Vulnerability Root Cause Mapping with CWE

Challenges, Solutions, and Insights from Grounded LLM-based Analysis

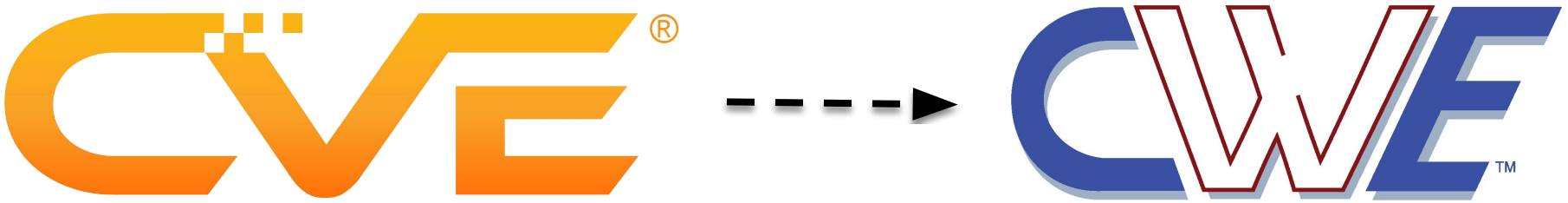
Chris Madden, Yahoo
Alec Summers, MITRE

Key Takeaways

- **Root cause mapping is an important part of vulnerability management and disclosure**
- **The better the vulnerability data, the better the root cause mapping (both manual and AI-driven)**
- **CWE mapping is hard, but resources and community collaboration has developed guidance material, and is actively developing AI-driven mapping capabilities**
- **LLMs (especially those grounded and trained) are good enough *and* cheap enough to be used for Root Cause Mapping and CVE Enrichment in an interactive or bulk (e.g., all CVEs) supervised manner**

Background

- **Root Cause Mapping (RCM) is the identification of the underlying cause(s) of a vulnerability**
- **This is best done by correlating CVE Records and/or bug or vulnerability tickets with CWEs**



The intersection of the CVE and CWE Programs can result in meaningful improvements across product security

RCM Value

- **Enables vulnerability trend analysis and greater visibility into their patterns over time**
- **Illuminates where investments, policy, and practices can address the weaknesses responsible for product vulnerabilities so that they can be eliminated**
- **Provides further insight to potential "exploitability" based on weakness type**
- **Provides valuable feedback loop into an SDLC or architecture design planning**

RCM Challenge

- **Multi-faceted problem**
  - RCM is a different way of thinking (i.e., not vulnerability- or attack-defined mindset)
  - Technically difficult / time-consuming
  - Missing demand historically from downstream consumers
    - this is changing :-)

- **The CWE repository is a vast body of knowledge that is technically detailed and densely presented material**
  - Designed to support a variety of technical stakeholders to identify and describe "weakness types" (i.e., root causes of vulnerabilities) in a hierarchical information corpus
  - Difficult to understand and navigate by the lay person
  - Result: "super users" and everyone else

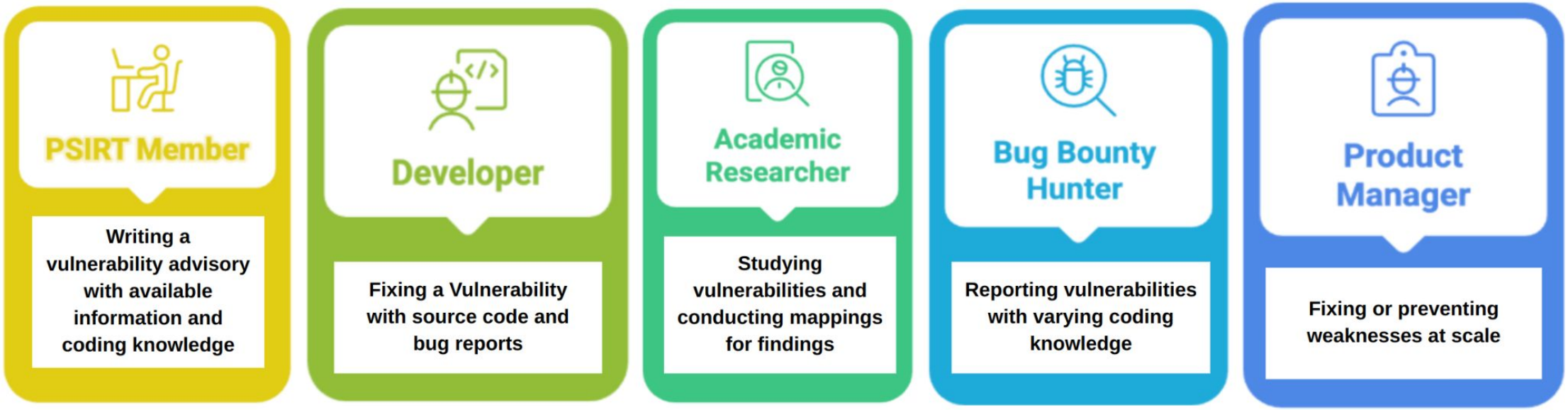
2024 vs 2025

- **CWE is making headway in addressing usability at macro- and micro-levels**
  - i.e., corpus-wide vs. individual weakness content
- **Centralized vs Decentralized approach for RCM at scale**
  - We are seeing broad adoption of CWE mapping throughout CNA community
- **Root Cause Mapping Working Group**
  - Bringing together CVE and CWE communities
  - VulnCon 2024 announcement, VulnCon 2025 share results

Increasing Adoption and *Quality* of RCM

- **Seeing broad adoption of RCM as part of vulnerability disclosure across majority of CNAs**
- **RCM guidance and Mapping Labels / Notes have helped, but CWE is still challenging to navigate**
  - Traditional search helps but does not fulfill need for more precise information retrieval or language interpretation
- **Opportunity exists for CWE Program to work with researchers to enable greater interactivity with its information for faster and more precise RCM**
  - Leverage advancements in large language model (LLM) capabilities

CWE ChatBot: Users and Use Cases

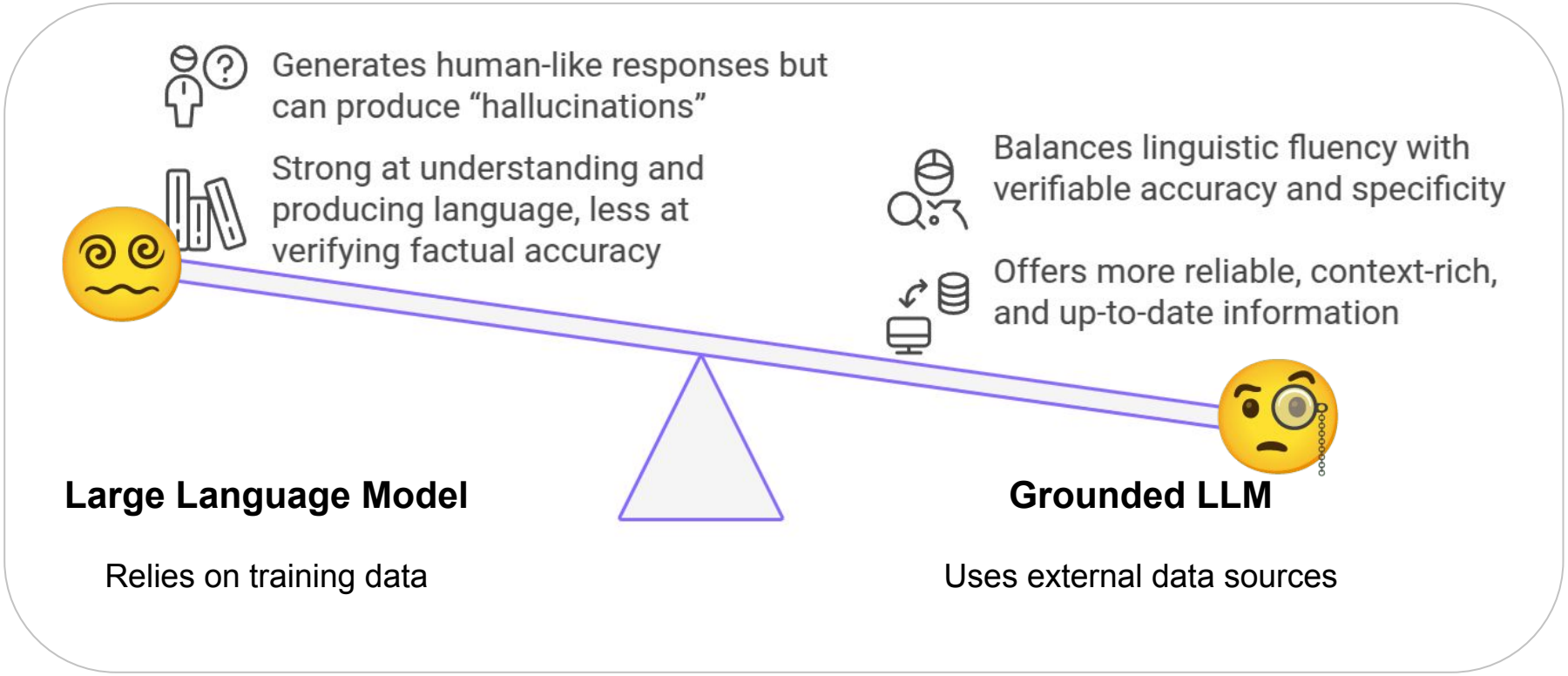


Advancements in LLM technologies' capabilities make it possible to ***interact*** with the CWE corpus as opposed to simply search or browse it

Exploring LLM capabilities in vulnerability root cause mapping

LLM vs grounded LLM

LLM vs Grounded LLM



bad at declining to answer questions they couldn’t answer accurately, offering incorrect or speculative answers instead.
…inaccurate answers with alarming confidence

https://www.cjr.org/tow_center/we-compared-eight-ai-search-engines-theyre-all-bad-at-citing-news.php March 2025

Grounding LLMs significantly reduces hallucinations.

Exploring LLM capabilities in vulnerability root cause mapping

LLM Capabilities in Vulnerability Root Cause Mapping

CWE assignment, info

Enrich and preserve CVE info

6 Bulk CVE Checking

Conduct bulk checks on published CVEs for accuracy and completeness.

Existing CVE-CWE mappings are checked. A shortlist of CWEs can be proposed for a CVE with rationale, confidence. Non-interactive.

5 Interactive QnA

Engage in interactive discussions and Q&A sessions using a CWE list.

ChatBot: A user can ask any questions of the CWE List e.g.

- What CWEs related to XSS?
- What is the best CWE for _____
- Please explain CWE-123.

4 Assign CWEs

Assign relevant CWE identifiers based on CVE information.

A shortlist of CWEs can be proposed for a CVE with rationale, confidence

3 Summarize Content

Summarize reference content using vulnerability keyphrases.

Reference Content can be summarized by vulnerability keyphrases (and preserved)

2 Extract Keyphrases

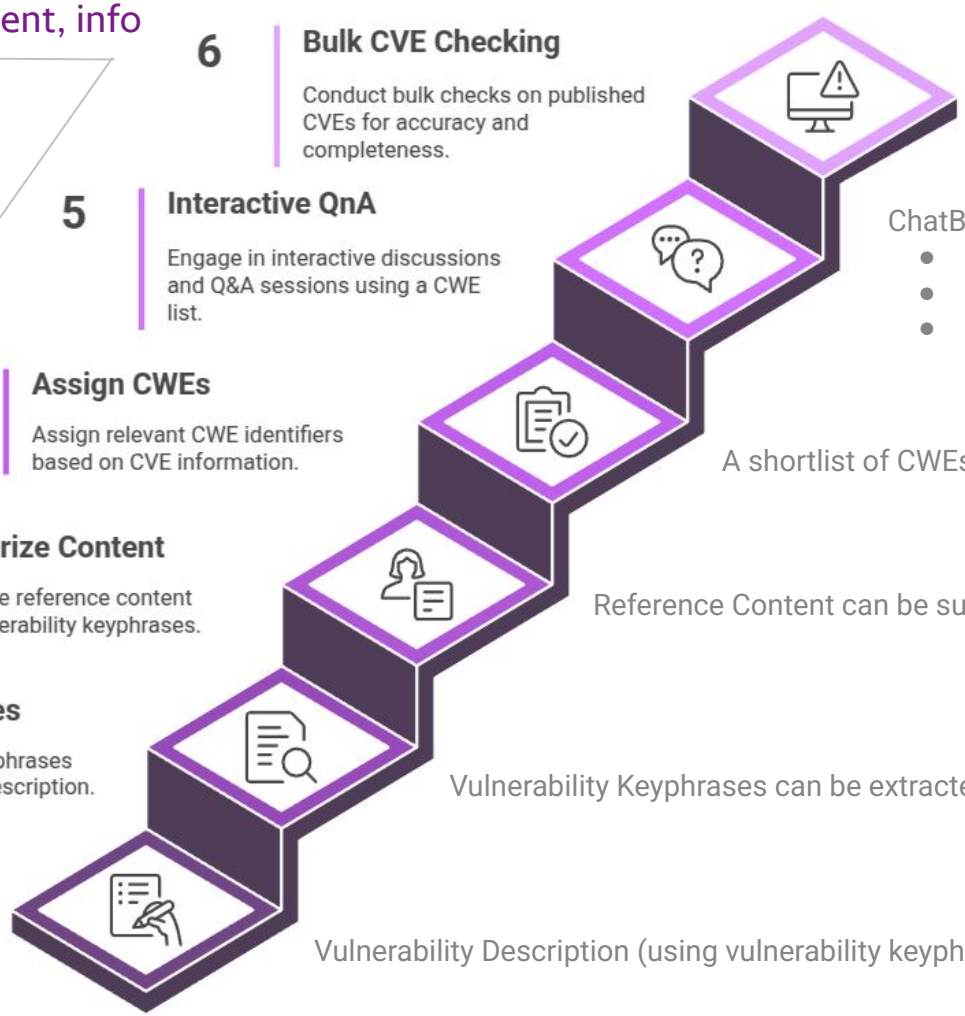
Identify and extract keyphrases from the vulnerability description.

