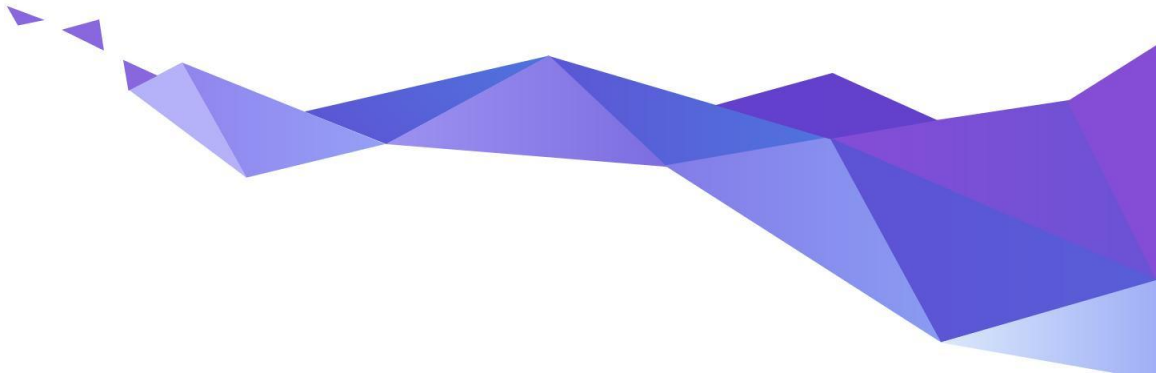
Practical SBOM Management with Zephyr and SPDX

Benjamin Cabé
benjamin@zephyrproject.org



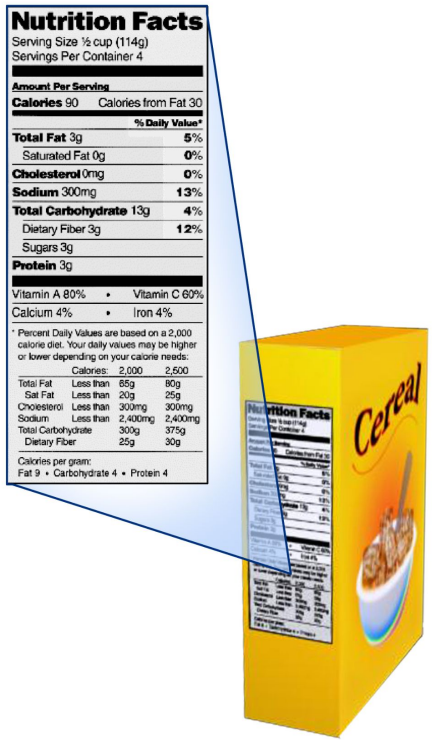
Agenda

- What is an SBOM, and why you should care
- Challenges of capturing an SBOM in the context of embedded development
- SPDX to the rescue
- Generating SBOM for your Zephyr application
- Useful tools to make the most of your SBOM
- Q & A

Software Bill of Materials?

Inventory of all the components (and their dependencies) in a software product

Essential for extending **supply chain management** to software



Why you need SBOMs...

Manage security vulnerabilities

Comply with regulations

Improve decision making

Streamline security audits & certifications

Embedded can make things tricky

- Mixing open-source / proprietary software
- Toolchain Integration
- Longevity
- Complex supply chain (silicon vendor HAL, RTOS, modules, libraries, custom app code, ...)

Standardizing SBOM description

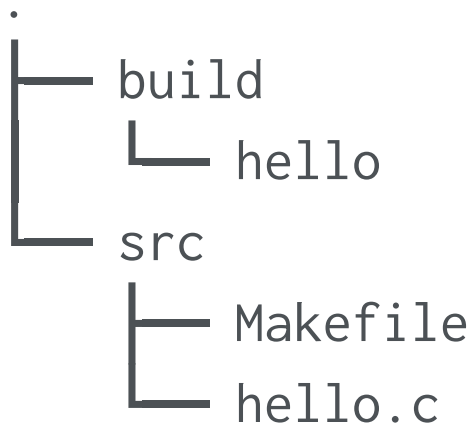
Zephyr®

"

SPDX (System Package Data Exchange) is an **open standard** for communicating **software bill of material information**, including **provenance**, **license**, **security**, and other related information.

