


anchore

THE SOFTWARE BILL OF MATERIALS AND ITS ROLE IN CYBERSECURITY

HOW TO USE SBOMS TO
STRENGTHEN THE SECURITY OF
YOUR SOFTWARE SUPPLY CHAIN
FOR CLOUD-NATIVE APPLICATIONS



CONTENTS

| | |
|---|----|
| Executive Summary | 3 |
| Introduction | 3 |
| Current State of the SBOM | 4 |
| The Role of the SBOM | 4 |
| The Role of the SBOM Standards | 6 |
| 5 Ways to Elevate the Role of the SBOM | 7 |
| Define Your Path to SBOM Adoption | 7 |
| Define the SBOM Elements and Standards Needed | 7 |
| Automate SBOM Generation in your DevOps/DevSecOps Toolchain | 8 |
| Implement SBOM Management | 8 |
| Watch for Unexpected SBOM Drift | 8 |
| Embedding SBOMs in DevSecOps | 9 |
| SBOM Adoption via Executive Order | 10 |
| Summary: SBOMs Have A Critical Role in Securing the Software Supply Chain | 11 |
| About Anchore | 12 |

Executive Summary

The software bill of materials (SBOM) is one of the most powerful security tools that you probably aren't already using. Large-scale software supply chain attacks such as SolarWinds and Codecov highlight the need for organizations to understand the components—and the security posture—of the software they create or use. SBOMs are critical not only for identifying security vulnerabilities and risks in software. They are also for understanding how that software changes over time, potentially introducing new risks or threats. Now is the time for your organization to incorporate and elevate SBOMs in your cybersecurity and development processes.

Introduction

Knowing what's in software is the first step to securing it. Increasingly organizations are developing and using cloud-native software that runs in containers. Consider the complexity of these containerized applications, with hundreds, sometimes thousands of components from commercial vendors, partners, custom-built software, and open source software (OSS). Each of these pieces is a potential source of risk and vulnerabilities. Generating SBOMs to create a trackable inventory of these components is a crucial and necessary step in securing the software supply chain.

Yet, despite the importance of SBOMs for container security practices, only 25% of the respondents to the [2021 Anchore Software Supply Chain Report](#) produce an SBOM for the containerized apps they build, and only 28% require an SBOM from their software suppliers.

SBOMs must become a priority. This white paper addresses the need to elevate the role of the SBOM, provides an introduction to best practices for using SBOMs, and offers insights into how your organization can promote the use of SBOMs as a foundation for cybersecurity.

Current State of the SBOM

The risks of software supply chain attacks are real, with almost two-thirds of enterprises impacted by a software supply chain attack in the last year according to the [2021 Anchore Software Supply Chain Report](#). To stem these rising threats, the United States Executive Order on the Nation's Cybersecurity outlines new requirements for SBOMs along with other security measures for software used by federal agencies. Until now, the use of SBOMs by cybersecurity teams has been limited to the largest, most advanced organizations. However, as a consequence of these two forces, the use of SBOMs is on the cusp of a rapid transformation.

