Securing the Software Supply Chain: Recommended Practices for Software Bill of Materials Consumption

Enduring Security Framework

November 2023

Executive Summary

Cyberattacks are conducted via cyberspace and target an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment or infrastructure; or destroying the integrity of the data or stealing controlled information.[1]

Cyberattacks such as those executed against SolarWinds and its customers and exploits that take advantage of vulnerabilities such as Log4j, highlight weaknesses within software supply chains, an issue which spans both commercial and open source software and impacts both private and Government enterprises. Accordingly, there is an increased need for software supply chain security awareness and cognizance regarding the potential for software supply chains to be weaponized by nation state adversaries using similar tactics, techniques, and procedures (TTPs).

In response, the White House released an Executive Order on Improving the Nation's Cybersecurity (EO 14028)[2] that established new requirements to secure the federal government's software supply chain. The Enduring Security Framework (ESF) [3], led by a collaborative partnership across private industry, academia and government, established the Software Supply Chain Working Panel which released a three part *Recommended Practices Guide* series to serve as a compendium of suggested practices to help ensure a more secure software supply chain for developers, suppliers, and customer stakeholders.

Similarly, the ESF Software Supply Chain Working Panel established this second phase of guidance to provide further details for several of the Phase I Recommended Practices Guide activities. This guidance may be used as a basis of describing, assessing and measuring security practices relative to the software lifecycle. Additionally, suggested practices listed herein may be applied across the acquisition, deployment, and operational phases of a software supply chain.

The software supplier is responsible for liaising between the customer and software developer. Accordingly, vendor responsibilities include ensuring the integrity and security of software via contractual agreements, software releases and updates, notifications, and mitigations of vulnerabilities. This guidance contains recommended best practices and standards to aid customers in these tasks.

This document will provide guidance in line with industry best practices and principles which software developers and software suppliers are encouraged to reference. These principles include managing open source software and software bills of materials to maintain and provide awareness about the security of software.

[1] Committee on National Security Systems (CNSS)

[2] https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

[3] ESF is a cross-sector working group that operates under the auspices of Critical Infrastructure Partnership Advisory Council (CIPAC) to address threats and risks to the security and stability of U.S. national security systems. It is comprised of experts from the U.S. government as well as representatives from the Information Technology, Communications, and the Defense Industrial Base sectors. The ESF is charged with bringing together representatives from private and public sectors to work on intelligence-driven, shared cybersecurity challenges.

DISCLAIMER

DISCLAIMER OF ENDORSEMENT

This document was written for general informational purposes only. References to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, do not constitute or imply an endorsement, recommendation, or favoring by the United States Government. This document is intended to apply to a variety of factual circumstances and industry stakeholders, and the information provided herein is advisory in nature. The guidance in this document is provided "as is." Once published, the information within may not constitute the most up-to-date guidance or technical information. Accordingly, the document does not, and is not intended to, constitute compliance or legal advice. Readers should confer with their respective advisors and subject matter experts to obtain advice based on their individual circumstances. In no event shall the United States Government be liable for any damages arising in any way out of the use of or reliance on this guidance.

PURPOSE

NSA, ODNI, and CISA developed this document in furtherance of their respective cybersecurity missions, including their responsibilities to develop and issue cybersecurity recommendations and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

CONTACT

Client Requirements / Inquiries: Enduring Security Framework nsaesf@cyber.nsa.gov

Media Inquiries / Press Desk:

- NSA Media Relations, 443-634-0721, MediaRelations@nsa.gov
- CISA Media Relations, 703-235-2010, CISAMedia@cisa.dhs.gov
- ODNI Media Relations, dni-media@dni.gov

Table of Contents

1 Introduction

Unmitigated vulnerabilities in the software supply chain pose a significant risk to organizations. This paper builds on the previously released Recommend Practices[4] for a software supply chain's development, production, distribution, and management processes, to increase the resiliency of these processes against compromise. This guidance also builds upon and supports the Office of Management and Budget (OMB) memorandum on *Enhancing the Security of the Software Supply Chain through Secure Software Development Practices* (M-22-18)[5].

Because the considerations for securing the software supply chain vary, this follow-on guidance focuses on Software Bill of Material (SBOM) Consumption and open source software (OSS). This information will help continue to foster communication between the different roles and among cybersecurity professionals that may facilitate increased resiliency and security in the software supply chain process.

All organizations are encouraged to proactively manage and mitigate risks as a part of evolving secure software development practices. An organization's role as a developer, supplier or customer of software in the software supply chain lifecycle will continue to determine the shape and scope of this responsibility.

It is recommended that acquisition organizations assign supply chain risk assessments to their buying decisions given the recent high profile software supply chain incidents. Software developers and suppliers should improve their software development processes and reduce the risk of harm to not just employees and shareholders, but also to their users.

1.1 Background

Known security risks related to the lack of transparency in software have become a major concern for public and private sector organizations in large part due to costly software supply chain compromises in 2020 and 2021. The SolarWinds supply chain compromise, which involved the insertion of malicious code into commercial monitoring software widely used by government agencies and other organizations, highlighted the way in which threat actors can compromise targets by gaining access to the software that the targets use. The issuance of Executive Order (EO) 14028 was a response to reduce this supply chain risk.

Common methods of compromise used against software supply chains continue to include exploitation of software design flaws, incorporation of vulnerable third-party components into a software product, infiltration of the supplier's network with malicious code prior to the final delivery of the software product, and injection of malicious software within the software deployed into the customer environment.

Public and private sector stakeholders should continually seek to mitigate security concerns specific to their area of responsibility. However, other concerns may require a mitigation approach that dictates a dependency on another stakeholder or a shared responsibility by multiple stakeholders.

[4] https://media.defense.gov/2022/Sep/01/2003068942/-1/-1/0/ESF_SECURING_THE_SOFTWARE_SUPPLY_CHAIN_DEVELOPERS.PDF; https://media.defense.gov/2022/Oct/31/2003105368/-1/-1/0/SECURING_THE_SOFTWARE_SUPPLY_CHAIN_SUPPLIERS.PDF; https://media.defense.gov/2022/Nov/17/2003116445/-1/-1/0/ESF_SECURING_THE_SOFTWARE_SUPPLY_CHAIN_CUSTOMER.PDF

[5] https://www.whitehouse.gov/wp-content/uploads/2022/09/M-22-18.pdf

Inadequately communicated or addressed software dependencies may lead to vulnerabilities and the potential for compromise. Transparency into the software supply chain is necessary to manage that risk.

1.2 Definitions

1.2.1 Definition of Software Product

The OASIS Common Security Advisory Framework (CSAF)[6] defines 'product' as "any deliverable (e.g., software, hardware, specification) which can be referred to with a name. This applies regardless of the origin, the license model, or the mode of distribution of the deliverable." A product comes from a supplier and applies to all software within the enterprise.

1.2.2 Definition of SBOM

A Software Bill of Materials (SBOM) has emerged as a key building block in software security and software supply chain risk management. An SBOM is a nested inventory, a list of ingredients that make up software components. The SBOM work has advanced since 2018 as a collaborative community effort, driven by National Telecommunications and Information Administration's (NTIA) multi-stakeholder process[7]. Note that EO 14028 directed the Secretary of Commerce to provide guidance about the minimum elements of a SBOM and other related parameters in 2021 and OMB has since indicated that CISA may publish "successor guidance" to update these[8]. For more information on SBOM and related Supply Chain Risk Management (SCRM) artifacts, see Section 2 of the *Securing the Software Supply Chain Recommended Practices Guide for Customers* released November 2022.

1.2.3 SBOM Formats

At the time of this documents publication, an SBOM has two widely used machine-readable formats: Software Package Data Exchange (SPDX)[9] and CycloneDX[10]. Software Identification tags (SWID)[11] have also been identified as a potential means of conveying SBOM data, but they are not as heavily used outside narrow uses cases such as firmware dependencies, where the data is conveyed in the hardware itself[12].

