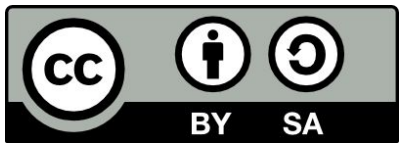
FOSDEM 2025

Implementing a triage process supporting all flavours of VEX

Licenced under Creative Commons Attribution-ShareAlike 4.0 International Licence



© Anthony Harrison 2025

- 40 years delivering mission critical solutions across multiple sectors
- Founder and Director APH10
- SBOM Europe
- Open Source Software
- STEM Ambassador
- Mentor

CoderDojo

© Anthony Harrison 2025

CVE-Bin-Tool

- Binary vulnerability scanner
- Intel OSS since 2020 based on NVD
  - Now supports OSV and other sources
- GPL 3.0 or later Licence
- SBOM support since 2021 (SPDX and CycloneDX)
- CISA KEV support since 2022
- EPSS support since 2023
- GSOC project for past 4 years
- OpenSSF Best Practices
- 1.3K+ stars on GitHub
- 200+ contributors

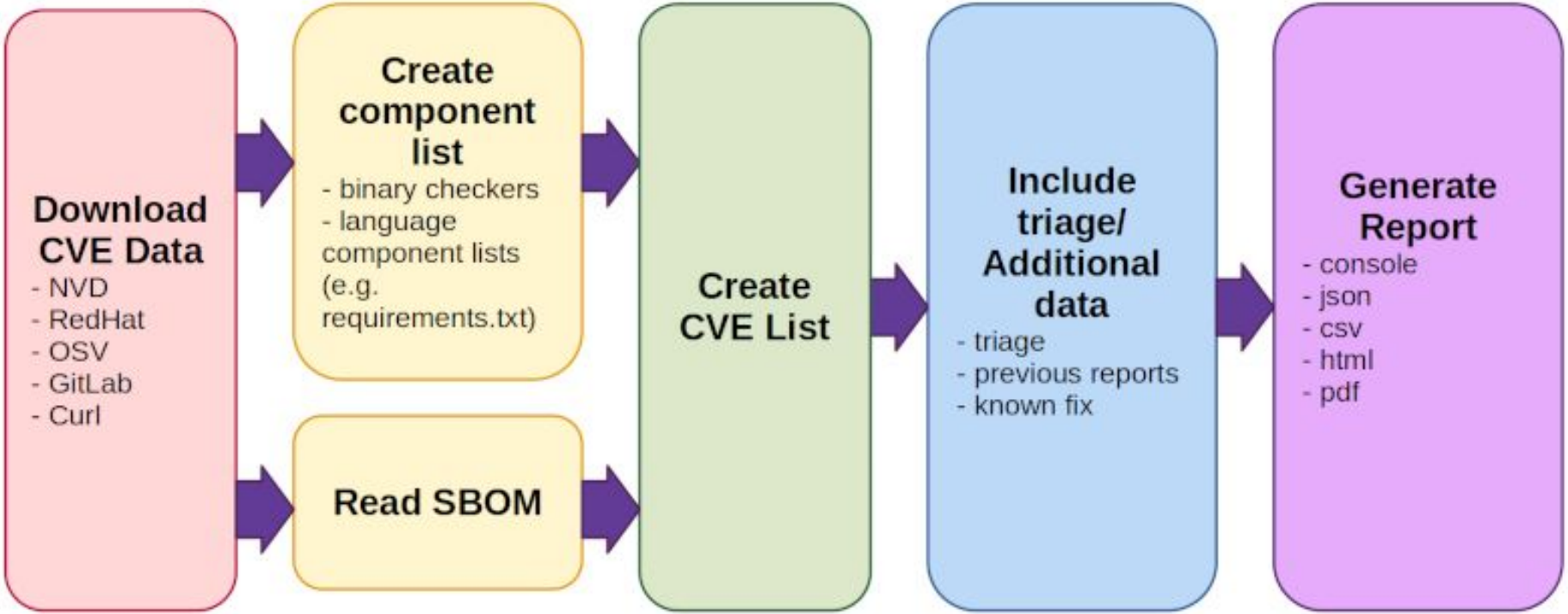