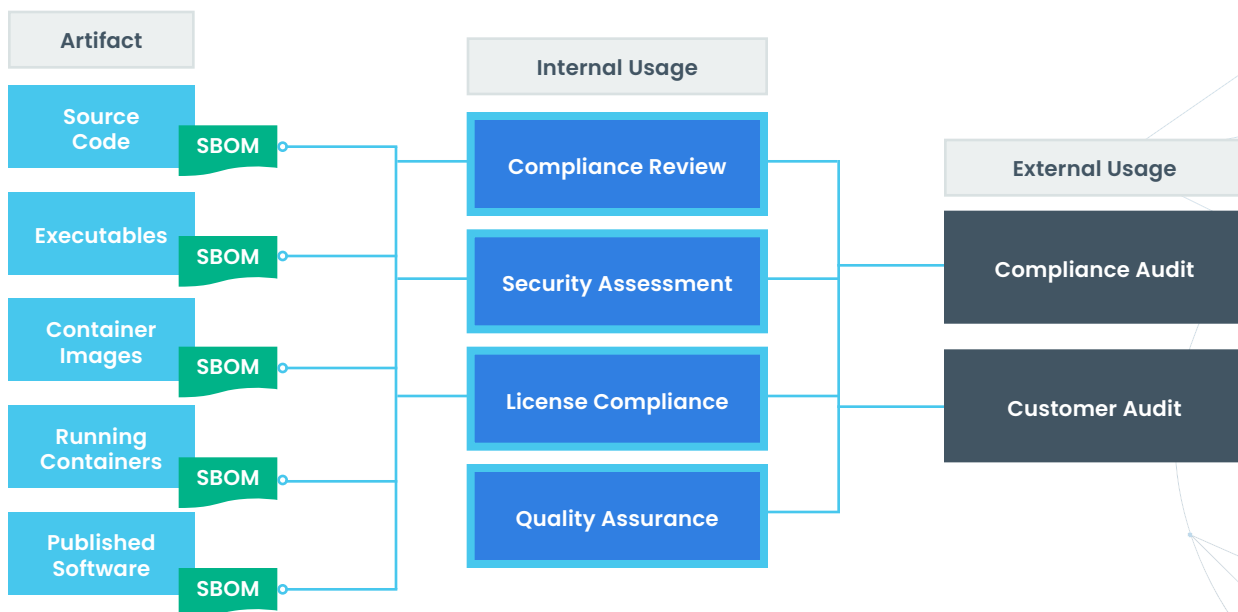
With governments and large enterprises leading the way, standardized SBOMs are poised to become a baseline requirement for all software as it moves through the supply chain. As a result, organizations that produce or consume software will need the ability to generate, consume, manage, and leverage SBOMs as a foundational element of their cybersecurity efforts.

The Role of the SBOM

An organization can use SBOMs for many purposes. The data inside an SBOM has internal uses:

- » Compliance review
- » Security assessments
- » License compliance
- » Quality assurance

Additionally, you can share an SBOM externally for compliance and customer audits. In this white paper, we will focus on the role of the SBOM for container security and development.



Within the security and development role, SBOMs serve a similar purpose as a bill of materials in other industries. For example, automotive manufacturers must track the tens of thousands of parts coming from a wide range of suppliers when manufacturing a modern car. All it takes is one faulty part to ruin the final product.

Cloud-native software faces similar challenges. Modern applications use significant amounts of open source software that depends on other open source components which in turn incorporate further open source components. They also include internally developed code, commercial software, and custom software developed by partners. Combining components and code from such a wide range of sources introduces additional risks and potential for vulnerabilities at each step in the software development lifecycle. As a result, SBOMs become a critical foundation for getting a full picture of the “ingredients” in any software application.

Collecting SBOMs from software suppliers and generating SBOMs throughout the process to track component inventory changes and identify security issues is an integral first step to ensuring the overall security of your applications.

Security and development teams can either request SBOMs from their software suppliers or generate an SBOM themselves. Having the ability to generate SBOMs internally is the more optimal approach. This way teams can produce multiple SBOMs throughout the development process to track component changes and search for vulnerabilities as new issues become known in older software.

SBOMs can help to alleviate the challenges faced by both developers and security teams by:

- » Identifying risk exposure inherent in open source and third-party tools
- » Reduce development time and cost through exposing and remediating issues earlier in the cycle
- » Identifying license and compliance requirements

Ultimately, SBOMs are a source of truth. To ensure product integrity, development and security teams must quickly and accurately establish:

- » Specific versions of the software in use
- » Location of the software in their builds and existing products

The Role of SBOM Standards

SBOM standards are schemas designed to provide formats for describing the composition of software and are consumable by other tools, such as vulnerability scanners. CycloneDX and Software Product Data Exchange (SPDX) are the most commonly used standards.

