CWE User Experience Working Group Meeting

November 29, 2023

Agenda

This meeting is being recorded :-)

- **Purpose**
- **Housekeeping**
- **Primary topics**
  - Data Mining Mitre.org Projects (Member presentation)
  - CVE –> CWE Root Cause Mapping
- **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 December Meeting**
  - The meeting currently set for 12/27 is being brought forward a week to 12/20
  - Updated invitation to follow this meeting

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

Topic 1

Data Mining Mitre.org Projects

UEWG Member – Blaine Wilson

Data Mining Mitre.org Projects

BMO We're here to help.™

V1.0
November 2023

Where did the data mining idea come from? (Mitigation example)

CAPEC 103: Clickjacking

- CAPEC Mitigations
  - When maintaining an authenticated session with a privileged target system, do not use the same browser to navigate to unfamiliar sites to perform other activities. Finish working with the target system and logout first before proceeding to other tasks.
  - Turn off JavaScript, Flash and disable CSS.
  - If using the Firefox browser, use the NoScript plug-in that will help forbid iFrames.

image source: resources.infosecinstitute.com

- Related CWE Weaknesses Mitigations
  - **CWE 1021: Improper Restriction of Rendered UI Layers or Frames**
    - Implementation Phase
      - The use of X-Frame-Options allows developers of web content to restrict the usage of their application within the form of overlays, frames, or iFrames. The developer can indicate from which domains can frame the content. The concept of X-Frame-Options is well documented, but implementation of this protection mechanism is in development to cover gaps. There is a need for allowing frames from multiple domains.
      - A developer can use a frame-breaker script in each page that should not be framed. This is very helpful for legacy browsers that do not support X-Frame-Options security feature previously mentioned. It is also important to note that this tactic has been circumvented or bypassed. Improper usage of frames can persist in the web application through nested frames. The frame-breaking script does not intuitively account for multiple nested frames that can be presented to the user.
      - This defense-in-depth technique can be used to prevent the improper usage of frames in web applications. It prioritizes the valid sources of data to be loaded into the application through the usage of declarative policies. Based on which implementation of Content Security Policy is in use, the developer should use the frame-ancestors directive or the frame-src directive to mitigate this weakness. Both directives allow for the placement of restrictions when it comes to allowing embedded content.+

What is the Data Mining idea?

Proactive vulnerability mitigation through Mitre CWE Data Mining

- To enhance the security posture of our software applications, this epic focuses on data mining techniques of the Mitre CWE data.
- The goal is to systemically discover, and address, known common software weaknesses by linking CWE common weaknesses to specific tasks and artifacts within my company's SDLC (including but not limited to the requirements, design, implementation, and testing phases) ensuring weaknesses are addressed and thereby reducing the risk of vulnerabilities in our software releases.
- This epic focuses on the tasks and user stories to achieve this proactive mitigation approach.

A lot of long words in there, miss

What is it that you want?

- Requirements for projects
- Test cases for testing teams
  - Application functionality testing
  - Security testing
- Mitigations for developers

image source: Pirates of the Caribbean: The Curse of the Black Pearl

User Story #1

AS A CWE data consumer

I WANT a repeatable way convert CWE data to a format allowing me to run queries

SO THAT I can easily visualize and consume the CWE data

WHEN I use the current data published by Mitre

THEN the data will be converted into my consumable format

WHEN Mitre publishes a new version of the documentation in the future

THEN the data will be converted into the same consumable format

WHEN the structure of the CWE data changes

THEN I can see the differences between the old and new data structures

User Story #1 - Options

1. Use XML schema to automatically create the database

Summary

- Programmatically walk the schema to create the database
- Walk the XML to import the data

Advantages

- One script works on many different types of XML documents

2. Manually review the XML to create the database scripts

Summary

- Manually read the XML to create the database scripts
- Manually create the script to import the data

Advantages

- Flatter database
- Failed import on XML schema changes (SQL scripts won't just start failing)

