IDC PERSPECTIVE

Operationalizing SBOMs to Secure Your Software Supply Chain

Katie Norton Jim Mercer

EXECUTIVE SNAPSHOT

FIGURE 1

Executive Snapshot: Operationalizing SBOMs to Secure Your Software Supply Chain

This IDC Perspective presents the challenges around operationalizing software bill of materials (SBOMs) that organizations face and that are impeding broader SBOM adoption. It highlights frameworks, open source projects, and commercial products working to provide organizations with solutions to these challenges.

Key Takeaways

- Organizations are challenged to respond quickly and effectively when they do not know whether their applications, which are increasingly composite and include third-party and open source components, contain a vulnerability.
- An SBOM is a machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships.
- SBOMs must be operationalized and integrated into daily operations, existing tools, and security ecosystems.

Recommended Actions

- As the dust settles on SBOMs, organizations must be attuned to industry consensus and regulatory initiatives that may require changes in tooling or workflow.
- In order to prepare for the next Log4j and future regulatory changes, organizations should begin working on an adaptable and flexible SBOM strategy as soon as possible.
- Align strategy with tooling that enables actionable decision making based on SBOM data, such as Dynamic SBOMs, VEX artifacts, and SBOM management solutions.
- SBOMs are not a cure-all and should be part of a comprehensive software supply chain strategy.

Source: IDC, 2023

March 2023, IDC #US50137723

SITUATION OVERVIEW

Today's applications are increasingly more composite, built using cloud-native architectures, and composed of multiple third-party and open source libraries. When an attack occurs, organizations might not be aware of whether their applications contain vulnerable components, making it hard to respond quickly and effectively. Attackers have increasingly taken advantage of this situation by exploiting vulnerabilities hidden in third-party libraries that organizations cannot identify due to their lack of visibility.

As a result, a software bill of materials (SBOMs) has "emerged as a key building block in software security and software supply chain risk management" (according to the Cybersecurity and Infrastructure Security Agency) insofar that it delivers the transparency required to ensure that if a vulnerability occurs in a software component, the organization can identify if it uses that component and should take action.

SBOMs are not new — most software composition analysis tools can generate them. However, SBOMs have recently garnered attention, primarily because of the May 2021 U.S. Executive Order 14028. This EO represents unprecedented government action to protect the U.S. government against software supply chain and infrastructure attacks. It emphasizes that the prevention, detection, assessment, and remediation of cyberincidents is a top priority and essential to national and economic security. The EO includes the directive to establish guidelines for practices that enhance the security of the software supply chain, one of which is providing a purchaser an SBOM for each software product.

An SBOM is akin to a manufacturing bill of material (BOM) — an inventory that tracks the parts needed to create a product. The BOM enables manufacturers to identify the affected products if a defect is found in a component. Similarly, an SBOM is a machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships.

The Log4j vulnerability from December 2021 demonstrates the value and use case for an SBOM. Organizations had to immediately find answers to the extent of their exposure to the Log4j vulnerability. Exposure was often not only in the software they build and maintain but in the commercial off-the-shelf (COTS) software they consume or share within an ecosystem of partners.

