CWE/CAPEC User Experience Working Group Meeting

October 19, 2022

Agenda

This meeting is being recorded :-)

- **Housekeeping**
- **Primary topics**
  - Review CWE & CAPEC recent content releases
  - Feedback: Personas and Presentation Filters
  - Discussion: Possible opportunities for UEWG Activity
- **Reminders and Adjourn**

UEWG: Reminders

- **Mission:** Identifying areas where CWE/CAPEC 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/CAPEC Board**
  - (next quarterly Board meeting TBD Q1-2023)
- **Please solicit participations from your contacts**
  - Contact: cwe@mitre.org & capec@mitre.org

Housekeeping

- **We are working to identify further UEWG opportunities and priorities for FY23**
  - Open discussion today, Topic #3
  - (Underway) Collaborative CWE/CAPEC content development space collaboration space

- **Board provided final comments on personas list and definitions**

Topic

CWE & CAPEC Recent Releases Review

Rich Piazza – CAPEC v3.8

Steve Christey Coley – CWE v4.9

CAPEC v3.8

- **Created the new view of Supply Chain CAPEC entries based on the CISA supply chain life cycle**
- **New CAPECs for Supply Chain domain**:
  - CAPEC-690: Metadata Spoofing, CAPEC-691: Spoof Open-Source Software Metadata,CAPEC-692: Spoof Version Control System Commit Metadata,CAPEC-693: StarJacking, CAPEC-695: RepoJacking
- **New CAPECs for Hardware domain**:
  - CAPEC-682: Exploitation of firmware or ROM code with un-patchable vulnerabilities
  - CAPEC-696: Load Value Injection
- **Other new CAPECs**:
  - CAPEC-694: System Location Discovery
  - CAPEC-697: DHCP Spoofing
- **Updated CAPEC to ATT&CK mapping**

Summary of CWE v4.9 Changes

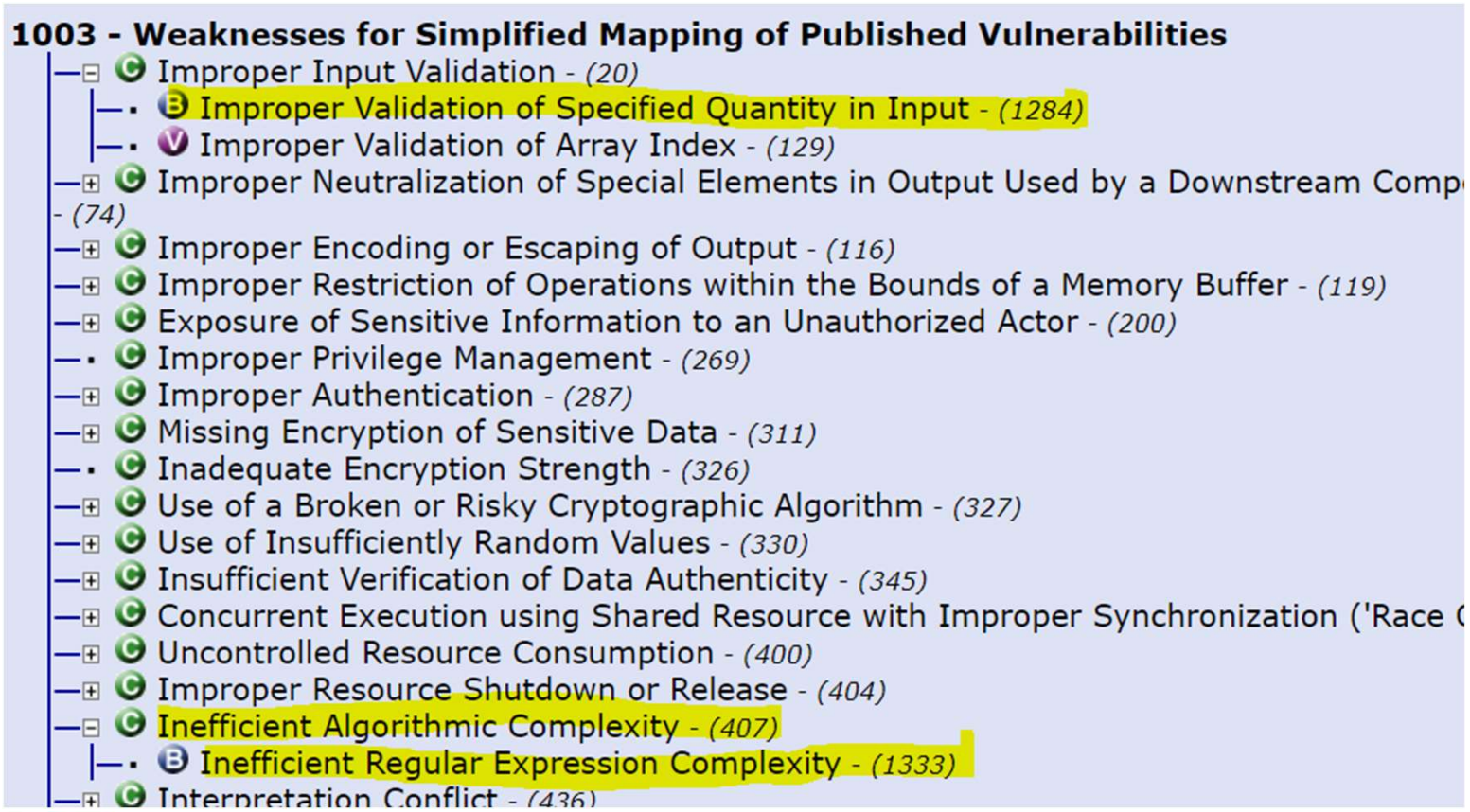
- **Maintenance-heavy release**
  - 693 entries modified (most in small ways)
  - 6 new entries
  - 1 view deprecated

