# A Shared Vision of Software Bill of Materials (SBOM) for Cybersecurity





Communications Security Establishment
**Canadian Centre for Cyber Security**

Centre de la sécurité des télécommunications
**Centre canadien pour la cybersécurité**


















**Publication: September 3, 2025**

**U.S. Cybersecurity and Infrastructure Security Agency**
**U.S. National Security Agency**
**Australian Signals Directorate's Australian Cyber Security Centre**
**Canadian Centre for Cyber Security**
**Czech National Cyber and Information Security Agency**
**French Cybersecurity Agency**
**Germany's Federal Office for Information Security**
**Indian Computer Emergency Response Team**
**Italy's National Cybersecurity Agency**
**Japan's Ministry of Economy, Trade and Industry**
**Japan's National Cybersecurity Office**
**Netherland's National Cyber Security Centre**
**New Zealand's National Cyber Security Centre**
**Poland's Research and Academic Computer Network**
**Cyber Security Agency of Singapore**
**Slovakia's National Security Authority**
**Republic of Korea's National Intelligence Service/National Cyber Security Center**
**Korea Internet and Security Agency**

# Introduction

This guidance, authored by the U.S. Cybersecurity and Infrastructure Security Agency (CISA) and the following partners[1], presents a shared vision of Software Bill of Materials (SBOM) and the value that increased software component and supply chain transparency can offer the global community.

- U.S. National Security Agency (NSA)
- Australian Signals Directorate's Australian Cyber Security Centre (ASD's ACSC)
- Canadian Centre for Cyber Security (Cyber Centre)
- Czech National Cyber and Information Security Agency (NÚKIB)
- French Cybersecurity Agency (ANSSI)
- Germany's Federal Office for Information Security (BSI)
- Indian Computer Emergency Response Team (CERT-In)
- Italy's National Cybersecurity Agency (ACN)
- Japan's Ministry of Economy, Trade and Industry (METI)
- Japan's National Cybersecurity Office (NCO)
- Netherlands' National Cyber Security Centre (NCSC-NL)
- New Zealand's National Cyber Security Centre (NCSC-NZ)
- Poland's Research and Academic Computer Network (NASK)
- Cyber Security Agency of Singapore (CSA)
- Slovakia's National Security Authority (NBÚ)
- Republic of Korea's National Intelligence Service/National Cyber Security Center (NIS/NCSC)
- Korea Internet and Security Agency (KISA)

The authoring organizations aim to further inform producers, choosers (i.e., procuring organizations), and operators of software about the advantages of integrating SBOM generation, analysis, and sharing into security processes and practices. Widespread adoption of SBOM will strengthen security, reduce risk, and decrease costs.

Most modern software is comprised of software components, modules, and libraries from open source and proprietary software worlds, rather than developers creating it from scratch. As concerns about the security and provenance of software grow, it is critical to understand the risks in the software's supply chain—including the risks of the underlying software components. The first step to addressing these risks is to increase transparency. This is especially important for software in critical infrastructure and systems that carry out essential functions that affect public safety.

[1] Hereafter referred to as the authoring organizations.

Software component and supply chain transparency are fundamental for a more secure software ecosystem, as identified in the international Secure by Design efforts.[2] **The authoring organizations understand the value of SBOM in securing the software supply chain and recognize the need for greater transparency in software development.**

# What is an SBOM?

An SBOM is a formal record of the details and supply chain relationships of various components used in building software. It can also be thought of as a "list of ingredients" for software. SBOMs have emerged as a key tool to address challenges in securing software because of the visibility they provide into the components of software.

An SBOM should be machine-processable in a widely used format and contain enough information about the open source and proprietary components in the software to correlate with other data sources, such as vulnerability databases and security advisories. Automation is a key goal for SBOM generation and use. By analyzing the SBOM data, a user should be able to determine whether a given component is present in a piece of software. SBOM data can also provide insight into the source of components. Sharing SBOMs downstream along the supply chain allows organizations to form a more complete picture of the software and to respond to information that may indicate new risks.

