
OWASP GLOBAL AppSec
VIE'26
NNA
JUN
25-26
25
years
of open source security

OWASP GLOBAL AppSec

VIENNA '26 JUN 25-26

25 years of open source security

OWASP CYCLONEDX SUNSHINE: SEE CYCLONEDX SBOMS COME TO LIFE AND CHAT WITH THEM

LUCA CAPACCI

$ whoami

Luca Capacci

- Staff Application Security Engineer @ Ivanti
- Maintainer @ OWASP CycloneDX
- 10+ years in the cybersecurity field
- *Opinions are my own and do not reflect those of my employer or any affiliated organizations*

AGENDA

1. INTRODUCTION:

- What is an SBOM and why it matters
- What is the OWASP CycloneDX project
- Brief introduction to the CycloneDX JSON/XML format
- The missing piece: an actionable visualization tool for CycloneDX files

2. OWASP CYCLONEDX SUNSHINE

- Sunshine to the rescue: visualize a CycloneDX file in an interactive and human-friendly way
  - Sunburst chart with dependencies, licenses and vulnerabilities
  - Table with dependencies, licenses and vulnerabilities
- Data enrichment with EPSS and CISA KEV
- Chart refocus to see only dependencies and vulnerabilities of a single component (with live demo)

3. WHAT'S NEW

- Advanced filters
- Identify and analyse n-tier dependencies within the SBOM
- "Query my SBOM" functionality AKA how to literally query your SBOM in SQL
- "Chat with my SBOM" functionality AKA how to get information from your SBOM in a convenient conversational way

4. WHAT'S NEXT

- Additional features?

The acronym **SBOM** stands for **Software Bill of Materials**, and it comes from the more general concept of a **BOM**, which stands for **Bill of Materials**.

Origin of "BOM" (Since around the 1960s)

- In **manufacturing**, a **Bill of Materials** is a comprehensive list of raw materials, parts, and components needed to build a physical product (like a car or a computer).

- It helps manufacturers understand **what's inside** a product and manage sourcing, costs, and quality.

SBOM (Since around the early 2000s)

An **SBOM** is a formal, machine-readable inventory of all software components (libraries, packages, frameworks) and related dependencies included in a software product.

The acronym **SBOM** stands for **Software Bill of Materials**, and it comes from the more general concept of a **BOM**, which stands for **Bill of Materials**.

Origin of "BOM" (Since around the 1960s)

- In **manufacturing**, a **Bill of Materials** is a comprehensive list of raw materials, parts, and components needed to build a physical product (like a car or a computer).

- It helps manufacturers understand **what's inside** a product and manage sourcing, costs, and quality.

SBOM (Since around the early 2000s)

An **SBOM** is a formal, machine-readable inventory of all software components (libraries, packages, frameworks) and related dependencies included in a software product.

OWASP CYCLONEDX SUNSHINE: SEE CYCLONEDX SBOMS COME TO LIFE AND CHAT WITH THEM

Post-Build Generation

Analyzing binaries and executables to generate SBOMs after the software has been built.

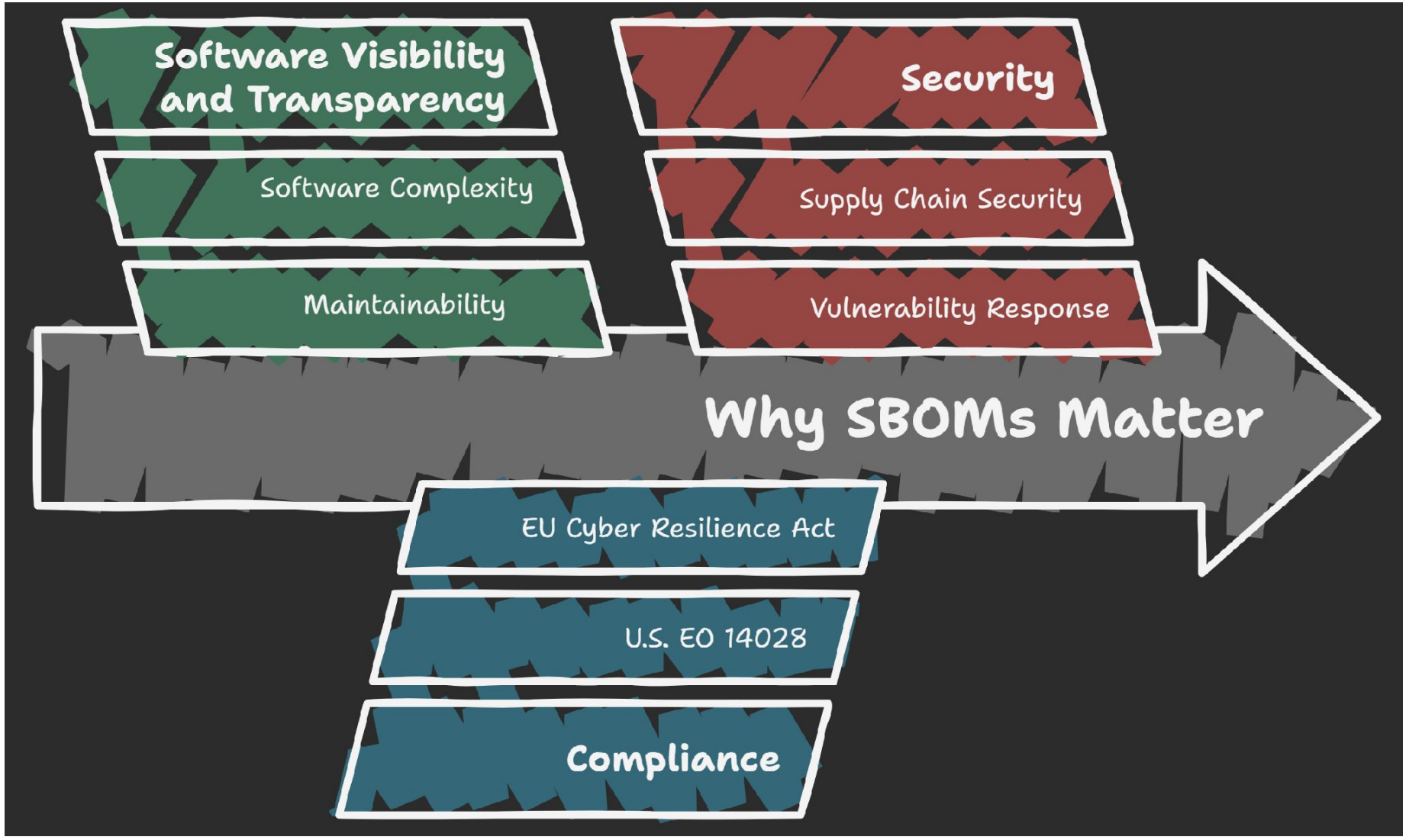
2

1

Build-Time Generation

Creating SBOMs as part of the automated build pipeline, capturing all components included in the final artifact.

