SBOM FAQ

Fork for October 22 Meeting

Table of Contents

OVERVIEW

Q: What is an SBOM?

A: A Software Bill of Materials (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. For details, see Section 2 of "Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)": https://www.ntia.gov/files/ntia/publications/framingsbom_20191112.pdf

Q: Who should have an SBOM?

A: In today's world, software touches every part of our life and spans across industries, with much of it built on third-party code and open source software. Anyone who is concerned about better supporting their software products internally, supporting their customers, and positively differentiating themselves in the marketplace should consider creating SBOMs and providing them to support their customers. Over time, more SBOM requirements may emerge, such as the FDA's mandate for medical device manufacturers in the draft Pre-Market Guidance for Cybersecurity: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/content-premarket-submissions-management-cybersecurity-medical-devices. For additional information on use cases and benefits of SBOMs, see: https://www.ntia.gov/files/ntia/publications/ntia_sbom_use_cases_roles_benefits-nov2019.pdf

Q: Who uses an SBOM and for what?

A: Most SBOM usage fits under one or more of three perspectives: those who produce software, those who choose software, and those who operate software.

- For those who produce software, SBOMs are used to assist in the building and maintenance of their software, including upstream components.
- For those who choose or purchase software, SBOMs are used to inform pre-purchase assurance, negotiate discounts, or plan implementation strategies.

- For those who operate software, SBOMs are used to inform vulnerability management, asset management, to manage licensing and compliance, and to quickly identify software or component dependencies and supply chain risks.

BENEFITS

Q: What are the benefits of an SBOM?

A: Benefits of SBOMs accrue to both software suppliers and consumers, and are similar for both. They include:

- Identifying and avoiding known vulnerabilities
- Quantifying and managing licenses
- Identifying both security and license compliance requirements
- Enabling quantification of the risks inherent in a software package
- Managing mitigations for vulnerabilities (including patching and compensating controls for new vulnerabilities)
- Lower operating costs due to improved efficiencies and reduced unplanned and unscheduled work.

These benefits can be seen by those who develop software, those who select or purchase software, and those who operate software, across every sector. For a complete discussion of all the benefits please see the Roles and Benefits for SBOM Across the Supply Chain Report: https://www.ntia.gov/files/ntia/publications/ntia_sbom_use_cases_roles_benefits-nov2019.pdf

Q: How does an SBOM help in the event of a cyberattack?

A: When flaws or vulnerabilities are discovered in a given component, SBOMs are used to quickly identify software that is affected by the vulnerable component, to assess its usage, and to understand the risk introduced by the vulnerable component. The ability to identify vulnerabilities allows software suppliers to produce patches or provide other remediation options; allows consumers to apply mitigations independently of the

software supplier; and allows the identification of software that is not affected. This enables focusing on the software that may be affected.

Q: In addition to vulnerability management, how can SBOMs help me?

A: With SBOMs, many parts of the organization – from procurement to operations – have the same understanding of their software assets. The benefits of increased visibility and accountability translate into providing more reliable services to their customers. Better visibility into software allows for additional business advantages such as consolidating assets, licensing clarity, reviewing impact of new policies and regulations and responding in a timely fashion to the ever-changing business environment.

Q: How have bills of material and supply chain transparency been helpful elsewhere?

A: Bills of Materials and supply chain management principles have been transformative to the automotive industry, the food industry, and general manufacturing for decades. Many of these principles were pioneered by Toyota in the 1940s and have been improved continuously across a growing list of industries. A key aspect of this revolution is transparency of the supply chain and knowledge of source, quality, and how to address defects efficiently . A promise of this SBOM initiative is to apply proven supply chain principles to modern software development. In fact, the financial sector has been experimenting with software supply chain transparency since as early as 2013.

COMMON MISCONCEPTIONS & CONCERNS

Q: Won't SBOMs be a "roadmap to the attacker"?

A: Theoretically, yes. In reality, the defensive benefits of transparency far outweigh this common concern as SBOMs serve more as a "roadmap for the defender". All information is dual-edged, but insufficient software transparency affords attackers asymmetrical advantages.

- First, attackers don't need SBOMs; mass, indiscriminate attacks like WannaCry serve to remind us that foreknowledge is not a prerequisite to cause harm.