| | SPDX | CycloneDX |
|---------------------------|--|---|
| Organization | SPDX Workgroup under the Linux Foundation | A “meritocratic, consensus-based community project” with an industry working group |
| Formats | RDF, XLS, SPDX, YAML, JSON | XML, JSON |
| Specs | spdx.github.io/spdx-spec BS ISO/IEC 5962 - 2020 | github.com/CycloneDX/specification |
| Unique Features | Extensive support for expressing license details | Extensible format and integrates SPDX license IDs, pURL, and other external identifiers |
| Original Use Cases | License management | For use with OWASP Dependency Track |
| Shared Use Cases | <ul style="list-style-type: none">» Tracking attributes of multiple software components such as vendor, license, and version.» Generically describe packages, containers, operating system distributions, archives, and other elements» Integrity verification of software components and sub-components | |

Both standards are identified by the United States National Telecommunications and Information Administration (NTIA) as formats that can meet the SBOM requirements for the US Executive Order.

5 Ways to Elevate the Role of the SBOM

Multiple high-profile software supply chain breaches and emerging government requirements create an imperative for SBOM adoption across organizations in all industries. Elevating the role of the SBOM and making it a foundation for software supply chain security will require collaboration between development, DevOps, security, and compliance teams.

1. Define Your Path to SBOM Adoption

Your organization should set policies and standards for the usage of SBOMs across development teams and business units. Here are some initial steps to take:

- » **Identify** which teams are currently generating or consuming SBOMs. Solicit their feedback on what's working and not working.
- » **Inventory** the tools being used and how they integrate with the DevOps toolchain. Get a sample of any SBOMs being used and compare the data elements and comprehensiveness.
- » **Define and document** a plan for how your organization will generate, manage, and use SBOMs. Your plan should include a central repository where you can store and manage SBOMs from across your organization for visibility and compliance.
- » **Share and evangelize** your plan with your software development teams and partners for feedback.
- » **Assess** the needs for any internal education about SBOMs and SBOM generation for your development teams. Develop any necessary training and documentation about SBOMs for delivery to your teams.

Putting in the tools, frameworks, and processes to generate, manage, and use high-quality and comprehensive SBOMs for the software in your supply chain ensures that you have ample coverage to improve your internal security vulnerability assessments, risk detection, component inventory changes, and compliance.

2. Define the SBOM Elements and Standards Needed

Familiarize yourself and your team with the different SBOM standards. Identify the SBOM standards that might be required by the consumers or customers of your software. Determine what standards are supported by your software suppliers, including open source communities.

Define the data elements needed in your SBOM for the internal use cases you are pursuing. For many organizations, the SBOM standards may not support all of the data elements you require.

Open source SBOM generation tools like [Syft](#) incorporate a richer set of data elements. With Syft, you can also generate SBOMs in other standards, like SPDX and CycloneDX to share with customers or consumers.

3. Automate SBOM Generation in your DevOps/DevSecOps Toolchain

With SBOMs becoming a critical foundation for securing the software supply chain, it is important to make SBOM generation a required and automated step in your development process.

In the case of cloud-native software, each time you build container images, use automated tooling to generate an SBOM for the image, including direct and transitive dependencies. As the container images are assembled into an application, there should be a complete SBOM that represents the full application. These image and application level SBOMs along with requisite security scans for vulnerabilities, malware, and secrets will enable your security team to fully assess the security posture of each software application, and continually monitor the components identified in the SBOM for new vulnerabilities.

4. Implement SBOM Management

Individual development teams may currently store the SBOMs they generate in the repositories they already use. However, in order to facilitate software supply chain security across your organization, you will need a centralized SBOM repository with SBOM management capabilities that include tagging, search, and reporting that is accessible across multiple teams. Your SBOM management system should include SBOMs coming from software suppliers, SBOMs generated internally, and the SBOMs you deliver to external software customers or consumers.

