



A crash course on SBOMs and OWASP CycloneDX for software engineering teams

Patrick Dwyer



@coderpatros



patrick.dwyer@owasp.org

<https://www.linkedin.com/in/coderpatros/>

<https://github.com/coderpatros>

- OWASP CycloneDX Project Co-Lead
- Contributor to multiple SBOM related projects and tools
- OSS Maintainer
- Senior Product Security Engineer (ServiceNow)
Secure Software Development Lifecycle Team

Agenda

- What are we talking about
- SBOM intro
- CycloneDX intro
- Use cases
- How to

Supply Chain Security



IT Asset Management

Modern Software Development

- Increasing reliance on 3rd party components
 - Libraries
 - Frameworks
 - Tools
 - Services
 - Infrastructure, virtualisation, containers, etc
- More modern practices like agile and DevOps
- Benefits include
 - Reduced time to market
 - Reduced cost
 - Improved quality (well... hopefully)

But what about risks?

Food Allergies

Food Labelling Standards

- Made in Australia from at least 95% Australian ingredients
- Malt extract from barley
- Allergen statement



Software Bill of Materials

An SBOM is a formal, machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships.

Credit: NTIA, Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)

https://www.ntia.gov/files/ntia/publications/framingsbom_20191112.pdf

Some External Factors

- US NTIA Software Component Transparency Initiative (first meeting July 19, 2018)
<https://www.ntia.doc.gov/SoftwareTransparency>
- US President Executive Order (EO) 14028, Improving the Nation's Cybersecurity (May 12, 2021)
<https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/>
- US Office of Management and Budget Memorandum M-22-18, Enhancing the Security of the Software Supply Chain through Secure Software Development Practices (September 14, 2022)
<https://www.whitehouse.gov/wp-content/uploads/2022/09/M-22-18.pdf>

SBOM Formats (NTIA)

SPDX is an open standard for communicating software bill of material information (including components, licenses, copyrights, and security references). The SPDX specification is developed by the SPDX workgroup, which is hosted by The Linux Foundation. It is specified in international standard ISO/IEC 5962:2021.

CycloneDX is an OWASP Flagship Standards Project. As a full stack bill of materials standard, it has been purpose-built for software security contexts and supply chain component analysis.

SWID tags record unique information about an installed software application, including its name, edition, version, whether it is part of a bundle and more. SWID tags support software inventory and asset management initiatives. The structure of SWID tags is specified in international standard ISO/IEC 19770-2:2015.


Introducing CycloneDX

- Flagship OWASP Standards Project
- Purpose built as a BOM format for cybersecurity use cases
- Lightweight, simplicity over complexity - easy to implement and adopt
- Optimized for highly automated processes
- Designed in May 2017
- Initial release March 2018
- Yearly releases since
- Formal governance and standards process
- Recommended by multiple world government agencies
- Large and growing industry and vendor support
 - <https://cyclonedx.org/about/supporters/>
- Estimated to be in use at 100k organizations

CycloneDX Supporters, Vendors, and Projects



Use Case Examples

GETTING STARTEDSPECIFICATIONABOUTTwitterGitHubEmailDiscordYouTube

Use Cases

The following examples provide guidance as to the minimal fields required to achieve specific use cases. Ideally, all optional fields would be populated in order to achieve all use cases. Many of the cases highlighted are directly or closely related to security.

Inventory

A complete and accurate inventory of all first-party and third-party components is essential for risk identification. BOMs should ideally contain all direct and transitive components and the dependency relationships between them.

CycloneDX is capable of describing the following types of components:

COMPONENT TYPE	CLASS
Application	Component
Container	Component
Device	Component
Library	Component
File	Component
Firmware	Component

Inventory

Known vulnerabilities

Integrity verification

Authenticity

Package evaluation

License compliance

Assembly

Dependency graph

Provenance

Pedigree

Service definition

Properties / name-value store

Packaging and distribution

Composition completeness

OpenChain conformance

Vulnerability remediation

Vulnerability disclosure

Security advisories

External references

A collection of common use cases achievable with CycloneDX along with concrete examples in XML and JSON.

<https://cyclonedx.org/use-cases/>

BOM Metadata

```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.3",
  "serialNumber": "urn:uuid:3e671687-395b-41f5-a30f-a58921a69b79",
  "version": 1,
  "metadata": {
    {
      "timestamp": "2020-04-13T20:20:39+00:00",
      "tools": [ ... ],
      "authors": [ ... ],
      "manufacture": { ... },
      "supplier": { ... },
      "component": { ... }
    }
  }
}
```


Component Inventory

```
{  
  ...  
  "components": [  
    {  
      "type": "library",  
      "group": "org.apache.logging.log4j",  
      "name": "log4j2-core",  
      "version": "2.14.1"  
    }  
  ]  
}
```

Supports:

- Applications
- Libraries
- Frameworks
- Containers
- Operating systems
- Firmware
- Devices
- Files
- Services