Anatomy of an SDPX file

Basic program example:

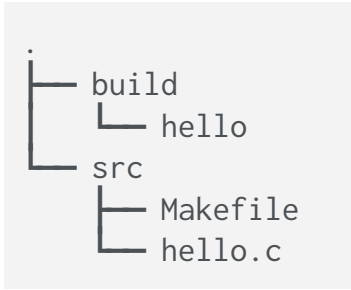


Anatomy of an SDPX file


```
SPDXVersion: SPDX-2.2
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: hello
Creator: Person: Jane Doe (jane@doe.net)
Creator: Tool: github.com/spdx/tools-golang/builder
Creator: Tool: github.com/spdx/tools-golang/idsearcher
Created: 2024-03-26T01:46:00Z

...
```

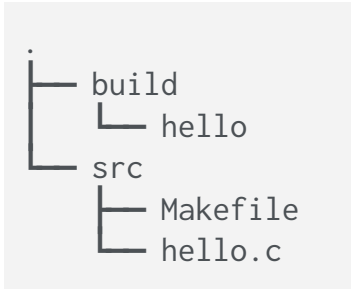
```
.
├── build
│   └── hello
└── src
    ├── Makefile
    └── hello.c
```

Anatomy of an SDPX file



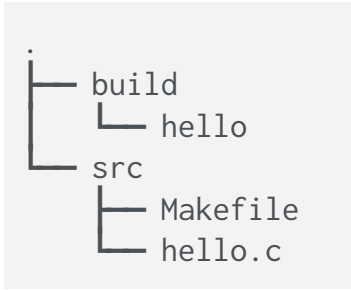

```
...

PackageName: hello
SPDXID: SPDXRef-Package-hello
PackageDownloadLocation: git+https://github.com/myrepo/hello.git
FilesAnalyzed: true
PackageVerificationCode: 9d20237bb72087e87069f96afb41c6ca2fa2a342
PackageLicenseConcluded: GPL-3.0-or-later
PackageLicenseDeclared: GPL-3.0-or-later
PackageCopyrightText: NOASSERTION

Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-Package-hello

...
```

Anatomy of an SDPX file




```
...

FileName: ./build/hello
SPDXID: SPDXRef-hello-binary
FileType: BINARY
FileChecksum: SHA1: 20291a81ef065ff891b537b64d4fdccaf6f5ac02
FileChecksum: SHA256: 83a33ff09648bb5fc5272baca88cf2b59fd81ac4cc6817b86998136af368708e
FileChecksum: MD5: 08a12c966d776864cc1eb41fd03c3c3d
LicenseConcluded: GPL-3.0-or-later
LicenseInfoInFile: NOASSERTION
FileCopyrightText: NOASSERTION

...
```

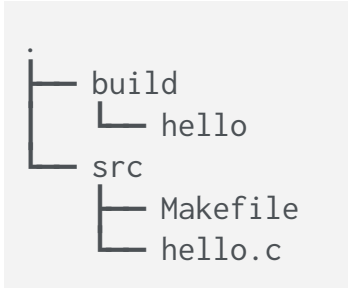
Anatomy of an SDPX file




```
...

FileName: ./src/hello.c
SPDXID: SPDXRef-hello-src
FileType: SOURCE
FileChecksum: SHA1: 20862a6d08391d07d09344029533ec644fac6b21
FileChecksum: SHA256: b4e5ca56d1f9110ca94ed0bf4e6d9ac11c2186eb7cd95159c6fdb50e8db5a823
FileChecksum: MD5: 935054fe899ca782e11003bbae5e166c
LicenseConcluded: GPL-3.0-or-later
LicenseInfoInFile: GPL-3.0-or-later
FileCopyrightText: Copyright (c) 2024 Acme.

...
```

Anatomy of an SDPX file

```
...

Relationship: SPDXRef-hello-binary GENERATED_FROM SPDXRef-hello-src
Relationship: SPDXRef-hello-binary GENERATED_FROM SPDXRef-Makefile
Relationship: SPDXRef-Makefile BUILD_TOOL_OF SPDXRef-Package-hello

...
```