5. Watch for Unexpected SBOM Drift

The use of SBOMs for containerized applications provides a unique opportunity to watch for **SBOM drift**—unexpected changes in the contents of a software application—which can indicate potential tampering, new versions, or changes in dependencies..

Generating an SBOM creates a snapshot of the components of your container at a specific time during the development process. By generating an SBOM for each build and at each step in the development process, you can look for differences over time. Some of those differences might be expected, but any changes should be investigated to determine if they introduce new risk

For example, you generate an SBOM at the early stages of a build. You identify all of the commercial software, code, and open source software being used in your containers. As your build progresses, these components can change. New pieces can be introduced to the container by developers or the software used could have an updated version. These changes need to be identified and addressed to prevent new risk and vulnerabilities from being introduced to your finished product.

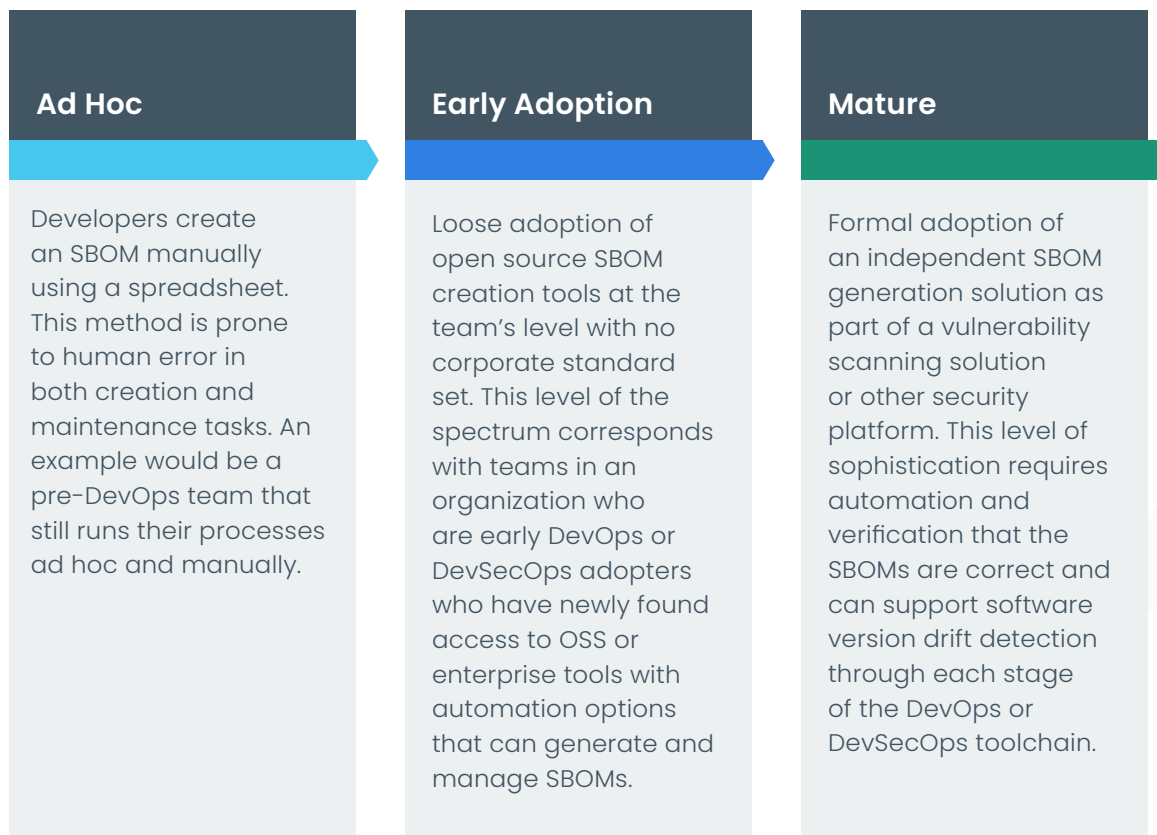
Embedding SBOMs in DevSecOps

A truly strong security posture is not limited to the responsibility of a single, cybersecurity team. Security must become a culture, where each group plays a role in ensuring the overall security of your organization and its products. This is especially evident for development teams and SBOMs are a key element in advancing your organization's DevOps to DevSecOps.

