SBOMs: Essential for Embedded too!

Kate Stewart, VP Dependable Embedded Systems

#ossummit  @_kate_stewart
● Why the interest in SBOMs?
● What makes up an SBOM?
● How to produce SBOMs for embedded systems?
Challenge:

Are my products affected by $vulnerability$?
Software Supply Chain Attacks are Growing…

Sources: Sonatype 2021 State of the Software Supply Chain
What is the Software Supply Chain?

Set of **software**, **suppliers**, **development processes**, and **distribution processes** used to **create software products**
Cost to Remediate?

Cost to developers to fix is small, cost to users....

Flaws in open source protocols expose millions of embedded devices

Amnesia:33 vulnerabilities could impact numerous industries, from health care to retail and beyond


Cleaning up SolarWinds hack may cost as much as $100 billion

Government agencies, private corporations will spend months and billions of dollars to root out the Russian malicious code

Of organizations surveyed, 95% are concerned about software security.

License: CC-BY-4.0
Of organizations surveyed, 
98% use open source software.
“98% of codebases audited in 2020 contained open source components.

Open source made up 75% of the audited codebases.”

Source: 2021 Synopsys Open Source Security and Risk Analysis Report
Transparency is key to improving supply chain security!

License: CC-BY-4.0
Guidelines for Securing the Internet of Things

This ENISA study defines guidelines for securing the supply chain for IoT. ENISA with the input of IoT experts created security guidelines for the whole lifespan: from requirements and design, to end use delivery and maintenance, as well as disposal. The study is developed to help IoT manufacturers, developers, integrators and all stakeholders that are involved to the supply chain of IoT to make better security decisions when building, deploying, or assessing IoT technologies.

Published: November 09, 2020
Language: English

Regulatory Agencies ...

Healthcare Sector Spearheads SBOM Adoption to Support Cybersecurity

Healthcare is pioneering SBOM adoption due to growing cybersecurity concerns and the FDA’s recent medical device security guidance, the Linux Foundation found.

9 rules imposed by NERC CIP standards

- CIP-001 Sabotage reporting
- CIP-002 Critical cyber asset identification
- CIP-003 Security management controls
- CIP-004 Personnel and training
- CIP-005 Electronic security perimeters
- CIP-006 Physical security of critical cyber assets
- CIP-007 Systems security management
- CIP-008 Incident reporting and response planning
- CIP-009 Recovery plans for critical cyber assets


Source: https://searchcompliance.techtarget.com/definition/NERC-CIP-critical-infrastructure-protection
Executive Order on Improving the Nation’s Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Section 1. Policy. The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people’s security and privacy. The Federal Government must improve its efforts to identify, deter, protect against, detect, and respond to these actions and actors. The Federal Government must also carefully examine what occurred during any major cyber incident and apply lessons learned. But cybersecurity requires more than government action. Protecting our Nation from malicious cyber actors requires the Federal Government to partner with the private sector. The private sector must adapt to the continuously changing threat environment, ensure its products are built and operate securely, and partner with the Federal Government to foster a more secure cyberspace. In the end, the trust we place in our

Source: https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/
● Why the interest in SBOMs?
● What makes up an SBOM?
● How to produce SBOMs for Embedded Systems?
What do SBOM’s look like?

Lots of different forms...

Need to standardize to automate at scale.

Source: https://simplerisk.freshdesk.com/support/solutions/articles/6000230367-what-third-party-software-is-used-by-simplerisk-and-how-is-it-licensed-
Software Bill of Materials (SBOM)

An SBOM is a formal record containing the details and supply chain relationships of various components used in building software.

These components, including libraries and modules, can be open source or proprietary, free or paid, and the data can be widely available or access-restricted.

Source: NTIA's SBOM FAQ
What does the SBOM represent?

Source: NTIA’s Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)
Challenge: Finding vulnerability in complex software

- Need to search the full dependency graph
- Not all dependencies are the same, there are different dependency types (test, build tools, static linked …)
- No single “UPC code” to clearly identify packages

Source: [https://deps.dev/maven/org.spdx%3Atools-java/1.0.2/dependencies/graph](https://deps.dev/maven/org.spdx%3Atools-java/1.0.2/dependencies/graph)
Producing & Consuming SBOMs

upstream / supplier

Open Source Projects  →  3rd Party

Development → Build → Test → Packaging → Policy Checks → Release/Operate

downstream / consumer

Open Source Projects  →  Customers

(Images and text content)
Challenge: Scaling up to the Open Source Ecosystem

Remote code injection in Log4j

Summary

<table>
<thead>
<tr>
<th>TOTAL PACKAGES AFFECTED</th>
<th>PACKAGES WITH A KNOWN FIX</th>
<th>TOTAL ECOSYSTEM AFFECTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>36.28k</td>
<td>8.97k</td>
<td>8.27%</td>
</tr>
</tbody>
</table>

Affected Version: < 2.15.0

Source: https://deps.dev/advisory/GHSA/GHSA-jfh8-c2jp-5v3q on 2022/1/10
Industry Awareness of SBOM is emerging...