User Story #1 - Actual

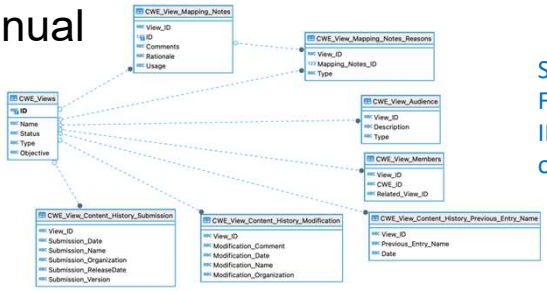
1. Started with Option 1 (automatically build the database)
2. Discovered multiple namespaces/schemas in the XML and HTML is being parsed as XML
3. Switched to Option 2 (manually build everything)
4. Encode HTML
5. Create python script to display XML structure
6. Create DDL SQL Script
7. Create python functions per XML element to validate attributes & sub-elements and load data
8. Have the CWE database
9. Started on CAPEC
10. Remembered how long it took for CWE and switched back to Option 1
11. Create generic python script to convert XML to JSON while encoding HTML
12. Create generic python script to import JSON
13. Have the CWE and CAPEC database

Same simple query on the two database schemas

```
<View ID="699" Name="Software Development" Type="Graph"
   <Objective>This view organizes weaknesses around con
   including both architecture and implementation. Acco
   variety of categories that are intended to simplify
   <Audience>
      <Stakeholder>
         <Type>Software Developers</Type>
         <Description>Software developers (including ar
         their software application. The use of concept
         specific phase of the development lifecycle.</
      </Stakeholder>
      <Stakeholder>
         <Type>Educators</Type>
         <Description>Educators use this view to teach
      </Stakeholder>
   </Audience>
```

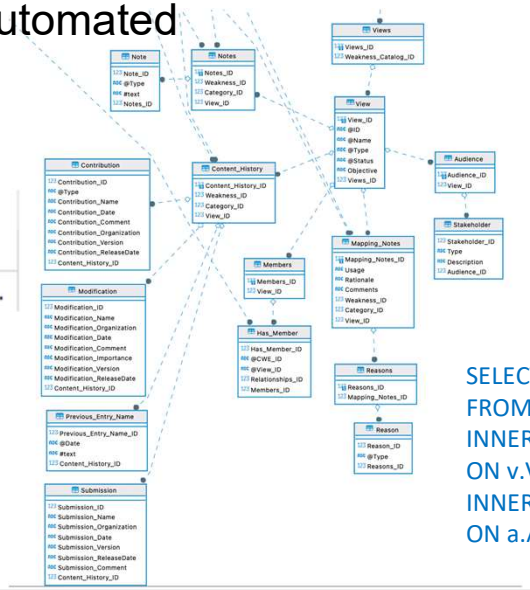
	Name	Type	Description
1	Software Development	Software Developers	Software developers (including architects, ...
2	Software Development	Educators	Educators use this view to teach future develope...

Manual



```
SELECT v.Name, va.Type, va.Description
FROM Views AS v
INNER JOIN View_Audience AS va
on v.ID=va.View_ID
```

Automated

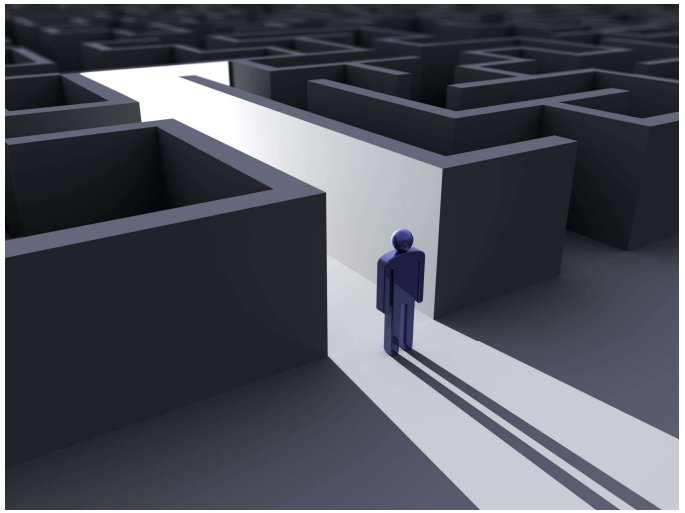


```
SELECT v."@Name“ AS Name, s.Type, s.Description
FROM View AS v
INNER JOIN Audience AS a
ON v.View_ID=a.View_ID
INNER JOIN Stakeholder AS s
ON a.Audience_ID=s.Audience_ID
```

