sonatype

The 10 Most Frequently Asked SBOM Questions Answered

What are the minimum requirements for an SBOM?

In order to validate the authenticity of a particular software component, an SBOM has to provide identification, including the supplier and name, the component name, the version, dependency relationships, the author of the SBOM, and the time the data was added to the SBOM. Also required is support for automatically generating and parsing, with the goal of interoperability across organizations. **System integrators and software vendors** that supply software to the US government must also attest to compliance using instructions outlined in the Secure Software Development Attestation Form provided by the Cybersecurity and Infrastructure Agency (CISA) and the Office of Management and Budget (OMB).

What impact does EO 14028 have on supply chain regulations, and what actions are being taken in other places around the world to address this problem?

Recent legislation, including Executive Order 14028 and NIST SP 800-218 in the US and NIS2 and the Cyber Resiliency Act (CRA) in the European Union, are driving important awareness of the effectiveness of SBOMs. Other countries, including Germany, Japan, and Korea, are all looking at ways to legislate the use of SBOMs. In 2023, CISA launched its Secure by Design initiative with the participation of cybersecurity authorities from around the world to underscore the role security has throughout the software development lifecycle.

Are you seeing industries standardizing around SBOM formats?

Yes, standardization is one of the key pillars of SBOM management. The National Telecommunications and Information Administration (NTIA) recognizes three main formats for SBOMs: CycloneDX, SPDX, and SWID. These formats allow SBOMs to be automatically generated during the development process and ensure compatibility with other tools and systems so vendors and customers work with compatible tools.

Could you explain the VEX process and how it applies to SBOM vulnerabilities?

An SBOM provides a comprehensive list of software components, while VEX (Vulnerability Exploitability eXchange) documentation offers crucial insights into known vulnerabilities and exposures within these components. This integration of VEX data with SBOMs is not just a technical detail, but a significant step towards enhancing software security. By pinpointing the components that are truly at risk, you can effectively prioritize vulnerabilities that demand immediate attention. VEX-enriched SBOMs play a pivotal role in identifying and mitigating the most critical vulnerabilities within a specific application, thereby aiding organizations in adhering to regional regulations.

My organization has implemented a wide range of tools for preemptively addressing security issues. Would Sonatype have any further advantages in helping solve this?

Yes. The danger of having a scattershot collection of tools is that organizations often rely on providing reports to developers. But without guidance or policy control, developers just get inundated. The number of dependencies in a software application is overwhelming. A system that can provide prioritized, actionable guidance is necessary. Having the right tools in place is a good start, and Sonatype can help make sure the data those tools provide is actionable.

How does the use and maintenance of the SBOM change in a SaaS development model vs. a COTS model?

It's important to build the SBOM generation and evaluation into your normal build testing workflow. From a COTS perspective, the ability to consume and evaluate SBOMs for policy guidelines is critical. Sonatype makes it possible to merge these two use cases using the same tool to evaluate both types of SBOMs for policy and potential security issues. Remediation is mostly done by the vendor In the COTS model, so it's important to set standards for vendor management.

As the use of AI in software development increases, how will SBOM management need to evolve?

In Sonatype's most recent State of the Software Supply Chain Report, three out of four DevOps leads reported concerns about the impact of generative AI on security, especially in open-source code. With the rise of AI, we need to be aware that if used incorrectly, it's going to make this problem worse, not better. SBOMs will need to accurately reflect the inclusion of AI components within software systems and emphasize transparency, providing insights into how AI components contribute to overall system functionality. SBOM tools also need to integrate with security scanning and vulnerability management systems to identify and address potential risks associated with AI components.

For an organization with thousands of products, how can we automate the centralized storage, management, and distribution of our SBOMs?

There should be an emphasis on automation. Use tools to automate the process of collecting, managing, maintaining, and storing your SBOMs. It's not unusual for developers to publish multiple times a day, and application teams can't keep up. The more you can automate and integrate SBOMs and build them into the CI/CD pipeline, the more you can alleviate that pressure. This could include automating policy support so development teams know upfront during the design stage what open source or third-party products are approved and which ones to avoid.

How can I convince my organization about the value of SBOMs?

SBOMs are increasingly becoming necessary for doing business. For example, the FDA is no longer approving new medical devices without an associated third-party risk model. There are also emerging requirements like IEC (International Electrotechnical Commission) 63000 that define requirements for listing software of unknown origin. An SBOM is an automated, standardized answer to that requirement. Organizations that are taking SBOM management seriously recognize it as more than a compliance effort - it's about using it as a testing framework for developers to act on quickly. The faster you can alert developers to potential vulnerabilities, the quicker it can be addressed.

How can an SBOM help identify snippets of a third-party component where the developer only uses part of it and removes the unused parts?

If the developer repackages dependencies, they will become apparent in the dependency hash section as they do not correspond with canonical package identities. Sonatype can highlight this and either mark it as 'similar to package-foo' or a 'completely unknown component.'

Simplify SBOM Compliance and Security Monitoring with Sonatype SBOM Manager

More than 70 percent of Fortune 100 companies manage their software supply chains with Sonatype, and our **SBOM Manager** has been developed to take the uncertainty out of SBOM collection and monitoring compliance.

Request a demo today

sonatype