Sharing and Exchanging SBOMs

Draft version 0.2 – July 6, 2020

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

Overview

Transparency in the supply chain means that SBOM data should be accessible to the right people at the right time. We acknowledge that there won't be a single one-size-fits-all solution.

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 [add ref]) nor does it deal with how the receiver uses the SBOM (see [add ref to use case doc]). One key goal of this document is to provide a small set of options for discovery and access of SBOM data to allow flexibility for different use cases while trying to minimize the burden on diverse producers and consumers of SBOM data.

We use the term author to represent the creation of the SBOM, even if there are several intermediaries (e.g. mid-stream developer) involved. We use the term consumer or receiver to represent the recipient of the SBOM transfer. We define upstream as towards the Author and downstream as towards the consumer. We use the term mid-stream developer to indicate intermediaries who make use of upstream components which they combine into something delivered downstream.

Getting SBOM data to the right people at the right time consists first of knowing the SBOM exists and how to access it. We use the term discovery to mean the mechanism used by the consumer to know the SBOM exists and how to access it. One key goal of the process 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:

- SBOM creation by the Author - beyond our scope. This is the act of generating and placement of an SBOM, which we assume has happened.
- Advertisement or Discovery - publishing and locating the SBOM and how to access it
- Access - retrieval of the SBOM
- Use and further sharing – beyond our scope

Advertisement And Discovery

Advertisement or discovery is how an author or a device informs consumers that an SBOM is available and how the consumer learns of the information. It should identify a locator and mechanism to retrieve the SBOM.

When an SBOM is to be used in an operational deployment, it might be included as a URL of an SBOM in product literature or packaging, or as part of a Manufacturer Usage Description (MUD) [RFC8520]. 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.

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 source distributions for use by parties in the middle of the supply chain. Alternatively, the SBOM itself may be generated by mid-stream developers based on compile-time options. For example, CURL may be compiled with varying library dependencies.

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):

- SBOM is provided directly to the receiver using EMail or similar “out of band” mechanisms.
- SBOM is resident on device executing the software the SBOM describes
- SBOM resides on server provided by the author
- The SBOM is part of a source distribution

A trivial case of access is the author transmitting a file via EMail. This would be the case where the manufacturer has the SBOM but no automated infrastructure for sharing it.

When the SBOM may be found on a web site (be it published on the web, through a customer portal), the SBOM is retrieved using HTTP. 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 HTTP 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.

When the SBOM is co-resident on a device, it may be retrieved using one of a number of protocols, such as HTTP, 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/HTTPS/CoAP/CoAPS, it would be found at a well known location such as /.well-known/sbom. Such names are unique to each origin HTTP or COAP service.   Each protocol has its own security model. For example, in the case of HTTP, a web token created through a device onboarding process might be used, or a basic username/password mechanism might be employed. OpenC2 has a number of protocol bindings, such as HTTP, MQTT, and OpenDDS. OpenC2 super-imposes its security model on those bindings. See the OpenC2 FAQ for more info.

Other access mechanisms may exist but we want to avoid having too many potential solutions as a burden on producers and consumers of SBOM data.

One open question is whether an SBOM consists of a single or multiple packages. If SBOMs must link to one another, then those links must be shareable/retrievable.

Next Steps and Future Work

Validating SBOMs

It may be desirable to validate the authenticity of the SBOM that has been received. The classic method to do this is to check a signature that is bound to a known and trusted identity.   The signature must also be bound to the actual running software. While the actual validation is (for now) beyond the scope of this work, the means by which signatures are stored and transmitted need to be considered. One key aspect is whether the signatures are internalized or externalized objects in the format. In the case of SPDX, for instance, the signature is externalized GPG.[1] When the signature is externalized, its location and retrieval method must be understood and the signature retrieved.

VEX

Both software producers and consumers note that not all vulnerabilities are created equal, and code that is vulnerable in one context may not be vulnerable in another. If an upstream supplier determines that a potential risk from a vulnerability is not manifest or exploitable, they may wish to communicate that risk information downstream to further users to help them make their own risk-based decisions.

VEX is an optional machine readable companion artifact to the SBOM artifact that allows a manufacturer to communicate the current status of a vulnerability discovered in one of the

[1] https://wiki.spdx.org/view/Technical_Team/SDPX_2.0_Provenance

software components itemized in an SBOM. This functionality would increase the value of an SBOM to the end consumer by:

- Eliminating a majority of false positives
- Address instances where a vulnerability cannot be exploited, due to implementation or other controls
- Conveying the manufacturer's progress on a mitigation for the vulnerability
- Recommended workaround(s)
- Availability of a patch or mitigated new version

While an SBOM is usually static to a given build, the VEX can be highly dynamic in the changes to its contents, such as the manufacturer's current status with regard to the vulnerability:

- Aware of the vulnerability
- Triaging the vulnerability
- Investigating the root cause of the vulnerability
- Investigating work arounds for the vulnerability
- Exploitability of the vulnerability
- Communicating details of the vulnerability to regulatory agencies and customers
- Etc.

It is this difference in supported activity that may force a VEX and an SBOM to be two seperate artifacts, however there are use cases where this two artifacts can be expressed as one artifact, such as when the SBOM is available from a manufacturers server as opposed to be resident in a device.

Currently there are two potential candidates for a VEX, the most promising is in the CycloneDX SBOM standard, and the second is the SARIF static analysis interchange format.

It is an open question as to whether VEX is the appropriate name to be used for this function. It is also an open question as to where VEX should be standardized.

DBoM

The Digital Bill of Materials (DBoM) Consortium is a Linux Foundation project that maintains a digital commons where SBOM and other BOM information can be stored in defined taxonomies and accessed using defined policies. An individual DBoM is the collection of records stored in the digital commons that are associated with an individual artifact (a piece of software, hardware, device, virtual artifact).

The DBoM Consortium has created a distributed ledger digital commons based on open source components, populated with standardized records including SBOMs, and accessed under policies based on the Data Trust Forum developed by Bosch and partners based on GDPR.

Individual enterprises can use the DBoM through vendors or other partners or can choose to run a DBoM Node. DBoM Nodes are available as open source and can be created without external support by any entity or can be hosted as a service.