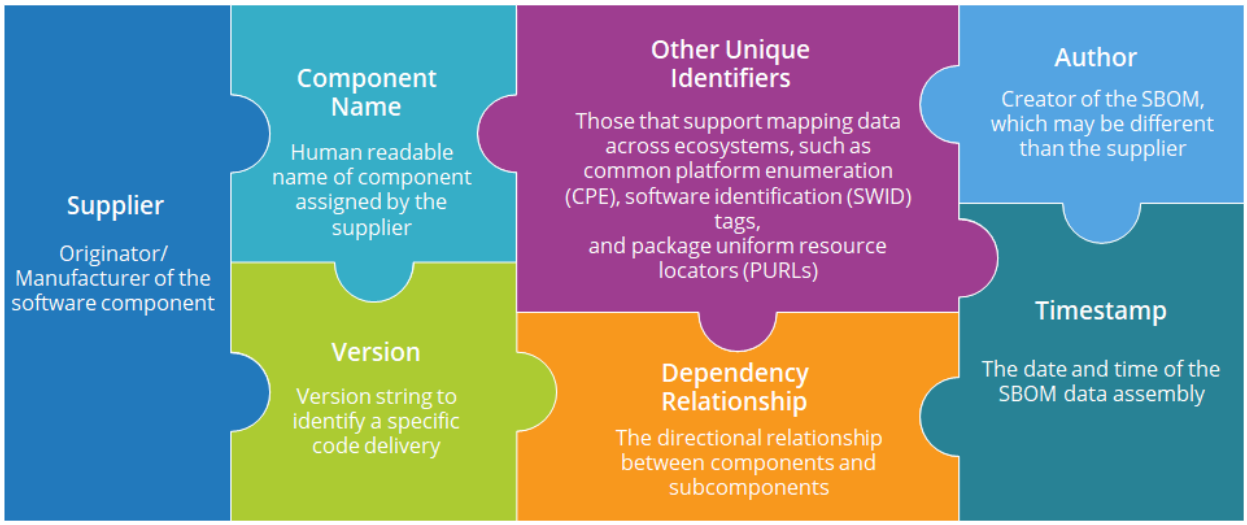
The challenge was that most Log4j usage is buried beneath several levels of transitive dependencies — making it almost impossible for a human to assess exposure manually simply by reviewing the code. However, with an SBOM, an organization could readily identify how and where to respond (see *The Log4j Vulnerability: Widespread Impact and How DevOps Teams Can Respond* [IDC #IcUS48600422, December 2021]).

While vulnerability management is probably the most prominent use case for SBOMs, it is not the only one. An SBOM can also benefit license management, as it can provide an organization with visibility into the license of every component of a product, in addition to the product itself. This visibility can assist legal and compliance departments to reduce potential misuse and thus risk that stems from unauthorized use of third-party software.

An SBOM forms a foundational data layer on which other security tools, practices, and assurances can be built. Figure 1 depicts the minimum data to be included in an SBOM as defined by the U.S. Department of Commerce and the National Telecommunications and Information Administration (NTIA) as of July 2021.

FIGURE 2

Pieces of the SBOM Puzzle



Source: IDC, 2023

An SBOM should be available in standard, machine-readable formats, such as Software Package Data Exchange (SPDX), CycloneDX, and SWID tags, to enhance automation and integration.

An SBOM is important primarily to two personas — those who produce software and those who consume software. For software producers, SBOMs assist in building and maintaining their software, including upstream components. For software consumers, SBOMs inform vulnerability, license, and asset management, and enable quick identification of software or component dependencies and supply chain risks.

Although an SBOM is still a novel idea for most organizations, the ability to produce and consume SBOMs is expected to become more significant in the future. IDC predicts that 55% of organizations will require a signed SBOM for externally consumed apps and software components by 2024 (*IDC FutureScape: Worldwide Developer and DevOps 2023 Predictions* [IDC #US48597522, October 2022]).

The current state of SBOM adoption can be likened to the U.S. government mandate to include an ingredient label on prepared foods that resulted from increased consumer demand for prepared products in the 1960s. The ingredient label enabled individuals to make informed decisions about the food they consumed and allowed food producers to be transparent and create trusting relationships with their customers. Similarly, an SBOM enables software consumers to make informed decisions and software producers to be transparent about the components used in their applications. Like the ingredient label, an SBOM is a critical first step in creating a more secure and transparent marketplace for software.

However, it is important to recognize that an SBOM is just a machine-readable document that does not provide much value in and of itself. Turning back to the ingredient label analogy, having a pantry full of foods, each with its ingredients printed on the box, does not enable a consumer to make actionable food choices easily. Likewise, an SBOM is entirely ineffective when viewed as a "check box" requirement, produced and collected without any plan for management and action.

For SBOMs to be an effective mechanism for aiding in the security of the software supply chain at scale, they must be operationalized and integrated into daily operations, existing tools, and security ecosystems. This IDC Perspective presents the challenges around operationalizing SBOMs that organizations face and that are impeding broader adoption. Frameworks, open source projects, and commercial products looking to address these challenges are highlighted to help organizations with implementing an SBOM strategy.