What is your organization’s familiarity with a software bill of materials (SBOM)?

- Never heard of the term: 7%
- Heard of the term but I’m not familiar with it: 8%
- Somewhat familiar with the term: 16%
- Familiar with the term: 35%
- Very familiar with the term: 31%
- Don’t know or not sure: 2%

Sample Size = 361

© 2021 The Linux Foundation
Doing this at scale requires standards and tools!

source: https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/
What is a Minimum SBOM?

<table>
<thead>
<tr>
<th>Minimum Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Fields</strong></td>
</tr>
<tr>
<td>Document baseline information about each component that should be tracked: Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, and Timestamp.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Automation Support</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Support automation, including via automatic generation and machine-readability to allow for scaling across the software ecosystem. Data formats used to generate and consume SBOMs include SPDX, CycloneDX, and SWID tags.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Practices and Processes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Define the operations of SBOM requests, generation and use including: Frequency, Depth, Known Unknowns, Distribution and Delivery, Access Control, and Accommodation of Mistakes.</td>
</tr>
</tbody>
</table>

What is the Minimum SBOM?

<table>
<thead>
<tr>
<th>Data Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier Name</td>
<td>The name of an entity that creates, defines, and identifies components.</td>
</tr>
<tr>
<td>Component Name</td>
<td>Designation assigned to a unit of software defined by the original supplier.</td>
</tr>
<tr>
<td>Version of the Component</td>
<td>Identifier used by the supplier to specify a change in software from a previously identified version.</td>
</tr>
<tr>
<td>Other Unique Identifiers</td>
<td>Other identifiers that are used to identify a component, or serve as a look-up key for relevant databases.</td>
</tr>
<tr>
<td>Dependency Relationship</td>
<td>Characterizing the relationship that an upstream component X is included in software Y.</td>
</tr>
<tr>
<td>Author of SBOM Data</td>
<td>The name of the entity that creates the SBOM data for this component.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Record of the date and time of the SBOM data assembly.</td>
</tr>
</tbody>
</table>

Of organizations surveyed, 47% are using SBOMs today.
Based on organizations surveyed, it’s forecasted 78% will use SBOMs in 2022.

Derived from: NTIA's \textit{Survey of Existing SBOM Formats and Standards}
Source SBOM - software sources used to build an executable image.

Analyzed SBOM - executable image to be integrated into deliverable. Created from 3rd party heuristics.

Build SBOM - List of components and relationships between dependent components assembled to create a product released from Supplier.

Deployed SBOM - Tracking configuration options on how a product has been deployed by User.
● Why the interest in SBOMs?
● What makes up an SBOM?
● How to produce SBOMs for embedded systems?
Embedded Projects Generating SBOMs

Zephyr’s west spdx

Presentation / Demo:
https://www.youtube.com/watch?v=KYC3YpSu9zs

Yocto builds

Presentation / Demo:
https://www.youtube.com/watch?v=y0N4FnkwTOY
Case Study: Zephyr

- **Open source** real time operating system
- **Vibrant Community** participation
- Built with **safety and security** in mind
- **Cross-architecture** with broad SoC and development board support.
- **Vendor Neutral** governance
- **Permissively** licensed - Apache 2.0
- **Complete**, fully integrated, highly configurable, **modular** for flexibility
- **Product** development ready using LTS includes security updates
- **Certification** ready with Auditable
Example of Products Running Zephyr Today

- Proglove
- RUUVI Tag
- Distancer
- Keeb.io BDN9
- Hati-ACE
- Oticon More
- Intellinium Safety Shoes
- GNARBOX 2.0 SSD
- Anicare Reindeer Tracker
- Safety Pod
- BLiXT solid state circuit breaker
- Moto Watch 100
- Point Home Alarm
- Rigado IoT Gateway
- HereO Core Box
- Sentrius
- See.Sense AIR
- Vestas Wind Turbine
One Line Config Change to Create SBOMs!

