CWE User Experience Working Group Meeting

November 19, 2025

Agenda

This meeting is being recorded :-)

- **Primary topics**
  - CWE Completeness Considerations (Steve Christey Coley)
  - Open Discussion
- **Reminders and Adjourn**

Topic 1

CWE Completeness Considerations

Steve Christey Coley

What is "Completeness?"

- **CWE corpus having entries for all weakness-related concepts**
  - Minimize gaps (at least to the Base level?)
  - CWEs for weaknesses within scope of the CWE program (e.g., AI)
- **Individual CWE entries**
  - Entry has all "high-priority" top-level fields (elements) filled
  - Entry has all "preferred" elements filled, where possible
  - Entry has other elements filled as-needed
- **Breadth – all entries have the element**
  - (or all entries in a particular set of CWEs e.g., MIHW, Top 25)
- **Depth**
  - All sub-elements are filled (where feasible)
  - Multiple sub-elements
- **Correctness / Quality**

Slide 4

BH1 Should this be "at least..."
Bob Heinemann, 2025-11-18T22:48:06.393

SC1 0 Fixed, thanks
Steven M Christey, 2025-11-18T22:53:40.633

Aspects of Completeness

- **Breadth – % of entries with the top-level element**
  - For some elements, you want to MINIMIZE their presence (maintenance notes, extended description)
- **Depth - % of sub-elements filled (where feasible)**
  - All sub-elements are filled (where feasible)
  - Elements that can have multiple instances (e.g., mitigations) "should" have all possible members (where feasible)
- **Correctness / Quality**
  - Data is correct (for some definition of "correct") without any key omissions
  - Element meets "quality" expectations
- **Can consider these relative to abstraction (Pillar, etc.) and/or use case**

Top-Level Elements (Excluding Obscure Elements)

- **Multiple use cases**
  - **Name, Description, Diagram**, **Related Weaknesses, Common Consequences, Applicable_Platforms, References**, *Extended Description*
- **Mapping-focused**
  - Mapping Notes, Alternate Terms, Taxonomy Mappings, Terminology Notes
- **Development-focused**
  - **Modes_Of_Introduction**, Exploitation_Factors, **Demonstrative_Examples**, **Observed_Examples, Potential_Mitigations**, Functional Areas, **Detection_Methods**, *Likelihood_Of_Exploit*
- **Miscellaneous**
  - Affected Resources, Weakness Ordinalities, *Related Attack Patterns*
- **Notes**
  - Other, Background Details, *Maintenance*
- **Content History**

"Depth" Examples

- **NOTE: numbers not official, based on internal snapshot November 4, 2025**
- **Applicable Platforms: 738 weaknesses**
  - 197 have a specific affected Language
    - 573 have a language "class" (not-language-specific, compiled, interpreted, assembly)
  - 231 with an affected Technology
- **Demonstrative Examples: 580 weaknesses**
    - 327 CWEs with only 1 demox
    - 1323 "Example_Code" segments across all demox; 14 without Language specified
    - 238 CWEs with Java demox; 194 C; 30 Python; 4 Go; …
  - Community involvement could help here (e.g., Python secure coding project)
- **References – authors identified, URL is not 404, etc.**
- **Most Potential Mitigations don't have "Effectiveness"**

Element Priority Varies by Task / User Type

- **Mapping? Don't care about code examples**
- **Many elements are "only" useful to developers and/or product security managers**
- **The UEWG spent time discussing weakness element importance as part of creating the current content filters (2023)**
- **Old worksheet (early 2010's) ranked elements based on stakeholder "tiers" and importance to those stakeholders:**
  - https://cwe.mitre.org/data/reports/stakeholder_field_priorities.html

Current Goals – Completeness Required (With Importance to Each Task / User Type)

Element	Completeness Goals	Mapper	Developer	Bug Hunt	Academi	PSIRT
Name	Yes	High	High	High	High	High
Description	Yes	High	High	High	High	High
Diagram	Yes	-	Medium	Low	Low?	Medium
Related Weaknesses	Yes	High	High	High	High	High
Applicable Platforms	Yes	-	High	Low	Low	High
Modes of Introduction	Yes	-	High	Low	Medium	High
Common Conseq	Yes	-	High	Medium	Low	High
Detection Methods	Yes	-	High	Medium	Low	High
Potential Mitigations	Yes	-	High	Low	-	High
Demonstrative Examples	Yes	-	High	High	?	Medium
Observed Examples	Yes - when available	-	High	High	Medium	Medium
References	Yes	-	Medium	High	High	Medium
Mapping Notes	Yes	High	-	High	Low	Medium
Content History	Yes	-	-	-	-	-

Current Goals – Completeness Not Required (With Importance to Each Task / User Type)

