CWE User Experience Working Group Meeting

January 31, 2024

Agenda

This meeting is being recorded :-)

- **Purpose**
- **Housekeeping**
- **Primary topics**
  - New to CWE Updates
  - Software Licensing 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 Q1-2024
- **Please solicit participation from your contacts**
  - Contact: cwe@mitre.org

Housekeeping

- **CWE UEWG February Meeting**
  - The next regularly scheduled meeting is Wed 2/28

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

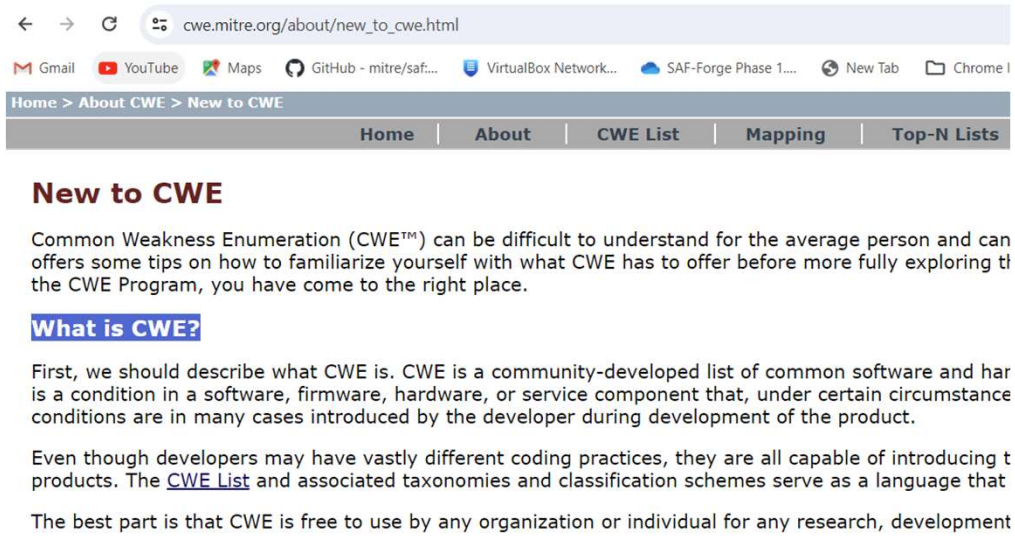
Topic 1

New to CWE Updates

Chris Coffin

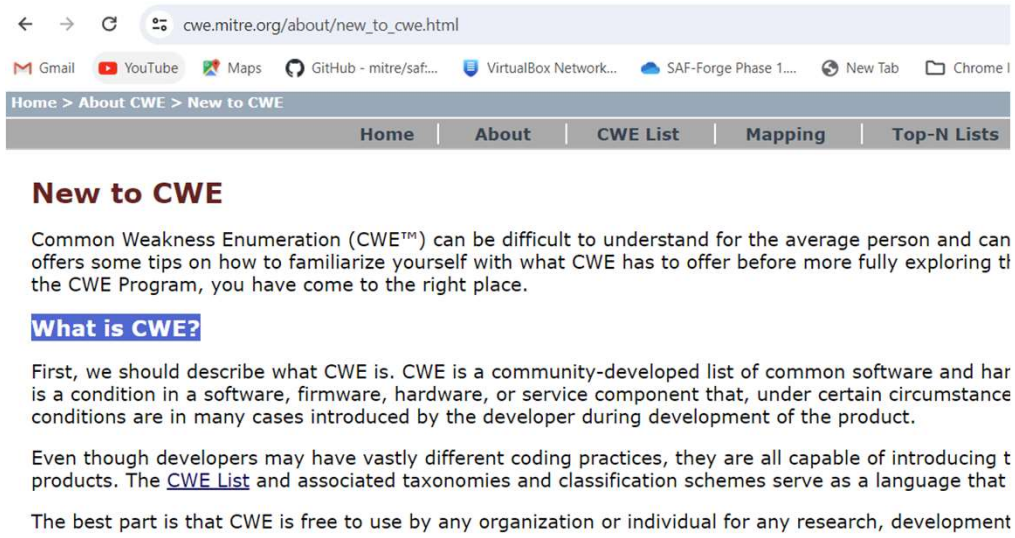
What is it?