Vulnerability Keyphrases can be extracted from an existing Vulnerability Description

1 Create Description

Develop a detailed vulnerability description from reference content.

Vulnerability Description (using vulnerability keyphrases template) can be created from advisories or patches



The foundation is good CVE data! Complete, accurate and timely data.

Create CVE Description

Prompt to an LLM

"using the information in <LINK TO ADVISORY>, provide a CVE Description that fits the following template
[VULNERABILITY TYPE] in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows [ATTACKER] to [IMPACT] via [VECTOR]"

https://www.cve.org/Resources/General/Key-Details-Phrasing.pdf

1/23 Original CVE

"Mail Spoofing"

"Mail Spoofing in Email Functionality in _____ versions prior to _____ allows remote attackers to send emails with forged sender addresses, potentially leading to phishing attacks or unauthorized information dissemination."

1/24 PoC CVE

1/29 New CVE

"A mail spoofing vulnerability in _____ allows attackers to forge email headers, making it appear as though messages are sent from trusted sources."

LLMs make it easy to create good quality CVE Descriptions from existing reference / advisory material.

1 **Create Description**

Develop a detailed vulnerability description from reference content.

Extract Key Phrases from CVE Description

Description

KeyPhrases

MITRE Technical Impacts


```
{
    "cveId": "CVE-2020-3118",
    "version": "1.0.0",
    "timestamp": "2024-11-03T18:37:50.907685+00:00",
    "description": "A vulnerability in the Cisco Discovery Protocol implementation for
Cisco IOS XR Software could allow an unauthenticated, adjacent attacker to execute
arbitrary code or cause a reload on an affected device. The vulnerability is due to
improper validation of string input from certain fields in Cisco Discovery Protocol
messages. An attacker could exploit this vulnerability by sending a malicious Cisco
Discovery Protocol packet to an affected device. A successful exploit could allow the
attacker to cause a stack overflow, which could allow the attacker to execute arbitrary
code with administrative privileges on an affected device. Cisco Discovery Protocol is a
Layer 2 protocol. To exploit this vulnerability, an attacker must be in the same
broadcast domain as the affected device (Layer 2 adjacent).",
    "keyphrases": {
        "rootcause": "improper validation of string input",
        "weakness": "stack overflow",
        "impact": [
            "execute arbitrary code",
            "cause a reload"
        ],
        "vector": "malicious Cisco Discovery Protocol packet",
        "attacker": "unauthenticated adjacent attacker",
        "product": "Cisco IOS XR Software",
        "version": "",
        "component": "Cisco Discovery Protocol implementation"
    },
    "mitreTechnicalImpacts": [
        "Denial-of-Service: resource consumption",
        "Denial-of-Service: unreliable execution",
        "Execute unauthorized code or commands"
    ]
}
```


https://github.com/CyberSecAI/cve_info/blob/main/2020/3xxx/CVE-2020-3118.json

- [VULNTYPE] in [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] allows [ATTACKER] to [IMPACT] via [VECTOR].
- [COMPONENT] in [VENDOR] [PRODUCT] [VERSION] [ROOT CAUSE], which allows [ATTACKER] to [IMPACT] via [VECTOR].

https://www.cve.org/Resources/General/Key-Details-Phrasing.pdf

"While there are a large number of weaknesses in CWE, there appear to be only eight different consequences or technical impacts to which these failures lead"

https://cwe.mitre.org/community/swa/priority.html

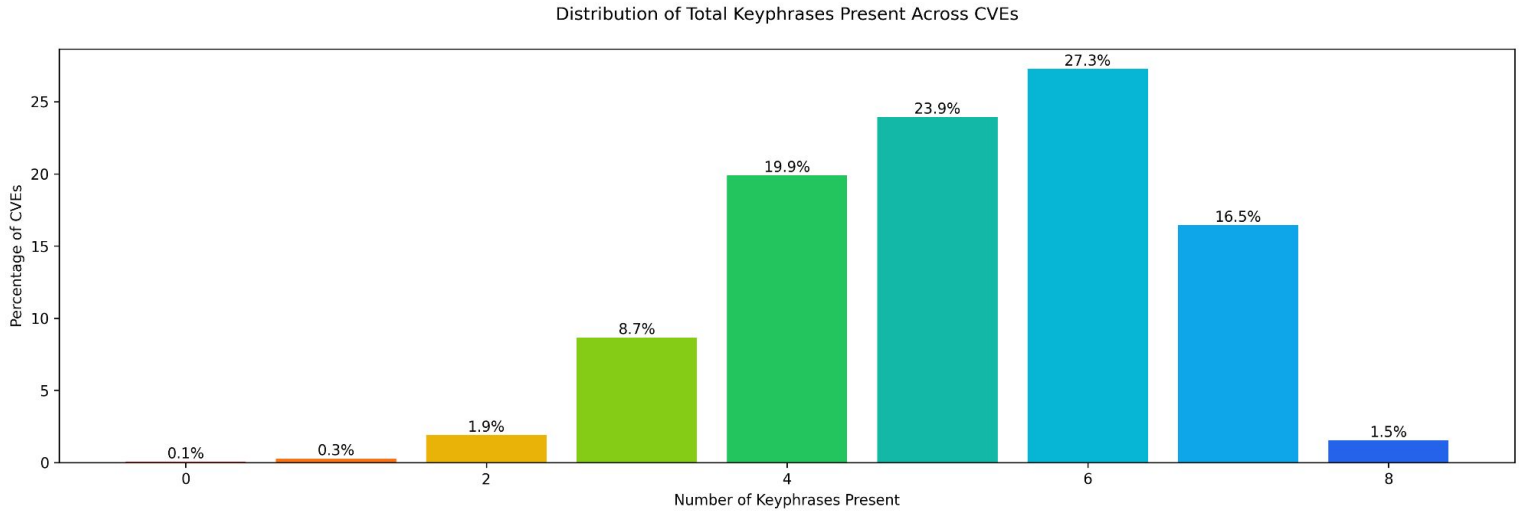
CVE 25 YEARS

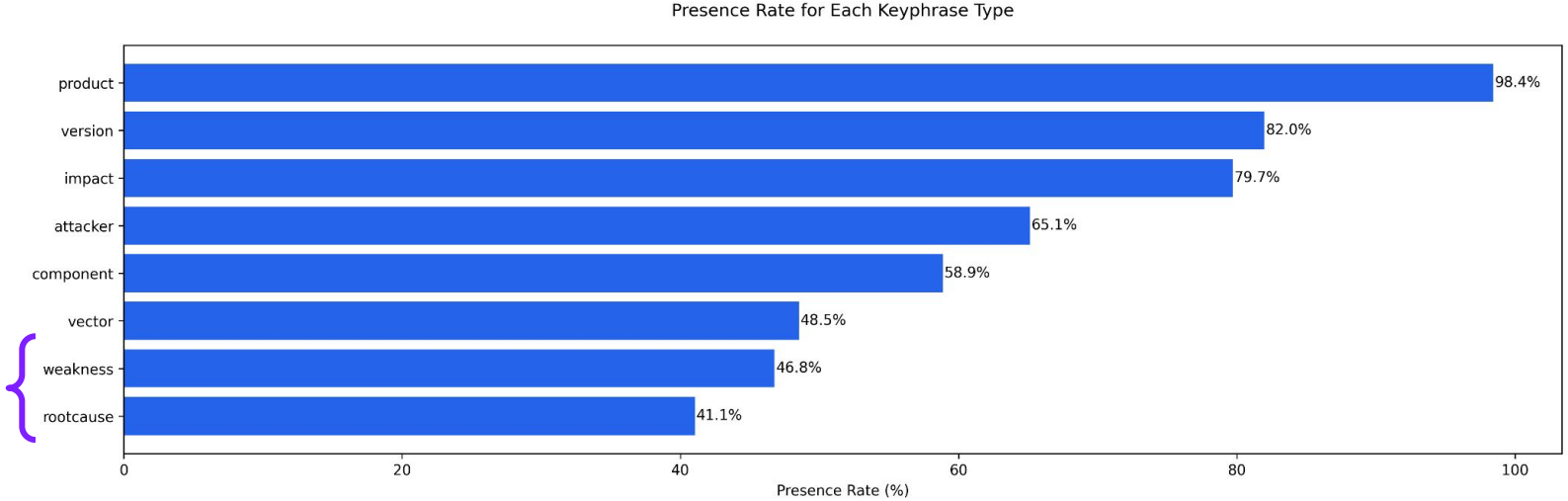
LLM used here to extract KeyPhrases from CVE Descriptions for ~260K CVEs to end of 2024

2 **Extract Keyphrases**

Identify and extract keyphrases from the vulnerability description.

CVE Description KeyPhrases as a Measure of Quality





https://github.com/CyberSecAI/VulnerabilityDescriptionQualityChecker is an analysis of the data in https://github.com/CyberSecAI/cve_info

https://www.cve.org/Resources/General/Key-Details-Phrasing.pdf

CVE Description Keyphrases can be used as a measure of CVE Description quality.
Rootcause and Weakness are the least common Keyphrases

Summarize CVE Reference Content

Input CVE Data

- **CVE Description**
- **Key Phrases** rootcause, weakness
- **CVE Links** Content Summary
- **Similar CVE** CWE consensus

CVE Description

A Python **command injection** vulnerability exists in the `SagemakerLLM` class's `complete()` method within `./private_gpt/components/llm/custom/sagemaker.py` of the imartinez/privategpt application, versions up to and including 0.3.0. The vulnerability arises due to the **use of the `eval()` function** to parse a string received from a remote AWS SageMaker LLM endpoint into a dictionary. This method of parsing is unsafe as it can execute arbitrary Python code contained within the response. An attacker can exploit this vulnerability by manipulating the response from the AWS SageMaker LLM endpoint to include malicious Python code, leading to potential execution of arbitrary commands on the system hosting the application. The issue is fixed in version 0.6.0.

https://nvd.nist.gov/vuln/detail/CVE-2024-4343

KeyPhrases

```
"keyphrases": {
    "rootcause": "use of eval function",
    "weakness": "command injection",
```

https://github.com/CyberSecAI/cve_info/blob/main/2024/4xxx/CVE-2024-4343.json

Hyperlink
https://github.com/imartinez/privategpt/commit/86368c61760c9cee5d977131d23ad2a3e063cbe9
https://huntr.com/bounties/1d1e8f06-ec45-4b17-ae24-b83a41304c15

Summary of CVE Reference Link Content

The provided content relates to **CVE-2024-4343** by detailing a commit that addresses a vulnerability involving the use of `eval()`.

Root cause of vulnerability:

The code was previously using Python's `eval()` function to parse the response from an AWS Sagemaker endpoint. `eval()` can execute arbitrary code, making it unsafe.

Weaknesses/vulnerabilities present:

- **Code injection:** The use of `eval()` allowed for the potential injection of malicious code via a crafted response from the Sagemaker endpoint.

Impact of exploitation:

An attacker could potentially execute arbitrary code on the server if they could control the response from the Sagemaker endpoint. This could lead to various impacts, including data breaches, service disruption, and complete system compromise.

Attack vectors:

The attack vector involves manipulating the response from the AWS Sagemaker endpoint. This could be done by compromising the Sagemaker endpoint or through some other vulnerability in how the endpoint is interacted with.

Required attacker capabilities/position:

The attacker would need to control the response from the AWS Sagemaker endpoint. This could involve compromising the Sagemaker endpoint, or exploiting a vulnerability in how the application communicates with it. The attacker does not need to be local.

Mitigation:

The vulnerability was mitigated by replacing `eval(response_str)` with `json.loads(response_str)`. This ensures that the response is parsed as JSON data rather than being interpreted as executable code.

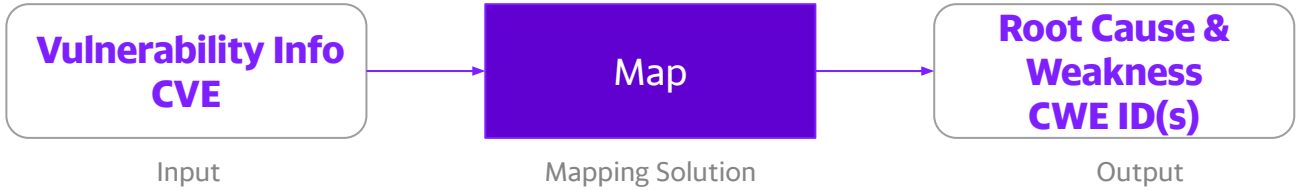
https://github.com/CyberSecAI/cve_info_refs/blob/main/2024/4xxx/CVE-2024-4343/refined/refined.md

CVE Reference Link content often contains more info about the weakness/rootcause that the CVE Description alone.
This was not a "command injection" vulnerability: Description is wrong!
LLMs used here to Summarize Content from CVE References for ~260K CVEs to end of 2024

3 **Summarize Content**

Summarize reference content using vulnerability keyphrases.

Assign CWEs to a CVE



4 **Assign CWEs**

Assign relevant CWE identifiers based on CVE information.

A shortlist of CWEs can be proposed for a CVE with rationale, confidence

Grounded ChatBot Proof Of Concept Setup

💁 A CWE-Expert can be built for free in a browser in less than 1 minute using the instruction and CWE Corpus files provided here.