Element	Completeness Goals	Mapper	Developer	Bug Hunt	Academi	PSIRT
Extended Description	No - avoid if possible	Low	Medium	Low	Medium	Medium
Weakness Ordinalities	No	Medium	Medium	Low	Low	Medium
Background Details	No	-	Low	-	-	Low
Alternate Terms	No - when available	High	Medium	High	Low	High
Functional Areas	No	Low	Medium	Low	Low	Medium
Affected Resources	No	Low	Medium	Low	Low	Low
Taxonomy Maps	No	Medium	-	-	Medium	Medium
Notes - Terminology	No - when available	High	Medium	Medium	Low?	Medium
Notes - Research Gaps	No	-	-	-	Medium	-
Notes - Applicable Platform	No - when available	-	Medium	Low	-	Medium
Notes - Maintenance	No - avoid if possible	-	-	-	-	-
Notes - Theoretical	No	Low	-	-	Low	-
Notes - Other	No - avoid if possible	-	-	-	-	-
Exploitation Factors	No - Discontinued	-	-	-	-	-
Likelihood Of Exploit	No - Discontinued	-	High	Medium	-	High
Attack Patterns	No - Discontinued	-	Medium	Low	-	Medium

Completeness Status – Snapshot as of November 4 (not all elements are listed)

	All Active Weaknesses			Most-Requested (not Top 25)			Top 25 (2024)		
Field	**# Filled**	**# Unfilled**	**% Filled**	**# Filled**	**# Unfilled**	**% Filled**	**# Filled**	**# Unfilled**	**% Filled**
Name	944	0	100%	102	0	100%	25	0	100%
Description	944	0	100%	102	0	100%	25	0	100%
Extended_Description	664	280	70%	49	53	48%	7	18	28%
Diagram	64	880	7%	40	62	39%	24	1	96%
Relationships	944	0	100%	102	0	100%	25	0	100%
Common_Consequences	923	21	98%	102	0	100%	25	0	100%
Applicable_Platforms	738	206	78%	90	12	88%	25	0	100%
Modes_of_Introduction	831	113	88%	100	2	98%	25	0	100%
Detection_Methods	330	614	35%	76	26	74%	25	0	100%
Potential_Mitigations	675	269	72%	89	13	87%	24	1	96%
Demonstrative_Examples	580	364	61%	85	17	83%	24	1	96%
Observed_Examples	554	390	59%	93	9	91%	25	0	100%
Weakness_Ordinalities	282	662	30%	31	71	30%	11	14	44%
Likelihood_of_Exploit	185	759	20%	45	57	44%	23	2	92%
References	855	89	91%	85	17	83%	25	0	100%
Mapping_Notes	944	0	100%	102	0	100%	25	0	100%
Terminology_Notes	34	910	4%	6	96	5%	7	18	28%
Alternate_Terms	103	841	11%	31	71	30%	21	4	84%
Related_Attack_Patterns	336	608	36%	76	26	74%	21	4	84%
Relationship_Notes	183	761	19%	25	77	24%	9	16	36%
Taxonomy_Mappings	657	287	45%	85	17	83%	24	1	96%
Maintenance_Notes	191	753	13%	23	79	22%	7	18	28%
Affected_Resources	115	829	8%	15	87	14%	7	18	28%

Prioritizing Completeness Improvements

- **Should we choose a specific element, or a set of elements?**
- **Should we prioritize a specific task or use case over others?**
- **Which CWE(s) do we focus on?**
  - All, top 25, most requested, …
- **Should expectations vary for abstraction (Pillar, Class, Base, Variant)?**
- **Which elements would benefit most from public contributions?**
- **Should we focus more on depth than breadth?**
  - Affected languages (and which ones?); more variety in demonstrative examples; effectiveness estimates for mitigations/detection; etc.
- **Where should quality goals fit in?**

Older/Draft Slides

Prioritization Worksheet (incomplete – not all elements listed, not all stakeholders, etc.)

Element	Mapper	Developer	Bug Hunter	Academic	PSIRT
Diagram	-	Medium	Low	Low?	Medium
Common Conseq	-	**High**	Medium	Low	**High**
Platforms	-	**High**	Low	Low	**High**
References	-	Medium	**High**	**High**	Medium
Mapping Notes	**High**	-	**High**	Low	Medium
Alt Terms	**High**	Medium	**High**	Low?	**High**
Taxonomy Maps	Medium	-	-	Medium	Medium
Terminology Notes	**High**	Medium	Medium	Low?	Medium
Modes of Introduction	-	**High**	Low	Medium	
Demox	-	**High**	Medium	?	Medium
Obex	-	**High**	**High**	Medium	Medium

Topic 2

Open Discussion

Chris Coffin

Next Meeting – TBD @ 12pm

PLEASE CONTACT WITH ANY QUESTIONS OR THOUGHTS

CWE@MITRE.ORG

Backups

Reminder

- **VulnCon 2025 is April 7-10, 2025**
- **https://www.first.org/conference/vulncon2025/**
- **Call for papers was extended to January 31st**