1.2.4 Using SBOMs & Risk Scoring

SBOMs may be correlated with other data and threat feeds to augment the value and scope of the content provided. Organizations will be consuming vast numbers of SBOMs which may not scale for some use cases with current technology tools and services. The application of Risk Scoring may be used to create a high-level abstraction based on SBOM content that can quickly be compared to

[6] https://docs.oasis-open.org/csaf/csaf/v2.0/csaf-v2.0.html
[7] https://www.ntia.doc.gov/blog/2019/stakeholders-prepare-further-work-software-transparency-2020
[8] For the 2021 guidance, see U.S. Department of Commerce, National Telecommunications and Information Administration, *The Minimum Elements for a Software Bill of Materials (SBOM)* (July 12, 2021), https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf
For the question of successor guidance, see OMB Memo 22-18, https://www.whitehouse.gov/wp-content/uploads/2022/09/M-22-18.pdf
[9] https://spdx.github.io/spdx-spec/
[10] https://cyclonedx.org
[11] https://csrc.nist.gov/projects/software-identification-swid/guidelines
[12] https://datatracker.ietf.org/doc/draft-ietf-sacm-coswid/

external data sources and permit timely action based on the received SBOMs and prioritization. A Risk Scoring methodology may contribute to successful consumption of SBOMs to simplify raw SBOM data for quick turn automated/manual analysis/utilization of SBOMs. SBOM information, coupled with information from other sources, will enable correlation of data and resulting Risk Scores in the four categories outlined in Section 4.

1.2.5 Definition of Vulnerability Exploitability eXchange

An SBOM-related concept is the Vulnerability Exploitability eXchange (VEX)[13]. A VEX document is an assertion, a form of a security advisory that indicates whether a product or products are affected by a known vulnerability or vulnerabilities. Thus, it offers the novel benefit of showing that a product is *not* affected by a specific vulnerability.

1.3 Document overview

This document contains the following additional sections and appendices:

Section 2. Software Bill of Materials Consumption

Section 3. SBOM Lifecycle in the Enterprise

Section 4. SBOM Risk Scoring

Section5. Operationalizing SBOM

Appendix A: References/Addendum

Appendix B: Acronym List

Appendix C: Glossary

2 Software Bill of Materials Consumption

This follow-on work focuses on the consumption of Software Bill of Materials (SBOMs) received by a wide variety of customer organizations and provides guidance for the use of SBOMs that will be consumed by customers. An SBOM conveys information about what is in the software. The mere act of knowing that a supplier can provide a quality SBOM offers benefits to the software user, since it offers a certain level of confidence that the software supplier is more likely to be able to respond to supply chain concerns. However, full leverage of the power of SBOM requires the capabilities to turn the SBOM data into security intelligence, which can then drive security actions.

From a security perspective, SBOMs are valuable because they ensure that the software is up-to-date and patched against known security vulnerabilities. According to a Synopsys 2022 Open Source Security and Risk Analysis Report[14], 97% of the codebases they audited in 2021 contained open source software. Key findings from this report include:

- While the use of open source software in of itself may feed into risk calculations, 81% of the codebases had at least one known open source vulnerability.

[13] https://ntia.gov/files/ntia/publications/vex_one-page_summary.pdf
[14] Synopsys 2022, *Open Source Security and Risk Analysis Report*, https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html?intcmp=sig-blog-supplychain

- 85% of the code bases audited contained open-source software that had not been updated by software developer(s) in more than 4 years. This usually indicates that the software is not being actively maintained and may have unpatched vulnerabilities.
- The security and compliance advantages of SBOMs have always been important. However, SBOMs have become especially critical today, for three main reasons:
    - The prevalence of open source software, which according to the Linux Foundation, 72 percent of companies now use internally or as part of commercial products. [15]
    - SBOMs help businesses ensure that their use of (open source) software complies with the business' risk appetite.
    - SBOMs, coupled with information from other sources, help reduce the window of exposure once a vulnerability is identified within a software package or listed component of the software by informing the organization and allowing for faster adoption of mitigating controls and measures to lower risk.
- Unpatched vulnerabilities provide a critical opportunity to improve supply chain security.

There are other risks that transparency via SBOM can address. For example, licensing information derived from SBOMs can help businesses ensure that they comply with licensing requirements when using open source and 3rd party licensed software. For example, an open source library that a software vendor incorporates into a product may include licensing terms mandating that the original authors of the library receive attribution within documentation related to the product. An organization that uses the application could also gain attribution information if information is provided by the SBOM. This greater visibility provides another potential avenue for license compliance. SBOM data can also provide insight into how up to date the components in an application are, and the corresponding risk of technical debt when components have not been kept up to date. This might, in turn, offer some insights into the cost of maintenance and potential cost of ownership or future contracts.

2.1 Security risks related to the origins of software SBOM Consumption

The SBOM provides transparency for improved software asset management, patch management, and vulnerability management by customer organizations, as well as the potential to derive enhanced supply chain risk data. An effective developer- or supplier-provided SBOM enumerates third-party software dependencies (both open source and proprietary) incorporated into the supplier product. An alternate approach may have any additional dependencies needed at runtime – and downloaded by an OSS vendor's package manager toolset –enumerated in an SBOM created separately e.g., by the package manager toolset. [16]. The set of SBOMs provide the requisite transparency for software asset management and vulnerability management.

[15] https://www.linuxfoundation.org/press/press-release/corporate-open-source-programs-are-on-the-rise-as-shared-software-development-becomes-mainstream-for-businesses

[16] The baseline computing environment may include some of the software drivers, libraries, or runtime dependencies. The provider of the SBOM may not always be able to predict which of these dependencies are or aren't included in the target computing environment. The provider may not be responsible for maintaining the dependencies.

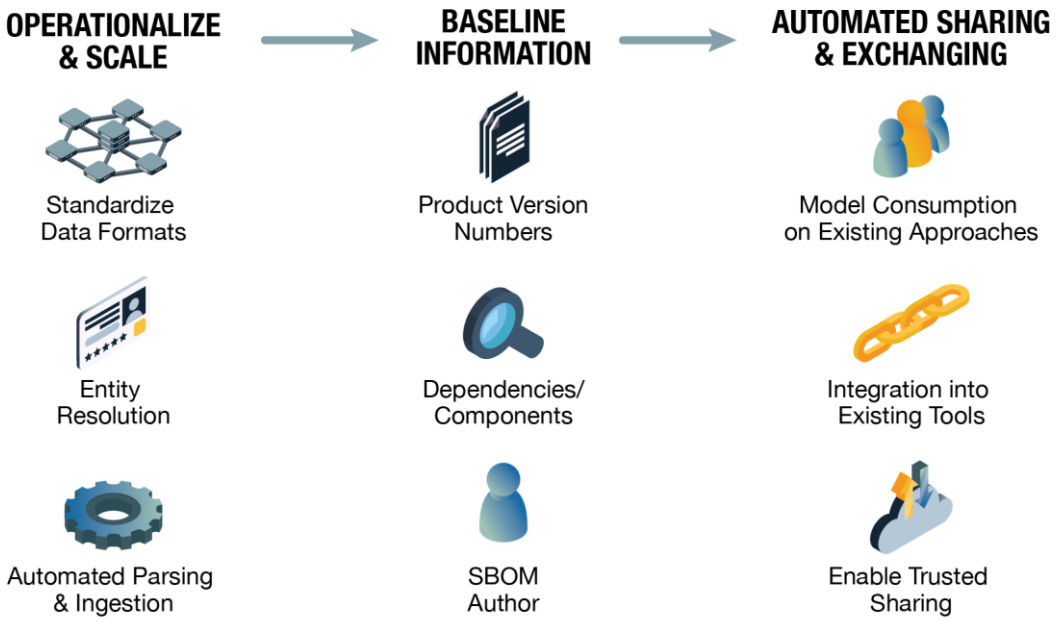

Figure 1: Necessary Elements for Automated, Scalable SBOM Consumption

2.1.1 How to Operationalize and Scale the use of an SBOM

The inclusion of SBOMs with all software releases will result in a customer having to consume thousands of SBOMs to understand the organization's risk exposure. Some basic benefits can be realized with relatively simple tooling: a simple script can help address straightforward questions like "which products have an SBOM that contains the recently announced major vulnerability?" However, to realize the broader benefits that SBOMs can provide, organizations should maximize automated SBOM processing, analysis, and correlation. This requires automated data exchanges and interoperability across the software supply chain, which then requires standardized data formats, entity resolution, and automated parsing and ingestion of the SBOM. Some tools are available to produce, consume, and transform these SBOMs, with more to come as users and the industry gain experience with them. The SBOM formats identified below address the core problem of identifying software components and associated metadata and include the requisite fields to cover the needs for the baseline SBOM as defined by the NTIA guidance or further guidance from CISA or other organizations.