You can ask your CWE-Expert

- any questions about CWEs
- to assign CWEs to vulnerability descriptions

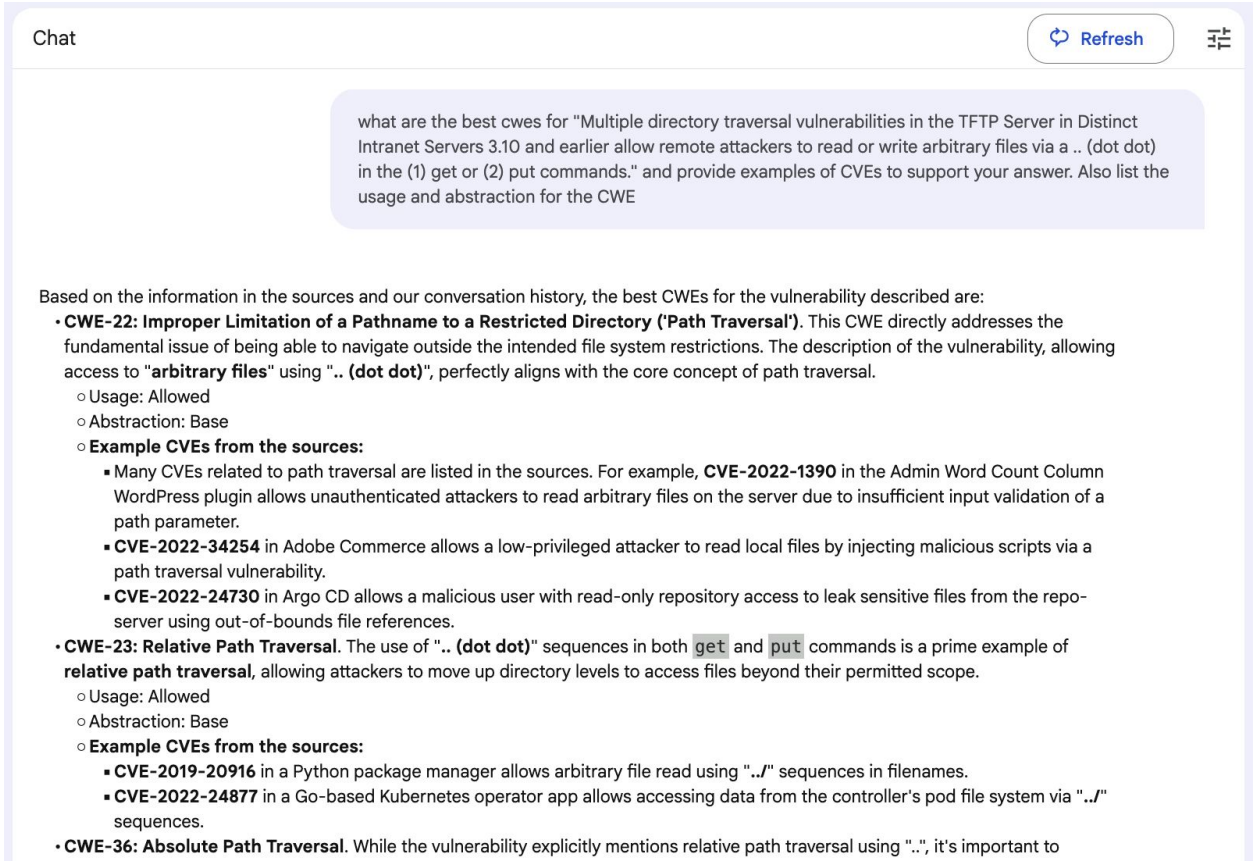
Demos

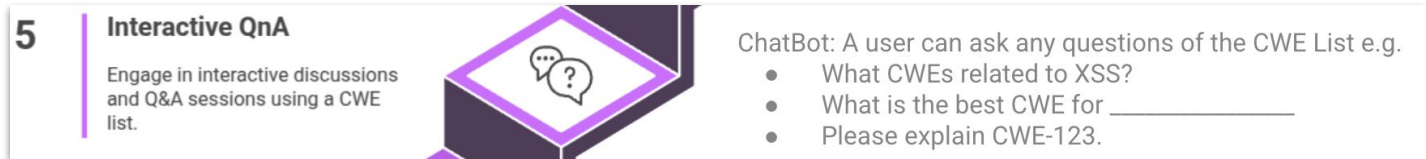
- Youtube video showing general CWE questions, and CWE assignment for several CVEs
- Some nuanced CWE assignments

This repo contains

1. The MITRE CWE specification stripped down to the minimum required for CWE mapping as the corpus for the CWE Expert.
2. Instructions on how to build a CWE Expert
3. Example prompts/questions to be used with the CWE Expert.

> ⓘ Note
>
> The code that creates the CWE Specification Corpus is cwe_json.ipynb.



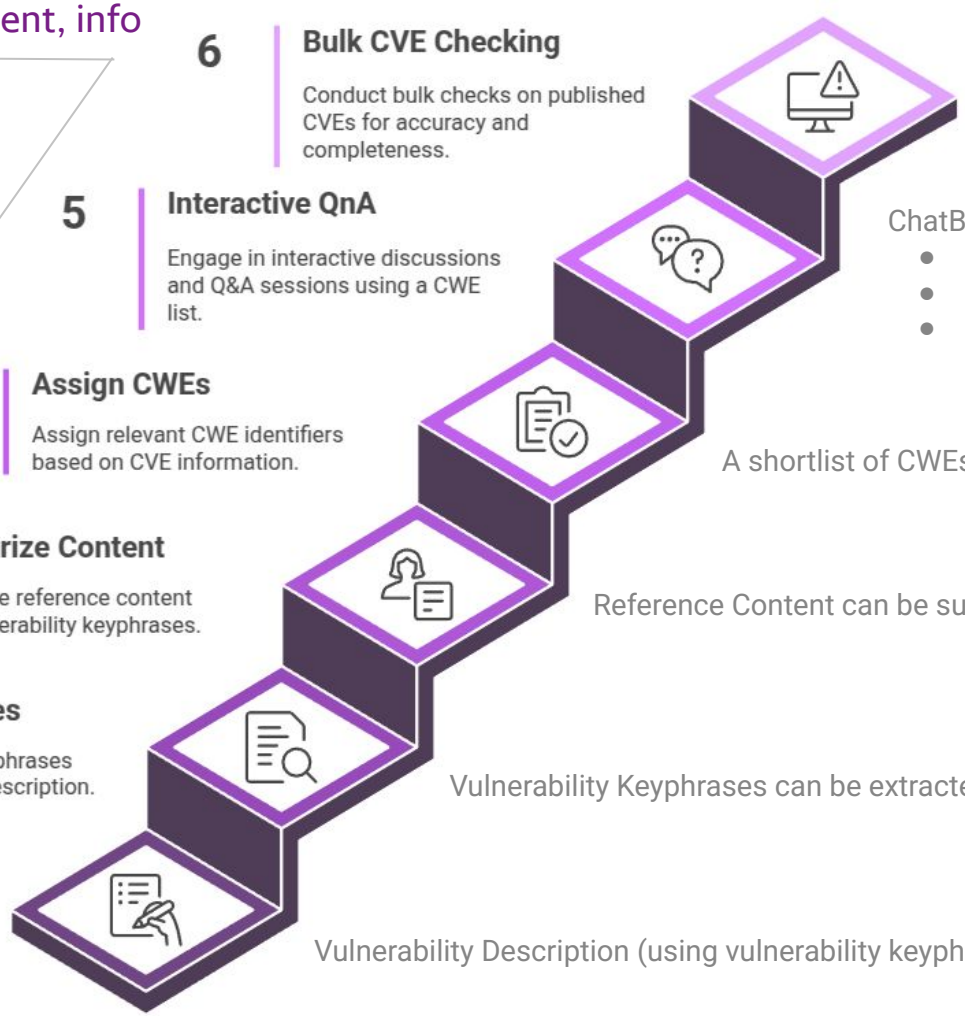


https://github.com/CyberSecAI/CWE-Expert/

LLM Capabilities in Vulnerability Root Cause Mapping (recap)

CWE assignment, info

Enrich and preserve CVE info



6 Bulk CVE Checking

Conduct bulk checks on published CVEs for accuracy and completeness.

Existing CVE-CWE mappings are checked. A shortlist of CWEs can be proposed for a CVE with rationale, confidence. Non-interactive.

5 Interactive QnA

Engage in interactive discussions and Q&A sessions using a CWE list.

ChatBot: A user can ask any questions of the CWE List e.g.

- What CWEs related to XSS?
- What is the best CWE for _____
- Please explain CWE-123.

4 Assign CWEs

Assign relevant CWE identifiers based on CVE information.

A shortlist of CWEs can be proposed for a CVE with rationale, confidence

3 Summarize Content

Summarize reference content using vulnerability keyphrases.

Reference Content can be summarized by vulnerability keyphrases (and preserved)

2 Extract Keyphrases

Identify and extract keyphrases from the vulnerability description.

Vulnerability Keyphrases can be extracted from an existing Vulnerability Description

1 Create Description

Develop a detailed vulnerability description from reference content.

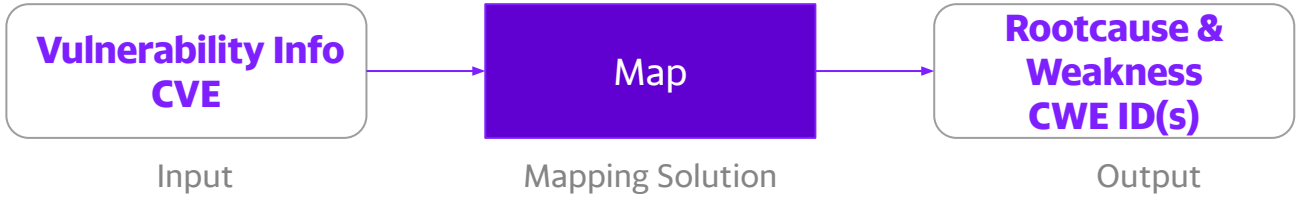
Vulnerability Description (using vulnerability keyphrases template) can be created from advisories or patches

The foundation is good CVE data! Complete, accurate and timely data.

Exploring LLM capabilities in vulnerability root cause mapping

Setup and methodology for testing capabilities for CWE Assignment

Bulk CVE Checking and Assignment of CWEs

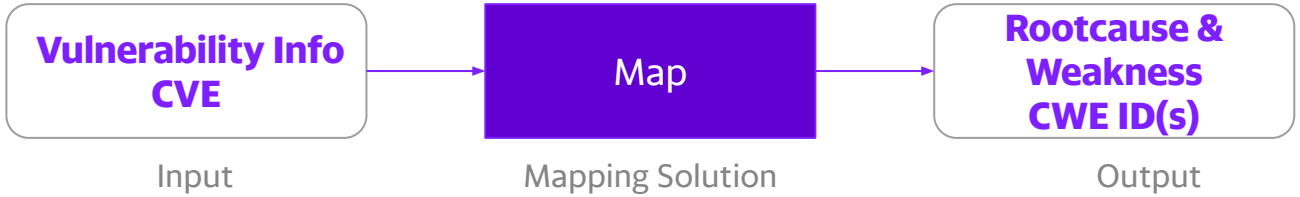


6 **Bulk CVE Checking**

Conduct bulk checks on published CVEs for accuracy and completeness.

Existing CVE-CWE mappings are checked. A shortlist of CWEs can be proposed for a CVE with rationale, confidence. Non-interactive.

Input and Output Data > Mapping Solution



A Mapping Solution is only as good as the INPUT data

Improve input data quality to ensure accurate mappings.

A Mapping Solution is only as good as the OUTPUT result, confidence, and supporting data

The Mapping Solution should assign CWE(s) with associated ranking, confidence and rationale.

Get as much useful (CVE, CWE) data as possible to inform the CWE assignment.

Optimize the system for CVE, CWEs

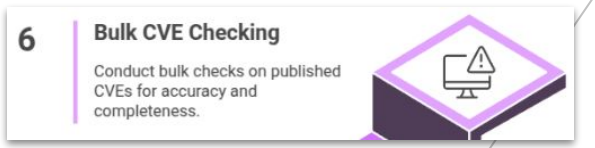
Utilize LLMs

Leverage Large Language Models for assistance in mapping.