Identify Known vulnerabilities

```
{  
  ...  
  "components": [  
    {  
      "type": "library",  
      "group": "org.apache.logging.log4j",  
      "name": "log4j2-core",  
      "version": "2.14.1",  
      "cpe": "cpe:2.3:a:apache:log4j:2.14.1",  
      "purl": "mvn:org.apache.logging.log4j/log4j-core@2.14.1",  
      "swid": { ... }  
    }  
  ]  
}
```

Integrity

```
{  
  ...  
  "components": [  
    {  
      "type": "library",  
      "group": "org.apache.logging.log4j",  
      "name": "log4j2-core",  
      "version": "2.14.1",  
      "hashes": [  
        { "alg": "SHA3-512", "content": "..." }  
      ]  
    }  
  ]  
}
```

Component Pedigree

```
"pedigree": {  
  "ancestors": [  
    {  
      "type": "library",  
      "group": "org.apache.logging.log4j",  
      "name": "log4j2-core",  
      "version": "2.14.1"  
    }  
  ],  
  "patches": [  
    {  
      "type": "backport",  
      "diff": { ..., "resolves": [{ "type": "security", "id": "CVE-2021-44228", ... }] }  
    }  
  ]  
}
```

Provenance

- Component downloaded location
- Supplier
- Author
- Publisher

Services

- Provider
- Endpoints
- Authentication requirements
- Trust boundary traversal
- Data flow and classification

Authenticity

- XML Signature
- JSON Web Signature (JWS)
- JSON Signature Format (JSF)
- Digital signatures can be applied to a BOM or to an assembly within a BOM
- Signatures can be external to the BOM or enveloped (included within)

Composition

- Assemblies
- Dependency graph
- Completeness
 - complete
 - incomplete
 - first-party/third-party
 - unknown

Vulnerability Exploitability Exchange

- Is this exploitable in the context of the assembled software/system?
- If it's not exploitable why?
- If it is what can be done about it?

and many, many more...

Tool Center

The screenshot shows the CycloneDX Tool Center interface. At the top is a dark navigation bar with the CycloneDX logo and links for GETTING STARTED, SPECIFICATION, ABOUT, and social media icons. Below this is a 'Tool Center' header. A filter bar allows users to show all tools (79) or filter by license (Open source: 65, Proprietary: 14) and function (Build integration: 34, Analysis: 21, Author: 1, GitHub action: 7, Transform: 5). Further filters include Library (8), Signing / Notary (2), and Distribute (1). The main area displays six tool cards:

- Auditjs** (Sonatype): Audits an NPM package.json file to identify known vulnerabilities. 39 Forks, 158 Stars. Tags: opensource, build-integration.
- BOM Repository Server** (CycloneDX): A lightweight repository server used to publish, manage, and distribute CycloneDX SBOMs. 0 Forks, 8 Stars. Tags: opensource, distribute.
- Chelsea** (Sonatype): Dependency vulnerability auditor for Ruby. 3 Forks, 7 Stars. Tags: opensource, build-integration.
- CodeNotary vcn** (CodeNotary): Protects an organizations software development pipeline from supply chain attacks. CodeNotary natively supports CycloneDX SBOMs. 20 Forks, 115 Stars. Tags: opensource, signing-notary.
- CodeSentry** (GammaTech): Software Composition Analysis (SCA) platform that leverages binary analysis to identify components, inherited risk, and communicates inventory through CycloneDX SBOMs. Tags: proprietary, analysis.
- Contrast Security** (Contrast Security): Automatically generates component inventory from runtime analysis (IAST or RASP) and generates CycloneDX SBOMs. Tags: proprietary, analysis.

Community effort to establish a marketplace of free, open source, and proprietary tools and solutions that support CycloneDX.

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

How to make an SBOM

- Normal SCA tools
- Container scanners
- SCM integrated solutions (i.e. GitHub Dependabot)
- Build process tools

Build and ship SBOMs like you build and ship software

- Modern systems/software are increasingly complex
- Package managers and build systems are too
- The most accurate BOMs are generated by process integrated in your build
- Leverage your existing CI/CD infrastructure
- Build & release artifacts
- Container registries
- CycloneDX BOM Repository Server
- BOM retrieval via /.well-known/sbom well known URI
- BOM reference in cloud resource tags
- OWASP Dependency-Track Project

Sharing SBOMs with customers

- Analyse your SBOMs from a customer perspective
- Component pedigree
- Vulnerability Exploitability Exchange (VEX)

Community Participation

- Website (introduction, use cases, tool center, and specification)
 - <https://cyclonedx.org/>
- GitHub
 - <https://github.com/CycloneDX>
- Slack
 - <https://cyclonedx.org/slack>
 - <https://cyclonedx.org/slack/invite>
- Mailing List
 - <https://cyclonedx.org/discussion>

Thank You