- A short introduction to CWE for new and casual viewers of the CWE web site
- Can be found under About menu and in the top right of the home page
- Also provides an example CWE (CWE-798) and discusses how CWE is used today



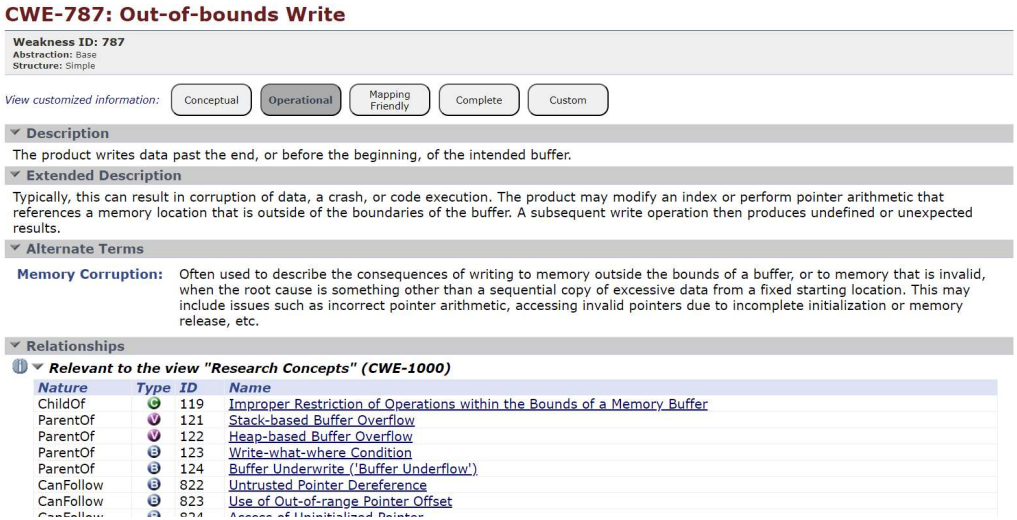
New to CWE – Additional Examples

- 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.

