SESSION ID: DSO-M03

Tooling Up – Getting SBOMs to scale

Allan Friedman
Senior Advisor & Strategist
CISA
@allanfriedman

Kate Stewart
VP, Dependable Embedded Systems
Linux Foundation
@_kate_stewart
Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

©2022 RSA Conference LLC or its affiliates. The RSA Conference logo and other trademarks are proprietary. All rights reserved.
Paying attention or looking for lunch?

- Remind me what an SBOM is again?
- Motivation
- Tooling taxonomies
- Challenges & open questions for SBOM automation
- Next steps for the tooling ecosystem
- What you can do
SBOMs provide transparency in the SW market
SBOMs provide transparency in the SW market
So what’s an SBOM again?
A Software Bill of Materials (SBOM) is effectively a list of ingredients or a nested inventory. It is “a formal record containing the details and supply chain relationships of various components used in building software.”
Motivation

- We’re ready for SBOM!
  - Many of you want this.
  - Some of you will have to do it...
  - See: Executive Order 14028*

- Doing this at scale requires tools

*https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/
Based on organizations surveyed, it’s forecasted 78% will use SBOMs in 2022.

Of organizations surveyed, 98% use open source software.

Goal:
Supporting an accessible, competitive marketplace

- A resource where tool providers can list themselves
- A resource where those looking for tools can find them
- A fair marketplace with transparent governance
- Standard/format neutral
- Includes open source and proprietary solutions
- Welcoming of novel solutions over time
SBOMs in the lifecycle of software

Source: NTIA's Survey of Existing SBOM Formats and Standards
Source SBOM - software sources imported used to build binary executable image.

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

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

Deployed SBOM - Tracking configuration options on how a product has been deployed by User.
# Minimum SBOM

<table>
<thead>
<tr>
<th>Data Fields</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation Support</td>
<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>
<tr>
<td>Practices and Processes</td>
<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 about the Tools?
# Taxonomy for Classifying SBOM Tools

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produce</td>
<td>Build</td>
<td>SBOM is automatically created as part of building a software artifact and contains information about the build</td>
</tr>
<tr>
<td></td>
<td>Analyze</td>
<td>Analysis of source or binary files will generate the SBOM by inspection of the artifacts and any associated sources</td>
</tr>
<tr>
<td></td>
<td>Edit</td>
<td>A tool to assist a person manually entering or editing SBOM data</td>
</tr>
<tr>
<td>Consume</td>
<td>View</td>
<td>Be able to understand the contents in human readable form (e.g. picture, figures, tables, text.). Use to support decision making &amp; business processes</td>
</tr>
<tr>
<td></td>
<td>Diff</td>
<td>Be able to compare multiple SBOMs and clearly see the differences (e.g. comparing two versions of a piece of software)</td>
</tr>
<tr>
<td></td>
<td>Import</td>
<td>Be able to discover, retrieve, and import an SBOM into your system for further processing and analysis</td>
</tr>
<tr>
<td>Transform</td>
<td>Translate</td>
<td>Change from one file type to another file type while preserving the same information</td>
</tr>
<tr>
<td></td>
<td>Merge</td>
<td>Multiple sources of SBOM and other data can be combined together for analysis and audit purposes</td>
</tr>
<tr>
<td></td>
<td>Tool support</td>
<td>Support use in other tools by APIs, object models, libraries, transport, or other reference sources</td>
</tr>
</tbody>
</table>

Other ways of classifying SBOM tools
Other ways of classifying tools

- Generation vs. Consumption
- By the Lifecycle of software

- Technical ecosystem
- Sector-specific tools
- Open source vs proprietary
- First party / third party
- Data management and configuration management
  - Tracking what has been updated
  - Also an asset management story
Generation made easy – single line cmds