Current SPDX support in Zephyr

1. Create a build directory with CMake file API enabled
2. Build project with “build metadata” enabled
3. Compute SBOM(s)

```
west spdx --init -d BUILD_DIR
west build -d BUILD_DIR -- -DCONFIG_BUILD_OUTPUT_META=y
west spdx -d BUILD_DIR
```

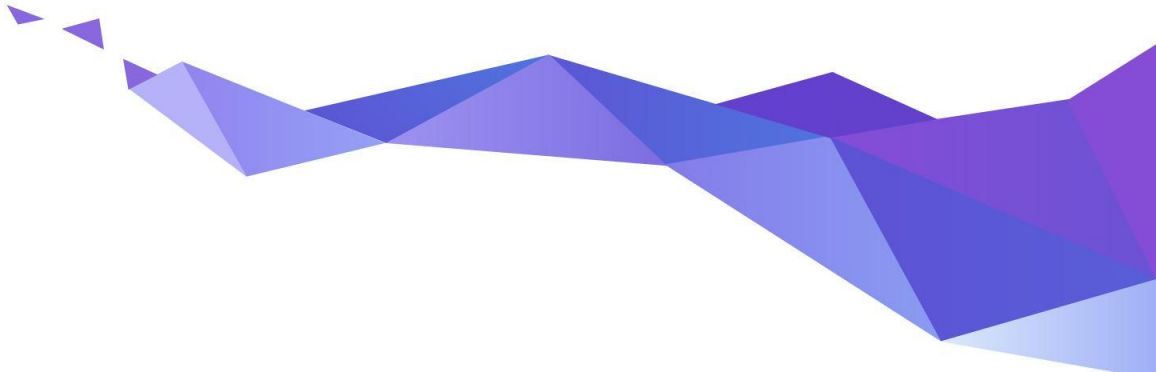
Current SPDX support in Zephyr

- `zephyr.spdx`
  - SBOM for the Zephyr source files actually used by your application
- `app.spdx`
  - SBOM for the source files of your application
- `build.spdx`
  - SBOM for all the build objects, inc. of course your final image

Making sense of the SBOMs?

- Ensure completeness
- Understand licenses
- Find vulnerabilities

Useful SBOM tools



ntia-checker

Zephyr®

NTIA conformance checker

Checks if your SDPX document contains all the "minimum elements” as per NTIA recommendations.

https://github.com/spdx/ntia-conformance-checker

ntia-checker

```
$ ntia-checker --file build/spdx/app.spdx

Is this SBOM NTIA minimum element conformant? False

Individual elements                                | Status
---------------------------------------------------------
All component names provided?                      | True
All component versions provided?                   | False
All component identifiers provided?                | True
All component suppliers provided?                  | False
SBOM author name provided?                         | True
SBOM creation timestamp provided?                  | True
Dependency relationships provided?                 | True
```