Home Page Redesign

- **The Problem: Lack of direction “where do I go now?” - especially for new users**

- **Proposed solution: Focus on grouping sections by common user actions**
  - Learn About CWE
  - Access Content
  - Contribute
    - Contribute content
    - Participate in working groups

Reorganize New User Content

- **The Problem: New user content is in multiple places (under About and under Root Cause Mapping)**

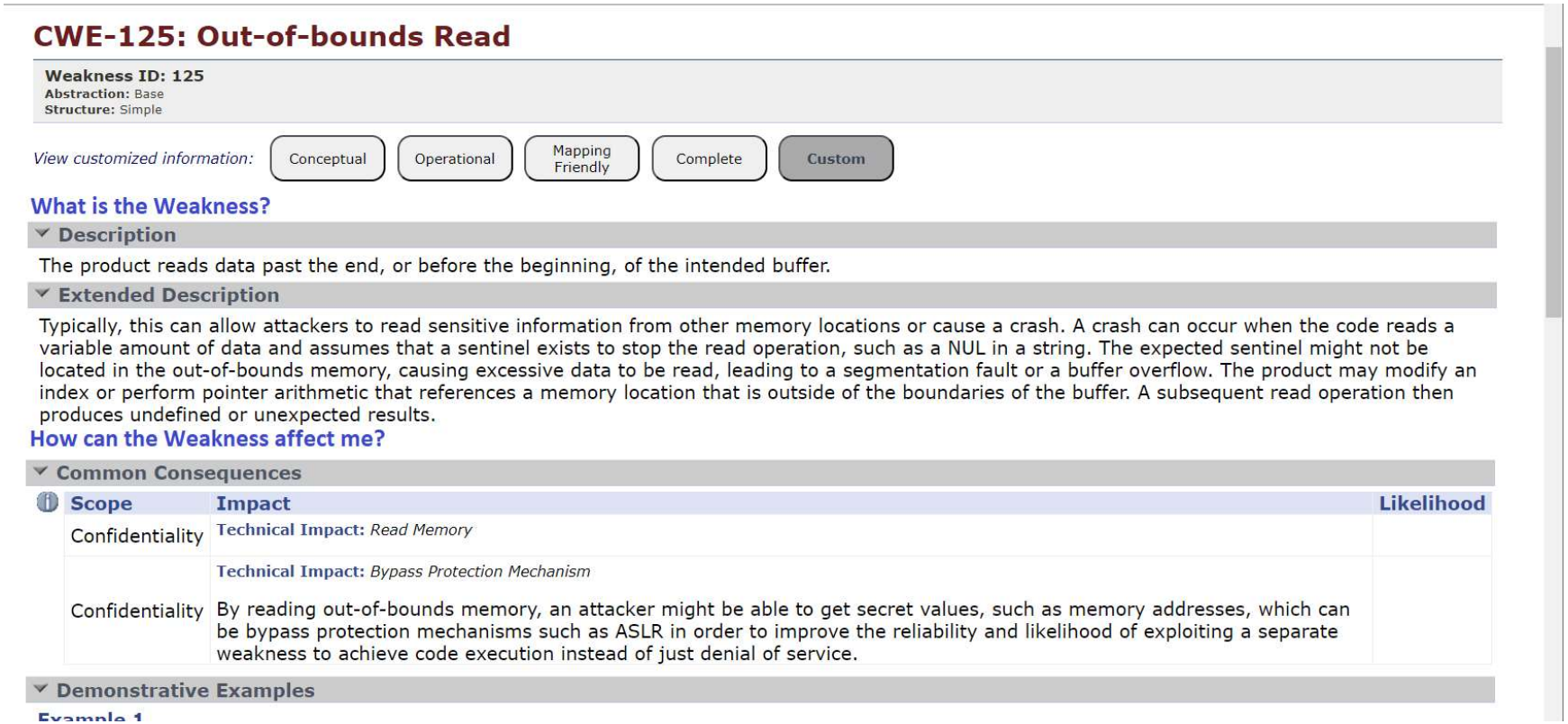
- **Proposed Solution: Reorganize new user content**
  - Remove duplication (e.g., New to CWE and Root Cause Mapping Guidance)
  - Tie together pieces from different places
  - Specific information focused on different types of users

New User Guidance on CWE Entry Pages

- **The Problem: many new users enter the CWE website through a link to a specific CWE weakness page. No clear guidance without taking the time to read through all the new user info (which many won't have the time to do)**

- **Proposed Solution: Add some content and structure to the CWE pages**
  - Make it easier for the novice to find what they need quickly
  - Could add some additional categorization to the CWE weakness entries as proposed on the following two slides

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-

Improve Search

- **The Problem: Search is a generic option that searches the whole website, without the ability to focus in on key items of interest**

- **Proposed Solution: Build a more robust search capability**
  - Advanced search, filter results
    - Select fields to search in
    - Search alternate terms
    - Show only mappable CWEs

What's new in CWE Release 4.15?