OWASP CYCLONEDX SUNSHINE:
SEE CYCLONEDX SBOMS COME TO
LIFE AND CHAT WITH THEM



U.S. EO 14028 (2021)
It mandates that software vendors **"shall provide a Software Bill of Materials (SBOM)"** as part of federal procurement requirements.

EU Cyber Resilience Act (2024)
Software and connected device manufacturers **must maintain a technical documentation file that includes "information about components used"**.

It means that manufacturers are expected to document third-party components, their vulnerabilities, and secure lifecycle practices, if they intend to sell such type of products within the European Union.

OWASP CYCLONEDX SUNSHINE: SEE CYCLONEDX SBOMS COME TO LIFE AND CHAT WITH THEM

CycloneDX

The **OWASP CycloneDX Project** is an open-source standard and ecosystem focused on **creating, consuming, and automating Software Bill of Materials (SBOMs)** to support cybersecurity, software supply chain risk management, and compliance.

It's part of the broader OWASP (Open Worldwide Application Security Project) community, and it's specifically designed to help **developers, security teams, and organizations** understand and secure what's inside their software.

https://cyclonedx.org/
https://cyclonedx.org/tool-center/

OWASP CYCLONEDX SUNSHINE:
SEE CYCLONEDX SBOMS COME TO
LIFE AND CHAT WITH THEM

Why CycloneDX

Lightweight

Fast generation and processing

Security-first design

Built with vulnerability tracking

Broad tooling support

Compatible with various tools

Continuous monitoring

Enables real-time SBOM updates

Ecosystem integration

Works well with CI/CD tools and security scanners

OWASP CYCLONEDX SUNSHINE: SEE CYCLONEDX SBOMS COME TO LIFE AND CHAT WITH THEM

Component name

log4j

Supplier or source

Apache Software Foundation

Licensing info

Apache License v2

VEX-related information

log4j-core@2.14.1 is affected by CVE-2021-44228

Version

2.14.1

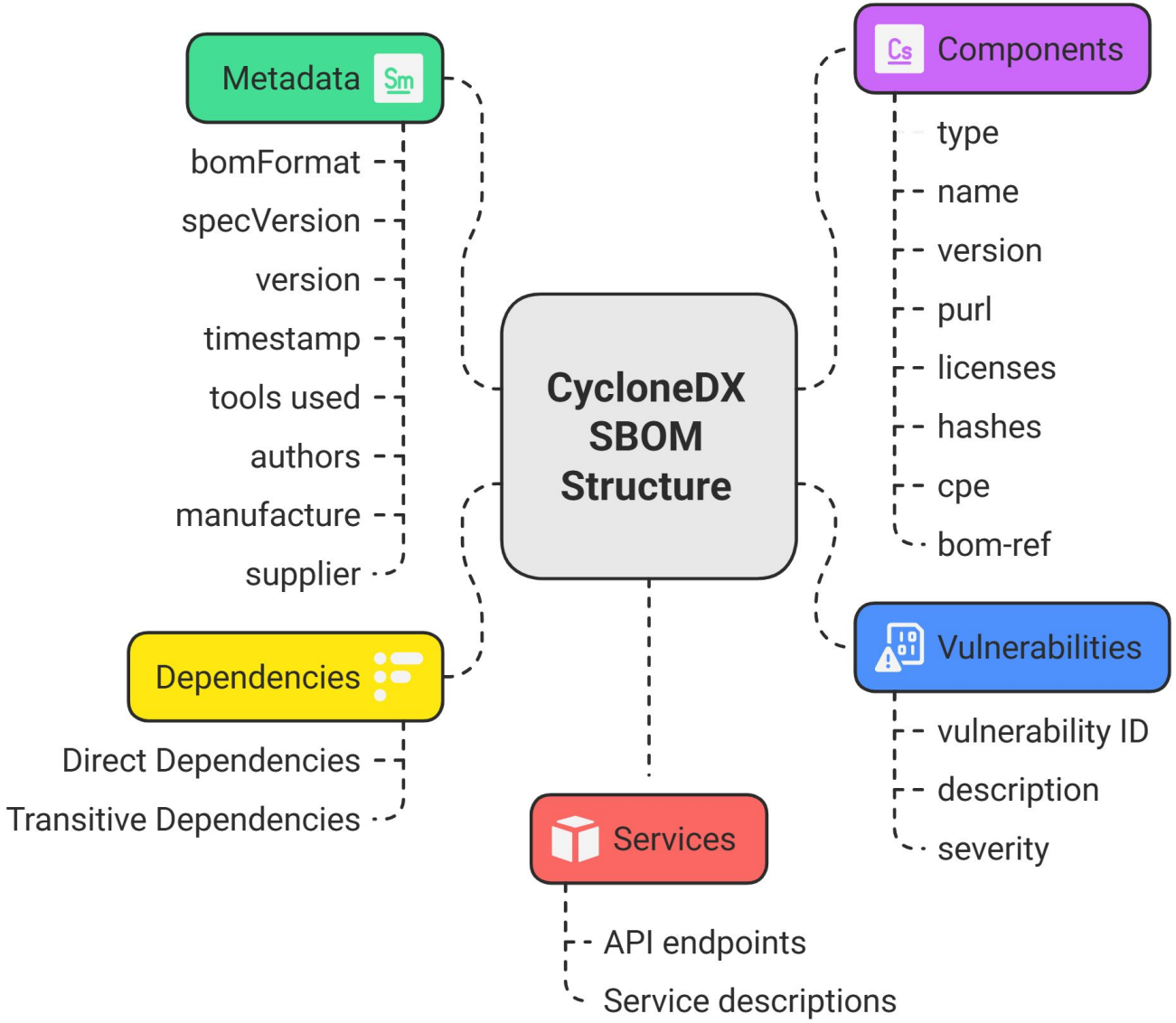
Dependency relationships

log4j-core depends on log4j-api

