Professional Perspective

Enhancing Cybersecurity Readiness With a Software Bill of Materials

David J. Kappos, Sasha Rosenthal-Larrea, Kathryn-Ann Stamm and Lucille Dai-He, Cravath, Swaine & Moore

Bloomberg Law

Enhancing Cybersecurity Readiness With a Software Bill of Materials

Contributed by ***David J. Kappos***, ***Sasha Rosenthal-Larrea***, ***Kathryn-Ann Stamm*** *and Lucille Dai-He*

Cravath, Swaine & Moore

Whether a company is governing code dependencies in an internal software project, onboarding a new technology vendor or evaluating an acquisition target, it is important to have a clear understanding of what is inside and outside the transaction perimeter. Just as physical products include materials of different quality from different sources, software products comprise building blocks of code that can vary in security, compatibility and operational risk.

Software products are rarely developed completely from scratch; the vast majority of software contains some combination of open-source, third-party proprietary and in-house developed proprietary components. It is important to document and understand what components are included within a given software product in order to make informed decisions about the product, understand any legal limitations on the code, address potential vulnerabilities and monitor compliance with the terms and conditions associated with each component on an ongoing basis.

An emerging and effective way to track the components of a software product is through a software bill of materials ("SBOM"), which is formal, machine-readable metadata providing a comprehensive inventory of all components in a software package, giving visibility into the entire software supply chain, including all third-party components which were not developed in-house. As the digital landscape evolves and threats become more sophisticated, organizations can proactively address cyber risk with SBOMs by (1) adopting software procurement policies that require SBOM disclosure as a condition for technology procurement and (2) building internal processes for generating and managing SBOMs consistent with industry standards.

Given recent regulatory requirements and ongoing market pressures from prominent software providers, SBOMs are set to become a standard practice for safeguarding against cyber risks. Diligence teams should consider SBOM adoption when evaluating a potential target's software security practices, and can use the information documented within SBOMs to evaluate potential security and operational risks.

Introduction to SBOM Analysis

An SBOM, in its simplest form, is a nested inventory for software, listing all components and third-party dependencies within a piece of software. If not using SBOMs, companies will typically leverage a combination of tools and practices to gain an understanding of their software supply chain, including code scanning tools, package managers, version control systems, digital signatures, provenance tools, etc. In contrast, SBOMs are machine-readable and follow standardized formats.

Widely accepted formats for SBOMs include Software Package Data eXchange ("SPDX"), CycloneDX and Software Identification Tags ("SWID Tags"). For example, a partial excerpt of an SPDX SBOM may look like this:

```
#Package Information
PackageName: Example
SPDXID: SPDXRef-Package-Example
PackageVersion: 2.11.1
PackageVerificationCode: d6a770ba38583ed4bb4525bd96e50461655d2758
PackageLicenseConcluded: GPL-3.0-or-later
PackageLicenseInfoFromFiles: GPL-3.0-or-later
PackageLicenseDeclared: GPL-3.0-or-later
#File Information
FileName: ./docs/example1
SPDXID: SPDXRef-example-one
FileType: ARCHIVE
FileChecksum: SHA1: a14bc8609e0c9be98f7fd1ae6348417fc4b6ed3a
LicenseConcluded: GPL-3.0-or-later
LicenseInfoInFile: GPL-3.0-or-later
FileCopyrightText: NOASSERTION
FileContributor: Jane Doe
#File Information
FileName: ./docs/example2
SPDXID: SPDXRef-example-two
FileType: SOURCE
FileChecksum: SHA1: 089e0b355e42624e02b9e1f995e03aab7922c369
LicenseConcluded: Apache-2.0
LicenseInfoInFile: Apache-2.0
FileCopyrightText: <text>Copyright 2010, 2011 Acme Corporation.</text>
FileContributor: Acme Corporation
```

Note: This illustrative example is not from a real file, but is based on an SPDX format example available on the SPDX GitHub repository. *See* SPDX, SPDXTagExample-v2.3.spdx Github (Aug. 26, 2021), **https://github.com/spdx/spdx-spec/blob/development/v2.3.1/examples/SPDXTagExample-v2.3.spdx**.

This machine-readable file provides metadata about the software being examined. SBOM files are not intended to be read directly in this format, and there are separate tools that translate the files to more human-readable outputs and user-friendly dashboards for SBOM analysis. Nevertheless, the SBOM file excerpt reveals key pieces of information that have been highlighted, including the component software files governed by open source, license information for such files, the package version number and the contributors.

SBOMs need to be created and maintained, and there are many software development tools on the market that automate this process. An SBOM file should be generated each time software is released and shared across the supply chain to aid supply chain management. While it is also possible to generate SBOMs retroactively by analyzing existing software for dependencies, if an SBOM is not generated based on information tracked throughout the development and release of the software, there will be a risk of gaps in key provenance information, and it may be impossible to correctly identify the dependencies used at build time.