© Anthony Harrison 2025

GSOC 2024 Team

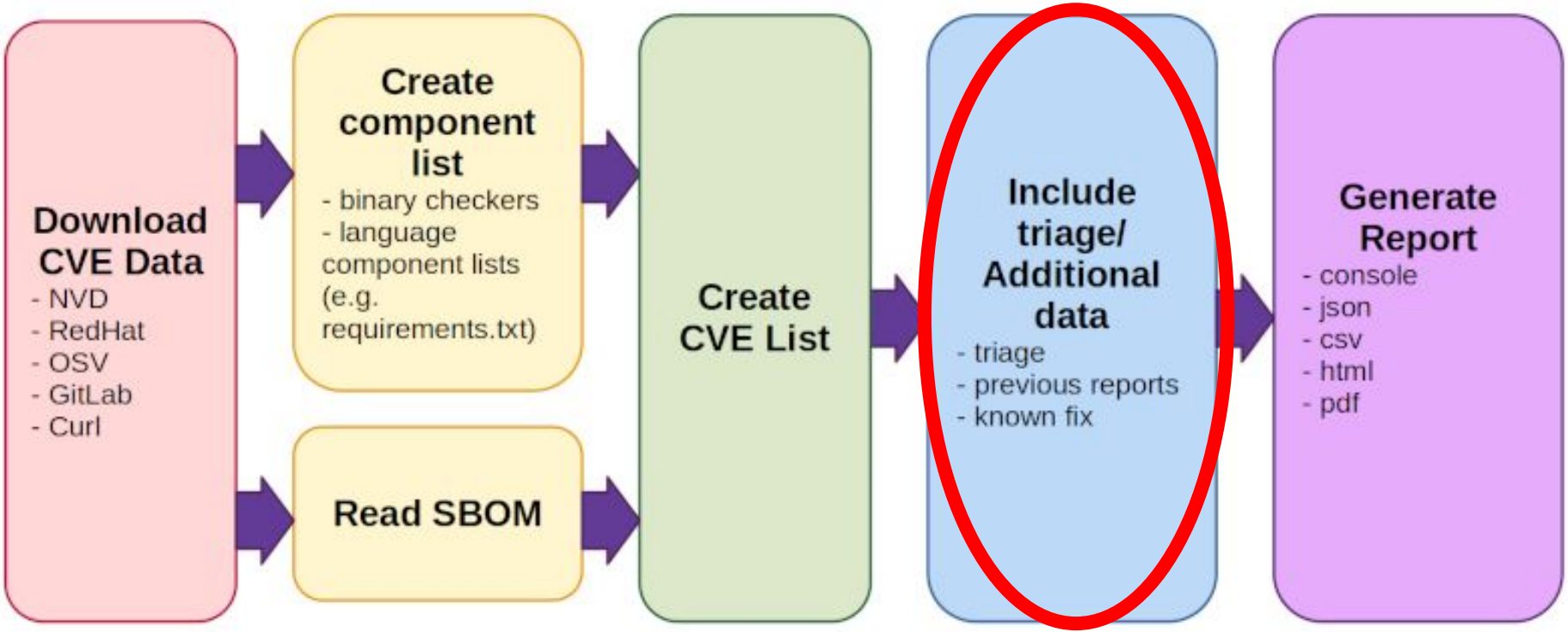
MENTEES

MENTORS

© Anthony Harrison 2025



© Anthony Harrison 2025



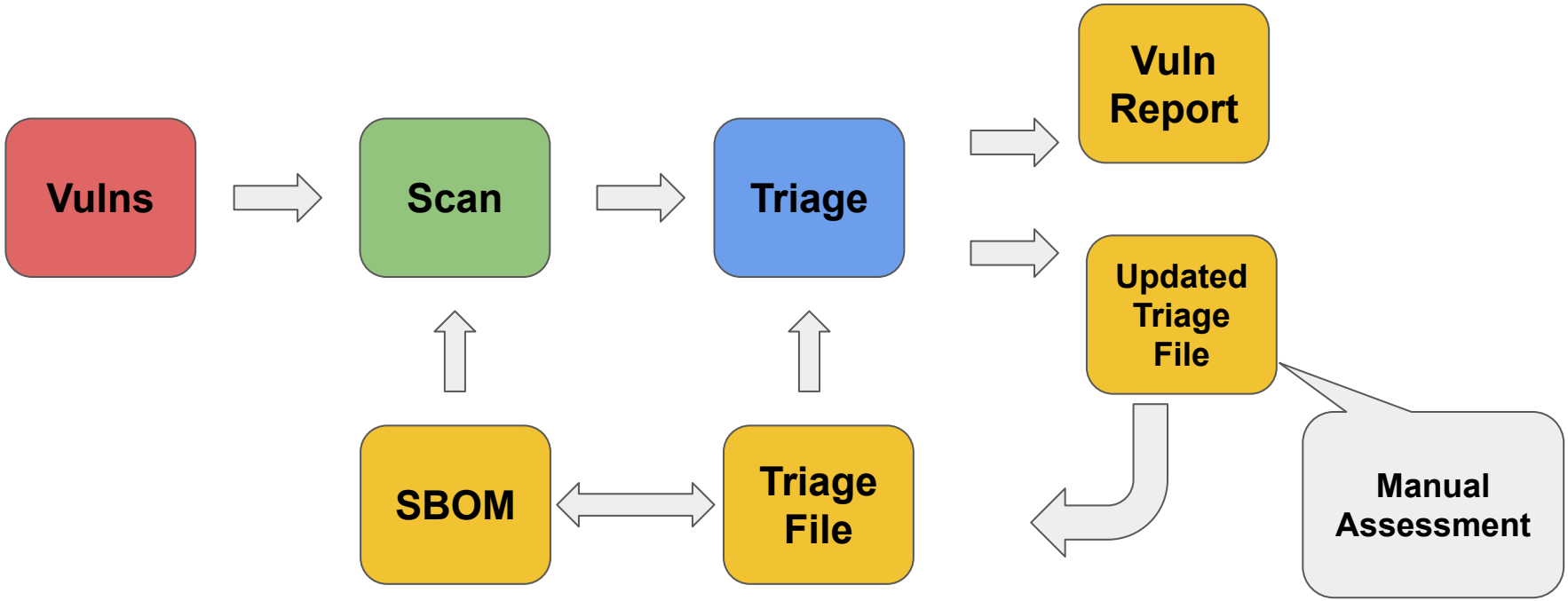
© Anthony Harrison 2025

GSOC 2024

- **Improved Triage Support**
- Existing triage process
  - Bespoke format and CycloneDX
- Proposed triage process
  - Take a triage file and associated SBOM
  - Ensure that the triage file is linked with the SBOM
  - Scan the SBOM for vulnerabilities
  - Remove the vulnerabilities which have been marked as not applicable
  - Report the remaining vulnerabilities
  - Support at least CSAF, CycloneDX and OpenVEX VEX formats

© Anthony Harrison 2025

Triage Process



© Anthony Harrison 2025

VEX

- VEX stands for **"Vulnerability Exploitability eXchange"**
- A form of a security advisory that indicates whether a product or products are affected by a known vulnerability or vulnerabilities
- A document format used to communicate whether a specific vulnerability is actually exploitable in a given software context
- It is NOT just a list of vulnerabilities

© Anthony Harrison 2025

VEX Comparison

	CSAF	CycloneDX	OpenVEX	SPDX
Vuln Id	Y	Y	Y	Y
Status	Y	Y	Y	Y
Component	N	Y	Y	Y
SBOM Link	Feasible	Y	N	Y
Remediation	Y	Y	C	C
Not exploitable justification	Y	Y	Y	Y

© Anthony Harrison 2025

VEX Use cases

- CSAF
  - Focus on system security aspects
- CycloneDX
  - Focus on software supply chain security in conjunction with software bill-of-materials (SBOMs)
- OpenVEX
  - Focus on vulnerability information exchange
- SPDX
  - Focus on software supply chain security