“you can't train on "bad mappings" to learn how to do good mappings” Steve Christey Coley
There’s a LOT more information (context) in a CVE, and a CWE, that should be used to map CWE(s) to a CVE

Mapping Solution Data Layers

CWE assignment, info



Agent CVE ⟶ CWE Reports

LLM: Precision

Select CWE(s) (~~1)

Agent Input, Response

Critic Input, Response

Resolver Input, Response

Best Candidate CWEs from Retrievers with CWE context

Retrieval: Recall

Ranked Results (Top ~~10)

Sparse Results (15)

Dense Results (15)

Graph Results (15)

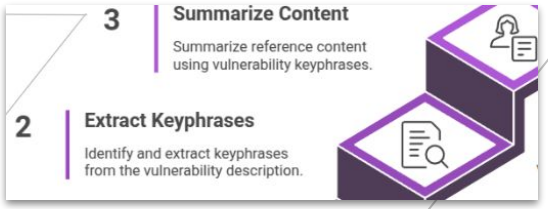
CWE Corpus

CWE List (~~1000)

Known Good Mapping Examples

Mapping Guidance

Enrich and preserve CVE info



Enriched CVE Data

CVE Description Key Phrases

CVE Links Content Summary

Similar CVE CWE consensus

Existing CVE Data

CVE Description

CVE Reference Links

All Published CVE Descriptions, CWEs

For all ~260K CVEs published before 2025:

https://github.com/CyberSecAI/cve_info Key Phrases
https://github.com/CyberSecAI/cve_info_refs References Text Summarized
https://github.com/CyberSecAI/cve_dedup Similar CVE and consensus CWEs

CVE 25 YEARS

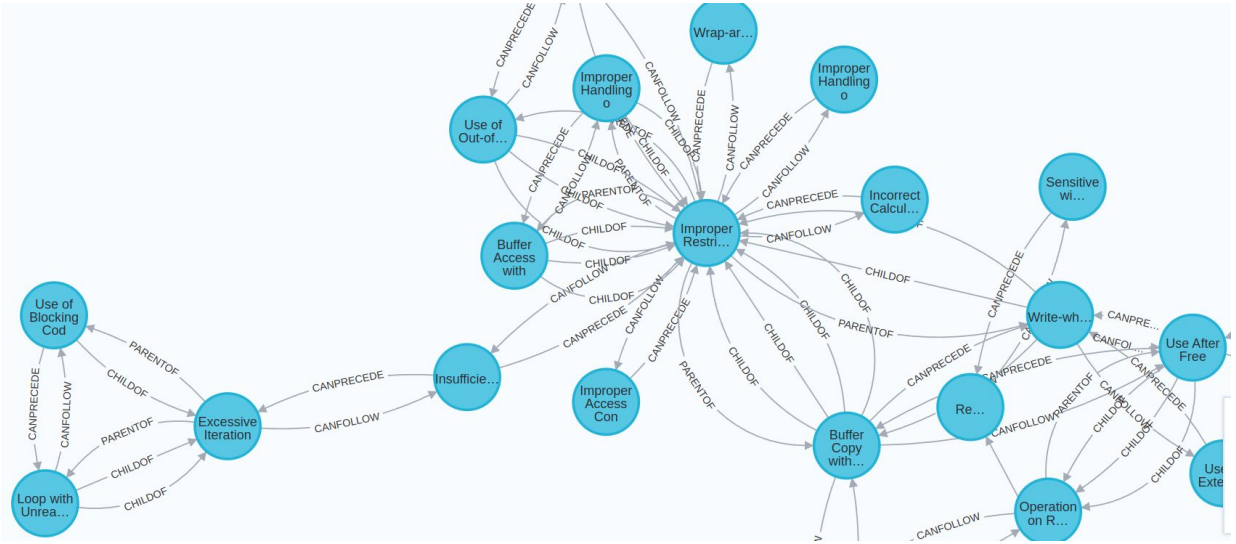
CWE Input Data

MITRE CWE list is a rich document containing detailed information on CWEs (~2800 pages)
The RelatedNatureEnumerations form a (view-dependent e.g. CWE-1003) graph of

- 1309 **ChildOf**/ParentOf
- 141 **CanPrecede**/CanFollow
- 13 **Requires**/RequiredBy

It contains ~3K Known-good CVE-CWE Mappings: Observed Example CVEs for various CWEs
2023 Top25 Mappings has ~6K Known-good CVE-CWE Mappings

MITRE provides CWE "Root Cause Mapping" Guidance

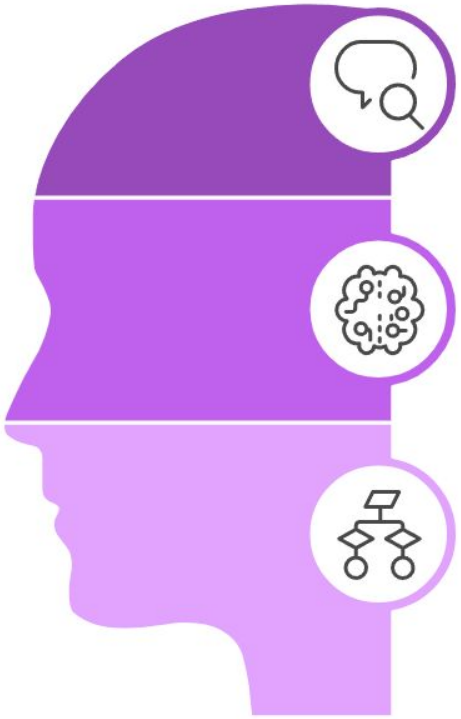


For a given CWE ID, the reciprocal RelatedNatureEnumerations relation is not defined in the CWE XML or JSON.
The existing RelatedNatureEnumerations can be used to build the reciprocal e.g. ChildOf to determine ParentOf.

Relevant CWE Info Related to the CVE Input Info

Retrievers get the CWE info (relevant to the input CVE info) to the LLM when it needs it (ala Retrieval Augmented Generation)

- For the input CVE, we want to go from ~1000 CWEs (from the CWE List) to ~~10 relevant CWEs so the LLM has the relevant extracts from the CWE List for those CWEs



Sparse Retriever

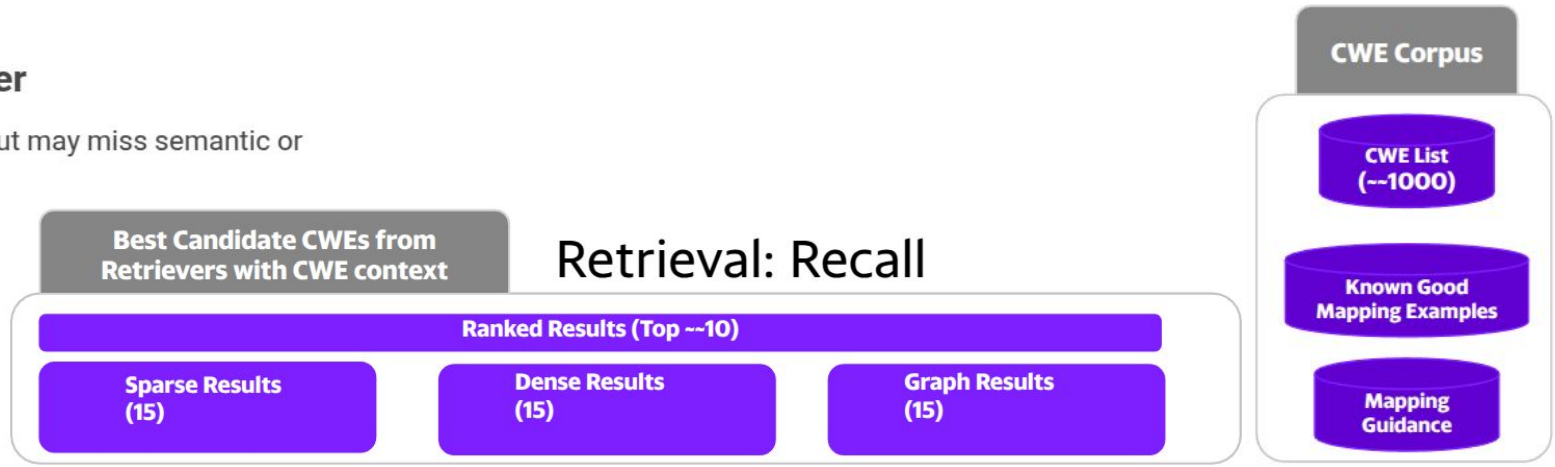
Excels at keyword matching but may miss semantic or structural relationships

Dense Retriever

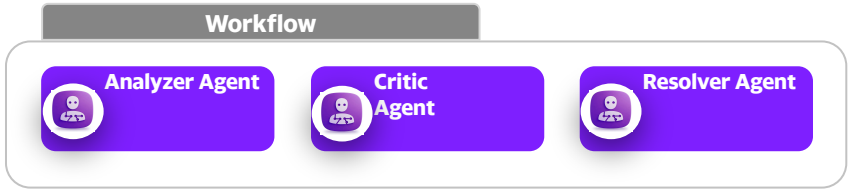
Captures semantic meaning but may miss exact keyword matches or structural relationships

Property Graph Retriever

Captures hierarchical context but may miss semantic or keyword relevance



Analysis LLM Workflow and Prompt



An Analyzer, Critic, Resolver pattern was evaluated initially.

- This added very little value (over asking the Analyzer to review its own work as part of the initial prompt) but was 3x time and $ cost.

Analyzer only was chosen

Analyzer Prompt (extract)

You are a security vulnerability analyzer specialized in Common Weakness Enumeration (CWE) classification. Your task is to analyze a given vulnerability description and assign the most appropriate CWE(s) based on root cause evidence, technical details, and established mapping best practices.

Your Analysis Must Include:

1. **Thorough Analysis of the Vulnerability Description**
   - Review all provided sections, especially:
     - **Vulnerability Description**
     - **Vulnerability Description Key Phrases** Focus on the rootcause or weakness phrases if present.
     - **CVE Reference Links Content Summary:** Focus on any rootcause and weakness mechanisms.
   - Clearly extract and reference supporting evidence from the text.
     - If evidence is insufficient, state that explicitly.
2. **Selecting the Right CWE Entry** based on **CWE for similar CVE Descriptions**, **Retriever Results**, and **Complete CWE Specifications**
   - **Rootcause**
     - The CWE should identify the issue that led to the vulnerability where this issue often has terms like “missing”, “improper”, “insufficient”, "insecure", "incorrect", "lack of", "does not properly", "fails to"
   - **CWE for similar CVE Descriptions** if present
     - **Primary CWE Match** is the best CWE Match for similar **Vulnerability Descriptions**
     - **Top CWEs** gives the top CWE Matches for similar **Vulnerability Descriptions**

https://cybersecai.github.io/prompt_engineering/prompt_engineering/

Prompts/Prompt Engineering makes a HUGE difference to the LLM behavior.

Exploring LLM capabilities in vulnerability root cause mapping

Case study examples

CVE-2022-40740 Retrieval Output (Input to LLM)

Vulnerability Description

Realtek GPON router has insufficient filtering for special characters. A remote attacker authenticated as an administrator can exploit this vulnerability to perform **command injection** attacks, to execute arbitrary system command, manipulate system or disrupt service.

Vulnerability Description Key Phrases

- **rootcause: insufficient input filtering**
- **weakness: command injection**
- **impact:** execute arbitrary system command
- **attacker:** remote authenticated administrator
- **product:** Realtek GPON router

CVE Reference Links Content Summary


```
{
  "Root cause of vulnerability": "The Realtek GPON router does not filter special characters in specific function fi
  "Weaknesses/vulnerabilities": [
    "Command Injection"
  ],
  "Impact of exploitation": "A remote attacker, after gaining administrator access, can exploit this vulnerability t
  "Attack vectors": "Remote command injection via a network.",
  "Required attacker capabilities/position": "The attacker must have administrator-level access to the router's inte
}
```


Retriever Results

Top Combined Results

Rank	CWE ID	Name	Abstraction	Usage	Retrievers	Individual Scores
1	77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	Class	Allowed-with-Review	alternate_terms	1.000
2	78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	Base	Allowed	alternate_terms	0.700
3	790	Improper Filtering of Special Elements	Class	Allowed-with-Review	sparse	0.291
4	791	Incomplete Filtering of Special Elements	Base	Allowed	sparse	0.271
5	138	Improper Neutralization of Special Elements	Class	Discouraged	sparse	0.225
6	121	Stack-based Buffer Overflow	Variant	Allowed	dense	0.515
7	184	Incomplete List of Disallowed Inputs	Base	Allowed	graph	0.002
8	20	Improper Input Validation	Class	Discouraged	sparse	0.207
9	89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	Base	Allowed	sparse	0.205
10	94	Improper Control of Generation of Code ('Code Injection')	Base	Allowed-with-Review	sparse	0.195

Complete CWE Specifications

CVE-2022-40740 LLM Output

Summary

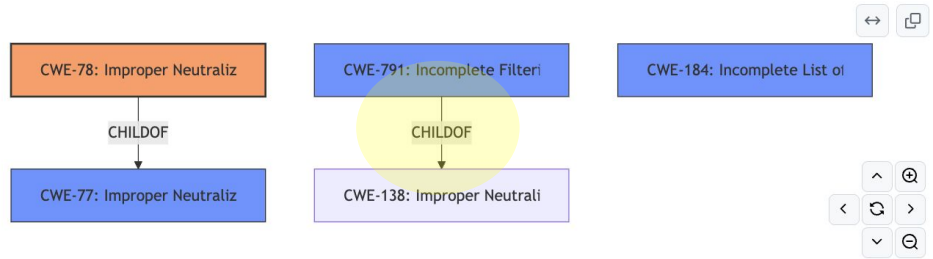
CWE ID	CWE Name	Confidence	CWE Abstraction Level	CWE Vulnerability Mapping Label	CWE-Vulnerability Mapping Notes
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	1.0	Base	Primary	Allowed
CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	0.7	Class	Secondary	Allowed-with-Review, consider if not OS Command Injection.
CWE-791	Incomplete Filtering of Special Elements	0.6	Base	Secondary	Allowed, consider if filtering is the root cause and is incomplete.
CWE-184	Incomplete List of Disallowed Inputs	0.5	Base	Secondary	Allowed, consider if a denylist is the root cause.

Evidence and Confidence

- **Confidence Score:** 1.0
- **Evidence Strength:** HIGH

Relationship Analysis

The primary CWE is CWE-78, which is a Base level weakness and a child of CWE-77. CWE-77 is a more general Class-level weakness. The vulnerability involves **command injection** attacks, which aligns with the descriptions of both CWE-78 and CWE-77. Because the **command injection** leads to arbitrary system commands, CWE-78 is a more specific fit. CWE-791 and CWE-184 are also related in that they describe **insufficient filtering**, which is the root cause here.



Vulnerability Chain

The vulnerability chain starts with **insufficient input filtering**, which allows an attacker to perform **command injection**. The **command injection** leads to arbitrary system command execution, system manipulation, or service disruption.

Summary of Analysis

The initial assessment strongly points towards **CWE-78 (Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection'))** as the primary CWE. The vulnerability description explicitly mentions **command injection** leading to arbitrary system command execution, and the CVE Reference Links Content Summary confirms that the root cause is the Realtek GPON router **does not filter special characters**.

The vulnerability description states: "Realtek GPON router has **insufficient filtering** for special characters. A remote attacker authenticated as an administrator can exploit this vulnerability to perform **command injection** attacks, to execute arbitrary system command, manipulate system or disrupt service."