Unique identifiers

purl:pkg:maven/org.apache.logging.log4j/log4j-core@2.14.1

OWASP CYCLONEDX SUNSHINE: SEE CYCLONEDX SBOMS COME TO LIFE AND CHAT WITH THEM




```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.5",
  "version": 1,
  "metadata": {
    "timestamp": "2025-05-03T12:00:00Z",
    "tools": [
      {
        "vendor": "CycloneDX",
        "name": "cyclonedx-cli",
        "version": "0.24.0"
      }
    ],
    "component": {
      "type": "application",
      "name": "my-app",
      "version": "1.0.0"
    }
  },
  "components": [
    {
      "type": "library",
      "name": "log4j-core",
      "version": "2.14.1",
      "purl": "pkg:maven/org.apache.logging.log4j/log4j-core@2.14.1",
      "licenses": [
        {
          "license": {
            "id": "Apache-2.0"
          }
        }
      ],
      "hashes": [
        {
          "alg": "SHA-256",
          "content": "abc123..."
        }
      ]
    }
  ],
  "dependencies": [
    {
      "ref": "pkg:maven/org.apache.logging.log4j/log4j-core@2.14.1",
      "dependsOn": []
    }
  ]
}
```

Example of VEX-related information:

```
        "bom-ref" : "product-ABC"
      }
    },
    "vulnerabilities": [
      {
        "id": "CVE-2021-44228",
        "source": {
          "name": "NVD",
          "url": "https://nvd.nist.gov/vuln/detail/CVE-2021-44228"
        },
        "analysis": {
          "state": "not_affected",
          "justification": "code_not_present",
          "response": ["will_not_fix"],
          "detail": "This version of Product ABC is not affected by the vulnerability. Class with vulnerable code was removed
        },
        "affects": [
          {
            "ref": "product-ABC"
```

Ever looked at a CycloneDX file and thought… *there's gotta be a better way to read this?*

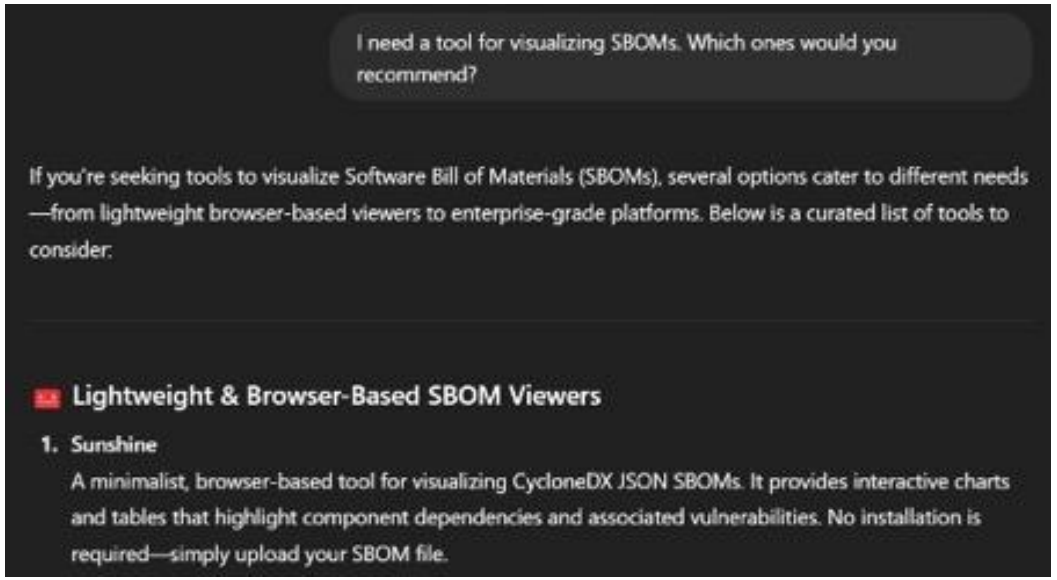
OWASP CYCLONEDX SUNSHINE: SEE CYCLONEDX SBOMS COME TO LIFE AND CHAT WITH THEM

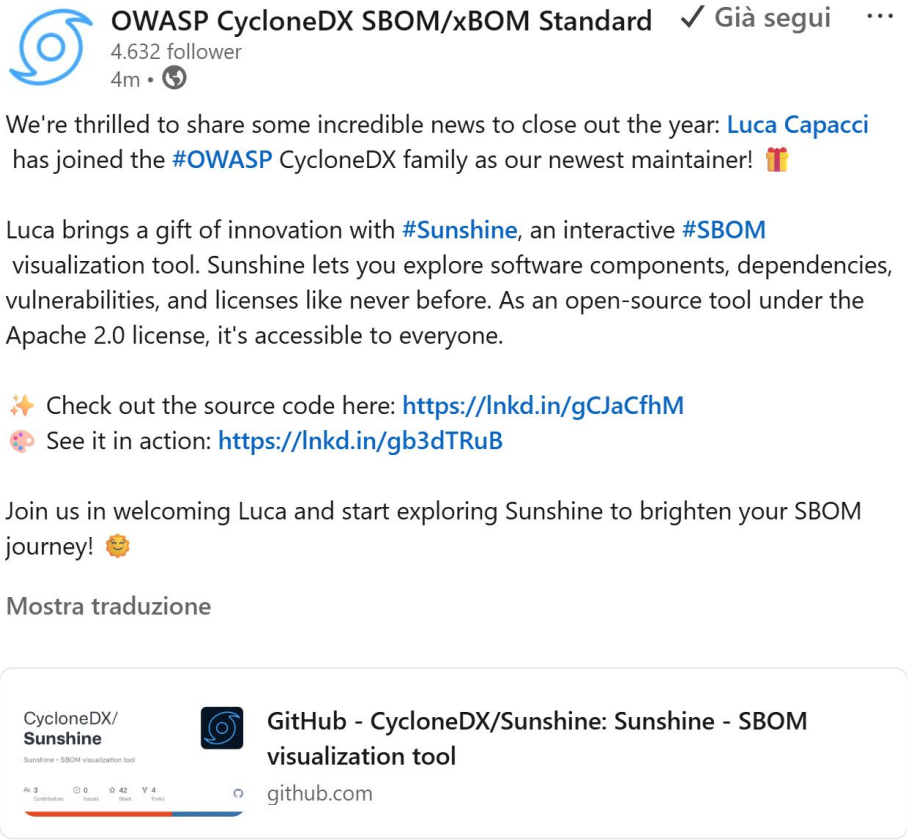
Sunshine to the rescue!