2.1.2 Baseline Component Information

The primary purpose of an SBOM is to identify components and their relationships to one another. To do so, some baseline component information is required. SBOM attributes, identified in Section 4, such as product version number, dependency identifier, and SBOM author, enable the identification of the baseline attributes of a particular component. As the SBOM ecosystem matures, best practices and requirements may advance. For example, Commerce's 2021 guidance treats the hash of a component as a "recommended data field," rather than a required data field. Future guidance, from governments or other bodies, may make hash or similar strong identifiers compulsory for certain SBOM types (Ref the SBOM Type Paper above).

2.1.3 Automated Sharing and Exchanging

SBOM data across the supply chain requires a combination of technical components, systems, and capabilities; standardized data formats; SBOM consumption tools; and integration into operational processes. Due to the diverse needs of the software ecosystem, there is no one-size-fits-all solution for SBOM consumption. However, modeling SBOM consumption processes on existing approaches and methods will enable interoperability between vendors, reduce variance, minimize the need for new SBOM consumption tools, and thereby simplify processes required for timely and scalable SBOM consumption through proper application of SBOM consumption automation through tool integration. Section 3 identifies detailed processes for consuming an SBOM.

For software that is installed on systems at the customer premises, the SBOM metadata can be distributed along with the binary, and along with the software. Some software producers may be comfortable sharing their SBOMs publicly[17], while other producers may not want to share this data beyond their customers and use access control mechanisms to protect this data.

More broadly, the SBOM sharing model should include a discovery mechanism, any potential access control model, and some transport mechanism. The consumer should have some way of knowing that an SBOM exists, where it is, and if it has been updated[18]. The customer should manage that access technology, such as credentials or decryption keys. Making use of SBOMs requires integration into the customers' existing operating infrastructure. Full consumption capability would also allow integration into a diverse set of tools that the supplier might not anticipate, such as the consumer's asset management solutions. As noted below, SBOM data can be integrated into existing security tools such as asset management or vulnerability management tools.

3 SBOM Lifecycle in the Enterprise

This section describes workflows for the acquisition, management, and use of SBOMs by software consumers. "Software consumer" is broadly defined to include commercial and non-commercial entities acquiring third-party software capabilities from a supplier, developer, or from open source.

[17] Real-time SBOM of their product at https://www.jupiterone.com/sbom

[18] There will be a joint CISA/Energy doc we can cite for this. Should be published by April 19. If it gets held up, probably don't need the footnote.

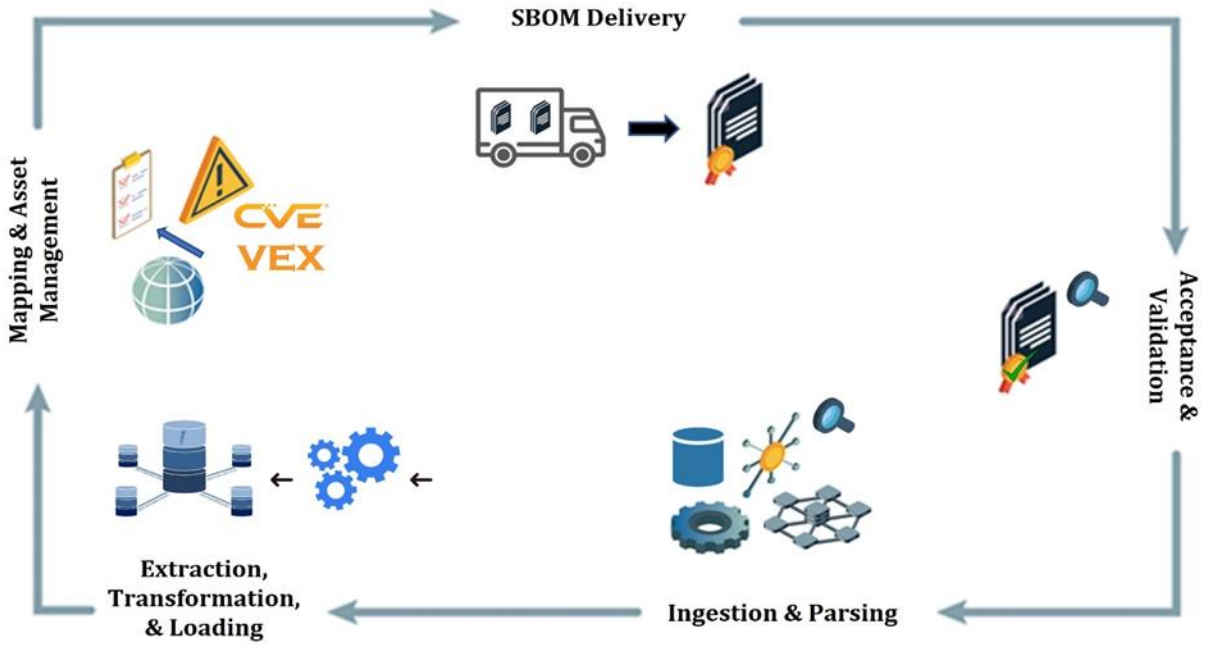

Figure 2: SBOM Lifecycle

3.1 SBOM Delivery for Software

There are many types of software and many methods by which software is delivered. A customer may receive SBOMs in various ways through a variety of mechanisms:

- As part of contractual procurement of a commercial product,
- As part of or alongside the download of commercial closed-source software,
- As part of a contractual procurement of professional services that includes the development and delivery of software capabilities,
- As part the acquisition of open-source software applications or components,
- During the discovery processes as a device connects to a network,
- Delivered directly to the customer via the supplier/developer,
- Through a customer portal or some other pre-arranged mechanism, or
- Through a service or repository designed to help deliver software metadata.

Note that some SBOM users will have SBOM use cases that occur before or without purchasing the software in question, such as acquisition risk analysis or providing data validation or enrichment.

3.1.1 Acceptance/Validation

Upon receipt of the SBOM, the customer may validate the integrity and authenticity of the SBOM using the methods described within the SBOM or via a pre-agreed-upon process. To maximize the effectiveness of this process, standardization of how to consistently produce a hash across diverse software ecosystems should be a priority for the SBOM community and should be shared across the popular SBOM formats.

The SBOM may have embedded information describing how the hashing/signing method to be used for validating the integrity and authenticity of the SBOM. The customer and supplier may also use a pre-agreed upon method not embedded within the SBOM. If integrity and authenticity checks are available for the SBOM Delivery method/infrastructure, it is recommended that the consumer verify the integrity and origin of the SBOM as well. The Securing the Software Supply Chain (for the Developer, Supplier, and Customer) guidance released by ESF recommends verifying SBOMs (e.g., for veracity and accuracy) and resolving any mismatches prior to ingestion. This may also include analyzing the SBOM data for completeness or "known unknowns" with intentional gaps in the dependency tree (see section 4). Software Composition Analysis (SCA) and/or software scanning tools may be used to determine the components of a software product or package to verify the accuracy and veracity of an SBOM and may also be used to validate or verify VEX information in support of SCRM risk decisions.

If the SBOM supplied to the customer is not complete, has minimal depth, or does not include dependencies of proprietary components, adjustments to vendor contracts or other risk mitigation steps may be warranted.

3.1.2 SBOM Ingestion and Management for Enterprise

Data from SBOMs feeds into many enterprise workflows, including procurement, asset management, vulnerability management, and overarching supply chain risk management and compliance functions. Therefore, the SBOM is often less useful as a file than as a collection of data that can be parsed, extracted, and loaded into automated processes or systems of record. Enterprises may have multiple options available for the consumption of SBOMs including internally developed tools/scripts, open-sources tools, and commercial product offerings and services, as well as various combinations of these options.

Organizations may require a data management layer to track SBOMs, map them to assets, and allow other tools to link to and correlate with SBOM data. By enabling better and more flexible and automatable data management, this layer can support multiple workflows and enterprise processes including supply chain risk management, vulnerability management, future procurement analysis, enterprise risk management and risk scoring (See Section 4 of this document for more on SBOM based Risk Scoring). There are alternate approaches for leveraging SBOM content such as SBOM-specific repositories, managed service models, and SBOM file-based storage and retrieval methods. As of 2023, tools supporting SBOM data management are just beginning to emerge.

In some cases, the data can be stored adjacent to the software in question, for easier access by scanning tools. Consumers may wish to consolidate this into SBOM repositories. For some consumers, the software may reside on sensitive networks or on systems that do not allow for direct scanning, such as industrial control systems.

