Sharing and Exchanging SBOMs

Draft version 0.2
2020-10-22

This document is available at:
https://docs.google.com/document/d/1XuGix4AIcXKqPlPVMvjQEj7zybvnYy4DkgpYfF2EwRc

Overview

Transparency in the supply chain enables better risk decision-making for suppliers and users of software. This means that information about the underlying software components in a piece of software—a Software Bill of Material (SBOM)—should be accessible to the right entities at the right time. An SBOM identifies and lists software components used in building a piece of software, as well as information about those components, and the relationships between them. An SBOM is effectively a nested inventory, or a list of ingredients that make up software components.[1]

Deleted: producers

Deleted: consumers

Sharing this SBOM data across the supply chain will involve a combination of technical platforms, predictable data formats, and operational processes. There will not be a single one-size-fits-all solution across the diverse needs of the entire software ecosystem, but modeling SBOM processes on existing approaches and methods will simplify this process, and minimize the amount of new tools and processes needed for better supply chain management.

Goal

The goal of this document is to provide a small set of options for discovery and access of SBOM data as architectural building blocks to allow flexibility for different use cases while trying to minimize the burden on diverse producers and consumers of SBOM data. SBOMs may be delivered in any number of ways based on standardized automated exchange mechanisms or through mechanisms specified by prearranged agreement such as a contract. This overview focuses primarily on standardized approaches.

This document deals with getting the SBOM from the upstream author to the downstream consumer of the SBOM. It does not deal with creating the SBOM (see *Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)*[2]) nor does it deal with how the receiver uses the SBOM (see *Roles and Benefits for SBOM Across the Supply Chain*[3]).

[1] For more information about the basics of a Software Bill of Materials, see www.ntia.gov/SBOM
[2] https://www.ntia.gov/files/ntia/publications/framingsbom_20191112.pdf
[3] https://www.ntia.gov/files/ntia/publications/ntia_sbom_use_cases_roles_benefits-nov2019.pdf

Terminology

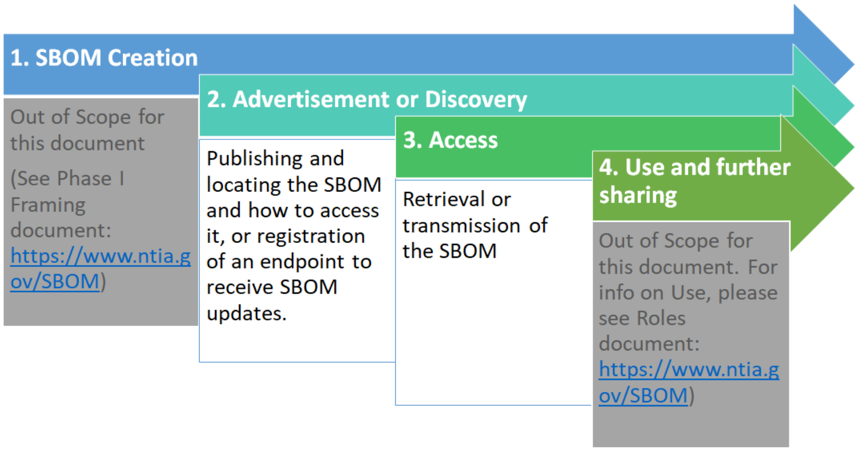
To best understand how to interpret the information provided in this document, it is important to understand the terms and how they are used in context. See *Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)*[2] for a full list of NTIA Software Transparency terms and their definitions.

Table 1: General terminology

Term	Meaning
Author	the creator of the SBOM, even if there are several intermediaries (e.g. mid-stream developer) involved
Consumer (this is sometimes called a “receiver”, “operator” or “leaf entity”)	represent the recipient of the SBOM transfer
Upstream	towards the author
Downstream	towards the consumer
Mid-stream developer	intermediaries who make use of upstream components which they combine into something delivered downstream
Discovery	the mechanism used by the consumer to know the SBOM exists and how to access it
Access	transfer of the SBOM using the method derived from discovery

Getting SBOM data to the right people at the right time consists first of knowing the SBOM exists and how to access it. “Discovery” is used to describe the mechanism used by the consumer to know the SBOM exists and how to access it. One key goal is automated SBOM discovery. We use the term advertisement to mean the mechanism by which the author makes known how the consumer may access the SBOM, e.g., either through a well known location or through an announcement of some form.

The overall process consists of:


1. SBOM Creation
Out of Scope for this document
(See Phase I Framing document: https://www.ntia.gov/SBOM)
2. Advertisement or Discovery
Publishing and locating the SBOM and how to access it, or registration of an endpoint to receive SBOM updates.
3. Access
Retrieval or transmission of the SBOM
4. Use and further sharing
Out of Scope for this document. For info on Use, please see Roles document: https://www.ntia.gov/SBOM)

Advertisement And Discovery

Advertisement is how an author or a device informs consumers that an SBOM is available and discovery is how the consumer learns of the information. At least one or the other is used as a means to locate one or more SBOMs.