Sunshine makes our SBOM interactive and easier to read FOR HUMANS.

GitHub Repository

https://github.com/CycloneDX/Sunshine/





CLI Version

```
# Setup
git clone https://github.com/CycloneDX/Sunshine.git
cd Sunshine
python3 -m venv venv
source venv/bin/activate
pip3 install -r requirements.txt
```

```
# Basic usage without data enrichment
python sunshine.py -i your-input.json -o your-output.html

# Basic usage with EPSS and CISA KEV data enrichment
python sunshine.py -i your-input.json -o your-output.html -e
```

```
# Advanced options
-e, --enrich                          enrich CVEs with EPSS and CISA KEV
-k, --only-in-cisa-kev                show only vulnerabilities in CISA KEV
-cs, --only-critical-severity         show only vulnerabilities with critical severity
-hs, --only-high-severity-or-above    show only vulnerabilities with high severity or above
-ms, --only-medium-severity-or-above  show only vulnerabilities with medium severity or above
-ls, --only-low-severity-or-above     show only vulnerabilities with low severity or above
-c, --min-cvss MIN_CVSS               show only vulnerabilities with score equal to or greater than the selected
                    value, which can be in rage 0.0-10.0
-p, --min-epss MIN_EPSS               show only vulnerabilities with EPSS equal to or greater than the selected
                    value, which can be in rage 0.00-1.00
-n, --no-segment-limit                prevent the automatic conversion of charts with many segments
                                       into still images
-nl, --no-logo                        prevent the display of the banner logo on startup
```

Web Version - online

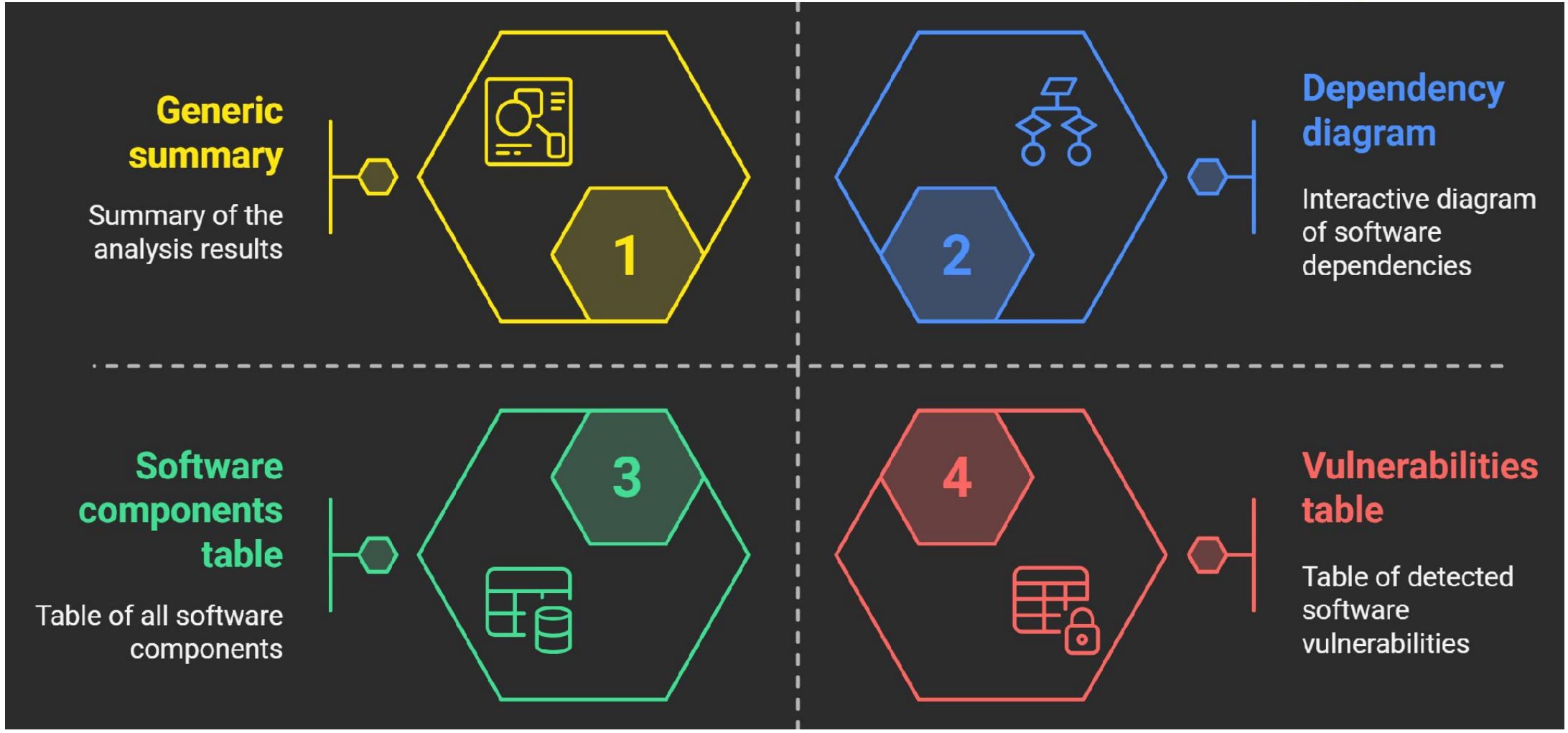
- visit the URL: https://cyclonedx.github.io/Sunshine/
- use Sunshine directly in your browser:
  - upload your SBOM in json format;
  - get the results on a web page.