- **First installment of major usability improvements that are underway to enhance the understandability, navigability, and usability of CWE content**
- **15 CWE Entry pages now include a concise summary of the weakness along with a visual aid at the top of each entry page**
  - These 15 CWEs were selected based on their inclusion in the CWE top 25 list
  - Full list can be found at: https://cwe.mitre.org/news/archives/news2024.html#july16_CWE_Version_4.15_Now_Available
- **The Alternate Terms, Common Consequences, and Mitigations sections were reordered so that they appear directly below the new summary section**

Open Discussion Topics?

- **Increase motivations to fix weaknesses versus vulnerabilities**
  - Could we find a way to incorporate cost and expense associated with a weakness into CWE?
- **What CWE personas most benefit from weakness visualizations?**
  - Technical writers and security architects whose audience is less technical could benefit from reusing visualizations
  - New or junior level Software or Hardware developers could better understand security issues in visual form
- **CWE persona for technical manager?**

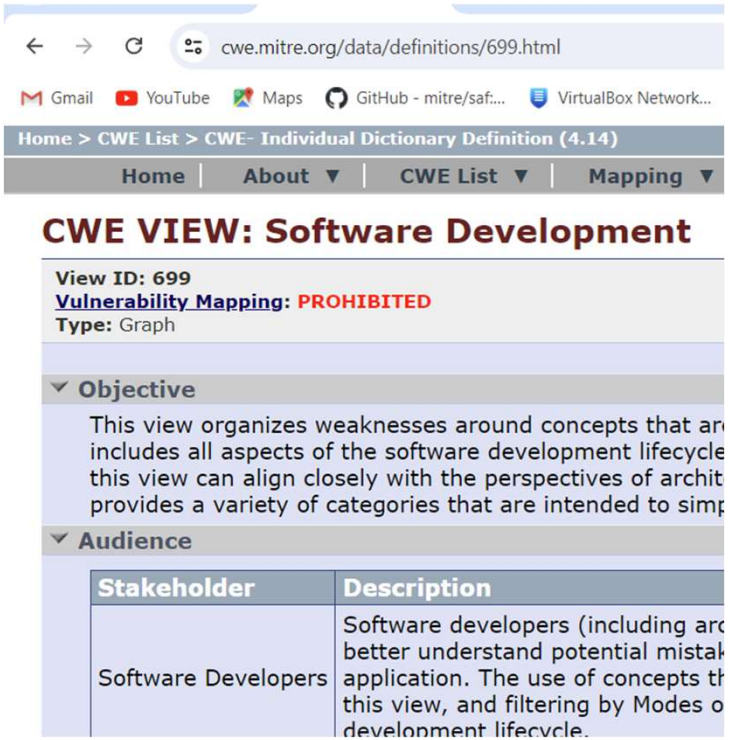
Usability: Macro- and Micro-Level Review

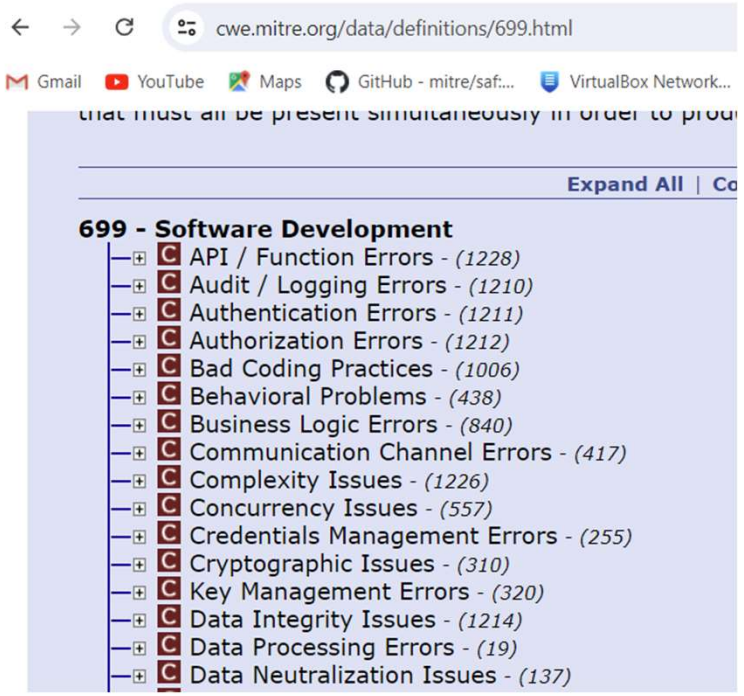
- **Macro –**
  - How to organize weaknesses at structural, site-wide level?
    - Organizing Views
  - Site-wide navigation

- **Micro –**
  - How to define a weakness understandably/accurately
  - Remove redundant and unnecessary information
    - Moving information to a better place
  - Better leverage the schema

View and Category IDs

- **Discussions around removing View and/or Category CWE IDs from page (under title) to prevent mappings**
- **Already removed IDs from titles**
- **IDs will still be used in URL**





Other Macro Changes?