Obtaining and understanding the content of an SBOM enables faster responses to known vulnerabilities by enabling users to search for specific attributes and unique identifiers to detect code dependencies in a given product or software package. Prior to SBOM adoption, the common method of identifying where a specific software component appears in an organization's software ecosystem was to use third-party scanning tools to scan through snippets of code to match them against known open-source component databases. Such scanners are known to generate false positives, making it difficult to quickly distinguish vulnerable components from low-risk components. Well-maintained SBOMs directly identify software components using unique metadata rather than code excerpts, enabling faster and more consistent identification of software components. Tracking SBOM information and implementing SBOM infrastructure can help organizations evaluate software building blocks in the short term as well as mitigate vulnerabilities in the long term.

In the short term, whether evaluating software dependencies for a project or software tools to onboard from a vendor, an important part of software management is making sure the software complies with legal, regulatory and industry standards. This includes compliance with licenses to which open-source and third-party software are subject. Many software dependencies include open-source software, which is usually licensed under one of several recognizable industry-standard licenses. The type of license under which open-source software is made available is important, and SBOMs facilitate identification and audits to ensure compliance with applicable license terms. For example, copyleft licenses, such as the GNU General Public License ("GPL"), require derivative works to be licensed on an open-source basis, which limits commercial opportunities for projects incorporating open-source software licensed under such terms. Awareness of the use of software components subject to these licenses is an important step to reduce risk; these licenses can have significant ramifications for the software products integrating open-source software.

In addition to licensing information, SBOMs can express other valuable information for identification and tracking of software quality issues. For example, SBOMs can enable organizations to identify end-of-life components (software that is no longer updated or maintained by its original developers), technical debt (imperfect code released to prioritize speedy delivery, that should be reworked retroactively) and software with foreign origin or influence. The latter is becoming increasingly important as governments are increasing their focus on the national security implications of foreign technology.

In the long term, SBOMs can enable organizations to manage a software product's inventory to respond quickly to vulnerabilities across software dependencies. One of the key objectives of cybersecurity teams is to respond quickly to cyberattacks and ideally identify and mitigate risks before they materialize. Some of the most devastating attacks are known as "zero-day exploits"—cyberattacks that take advantage of a vulnerability in computer software that was previously unknown, where developers have not had the opportunity to patch the vulnerability. A notable example is the Apache Log4j vulnerability discovered in December 2021 that sent shockwaves throughout the information security world. Log4j was a very popular logging tool used by tens of thousands of software packages across the industry, impacting even the most prominent technology companies' software. But organizations' insufficient visibility into their software's direct and indirect dependencies made it difficult to track where the Log4j library existed across the enterprise, which in turn hindered efforts to patch the vulnerability.

SBOMs provide a clear understanding of the component parts in a software product, so when a vulnerability like Log4j is discovered, company cybersecurity teams can quickly and effectively find and patch the vulnerability, even across numerous integrated software products. If a vulnerability is discovered in a particular version of an open-source library, the affected organization can use SBOMs to search for each place where that particular version appears as a dependency and upgrade to the latest patched version. After a vulnerability is identified, metadata documented within SBOMs, such as time stamps and version numbers, can help organizations quantify **exposure risk** by measuring how long vulnerabilities in the software product have been allowed to persist, allowing adversaries to develop and test exploits. Without standardized SBOMs, this is at best a much more lengthy and onerous process. It may take the organization much longer to realize that a particular software product is impacted by the vulnerability, at which point it may be too late.

Regulations Governing SBOM Adoption & Implementation

President Joe Biden's 2021 **Executive Order 14028** on Improving the Nation's Cybersecurity (the "Executive Order") set into motion **increased attention** to SBOMs, which it defined as "a formal record containing the details and supply chain relationships of various components used in building software." This Executive Order charged multiple agencies with issuing guidance around practices to enhance software supply chain security, including standards and procedures regarding SBOMs.

Following the Executive Order, the National Telecommunications and Information Administration ("NTIA") **published** helpful guidance and informational resources regarding SBOMs. According to the NTIA **guidance**, the minimum data fields for an SBOM are: the supplier name, component name, version of the component, other unique identifiers, dependency relationship, author of SBOM data and timestamp. These fields help map software components to other sources and distinguish a component from others (such as other versions of the component from the same source). NTIA recommends that SBOMs be conveyed across organizations using at least one of three primary data formats for SBOMs: SPDX, Cyclone DX and SWID Tags. These standard formats allow for SBOMs to be interoperable across organizations. NTIA has also promulgated best practices related to practice and process. A new SBOM should be created each time a software component is updated. SBOMs should include the appropriate level of depth, not only top-level components, but also their dependencies and flagging known unknowns. Appropriate access permissions and roles should be put in place, and SBOMs should be made available in a timely fashion. These NTIA guidelines are non-binding, but they provide a set of industry best practices that companies should work towards.