3.1.2.1 Extraction, Transformation, and Loading of SBOMs

Extraction, Transformation, and Loading of SBOMs into enterprise processes and platforms requires a mapping process to correlate specific components to one or more applications, systems, or endpoints. Much of the value of these processes and platforms derives from the mapping and update functionality that maintains the accuracy of software inventories and configurations across an enterprise. In certain ways, SBOM data is similar to attribute data for any software asset. However, SBOM data does differ significantly in its volume and granularity. There is more data and more detailed data per software asset than for a conventional commercially procured software capability with no SBOM.

Therefore, the workflow volume, particularly in automated workflows, may scale significantly, and enterprises should plan for this. This step may be part of the management step described above, or further downstream.

Customers may desire to store the original SBOM document/file after parsing, for regulatory reasons or the customer currently lacks the capacity to further process the SBOM. Either way, the process for SBOM storage uses a content management approach: SBOM time of receipt, file location, file content, data retention, and life cycle policies for the storage of that file. This may entail:

- Storage in an enterprise inventory or information technology (IT) asset management database (e.g., the same database or file system that stores the serial numbers of computers and software licenses).
- Leveraging a Unified Endpoint Management (UEM) system that is capable of bill of material processing. It can keep track of the hardware assets, the associated software running on its network, and monitor for significant security posture changes.
- Leveraging[19] a security information and event management (SIEM) software solution that can, among other things, collect, store, aggregate, and analyze data from networked devices, servers, etc.

If specific security measures for the storage of supplier SBOM information are contractually specified, it is recommended that the consumer ensure that the security controls for systems into which SBOM data flows meet or exceed controls specified in the terms and conditions of the supplier of the SBOM.

Life cycle policies for storage of SBOMs as files should correlate to the life cycle of the software deliverable represented by that SBOM. The file should persist until the consumer has properly decommissioned that software asset, e.g., automated scans or manual inventories indicate that the software asset corresponding to the SBOM is no longer present or installed on the consumer's infrastructure, and the information is no longer relevant for legal or forensic purposes (e.g., discovery of a breach that occurred before a software asset was updated or removed). Decommissioning - verification that an asset has been removed from a system or facility -- is an order of magnitude more difficult than deploying assets. It requires a higher level of transparency and positive control than most IT enterprises possess. For this reason, especially given the compressibility of much SBOM data, many organizations might want a default archival retention policy for SBOMs. To support this, SBOMs should be correlated with version information to ensure distinctions between current and archived data. Customers need the capability to discover and access SBOMs relevant to their environment.

3.1.3 Mapping & Asset Management

As mentioned, content from SBOMs feed into existing enterprise workflows including procurement, asset management, vulnerability management, and overarching supply chain risk management and compliance functions. A priority workflow will be the consumption of SBOM data into an Asset Management repository, tools or systems that can map the SBOM elements and data to software products and components leveraged and deployed across the enterprise[20]. Some asset management and vulnerability management systems are just beginning to integrate SBOM data in 2023.

[19] International Medical Device Regulators Forum (IMDRF) Medical Device Cybersecurity Guide Working Group Proposed Document, "Principles and Practices for Software Bill of Materials for Medical Device Cybersecurity", July 1, 2022. Available: https://www.imdrf.org/consultations/principles-and-practices-software-bill-materials-medical-device-cybersecurity

[20] Note that asset management tools can also drive SBOM collection by identifying systems on a network, potentially triggering automated SBOM discovery and retrieval.

SBOM content can support additional enterprise processes (Security, Supply Chain Risk Management (SCRM)[21], Quality, Licensing, Product comparison, etc.), once an SBOMs is processed, SBOM information can be provided to the variety of SBOM information consumers within the customer enterprise:

- Configuration management databases (CMDBs),
- Software asset management (SAM) systems,
- Security operations centers (SOCs),
- Procurement workflows, which may include pre-procurement diligence, contractor/vendor management systems, and third-party risk and compliance management and reporting, and
- Software supply chain risk assessment and management functions

Section 4 describes the processes for SBOM consumption that can be evaluated for potential automation. It is recommended that distribution be done through automated processes versus manual propagation methods.

3.2 Use of SBOM Content

SBOMs can inform the risk decision associated with acquiring a product if the actual acquisition occurs soon after the evaluation of the SBOM, which may evolve over time. Initially, risk can be assessed based on the content of the SBOM (including known vulnerabilities associated with SBOM's components), but over time risk can be re-assessed due to changes in environment or newly discovered "zero-day" vulnerabilities.

"Zero-day" vulnerabilities can be identified in vulnerability databases e.g., by newly registered Common Vulnerabilities and Exposures (CVEs) associated with components or products. Notice of new threats may come from media (news or social), the supplier, or other third parties. Ideally, "Zero-day" vulnerabilities are announced using a standardized machine-readable format.

3.2.1 Intrinsic Value of Having an SBOM

SBOMs provide improved visibility into the pedigree of the software that the customer organization is evaluating, deploying, and/or operating within their environment. This increased visibility into all software is critical for proper supply chain risk management and overall enterprise risk management. SBOM content provided to and consumed by customers informs risk management for customer organizations without impacting sensitive intellectual property interests of software suppliers. Even before the SBOM data is consumed by the customer, the customer benefits from the supplier having the SBOM data and the potential to use it to inform risk decisions. This does not guarantee that the supplier will use this data, but having the data is a necessary first step.

If a supplier does not have visibility into their software supply chains, customers should be cautious of the trust placed on that software and its supply chain. While there may not be any currently known active exploits or vulnerabilities, such a supplier may not be in a position to make any claims or assurances. A supplier that provides an SBOM signals its visibility, and the quality of this visibility, into its supply chains.

[21] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161r1.pdf

3.2.2 Known Vulnerabilities

The Department of Homeland Security (DHS)/Cybersecurity and Infrastructure Security Agency (CISA) sponsors the CVE List[22] that is maintained by the MITRE Corporation. The CVE List is a list of publicly disclosed computer security flaws/vulnerabilities that have been assigned a CVE identification number. The CVEs in the CVE List are fed into the National Vulnerability Database (NVD)[23], maintained by the National Institute of Standards and Technology (NIST), a non-regulatory agency of the United States Department of Commerce. The NVD performs analysis on the CVEs which provides metadata results such as association impact metrics (Common Vulnerability Scoring System - CVSS[24]), vulnerability types (Common Weakness Enumeration - CWE[25]), and applicability statements (Common Platform Enumeration - CPE[26]), as well as other pertinent metadata.

CVEs help IT professionals coordinate their efforts to prioritize and address vulnerabilities to make computer systems more secure. The mission of the CVE Program is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities in addition to a dynamic catalog of Known Exploited Vulnerabilities (KEV)[27].

3.2.2.1 Clarifying Vulnerability Risk with VEX

Not all vulnerabilities in the dependencies of a software product actually affect the security of that product. From this perspective, vulnerabilities identified in the SBOM may overstate the actual risks of a product. Addressing these would not be efficient for suppliers and customers, both of whom have finite resources.

To address these risks, among others, NTIA and CISA have facilitated public and private sector collaboration on Vulnerability-Exploitability eXchange (VEX)[28]. A VEX document is a machine-readable security advisory that can be used to clarify and prioritize vulnerability risk. VEX is not technically part of SBOM, and both can exist independently, but using both together will maximize both the efficiency and security benefits of SBOM consumption.

A VEX implementation, framework and/or specification [29]provides machine readable information indicating whether a product (or one of its components) is impacted by a specific vulnerability, and if "AFFECTED" whether there are actions recommended to remediate or mitigations exist which address the vulnerability. The goal of VEX is to allow a software supplier or other parties to assert the status of specific vulnerabilities in a particular product. VEX documents allow both suppliers and consumers to focus on vulnerabilities that pose the most immediate risk, while not investing time in searching for or patching vulnerabilities that are not exploitable and therefore have no impact. To this end, a VEX indicates a status per vulnerability [30]:

- **NOT AFFECTED** – No remediation is required regarding this vulnerability.
- **AFFECTED** – Actions are recommended to remediate or address this vulnerability.