Web Version – local server

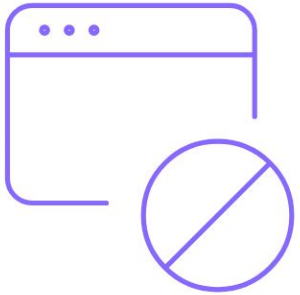
```
# Setup
git clone https://github.com/CycloneDX/Sunshine.git
cd Sunshine
python3 -m venv venv
source venv/bin/activate
pip3 install -r requirements.txt

# Execution
python3 -m http.server 8000
```

Then, just open a browser at URL http://127.0.0.1:8000

OWASP CYCLONEDX SUNSHINE: SEE CYCLONEDX SBOMS COME TO LIFE AND CHAT WITH THEM



Sunshine Capabilities

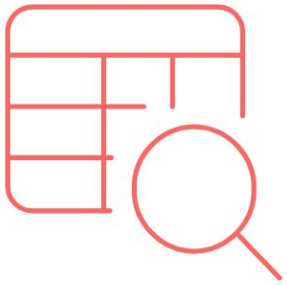


Privacy First

Handles data directly in the browser, maintaining privacy.

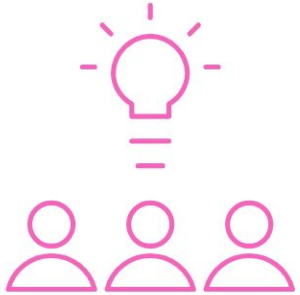
SBOM Integrity

Performs SBOM integrity checks and recovery processes efficiently.



Data Enrichment

Enriches data using EPSS and KEV information.



Awareness Raising

Raises team awareness regarding component management and security.

Practical applications at this point

- Sunshine's output can be used by pentesters as input for their activities
  - **offensive** perspective

- Sunshine's output can be used as a support for internal analysis
  - **defensive** perspective

OWASP CYCLONEDX SUNSHINE:
SEE CYCLONEDX SBOMS COME TO
LIFE AND CHAT WITH THEM

DEMO TIME!

Sunshine workflow

1. We generate one or more SBOMs in CycloneDX format.
2. If necessary, **merge** multiple SBOMs using a dedicated tool such as CycloneDX Merge: https://cyclonedx.github.io/cyclonedx-web-tool/merge
3. Sunshine performs an **integrity check** and, if necessary, carries out **recovery**:
   - automatic recovery of missing bom-refs
   - automatic recovery of broken dependency references
   - circular dependencies detection
4. An interactive **dependency diagram** and **components/vulnerabilities tables** are generated

Dependency diagram

The colors of the segments indicate the vulnerability status of the components:

- **Dark red**: affected by at least one critical severity vulnerability.
- **Red**: affected by at least one high severity vulnerability.
- **Orange**: affected by at least one medium severity vulnerability.
- **Yellow**: affected by at least one low severity vulnerability.
- **Green**: affected by at least one informational severity vulnerability.
- **Light blue**: not directly affected by vulnerabilities but has at least one vulnerable dependency.
- **Grey**: neither the component nor its dependencies are affected by any vulnerabilities.

A real snapshot of the current state of the software

(the original SBOM contained 29,000 lines)

Dependency diagram

The chart is interactive:

- **Hovering**: displays details about a component, including its name, version, and list of vulnerabilities.
- **Clicking**: refocuses the chart. The clicked segment becomes the center (second innermost circle), showing only that component and its dependencies. In this view, the innermost circle is always blue. Clicking the blue circle navigates back up one level in the dependency hierarchy.

Components table

Show 10 entries

Copy CSV Excel Print

Component	Depth	Depends on	Dependency of	Direct vulnerabilities	Transitive vulnerabilities	License
Search Component	Searc	Search Depends on	Search Dependency of	high	Search Transitiv	Search License
aiohttp 3.10.2	root	multidict 6.6.3 aiosignal 1.4.0 attrs 25.3.0 frozenlist 1.7.0 yarl 1.20.1 aiohappyeyeballs 2.6.1 async_timeout 4.0.3	-	High → CVE-2024-52304	-	Apache-1.1 Apache-2.0 MIT
bcprov-jdk18on 1.76	2	-	bcpkix-jdk18on 1.76	High → CVE-2024-30172 Medium → CVE-2024-29857 Medium → CVE-2024-30171 Medium → CVE-2024-34447 Low → CVE-2025-8885	-	MIT
certifi 2024.2.2	root	-	-	High → CVE-2024-39689	-	MPL-2.0
commons-io 2.13.0	1	-	resteasy-client 6.0.0.Final	High → CVE-2024-47554	-	Apache-2.0