- **Any other thoughts or ideas on Macro usability changes?**

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 Q2-2024
- **Please solicit participation from your contacts**
  - Contact: cwe@mitre.org

Housekeeping

- **CWE UEWG May Meeting**
  - The next regularly scheduled meeting is Wed 7/31
- **The CWE Program is continuously seeking feedback on UEWG activities and priorities during these sessions or via email: cwe@mitre.org**

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 (w**hat to do about the weakness)
- Remaining Elements follow

Prioritizing User Experience Improvements

- **User Experience Working Group was established in response to negative feedback related to CWE's usability**
  - This group has accomplished much in response (e.g., submission guidelines, mapping guidance, "New to CWE?" launch, weakness filtering)
  - We must now begin to address usability and understandability in further ways throughout the corpus structure and down to individual weakness language
- **CWE entry clean-up (across the entire corpus)**
  - Understandability: revising all entries for simpler language; remove redundancy
  - Simplifying descriptions/extended descriptions to leverage other elements appropriately
    - E.g., removing example instance, impacts, etc. language from descriptions
    - Only having one description written in concise, accurate language focusing on the weakness
  - Completeness: ensuring all entries have basic required elements populated
  - Visualizations: adding to entries to explain topics visually (leverage Abhi's top25 examples, build from there)

Usability: Macro- and Micro-Level

- **Macro –**
  - How to organize weaknesses at structural, site-wide level?
    - Organizing Views
  - Site-wide navigation

- **Micro –**
  - How to define a weakness understandably/accurately
  - Remove redundant and unnecessary information
  - Better leverage the schema

CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Weakness ID: 89
Abstraction: Base
Structure: Simple

Description

The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component.

Extended Description

Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, possibly including execution of system commands.

SQL injection has become a common issue with database-driven web sites. The flaw is easily detected, and easily exploited, and as such, any site or product package with even a minimal user base is likely to be subject to an attempted attack of this kind. This flaw depends on the fact that SQL makes no real distinction between the control and data planes.

Relationships

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

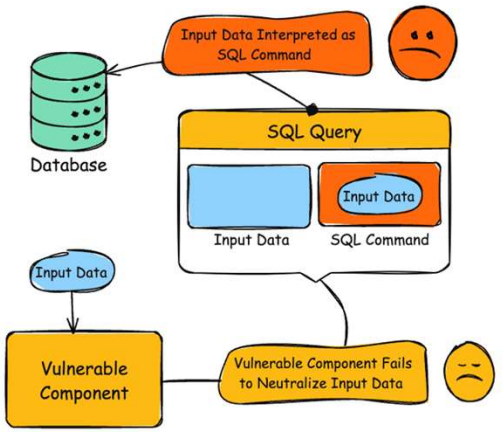
Nature	Type	ID	Name
ChildOf	C	943	Improper Neutralization of Special Elements in Data Query Logic
ParentOf	V	564	SQL Injection: Hibernate
CanFollow	V	456	Missing Initialization of a Variable

CWE-89: Improper Neutralization of Special Elements used in an SQL Command

Weakness ID: 89
Usage for Mapping: ALLOWED ***(this CWE ID may be used to map to real-world vulnerabilities)***
Abstraction: Base

Description

The product uses user-controllable input to construct an SQL command incorrectly or without neutralizing its special elements. This can cause inputs to be interpreted as SQL code instead of ordinary user data. This weakness depends on the fact that SQL makes no real distinction between the control and data planes.



Alternate Terms

'SQL Injection'

Common Consequences

Scope	Impact
Confidentiality	**Technical Impact:** *Read Application Data* A user could construct an SQL command to reveal sensitive information stored in a database

Overall Prioritization

- **At what level should we begin our focus?**
- **Macro (CWE organization and navigation) vs Micro level (weakness understandability)**
- **Micro could include CWE record:**
  - Understandability
  - Completeness
  - Visualizations

A possible path forward

- **Discuss a single CWE and how we could improve usability by**
  - Merging the description and extended descriptions
  - Removing redundant information
  - Moving information that belongs in other parts of the schema
  - Removing vulnerability specific information
- **We could walk though one example in the meeting**
- **Would anyone be interested in trying this exercise with another CWE on their own**
  - Could present their results in the next meeting