Static SBOMs and Modern Software Delivery Practices Are at Odds

Today's best practices for software development enable continuous delivery, which results in faster iteration and shorter release cycles. It also means that the delivery of a single SBOM is meaningless — it is a snapshot in time that can quickly become out of sync with the actual application it is supposed to describe. Over a third of the respondents in IDC's *DevOps Practices, Tooling, and Perceptions Survey* deploy code release to production with every change, and another 23% deliver multiple times per day or daily (IDC #US49379723, January 2023).

Whenever a software artifact changes, a new SBOM must be generated to reflect the changes. For both software producers and consumers alike, it becomes increasingly difficult to reconcile SBOMs as they continue to grow. Complicating this are applications that self-update or that only run for a brief period, such as serverless applications. IDC's 2023 *DevSecOps Adoption, Tools, and Techniques Survey* (forthcoming in April 2023) found keeping SBOMs accurate given constant changes were identified as a top challenge regarding managing SBOMs.

Consequently, the market has seen the emergence of solutions that create a dynamic SBOM generated against running production applications. A dynamic SBOM provides an accurate picture of the component the application is using in production by examining the application in a runtime state. As such, it can identify the third-party components being used (i.e., code reachable) and any components linked to the application while running in production. Some examples include serverless libraries, just-in-time (JIT) compiled languages, such as JavaScript applications that often dynamically load libraries at runtime (i.e., lazy loading) to improve performance, or even Java that runs in a Java virtual machine (JVM).

Where a dynamic SBOM shines is its ability to help an organization understand its real-time attack surface and prioritize when there is a high-priority common vulnerability exposure (CVE) (i.e., Log4j/Log4Shell). A dynamic SBOM can identify if a vulnerable component is being executed in runtime and is potentially exploitable.

Table 1 lists the representative solutions providing runtime/dynamic SBOM solutions.

TABLE 1

Representative Dynamic SBOM Solutions

Vendor	Solution
Apiiro	Cloud Application Security Platform
Bionic	ASPM
Codenotary	TrueSBOM
Contrast Security/Eclipse Foundation	jbom (open source)
Deepfence	ThreatMapper
Oxeye	Oxeye Platform
Rezilion	Rezilion Platform
Tauruseer	SPACE Platform

Source: IDC, 2023

SBOMs Lack Necessary Context About Exploitability

Software producers and consumers recognize that not all vulnerabilities are created equal, and a code vulnerable in one context may not be vulnerable in another. If an SBOM reports a vulnerable component, it does not necessarily mean your application is vulnerable. For example, vulnerable dependencies in application code are not always exploitable in runtime environments. Therefore, an SBOM without context can lead to confusion, as each consumer can interpret it differently. The fear of receiving "false positives" keeps many software suppliers from distributing an SBOM to their customers. Software suppliers must have a standard method of notifying users when components are not exploitable.

The Vulnerability Exploitability eXchange (VEX), initially introduced by the NTIA in 2021 and expounded upon by the U.S. Cybersecurity and Infrastructure Security Agency (CISA) in 2022, has emerged as a companion to an SBOM that communicates the status of vulnerabilities found in software components listed in the SBOM. In contrast to an SBOM, which is updated whenever a component changes, a VEX artifact is updated whenever a vulnerability or threat is discovered. Suppliers can issue VEXs to indicate if a component is not affected, affected, fixed, or under investigation. Ultimately, VEX reduces the time and effort users spend investigating non-exploitable vulnerabilities and informs consumers of the vendor's actions or which actions the consumer must make to reduce risk.

In the same way an SBOM requires industry collaboration for mass adoption, VEX will require the private and public sectors to create formats and tools that can be easily integrated into current development practices. Currently, two standard formats exist for VEX artifacts: the Common Security Advisory Framework (CSAF) and CycloneDX. In January 2023, a third specification, OpenVEX, was announced, providing a lightweight and SBOM-agnostic VEX document. Tools to help with VEX have also emerged, such as Chainguard's open source tool vexctl, which is a command line interface (CLI) that enables scanners to consume VEX data to turn off false positives. Other vendors have announced the capability to automatically generate VEX documents within their solutions, including Endor Labs and Cybellum.