- Second, attackers and their tools can more easily identify software components; conversely, it is often quite challenging, disruptive, inefficient, and even unlawful for defenders to determine the same.
- Third, attackers of any single product can already find *human-readable* target components – licensing requirements have been increasingly requiring disclosure for decades.

SBOMs seek to level the playing field for defenders by providing additional transparency – at enterprise scale – with standard, *machine-readable* decision support.

Q: Does an SBOM require source code disclosure?

A: No. Your proprietary source code remains yours to share or to keep confidential at your discretion. Concerns about exposing the internals of your code's operation are likewise unfounded, as these software components are just "a piece of the puzzle", not anything close to the "whole completed puzzle" that represents your program.

Q: Does a list of the software components I include expose my intellectual property?

A: No. First, there is a big difference between knowing the 3rd party ingredients and the ultimate recipe or execution. Further, any intellectual property associated with these supply chain components belongs to those upstream, 3rd party, commercial and open source software suppliers. In fact, the licenses those suppliers attached to their components may obligate you to publicly disclose that you include their software in any form.

Q: Does an SBOM increase my exposure to license violations?

A: No. License obligations are incurred whenever licensed software is present and these obligations are independent of SBOMs. SBOMs provide inventory of software that may otherwise be hidden. They therefore make visible potential license violations, and the awareness necessary to mitigate them.

Q: Does an SBOM enable patent or license "trolls"?

A: See: "Does an SBOM increase my exposure to license violations?" above.

Q: Will SBOMs increase my licensing costs or licensing commitments?

A: No. The awareness gained by creating an SBOM allows the manufacturer to address unknown licensing commitments that may be lurking in your programs. This permits the manufacturer to address these issues, either through licensing fees or securing more favorable licensing terms, thus avoiding fines, lawsuits, and licensing commitments such as exposure of your proprietary code.

CREATION

Q: Who creates and maintains an SBOM?

A: An SBOM is created and maintained by the producers of software. "Producers" includes manufacturers, suppliers, and individual authors. Ideally, producers assemble SBOMs provided to them by their suppliers; in the absence of SBOM data, producers may have to generate the SBOM data for some components.

Q: What should be included in an SBOM?

A: An SBOM should contain some combination of the following baseline information: author name, supplier name, component name, version string, component hash, unique identifier, and relationship. Licensing, pedigree, provenance, should also be included, if available. For detailed information about SBOM baseline component information, see section 2.2 of "Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)":
https://www.ntia.gov/files/ntia/publications/framingsbom_20191112.pdf

Q: What data formats exist for conveying SBOM data? NEW

A: There are currently three SBOM formats in common use that support the baseline component information for SBOMs:

- Software Identification (SWID) Tagging
  https://csrc.nist.gov/projects/Software-Identification-SWID
- Software Package Data Exchange (SPDX)

  https://spdx.dev/

- Cyclone Dx
  https://cyclonedx.org/

Additional details about SBOM formats can be found in the Survey of Existing SBOM Formats and Standards:
https://www.ntia.gov/files/ntia/publications/ntia_sbom_formats_and_standards_whitepaper_-_version_20191025.pdf

Q: Are there tools that translate between SBOM formats? NEW

A: Yes. The SBOM community has emphasized the necessity for interoperability between different SBOM data formats. There are tools that translate between the SBOM formats, for example: https://democert.org/sbom/. For additional information about tools for each SBOM format, please see the related link to the draft Formats & Tooling documentation:

- Software Identification (SWID) Tagging
  http://tiny.cc/SWID
- Software Package Data Exchange (SPDX)
  http://tiny.cc/SPDX
- Cyclone Dx
  http://tiny.cc/CycloneDX

Q: When is an SBOM created, changed, or maintained?

A: A new SBOM should be created for every new release of a component. Changes to components require corresponding changes to SBOMs to be valid. For details on when to create an SBOM, see Section 4.2 of “Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)”:
https://www.ntia.gov/files/ntia/publications/framingsbom_20191112.pdf

Q: Some software components are made up of other software components themselves. Can an SBOM show that hierarchy?

A: Yes. SBOMs can provide hierarchical information. Each component that is included in an SBOM should have an SBOM of its own if it also includes components. The SBOMs supplied with each software component taken together represent all levels of the hierarchy required to sufficiently understand the software and its various parts. Such

an SBOM is analogous to a manufactured product multilevel bill of materials. https://en.wikipedia.org/wiki/Bill_of_materials#Multi-level_BOM

Q: How deep in the dependency graph should an SBOM enumerate?