Next Steps

Collaboration

- Does anyone want to participate?
- What does collaboration look like? (what are we producing)
- Where do we store stuff?
- Part of CWE/CAPEC User Experience Working Group?



Database

- Need to pick import / database option
  - Manual: 1,549 lines, no re-use, not many comments
  - Automatic: 394 lines, re-use, comments, needs better schema validation
- Need help validating the database

User Stories

- Need help coming up with user stories on why to extract data

Development

- Could use help in maintaining data extraction scripts
- Could use help in writing the SQL scripts to extract data based on user stories

Demo

Next set of stories to flush out data gaps

User Story #2

AS A CWE data consumer

I WANT TO flag all common weaknesses with no mode of introduction data

SO THAT I can take each weakness back to the CWE community to ideate and document modes of introduction

WHEN a common weakness has an undocumented mode of introduction

THEN the mode of introduction will be documented

WHEN a common weakness does not have a clear mode of introduction

THEN the weakness has been verified to not have a clear mode of introduction

```
SELECT "@ID" AS ID, "@Name" AS Name FROM Weakness AS w WHERE NOT EXISTS
(SELECT 1 FROM Modes_Of_Introduction WHERE Weakness_ID = w.Weakness_ID)
```

Next set of stories to flush out data gaps - continued

User Story #3

AS A CWE data consumer

I WANT TO flag all common weaknesses with no mitigation data

SO THAT I can take each weakness back to the CWE community to ideate and document mitigations

WHEN a common weakness has an undocumented mitigation

THEN the mitigation will be documented

WHEN a common weakness does not have a clear mitigation

THEN the weakness has been verified to not have a clear mitigation

SELECT "@ID" AS ID, "@Name" AS Name FROM Weakness AS w WHERE NOT EXISTS (SELECT 1 FROM Potential_Mitigations WHERE Weakness_ID = w.Weakness_ID)

Circle back to Mitigations Example

```
SELECT "@ID" AS ID, "@Name" AS Name, ms.Mitigation, "CAPEC" AS Source FROM CAPEC_Attack_Pattern AS ap
INNER JOIN CAPEC_Mitigations AS ms ON ms.Attack_Pattern_ID=ap.Attack_Pattern_ID
WHERE ap."@ID" = "103"
UNION ALL
SELECT ap."@ID" AS ID, ap."@Name" AS Name, m.Description as Mitigation, "CWE" AS Source FROM CAPEC_Attack_Pattern AS ap
INNER JOIN CAPEC_Related_Weaknesses AS rws ON rws.Attack_Pattern_ID=ap.Attack_Pattern_ID
INNER JOIN CAPEC_Related_Weakness AS rw ON rw.Related_Weaknesses_ID=rws.Related_Weaknesses_ID
INNER JOIN CWE_Weakness AS w ON w."@ID" = rw."@CWE_ID"
INNER JOIN CWE_Potential_Mitigations AS pms on pms.Weakness_ID = w.Weakness_ID
INNER JOIN CWE_Mitigation AS m on m.Potential_Mitigations_ID = pms.Potential_Mitigations_ID
WHERE ap."@ID" = "103";
```