The "CVE Reference Links Content Summary" states: "Root cause of vulnerability: The Realtek GPON router **does not filter special characters** in specific function fields." "Weaknesses/vulnerabilities: Command Injection"

CWE-78 is at the Base level of abstraction, which is the preferred level. The mapping guidance for CWE-78 states that it is "Allowed" and the rationale is "This CWE entry is at the Base level of abstraction, which is a preferred level of abstraction for mapping to the root causes of vulnerabilities."

CWE-77 was considered but not chosen as the primary because it is a more general class that applies to command injection in general, not specifically OS commands.

CWE-791 (Incomplete Filtering of Special Elements) was considered as a secondary CWE. This could be a contributing factor, as the filtering is present but insufficient.

CWE-184 (Incomplete List of Disallowed Inputs) was considered as a secondary CWE. If the **insufficient filtering** is specifically due to a denylist, then this CWE would be applicable.

Other CWEs considered but rejected:

- CWE-20 (Improper Input Validation): This is too general and discouraged.
- CWE-89 (Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')): This is specific to SQL injection and not applicable here.
- CWE-94 (Improper Control of Generation of Code ('Code Injection')): The vulnerability is not about generating code, but about executing commands.

1. Original assignment was CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') only
2. 2023 Top25 assigned **CWE-790: Improper Filtering of Special Elements**
   - **CWE-791** is a ChildOf CWE-790 (which is a ChildOf CWE-138)

CVE-2022-47717 Retrieval Output (Input to LLM)

Vulnerability Description

Last Yard 22.09.8-1 is vulnerable to Cross-origin resource sharing (CORS).

KeyPhrases

Vulnerability Description Key Phrases

- **product:** Last Yard
- **version:** 22.09.8-1
- **component:** Cross-origin resource sharing (CORS)

no rootcause or weakness KeyPhrases

Summary of CVE Reference Link Content

CVE Reference Links Content Summary

The provided content relates to CVE-2022-47717.

Root cause of vulnerability: The vulnerability stems from an overly permissive Cross-Origin Resource Sharing (CORS) configuration. Specifically, the `Access-Control-Allow-Origin` header is set to `*`, which allows any origin to access the resource.

Weaknesses/vulnerabilities present:

- **Insecure CORS Configuration:** The use of `*` in the `Access-Control-Allow-Origin` header opens up the application to cross-origin attacks.

Impact of exploitation:

- **Data theft:** A malicious website can make cross-origin requests to the vulnerable server and potentially steal sensitive data.
- **CSRF bypass:** The attacker can potentially bypass CSRF protections because the vulnerable server allows requests from any origin.

Attack vectors:

- **Cross-site requests:** A malicious web page can send a request to the vulnerable server using JavaScript.

Required attacker capabilities/position:

- The attacker only needs a web server to host the malicious HTML/Javascript code.
- The attacker needs the victim to visit the malicious page using a browser.

Retriever Results

Top Combined Results

Rank	CWE ID	Name	Abstraction	Usage	Retrievers	Individual Scores
1	923	Improper Restriction of Communication Channel to Intended Endpoints	Class	Allowed-with-Review	sparse	0.034
2	1385	Missing Origin Validation in WebSockets	Variant	Allowed	sparse	0.033
3	942	Permissive Cross-domain Policy with Untrusted Domains	Variant	Allowed	sparse	0.028
4	113	Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Request/Response Splitting')	Variant	Allowed	sparse	0.027
5	657	Violation of Secure Design Principles	Class	Discouraged	sparse	0.025
6	352	Cross-Site Request Forgery (CSRF)	Compound	Allowed	dense	0.465
7	1275	Sensitive Cookie with Improper SameSite Attribute	Variant	Allowed	graph	0.003
8	93	Improper Neutralization of CRLF Sequences ('CRLF Injection')	Base	Allowed	sparse	0.024
9	79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	Base	Allowed	sparse	0.023
10	918	Server-Side Request Forgery (SSRF)	Base	Allowed	sparse	0.022

Retrieval Results

Complete CWE Specifications

CVE-2022-47717 LLM Output 🧐

Summary

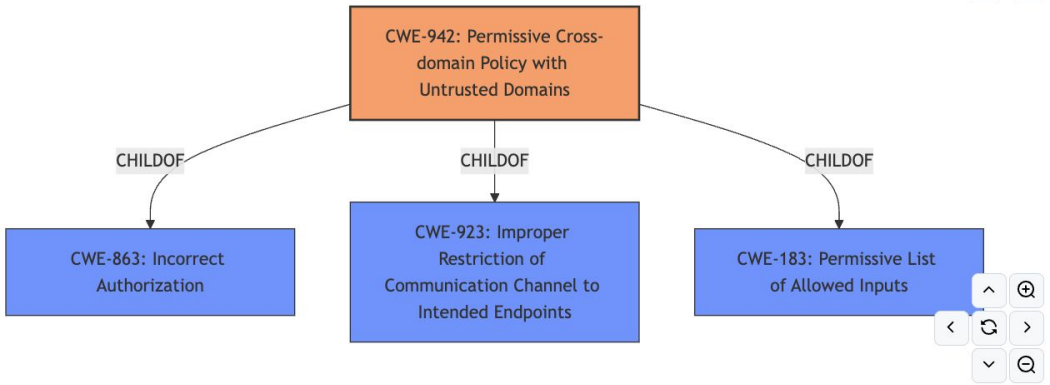
CWE ID	CWE Name	Confidence	CWE Abstraction Level	CWE Vulnerability Mapping Label	CWE-Vulnerability Mapping Notes
CWE-942	Permissive Cross-domain Policy with Untrusted Domains	0.9	Variant	Allowed	Primary CWE

Evidence and Confidence

- **Confidence Score:** 0.9
- **Evidence Strength:** HIGH

Relationship Analysis

The primary relationship is that CWE-942 is a variant-level CWE. It also has ChildOf relationships to CWE-863, CWE-923, and CWE-183. The relationships confirm that the selected CWE is the most specific and accurate representation of the vulnerability.



Vulnerability Chain

The chain of the vulnerability is that an overly permissive CORS policy (`Access-Control-Allow-Origin: *`) allows any origin to access the resource, leading to potential data theft and CSRF bypass.

Summary of Analysis

The analysis identified CWE-942 as the primary weakness. This assessment is based on the provided evidence, which explicitly states that the vulnerability stems from an overly permissive CORS configuration. The **root cause of the vulnerability** is that the `Access-Control-Allow-Origin` header is set to `*`, which allows any origin to access the resource.

The evidence:

> **Root cause of vulnerability:** The vulnerability stems from an overly permissive Cross-Origin Resource Sharing (CORS) configuration. Specifically, the `Access-Control-Allow-Origin` header is set to `*`, which allows any origin to access the resource.

CWE-942 (Permissive Cross-domain Policy with Untrusted Domains) closely matches the vulnerability description, especially the part "The product uses a cross-domain policy file that includes domains that should not be trusted." In this case, the policy effectively trusts all domains due to the wildcard.

Relevant CWE Information:

Enhanced Context (25 CWEs)

The following CWEs were identified as potentially relevant to this vulnerability:

- Original assignment was CWE-668: Exposure of Resource to Wrong Sphere
- Top25 2023 assigned **CWE-923**: Improper Restriction of Communication Channel to Intended Endpoints
  - CWE-942 is a ChildOf CWE-923
- CISA-ADP assigned **CWE-942** on 3/27/2025

https://nvd.nist.gov/vuln/detail/CVE-2022-47717#VulnChangeHistorySection

https://github.com/CyberSecAI/cwe_agent_assign_reports_2023/blob/main/CVE-2022-47717/CVE-2022-47717_combined_report.md

CVE-1999-0618 Low CVE Information

Description

The rexec service is running.

Hyperlink
https://www.cve.org/CVERecord?id=CVE-1999-0618
https://www.cve.org/CVERecord?id=CVE-1999-0618

Weakness Enumeration

CWE-ID	CWE Name
NVD-CWE-Other	Other

https://nvd.nist.gov/vuln/detail/CVE-1999-0618

CVE-1999-0618

Vulnerability Description

The rexec service is running.

Vulnerability Description Key Phrases

- **component:** rexec service

Final Resolution for CVE-1999-0618

Summary

CWE ID	CWE Name	CWE Abstraction Level	CWE Vulnerability Mapping Label	CWE-Vulnerability Mapping Notes
N/A				Insufficient information to assign a CWE.

Evidence and Confidence

- **Confidence Score:** 0.0
- **Evidence Strength:** LOW

Relationship Analysis

Given the lack of specific information about the vulnerability beyond the rexec service running, analyzing CWE relationships is not particularly fruitful. The provided relationships are likely the result of automated tools attempting to link the concept of "rexec" to various CWEs, but they lack the necessary context to be useful. Without more details on how the rexec service is being used or what specific misconfiguration exists, it's impossible to determine which, if any, of these relationships are relevant. The analysis correctly identifies that the given vulnerability description doesn't indicate a specific coding error or weakness, making it impossible to pinpoint a precise vulnerability.

https://github.com/CyberSecAI/cwe_assign_reports/blob/master/CVE-1999-0618/CVE-1999-0618_resolution.md

With low information, LLMs tend to infer from their training data but don't automatically indicate their confidence.
So we can be easily fooled by their fluency (hallucinations).

Evaluating a Grounded LLM Tool against traditional CWE Top 25 methodology

2019-2023 CWE Top 25 (manual mapping review)

- **From 2019–2023, the CWE team released five Top 25's based on analyzing public vulnerability data**
  - The team would spend significant time towards verifying RCM accuracy in the dataset
- **The goal was to identify those CVE Records for manual review based on automated keyword search and matching, e.g., CVE Records that:**
  - had descriptions that suggested inaccurate root cause mappings in the Record
  - mapped to abstract, high-level CWEs with less actionability
- **This resulted in thousands of 're-mappings' in the data, and a better data set to rank the CWE Top 25**
  - note: the 2024 CWE Top 25 leveraged the CNA community's expert perspective to review the dataset

Limitations of CWE manual processes

- **Resource intensive**
  - Time and staff
- **Uncertainty**
  - 3rd-party perspective lacks full context and relies on public disclosure information (sometimes quite vague)
- **Incomplete**
  - Impractical to analyze entire dataset manually

– but it did give us some good mappings to use for experimentation –

Evaluating a Grounded LLM Tool against traditional CWE Top 25 methodology

LLM

2023 Top25 LLM and Retriever Costs

Analysis of 7K CVEs from 2023 Top25:

- **~$15**
  - $2 for embeddings
  - $13 for analysis
- **27 hours (unattended)**
  - on a basic computer (with no optimization)
  - calling 3rd party LLM via API

$ cost is low
- it's based on input and output words/tokens count

Batch Time can be reduced by running multiple processes in parallel

The Reference Content Summarization is the most expensive in terms of words/tokens and money given the amount of content in reference links (GB).
- An Experimental LLM was used for this (as it was free to use during the experimental phase).

KeyPhrase Extraction

Extracting key phrases from CVEs using Gemini 1.5 Flash, costing around $10 and taking approximately 5 days.

Reference Content Summarization

Summarizing reference content uses Gemini 2.0 Flash Experimental with zero cost and a time of about 5 days.

260K CVEs to end 2024

https://github.com/CyberSecAI/cve_info Key Phrases
https://github.com/CyberSecAI/cve_info_refs
References Text Summarized

Retrieval

Retrieving shortlist of CWEs for 7K CVEs, costing about $2 total, and taking 4 seconds per CVE (Dense and Graph take significantly longer than Sparse).
OpenAI text-embedding-3-small for semantic search.

Analyst LLM

Assign CWEs for 7K CVEs using Gemini 2.0 Flash, costing approximately $13 total, and requiring 14 seconds per CVE (27 hours total)

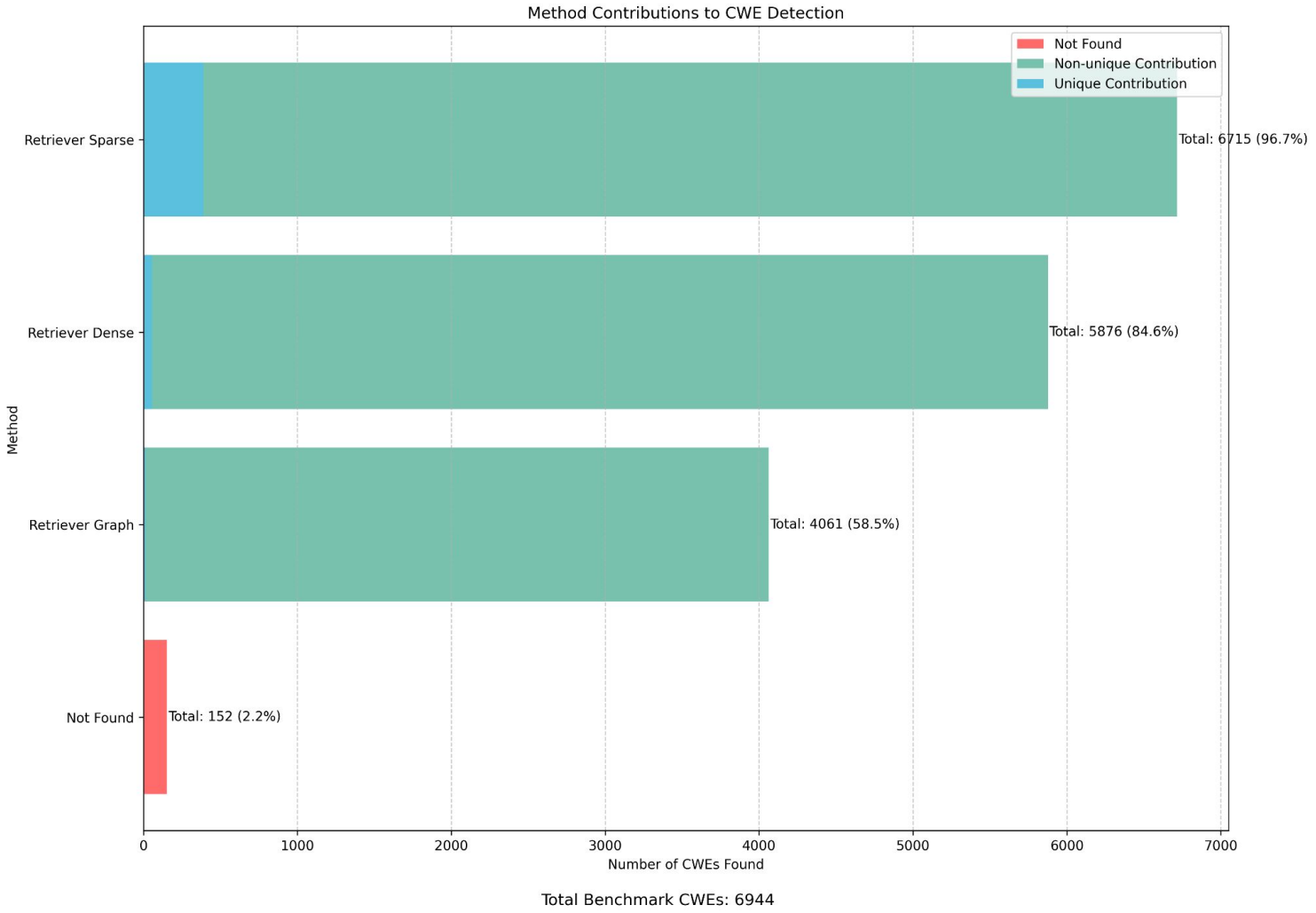
7K CVEs in 2023 Top25

Retriever Performance

2023 Top25 benchmark dataset

- **Total CVEs Analyzed:** 6944
- **CVEs with At Least One CWE Match:** 6656 (95.85% of 6944)
- **Unique CWEs:** 411