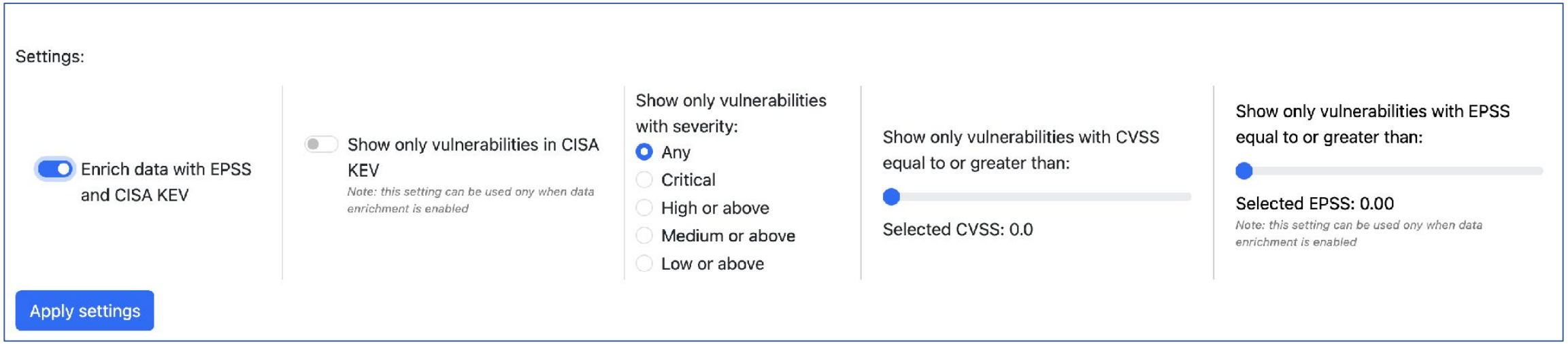
Vulnerabilities table

Show 10 entries

Copy CSV Excel Print

Vulnerability	Severity	Score	Vector	Directly vulnerable components	Transitively vulnerable components
Search Vulnerability	Search Severity	Search Score	Search Vector	Search Directly vulnerable compc	Search Transitively vulnerable compon
CVE-2024-30105	High	7.5	AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	system.text.json 7.0.3	microsoft.fast.components.fluentui 3.5.5
CVE-2024-27086	Low	3.9	AV:L/AC:L/PR:L/UI:R/S:U/C:N/I:L/A:L	microsoft.identity.client 4.59.0	-
CVE-2024-35255	Medium	5.5	AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N	microsoft.identity.client 4.59.0	-
CVE-2023-36558	Medium	5.5	AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N	microsoft.aspnetcore.components 7.0.0	fluxor.blazor.web 5.9.1 fluxor.blazor.web.reduxdevtools 5.9.1 microsoft.aspnetcore.components.web 7.0.0 microsoft.aspnetcore.components.forms 7.0.0

Advanced filters



Query my SBOM

SQL QUERY

```
SELECT * FROM Components WHERE "Direct vulnerabilities" LIKE '%High%'
```

Run SQL | Export CSV

Component	Depth	Depends on
aiohttp 3.10.2	root	attrs 25.3.0, async_timeout 4.0.3, frozenlist 1.7.0, multidict 6
bcprov-jdk18on 1.76	2	-
certifi 2024.2.2	root	-
commons-io 2.13.0	1	-
httpclient5 5.4	1	httpcore5 5.3, httpcore5-h2 5.3
httpclient5 5.4.1	root	slf4j-api 2.0.16, httpcore5-h2 5.3.1, httpcore5 5.3.1
idna 2.10	root	-
jinja2 3.1.4	root	-
jose4j 0.9.3	1	-