[22] https://www.cve.org/
[23] https://nvd.nist.gov/general
[24] https://www.first.org/cvss/
[25] https://cwe.mitre.org/
[26] https://nvd.nist.gov/products/cpe
[27]https://www.cisa.gov/known-exploited-vulnerabilities-catalog
[28] https://www.cisa.gov/sites/default/files/publications/VEX_Use_Cases_Aprill2022.pdf
[29] At the time of writing, VEX implementations have been proposed in the as a profile in OASIS CSAF automated advisory, a part of the OWASP CycloneDX SBOM format, and the OpenSSF project OpenVEX.
[30] https://www.ntia.gov/files/ntia/publications/vex_one-page_summary.pdf

- **FIXED** – These product versions contain a fix for the vulnerability.
- **UNDER INVESTIGATION** – It is not yet known whether these product versions are affected by the vulnerability. An update will be provided in a later release.

Additionally, when the product is indicated as "NOT AFFECTED", VEX permits the document to include a justification statement of why the VEX document creator chose to assert that the product's status is NOT AFFECTED. Status justifications range from indicating the product is not affected by the vulnerability because the component is not included in the product to the vulnerable code can never be executed in the context of the application.

While VEX is a recent development, effective enterprise security includes implementing or leveraging an existing capability that facilitates identifying whether products are affected by vulnerabilities or recommend mitigations.

A VEX can inform remediating actions. However, it is important to verify the veracity of the information within the VEX, including any recommended actions. Ideally, a VEX originator should be authenticated and screened, and the VEX itself should be checked for integrity. VEX will help scale SBOM consumption and allow organizations to focus on vulnerabilities that actually pose real risks to organizations. VEX tooling and VEX integration into existing security tools is just emerging in 2023. Generally, we suggest requirements provide VEX or VEX-like information in contracts between consumer and developer/supplier. A supplier may want different policies on when to issue a VEX, such as in response to a high-profile bug, a vulnerability that has been publicly exploited, or when a new CVE is in a component in the supplier's SBOM.

3.2.3 Query/Reporting

Effective enterprise vulnerability management requires determining the risk associated with a vulnerability. Enterprises can learn about vulnerabilities via:

- Querying vulnerability repositories using the SBOM; or
- Receiving a VEX (or similar information) from a supplier, associated with a component or product.[31].

It is recommended that customers correlate the SBOM or VEX against the vulnerability repositories, the results of which contributes to a risk score. The determined risk score is used in determining the appropriate risk response or action (see section 3.2.4). The concept of weighting results is one that is proposed to be a consumer responsibility. CISA prioritizes remediation of the vulnerabilities listed in the aforementioned KEV catalog.

3.2.4 Action

NIST SP 800-40r4, *"Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology"*[32] provides an overview of the software vulnerability lifecycle and outlines possible risk response approaches for software vulnerabilities (Accept, Mitigate, Transfer, and Avoid) to be taken by the Customer.

Depending on the specific deployment of the software product and the implementation of other security controls, the customer may assess the risk as acceptable with no additional action needed. A

[31] Typically, a VEX is associated with a "zero-day" vulnerability.
[32] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r4.pdf

customer may avoid the exploitation of the new vulnerability through a variety of mitigations including by:

- Uninstalling the vulnerable software,
- Decommissioning assets with the vulnerabilities, or
- Disabling computing capabilities in assets that can function without them.

Patching is not always an immediate option. The system may be vital for the organization's mission, and downtime would interfere with operations, especially in the operational technology domain. The system may not be supported, or the vendor may simply not exist anymore or lack the capability to provide a patch. Other mitigations for potential risk span beyond the software in question. At the network level, tuned network tools could segment the vulnerable software from the Internet, or from the rest of the enterprise network. The organization could assume that while the known vulnerability is not currently being exploited, threat intelligence efforts should detect any potential exploitation in the broader community in real time and set up automated defensive measures in response. Smaller organizations without access to specific threat intelligence can conduct software composition analysis correlating with current CVEs and/or perform active and passive code scans to address the risk of deploying the software. Organizations can set up automated defensive measures through the implementation of the best practices cited in Sections 4 and 5.

Alternatively, the organizational risk may be transferred, for example, by purchasing cybersecurity insurance or by replacing conventional software installations with software-as-a-service (SaaS) or Cloud usage. Finally, exploitation may be mitigated e.g., by elimination (patching the vulnerable software, disabling a vulnerable feature, or upgrading to a newer software version) or deploying additional security controls to reduce vulnerability exploitation. Customers should create a set of questions to ask their SaaS provider to ensure their practices are secure and to inform the customer's risk management decisions.

It is important to note that the action recommended in the VEX or other advisories or guidance may not be immediately applicable. For example, applying a recommended patch may disrupt operations.

3.3 SBOM Update for Existing Software

A new SBOM for an existing software product should be provided/acquired for software updates, software upgrades, or for augmenting the completeness of a prior SBOM for an existing software product. The process for acquiring, validating, processing, and storing the new SBOM will follow the same processes and methods described in Section 3.1. with one additional step. The new/updated SBOM should be compared to the last SBOM to identify new components / dependencies that have been introduced or removed since the last version. These changes should be validated against current threat information (ex. CVEs and/or VEX) to ascertain, inform, and update the current risk posture and Risk Scoring (see Section 4). The ongoing use of the updated SBOM content will follow the processes and methods described in Section 3.2.

3.4 Example of SBOM in use at Customer

By leveraging SBOM content in the enterprise threat, risk, and vulnerability management process, a customer is likely to reduce the window of exposure to a given vulnerability and accelerate their remediation processes.

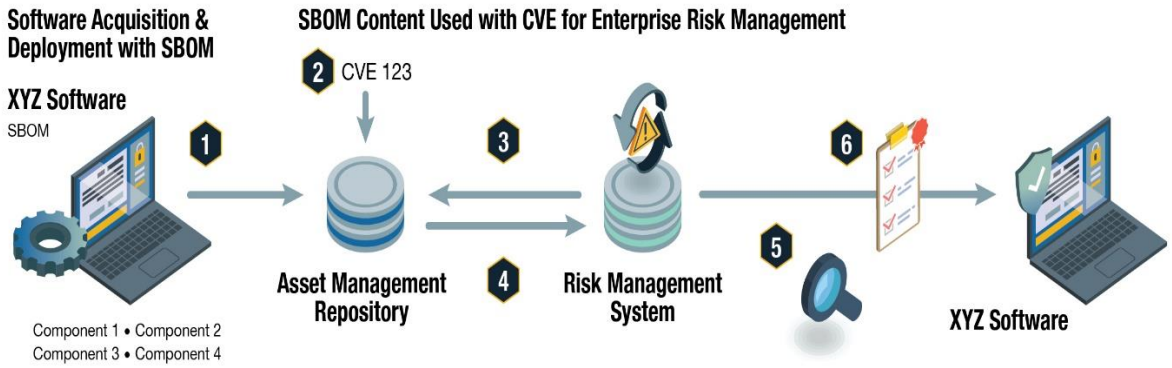

Figure 3: Example of SBOM in Use

The following use case illustrates the processes and best practices, associated with SBOM consumption as depicted in Figure 3 above:

1. Software product, XYZ Software has been acquired and deployed throughout the enterprise. At the time of acquisition, an SBOM for the product was received, validated, and processed. The SBOM content was loaded into an enterprise asset management repository with linkage to the enterprise vulnerability and threat management systems and processes.
2. At some future date, a CVE is published with an actively exploited vulnerability targeting a commonly used open-source utility.
3. The organization's threat/risk management systems will consume the CVE data and then query the asset management repository to identify any software products that are being leveraged/deployed across the enterprise that contain or are dependent upon the vulnerable utility.
4. In this example, XYZ Software contains the vulnerable open-source utility.
5. The security teams at the customer now have visibility into the software products within their organization that contain the vulnerable utility and can both assess the risk(s) of the software products with this newly reported vulnerability, develop a risk response, and (if it decides to mitigate the risk caused by the new vulnerability), begin to prepare and take mitigating actions to reduce the risk of exposure.
6. All of this can be done in near real time from the publication of the CVE. Whereas in current models, without the SBOM content, the customer should wait for the supplier/developer of the software product to verify the exposure and then notify their customer base of the potential vulnerability.

   This notification often will not occur until after the supplier/developer has developed a patch or mitigating control. As the supplier/developer provides additional information (ex. VEX) on the vulnerability and/or a patch for the vulnerability, the customer organization can adjust and update their mitigations and responses to the risks.