Source SBOM

Build SBOM

Learn more at: https://www.youtube.com/watch?v=KYC3YpSu9zs
SBOMs Included By Default ... Automatically

Case Study: Yocto & OpenEmbedded

Yocto Project
- Linux Foundation project
- Poky reference distribution
- Runs QA tests
- Manages release schedule
- Provides funding for personnel
- Documentation

OpenEmbedded
- Community project
- OpenEmbedded core layer
- Build system (bitbake)

Poky
Yocto Project Open Source Reference Embedded Distribution

Open Source Build Engine and YP-Compliant Metadata for Embedded Linux

YOCTO PROJECT (YP)
Umbrella Open Source Project that Builds and Maintains Validated Open Source Tools and Components Associated with Embedded Linux
Simplified Build Flow

Host Tools

Source

Recipe Metadata

SHA256

Native tools & Cross Compiler

Target Packages

Recipe Metadata

SHA256

Target Image

Recipe Metadata

SHA256

Recipe

Metadata

Source

SHA256

Copyright 2022 Joshua Watt, Creative Commons BY-SA 4.0 International License
Recipe Metadata

Recipes already contain much of the data desired in a SBOM

- Version
- Source code URL
- Licenses
- Build time dependencies
- Run time dependencies
- CVEs patched
- Source Files
- Package Files
- ...

All of this information is authoritative (no guessing)
SPDX SBOM Generation with one line config change

Host Tools

Recipe Metadata

Source

Native tools & Cross Compiler

SPDX

Recipe Metadata

Target Packages

SPDX

Recipe Metadata

Target Image

SPDX

SPDX Archive

Copyright 2022 Joshua Watt, Creative Commons BY-SA 4.0 International License
SBoM Relationships

- **Package SPDX**
  - **Runtime SPDX**
    - **Image SPDX**
  - **Package Files**
  - **Recipe SPDX**

  - **Image JSON Index**

Chain of Ownership:
- **Package SPDX** to **Runtime SPDX** via **GENERATED_FROM** (recipe)
- **Runtime SPDX** to **Package SPDX** via **AMENDS**
- **Package SPDX** to **Recipe SPDX** via **BUILD_DEPENDENCY_OF(*)**
- **Recipe SPDX** to **Source Code** via **CONTAINS**
- **Recipe SPDX** to **Package SPDX** via **GENERATED_FROM** (debug source)

Relationships:
- **RUNTIME_DEPENDENCY_OF(*)**: Package SPDX to Runtime SPDX
- **CONTAINS**: Runtime SPDX to Image SPDX
- **OTHER**: Image JSON Index to Image SPDX
- **GENERATED_FROM**: Package Files to Package SPDX
- **BUILD_DEPENDENCY_OF(*)**: Recipe SPDX to Package SPDX
- **GENERATED_FROM**: Package SPDX to Recipe SPDX (debug source)
- **CONTAINS**: Recipe SPDX to Source Code
What can we generate SPDX SBOM documents for?

TL; DR - Anything we can build

- "On target" C/C++/Fortran etc. ✅
- "Host" build tools ✅
- Linux Kernel ✅
- Target images ✅
- SDKs ✅
- Container Images ✅
- VM Images ✅
- Rust 🚧
- Go 🚧

To learn more about Yocto & SBOM generation see:
https://www.youtube.com/watch?v=8X5PWa7A6pY
Yocto SPDX Features

- Declared License
  - With License Text if not a known SPDX license
- Homepage URL
- Download URL(s)
- CVEs fixed
- CPE
- Summary
- Description
- Source File Listing with Checksums
- Source file SPDX licenses
- Packages
- Package files with Checksums

- Package file GENERATED_FROM (from debug data)
- Build time dependencies
- Runtime dependencies
- Source code archive for analysis by other tools (e.g. Fossology)
SPDX: Embedded Focus from the Start

🎵 Born out of a need to exchange OSS component and license information - components and dependencies are the underpinning!

🎂 Recently celebrated its 10th birthday - 10 years of use cases produced rich set of dependency relationships.