Topic 2

New to CWE Updates

Chris Coffin

Prioritizing User Experience Improvements

- **To come: New Strategy and Implementation Plan for usability improvements across CWE corpus aligned to required elements and their requirements as outlined in CWE content suggestion guidelines**

- **Community partners in UEWG are invited to help us tackle this important work**
  - More intentional focus in UEWG
  - CWE Content Web Submission Form for suggested mods on any weakness
  - Content Development Repository (CDR), GitHub repo for community collaboration is coming soon and will provide transparent, collaborative platform on which to work

Recent Discussions on SW Licensing Issues

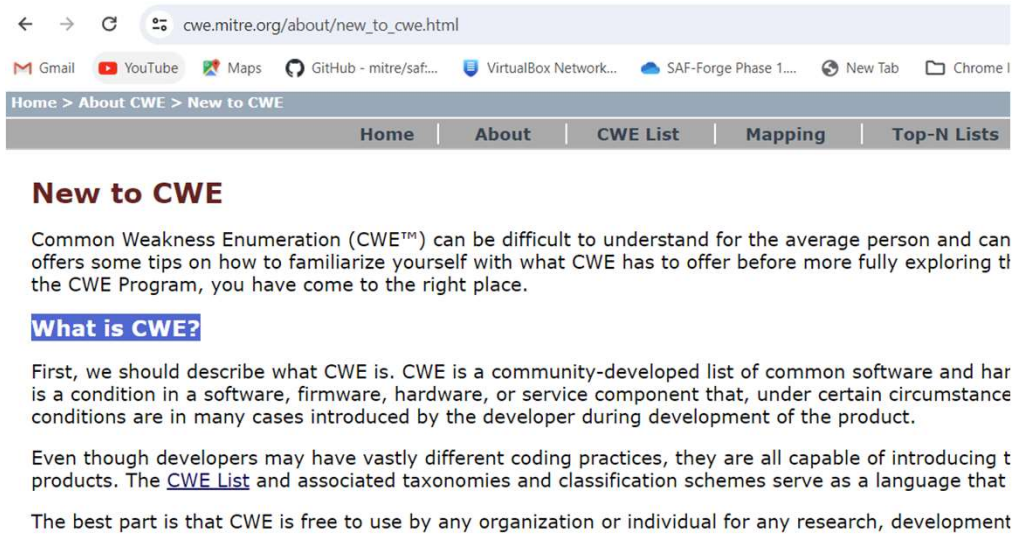
- **Since November, a resurrection of debate around SW licensing issues being in/out of CWE scope on the CWE-Research email listserv**
- **In 2018, it was determined that "improper licensing" was outside of CWE's scope**
  - Rationale: Impact to software and its usage not through the technical exploitation of a software security weakness in architecture, design, or code. Rather, it is through policy/programmatic exploitation.
  - Availability concern comparable with supply chain issues where one disrupts a supplier to stop/limit a product from being delivered, and hence make it not available
- **Recent arguments are varied:**
  - Pro: Misusing SW licenses can negatively affect maintainability, availability
  - Against: Invalid/Improper licenses cannot lead to a vulnerability
  - Pro: CWE absorbed CQE content ~2018; improper licensing is a code quality issue
  - Against: Licensing issues are not a property of SW, but of the society and economy around the SW

CWE Scope

- **CWE's definition of "Weakness":**
  - A condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities
- **Community members sometimes suggest new entries to CWE that do not satisfy this "weakness" definition, but they want CWE to treat them as "weaknesses"**
- **Other times, people effectively suggest the expansion of CWE's scope beyond "traditional" software/hardware**
- **"Scope exclusions" attempt to formalize decisions about what can or cannot be included in CWE as an official "weakness" entry**
  - Have already been used in external submissions