- **Diff report available:**
  - https://cwe.mitre.org/data/reports/diff_reports/v4.8_v4.9.html
  - One comment: the diff reports don't work very well to show details about how entries were changed

Software Content Changes for CWE 4.9

- **External content submissions**
  - Miscellaneous modifications
  - New entry CWE-1389: Incorrect Parsing of Numbers with Different Radices
- **New areas: cloud, ICS/OT**
  - ICEFALL (ICS/OT) CVEs – observed examples
  - Cloud – modify existing entries
    - Demonstrative examples (real-world incidents) have few weakness details
- **Integrate outputs from Top 25 remapping**
  - Modify view 1003
  - Mapping notes - prohibit/discourage mapping to commonly-misused CWEs

View 1003 changes



Software Content Changes for CWE 4.9 (Continued)

- **New CWE entries**
  - Access control subtree overhaul
    - "Weak Authentication," "Weak Credentials," etc. (see next slide)
- **Expand observed example coverage**
  - CVEs in software written in Go or Python
- **xhtml cleanup for CWE REST API**
  - Fix minor inconsistencies and rendering problems in xhtml
- **Software Fault Patterns view updates (888)**
- **Minor schema changes**
  - Consistent capitalization, changing terms, etc.
- **Many other modifications**

Access Control Enhancements – Weak Authentication

- **“Improper” -> “Missing” or “Incorrect” (Weak)**
- **“Incorrect AuthN” could only use more-general CWE-287**
- **Others like authZ have had this distinction for a long time**
- **Use of Weak Credentials (CWE-1391) is a key breakdown from other authN issues**
- **Entries are incomplete (to address in 4.10)**
- **“software” -> “product”**

Example ICS/OT Change – CWE-798: Hard-Coded Credentials

CWE-798: Use of Hard-coded Credentials

Weakness ID: 798
Abstraction: Base
Structure: Simple

View customized information: Theoretical | Operational | Mapping-Friendly | **Complete**

change →

Description

The software contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

Applicable Platforms

Languages

Class: Language-Independent *(Undetermined Prevalence)*

Technologies

Class: Mobile *(Undetermined Prevalence)*

Class: ICS/OT *(Often Prevalent)*

References

[REF-7] Michael Howard and David LeBlanc. "Writing Secure Code". Chapter 8, "Key Management Issues" Page 272. 2nd Edition. Microsoft Press. 2002-12-04. <https://www.microsoftpressstore.com/store/writing-secure-code-9780735617223>.

