Rezilion

EBOOK

The Future of Software Security is the SBOM

INTRODUCTION

1. Why SBOMs are Important

2. Why Security Leaders Should Care

3. Building and Maintaining an SBOM

4. SBOMs Need to be Dynamic

5. The Importance of Context

Conclusion—Dynamic SBOM as a Competitive Advantage

This ebook describes:

- Why SBOMs are so important today;
- Why security leaders need to care about them;
- What's involved in building and maintaining SBOMs;
- The importance of context when creating and using these records;
- Why SBOMs need to be dynamic in order to be most effective;
- How SBOMs can provide a competitive advantage for companies

IF YOU'RE NOT FAMILIAR WITH A SOFTWARE BILL OF MATERIALS (SBOM) and what it means in terms of providing better security for software products, it is time to get up to speed in a hurry.

For those not familiar with the concept, an SBOM is a formal record that contains the details and supply chain relationships of the various components used to build a software product. It's essentially a list of the ingredients that go into software, including many open source components, and where they come from.

These records can help organizations or individuals that purchase software to avoid using products that could cause harm through vulnerabilities and other issues. SBOMs can also be helpful to developers of software, who oftentimes leverage open source and third-party components to create products, by enabling to make sure the components are up-to-date and can respond quickly to new vulnerabilities.

INTRODUCTION

1. Why SBOMs are Important
2. Why Security Leaders Should Care
3. Building and Maintaining an SBOM
4. SBOMs Need to be Dynamic
5. The Importance of Context

Conclusion—Dynamic SBOM as a Competitive Advantage

SBOM IS NOT THE LATEST BUSINESS BUZZWORD—OR ACRONYM, IN THIS CASE. The idea of using these resources is gaining a lot of momentum in the public and private sectors, probably in large part because of the importance of software in the age of digital business and the potential damage software vulnerabilities can cause.

Consider the case of the recently discovered vulnerability with Log4j, a Java-based logging framework. Security researchers in December 2021 identified a zero-day security vulnerability involving arbitrary code execution in Log4j. Security experts said the flaw was one of the biggest and most critical discovered in recent years.

It's a dramatic example of the fact that new software vulnerabilities are constantly being found, in many cases without the software producers being aware of them until they are discovered and the software is already in use broadly.

Vulnerabilities are part of the software development process—especially with the rush to get software out the door as quickly as possible. Having the ability to identify and address the most serious vulnerabilities quickly, and document them in an SBOM in a timely manner, is vital. Building security into the development lifecycle is extremely important, as is integrating SBOMs into that lifecycle and producing them automatically at various stages of development.

Clearly, the U.S. federal government is embracing the idea of using SBOMs. The interest began some time ago, with the Cyber Supply Chain Management and Transparency Act of 2014. The Congressional legislation proposed to require government agencies to obtain SBOMs for any new products they bought. The bill would have required obtaining SBOMs for any software, firmware, or product in use by the U.S. government. And while the legislation ultimately did not pass, it brought awareness of the idea to government officials.

Fast forward to May 2021, when the announcement of the U.S. Executive Order on Improving the Nation's Cybersecurity included a requirement of software providers to include purchasers with SBOMs for each product. The executive order from the White House brought a new level of awareness and sense of urgency about the need for SBOMs.

Most organizations are expected to produce or use SBOMs in 2022, according to a February 2022 report released by the Linux Foundation, a nonprofit organization 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 up from 66% the previous year.

Organizations concerned about application security are making SBOMs a cornerstone of their cybersecurity strategy, according to the report, "The State of Software Bill of Materials and Cybersecurity Readiness." Other key findings from the survey are that 82% of respondents are familiar with the term "software bill of materials"; 76% are actively engaged in addressing SBOM needs; and 47% are now producing or using SBOMs.

The top three benefits of SBOMs are:

- They make it easier for developers to understand dependencies across components in an application;
- They make it easier to monitor components for vulnerabilities;
- They make it easier to manage license compliance.

INTRODUCTION

1. **Why SBOMs are Important**

2. Why Security Leaders Should Care

3. Building and Maintaining an SBOM

4. SBOMs Need to be Dynamic

5. The Importance of Context

Conclusion—Dynamic SBOM as a Competitive Advantage

GIVEN THAT SBOMS CAN BE A POWERFUL means to enhance software security, CISOs and other cybersecurity leaders should be promoting their use within their organizations and among their organizations' software providers and business partners.

