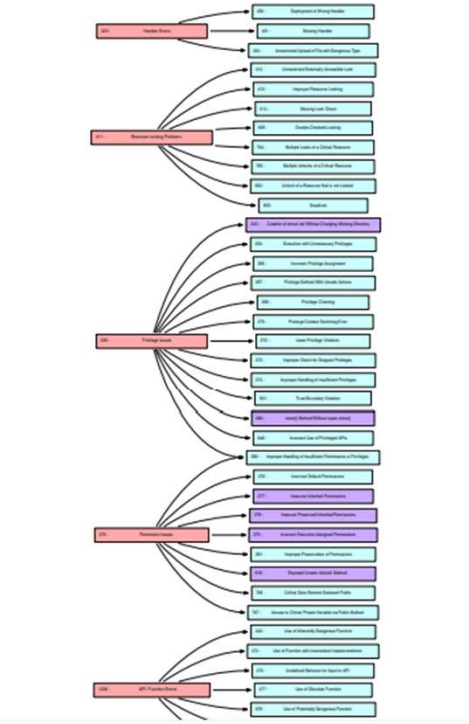
CWE Graphical Depiction PDFs

- **Provides a way of quickly seeing the structure implied by the parent relationships in CWE views**
- **Some files provide "coverage graphs," in which the members of a smaller view are highlighted within the context of a larger view**
- **Use color to graphically highlight levels of abstraction**
- **Cited in advice to people for conducting mapping**
  - https://cwe.mitre.org/documents/cwe_usage/guidance.html
- **Only hundreds of downloads per month**
  - No visibility into who the downloaders are
  - Not easy to find or notice these PDFs with the current site layout
- **https://cwe.mitre.org/data/pdfs.html**

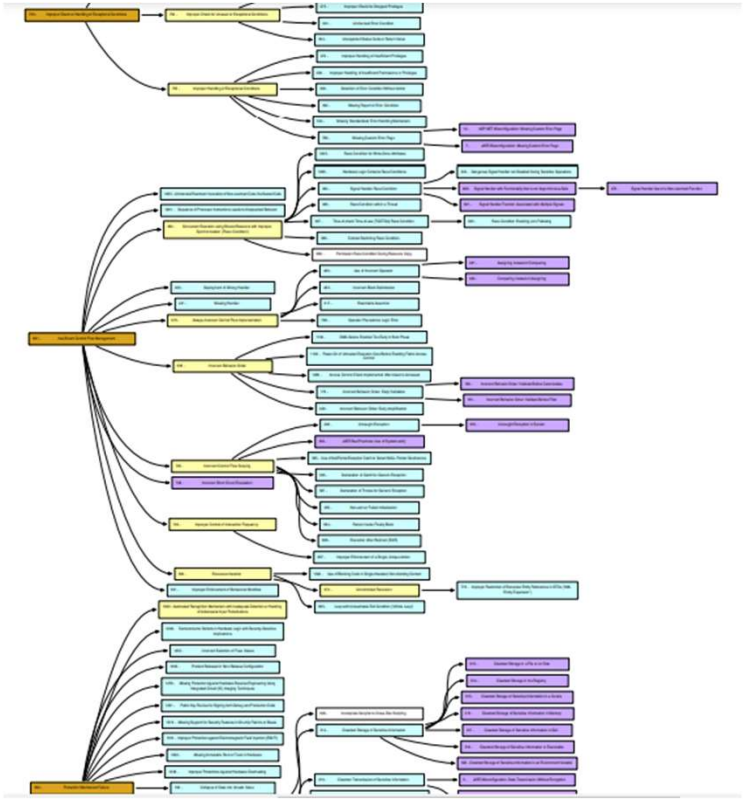
CWE Graphical Depiction PDFs – Example 1 – Hardware View (CWE-1194)



CWE Graphical Depiction PDFs – Example 2 – Development View (CWE-699)



CWE Graphical Depiction PDFs – Example 3 – Research View (CWE-1000)



- Colors indicate level of abstraction
- All 933 weaknesses (in CWE 4.12) are in the (large/long) graph
- Can see places where amounts of detail vary (typically due to lack of focused classification research across the entire community)

Questions for Graphical Depiction PDFs

- Are UEWG members aware of these depictions?
- Do any UEWG members currently use these graphical depictions?
- What kinds of users would care about hierarchical depictions of views?
- Do they provide value in their current form?
- Could they be improved upon?
- https://cwe.mitre.org/data/pdfs.html