ID	Name	Mitigation	Source
103	Clickjacking	If using the Firefox browser, use the NoScript plug-in that will help forbid iFrames., Turn off JavaScript, Flash and disable CSS., When maintaining an authenticated session with a privileged target system, do not use the same browser to navigate to unfamiliar sites to perform other activities. Finish working with the target system and logout first before proceeding to other tasks.	CAPEC
103	Clickjacking	<xhtml:p>The use of X-Frame-Options allows developers of web content to restrict the usage of their application within the form of overlays, frames, or iFrames. The developer can indicate from which domains can frame the content.</xhtml:p> <xhtml:p>The concept of X-Frame-Options is well documented, but implementation of this protection mechanism is in development to cover gaps. There is a need for allowing frames from multiple domains.</xhtml:p>	CWE
103	Clickjacking	<xhtml:p>A developer can use a "frame-breaker" script in each page that should not be framed. This is very helpful for legacy browsers that do not support X-Frame-Options security feature previously mentioned.</xhtml:p> <xhtml:p>It is also important to note that this tactic has been circumvented or bypassed. Improper usage of frames can persist in the web application through nested frames. The "frame-breaking" script does not intuitively account for multiple nested frames that can be presented to the user.</xhtml:p>	CWE
103	Clickjacking	This defense-in-depth technique can be used to prevent the improper usage of frames in web applications. It prioritizes the valid sources of data to be loaded into the application through the usage of declarative policies. Based on which implementation of Content Security Policy is in use, the developer should use the "frame-ancestors" directive or the "frame-src" directive to mitigate this weakness. Both directives allow for the placement of restrictions when it comes to allowing embedded content.	CWE

Appendix

AS0

Slide 19

AS0 Do we need this slide?
Alec J Summers, 2023-11-27T18:49:09.615

Topic 2

Root Cause Mapping

Alec Summers

Root Cause Mapping (RCM)

- **Root cause mapping is the identification of the underlying cause of a vulnerability**
  - This is best done by correlating CVE Records with CWE entries
  - Today, this is not done accurately at scale by the ecosystem

- **Root Cause Mapping is valuable because it:**
  - Enables trend analysis (e.g., how big of a problem is memory safety compared to other problems like injection)
  - Illuminates where investments, policy, and practices can address the weaknesses responsible for product (e.g., the vulnerable thing) vulnerabilities so that they can be eliminated

RCM Assertions

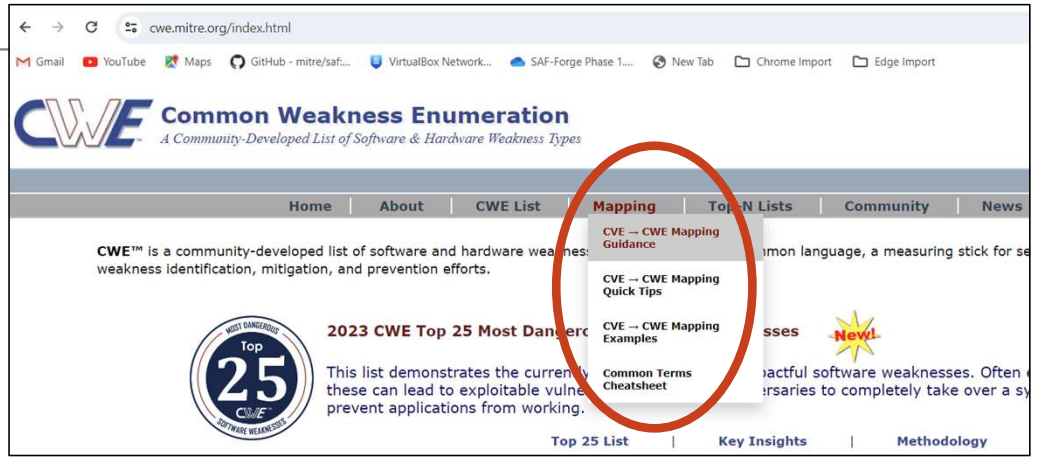
- **Centralized root cause mapping is expensive and difficult to scale**
  - E.g., Top 25 list development
- **Effective root cause mapping is best done by the organizations that own the product**
- **Start with CVE Numbering Authorities (CNAs) because they are trusted partners**
  - If we cannot get the CNAs to do root cause mapping effectively, we cannot get anyone to do so at scale
- **Minimal understanding of vulnerabilities and weaknesses in large section of CNA community and Vuln Mgt Ecosystem**
  - Majority of CNAs don't know how to do RCM effectively
  - Many choose broad CWEs because it's easy
- **NVD will never be able to be more accurate in mapping relying solely on the CVE description.**
  - Most useful thing would be to link to the code where the vuln was discovered and then the patch code (where possible)