Additionally, correlating SBOM content across software products deployed within the enterprise can provide powerful insights to the incident response teams, forensics teams, risk management, and procurement.

4 SBOM Risk Scoring

4.1 Turning SBOM into Risk Information

This section provides information on how to quantify the risk of a software product or component based on an SBOM.

Beyond tracking well-defined, known risks, such as checking against lists of known vulnerabilities, SBOM data can be used as a starting point for more context-specific risk analysis. Components mentioned in an SBOM can be further investigated to understand details about the source of the components, for example jurisdiction, maturity (e.g., an open source project with only one maintainer), or the financial stability of the supplier. This data may not be directly available in the SBOM, but SBOM data can be used as the starting point for this type of enrichment that can feed into more enhanced risk analysis.

4.2 Rationale for Risk Scoring

Within the software supply chain there is a lack of a consistent approach to communicating risk with a given product or component to the customer. Current methods are inadequate in many ways including:

- Out of date contractual or license-based support that may impact availability of downstream patches and product updates.
- Lack of transparency between supplier and customer.
- Exponential growth in complexity of dependencies within software products.
- Exponential growth in open source component usage by suppliers without transparent means to understand what is being acquired.

4.3 Risk Scoring Definition

Risk Scoring allows organizations to understand their supply chain risk based on defined risk factors and anticipate the potential of future risk of a given software product in the enterprise. A risk score is a metric used to predict aspects of the software and/or it's components current and future risk. This metric is developed using indicators from the SBOM, VEX, etc. as well as other feeds and content in support of SCRM. A risk analysis and the criteria that will be utilized to assess the scoring of the product and/or software components will include a rubric with categorical definitions to encourage the transparency of assessment results. When applying or assessing a risk score the context of where the software is being used, how the software is accessed or isolated, or what processes and systems it is supporting should be a factor in considering the associated risk. The risk score informs the overall risk determination for a software product or software component.

It is important to note, that in many complex systems and systems of systems, there may be multiple SBOMs as a part of the collective solution and therefore, a collection of Risk Scores. Organizations can choose to combine the risk score at the aggregate level or manage the risk scoring at the individual SBOM level.

4.4 Risk Scoring Recommendation

Four factors – Vulnerabilities, License, Community, and Dependencies – are identified as a good starting point for risk score development. Completeness and/or coverage of these factors will affect the associated SBOM risk score.

The table below shows a set of tangible sources and metrics to instantiate a Risk Score within a given organization. The bolded items are the primary cyber risk focused factors.

Table 1: SBOM Risk Scoring Process

Criteria factors	Description	Source	Scoring metric
Vulnerabilities (Section 4.3.1)	The foundational bar for measuring potential and realized cybersecurity issues within a given software product. The Common Enumeration of Vulnerabilities or CVE, National Vulnerability Database (NVD) and related Common Vulnerability Scoring System (CVSS) have been in use for decades now and are universally adopted, in order to drive prioritization decisions: • has the vulnerability or its risks been mitigated or is a mitigation available? • Is the software affected by other vulnerabilities?	CVSS scores in/from SBOM LFX VEX	CVSS score; Linux Foundations LFX platform, VEX implementation, framework or specification (see section 3.2.2.1)
Licenses (Section 4.3.2)	An integral part of software acquisition/procurement processes. Both proprietary and open source software licenses have implied or explicit requirements for the customer. License information included in SBOMs provide a level of transparency into the software components and dependencies that are critical to SCRM and acquisition risk assessments. The License information helps SCRM processes assess a product's viability by understanding if any unacceptable copyright or license terms are present in the product.	License info from the SBOM	Found on SPDX License list
Community / Supplier(s) (Section 4.3.3)	The concept of focusing on the entire supply chain of a given software product. The software product being consumed is a sum of its parts and increasingly 3rd party suppliers are used in the creation of products. The	Independent evaluation sources (e.g., libraries.io, Linux Foundation, etc.)	Examples include, libraries.io SourceRank (Tidelift, n.d.); Linux Foundations LFX platform (Foundation, n.d.)

	following factors are part of a complete picture for a software supply chain: - 3rd party suppliers - Geolocation - Update frequency -Response to known vulnerabilities		
Context of the Dependencies (Section 4.3.4)	Components should be part of the software product for it to function. It is critically important to understand what dependencies exist in a particular product to determine the risk of using a particular software product. The SBOM is the emerging standard way of communicating dependencies between supplier and customer.	Packages in the SBOM, or independent evaluation sources	Examples include, libraries.io SourceRank; Linux Foundations LFX platform

4.4.1 Risk Score Guidance Recommendation

Development of a vendor neutral guidance or open standard to identify the risk factors that can be aggregated into a Risk Score could be helpful. Such a Risk Score could include a breakout of the distinct values used to calculate an overall aggregate value, in addition to the aggregate value itself.

4.4.2 Vulnerabilities

Vulnerabilities in the form of CVE references are optional fields within an SBOM at the time of SBOM creation. While CVE references may be derived from the SBOM, new CVEs are discovered frequently, and other sources should be consulted (using the information in the SBOM). From the CVEs, a set of CVSS scores can be obtained to use within the Risk Score. Mathematically these can be summarized to create an aggregate Vulnerability factor result or score. Example: 1 high + 1 medium = CVSS score (7.7) + CVSS score (4.1) = vulnerability factor (11.8) for 2 vulnerabilities. Other risk scoring mechanisms could also be used, including the Stakeholder-specific vulnerability categorization[33] . Additionally, VEX content will provide critical information as to the applicability of the vulnerability to the software product and will reduce false positives.

A customer may require or request current vulnerability data as part of a software acquisition / delivery alongside the SBOM (or equivalent bill of materials). It is important to note that this list of vulnerabilities will be dynamic and out of date quickly. Alternatively, the processes described above (Section 3) to either use VEX or direct CVE mapping/query is a better approach to SCRM and ongoing risk / vulnerability management.

Consumers may want to explore tools that can link SBOM data to vulnerability databases, such as the open-source DaggerBoard[34] tool developed by New York Presbyterian Hospital. The SBOM information should be correlated against VEXs, when available, CVEs, vulnerability scanning tools,

[33] SSVC - https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=653459

[34] https://github.com/nyph-infosec/daggerboard

such as the DaggerBoard tool, as well as with software composition analysis tools and automated risk scoring algorithms.

The presence of vulnerabilities in a product's dependencies tells only part of the story. Vulnerable components may not actually put a product or its users at risk. The Vulnerability Exploitability eXchange (VEX) provides further context The CISA-facilitated SBOM community is refining VEX as a form of security advisory that augments the SBOM usage by communicating machine readable vulnerability information and their applicability to a specific product. In particular, VEX introduces the capability to flag that a product is *not* affected by a particular vulnerability. The VEX would enhance the Risk Score Vulnerability factor by introducing context to the raw CVSS scores. This allows for much greater accuracy by making the potential for exploitability of a particular product visible to the consumer. Linking SBOMs to vulnerabilities enables risk flags, while VEX documents allow a consumer to prioritize vulnerabilities. We suggest that customers use the latest vulnerability information to inform the risk decision process.

4.4.3 Licenses

Most software is distributed under a license agreement of one kind or another. Consumers should be cognizant of which license agreements they are entering into to help avoid potential legal challenges for their organizations. Typically, licenses are analyzed within the Supply Chain procurement process and the SBOM provides expanded visibility of licensing information.

As recognized in NTIA's The Minimum Elements For a Software Bill of Materials (SBOM)[35], SBOMs may convey data about the licenses for each component. Information about licenses can be used in the following ways to create a License risk factor result:

- how many different licenses are involved? Higher numbers imply more license complexity that could lead to increased risk; and
- key license types that limit or expose the consumer to unfavorable terms.

4.4.4 Community

Community (ex. open source or supplier) is the most difficult factor to analyze and obtain accurate information. However, several clauses in the SBOM enable analysis to be performed to inform risk decisions.