SBOMs are formal records that include the details and supply chain relationships of all the various components used to build software, so they provide extensive histories of the software that can help identify potentially risky components or sources.

They allow software developers who rely on open source and third-party components to ensure that components are up to date and able to respond to newly discovered vulnerabilities. At any time, software providers might not be aware of vulnerabilities in their software and or whether they are exploitable.

That fact was brought to light in dramatic fashion with the Log4j incident, and given that new vulnerabilities are constantly arising, it's virtually impossible to know about all vulnerabilities in an environment at any given time.

How important are SBOMs to software security? The U.S. Dept. of Commerce has said that an SBOM creates a foundational data layer on which security tools, practices, and assurances can be built. The essential pieces that support basic SBOM functionality serve as the foundation for an evolving approach to software transparency, the department said.

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

Software development teams are being encouraged to build security into the development process via efforts such as DevSecOps, and one of the ways they can achieve this is by referring to SBOMs for potential vulnerabilities. They can then take into account the context of the vulnerabilities and if necessary, fix them before moving ahead with the development process.

With this process, product security is baked into the development process, which can end up saving organizations a lot of money by avoiding the need for additional development costs. More importantly, they can avoid the possible financial impact of security breaches, regulatory fines, and other negative effects of insecure software.

Software plays such an important role in digital business, and to neglect security by failing to address exploitable vulnerabilities can put organizations at great risk. SBOMs can play a major role in helping development teams enhance the cybersecurity of software. Because of this, security leaders should make it a high priority to leverage these documents.

INTRODUCTION

1. Why SBOMs are Important

2. **Why Security Leaders Should Care**

3. Building and Maintaining an SBOM

4. SBOMs Need to be Dynamic

5. The Importance of Context

Conclusion—Dynamic SBOM as a Competitive Advantage

INTRODUCTION

1. Why SBOMs are Important

2. Why Security Leaders Should Care

3. **Building and Maintaining an SBOM**

4. SBOMs Need to be Dynamic

5. The Importance of Context

Conclusion—Dynamic SBOM as a Competitive Advantage

IT'S IMPORTANT THAT SECURITY AND BUSINESS LEADERS as well as development teams understand that building and maintaining an SBOM is not easy; in fact, it can be a significant challenge.

These are formal records that consist of the minute details and supply chain relationships of all the components used in the creation of a software product.

For some applications or operating systems, putting together an SBOM can be a complex undertaking that takes time. It involves gathering lots of information from an array of sources. After an SBOM has been created, the document needs to be updated whenever changes are made to any of the software components. Changes might include updates in code, vulnerability patches, new features, and other modifications in the software or sourcing.

Information integrity is vital with SBOMs, so everything included in the record must be auditable. That includes all of the various version numbers and licenses. All of the data needs to be provided by a reputable source and it must be verifiable by a third party.

Today, most SBOMs are static documents that do not automatically incorporate updates. All of the SBOM maintenance and updating is typically done on a manual basis. Because modification can happen at any time, this is a continuous, labor-intensive and tedious process. The fact that changes need to be tracked in real-time for the SBOM to be effective makes the challenge even greater.

As an organization's environment continues to change, the organization needs to create new SBOMs. This results in an-ever increasing number of SBOMs, and this also adds to the maintenance challenge.

The move to DYNAMIC SBOMs will most likely become a REQUIREMENT at some point, especially for those organizations that regularly BUILD, UPDATE, and use many software products.

INTRODUCTION

1. Why SBOMs are Important

2. Why Security Leaders Should Care

3. Building and Maintaining an SBOM

4. **SBOMs Need to be Dynamic**

5. The Importance of Context

Conclusion—Dynamic SBOM as a Competitive Advantage

ONE WAY TO ADDRESS THESE CHALLENGES IS to embrace the idea that SBOMs need to be dynamic rather than static. Organizations can implement technology tools that enable them to have dynamic SBOMs that incorporate updates automatically whenever changes occur in software or components.