What is DevSecOps?

DevSecOps is the practice of integrating security checks at every phase of the software development cycle.

The first step to that level of collaboration between development and security is to chart where your organization is in its SBOM maturity. It's a spectrum



Security as a culture, with a focus on the transformation of DevOps to DevSecOps is the evolution required to secure modern, continuous development pipelines and the overall software supply chain.

SBOM Adoption via Executive Order

On May 12, 2021, President Joe Biden released the Executive Order (EO) on Improving the Nation's Cybersecurity¹ with a specific requirement for SBOMs.

For reference, here's the SBOM definition from the EO:

The term "Software Bill of Materials" or "SBOM" means a formal record containing the details and supply chain relationships of various components used in building software. Software developers and vendors often create products by assembling existing open source and commercial software components. The SBOM enumerates these components in a product. It is analogous to a list of ingredients on food packaging. An SBOM is useful to those who develop or manufacture software, those who select or purchase software, and those who operate software. Developers often use available open source and third-party software components to create a product; an SBOM allows the builder to make sure those components are up to date and to respond quickly to new vulnerabilities. Buyers can use an SBOM to perform vulnerability or license analysis, both of which can be used to evaluate risk in a product. Those who operate software can use SBOMs to quickly and easily determine whether they are at potential risk of a newly discovered vulnerability. A widely used, machine-readable SBOM format allows for greater benefits through automation and tool integration. The SBOMs gain greater value when collectively stored in a repository that can be easily queried by other applications and systems. Understanding the supply chain of software, obtaining an SBOM, and using it to analyze known vulnerabilities are crucial in managing risk.

The EO requires software publishers to provide an SBOM directly to the federal agency purchaser or publish the SBOM to a public website. It's a positive first step that opens the door for the automation of SBOM generation but also crystallizes the need for an SBOM to serve as a starting point for vulnerability scans and other security assessments.

While the EO does not directly apply to purchases of software by the commercial sector, this requirement may usher in SBOMs as a future compliance standard, leading some private sector organizations to take notice and begin implementation of SBOM requirements internally.

1. <https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/>

Summary: SBOMs Have A Critical Role in Securing the Software Supply Chain

In recent years we have seen threat actors shift their focus to third-party software suppliers. Rather than attacking their targets directly, they aim to compromise software at the build-level, introducing malicious code that can later be executed once that software has been deployed giving the attacker access to new corporate networks. Now, instead of taking down one target, supply chain attacks can potentially create a ripple effect that could affect hundreds, even thousands of unsuspecting targets. Open source software can also be an attack vector if it contains un-remediated vulnerabilities.

SBOMs are a critical foundation for securing against software supply chain attacks. By generating SBOMs into the development cycle, developers and security teams can identify and manage the software in their supply chain and catch these bad actors early before they reach runtime and wreak havoc. Additionally, SBOMs allow organizations to create a data trail that can provide an extended view of the supply chain history of a particular product.



About Anchore

Anchore is a leader in software supply chain security and enables organizations to protect cloud-native applications against software supply chain attacks. Anchore technology embeds continuous security and compliance checks at every stage of the software development process to prevent security risks from reaching production. Large enterprises and government agencies use Anchore solutions to generate a comprehensive software bill of materials, pinpoint vulnerabilities, identify malware and discover unprotected credentials that can lead to hacks and ransomware. With an API-centric approach, Anchore solutions integrate into the tools developers already use to detect issues earlier, saving time and lowering the cost to fix vulnerabilities. To learn more visit www.Anchore.com.



anchore

✉ info@anchore.com

🌐 anchore.com