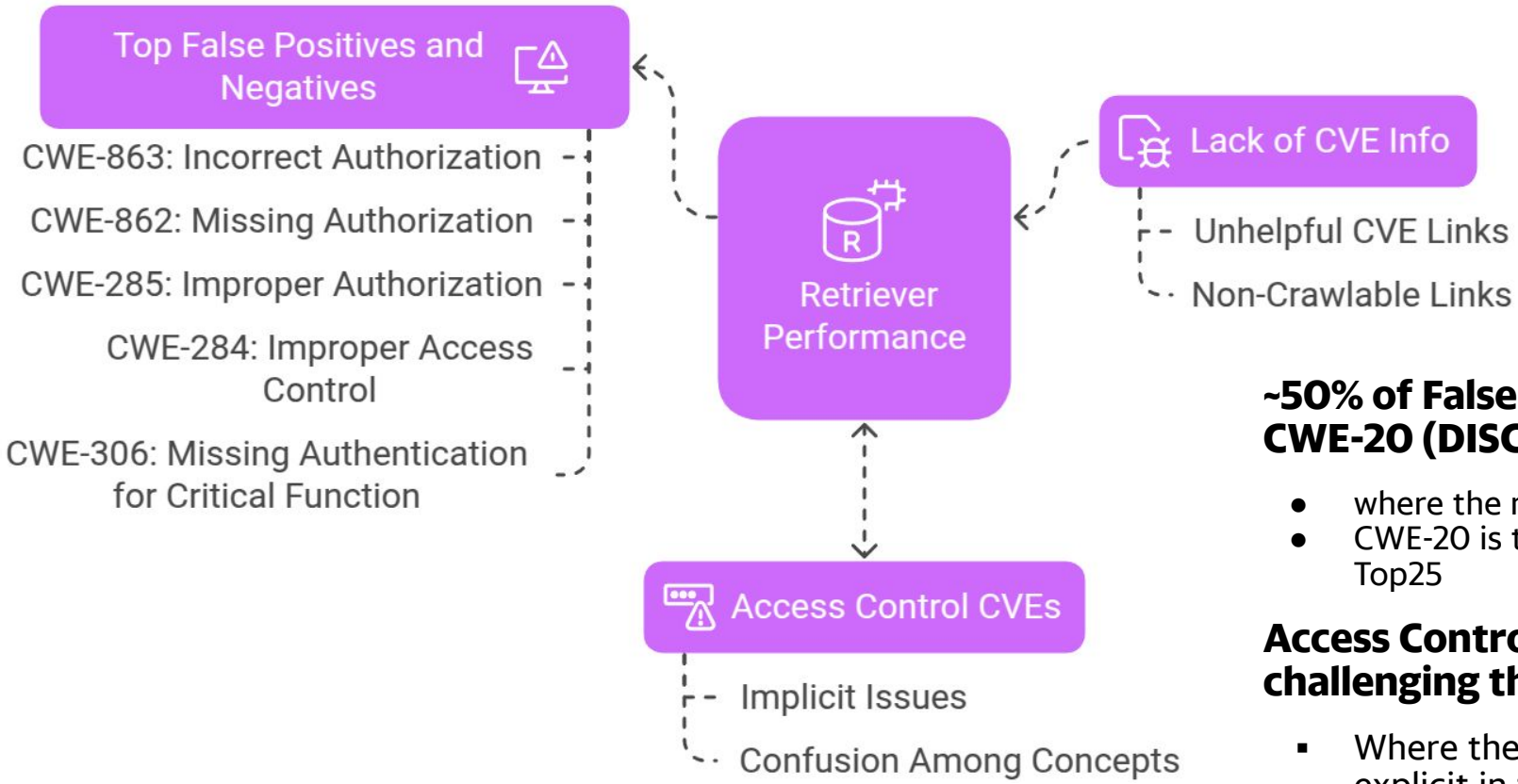
- **Overall Precision:** 0.10 (~ 1 in 10)
- **Overall Recall:** 0.96
- **Overall F1 Score:** 0.18



"Total CVEs Analyzed" is the subset when these are removed:

- CVE since Rejected
- CWE since Prohibited (2)
- No CWE assigned due to lack of Info

Retriever Performance



~50% of False Negatives (Misses) were for CWE-20 (DISCOURAGED, Class)

- where the retriever had more specific allowable CWEs
- CWE-20 is the most common (8%) CWE in the 2023 Top25

Access Control related CVEs were more challenging than others

- Where the access control issue was implicit not explicit in the CVE info
- People also struggle with Access Control vs Authentication vs Authorization
- This was the topic of a recent RCMWG meeting

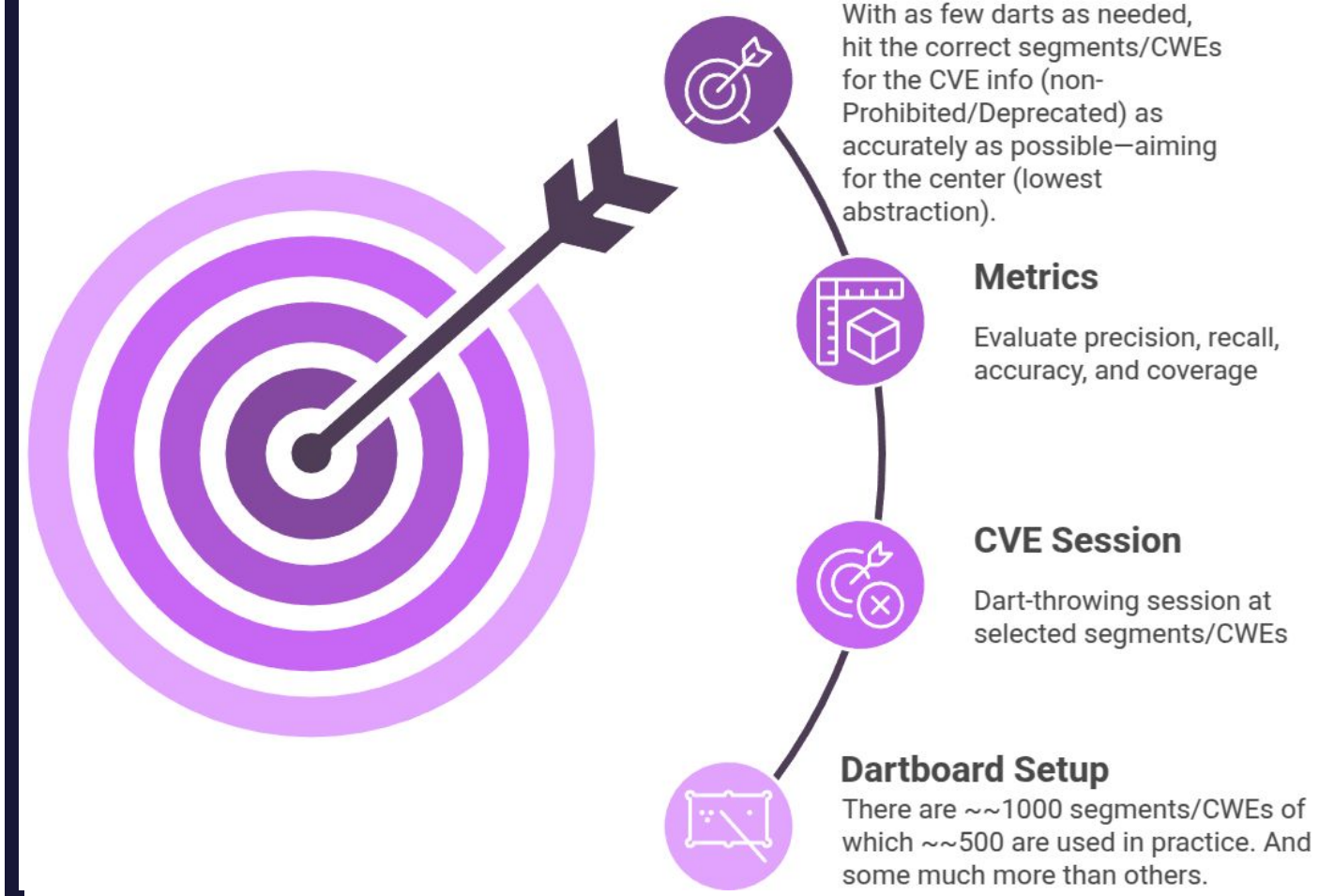
https://github.com/CVEProject/Root-Cause-Mapping-Working-Group/

NOTE: Retrievers are pre-LLM i.e. use traditional code methods to find the relevant CWEs - so fast and cheap relative to LLMs!

Evaluating a Grounded LLM Tool against traditional CWE Top 25 methodology

Comparison of results: CWE mappings vs. LLM inferences

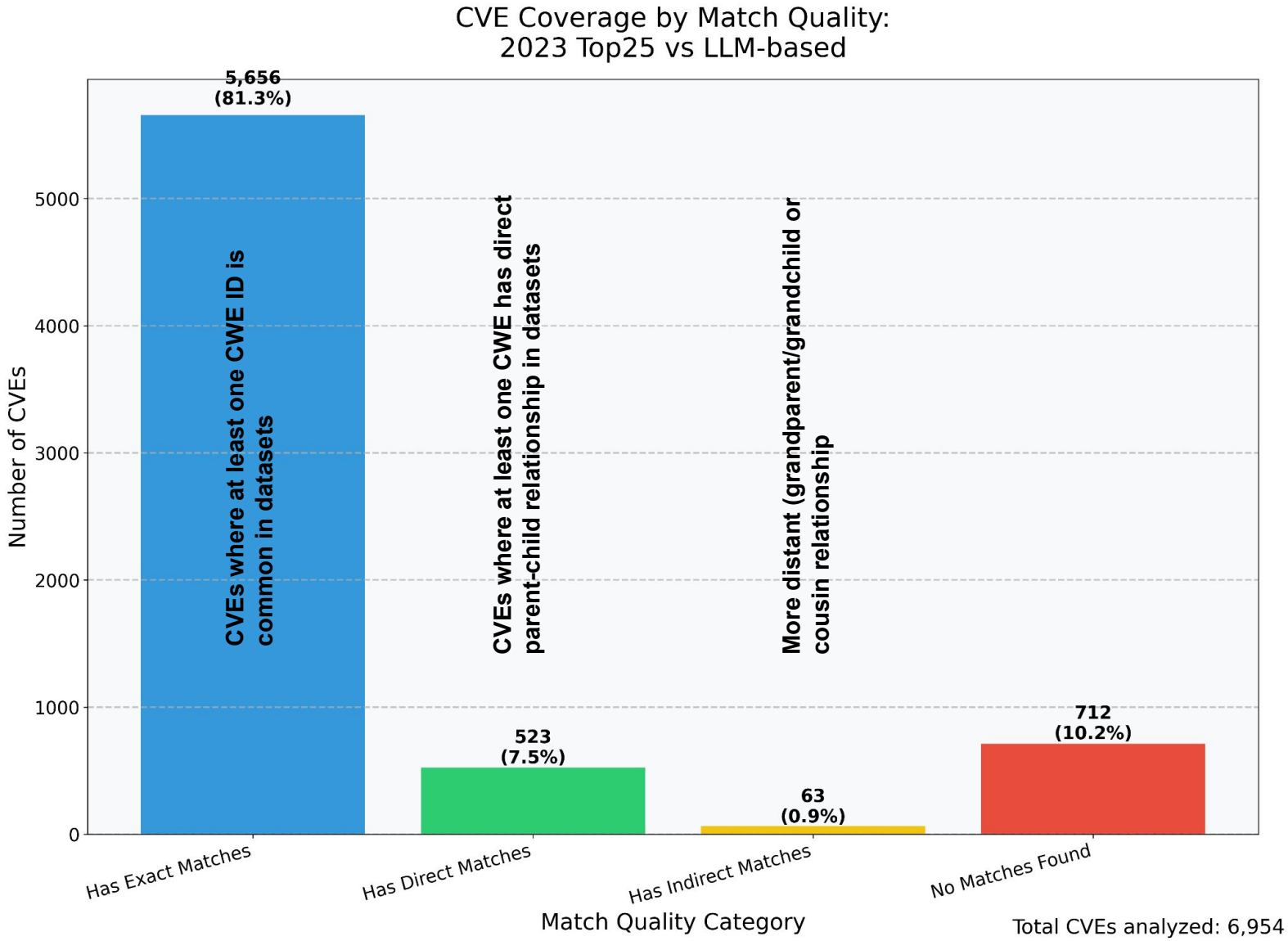
The Game we're Playing



- **The "dataset" is the problematic subset of the 2023 Top25. ~7K real live CVEs with varying data quality.**
- Most times only 1 dart per CVE is thrown by the expert, but sometimes 2, and rarely 3 or more.
- **The LLM CWE assignment solution is setup for Supervised / assistant mode per requirements. We want to provide a very short list of candidate CWEs i.e. we deliberately sacrifice Precision for Recall!**
- We are **measured on the match to the experts darts** (hit the same segment or not, and not whether the experts' darts hit the right segment or not).