SBOMs Require Management Tools to Be Actionable

In the event of a security incident or vulnerability, organizations need the ability to query all their software's SBOMs instantly. Being able to query SBOMs across the application portfolio enables the organization to determine the impact rather than wait for each application development team to provide them with individual assessments or waste valuable time scanning each application again. A static SBOM in a build directory or a document storage system offers little benefit. When an SBOM is generated for every deployment or delivered release of an application, it will create SBOM sprawl over time as more and more SBOMs are produced and ingested.

Organizations need SBOM management solutions that provide a centralized repository or database. As modern applications typically include open source and third-party commercial libraries along with internally developed code, these solutions must be able to ingest SBOMs external to the organization. Further, the solution must be able to reconcile and normalize SBOM data to provide a unified, organizationwide view. This requirement also necessitates support for all the major SBOM formats, VEX documents, and the ability to collect component-level SBOMs. SBOM management solutions should include features such as search and comparison for drift detection.

Table 2 lists the representative solutions providing SBOM management.

TABLE 2

Representative SBOM Management Solutions

Vendor	Solution
Anchore	Anchore Platform
Chainguard	Chainguard Enforce
Codenotary	TrustCenter
Cybeats	SBOM Studio
Cybellum	Cybellum Platform
Dependency Track	Dependency Track Platform
Endor Labs	Endor Labs Platform
FOSSA	FOSSA Platform
Revenera	SBOM Insights
RKVST	RKVST Platform
Scribe	Scribe Platform

Source: IDC, 2023

ADVICE FOR THE TECHNOLOGY BUYER

- **Be attuned to industry consensus and regulatory initiatives.** The dust has not settled on how SBOMs will be used and regulated. Consolidation around a particular SBOM format, the emergence of new standards and best practices, and new mandates or regulations may require shifting of tooling and workflows.
- **Do not wait to get started on an SBOM strategy.** Many organizations currently are not regularly generating or managing SBOMs for their applications. It is important to get started today on a flexible and adaptable strategy to avoid being in the hot seat when and if eventual regulation or industry best practice comes to fruition.
- **Embrace tools that make SBOMs actionable.** Dynamic SBOMs, VEX artifacts, and SBOM management solutions aim to allow organizations to make informed decisions based on the data contained within their SBOMs.
- **SBOMs are not a software supply chain security magic bullet.** Just because an organization knows what third-party components are in its software does not mean it knows all the vectors introducing risk into the business. For example, SBOM management would not have stopped the well-known SolarWinds breach. SBOMs should be part of a comprehensive software supply chain strategy that includes code security and management, digital provenance, secure build images, and more.

LEARN MORE

Related Research

- *IDC FutureScape: Worldwide Developer and DevOps 2023 Predictions* (IDC #US48597522, October 2022)
- *OMB Introduces Memo to Ensure Secure Development of Software Purchased by the Federal Government* (IDC #lcUS49707622, September 2022)
- *How Many Organizations Have Been Impacted by an Open Source Software Vulnerability, and What Is the Effect on Usage?* (IDC #US49613622, August 2022)
- *The Log4J Vulnerability: Widespread Impact and How DevOps Teams Can Respond* (IDC #lcUS48600422, December 2021).
- *IDC TechBrief: Software Composition Analysis of Open Source Software* (IDC #US46133920, March 2020)

Synopsis

This IDC Perspective presents the challenges around operationalizing software bill of materials (SBOMs) that organizations face and that are impeding broader adoption. Frameworks, open source projects, and commercial products looking to address these challenges are highlighted to help organizations with implementing an SBOM strategy.

"The SBOM has been all the rage since the Executive Order, but both quantitative and qualitative data suggest that organizations are struggling with implementing the practices and tools necessary to make the SBOM actionable in securing their software supply chains," says Katie Norton, senior research analyst, DevOps and DevSecOps practices at IDC. "However, an ecosystem of frameworks, projects, and tools is forming that can help organizations establish an SBOM strategy that will set them up for success when the next Log4J or government regulation comes around."

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

Global Headquarters

140 Kendrick Street
Building B
Needham, MA 02494
USA
508.872.8200
Twitter: @IDC
blogs.idc.com
www.idc.com