Rezilion

COMPETITIVE ADVANTAGE SERIES WHITE PAPER

A Dynamic SBOM Isn't Just an Artifact, It's a Competitive Advantage

AS MORE ORGANIZATIONS CONCERN THEMSELVES WITH THE CONCEPT OF THE SOFTWARE BILL OF MATERIALS (SBOM), here's one question that might come up: can SBOMs provide modern enterprises with a competitive advantage? The answer is yes—if the SBOMs are created and managed the right way.

What is an SBOM anyway?

SBOMS ARE ARTIFACTS THAT CONTAIN A COMPREHENSIVE INVENTORY OF ALL COMPONENTS PRESENT IN A SOFTWARE, their dependencies and hierarchical relationships along with details of supply chain relationships of the various components. These software components are packages, files, libraries, and containers, as well as third party and open source components that are used by development teams to build software products. An SBOM is a machine-readable collection of metadata that provides a complete list of all the ingredients that go into a piece of software.

When creating software, teams in many cases rely on assembling open source and proprietary, commercial off-the-shelf (COTS) software components. An SBOM enumerates these components, offering a way to trace the origin of the software's makeup. There are many benefits to SBOMs, such as the reduction of security, license, and compliance risk.

The biggest limitation of the SBOM today is that they are static documents, while software is a dynamic entity. Software is constantly changing, be it development or production and SBOMs need to reflect such change, preferably in real time.

INSIDE

Software is CONSTANTLY CHANGING, be it development or production and SBOMs need to REFLECT SUCH CHANGE, preferably in real time.

If SBOMs cannot be easily updated, then the user cannot really use them in a meaningful way and their value is greatly diminished. It is important to understand that for effective use and adoption of SBOMs its management has to be automated and efficient. The current manual and static approach to SBOMs, though valid, will not fulfill its true potential to help organizations be more secure.

This whitepaper describes why SBOMs are quickly becoming an important tool to address the growing challenges of software security and reliability, why the limitations of current SBOMs can be addressed by a dynamic SBOM, and how organizations can use dynamic SBOM as a competitive advantage.

SBOMs Are Gaining Momentum—But Why?

One of the big appeals of an SBOM is its broad value to a range of users. An SBOM is equally important for organizations developing software, for entities buying such software, and for enterprises who actually end up using that software for their daily operations. All of these users address specific use cases namely security, compliance, and risk.

For instance, an SBOM can allow developers who rely on open source and third-party components to be sure that the components are up-to-date, that they are not in violation of any licenses, and that they can respond to new vulnerabilities. Software buyers can leverage an SBOM to perform a vulnerability analysis when evaluating the security risks of a software product.

The National Telecommunications and Information Administration (NTIA) report, Roles and Benefits for SBOM Across the Supply Chain, provides numerous examples of how SBOMs can be beneficial.

For producers of software, the benefits of an SBOM can include:

- Reduced unplanned, unscheduled work
- Reduced code bloat and an understanding of the dependencies within broader complex projects
- Knowledge of and compliance with license obligations
- The ability to monitor components for vulnerabilities
- Easier code reviews
- Awareness of banned software components
- The ability to provide an SBOM to a customer

For software buyers, the potential benefits of an SBOM include:

- The ability to identify potentially vulnerable components
- A more targeted security analysis
- The ability to verify the sources of software components
- License analysis and easier compliance with policies
- Awareness of end-of-life components
- An understanding of the software's integrations
- Pre-purchase and pre-installation planning

And for software operators, the possible benefits of an SBOM include:

- Being able to quickly evaluate whether they are using a particular component
- The ability to make more informed risk-based decisions
- Availability of alerts about potential end-of-life for products or components
- Ability to support compliance and reporting requirements
- Reduction of costs through a more streamlined and efficient administration

Understanding SBOM Formats

THERE ARE SEVERAL FORMATS THAT ARE BEING DISCUSSED in order to export and share SBOMs with other stakeholders or interested parties.
The top three formats are listed below.

1. **Software Package Data Exchange (SPDX)** is an open standard for an SBOM from the Linux Foundation that enables the presentation of components, licenses, copyrights, security references, and other data relating to software.

2. **CycloneDX** is an SBOM standard created by OWASP that features significant cybersecurity capabilities aimed at driving innovation and increasing operational efficiency of the SBOM across the software supply chain. The latest version adds the ability to communicate vulnerabilities and their exploitability.

3. **Software Identification** (SWID) Tags from NIST provide a transparent way for organizations to track the software installed on their managed devices. SWID Tag files contain descriptive information about a specific release of a software product.

SBOM has emerged as a MAJOR BUILDING BLOCK in SOFTWARE SECURITY and software supply chain risk management.

The Linux Foundation has been working on SPDX for communicating SBOM information including components, licenses, copyrights, security references, and other metadata relating to software.