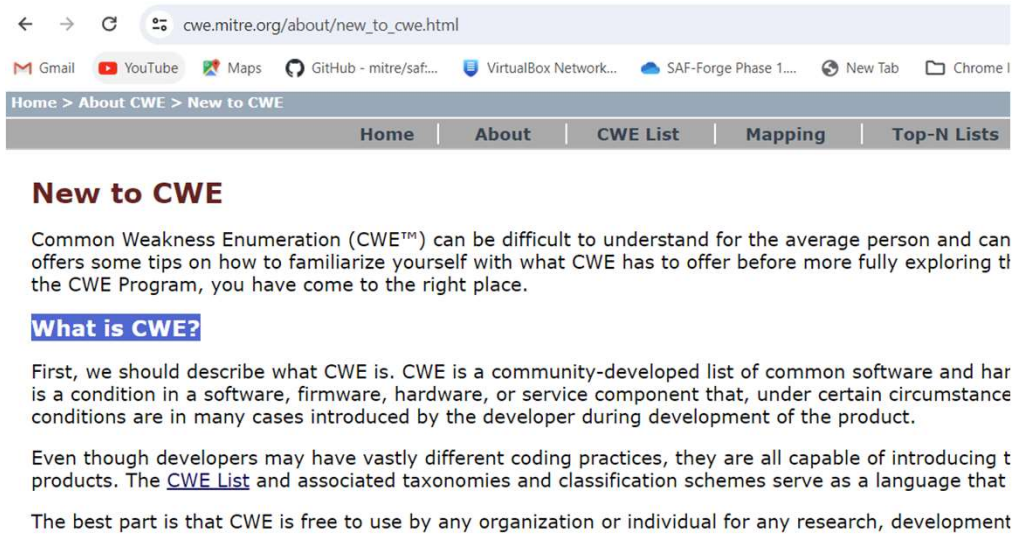
Relationships

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

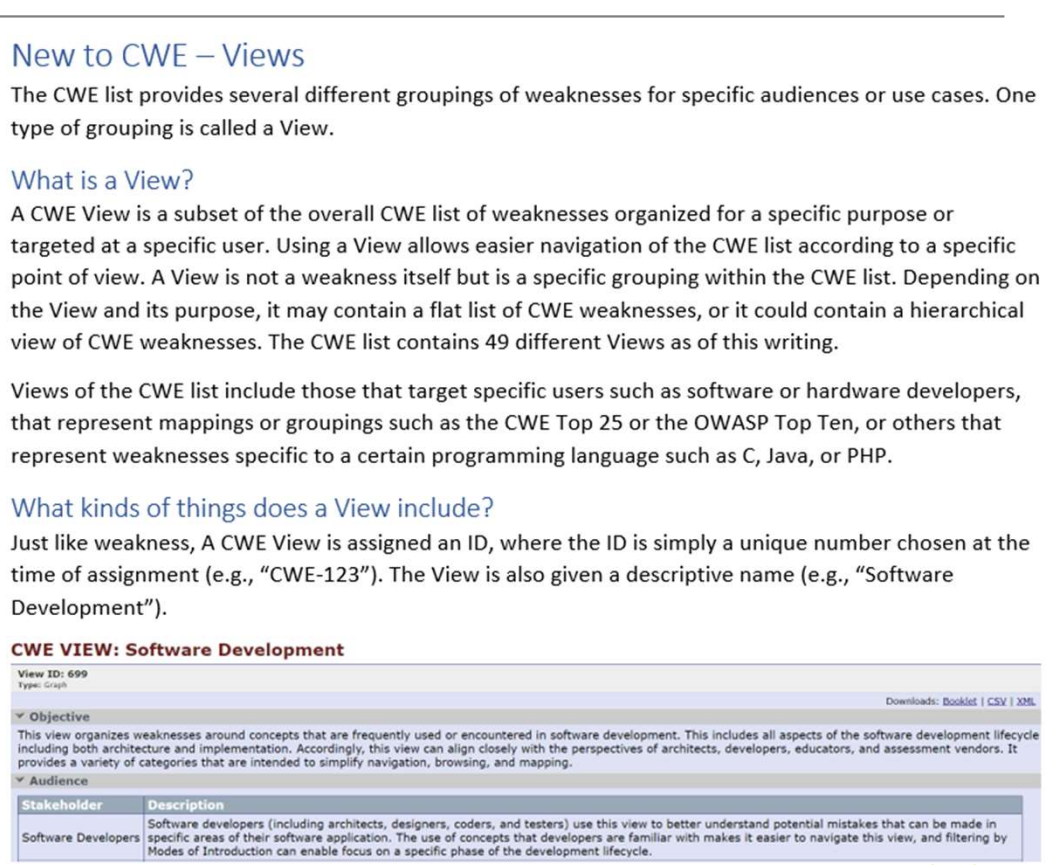
New to CWE Series – Future Topics

- 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 – Views

- Currently drafting a new document in the series to cover Views
- Will send out a draft when we are closer to completion
- Sections currently include:
  - What is a View?
  - What kinds of things does a View include?
  - What are some examples of Views?
  - Where can Views be found on the CWE website?
  - How can CWE Views help me?
- Thoughts on the topics?
  - Topics follow closely to the current New to CWE document that exists today
  - Any additions or removals?



New to CWE – Views

The CWE list provides several different groupings of weaknesses for specific audiences or use cases. One type of grouping is called a View.

What is a View?

A CWE View is a subset of the overall CWE list of weaknesses organized for a specific purpose or targeted at a specific user. Using a View allows easier navigation of the CWE list according to a specific point of view. A View is not a weakness itself but is a specific grouping within the CWE list. Depending on the View and its purpose, it may contain a flat list of CWE weaknesses, or it could contain a hierarchical view of CWE weaknesses. The CWE list contains 49 different Views as of this writing.

Views of the CWE list include those that target specific users such as software or hardware developers, that represent mappings or groupings such as the CWE Top 25 or the OWASP Top Ten, or others that represent weaknesses specific to a certain programming language such as C, Java, or PHP.

What kinds of things does a View include?

Just like weakness, A CWE View is assigned an ID, where the ID is simply a unique number chosen at the time of assignment (e.g., "CWE-123"). The View is also given a descriptive name (e.g., "Software Development").

CWE VIEW: Software Development

View ID: 699
Type: Graph

Downloads: Booklet | CSV | XML

Objective

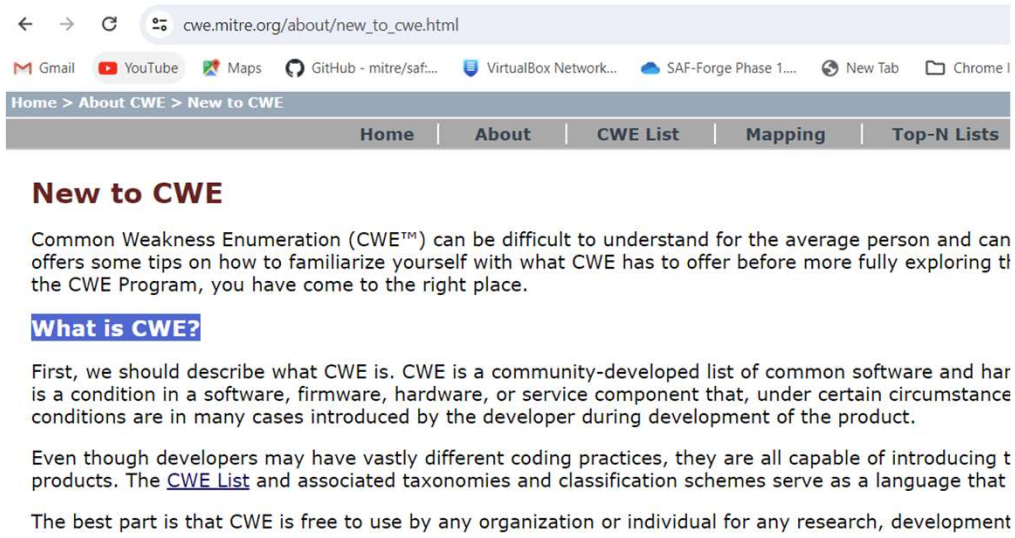
This view organizes weaknesses around concepts that are frequently used or encountered in software development. This includes all aspects of the software development lifecycle including both architecture and implementation. Accordingly, this view can align closely with the perspectives of architects, developers, educators, and assessment vendors. It provides a variety of categories that are intended to simplify navigation, browsing, and mapping.