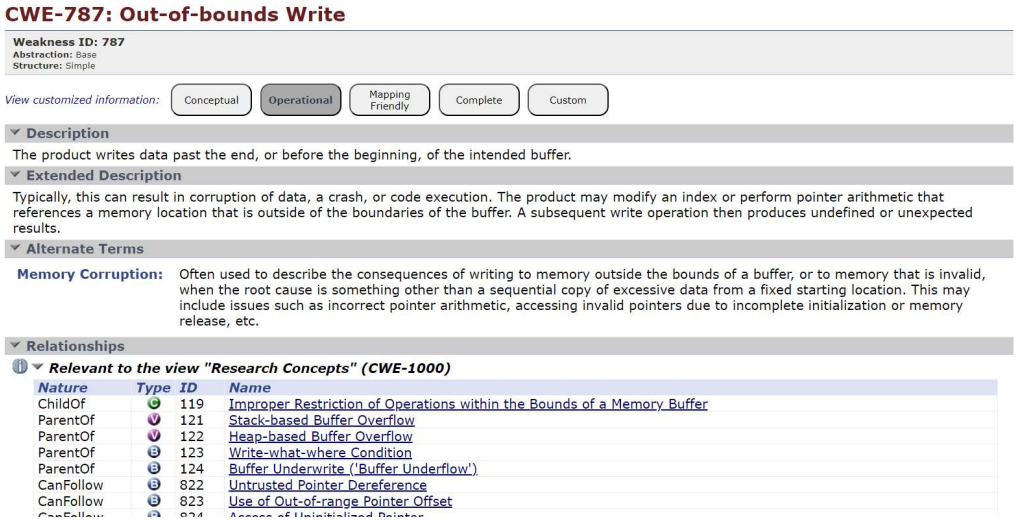
New to CWE – Additional Examples

- The current New to CWE page has one example “CWE-798: Use of Hard-coded Credentials”
- More examples are needed
- Should be simple to understand to closely match the intended audience of the New to CWE page
- Examples for consideration?
  - Might focus on 2023 Top 25 CWEs?
  - CWE-434: Unrestricted Upload of File with Dangerous Type
  - CWE-287: Improper Authentication
  - Others?



New to CWE – Additional Examples – Selection Criteria (Continued)

- What CWE characteristics are important when determining the best examples for new CWE users?
- Is the description easy to understand for a non-developer or industry expert?
- Does it include good examples (demonstrative and observed)?
- Is it a well-known/well-understood weakness?
- Does it exist in the Top 25 list (now or previous years)?
- Others?



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.