A: It depends on the intended audience and the medium of communication. In the case of a machine-readable SBOM, the minimum viable model is one layer deep with the goal of recursing up the supply chain. Many use cases (e.g. the FDA Premarket Submissions for Management of Cybersecurity in Medical Devices) would like to see it as complete as possible, but they understand that complete SBOMs will take time. For most use cases, more complete SBOMs result in greater value.

DISTRIBUTION & SHARING

Q: If I make an SBOM, do I have to make it public?

A: No. The act of making an SBOM is separate from sharing it with those who can use this data constructively. The author may advertise and share the SBOM at their discretion. In the case of publicly available open source software, it makes sense to make the SBOM public. In other cases, sector specific regulations or legal requirements may require more or less access to the SBOM. Moreover, SBOM data that is more broadly available is more likely to have a positive impact across the supply chain for the myriad benefits discussed above.

Q: How will SBOM data be shared?

A: Since SBOMs will be used by a wide range of software, in a diversity of contexts, there may not be a single way to share SBOM data. However, the data can be advertised, shared, and access-controlled (if needed) in a set of predictable and discoverable ways, including:

- Distributed with the source or binary
- Manufacturer website
- Some centralized or trusted third party's website
- Full content from device (e.g. OpenC2)

- Pointer from device (e.g. Manufacturer Usage Description Specification: (https://www.rfc-editor.org/info/rfc8520)
- Human-readable files provided to the purchaser (e.g. OpenChain)

Note the sharing mechanism is independent of who the SBOM is shared with. NTIA stakeholders continue to review how SBOM data can be shared effectively.

ROLE SPECIFIC

Q: How can SBOMs be leveraged as a Purchaser?

Having an SBOM informs and enables the following, prior to purchasing decisions:

- Catalog various parts of the software and their inter-relationships
- Understand chain of licensing of the software product
- Understand complexity of the software (dates, versions of various parts of the software)

Q: How can SBOMs help an engineer provide surveillance for deployed technology in the field for emerging vulnerabilities?

A: Periodic (ideally automated) comparisons against disclosed vulnerabilities (NVD, Vulners, etc.) can provide an early alert to a potential risk lurking in your environment. A subsequent investigation into the impact of the disclosed vulnerability upon your program can be performed so that, if necessary, a patch can be distributed to the field before your product is ever attacked. This improves customer satisfaction and can improve your position in the market. For additional benefits for engineers and other personas, see: https://www.ntia.gov/files/ntia/publications/ntia_sbom_use_cases_roles_benefits-nov2019.pdf

---

“How does SBOM relate to…”

Q: How does SBOM relate to the Manufacturer Disclosure Statement for Medical Device Security (MDS^2)? NEW

A: The Manufacturer Disclosure Statement for Medical Device Security (MDS^2) provides medical device manufacturers with a means for disclosing to healthcare providers the security-related features of the medical devices they manufacture.

The working group that established and modified the latest version of the MDS^2 were both aware of and participants in the NTIA SBOM multistakeholder process. The SBOM section of the MDS^2 was created with these parallel efforts in mind. Where the new MDS^2 was published prior to the availability of formal SBOM documentation, it was designed to be flexible enough to accommodate any emerging guidance and standards from the NTIA SBOM multistakeholder process. For additional details, see: https://www.nema.org/standards/view/manufacturer-disclosure-statement-for-medical-device-security

Q: How does SBOM relate to OpenC2? NEW

A: 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). For additional details, see: https://openc2.org/

Q: How does SBOM relate to Manufacturer Usage Descriptions (MUD)? NEW

A: 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. For additional details, see: https://csrc.nist.gov/publications/detail/sp/1800-15/draft

Q: How does SBOM relate to DBOM? NEW

A: DBoM is a common backbone for attestation sharing including data such as SBOMs among supply chain partners.

GET INVOLVED

Q: Where can I find more information about the NTIA SBOM process? How do I get involved?

A: To learn more about the NTIA Multistakeholder process for SBOM – including scope, definitions, tools, formats, community-drafted documents, and existing state of practices – visit: www.ntia.gov/sbom, www.ntia.gov/softwareTransparency, or reach out to Allan Friedman at afriedman@ntia.gov.