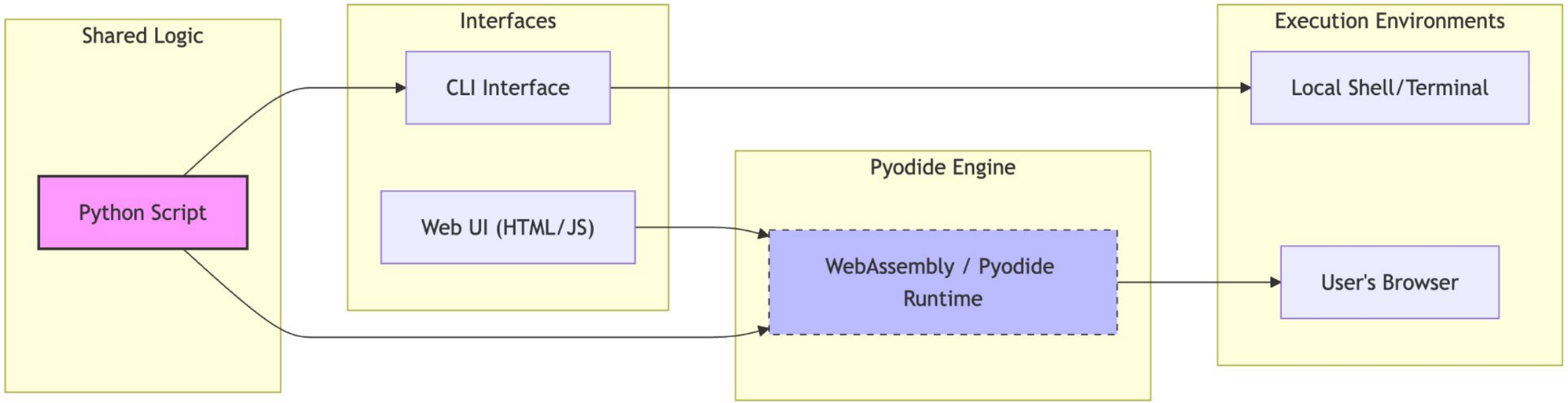
Chat with my SBOM

Privacy first architecture

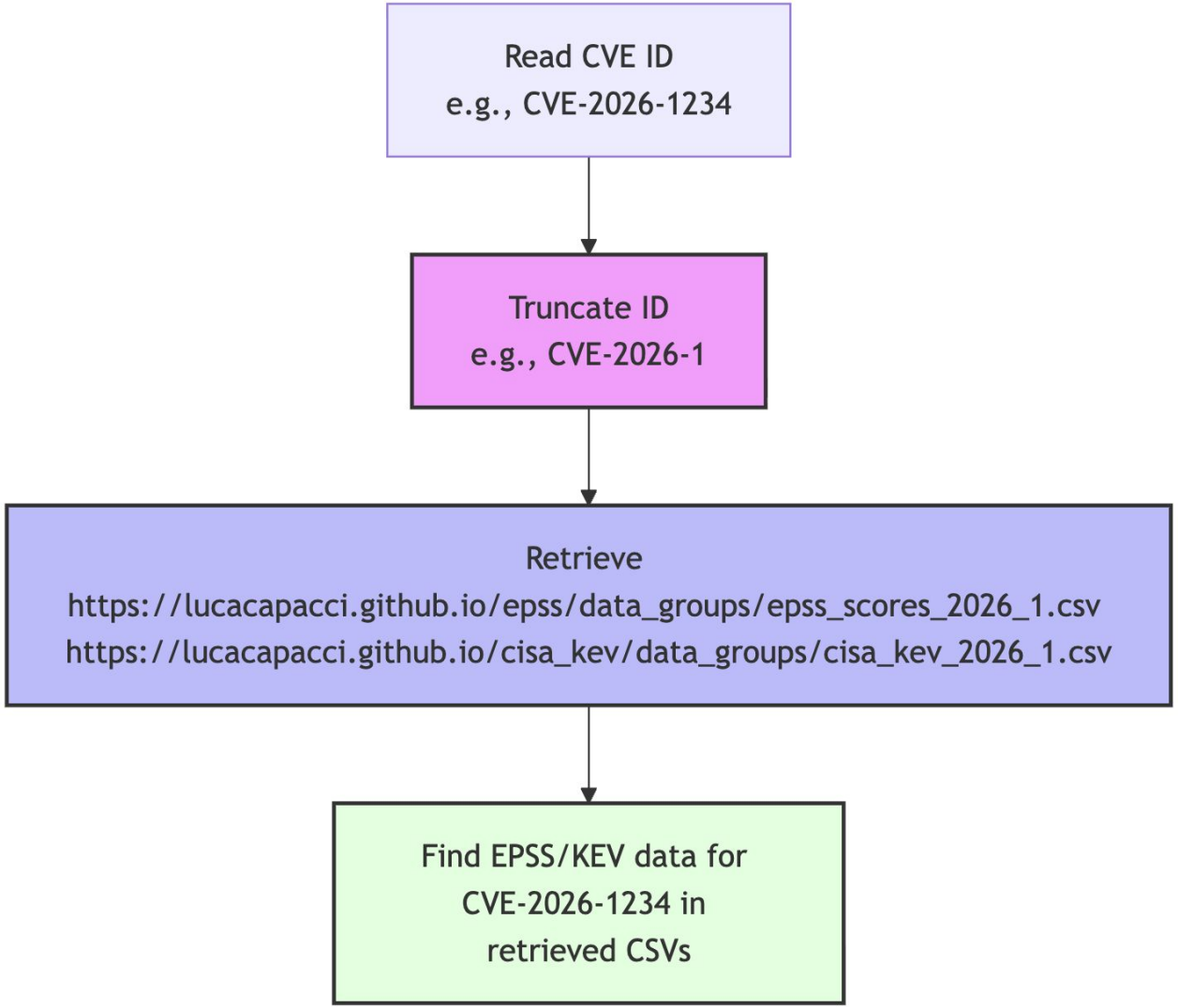
Your SBOM details never leave your browser!

- One code base to rule them all:
  - CLI version → 100% python
  - Web UI version → HTML+JS that executes the very same python script used in the CLI version inside the user's browser, leveraging Pyodide
- Interactive charts → Apache ECharts
- Interactive tables → DataTables
- Data enrichment → Grouped data retrieved from:
  - https://lucacapacci.github.io/cisa_kev
  - https://github.com/lucacapacci/epss
- SQL query engine → DuckDB Wasm
- Chatbot → WebLLM

One code base to rule them all

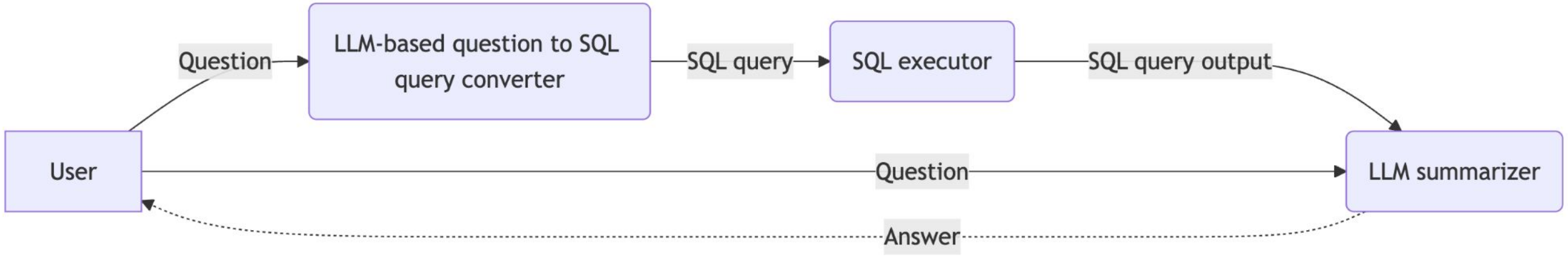


Data enrichment



Chatbot

- In-browser language model inference engine → WebLLM
  - Lightweight model: Qwen2-1.5B-Instruct-q4f16_1-MLC
  - Low temperature
- In-browser SQL query engine → DuckDB Wasm



Chat with my SBOM - example #1

Question-to-SQL-query converter

You:
Which are the most used licenses?

Sunshine:
SELECT "License", COUNT(*) as "Count" FROM "Components" GROUP BY "License" ORDER BY "Count" DESC;

Chat with my SBOM - example #1

Summarizer

You:
Which are the most used licenses?

Sunshine:
The most used licenses in the provided data are:

- MIT: 1210 occurrences
- Apache-2.0: 86 occurrences
- ISC: 79 occurrences
- BSD-3-Clause: 48 occurrences

These licenses appear to be highly prevalent, indicating a strong presence of open-source software under these licenses.

Chat with my SBOM - example #2

Question-to-SQL-query converter

You:
How many components are affected by high or critical vulns?

Sunshine:
SELECT COUNT(*) FROM "Components" c INNER JOIN "Vulnerabilities" v ON v."Directly vulnerable components" LIKE '%' || c."Component" || '%' WHERE (v."Severity" = 'High' OR v."Severity" = 'Critical')

Chat with my SBOM - example #2

Summarizer

You:
How many components are affected by high or critical vulns?

Sunshine:
There are 69 components affected by high or critical vulnerabilities.

OWASP CYCLONEDX SUNSHINE: SEE CYCLONEDX SBOMS COME TO LIFE AND CHAT WITH THEM

What's next?