Example 1: URL

When an SBOM is to be used in an operational deployment, it might be included as a URL in product literature or packaging, or as part of a Manufacturer Usage Description (MUD) [RFC8520][4]. MUD provides a means for devices to describe their capabilities and needs for deployments. A MUD extension for SBOM provides a choice for one or more ways to retrieve the SBOM. An SBOM could also be searched for in a search engine.

An SBOM could also be retrieved from the device itself, via ".well-known" URL [RFC8615]. This would require a short RFC to define the desired URL, and possibly to register MIME types for any SBOM formats that do not yet have a definition.

Example 2: Manifest

When an SBOM is to be used by downstream developers, the software package could include a manifest in a well known location. For instance, one could imagine SBOM.{fmt} in a top-level directory of a software repository. This form is well suited for distributions for use by parties in the middle of the supply chain to indicate licensing requirements and package contents. This is useful for including the generated SBOM in package management tools.

Commented [1]: Isn't this more of an example of a URL?

ie. an example of my reading of "top-level directory of a software repository" would be https://github.com/WordPress/WordPress/tree/5.5.1/sbom.{fmt}

An example of a manifest might be:

* included in a well known layer in an OCI image
* included in a well known location in a package (deb, rpm, npm, etc.)

[4] https://datatracker.ietf.org/doc/draft-lear-opsawg-mud-sbom/

Possible ref: https://reuse.software/

Example 3: Publish/Subscribe System

Another means to share SBOMs is via a publish/subscribe system. In this case, a consumer would subscribe to a supplier service for updates that would be published. An example of this would be a shared channel established between supply chain partners where SBOMs are published by upstream authors.

Access

Access is the transfer of the SBOM using the method derived from discovery. The input to this process is the location and access method to retrieve the SBOM. Several transfer mechanisms will be discussed under different scenarios depending on where it resides (note these are not mutually exclusive):

- Example Method 1: SBOM is provided directly to the receiver using email or similar “out of band” mechanisms
- Example Method 2:SBOM is resident on device executing the software the SBOM describes
- Example Method 3:SBOM resides on server provided by the author
- Example Method 4:The SBOM is part of a software distribution, e.g. software update

Commented [2]: The other examples have a accompany paragraph, but one is missing for this example.

Example Method 1: SBOM is provided directly to the receiver using email or similar “out of band” mechanism

Commented [3]: One discussion point for the boader team: is this the common method and should it be moved down, as it is less preferred? The argument for is that we should lead with more preferred methods. The argument to leave it as is would be to show a progression of some form.

This would be the case where the manufacturer has the SBOM but no automated infrastructure for sharing it. This is a less preferred method because it is not easily amenable to automation (a goal of this project).

Example Method 2:SBOM is resident on device executing the software the SBOM describes

When the SBOM is co-resident on a device, it may be retrieved using one of a number of protocols, such as HTTP, Constrained Application Protocol (CoAP) or an OpenC2 binding. This is useful in cases of highly tailored systems that have the ability to expose an API. In the case of HTTP, CoAP and their secure variants, it would be found at a well known location such as /.well-known/sbom (as described above). Such names are unique to each origin HTTP or COAP service. OpenC2 has a number of protocol bindings, such as HTTP/S, Message Queuing Telemetry Transport (MQTT), OpenDxl, and OpenDDS. OpenC2 super-imposes its security model on those bindings. See the OpenC2 FAQ for more info.

Example Method 3 SBOM resides on server provided by the author

One common form of sharing is the use of source is through Continuous Integration (CI)/Continuous Deployment (CD) processes available on development platforms that can detect through well known locations or tags the existence of SBOMs.

When the SBOM may be found on a web site (be it published on the web or through a customer portal), the SBOM is retrieved using HTTP over Transport Layer Security (HTTPS). This is useful in the case of small or legacy systems that have no APIs to transmit SBOMs, or when a software package is being included by a developer and the corresponding SBOM is to be included downstream. Authors may leverage the security model of HTTPS to test for entitlement or otherwise limit distribution as they see fit. Automated tooling must take care to identify portal requirements, and may need to alert the administrator of any registration requirements.

Status Of This and Future Work

While SBOMs can be shared now, we continue to work to improve their applicability to various market segments, associated processes, and tooling. Further information on future work can be found at https://docs.google.com/document/d/1KnFbbpoznx_7GxvqPbhNoGE-OZNvGirnx2hz-sru9PU.