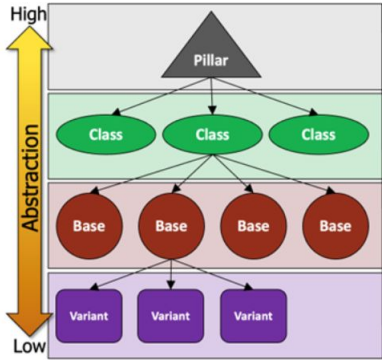
Coverage (CVE has at least one matching CWE)

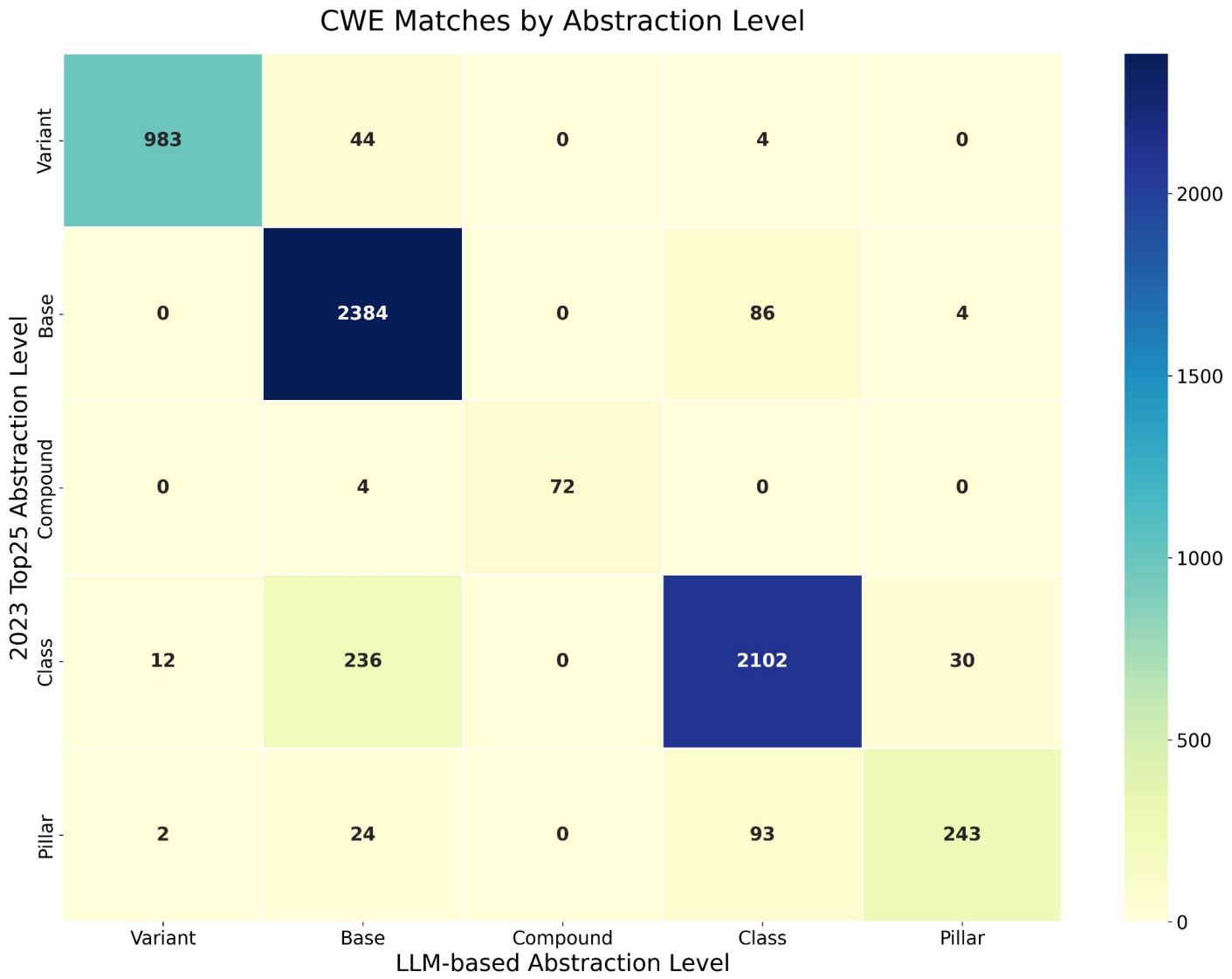
1. **High Agreement Rate (81%):** Over 5,600 CVEs have CWE matches between datasets, indicating strong consensus on vulnerability classification.
2. **Hierarchical Relationships**: About 8% of CVEs (523 + 63) don't have exact matches but have related CWEs through parent-child relationships in the CWE hierarchy.
3. **Small Discrepancy Rate (10%)**: Only about 10% of CVEs have completely unrelated or missing CWE assignments between datasets.
4. **Direct > Indirect**: When there isn't an exact match, direct parent-child relationships (523) are much more common than more distant relationships (63), suggesting that **when datasets disagree, they still tend to classify vulnerabilities in similar parts of the CWE hierarchy.**



Abstraction Heatmap (aka Confusion Matrix)

1. There is **strong agreement** between the 2023 Top25 and LLM-based approaches not just in which CWEs they assign, but also in the **level of abstraction** they choose.
2. The dominant pattern along the diagonal shows that **when both datasets identify the same type of vulnerability, they tend to describe it at the same level of abstraction**.
3. This **consistency is particularly strong at the Base and Class levels, which are the most practically useful levels** for vulnerability classification.





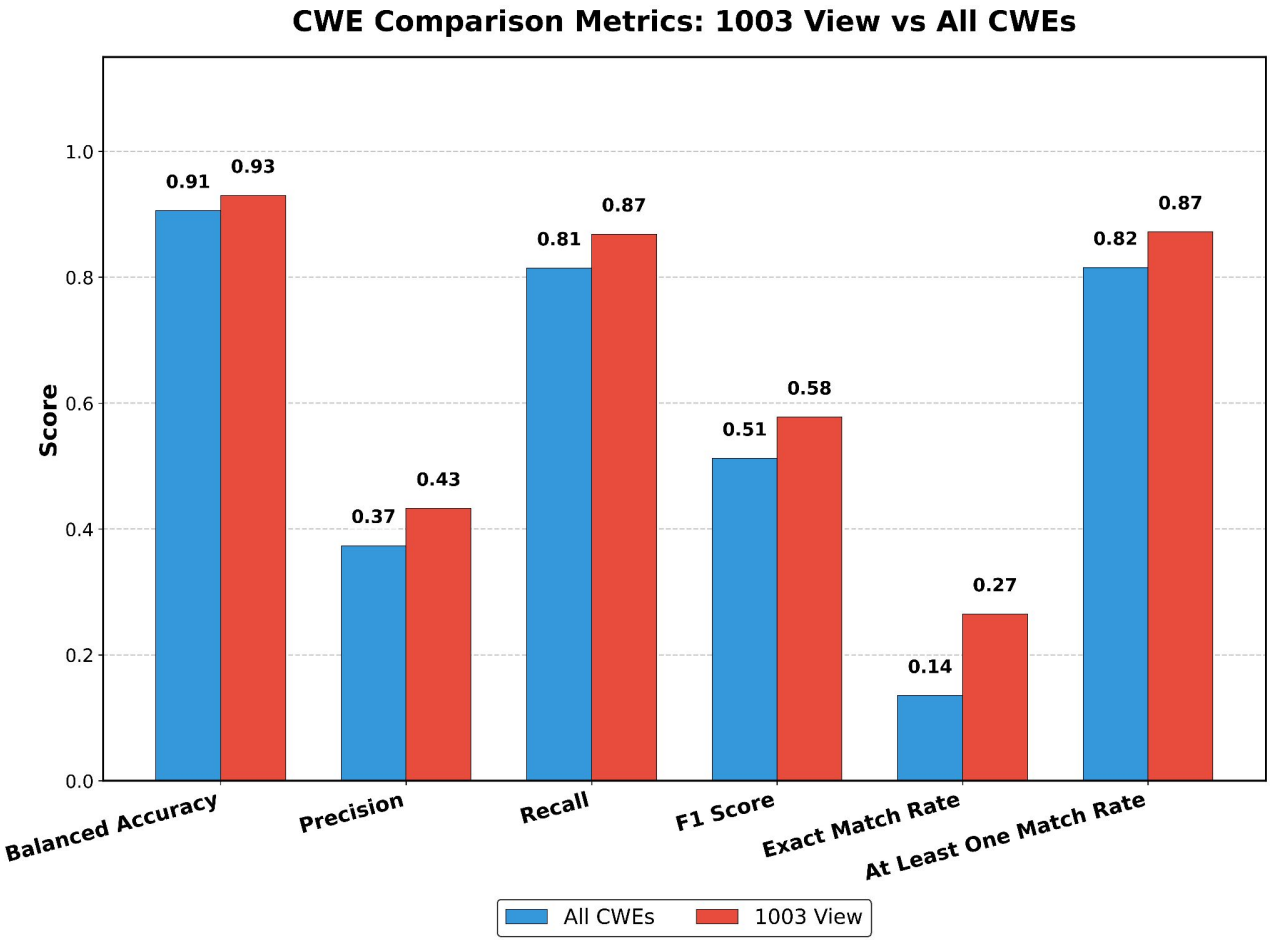
https://cwe.mitre.org/documents/cwe_usage/guidance.html

Comparison of results: CWE mappings vs. LLM inferences

We want to provide a very short list of candidate CWEs to the user i.e. we deliberately sacrifice Precision for Recall!

Key Metrics Calculated:

1. **Balanced Accuracy:** The average of Recall (Correctly identifying when a CWE applies) and Specificity (Correctly identifying when a CWE doesn't apply).
   - because Accuracy is > 99.9%, because we're picking ~3 out of 1000 CWEs
2. **Precision**: The fraction of Agent_Resolution CWEs that are in 2023 Top25
   - Measures how many of the LLM predictions are correct.
3. **Recall**: The fraction of 2023 Top25 that are in LLM results
   - Measures how many of the Benchmark CWEs the LLM correctly identified
4. **F1 Score**: The harmonic mean of precision and recall
   - Balanced measure of precision and recall
5. **Exact Match Rate**: The percentage of CVEs where 2023 Top25 and LLM are exactly the same
   - Perfect alignment between benchmark and LLM
6. **At Least One Match Rate**: The percentage of CVEs where there's at least one matching CWE
   - Indicates if the agent is at least partially correct

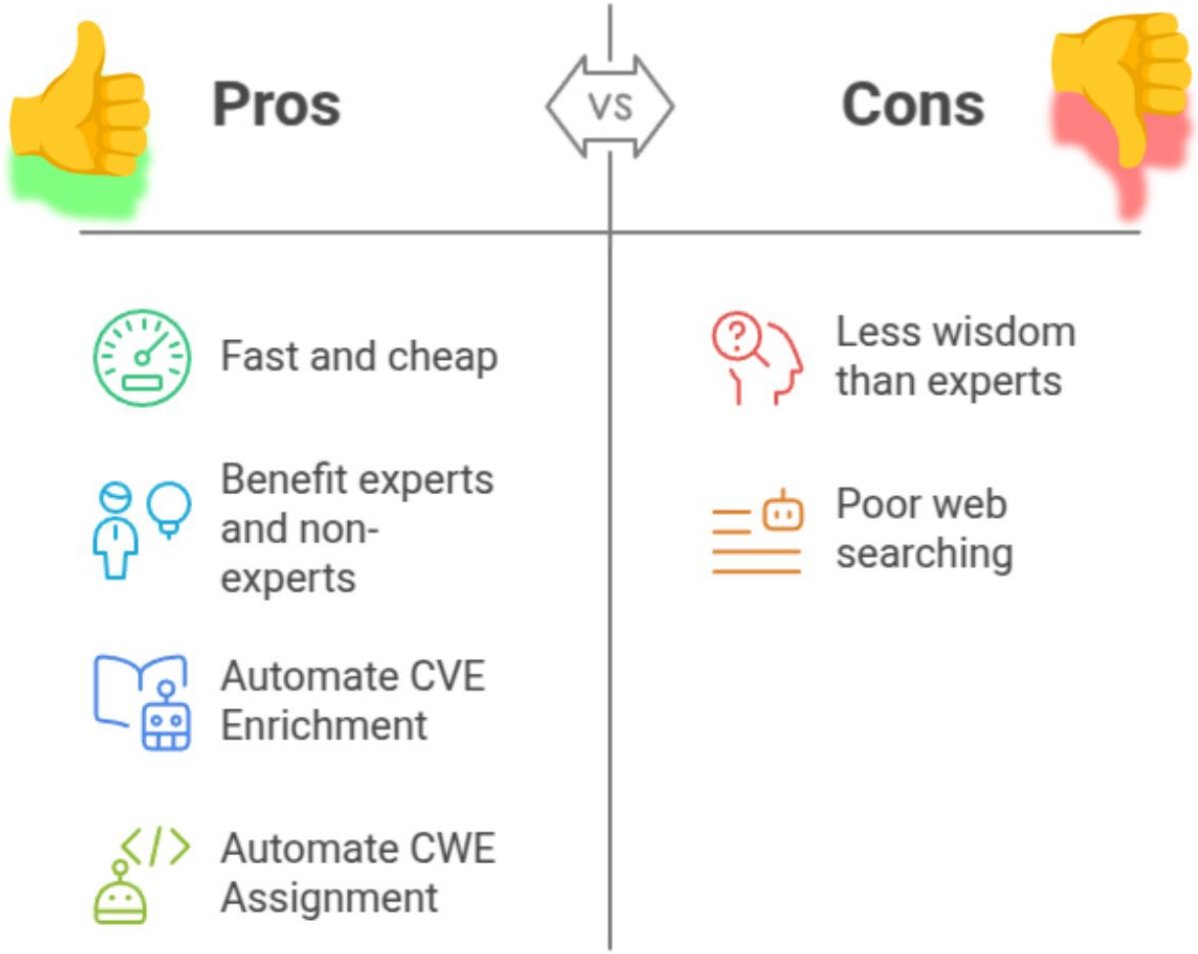


2023 Top25 Unique CWEs

- "All CWEs": 411 unique CWEs in the 2023 Top25 dataset
- 1003 View: 128 of 130 from CWE-1003 View where the results and 2023 Top25 CWEs are remapped to their lowest 1003 view parents (if any)