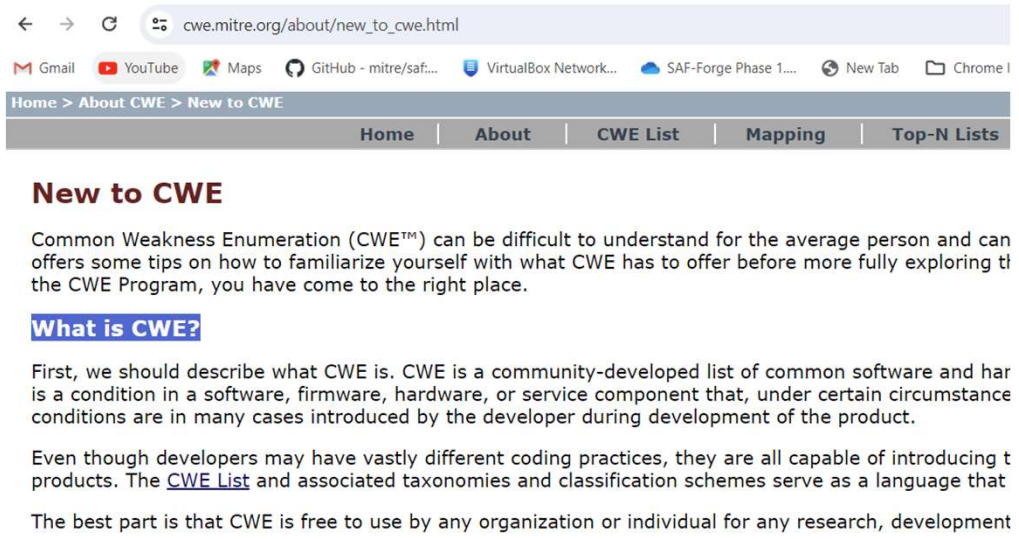
Relationships

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

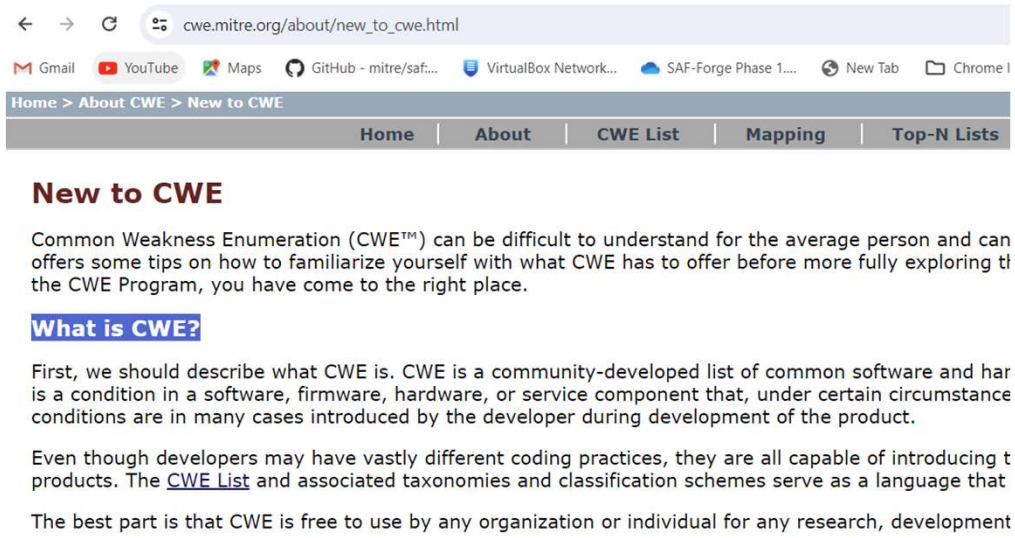
New to CWE Series – Future Topics

- The CWE team is considering developing additional documents as part of a “New to CWE” series
- Keep them short so that new and casual users can read quickly
- Each document would link to others in the series where appropriate
- Topics for consideration?
  - Categorization - Views and Categories
  - How do I navigate the CWE corpus?
  - CWE Hierarchy - Pillars, Classes, Bases, and Variants
  - Others?

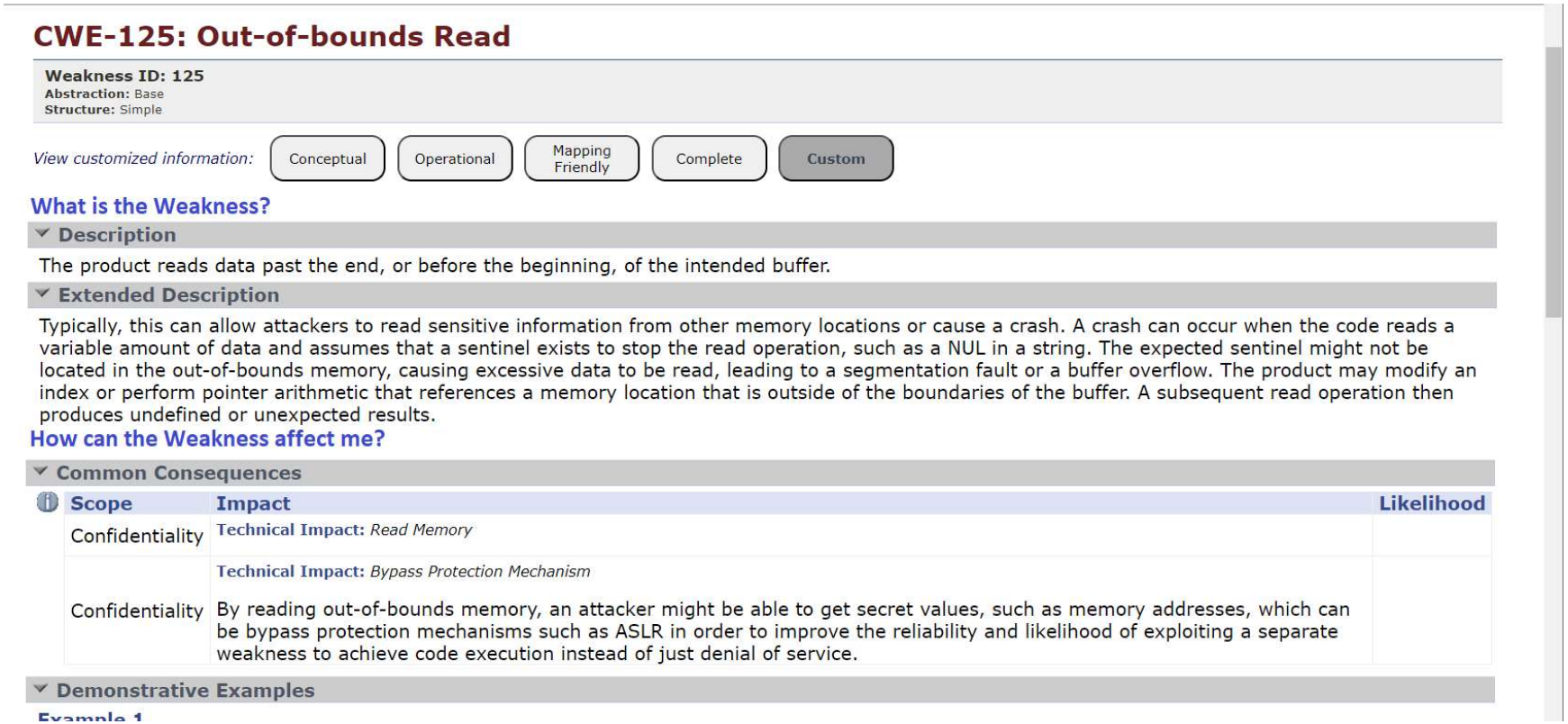


New to CWE Series – How to Navigate CWE

- The CVE -> CWE Mapping Guidance exists on the website today
- The Mapping Methodologies section of the guidance could be reused with some modification
  - Keyword search
  - Views
  - PDF Visualization
- The goal is to create a “New to CWE: How to Navigate CWE” document that could be referenced by the current “New to CWE” page that exists today



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)