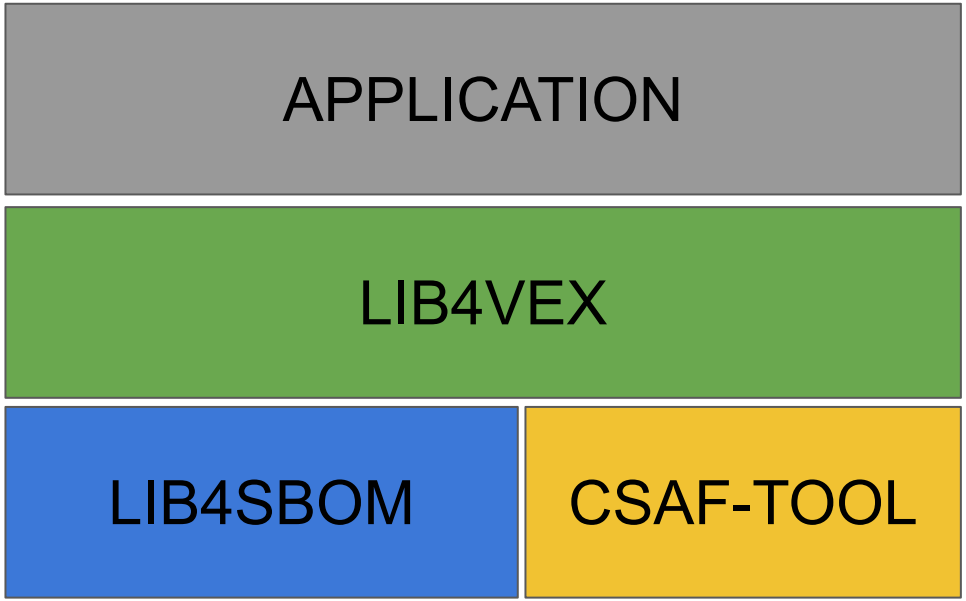
© Anthony Harrison 2025

Lib4VEX

- Python library
- Generate VEX documents in JSON format
- Parse VEX documents and extract vulnerability information
- Supports CSAF, CycloneDX, OpenVEX and SPDX formats
- Generated VEX document can be output to a file or to the console
- VEX document supports a single version of a product
  - A product has a single SBOM

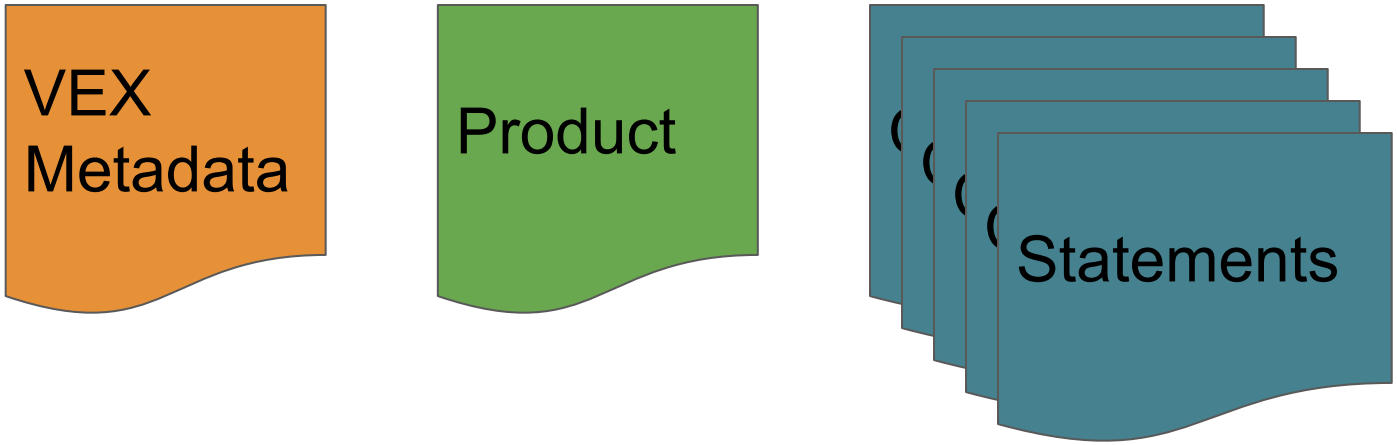
© Anthony Harrison 2025

Lib4VEX



© Anthony Harrison 2025

VEX Content



© Anthony Harrison 2025

Lib4VEX - Design Decisions

- Provides abstraction for Product definition
  - Optional link to SBOM
  - Only vulnerabilities for components included in the SBOM are included in the VEX document.
- Provides abstraction for Vulnerability
  - Status and justification enumerations validated against VEX format
- The VEX document MAY contains all reported vulnerabilities and the respective status.
  - Essentially merging multiple VEX documents together
- The latest VEX is indicated by the latest timestamp.
  - The previous VEX documents are retained for audit purposes.