# The SBOM Value Proposition

SBOMs enable greater visibility across an organization's software supply chain and enterprise system by documenting information about software dependencies. Organizations can leverage this transparency to increase the efficacy of risk management practices, particularly vulnerability management and supply chain management, improve software development processes, and support an organization's license management.

## Risk Management Practices

### Vulnerability Management

Identifying and responding to vulnerabilities is a key step in secure software development[3] and for limiting risk throughout the software lifecycle. With SBOM data, software producers and operators can map the software's dependencies to relevant lists of existing vulnerabilities and use continuous monitoring to track

[2] The Secure by Design initiative seeks to foster a cultural shift across the technology industry to normalize the development of products that are secure out of the box. During the design phase of a product's development lifecycle, companies should implement Secure by Design principles to significantly decrease the number of exploitable flaws before introducing them to the market for widespread use or consumption. For more information, see CISA's Secure by Design webpage.

[3] "SP 800-218, Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities," U.S. National Institute for Standards and Technology, February 2022, https://csrc.nist.gov/pubs/sp/800/218/final.

new vulnerabilities that may arise. SBOM's transparency increases the speed and efficiency of responding to vulnerabilities with mitigations that are more tailored and focused on addressing specific risks. By using security advisories provided by software suppliers (ideally, in an automation-friendly format like the Common Security Advisory Framework[4]), an organization can also verify if the vulnerability makes their software vulnerable to exploitation. If mitigation measures are needed, an organization can conserve resources by deploying a more targeted set of measures based on the SBOM data and security advisories.

### SBOM Improved Organizations' Visibility into How They May Have Been Impacted by Log4Shell

In December 2021, an arbitrary code execution vulnerability (commonly known as Log4Shell) was discovered in Apache Log4j—an open source log output library. This library was widely used as a standard module for log output in Java systems around the world. Security organizations soon observed widespread exploitation. Because Log4j was usually used as a transitive dependency (a dependency of other dependencies), it was not always easy to identify. Organizations without SBOM capability often had to engage in time-consuming manual searches and risked remaining vulnerable. Organizations with SBOMs were able to report a relatively straightforward and efficient response (see **Figure 1**).

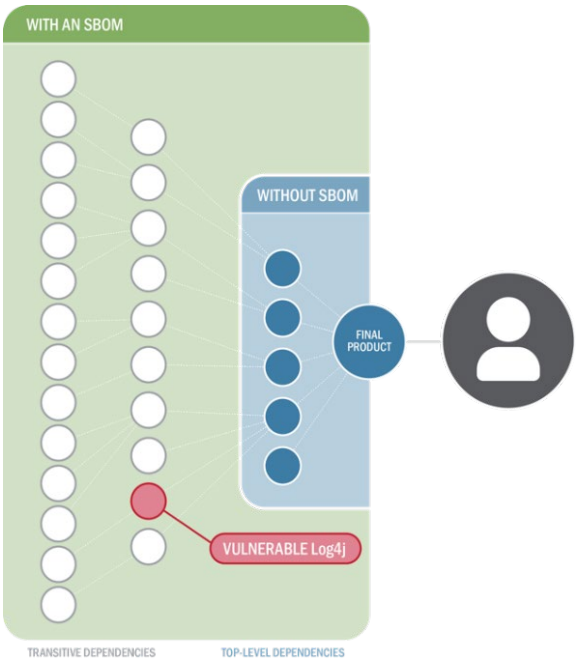


*Figure 1. The Effect of SBOM on the Log4j Vulnerability*

## Supply Chain Risk Management

An organization may have organization-, industry-, or government-specific policies that require them to report or limit the presence of certain types of software components. These requirements may be related to where, how, by whom, etc., a software component was developed. Organizations can leverage this data when making purchase decisions to better align their software selection with known supply chain concerns. Software suppliers, in turn, will have incentives to integrate supply chain risk data into their development practices to minimize friction from their downstream users.

[4] "Common Security Advisory Framework (CSAF)," GitHub, accessed June 17, 2025, https://github.com/oasis-tcs/csaf.