[REF-729] Johannes Ullrich. "Top 25 Series - Rank 11 - Hardcoded Credentials". SANS Software Security Institute. 2010-03-10. <http://blogs.sans.org/appsecstreetfighter/2010/03/10/top-25-series-rank-11-hardcoded-credentials/>.

[REF-172] Chris Wysopal. "Mobile App Top 10 List". 2010-12-13. <http://www.veracode.com/blog/2010/12/mobile-app-top-10-list/>.

[REF-962] Object Management Group (OMG). "Automated Source Code Security Measure (ASCSM)". ASCSM-CWE-798. 2016-01. <http://www.omg.org/spec/ASCSM/1.0/>.

[REF-1283] Forescout Vedere Labs. "OT:ICEFALL: The legacy of "insecure by design" and its implications for certifications and risk management". 2022-06-20. <https://www.forescout.com/resources/ot-icefall-report/>.

Example ICS/OT Change (2) – CWE-798: Hard-Coded Credentials

```
</connectionStrings>
...
```

Username and password information should not be included in a configuration file or a properties file in cleartext as this will allow anyone who can read the file access to the resource. If possible, encrypt this information.

Example 5

In 2022, the OT:ICEFALL study examined products by 10 different Operational Technology (OT) vendors. The researchers reported 56 vulnerabilities and said that the products were "insecure by design" [REF-1283]. If exploited, these vulnerabilities often allowed adversaries to change how the products operated, ranging from denial of service to changing the code that the products executed. Since these products were often used in industries such as power, electrical, water, and others, there could even be safety implications.

Multiple vendors used hard-coded credentials in their OT products.

Observed Examples

Reference	Description
CVE-2022-29953	Condition Monitor firmware has a maintenance interface with hard-coded credentials
CVE-2022-29964	Distributed Control System (DCS) has hard-coded passwords for local shell access
CVE-2022-30997	Programmable Logic Controller (PLC) has a maintenance service that uses undocumented, hard-coded credentials
CVE-2022-30314	Firmware for a Safety Instrumented System (SIS) has hard-coded credentials for access to boot configuration
CVE-2010-2772	SCADA system uses a hard-coded password to protect back-end database containing authorization information, exploited by Stuxnet worm
CVE-2010-2073	FTP server library uses hard-coded usernames and passwords for three default accounts
CVE-2010-1573	Chain: Router firmware uses hard-coded username and password for access to debug functionality, which can be used to execute arbitrary code
CVE-2008-2369	Server uses hard-coded authentication key

AJS0

Hardware CWE 4.9 Changes

- Re-categorized several CWEs as a result of the new CWE-1388 category from 4.8
- Name changes for CWE-1206 Category (include "Thermal" as mentioned in description) and CWE-1320
- Schema changes: "Not Technology-Specific", new Language Class "HW Description Language"
- Fixed/replaced erroneous demonstrative examples for CWE-1311 and CWE-1303
- Fixed indenting in some code examples (related to REST API cleanup)
- Changed some code example languages from "Other" to Verilog/VHDL
- Added TI "Physical Security Attacks Against Silicon Devices" reference to several CWE entries dealing with Fault Injection, Side Channels, and glitching
- Other modifications

- Ongoing: reorg related to processor issues (Spectre, Meltdown, etc.)

Slide 14

AJS0 Incorporate edits to view-699... CWE-416: Use after free

Alec J Summers, 2022-10-19T16:26:20.102

Topic 2

Feedback: Personas, Definitions, and Presentation Filters

Alec Summers

AJS1

CWE/CAPEC User Personas

To be published on CWE/CAPEC sites in October pending approval

- **Educators:** Teachers, professors, or certification programs that educate developers and system designers how to design more secure products, develop more secure code, and/or how to find vulnerabilities.
- **Hardware Designers/Verification Engineer**: Those who create, integrate, or verify hardware components within the discipline of microelectronics and want to understand, detect, and avoid weaknesses.
- **Security Researchers/Analysts:** Those who look for ways to attack a product by finding weaknesses using manual and/or automated techniques, then reporting the findings to the vendor and/or the general public.
- **Software Developers/Assurance Engineers**: Those who create, maintain, and test software, using various languages and frameworks, who want to understand, detect, and avoid weaknesses
- **Technical Writers:** Those who communicate advanced technical concepts as clearly, accurately, and comprehensively as possible to their intended audience (e.g., code analysis tool users or system designers)
- **Tool Develo**AJS0**rs:** Developers of security products and services for finding weaknesses, attacking systems, and reporting findings to users

