DRAFT For review by NTIA SBOM Stakeholders - June 26, 2020 version

SBOM Overview Two-Pager

Background

Most software depends on third-party components (libraries, executables, or source code), but there is very little visibility into this software supply chain. It is common for software to contain numerous third-party components that have not been sufficiently identified or recorded.

Software vulnerabilities are both the byproduct of the human process of developing software and the increasingly frequent target of attacks into the software supply chain. If users don't know what components are in their software, then they don't know when they need to patch. They have no way to know if their software is potentially vulnerable to an exploit due to an included component – or even know if their software contains a component that comes directly from a malicious actor.

The reality is this: when a new risk is discovered, very few organizations can quickly and easily answer simple, critical questions such as: "Are we potentially affected?" and "Where is this piece of software used?" This lack of systemic transparency into the composition of software across the entire digital economy contributes substantially to cybersecurity risks as well as the costs of development, procurement, and maintenance.

An Ecosystem-Wide Solution

Software spans industry verticals and the underlying components can come from a common foundation of open source and commercial software. Because of this, any solution must work across the entire ecosystem. The solution we have been exploring is known as a software bill of materials (SBOM) – a "list of ingredients" in software.

Greater transparency allows earlier identification (and mitigation) of potentially vulnerable systems, supports informed purchasing decisions, and incentivizes secure software development practices.[1] The idea of a "list of ingredients" is not particularly new, but current trends in security make transparency absolutely essential:

- We need to rapidly respond to known or potential exploits targeting software components such as the Urgent/11 or Ripple20 vulnerabilities.
- IoT, industrial control, medical devices, and embedded systems are particularly important in safety-critical applications and are ever-more dependent on complex software. Introducing SBOMs into these technologies today will help us better respond to risks tomorrow.
- With SBOM data, we can prioritize open source security. Vulnerability management and resilience at scale is about more than specific products; we need to understand where risk is concentrated. For example, which open source software or third party components can give a malicious actor the greatest advantage?

The NTIA Process

In 2018, the National Telecommunication and Information Administration convened a cross-sector, industry-led, multi-stakeholder process on Software Component Transparency. The goal was to bring together perspectives from across the supply chain to understand the potential, the needs, and the

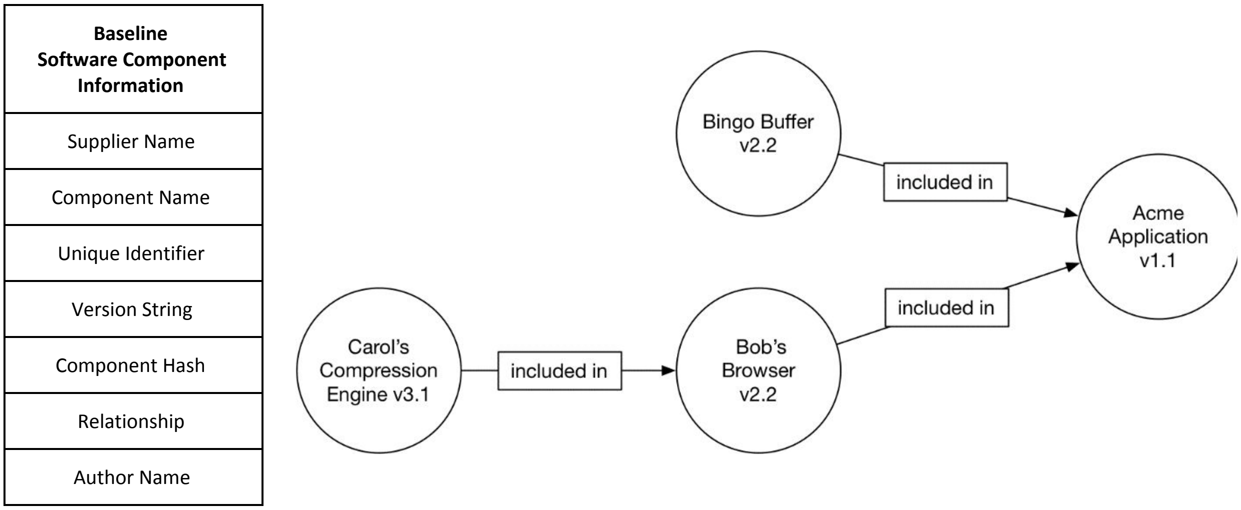
[1] https://www.ntia.gov/files/ntia/publications/ntia_sbom_use_cases_roles_benefits-nov2019.pdf. More resources at https://www.ntia.gov/SBOM

obstacles for widespread adoption of software transparency. This open, non-regulatory process fosters frank exchanges, and practical, engineering- and enterprise-oriented outcomes.

Since its inception, experts have made substantial progress in establishing a shared vision of transparency and a “software bill of materials.” Stakeholders have defined what an SBOM is, documented the security and economic benefits of using SBOMs for those who produce, buy, and operate software, and identified existing standards that can be used to automatically convey SBOM data. Practitioners from the healthcare field have successfully executed a proof-of-concept exercise, generating and using SBOM data from medical device manufacturers in real hospital security systems.

What is a Software Bill of Materials?

An SBOM is effectively a nested inventory: a list of ingredients that make up software components. An SBOM identifies and lists software components, information about those components, and the relationships between them.



Baseline Software Component Information
Supplier Name
Component Name
Unique Identifier
Version String
Component Hash
Relationship
Author Name

Figure 1: The baseline SBOM includes components in their assembled relationship. Each component has enough information to “uniquely and unambiguously identify” it (left), and the relationship of what upstream or child components are “included in” downstream or parent components (right).[2]

Data standards exist today that can capture this SBOM data. These include SPDX, SWID, and CycloneDX. More information, including interoperability guides between the formats, is available in “Survey of existing SBOM Formats and Standards.[3]

In Summary

Knowing the “ingredients” of software installed on any system or device can save hundreds of hours in the risk analysis, vulnerability management, and remediation processes. The Software Component Transparency initiative welcomes participation, and encourages the adoption of SBOMs throughout organizations and the software supply chain process.

[2] https://www.ntia.gov/files/ntia/publications/framingsbom_20191112.pdf
[3] https://www.ntia.gov/files/ntia/publications/ntia_sbom_formats_and_standards_whitepaper_-_version_20191025.pdf