Topic 4

Red Hat CWE Blog

Przemyslaw Roguski

Topic 5

Open Discussion

Chris Coffin

Miscellaneous Topics

- **Previous Meeting: Someone mentioned creating CWE quizzes that could generate interest in learning about CWE**
  - Are there any members interested in putting together some questions to be used for this purpose?
- **Other thoughts or topics?**

Next Meeting – October 25 @ 12pm

PLEASE CONTACT WITH ANY QUESTIONS OR THOUGHTS

CWE@MITRE.ORG

Backups

Grouping CWEs

- **CWE entries are currently "grouped" in different ways to provide useful subsets of the CWE corpus for different purposes:**
  - Views: a subset of CWE entries that provides a way of examining CWE content. The two main view structures are Slices (flat lists) and Graphs (containing relationships between entries), examples include:
    - CWE-1194: Hardware Design
    - CWE-699: Software Development
    - CWE-1400: Comprehensive Categorization for Software Assurance Trends
    - CWE-1003: Weaknesses for Simplified Mapping of Published Vulnerabilities (NVD)
  - Categories: a CWE entry that contains a set of other entries that share a common characteristic. A category is not a weakness, but rather a structural item that helps users find weaknesses that share the stated common characteristic.
    - CWE-1199: General Circuit and Logic Design Concerns
  - ~ Overall Hierarchy
    - CWE-1000: Research Concepts contains all CWE entries in one hierarchical structure

Grouping CWEs, cont.

- What groupings are most useful to new or casual CWE users? Experienced users?
- How can groupings be better presented/discovered/identified to the user?
- Should new users be guided to groupings of CWEs for learning about CWE? (e.g., links in user stories)
- Are there additional groupings that we are missing? Too many?

CSV Single Colon Separators Within Column Data

- **A double colon is used to separate csv fields/columns data, while a single colon is used to separate multi-value data within fields/columns**
- **The Observed Examples field contains Reference and link data that includes a url in some cases (colon within "http://...")**
- **Should a note be added to the download data that warns the user of this, or should we look into an alternative separator?**
- **Example taken from CWE - CWE-41: Improper Resolution of Path Equivalence (4.12) (mitre.org)**
  - ::REFERENCE:CVE-2000-1114:DESCRIPTION:Source code disclosure using trailing dot:LINK:https://www.cve.org/CVERecord?id=CVE-2000-1114::REFERENCE:CVE-2002-1986:DESCRIPTION:Source code disclosure using trailing

CWE User Pain Points

- **Pain point topics that the group is aware of or would like to discuss**
- **For those on the call, what were your biggest questions or concerns when beginning to use CWE?**
- **Are there common questions that CWE users have that are not covered in the current FAQ?**

- **Other potential opportunities:**
  - Features we could expand or improve to make CWE consumption easier?
  - Maybe engage the community in one or more ways to solicit this kind of feedback (see topic #3)

- **Other thoughts?**

Community Engagement Strategy

- **Develop a strategy for engaging the CWE user community for feedback**
- **What are the best methods to query the community on topics such as the pain points covered in topic #2**
- **What communication methods should be employed?**
  - E.g., polls, emails, web, social media
- **Should we target specific user types?**
- **Other thoughts?**

CWE Video Tips Series

- **Current video ideas:**
  - How to search CWE for a weakness
  - How to display only the information that you need with presentation filters
  - What is a weakness (vs a vulnerability)
  - How are weaknesses organized
  - What is a category (how is it different than a pillar)
  - What are views
  - How and why to use the research view
  - Use cases for CWE (could user stories be used?)
  - How do I submit an idea for a new weakness
  - How can I improve the quality of existing weaknesses

New to CWE – Future Content

- **The New to CWE content audience is different from what has been catered to previously**
- **The audience is the casual or new user to CWE or even the manager who makes security funding decisions**
- **The team has previously drafted material for the New to CWE audience that covers the CWE hierarchy**
  - Not yet released material
  - Do members agree that this topic should be covered for New to CWE?
- **Are there other topics that UEWG members feel strongly about or believe should be covered given the intended audience?**
- **Should there be a close coupling of the topics covered here with the CWE Video Tips series?**

CWE Naming and Vulnerability Mapping

- **Being thinking about solutions for common and well-known issues surrounding use of CWE names and how to more easily map vulnerabilities to CWEs**
- **Current CWE structure is difficult to understand and use**
- **Community needs better root cause information for vulnerabilities**
- **Does CWE naming need a change or update to support easier mapping?**
  - Remove CWE names for Views and/or Categories?
  - New naming that embeds a structure (e.g., CWE-1234-1)