When all participants along the supply chain have an SBOM for a piece of software, the time to identify and respond to vulnerabilities can be reduced significantly (see **Figure 2**). Without an SBOM, each actor is dependent on upstream suppliers for notification that the vulnerability impacts their software.

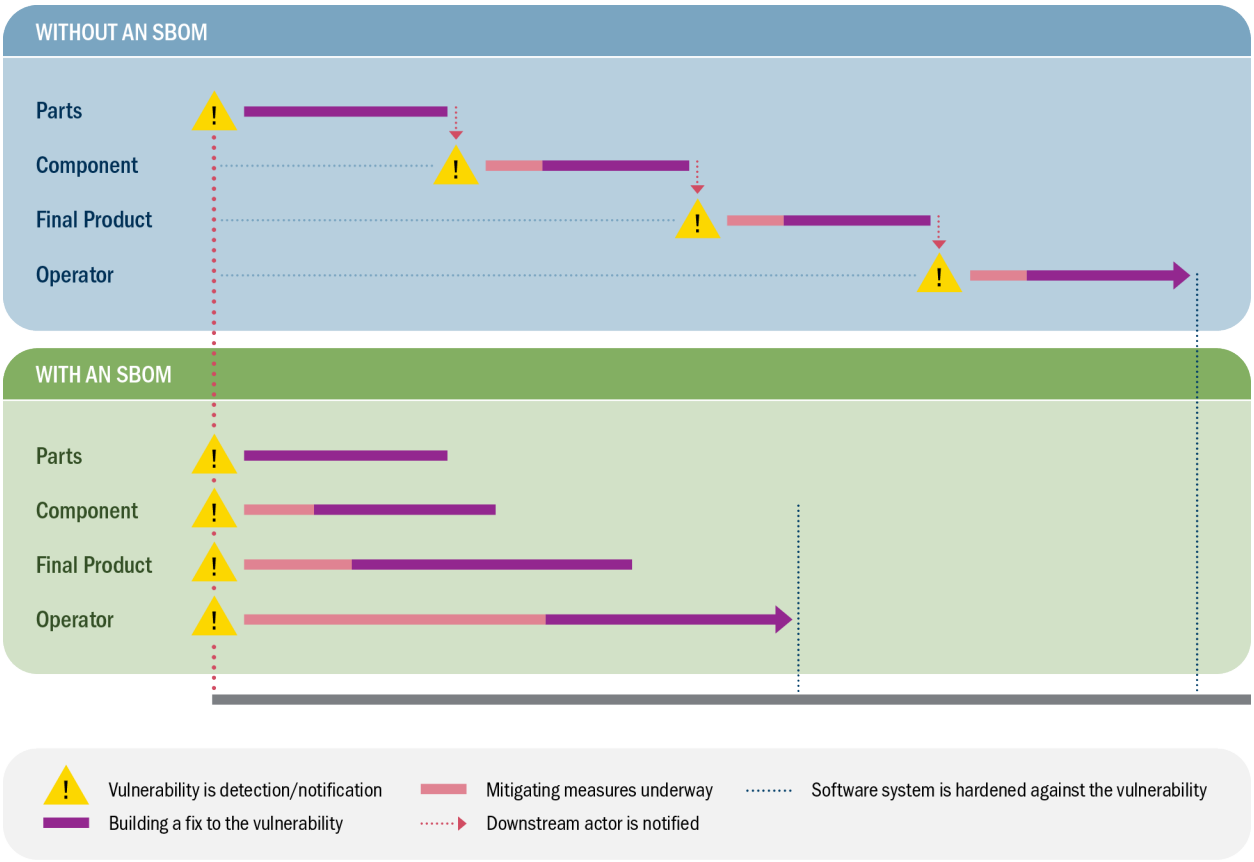


*Figure 2. SBOM Reduces Total Vulnerability Response Time*

## Improved Software Development Processes

Implementing and using SBOMs throughout the software development lifecycle lowers costs associated with managing components, downtime spent responding to vulnerabilities, and identifying the risk of end-of-life/end-of-support software components. Organizations save time by using SBOM data to identify components that have already been approved according to organization requirements and policies. Addressing issues with software components, such as known vulnerabilities or license concerns, early in the software development lifecycle reduces unplanned or unscheduled work normally required for vulnerability responses once the software has been deployed. Identifying a software component's end-of-life or security support information during the software development process allows developers time to plan for the end-of-life/end-of-support date. Post-deployment monitoring of SBOMs can help organizations identify components that have become vulnerable over time allowing them to quickly and efficiently build and deploy fixes as part of an effective DevOps development process. Software developers can use SBOM data to inform their decisions at each stage of the software development lifecycle to develop more secure software.

### Managing Software Licenses

SBOM data better equips an organization to identify license information for software components and to use the software component as allowed by the license. For example, tracking licenses and obligations will likely become both more complex and more important as open source software becomes increasingly embedded throughout the software ecosystem. Violating an open source license can result in sale suspension or recall of software, fines, and reputational damage. These consequences can impact downstream organizations through abrupt added costs or unplanned end of support.

## Who Should Care About SBOM?

Producers, choosers, and operators of software across the software ecosystem benefit from the increased transparency from SBOM data. Organizations may simultaneously take on the role of software producer and chooser, chooser and operator, or any combination of those roles. Similarly, individuals or offices within an organization may also act as a software producer, chooser, or operator, or any combination of those roles. National cybersecurity organizations also benefit from the adoption and implementation of SBOM.

### Producers

In an organization that produces software, developers gain better and more automated tracking of upstream components in their supply chain, which enables them to choose the best suited components for their needs. The security teams that support software development, such as the application security teams or the more formal product security teams, can better respond to vulnerability information and strategically deploy mitigation measures. License data provided in an SBOM can help a legal team manage license concerns more effectively. An organization that produces software can also benefit from reducing code bloat and predicting potential support or quality issues.