Additional information can be gathered from open sources[36] (such as, ClearlyDefined,[37] libraries.io,[38] the Linux Foundation LFX,[39] etc.) or directly from the source of open source repositories. These sources can be used as input into intelligence analysis and correlation to answer the key questions to create a Risk Score including:

- How actively used or supported by the community or supplier,
- How large (number of members in community/size of support team),
- Update frequency,

[35] The United States Department of Commerce, "The Minimum Elements For a Software Bill of Materials (SBOM)" https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

[36] https://www.synopsys.com/blogs/software-security/open-source-license-compliance-dependencies.html

[37] https://clearlydefined.io/?sort=releaseDate&sortDesc=true

[38] https://libraries.io/

[39] https://www.linuxfoundation.org/

- Adherence to the various open source best practices, such as the OpenSSF Best Practices[40], including use version control software, code review process documentation & actual practices, governance, etc.,
- Demonstrated ability to respond to security concerns,
- Potential bad actor influence,
- Geographic involvement and facilities (Foreign Ownership, Control or Influence (FOCI)).

Different organizations will care about different aspects of this risk. Jurisdictional risk may be a chief compliance concern for a defense contractor, while an organization that prizes resiliency may focus on the vulnerability disclosure policy of a supplier, and how quickly and effectively it has historically responded to reported vulnerabilities. We suggest that customer use these parameters to inform their risk decisions. The Phase 1 Customer document[41] introduces the concepts for the consumption of the SBOM.

4.4.5 Dependencies

Dependencies are a critical part of the SBOM and provide valuable information for making consumer decisions. The SBOM dependency information provides inputs into intelligence analysis enabling the development of a dependency factor score. This can be used to weigh many key questions including how many external packages/libraries/things are necessary to use a particular software product. Customers should use the dependencies to evaluate the risk from using a particular software.

4.4.5.1 Example:

- Two suppliers of competing products being considered for purchase.
- Supplier A provides an SBOM showing that their product uses a large quantity of older open source dependencies without a VEX.
- Supplier B provides an SBOM showing that their product uses a small quantity of older open source dependencies and includes a VEX product providing context for the vulnerability.
- This new level of dependency transparency enables the consumer to make better comparisons of the two suppliers in a procurement decision.

4.4.6 Limitations of Custom Risk Models

While many organizations face similar risks, each organization is unique, and has different risk tolerances. Collapsing risks into a single score can be very helpful for management modeling and executive dashboards, but it may not be the best from an action-oriented perspective. For example, CISA has begun to emphasize the Stakeholder-Specific Vulnerability Calculator (SSVC)[42] that uses decision trees to complement and supplement the Common Vulnerability Severity Score (CVSS). From a broader macro level, unique scoring calculation can be harder to attest to, write into contracts, and verify by others in the marketplace.

Services and tools exist that offer particular insights ranging from vulnerability prioritization to third party supply chain risk management. Many of these are starting to integrate SBOM. Organization may

[40] https://bestpractices.coreinfrastructure.org/en/criteria
[41] https://media.defense.gov/2022/Nov/17/2003116445/-1/-1/0/ESF_SECURING_THE_SOFTWARE_SUPPLY_CHAIN_CUSTOMER.PDF
[42] https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc

choose to build out these capabilities internally or to contract for managed/service model approaches to support these risk management functions.

4.4.7 Additional Information

For additional information on practices for SCRM and acquisition artifacts please see *"Securing the Software Supply Chain: Recommended Practices for Customers"*, Section 2.1 'Procurement and Acquisition'.

4.5 How SBOM Risk Scoring can be used by Organizations to Reduce Risk

The first step to enabling Risk Scoring is operationalizing the use of SBOMs, as described in Section 5. SBOMs can be used to calculate the Risk Score factors through an established automation framework. The automation framework can be extended to a workflow system to establish auditable decision-making.

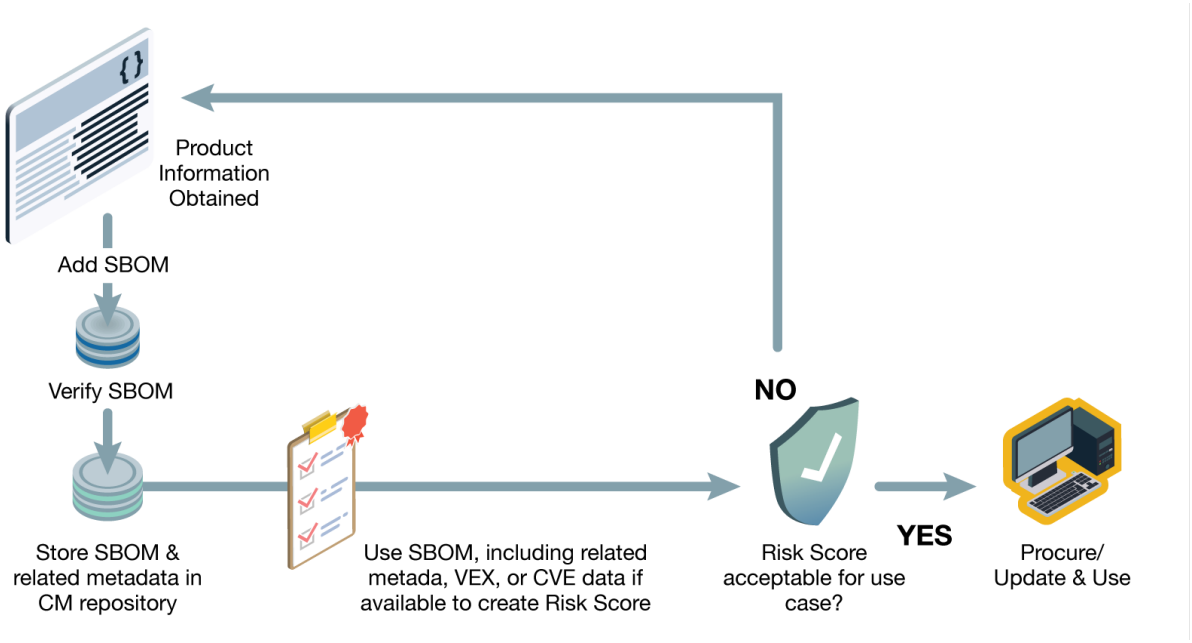

Figure 2: Operationalizing SBOMs for Risk Scoring

Additional information can be gathered from open sources.

During the establishment of tooling in support of Risk Scoring within the consuming organization, a weighting criterion should be developed. This criterion enables tailoring resulting scores based on the use case of the product. Effective weighting criteria would carefully consider the temporal point in time nature of scores against a given use case. There is no universal Risk Score that applies to all customer organizations.

4.5.1 Leveraging Risk Scoring for Supply Chain Risk Management and Enterprise Threat Management

The risk scoring methodology provides an approach to consistent communication of risk in support of SCRM practice within an organization. The risk score can be used in many types of analysis including:

- Compare products side by side on an even plane,
- Flexibility for consumer organizations to weight Risk Scores based on the needs of the organization,
- Summary of dependencies defined in the SBOM; how many levels and how many external organizations/projects does a product rely on,
- Share agreed upon risk levels within a community of practice or large enterprise.

Within an organization, Enterprise Threat Management (ETM) practice understanding the composition of the software within their environment is a key first step. The SBOM provides granular visibility at the software component level to expose potential threats posed by a new vulnerability. Unfortunately, once you start collecting SBOMs you quickly have a huge amount of data. By adding a Risk Scoring methodology to augment the SBOM, threats can be aggregated for higher-level analysis. This provides a time saving way for already over tasked ETM organizations to focus efforts surrounding software products.

5 Operationalizing SBOM

To fully maximize the value of SBOM requires organizational cyber policies and procedures that allow for successful agile automated implementation of SBOM consumption for all software in the enterprise. This is not dissimilar to other types of security data. Threat intelligence data, for example, has gone through a similar evolution and has similar diversity of maturity. Implementing appropriate operational processes will be a prerequisite to enabling decisions to minimize the risk made visible by SBOM content. The following set of mitigations will help reduce the risk associated with the software.

1. Identify vulnerable and/or exploitable versions of software and prioritize patching accordingly. Monitor those systems/software for malicious or anomalous behavior, determine a risk associated with these observations, and implement corrective behavior automatically. AI/ML models can help automatically detect malicious or anomalous behavior.
2. Use data collected from the SBOM and its analysis and incorporate it directly into your risk management processes to determine software supply risks and risk tolerance for your organization. This means also tying in mechanisms for applying countermeasures, mitigations, or other risk control activities.
3. The use of SBOM and SCRM processes works in concert with a “zero trust” architectural approach.

There are several ways to use the data provided by an SBOM once a vulnerability is discovered. First, evaluate the results in terms of likelihood and impact. Likelihood is a determination of the probability of an attack succeeding using the discovered vulnerability. Impact should consider both the immediate damage and long-term impact to the company brand, bottom line, and customer experience.

