

CycloneDX Software Bill of Materials



About

Patrick Dwyer

- CycloneDX Core Working Group
- OSS Maintainer
- Dev Lead for a Government Organisation



@coderpatros



patrick.dwyer@owasp.org

Food allergies

Food Labelling Standards

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



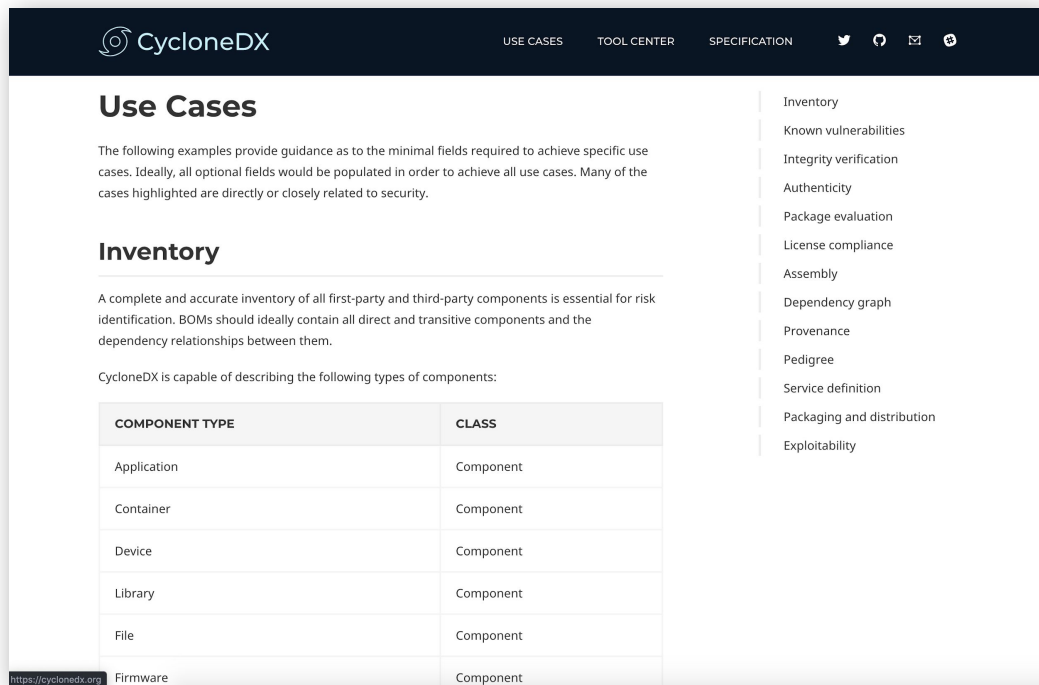
History

- Origins in the OWASP community
- Designed in May 2017
- Initial release in March 2018
- OWASP Dependency-Track was first adopter, many others followed
- CycloneDX v1.1 released in March 2019
- CycloneDX v1.2 released in May 2020
- Formal CycloneDX working group and standardization process in 2020
- Members of CycloneDX Core working group are OWASP leaders/members

The CycloneDX Approach

- Easy to adopt – easy to contribute
- Identify risk to as many adopters as possible, as quickly as possible
- Avoid any/all blockers that prevent the identification of risk
- Continuous improvement – Innovate quickly, improve over time
- Encourage innovation and competition through extensions
- Produce immutable and backward compatible releases
- Facts first – Dynamic facts and observations enabled through extensions
- Automation and optimization of BOM creation
- Full-stack BOM specification

Use Case Examples



The screenshot shows the CycloneDX website with a dark header. The 'USE CASES' tab is selected. The 'Use Cases' section includes an introductory paragraph and an 'Inventory' subsection. The 'Inventory' subsection contains a paragraph about BOMs and a table of component types. A sidebar on the right lists various use cases. The footer includes the URL <https://cyclonedx.org>.

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 |

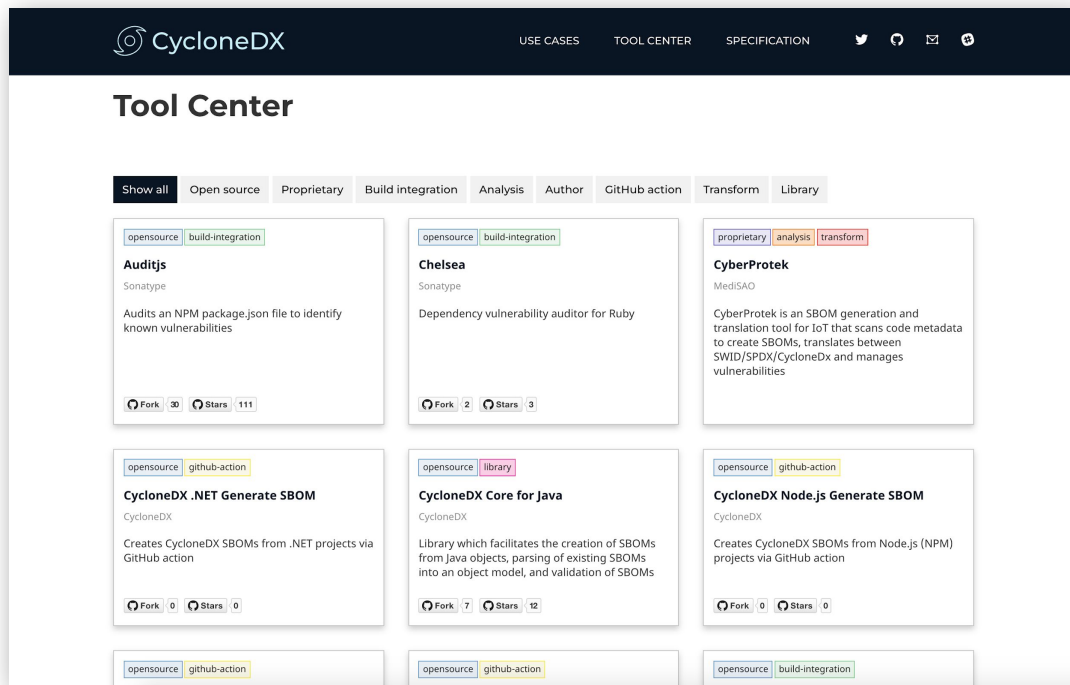
- Inventory
- Known vulnerabilities
- Integrity verification
- Authenticity
- Package evaluation
- License compliance
- Assembly
- Dependency graph
- Provenance
- Pedigree
- Service definition
- Packaging and distribution
- Exploitability

<https://cyclonedx.org>

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



Tool Center



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

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>