### Choosers

The increased transparency from SBOMs empowers an organization acquiring software (or those responsible for acquiring or selecting software for their organizations) to make risk-informed decisions about introducing a software component into the organization's system. Whether a software supplier can provide an SBOM for their software can itself help inform the software procurement decision.

### Operators

Once the software has been deployed, visibility across the network and system allows operators of software to better understand exposure to newly identified risks. An operator can more easily triage which software—and which missions supported by that software—must be addressed in response to new vulnerability information. When a patch or update is not available, a security team can take other precautions, including compensating controls, network isolation, or targeted use of threat intelligence.

## National Cybersecurity Organizations

SBOMs can play a crucial role in managing a country's cybersecurity risk. Beyond using SBOMs as part of the procurement process, governments are beginning to explore how enhanced transparency can improve overall risk posture. This includes assisting in coordinated vulnerability disclosure and response, as well as contributing to the evaluations and policies of market surveillance authorities, specific regulators, or certification approaches. Cybersecurity organizations can also benefit from software supply chain data when coordinating with other entities, such as sector coordinating bodies. The use of SBOM does not pre-suppose any policy; specific policies should ultimately be left to each country.

Beyond these core roles, supply chain visibility should also be tracked by executive teams as a key component of software security, covering the range of issues and cross-team concerns. The legal and compliance team may have a clear interest for license and compliance reasons. For organizations just beginning their SBOM journey, the strategy team should understand how to introduce and deploy this capacity. Organizational modernization experts should also consider implementing SBOM since more modern or advanced tools make SBOM generation easier and cheaper, leading to economies of scale and more advanced development practices.

# Understanding SBOM as Secure by Design

In the joint document Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Security by Design and Default, the second principle is to "Embrace Radical Transparency and Accountability." The secure by design approach encourages software manufacturers and producers to "have command of their supply chains." SBOM offers a way for software manufacturers and producers to adopt the secure by design principle of embracing radical transparency and accountability in their supply chains. By building and maintaining SBOMs for each product, requesting data from suppliers, and making SBOMs available for downstream customers and users, software manufacturers and producers demonstrate their due diligence in the products they create and their ability to respond to risk. SBOMs also enable software consumers to better respond to risks when new vulnerabilities emerge.