🎉 SPDX 2.2 became ISO standard in 2021 (ISO/IEC 5962:2021)

Open weekly working group and monthly general meetings. Contributors welcome!
SPDX Relationships Connect Dependencies Between Components and Between SBOMs

<table>
<thead>
<tr>
<th>DESCRIBES</th>
<th>DEPENDENCY_OF</th>
<th>PREREQUISITE_FOR</th>
<th>GENERATES</th>
<th>VARIANT_OF</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIBED_BY</td>
<td>RUNTIME_DEPENDENCY_OF</td>
<td>HAS_PREREQUISITE</td>
<td>TEST_OF</td>
<td>FILE_ADDED</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>BUILD_DEPENDENCY_OF</td>
<td>ANCESTOR_OF</td>
<td>TEST_TOOL_OF</td>
<td>FILE_DELETED</td>
</tr>
<tr>
<td>CONTAINED_BY</td>
<td>DEV_DEPENDENCY_OF</td>
<td>DESCENDENT_OF</td>
<td>TEST_CASE_OF</td>
<td>FILE_MODIFIED</td>
</tr>
<tr>
<td>DYNAMIC_LINK</td>
<td>OPTIONAL_DEPENDENCY_OF</td>
<td>DOCUMENTATION_OF</td>
<td>EXAMPLE_OF</td>
<td>PATCH_FOR</td>
</tr>
<tr>
<td>STATIC_LINK</td>
<td>PROVIDED_DEPENDENCY_OF</td>
<td>BUILD_TOOL_OF</td>
<td>METAFILE_OF</td>
<td>PATCH_APPLIED</td>
</tr>
<tr>
<td>AMENDS</td>
<td>TEST_DEPENDENCY_OF</td>
<td>EXPANDED_FROM_ARCHIVE</td>
<td>PACKAGE_OF</td>
<td>OTHER</td>
</tr>
<tr>
<td>COPY_OF</td>
<td>OPTIONAL_COMPONENT_OF</td>
<td>DISTRIBUTION_ARTIFACT</td>
<td>DATA_FILE_OF</td>
<td></td>
</tr>
<tr>
<td>DEPENDS_ON</td>
<td>DEPENDENCY_MANIFEST_OF</td>
<td>GENERATED_FROM</td>
<td>DEV_TOOL_OF</td>
<td></td>
</tr>
</tbody>
</table>

For more details see: [https://spdx.github.io/spdx-spec/relationships-between-SPDX-elements/](https://spdx.github.io/spdx-spec/relationships-between-SPDX-elements/)
ISO/IEC 5962:2021

- SPDX is became an ISO/IEC standard in 2021!
- Specification is freely available from ITTF site
- Future updates are live tracked at: https://spdx.github.io/spdx-spec
- For more information on participating in next revisions see https://spdx.dev/
I.1 Differences between V2.3 and V2.2.2

V2.3 has added new fields to improve the ability to capture security related information and to improve interoperability with other SBOM formats.

Key changes include:

- Added fields to Clause 7 (Package Information) to describe “Primary Package Purpose” and standardize recording of “Built Date”, “Release Date”, “Valid Until Date”.
- Added hash algorithms (SHA3-256, SHA3-384, SHA3-512, BLAKE2b-256, BLAKE2b-384, BLAKE2b-512, BLAKE3, ADLER32) to the set recognized by 7.10 (Package Checksum field) and 8.4 (File checksum field).
- Update Clause 7, 8, and 9 to make several of the licensing properties optional rather than requiring the use of “NOASSERTION” when no value is provided.
- Update Clause 11 to add the new relationship types: REQUIREMENT_DESCRIPTION_FOR and SPECIFICATION_FOR.
- Update Annex B (License matching guidelines and templates) to use the License List XML format.
- Update Annex F (External Repository Identifiers) to expand security references to include advisory, fix, URL, SWID. Expand persistent identifiers to include gitoid.
- Update Annex G (SPDX Lite Profile) to include NTIA SBOM mandatory minimum fields as required.
- Update Annex H to documented how the snippet information in files to be consistent with REUSE recommendations.
- Added Annex K (How To Use SPDX in Different Scenarios) to illustrate linking to external security information, and illustrate how the NTIA SBOM mandatory minimum elements map to SPDX fields.
To Learn More:

Software Bill of Materials (SBOM) Guidance:
- https://www.ntia.gov/SBOM
- https://www.cisa.gov/sbom

Zephyr SBOM generation:
- Documentation: Generate Software Bill of Materials with West
- Presentation: Generating SBOMs for IoT at Build Time
- Presentation: Zephyr Developer Summit see: https://sched.co/10CF4

Yocto SBOM generation:
- Documentation: create-spdx in Release 3.4 (honister)
- Presentation: Software Supply Chain with the Yocto Project
- Presentation: SPDX in the Yocto Project

SPDX:
- https://spdx.dev/
- Documentation: https://spdx.github.io/spdx-spec/v2.3