Audience

Stakeholder	Description
Software Developers	Software developers (including architects, designers, coders, and testers) use this view to better understand potential mistakes that can be made in specific areas of their software application. The use of concepts that developers are familiar with makes it easier to navigate this view, and filtering by Modes of Introduction can enable focus on a specific phase of the development lifecycle.

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



Topic 2

Software Licensing Discussion

Przemyslaw Roguski and Jonathan Hood

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

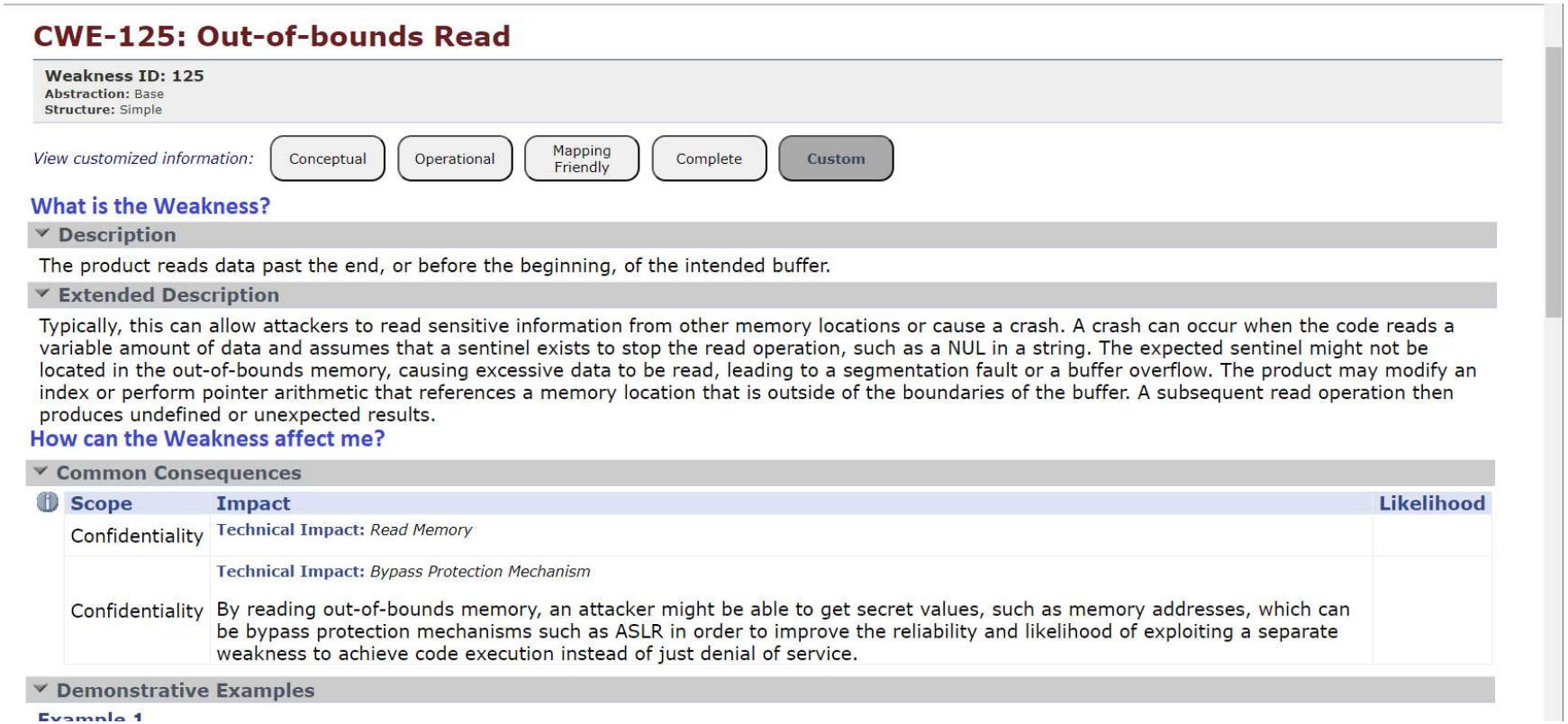
Next Meeting – February 28 @ 12pm

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)