To make good on this promise, automated tools and processes should drive SBOM implementation. This includes automated generation, management, and consumption. Generating an SBOM can occur at multiple points in the lifecycle of software creation. Ideally, SBOM generation is part of the software creation process, using build-time tools, but this is not always possible. SBOMs can be created from source repositories. If the software already exists, binary analysis tools can use increasingly accurate heuristics and datasets to determine the underlying components. Once created, software manufacturers and producers must manage SBOMs, including through version control and identifying which metadata is current and applies to systems deployed across organizations. Other automation-ready data like

Vulnerability Exploitability eXchange (VEX) can be used concurrently with SBOM data to maximize effectiveness and assist in prioritization.[5]

Lastly, the SBOM must be consumed. An SBOM is only data, after all. Much of the value described above requires converting this data into insights that can drive action. SBOM data can be mapped to other datasets, such as vulnerability databases, security advisories or information about supply chain risk, open source project details, and end-of-life/end-of-support information. Some of this data will change over time; components that previously introduced little risk may raise new concerns. Ideally, SBOM data will be integrated into other tools organizations already employ, such as supply chain risk management, vulnerability management, and asset management tools.

# Conclusion

Better software transparency will directly improve the quality of decisions made in the creation and use of software. The authoring organizations understand the value of SBOM in securing the software supply chain and recognize the need for greater transparency in software development. Additional steps to further clarify the shared vision could include the harmonization of technical implementations. Divergent implementations could hinder widespread adoption and sustainable implementation of SBOM. An aligned and coordinated approach to SBOM will improve effectiveness while reducing costs and complexities. When used widely across sectors, regions, and countries, supply chain illumination drives better "ingredients" for everyone to use and helps ensure that known risks are addressed early. SBOM adoption is an integral condition for software to be secure by design.

# Disclaimer

The information in this report is being provided "as is" for informational purposes only. CISA and the authoring organizations do not endorse any commercial entity, product, company, or service, including any entities, products, or services linked within this document. Any reference to specific commercial entities, products, processes, or services by service mark, trademark, manufacturer, or otherwise, does not constitute or imply endorsement, recommendation, or favor by CISA and the authoring organizations.

This document is without prejudice to any type of legislation that is applicable in the jurisdictions of the authoring organizations. This document does not bind the authoring organizations and is not intended to provide guidance on the implementation of such legislation.

[5] "Vulnerability Exploitability eXchange (VEX)—An Overview," U.S. National Telecommunications and Information Administration Open Working Group on Software Component Transparency, September 27, 2021, https://www.ntia.gov/sites/default/files/publications/vex_one-page_summary_0.pdf.

Additional resources on VEX can be found in the U.S. Cybersecurity and Infrastructure Security Agency SBOM Resources Library.

# Acknowledgements

The Directorate-General for Communications Networks, Content and Technology (DG CONNECT) of the European Commission[6] contributed to this guide.

# Version History

**September 3, 2025:** Initial version.

[6] This document does not interpret European Union law nor is it meant to be a guidance for implementation of Union law. The document does not bind the European Commission. The Directorate General for Communications Networks, Content and Technology (DG CONNECT) contributed to the drafting of the document in order to cooperate on and emphasize shared cybersecurity principles. However, as this document is a multilateral effort, not all of its elements reflect Union law. Entities falling within the scope of Union law might use this document for information purposes only.

# References

GitHub. "Common Security Advisory Framework (CSAF)." Accessed June 17, 2025. https://github.com/oasis-tcs/csaf.

U.S. National Institute for Standards and Technology. "SP 800-218, Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities." February 2022. https://csrc.nist.gov/pubs/sp/800/218/final.

U.S. National Telecommunications and Information Administration Open Working Group on Software Component Transparency. "Vulnerability Exploitability eXchange (VEX)—An Overview." September 27, 2021. https://www.ntia.gov/sites/default/files/publications/vex_one-page_summary_0.pdf.