- Docker
Generation made easy – single line cmds

- Docker

```bash
$ docker sbom neo4j:4.4.5
Syft v0.43.0
✓ Loaded image
✓ Parsed image
✓ Cataloged packages [385 packages]

<table>
<thead>
<tr>
<th>NAME</th>
<th>VERSION</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bsdutils</td>
<td>1:2.36.1-8+deb11u1</td>
<td>deb</td>
</tr>
<tr>
<td>ca-certificates</td>
<td>20210119</td>
<td>deb</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log4j-api</td>
<td>2.17.1</td>
<td>java-archive</td>
</tr>
<tr>
<td>log4j-core</td>
<td>2.17.1</td>
<td>java-archive</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Generation made easy – single line cmds

- Docker
- Yocto
- Zephyr
- ... more coming

New production tools are emerging daily, but challenge is organizations need a place to find them, and find the right type of tool for the task!
Consumption tools

Data → Intelligence → Action
Consumption tools

• Simple use case: detecting vulnerabilities
  – Grep NVD
  – Map to other sources of data
  – Entity disambiguation

• Integration into existing security tools
  – Asset management
  – Vulnerability management
  – CMDB
  – Data Lake

Data → Intelligence → Action
Consumption tools

- Simple use case: detecting vulnerabilities
  - Grep NVD
  - Map to other sources of data
  - Entity disambiguation
- Integration into existing security tools
  - Asset management
  - Vulnerability management
  - CMDB
  - Data Lake

Tools starting to do this:
- SW360
- OWASP DependencyCheck
- Daggerboard (coming soon!)

… and of course, commercial offerings!
Challenges & Open Questions for Automation
Delivering SBOMs: discovery and access
Plumbing

- **All infosec problems eventually become data management problems**
- How will I get my SBOMs
- How to store our piles of SBOMs?
- How do we find the relevant info in our SBOMs?
- Integration into existing data flows

Photo by jiawei cui: https://www.pexels.com/photo/chrome-pipe-lines-2310904/
Software by any other name...

“There are only two hard things in Computer Science: cache invalidation and naming things.”

- attributed to Phil Karlton
Challenge: Vulnerability vs Exploitability

Solution: “Vulnerability Exploitability eXchange” (VEX)
SBOM – to include vulnerabilities or not?

- Tooling across organizations: how to keep data current?
- Mapping VEX documents to SBOMs and other data
- Tools for VEX creation and consumption
  - Early days: https://secvisogram.github.io/
- Work flows for lifecycle
  - E.g. – VEX documents replace other VEXes.
  - E.g. - Do earlier VEX docs apply to later products?
Where to find more info on tools?

**CycloneDX:** [cyclonedx.org/tool-center/](cyclonedx.org/tool-center/)  
or [tiny.cc/CycloneDX](tiny.cc/CycloneDX)

**SPDX:** [spdx.dev/resources/tools/](spdx.dev/resources/tools/)  
or [tiny.cc/SPDX](tiny.cc/SPDX)

**SWID:** [tiny.cc/SWID](tiny.cc/SWID)

Need to see a summary in a neutral location that is Standard/Format Neutral to allow a more open process and wider set of visible reviews.

- Anyone can nominate tool to be added to a list
- Point to evidence of producing, consuming or transforming of SBOM documents to get tool on the list (this includes participating in Plugfest)
Translating Between SBOM Formats and Filetypes

- SwiftBOM: (SPDX(.spdx), SWID(.xml), CycloneDX(.xml,.json))
  - Demo at: https://democert.org/sbom/
  - Source code at: https://github.com/CERTCC/SBOM/tree/master/sbom-demo

- SPDX online tools: (SPDX (.spdx, .json, .yaml, .rdf, .xml, .xls) )
  - Demo at: https://tools.spdx.org/app/
  - Source code at: https://github.com/spdx/spdx-online-tools

- CycloneDX CLI: (CycloneDX (.xml, .json), SPDX(.spdx))
  - Source code at: https://github.com/CycloneDX/cyclonedx-cli
Next steps for the tooling ecosystem

• Join the “Tooling & Implementation” work stream through CISA
  • July 13, 2022 – 3:00-4:30pm ET
  • July 21, 2022 – 9:30-11:00am ET
  • Sign up: SBOM@cisa.dhs.gov

• “Plugfests” to be announced

• **Case studies** of organization adoption of tools & reference tooling workflows
What can your organization do?

- **Next week**: Understand origins of software your organization is using
  - Commercial - can you ask for an SBOM?
  - Open Source - do you have an SBOM for the binary or sources you’re importing?

- **Three months**: Understand what SBOMs your customers will require
  - Expectations - which Standards, dependency depth, licensing info?

- **Six months**: Prototype and Deploy
  - Implement SBOM through using an OSS tool and/or starting conversation with vendor

*If your organization think this is important enough to help:*

- Participate in ongoing discussions to determine best practices for ecosystem
- Contribute to open source project any code developed to support
What can you do?

- **Next week**: start playing with an Open Source SBOM tool and apply it to a repo
- **Three months**: Have an SBOM strategy that explicitly identifies tooling needs
- **Six months**:
  - begin SBOM implementation through using an OSS tool or starting conversation with vendor
  - Participate in a Plugfest, and try to consume another’s SBOM

*If you think this is important enough to help:*

- Tools exist, both open source and commercial. Make sure the ones you find most useful are listed.
- Work with the tools to help harden them, test and report bugs, push them to scale