The four-quadrant approach is one effective way to evaluate open source vulnerabilities found in COTS software. For example, software with some vulnerabilities -- where the vulnerabilities are considered to have low impact and are unlikely to be exploited -- could be approved for purchase, renewal, or maintenance contract by simply accepting the low risk level. Obviously, software with a high impact, high likelihood of attack vulnerabilities may need to be rejected.

However, it is often not possible to reject any software critical to the business. While using SBOM data in the COTS procurement process is a relatively new discipline, the assumption here is that both the customer and the vendor will act in good faith to improve the security of the product and reduce

security risk over time. This assessment process can be applied to all deployed software. Figure 3 depicts a continuous decision workflow to follow once SBOM results are in-hand.

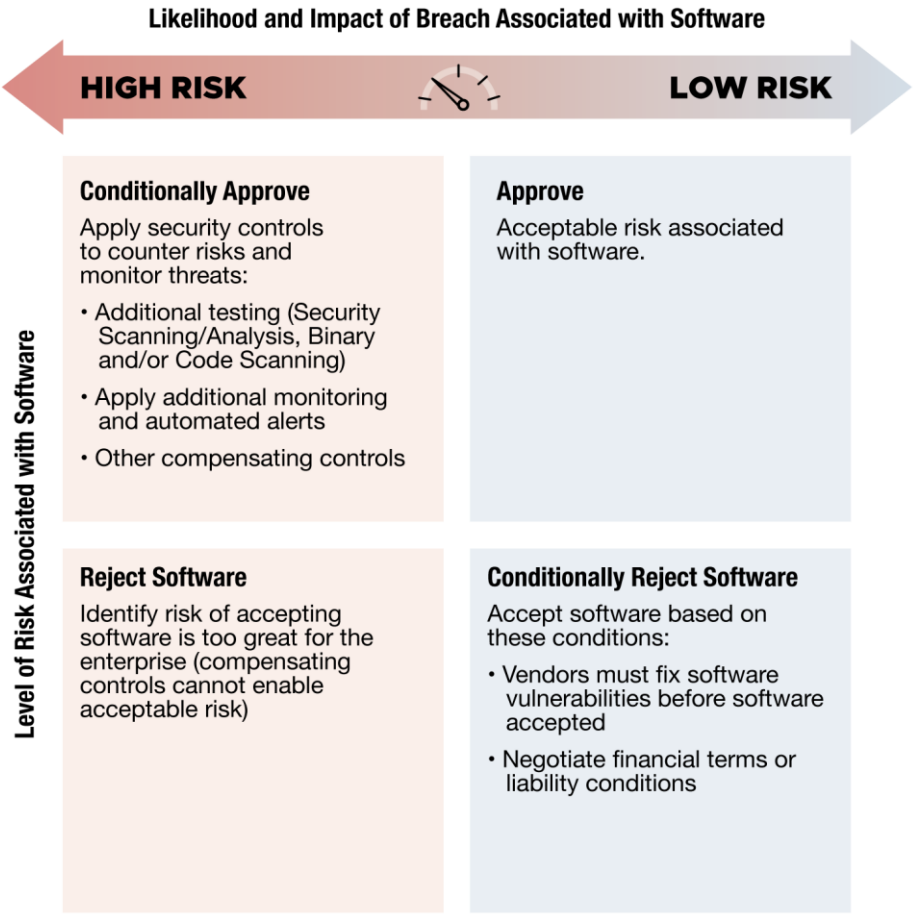

Figure 3: The 4 Quadrant Approach

6 Conclusion: SBOM Consumption Today and Tomorrow

- SBOM consumption is new and will evolve and scale.
- Emerging tools will help automate and scale.
- It is understandable that tools are not here yet—until recently, SBOM data was quite scarce, so there wasn't much need for open source software or proprietary SBOM consumption tools.
- Different organizations will focus on different risks that they can better manage with software transparency.
- The industry is still imagining use cases and expect more to emerge as SBOM becomes more common.
- There is still value in just asking for SBOMs, and just keeping SBOM data on hand to respond to emergency advisories.
- SBOM is just one part of software supply chain security— basic hygiene and paying attention to other C-SCRM guidance is still important.

- A vendor neutral open standard set of risk factors that can be aggregated into risk scoring for SBOMs should be developed.

Appendix A: References/Addendum

SPDX - https://spdx.github.io/spdx-spec/

SPDX2 - SPDX format 2.2.2

CycloneDX - https://cyclonedx.org

SWID ISO/IEC 19770 - https://www.iso.org/standard/65666.html & https://csrc.nist.gov/projects/software-identification-swid/guidelines

KEV - https://www.cisa.gov/known-exploited-vulnerabilities

CVE - https://cve.mitre.org/about/cve_and_nvd_relationship.html

VEX1 - https://ntia.gov/files/ntia/publications/vex_one-page_summary.pdf

VEX2- https://www.cisa.gov/sites/default/files/publications/VEX_Use_Cases_April2022.pdf

VEX3 - https://www.cisa.gov/sites/default/files/publications/VEX_Status_Justification_Jun22.pdf

SWID1 - https://datatracker.ietf.org/doc/draft-ietf-sacm-coswid/

LIO1 - https://libraries.io

LFX1 - https://lfx.linuxfoundation.org

CD1 - https://clearlydefined.io/

SBOM - https://www.cisa.gov/sbom

Risks and costs of treating SBOM data as confidential/classified/etc.

There are tradeoffs. See NTIA myths document: https://ntia.gov/files/ntia/publications/sbom_myths_vs_facts_nov2021.pdf

SPDX specific Risk Scoring fields and criteria

SBOM formats

SPDX format 2.2.2

https://www.ntia.gov/files/ntia/publications/sbom_formats_survey-version-2021.pdf)

Formats

https://becomingahacker.org/sboms-csaf-spdx-cyclonedx-and-vex-todays-cybersecurity-acronym-soup-5b2082b2ccf8

extra info SBOM formats 2021 https://www.ntia.gov/files/ntia/publications/sbom_formats_survey-version-2021.pdf

Appendix B: Acronym List

Acronym	Expansion
AI/ML	Artificial Intelligence / Machine Learning
CMDB	Configuration Management Database
CIPAC	Critical Infrastructure Partnership Advisory Council
CISA	Cybersecurity and Infrastructure Security Agency
CMT	Crisis Management Team
CNSS	Committee on National Security Systems
CPE	Common Platform Enumeration
CSAF	Common Security Advisory Framework
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DHS	Department of Homeland Security
EO	Executive Order
ER	Entity Resolution
ESF	Enduring Security Framework
ETL	Extraction, Transformation, and Loading
ETM	Enterprise Threat Management
FOCI	Foreign Ownership, Control or Influence
IAMR	Integrated Asset Management Repository
IMDRF	International Medical Device Regulators Forum
IT	Information Technology
KEV	Known Exploited Vulnerabilities
LFX	Linux Foundation
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NTIA	National Telecommunications and Information Administration
NVD	National Vulnerability Database
ODNI	Office of the Director of National Intelligence
OMB	Office of Management and Budget
ONCD	Office of the National Cyber Director
OSS	Open Source Software

Acronym	Expansion
OWASP	Open Web Application security Project
SAM	Software asset management
SBOM	Software Bill of Materials
SCA	Software Composition Analysis
SCRM	Supply Chain Risk Management
SIEM	security information and event management
SOC	Security Operations Center
SP	Special Publication
SPDX	Software Package Data Exchange
SWID	Software Identification
TTPs	Tactics, Techniques, And Procedures
UEM	Unified Endpoint Management
VEX	Vulnerability Exploitability eXchange

Appendix C: Glossary

Term	Definition
Compliance Risk	Risk of violating a regulatory requirement or a corporate policy
Decommissioning	Decommissioning is a strategic approach for systematically retiring outdated and costly legacy applications—without compromising business needs or compliance requirements.
Entity Resolution	The process of working out whether multiple records are referencing the same entity/component
License Risk	Risk of violating the terms of the component's license
SBOM Integrated Asset Management Repository	A database that allows the user to store, manage, retrieve and search SBOMs associated with integrated products.
Security Risk	Risk of a component exposing a vulnerability or being active exploited
open source	Open source software is typically developed via open collaboration, and its source code is made available by the authors/producers for anyone to use, examine, alter and/or redistribute.