Root Cause Mapping Guidance (1 of 2)

- **Purpose:**
  - Provide guidance on navigating the CWE site and aligning/mapping newly discovered vulnerabilities (i.e., CVEs) to their respective, underlying weaknesses
  - By aligning CVEs to the most applicable CWE entries, the community will be in a better position to mitigate or eliminate their associated operational risk most effectively
- **Audience:**
  - Vendors and researchers who produce or analyze CVE Records
- **Informed by:**
  - Two years of experience in analyzing and mapping thousands of CVE Records in the NIST National Vulnerability Database (NVD) to CWEs for calculating the annual CWE Top 25 list

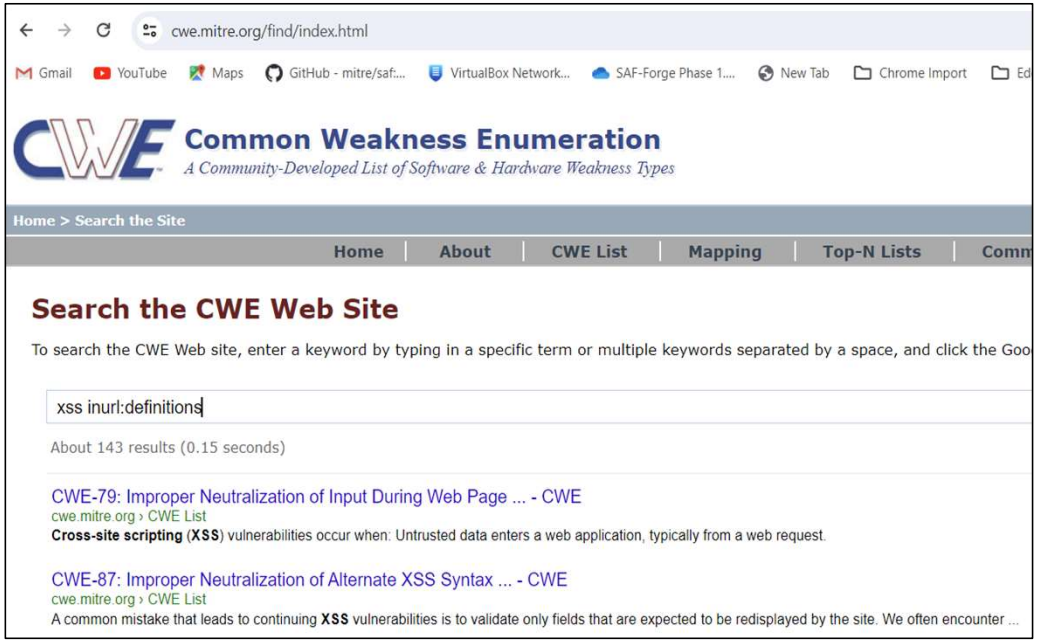
Current Root Cause Mapping Guidance (2 of 2)

- **Guidance page provides:**
  - Purpose
  - CWE Overview
  - Important Terms for Mapping Guidance
  - CWE Relationships Overview
  - Mapping Methodologies
    - Keyword Search Method
    - View-1003 Method
    - Other Useful Hierarchical Views (Views 1000, 699, and 1194)
    - Relationship Graph Visualization in PDF Format
    - Keyword Scraper



