SBOM Use Cases
Hannah Pearson-Kleinheider
Overview

- Use cases
  - What questions do you want to answer?
  - How will SBOMs provide value to your organization?

- Considerations for different use cases
  - Level of detail or “depth”
  - What information to gather
  - SBOM format and fields used

- Example: CyTRICS use of SBOMs
So … what are SBOMs good for?

- Supply Chain security
- Vulnerability management
- Risk assessment
- License management
- … and more!
SBOM Depth in General

- Varies depending on use case
- From most-to-least detail required:
  - Supply Chain security
  - Vulnerability management
  - Risk assessment
  - License management
- Why?
  - It all depends on what questions you want to answer …
Considerations for SBOM Depth

- Most important aspect: **accuracy**
  - BOM needs to be correct in order to be useful
  - Completeness (more depth and detail) is desirable, but never at the cost of accuracy

- Try exercises testing your ability to obtain answers from your BOMs
  - “Suppose some instances of component XYZ are compromised. What information is needed to identify the compromised components?”
Example: CyTRICS

- **Use cases**
  - Supply Chain security
  - Vulnerability assessment

- **Either way, a high degree of detail is required!**
  - Ability to identify compromised components
  - Find common components across an industry
    - Which components are sufficiently widespread that a single vulnerability could have significant impact?
  - Connect known vulnerabilities in components to systems that use those components
Final Thoughts

- Let your use case inform your data collection
- SBOMs are a tool that provides business value
  - Incident response
  - Risk assessment and mitigation
  - Customer support
  - Patching and vulnerability mitigation decisions
- Start using SBOMs now so you have them when you need them!
Questions?

hannah.kleinheider@inl.gov
STAMP: Software Trace of a Manufacturing Process/Product

Dr. Gabriela Ciocarlie, Dr. Dongyan Xu, Dr. Ananth Grama, Michael Focosi

Nov 15th, 2023
Preliminaries: The Software Bill of Materials (SBOM)

• A Software Bill of Materials (SBOM) is a formal record containing the details and supply chain relationships of various components used in building software.

• Section 10(j) of EO 14028 defines an SBOM as 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.gov/sbom]
The primary purpose of an SBOM is to uniquely and unambiguously identify components and their relationships to one another.

Baseline information may be augmented with several other fields to enable application-specialization.

[Source: ntia.gov/sbom]
Preliminaries: SBOMS and Software Lifecycle

Preliminaries: SBOM Tools

- A number of tools provide support for identifying software entities and conveying associated metadata (producing, consuming, and transforming software entity metadata)
  - CycloneDX (https://cyclonedx.org/)
  - Software Package Data Exchange (SPDX, https://spdx.github.io/spdx-spec/v2.3/)
  - Supply Chain Levels for Software Artifacts (SLSA, https://slsa.dev/)

Important Note: “SBOMs and the improved transparency that they are meant to provide for federal acquirers are a complementary, not substitutive, capability. Federal acquirers that are unable to appropriately ingest, analyze, and act on the data that SBOMs provide will likely not improve their overall C-SCRM posture.” EO 14028
Preliminaries: SBOM Benefits

SBOMs are primarily used by three primary groups:

- **Software developers** use SBOMs to assist in the building and maintenance of their software, including upstream components.

- **Software procurers** use SBOMs to inform pre-purchase assurance, negotiate discounts, or plan implementation strategies.

- **Software users/operators** use SBOMs to inform vulnerability management and asset management, to manage licensing and compliance, and to quickly identify software or component dependencies and supply chain risks.
The Manufacturer Disclosure Statement for Medical Device Security (MDS) provides medical device manufacturers with a means for disclosing to healthcare providers the security-related features of their medical devices.

- The SBOM section of the MDS was created with these parallel efforts in mind.

OpenC2 is a standardized language for the command and control of cybersecurity.

- OpenC2 has commands for obtaining the SBOM of a device, for analyzing the SBOM, and for taking appropriate actions based on the analysis (e.g. connect, patch, sandbox, or block).

Manufacturer Usage Descriptions (MUD) describe IoT devices, their capabilities, and their needs.

- An extension to those descriptions can inform local deployments on how to find an SBOM by pointing to a URL, indicating appropriate local mechanisms, or indicating a point of contact for further information.

DBOM is a common backbone for attestation sharing including data such as SBOMs among supply chain partners.
CyManII’s Cyber-Physical Passports

- Collect **provenance information** across the automation and supply chain networks with **non repudiation**

- Provide a **multiple-perspective** view of manufacturing processes and products across **cyber, physical, and energy** dimensions

- Enable **supply chain verification, analytics, and integrity checks** with privacy-preserving policies in place

Offer the “know how/when/where” of manufacturing products
The CPP follows a product along its lifecycle.
Passports have the same schema but capture different relevant information.

Contains trade secrets or commercial or financial information that is privileged or confidential and exempt from public disclosure.

Do not copy, cite, or distribute without permission of the author.
From CPP to STAMP...
STAMP: Software Trace of a Manufacturing Process/Product

- A STAMP or its associated part is a complete trace of all software components that impact the function or structure of a manufactured part, process, or operation.

- A STAMP is substantially more powerful than an SBOM, since the latter is defined for a software object, whereas the former is defined for a physical artifact.

This difference has significant implications for how STAMPs are constructed, manipulated, and analyzed.

SBOMs are typically (small) parts of STAMPs.
STAMP vs. SBOM

• An SBOM is defined for a software object, therefore all provenance links manifest in the object (either as code or library calls)

• A STAMP is defined for a physical artifact, process, or operation. The artifact may itself have software components for which the SBOM describes the software provenance

• The STAMP of a physical artifact also includes all software used in its design, design compilation, the software controlling the machine used to manufacture the part, etc.

• Since a part may be an assembly of sub-parts, the STAMP of a part includes references to STAMPs associated with its subparts

Linkages in SBOMs are explicit in the software target, while for STAMPs are established across physical objects
STAMPS: Construction and Data Structures

- Since linkages are established through part/process/operational composition, the core data structure must establish a linkage between an artifact and its STAMP, along with all of the constituent subparts.
  - The core data structure can then be used to construct the complete STAMP for a given part.
- **STAMPS in our system are also represented as CPP passports** (in addition to all of the physical artifacts).
- Our CPP system also provides **linkage across parts and sub-parts**, and can therefore be used to construct complete STAMPs.
STAMPS: Uses and Benefits

- Since STAMPs provide a comprehensive software trace of an artifact, they can be used to analyze the impact of software on the structure and function of a given part
  - This includes security as well as design flaws
- A direct application of STAMPs is in root-cause analysis of an observed structural or functional anomaly in a part
- Since STAMPs impact structure and function, they can be associated with functional digital twins or structural part descriptions to enable complex analyses
Use Case: Controller on a turbine in operation

• A STAMP for a turbine links to the STAMP of its constituent blades, even though there may be no software link between the controller for the turbine and any software associated with a blade.

• An SBOM captures a subset of the STAMP information, focusing more on software linkage, while STAMP provides additional process- or product-based linkage.
Supply Chain Transport and Tree Visualization with Software Passports
Thank you!

gabriela.ciocarlie@cymanii.org
dongyan.xu@cymanii.org
ananth.grama@cymanii.org