Slide 16

AJS0 Tool “Designers” instead of Developers
Alec J Summers, 2022-10-19T16:40:23.593

AJS1 Tools are going to change over time… not only discovering weaknesses, but testing if weaknesses are not present
Alec J Summers, 2022-10-19T16:41:16.857

AJS1 0 (Security designers?)
Alec J Summers, 2022-10-19T16:42:54.490

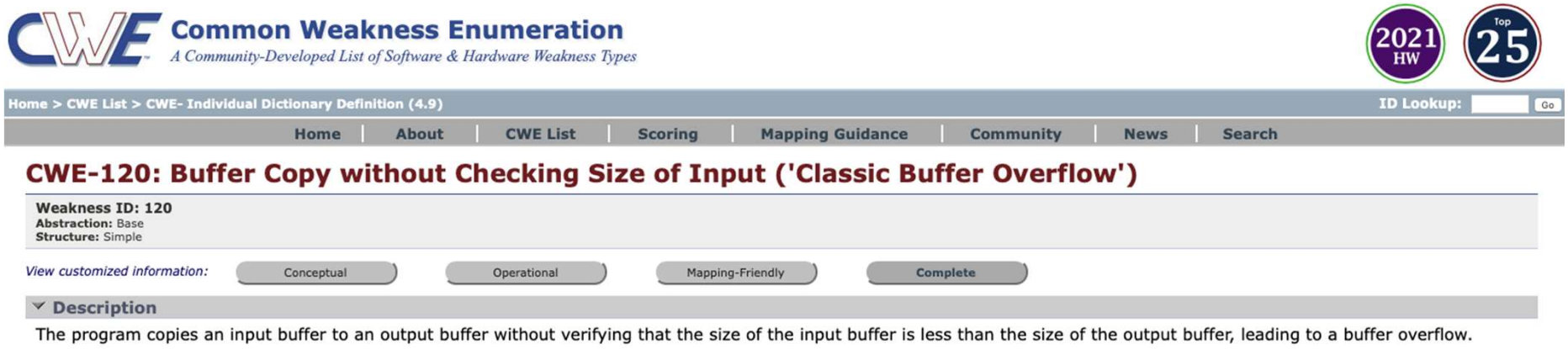
Modernizing Definitions on CWE/CAPEC Sites

Term	Definition	Authority	Authorities Doc
Vulnerability	A flaw in a software, firmware, hardware, or service component resulting from a weakness that can be exploited, causing a negative impact to the confidentiality, integrity, or availability of an impacted component or components	CVE	website
Weakness	A condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities	n/a	edited from def on CWE website
Attack Pattern	The common approach and attributes related to the exploitation of a weakness in a software, firmware, hardware, or service component.	n/a	edited from def on CAPEC website

“View Customized Information”

- **Theoretical:**
  - For users who are interested in the practical
- **Operational:**
  - For users who are concerned with the practical application and details about the nature of the weakness and how to prevent it from happening
- **Mapping-Friendly:**
  - For users who are mapping an issue to CWE/CAPEC IDs, i.e., finding the most appropriate CWE for a specific issue (e.g., a CVE record)
- **Complete:**
  - For users who wish to see all available information for the CWE/CAPEC entry