sbomqs // SBOM quality score

Assess the quality of your SBOM and how “consumable” it is

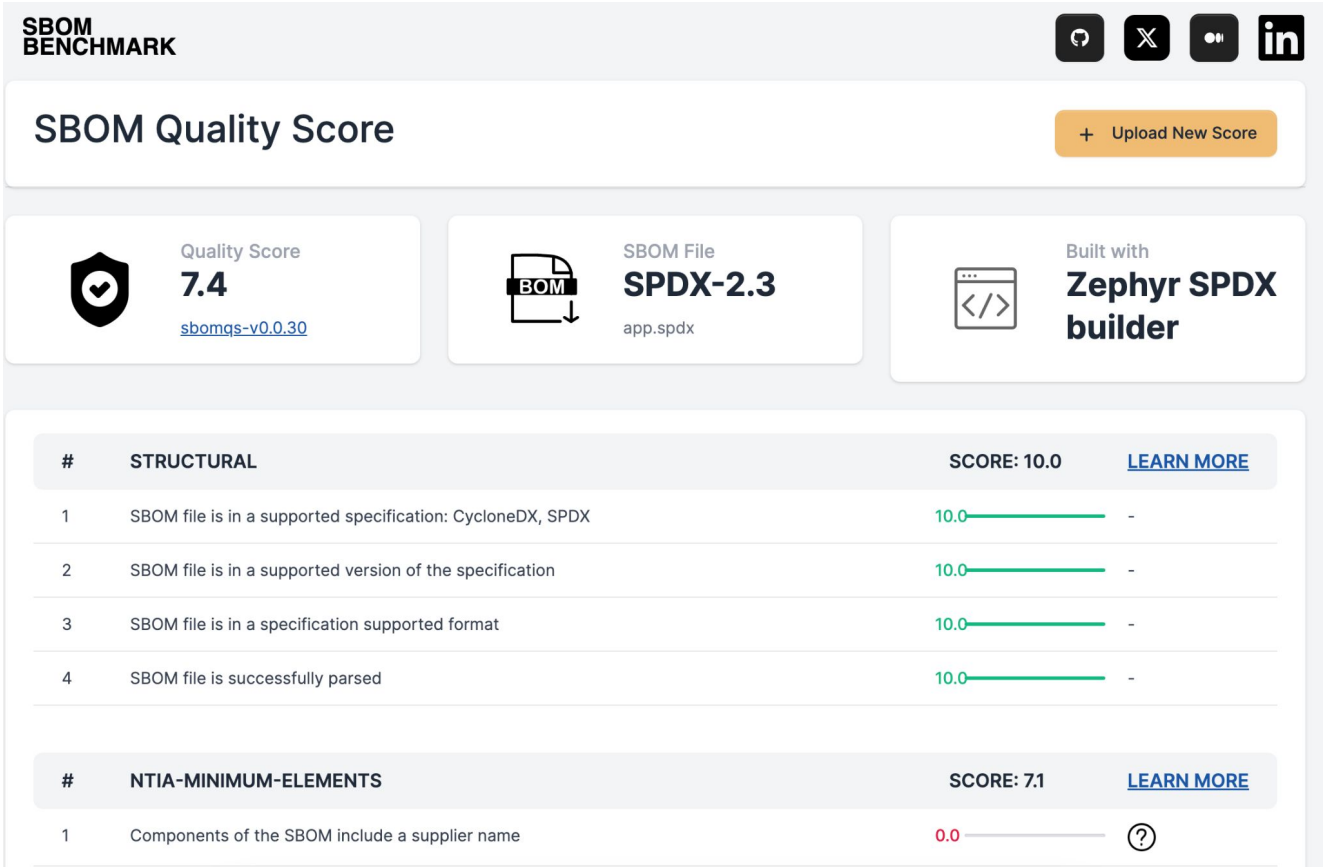
https://github.com/interlynk-io/sbomqs

sbomqs


```
$ sbomqs score build/spdx/build.spdx

SBOM Quality Score:6.3  components:72     build/spdx/build.spdx
+-----------------------+---------------------------------+-----------+---------------------------------+
|       CATEGORY        |            FEATURE              |   SCORE   |              DESC               |
+-----------------------+---------------------------------+-----------+---------------------------------+
| NTIA-minimum-elements | comp_with_name                  | 10.0/10.0 | 72/72 have names                |
+                       +---------------------------------+-----------+---------------------------------+
|                       | comp_with_supplier              | 0.0/10.0  | 0/72 have supplier names        |
+                       +---------------------------------+-----------+---------------------------------+
|                       | comp_with_uniq_ids              | 10.0/10.0 | 72/72 have unique ID's          |
+                       +---------------------------------+-----------+---------------------------------+
|                       | comp_with_version               | 0.0/10.0  | 0/72 have versions              |
+                       +---------------------------------+-----------+---------------------------------+
|                       | sbom_authors                    | 10.0/10.0 | doc has 1 authors               |
+                       +---------------------------------+-----------+---------------------------------+
|                       | sbom_creation_timestamp         | 10.0/10.0 | doc has creation timestamp      |
|                       |                                 |           | 2024-04-16T14:10:47Z            |
+                       +---------------------------------+-----------+---------------------------------+
|                       | ...                                | ...       | ...                                |
```

sbomqs

Zephyr®



gh sbom

Generates an SBOM using information from a Github repository's "Dependency Graph"