Very similar results and plot for 2022 Top25

Areas where LLMs complement or fall short

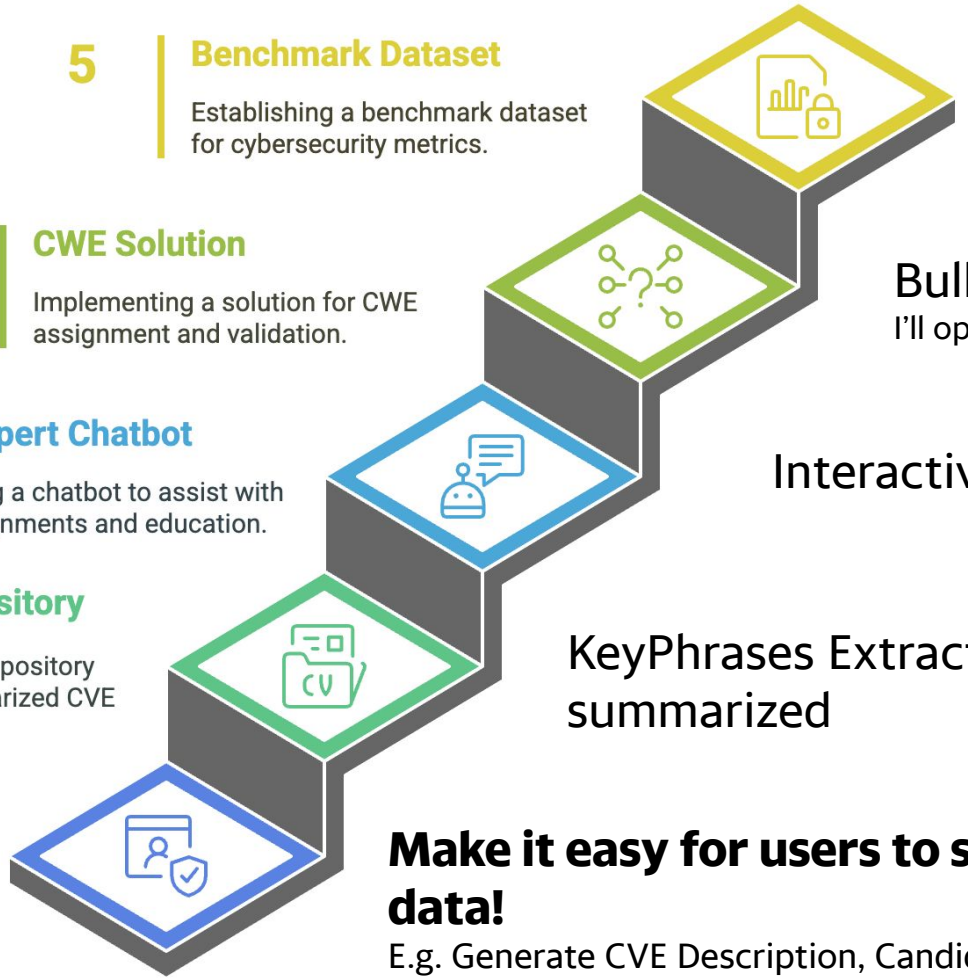


Wisdom: “having the power of discernment and judging properly as to what is true or right” https://www.dictionary.com/e/wisdom-vs-knowledge/

It is practical to analyze the entire CVE dataset

Practical Recommendations and Future Directions

Vision For The Future Of Root Cause Analysis In Cybersecurity



5 Benchmark Dataset

Establishing a benchmark dataset for cybersecurity metrics.

4 CWE Solution

Implementing a solution for CWE assignment and validation.

3 CWE Expert Chatbot

Developing a chatbot to assist with CWE assignments and education.

2 Central CVE Repository

Creating a centralized repository for enriched and summarized CVE information.

1 User Info Submission

Facilitating user-friendly submission of advisory and patch information using LLMs.

Bulk post-publish solution

I'll open-source my solution

Interactive Pre-publish solution

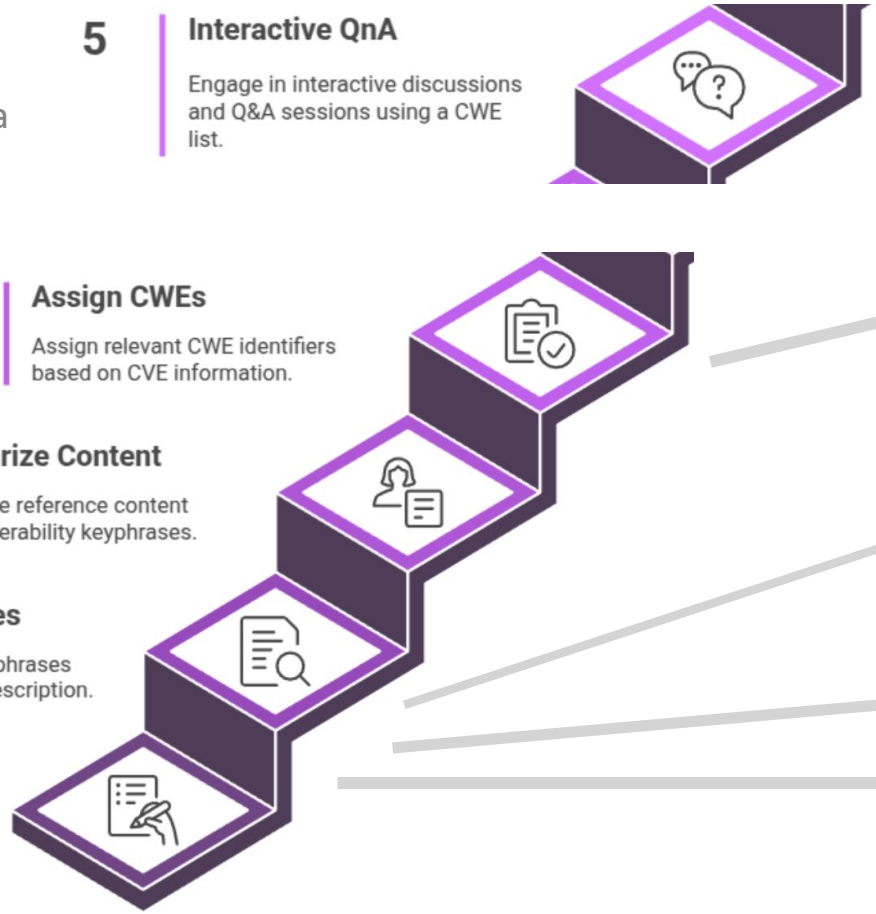
KeyPhrases Extracted + Reference Content preserved and summarized

Make it easy for users to submit complete, accurate and timely data!

E.g. Generate CVE Description, Candidate CWEs based on existing advisory, patch content.

User Info Submission

The Vulnogram Schema Source JSON can be auto-populated/created for a CVE.
Then opened in Vulnogram for editing.
No change required to Vulnogram.



Vulnogram CVE CVSS 4.0

NEW Open Download Post to CVE.org

Editor 4 Source Preview CVE Portal


```
{
  "dataType": "CVE_RECORD",
  "dataVersion": "5.1",
  "cveMetadata": {
    "cveId": "",
    "assignerOrgId": "00000000-0000-4000-9000-000000000000",
    "requesterUserId": "00000000-0000-4000-9000-000000000000",
    "serial": 1,
    "state": "PUBLISHED"
  },
  "containers": {
    "cna": {
      "providerMetadata": {
        "orgId": "00000000-0000-4000-9000-000000000000"
      },
      "problemTypes": [
        {
          "descriptions": [
            {
              "lang": "en",
              "cweId": "CWE-121",
              "description": "CWE-121 Stack-based Buffer Overflow",
              "type": "CWE"
            }
          ]
        }
      ],
      "impacts": [
        {
          "descriptions": [
            {
              "lang": "en",
              "value": "Denial-of-service: resource consumption"
            }
          ]
        }
      ],
      "affected": [
        {
          "versions": [
            {
              "status": "affected",
              "version": ""
            }
          ],
          "defaultStatus": "unaffected"
        }
      ],
      "descriptions": [
        {
          "lang": "en",
          "value": "A vulnerability in the ACME Discovery Protocol
              this vulnerability by sending a malicious Cisco Disco
              an attacker must be in the same broadcast domain as t
          "supportingMedia": [
            {
              "type": "text/html",
              "base64": false,
              "value": "<span style=\"background-color: rgb(255, 25
                  ACME Discovery Protocol messages. An attacker cou
                  Protocol is a Layer 2 protocol. To exploit this v
            }
          ]
        }
      ],
      "references": [
        {
          "url": "https://www.example.com/vuln/CVE-2020-12345"
        }
```


CVE 25 YEARS

CVE-CWE Mapping Leaderboard (Ideation)

Currently

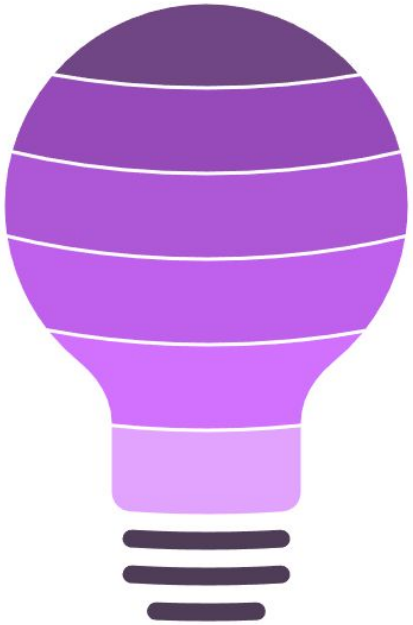
Many researchers build CVE-CWE mapping models, but there's no benchmark dataset or benefit to industry.

Benchmark Dataset

A dataset for standardizing mappings. Guidelines for evaluating solutions

Competitions

Events to improve mapping techniques



Open Source Solutions

Community-developed mapping solutions

Benefits of Participation

Gain recognition, learn new skills, and have fun in the process.

Competitions

- CVE-CWE mapping
- Reference Link Content Summarization
- Vulnerability KeyPhrase Extraction
- Vulnerability Description Creation

Rank	Model	Score	Organization	License
1	VulnMapPro v3.1	92.7	CyberSafe Labs	Apache 2.0
2	SecureMatchXL	90.4	OpenSec AI	MIT
3	CWEFinderNet	89.8	Massachusetts Institute of Technology (MIT)	GPL-3.0
4	ThreatLinker AI	87.2	RedShield AI	Apache 2.0
5	CVE2CWE-Transformer	84.3	Stanford University	BSD 3-Clause
6	VulnMatch	83.9	OpenThreat Systems	MIT
7	AutoCWE-Tagger	81.5	Carnegie Mellon University	Apache 2.0
8	CyberMatch GPT	78.6	SecureAI Alliance	GPL-3.0
9	CWEAligner	75.4	University of Cambridge	BSD 2-Clause
10	BasicMapBot	70.1	VulnScan Open Group	MIT

Email cwe@mitre.org if interested

with subject "CVE-CWE Mapping Leaderboard"

Example Leaderboard https://huggingface.co/spaces/lmarena-ai/chatbot-arena-leaderboard

Conclusion

- **The foundation of good vulnerability management and reducing cybersecurity risk is Complete, Accurate, and Timely CVE data!**
  - LLMs can make it easier to create this
  - LLMs and People rely on this for Root Cause Mapping and CWE assignment
- **LLMs are good enough and cheap enough to be used for Root Cause Mapping and CVE Enrichment in an interactive or bulk (e.g., all CVEs) supervised manner**
- **Community engagement will continue to drive advancements – model refinement, RCM WG discussions,**
- **ChatBot WIP**
  - for the CWE website, and possibly in the future…
  - incorporated into CVE Record generator clients

The mission of the CVE Program is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.

There is one CVE Record for each vulnerability in the catalog. The vulnerabilities are discovered then assigned and published by organizations from around the world that have partnered with the CVE Program. Partners publish CVE Records to communicate consistent descriptions of vulnerabilities. Information technology and cybersecurity professionals use CVE Records to ensure they are discussing the same issue, and to coordinate their efforts to prioritize and address the vulnerabilities.

Learn more www.cve.org