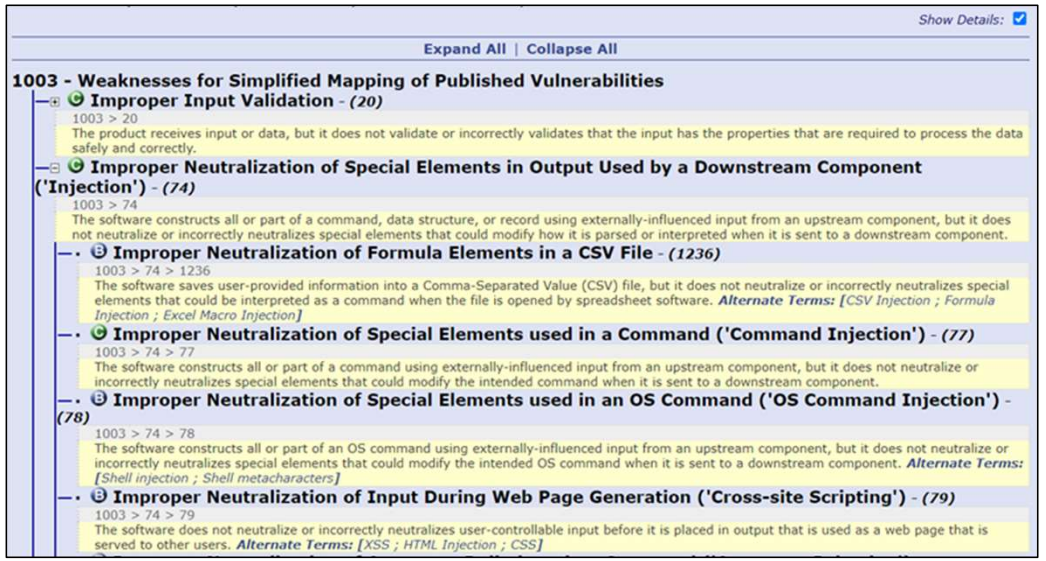
Keyword Search Method

- The in-site CWE search feature will find CWEs of interest by keyword
- You can limit the search to CWE definitions only by including "inurl:definitions" in your query
- Keywords will be generally be identified in the first few results, if not the first one
- Use the Relationships section of the resulting CWEs to find related CWEs



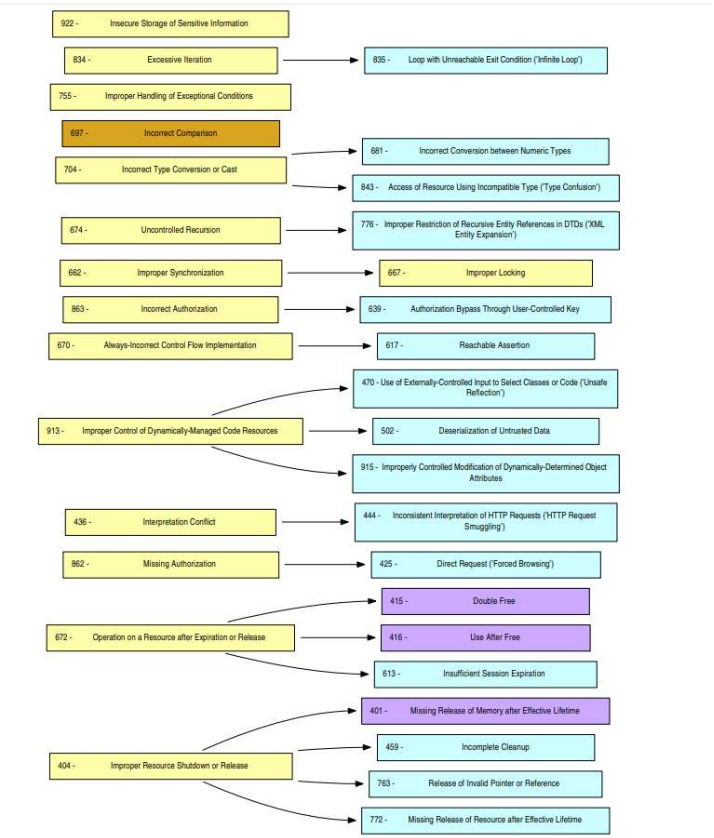
View-1003 Method (and other Views)