Screenshot



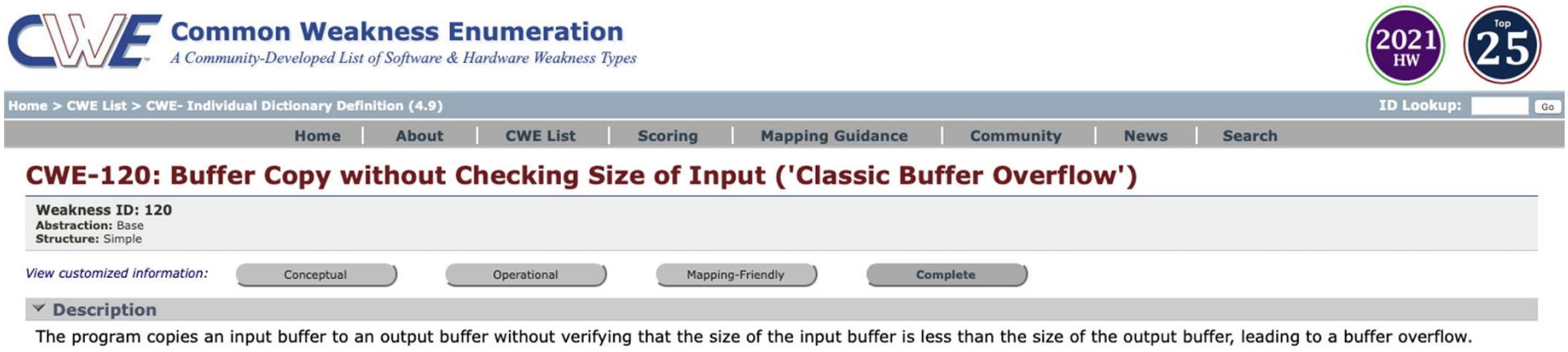
Adjusting our Presentation Filters

- **Better serve our user persona needs**
- **"Two-Types" Proposal by Premyslaw Roguski (Red Hat)**
  1. Theoretical: users who are more focused on the theoretical aspects of the weaknesses
     - Educators (teachers, professors, solution architects who design the systems' requirements)
     - Technical Writers (people responsible for security content, security blogs and articles)
     - Project and Program Managers who need some level of understanding about security and weaknesses
  2. Technical: users who are managing the security issues and need more details about the nature of the weakness and how to prevent this from happening
     - Tool Developers, Security Researchers, Pentesters, Incident Response Analysts

Proposed Data Elements for Each Group

- **Theoretical:**
  - Description, Extended Description, Alternate Terms, Common Consequences
- **Technical (operational?):**
  - Description, Extended Description, Alternate Terms, Modes of Introduction, Demonstrative Examples, Observed Examples (include recent CVE tracking data*), Potential Mitigations, Relationships
- **Mapping-Friendly:**
  - Description, Extended Description, Alternate Terms, ~~Modes of Introduction, Likelihood of Exploitation~~, Taxonomy Mappings, Memberships, Notes
- **Complete:**
  - all

Screenshot



Thoughts? Comments?

Customizing CAPEC Information

- **Currently**
  - Basic, Complete
- **Would the new CWE model for filters work for CAPEC?**
  - Conceptual
  - Operational
  - Mapping-Friendly
  - Complete

CAPEC CONTENT	
BASIC	COMPLETE
Description	Description
Extended_Description	Extended_Description
Relationships	Relationships
Execution_Flow	Execution_Flow
Prerequisites	Prerequisites
Mitigations	Mitigations
Related_Weaknesses	Likelihood_Of_Attack
	Alternate_Terms
	Typical_Severity
	Skills_Required
	Resources_Required
	Indicators
	Consequences
	Example_Instances
	Related_Weaknesses
	Taxonomy_Mappings
	References
	Notes
	Content_History

Topic 3

Possible UEWG Activities and Focus Areas

Open Discussion

AJS1

UEWG Activities Brainstorm

- **Other potential opportunities:**
  - Improve CWE and CAPEC usability
  - Features we could implement / expand

- **Engaging other users?**
- **Engaging other working groups?**

- **Other thoughts?**

Slide 25

AJS0 CWSS? What can be done here?
Alec J Summers, 2022-10-19T16:54:38.398

AJS1 Suzannah: Seek IRB approval to get feedback from students if they have any thoughts or difficulties
Alec J Summers, 2022-10-19T16:57:35.184

Next Meeting – November 16 @ 12pm

PLEASE CONTACT WITH ANY QUESTIONS OR THOUGHTS

CWE@MITRE.ORG **CAPEC@MITRE.ORG**