As more and more SBOMs are created and maintained, it makes sense that their future must be dynamic. If not, the administrative burden of managing these resources will become overwhelming for many organizations. The move to dynamic SBOMs will most likely become a requirement at some point, especially for those organizations that regularly build, update, and use many software products.

SBOMs in the future will also be integrated into the security lifecycle of software, and will be created automatically at pre-defined phases of code development. This is vital, considering that many software providers are not aware of what vulnerabilities might exist in their products and which of those flaws could be exploitable.

The dynamic SBOM has major implications for software security. New vulnerabilities are being discovered all the time, and oftentimes the companies that develop the software are not aware of them until they're discovered by others. If cybercriminals are able to exploit these flaws before they are fixed, there's a potentially serious cybersecurity risk.

Given that vulnerabilities are a constant with software development, organizations need timely information about weaknesses in software components. Integrating dynamic SBOMs into the development lifecycle and producing these documents automatically at various stages of development is important.

Dynamic SBOMs offer several benefits to producers, buyers, and users of software. For one thing, they are continuous, fluid documents that can incorporate changes as they happen. For another, the data in these SBOMs is itself dynamic, with various types of files coming from multiple sources. And dynamic SBOMs have the ability to cover a broad range of the environment versus static files that are limited in scope.

WHILE AN SBOM PROVIDES IMPORTANT INFORMATION organizations need to know about software components, this data can't do much good in a vacuum. In order for the SBOM to be truly effective it must have context as well.

Software products and vulnerabilities will have a range of impacts on organizations. In some cases, they will have little or no impact, while in others they could have a major impact. It all depends on factors such as the type of company, application, severity of the vulnerability, etc. That's why context is so important.

For an understanding of how context applies to an SBOM, consider the Vulnerability Exploitability eXchange (VEX), which was developed as part of the National Telecommunications and Information Administration (NTIA) Multistakeholder Process for Software Component Transparency. VEX was developed to fill a particular need regarding the use of SBOMs, NTIA says.

NTIA, an agency of the U.S. Department of Commerce, says the main use cases for VEX are to provide software users, developers, service providers, and others with additional information on whether a product is affected by a specific vulnerability in one of the components and, if affected, whether they need to take action to remediate.

Software suppliers can issue a VEX to reduce the effort expended by software users on investigating non-exploitable vulnerabilities that don't affect a software product. A VEX is an assertion about the status of a vulnerability in specific products, the agency says.

Although software suppliers can notify users of a non-exploitable vulnerability by email or other means, NTIA says, a VEX is machine-readable, and this enables automation and supports integration into broader tooling and processes. This will likely allow users to take a much more targeted approach to finding and fixing vulnerabilities in software.

Concepts such as VEX provide the needed context that gives SBOMs more value for organizations looking to improve software security.

INTRODUCTION

1. Why SBOMs are Important

2. Why Security Leaders Should Care

3. Building and Maintaining an SBOM

4. SBOMs Need to be Dynamic

5. **The Importance of Context**

Conclusion—Dynamic SBOM as a Competitive Advantage

The biggest upside of SBOMs from a strategic standpoint is that they can provide companies with a competitive advantage.

Perhaps the biggest upside of SBOMs from a strategic standpoint is that they can provide companies with a competitive advantage—if they are created and managed the right way.

These software records can add value for organizations in multiple ways. For one thing, they can enhance the security of software. Enterprises can use an SBOM to perform a vulnerability analysis and evaluate the risk a product presents.

This helps organizations understand what goes into the product and how that might impact security.

SBOMs can enable software developers who rely on open source and third-party components to ensure that the components are up-to-date and come from reliable sources. This can save time and cost on the backend by avoiding the need for revisions to fix vulnerabilities. And given the federal mandate to use SBOMs, any company wanting to provide software to U.S. government agencies needs to produce SBOMs.

By developing and maintaining SBOMs that are dynamic, organizations that produce, buy, and use software will have assurances that they are doing the best they can to minimize the risk and maximize the value of software products.

INTRODUCTION

1. Why SBOMs are Important

2. Why Security Leaders Should Care

3. Building and Maintaining an SBOM

4. SBOMs Need to be Dynamic

5. The Importance of Context

Conclusion—Dynamic SBOM as a Competitive Advantage

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 for a free 30-day trial at www.rezilion.com/get-started/.