- Use CWE View 1003 - Weaknesses for Simplified Mapping of Published Vulnerabilities
- Provides a subset of CWEs that cover the most commonly-used CWEs that are mapped by CVEs (used by NIST NVD)
- From the View main page, the CWE list can be expanded to show all CWEs, and the show details button can be selected to include the CWE descriptions
- Search the page using the complete list with descriptions
- Can also use View-1000, View-699, and View-1194 in the same way



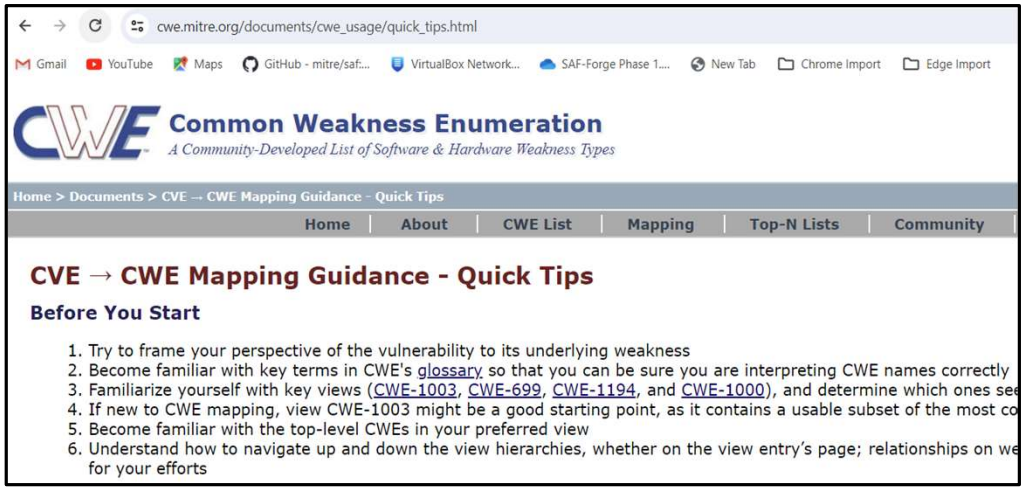
Relationship Graph Visualization in PDF Format

- Graphical displays of CWE Views are also available in PDF format
- https://cwe.mitre.org/data/pdfs.html
- Can be used to quickly browse the CWEs and their relationships



Other Root Cause Mapping Pages

- **Quick Tips** - A quick overview and key points with regard to CVE -> CWE mapping
- **Mapping Guidance – Examples** – Written examples and walks through them using the previously mentioned guidance
- **Common Terms Cheatsheet** – The most important CWE terms relevant to the Mapping Guidance discussion



Root Cause Mapping Working Group (RCM WG)

- **The Root Cause Mapping Working Group (RCM WG) was established by CVE® and CWE™ community stakeholders (e.g., Intel, Red Hat, Microsoft, NVIDIA, Rapid 7) to achieve the following goals:**
  - Determine the feasibility of effective, decentralized vulnerability root cause mapping
  - Identify the capabilities, processes, and information needed to make root cause mapping easier
- **Current objectives:**
  - Define the business use case for doing and disclosing effective root cause mapping to socialize and confirm with the broader community
  - Discuss what organizations are doing currently to fill gaps in the existing CWE structure
  - Develop a communications plan, examples of what good mapping looks like, and possible metrics
- **Looking ahead:**
  - CVE/FIRST VulnCon Panel Discussion Proposal
  - Opening the working group to the wider CNA community – Early 2024

Next Meeting – December 20 @ 12pm

PLEASE CONTACT WITH ANY QUESTIONS OR THOUGHTS

CWE@MITRE.ORG

Backups

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-

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)