The National Institute of Standards and Technology ("NIST") subsequently **published** its own Software Supply Chain Security Guidance, with a set of practices for the development of secure software, which acknowledges that SBOMs may "provide increased transparency, provenance, and speed at which vulnerabilities can be identified and remediated by federal departments and agencies." Following the Executive Order and the NIST guidance, federal agencies have begun requiring federal software product suppliers to produce SBOMs conforming with NTIA's minimum requirements to improve transparency and security in federal contracts. Looking ahead, a pending **proposal** by the Department of Defense ("DoD"), the General Services Administration ("GSA") and the National Aeronautics and

Space Administration ("NASA") seeks to add an SBOM requirement to the Federal Acquisition Regulation ("FAR"), formally requiring SBOMs from federal contractors across the board.

Several other regulatory bodies have also enacted rules requiring SBOMs in certain cases. Pursuant to the **Federal Food, Drug, and Cosmetic Act** (the "FD&C Act"), manufacturers seeking U.S. Food and Drug Administration ("FDA") clearance or approval for cyber devices must disclose SBOMs for cyber devices when seeking FDA clearance or approval. *See***Federal Food, Drug, and Cosmetic Act Section 524B(b)(3)**; Title 21 Chapter 9 § 360n–2(b)(3). In 2023, the FDA also released **guidance** aimed at cybersecurity in medical devices, recommending that SBOMs be submitted as part of security risk management documentation for premarket submissions for medical devices.

Pending legislation in the U.S. and other jurisdictions also stands to increase SBOM adoption. The Securing Open Source Software Act of 2023, **introduced** in March 2023, would direct the Cybersecurity & Infrastructure Security Agency ("CISA") to assess the open-source software components used by federal agencies by means of SBOMs, software inventories and other publicly available information. In the EU, the Cyber Resilience Act ("CRA") **proposal** highlights the importance of identifying and documenting software product components, including using SBOMs, to enhance supply chain understanding and vulnerability analysis. If adopted, the CRA would also confer powers upon the EU Commission to specify standardized formats and elements for SBOMs.

SBOMs in Legal Due Diligence

Cybersecurity is an increasingly important topic in legal due diligence. SBOMs allow more transparency into a potential acquisition target or vendor's software stack and can serve as a complement to existing cybersecurity diligence practices such as software composition analysis, penetration tests and other risk assessments.

Before making purchase decisions, acquirors must be informed of their risks and require adequate disclosure of various IT practices underlying the target or the vendor's representations and warranties. Key topics that potential buyers inquire into include: an IT system's usability, whether there have been any breaches of software components and what controls are in place. SBOMs can help to better understand the software components involved in a deal, including potential security risks and license compliance. Obtaining visibility into a target or its vendor's software components can expose potential vulnerabilities and dependencies, alerting the acquirer to software risks that may otherwise go undetected. Moreover, adherence to SBOM standards can serve as evidence of adhering to accepted industry practices set forth by NTIA and other regulatory bodies.

In the diligence process, SBOM analysis can be used to pressure test a target or vendor's representations and warranties regarding its security practices. For instance, many security certification standards require attestations about maintaining a secure software development life cycle ("SDLC"), but the terms "secure" and "life cycle" are seldom precisely defined. Yet, if the target or vendor's SBOM contains dozens of components that are several major versions behind and barnacled with vulnerable dependencies of their own, the acquiror or customer can ask the vendor to reconcile their representation with empirical evidence that its security practices do not meet industry standards. On the other hand, a target or vendor can disclose SBOM analysis documenting up-to-date and vulnerability-free components as supportive evidence of secure SDLC practices.

In this regard, SBOM analysis is salient not only as a security check, but also as an assessment of total cost of ownership and the ability of a supplier to update their software product in the future. For instance, there is high operational risk when a critical vulnerability cannot be remediated with an update because too many other components are out-of-date and incompatible with the new, patched versions of the critically vulnerable component. Similarly, end-of-life and outdated components require expensive refactoring in order to integrate the component with other systems or capabilities. Evidence of an unmaintainable software product–which transfers operational risk to the customer or acquiror–may inform contractual covenants for security response and indemnities for costs arising from vulnerable components. This increased operational risk may influence the price a customer is willing to pay for a software product. In acquisitions, this may similarly influence the target's valuation, given the level of effort that will be required to integrate and maintain the capability in a way that satisfies the acquiror's internal standards and regulatory requirements.

Conclusion

SBOMs are a useful tool to inventory software components and to get a grasp on the dependencies of a particular software product. As regulatory requirements and cybersecurity risks drive SBOM adoption, it is important for legal teams to understand the utility of SBOMs and the growing role SBOMs play in cyber readiness. Organizations should begin adding SBOM disclosures to their software procurement policies and leveraging SBOMs to evaluate security vulnerabilities, license compliance and incident response and remediation. Setting up infrastructure around SBOMs not only helps organizations make better procurement choices in the short run, but also mitigates risk in the long run by enabling teams to respond more efficiently to cybersecurity vulnerabilities.

Cravath, Swaine & Moore llp