Maybe not the most useful for a project such as Zephyr (vs. for example NodeJS projects), but can help surface indirect/unnecessary dependencies

https://github.com/advanced-security/gh-sbom

sbom2doc

Nice summary (in Markdown, text, PDF, or json) of all the components within an SBOM.

```
$ sbom2doc -i build/spdx/modules-deps.spdx
```

https://pypi.org/project/sbom2doc/

cve-bin-tool

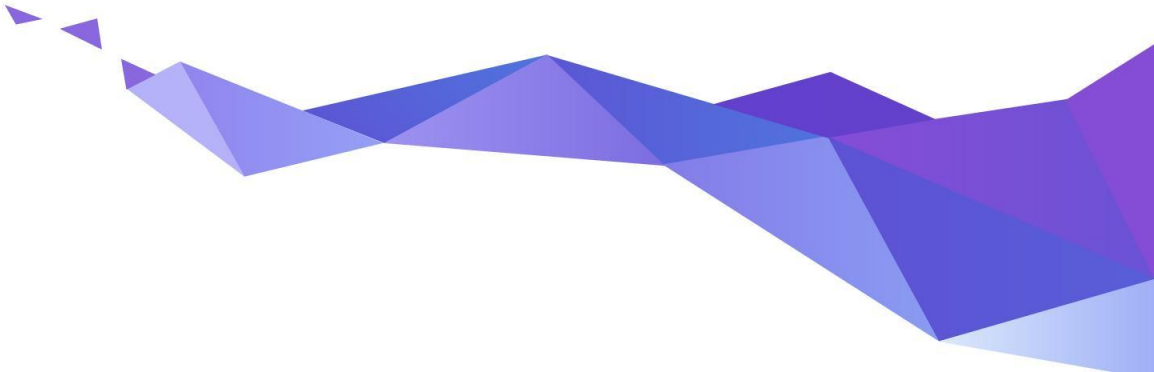
Zephyr®

Checks SBOM against NVD (National Vulnerability Database)

Can also scan binaries

https://github.com/intel/cve-bin-tool

Coming next?



PR #66495 - Adding support for CPE

CPE (Common Product Enumerator) and **PURL** (Package URL) allow to uniquely identify the version of a component

Example:

- `cpe:2.3:a:arm:mbed_tls:3.5.1:*:*:*:*:*:*:*`
- `pkg:github/Mbed-TLS/mbedtls@V3.5.1`

PR #66495 - Adding support for CPE

New metadata in module's `module.yml` to indicate CPE, when applicable.

Example: `modules/crypto/mbedtls/zephyr/module.yml`

```
build:
  cmake-ext: True
  kconfig-ext: True
security:
   external-references:
       - cpe:2.3:a:arm:mbed_tls:3.5.1:*:*:*:*:*:*:*
       - pkg:github/Mbed-TLS/mbedtls@V3.5.1
```

PR #66495 - Adding support for CPE

```
$ cve-bin-tool --sbom spdx \
               --sbom-file build/spdx/modules-deps.spdx
```

CPE SUMMARY

Vendor	Product	Version	Latest Upstream Stable Version	CRITICAL CVEs Count	HIGH CVEs Count	MEDIUM CVEs Count	LOW CVEs Count	UNKNOWN CVEs Count	TOTAL CVEs Count
arm	mbed_tls	3.5.1	3.6.0	0	3	1	0	0	4

NewFound CVEs

Vendor	Product	Version	CVE Number	Source	Severity	Score (CVSS Version)
arm	mbed_tls	3.5.1	CVE-2023-52353	NVD	HIGH	7.5 (v3)
arm	mbed_tls	3.5.1	CVE-2024-23170	NVD	MEDIUM	5.5 (v3)
arm	mbed_tls	3.5.1	CVE-2024-23744	NVD	HIGH	7.5 (v3)
arm	mbed_tls	3.5.1	CVE-2024-23775	NVD	HIGH	7.5 (v3)

SPDX 3.0

- Expand beyond “just” licensing
- **Profiles**. Only ship the information that’s useful for a particular audience/use case.
  - licensing profile: declared licenses, concluded licenses, …
  - build profile: build parameters, build ID, …
  - AI profile: models, datasets, …

https://spdx.github.io/spdx-spec/v3.0/

Help wanted!

- SPDX 3.0 migration efforts
- Capture more data in the SBOMs (e.g. copyright information, ...)
- Share your requirements (Discord, GitHub)

Thanks!
Questions?


zephyrproject.org


github.com/zephyrproject-rtos


chat.zephyrproject.org