The first version of the SPDX specification was intended to make compliance with software licenses easier. Subsequent versions added capabilities intended for other use cases such as being able to contain references to known software vulnerabilities.Recent versions fulfill the NTIA's Minimum Elements for a SBOM.

SPDX reduces redundant work by providing common formats for organizations to share important software data, thereby streamlining and improving compliance, security, and dependability, according to the foundation. The grass-roots effort includes representatives from a variety of organizations, including software, systems and tool vendors; foundations and systems integrators.

SBOMs have taken on greater importance in the public and private sectors in large part because they were specified among the requirements of an executive order (EO 14028) on improving the nation's cybersecurity, released by the White House in May 2021.

The requirement of the order is that organizations provide buyers of software products with an SBOM for each product directly or by publishing it on a public website.

The Commerce Department has said an SBOM provides organizations that produce, purchase, and operate software with information that improves their understanding of the supply chain. That in turn offers multiple benefits including the potential to track known emerging vulnerabilities and risks.

The U.S. Cybersecurity & Infrastructure Security Agency (CISA), a part of the Department of Homeland Security that leads national efforts to understand, manage, and reduce risk to the cyber and physical infrastructure, said the SBOM has emerged as a major building block in software security and software supply chain risk management.

The majority of organizations are expected to produce or use SBOMs in 2022 according to a February 2022 report released by the Linux Foundation, a nonprofit that provides open source products. The foundation's research surveyed 412 organizations worldwide as part of a study on organizational SBOM readiness and adoption and found that 78% expect to create or use SBOMs this year. That's an increase from 66% the year prior. Many organizations concerned about application security are making SBOMs a cornerstone of their cybersecurity strategy, according to the Linux Foundation report. Among other key findings from the survey: 82% of the respondents are familiar with the term software bill of materials; 76% are actively engaged in addressing SBOM needs; and 47% are currently producing or consuming SBOMs.

What these statistics tell us is that there is a growing awareness, need, and demand for SBOMs.

A DYNAMIC SBOM is a comprehensive and continuously updated view of the composition of an ENTIRE SOFTWARE ATTACK SURFACE and associated risk.

The top three benefits of SBOMs, according to the survey, are that they make it easier for developers to understand dependencies across components in an application, monitor components for vulnerabilities, and manage license compliance.

Is an SBOM Enough for Security?

There is no doubt SBOMs offer great benefits to drive security. At the same time, SBOMs have their own challenges. Indeed, the challenges with SBOMs are real. Some of the SBOM challenges include:

- SBOMs are manually created. Considering the number and frequency of changes needed, this is a labor-intensive and expensive process, especially because changes need to be tracked in real-time for SBOMs to be effective.
- SBOMs can be difficult to build and maintain. Due to the iterative nature of software, they need to be updated, whenever changes are made to any components of a software. This can include updates of the code, the release of vulnerability patches, the introduction of new features, and other modifications. All of these alterations need to be tracked in real time if SBOMs are to be effective.
- SBOMs need to be auditable. All of the data in an SBOM, including every version number and license, needs to be auditable. The data must be provided by a reputable source and verifiable by a third party.
- SBOMs are static, which is not ideal for an environment in which change is a constant. As noted, once an SBOM is developed it needs to be updated whenever changes are made to software components.
- SBOMs by itself are not enough, they need context in order to understand if a vulnerability is exploitable in their environment. In order to gain an understanding of how context applies to an SBOM, organizations need to consider the Vulnerability Exploitability eXchange (VEX). The primary use cases for VEX are to provide software users with additional information about whether a product is impacted by a specific vulnerability in an included component and, if affected, whether they need to take actions to remediate the issue. In many cases a vulnerability in a component will not be exploitable in the final product for various reasons. When software providers issue a VEX, that reduces or eliminates the effort needed by software users to investigate non-exploitable vulnerabilities that don't affect a software product. They can notify users of a non-exploitable vulnerability by email or other means. A VEX is machine-readable, which enables automation and supports integration into broader tooling and processes.

Despite these and other challenges, there is no doubt about the value of SBOMs in efforts to keep software as secure and reliable as possible, and to prevent vulnerabilities from enabling significant cybersecurity threats, such as ransomware.

Organizations that develop and purchase software need to accept the concept that for SBOMs to be really effective and useful, the creation and maintenance of SBOMs must be dynamic rather than static. In addition, organizations need to deploy technology that gives them the ability to have a dynamic SBOM that incorporates updates and changes automatically whenever changes occur.

Moving to a Dynamic Future

MOVE OVER SBOM, HERE COMES THE DYNAMIC SBOM. As the name suggests, dynamic SBOMs are a more enhanced version of static SBOMs. A dynamic SBOM is a comprehensive and continuously updated view of the composition of an entire software attack surface and associated risk.