© Anthony Harrison 2025

VEX Creation/Product Definition

```
vextype="csaf" # can be also cyclonedx, openvex, spdx

vexgen = VEXGenerator(vex_type=vextype, author="ACME_Division")
# Specify product
vexgen.set_product
(name="ACME", release="1.0", sbom="samples/example.json")
```

© Anthony Harrison 2025

VEX Metadata

```
metadata={}
metadata["id"]="ACME-1.0-VEX"
metadata["title"]="ACME-1.0-VEX Use Case complete"
metadata["comment"]="ACME PoC II VEX document. Demo only."
metadata["supplier"]="Fred Flintstone"
metadata["supplier_url"]="fredflintstone@acme.com"
metadata["status"]="draft"

metadata["revision_reason"] = "New vulnerability CVE-2024-6789”
```

© Anthony Harrison 2025

VEX Vulnerability - Definition

```
vulnerability = Vulnerability(validation=vextype)
vulnerabilities = []

# Specify vulnerability by product name/version
vulnerability.initialise()
vulnerability.set_id("CVE-2023-12345")
vulnerability.set_name("pyyaml")
vulnerability.set_release("6.0.1")
# Alternative via PURL
# vulnerability.set_value("purl", "pkg:pypi/pyyaml@6.0.1")
vulnerability.set_status("under_investigation") # VEX type specific
vulnerabilities.append(vulnerability.get_vulnerability())
```

© Anthony Harrison 2025

VEX Vulnerability - Update

```
# Specify vulnerability by product name/version
vulnerability.set_id("CVE-2023-12345")
vulnerability.set_name("pyyaml")
vulnerability.set_release("6.0.1")
vulnerability.set_status("known_not_affected")
# Justify decision
vulnerability.set_justification("vulnerable_code_not_in_execute_path")
vulnerabilities.append(vulnerability.get_vulnerability())
```

© Anthony Harrison 2025

VEX Parsing

```
from lib4vex.parser import VEXParser

vexparser = VEXParser(vex_type="csaf")

vexparser.parse("samples/csaf/acme_1.0_vex.json")

vexparser.get_metadata()

{'version': '2.0', 'title': 'ACME-1.0-VEX Use Case complete',...}

vexparser.get_product()

{'CSAFPID_0001': {'vendor': 'ACME_Division', 'product': 'ACME',
'version': '1.0', 'family': ''}}

vexparser.get_vulnerabilities()

[{'id': 'CVE-2023-12345', 'description': 'Product': 'CSAFPID_0001',...
'status': 'under_investigation'}, … ]
```

© Anthony Harrison 2025

Lib4VEX Project Page

https://github.com/anthonyharrison/lib4vex

Worked example available as a Tutorial

Example files available in CSAF, CycloneDX, OpenVEX and SPDX

© Anthony Harrison 2025

CVE Use Case

Generate initial triage file

```
cve-bin-tool --sbom-file <sbom>
             --vex-output <triage.json>
             --vex-type <type of vex>
             --product <product name>
             --version <product version>
             --vendor <product vendor>
```

© Anthony Harrison 2025

CVE Use Case

Use triage file in subsequent scans

```
cve-bin-tool --sbom-file <sbom>
             --vex-file <triage.json>
             --vex-type <type of vex>
             --vex-output <triage.json>
             --product <product name>
             --version <product version>
             --vendor <product vendor>
             -rr <reason for update>
```

© Anthony Harrison 2025

Implementation Decisions

All new vulnerabilities:

vex status = “in_triage” (value depends on VEX type)

Existing vulnerabilities:

vex status is unmodified

© Anthony Harrison 2025

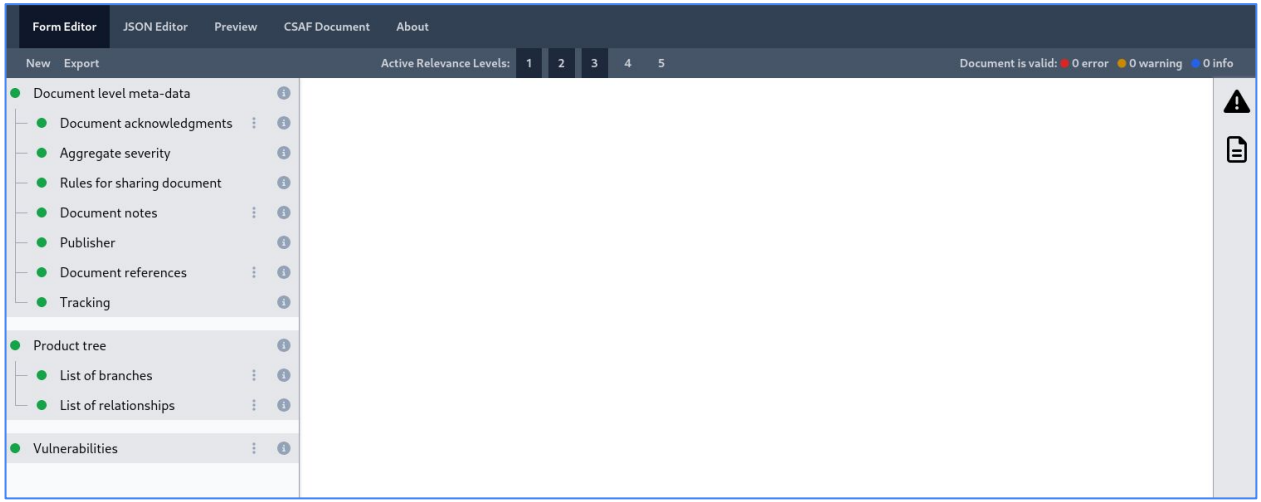
VEX Challenges

- Validation
- Scope
- VEX Users

© Anthony Harrison 2025

Validation - CSAF

Secvisogram Tool https://secvisogram.github.io/



© Anthony Harrison 2025

Validation - CycloneDX

Various validators available

Sbom-utility - https://github.com/CycloneDX/sbom-utility

works for multiple versions of CycloneDX

Online validator - https://cyclonedx.github.io/cyclonedx-web-tool/

Only works up to version 1.5

© Anthony Harrison 2025

Validation - OpenVEX

At the time the library was started there was no OpenVEX validator

Validation was based on ‘copying’ examples in the specification

© Anthony Harrison 2025

Validation - SPDX

At the time the library was started there was no SPX 3.0 validator

Validation was based on ‘copying’ examples in the specification

© Anthony Harrison 2025

Validation - SPDX

At the time the library was started there was no SPX 3.0 validator

Validation was based on ‘copying’ examples in the specification

UPDATE

Online validators now available (January 2025)

- Requires spec version 3.0.1 (was 3.0.0)
- Latest status - Lib4vex files now validate

© Anthony Harrison 2025

Scope

- VEX only reports vulnerable components
  - How do you know the VEX is complete?
- How do you demonstrate that a component is not vulnerable?
  - cve-bin-tool reports components with no known vulnerabilities
  - NOT addressed by any of the VEX formats

© Anthony Harrison 2025

VEX Users

Are all VEX Users JSON experts?

Triage process requires additional elements to be included not just a simple status change

© Anthony Harrison 2025

VEX Users

Are all VEX Users JSON experts?

Triage process requires additional elements to be included not just a simple status change

WE NEED A VEX Editor

© Anthony Harrison 2025

VEX Users

Are all VEX Users JSON experts?

Triage process requires additional elements to be included not just a simple status change

WE NEED A VEX Editor

But there is a documentation tool!

© Anthony Harrison 2025

VEX Users

Vex2doc - https://pypi.org/project/vex2doc/

Product Summary

Item	Details
Name	sbom2doc
Version	0.4.0

Vulnerabilities Summary

Bom-ref	Product	Release	Id	Source-name	Source-url	Description	Created	Updated	Status
reportlab@3.6.12	reportlab	3.6.12	CVE-2023-33733	NVD	https://nvd.nist.gov/vu…	Reportlab up to v3.6.12 allows attackers to execute arbitrary code via supplying a crafted PDF file.	2025-01-24T12:24:13Z	2025-01-24T12:24:13Z	in_triage

© Anthony Harrison 2025

VEX Support

	cvebintool	grype	trivy	Dependency track
CSAF	Y			
CycloneDX	Y		Y	Y
OpenVEX	Y	Y	Y	
SPDX	Y (*)			

© Anthony Harrison 2025

Links to Tools

Cve-bin-tool - https://github.com/intel/cve-bin-tool and https://pypi.org/project/cve-bin-tool/

Lib4vex - https://pypi.org/project/lib4vex/

Lib4sbom - https://pypi.org/project/lib4sbom/

Csaf-tool - https://pypi.org/project/csaf-tool/

Vex2doc - https://pypi.org/project/vex2doc/

Sbom-utility - https://github.com/CycloneDX/sbom-utility

CycloneDX Online validator - https://cyclonedx.github.io/cyclonedx-web-tool/

Secvisogram Tool - https://secvisogram.github.io/

© Anthony Harrison 2025

Contact Details

LinkedIn

GitHub

anthony@aph10.com

sbomeurope.eu

© Anthony Harrison 2025