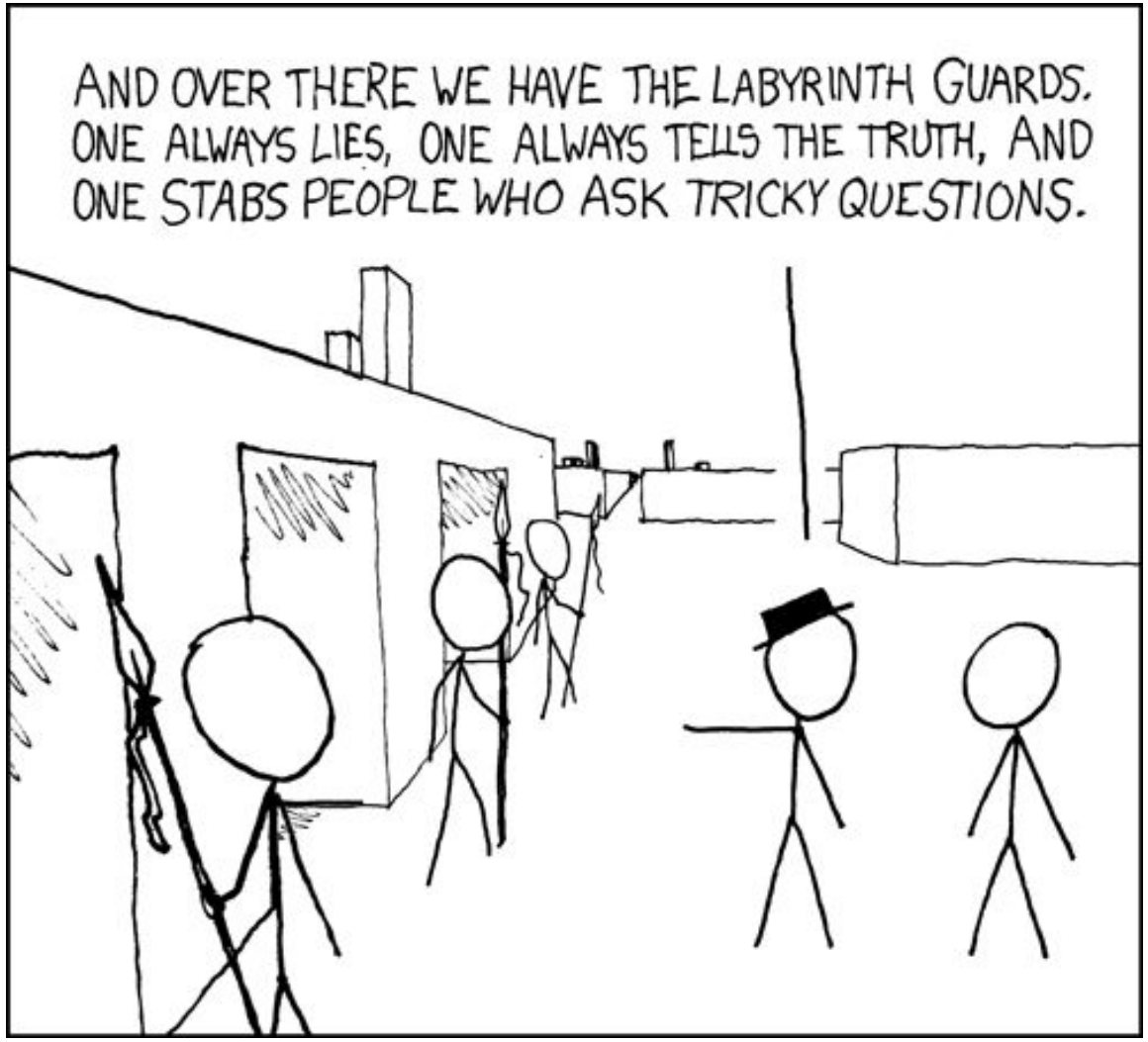
VULN ID	CVE/NVD PUBLISHED	PRODUCT	SEVERITY	SCORE	CVSS VECTOR	SSVC DECISION	SSVC VECTOR	EPSS	CISA KEV	POCS	CWE/CAPEC	LINKS
CVE-2025-29927	2025-03-21	Vercel Next.js	CRITICAL	9.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	ATTEND	CISA:2.0.3/E:P/A:Y/T:T	99.621%	—	1 2 +121	CWE-285, CWE-863	NVD CVE
CVE-2023-4863	2023-09-12	Bandisoft Honeyview, Bentley Seequent Leapfrog, Debian Linux, Fedoraproject, Google Chrome, Microsoft Edge Chromium, ... SHOW MORE	HIGH	8.8	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	ACT	CISA:2.0.3/E:A/A:N/T:T	99.739%	2023-09-13	1 2 +17	CWE-787	NVD CVE KEV
CVE-2021-23358	2021-03-29	Debian Linux, Fedoraproject, Tenable.sc, Underscorejs	HIGH	7.2	CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H	ATTEND	CISA:2.0.3/E:P/A:N/T:T	4.087%	—	1 2 +4	CWE-94	NVD CVE
CVE-2024-34351	2024-05-14	Vercel Next.js	HIGH	7.5	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N	ATTEND	CISA:2.0.3/E:P/A:Y/T:P	5.453%	—	1 2 +3	CWE-918	NVD CVE
CVE-2024-46982	2024-09-17	Vercel Next.js	HIGH	7.5	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	ATTEND	CISA:2.0.3/E:P/A:Y/T:P	58.768%	—	1 2 +3	CWE-639	NVD CVE
CVE-2023-45857	2023-11-08	Axios	MEDIUM	6.5	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N	ATTEND	CISA:2.0.3/E:P/A:Y/T:P	0.556%	—	1 2 +2	CWE-352	NVD CVE

OWASP CYCLONEDX SUNSHINE:
SEE CYCLONEDX SBOMS COME TO
LIFE AND CHAT WITH THEM

What's next?

SSVC DECISION	SSVC VECTOR	EPSS	CISA KEV	POCS	CWE/CAPEC	LINKS
ATTEND	CISA:2.0.3/E:P/A:Y/T:T	99.621%	—	1 2 +121	CWE-285, CWE-863	NVD CVE
ACT	CISA:2.0.3/E:A/A:N/T:T	99.739%	2023-09-13	1 2 +17	CWE-787	NVD CVE KEV

You can already try by copying any text containing CVEs and pasting it to:
https://vinthub.pages.dev/



https://xkcd.com/246/

Questions?

Suggestions?

Luca Capacci
luca.capacci@gmail.com

Mattia Fierro
fierro.mattia@gmail.com


OWASP GLOBAL AppSec
VIENNA'26 JUN 25-26
25 years
of open source security
THANK YOU!