Dynamic SBOMs offer several benefits:

- **They are continuous, fluid documents** that can accommodate changes as they occur, unlike static SBOMs that look at a single point in time. Organizations need that fluidity because of the changing nature of software.
- Another benefit is that the **data in these SBOMs is itself dynamic**; not just one kind of a file or package, but an array of data coming from multiple sources. That's important because in any software environment there is not just one kind of data or source. Dynamic SBOMs are large, comprehensive resources that cover everything about the software.
- A third aspect of dynamic SBOM that makes it appealing is its **ability to cover a broader range of the environmen**t, versus static files that are limited in scope. When trying to document the entirety of a software product, the broader the information and range of sources, the better.
- Another important aspect of dynamic SBOM that is worth noting is the importance of **built in vulnerability contex**t. While SBOMs provide a lot of the information organizations need in order to know about software products and components, this data in and of itself is not enough. For SBOMs to be really effective, they need to have access to context as well. Dynamic SBOMs offer built in context and do not require a separate document which adds complexity.

Dynamic SBOM	SBOM
Real time, continuous updates	Static point-in-time document
Automatically generated at predefined stages	Not fully automated
Covers the entire environment	Generally related to a software or application
Contextualization built in	Requires VEX for added context

Not all software products or vulnerabilities are equally significant. Depending on the scenario, organization, and other factors, a given software product might have a broad range of impacts or no impact at all. And of course, some vulnerabilities might be major problems while others are not a risk at all.

Dynamic SBOMs: A Competitive Advantage For Your Product Security Lifecycle

LOOKING AHEAD, as SBOMs become increasingly common their future must be dynamic. If not, the administrative burden of keeping these resources up-to-date and accurate will become overwhelming. The move to dynamic SBOMs will eventually become a requirement, in all likelihood, particularly for companies that build and update lots of software products on a regular basis.

Dynamic SBOMs will be integrated into the security lifecycle of software products. They will automatically be created at pre-defined stages of code development, which is vital given that software providers in many cases do not have any idea about what vulnerabilities might be present in their products and which of the vulnerabilities can be exploited.

Dynamic SBOMs form an integral part driving security through the different stages of the Software Development Lifecycle (SDLC).

- Dynamic SBOMs will be created for software under development which can be used by developers to prioritize and remediate vulnerabilities that are actually exploitable, saving them time and bringing focus back to building
- Dynamic SBOMs will be used to drive how decisions about security and patches are made based on risk and validation
- Dynamic SBOMs will be used to support the build process by acting as security gates with the objective to ensure that SBOMs from a previous stage can be verified for any changes that could violate any security and compliance policies, as well as monitor for known vulnerabilities.
- Its dynamic nature lends itself to be a security tool to support on-going and continuous monitoring for vulnerabilities.

Dynamic SBOMs will help development and security teams to discover, prioritize, remediate vulnerabilities, and release products faster so that they can stay competitive.

Log4j a Case Study for Dynamic SBOM

CONSIDER THE RECENTLY DISCOVERED VULNERABILITY WITH LOG4J, a Java-based logging utility that is part of the Apache Logging Services and one of several Java logging frameworks. In December 2021, security researchers discovered a zero-day security vulnerability involving arbitrary code execution in Log4j.

Security experts have called the flaw one of the biggest and most critical vulnerabilities discovered in recent years. Clearly, most vulnerabilities do not have this level of impact. New software vulnerabilities are constantly emerging, however, most are without the knowledge of the companies that developed the software and the countless organizations using it.

Vulnerabilities unfortunately are a part of the software development process. Mistakes happen, even with the most diligent and skilled development teams. The ability to identify and address the most serious vulnerabilities—and document them in SBOMs in a timely manner—is extremely important.

Building security into the development lifecycle through initiatives such as DevSecOps has never been more important for teams, and integrating SBOMs into the lifecycle and producing them automatically at various stages of development will become the standard going forward.

Indeed, the future of SBOMs is dynamic and it will be a competitive advantage for organizations that use it to drive security within their product lifecycle.

ABOUT REZILION

Rezilion's platform automatically secures the software you deliver to customers. Rezilion's continuous runtime analysis detects vulnerable software components on any layer of the software stack and determines their exploitability, filtering out up to 95% of identified vulnerabilities. Rezilion then automatically mitigates exploitable vulnerabilities across the SDLC, reducing vulnerability backlogs and remediation timelines from months to hours, while giving DevOps teams time back to build.

Learn more about Rezilion's software attack surface management platform at www.rezilion.com and get your 30 day free trial www.rezilion.com.

Enhance your vulnerability management strategy and eliminate software risk today with a Dynamic SBOM. For more information, visit www.rezilion.com/platform/dynamic-sbom/ and to sign up fora free 30-day trial at www.rezilion.com/get-started/.