Recommendations for Software Bill of Materials (SBOM) Management

Executive summary

The dramatic increase in cyber compromises over the past five years, specifically of software supply chains, prompted intense scrutiny of measures to strengthen the resilience of supply chains for software used throughout government and critical infrastructure. Several policies and working groups at multiple levels within the U.S. Government focus on this need to ensure the authenticity, integrity, and trustworthiness of software products. The office of the National Manager for National Security Systems (NSS), working in collaboration with other NSA organizations, researched and tested tools that manage Software Bills of Materials (SBOMs) as part of a Cybersecurity Supply Chain Risk Management (C-SCRM) strategy. This guidance includes important recommendations for SBOM management tool functionality derived from the research and evaluation of various SBOM management tools.

Fundamental to C-SCRM is leveraging a 'list of [software] ingredients' to understand and mitigate the cyber risks that software can pose to a user organization. SBOMs and SBOM management tools bridge this gap to support an improved cybersecurity posture. Specifically, users should leverage SBOMs, as part of a cybersecurity tool suite, to make:

- **Risk Management** decisions about acquiring and deploying software,
- **Vulnerability Management** decisions about software deployment and ongoing operations, and
- **Incident Management** decisions to detect and respond to new software vulnerabilities during vital operations.

This guidance can enable NSS software application owners and users to determine an appropriate management toolset that leverages SBOMs to achieve these tasks.

Contents

Introduction

As Software Bills of Materials (SBOMs) become more integral to Cybersecurity Supply Chain Risk Management (C-SCRM) standards, best practices for managing SBOMs will become critical in ensuring reliability of the software supply chain. Those standards and best practices are evolving, but no single solution, or set of solutions, has become ubiquitous. While a variety of software analysis tools exist for software developers, most address SBOMs peripherally or focus on SBOM production, rather than on managing software risk in acquiring and using software as a consumer.

The C-SCRM activities that this guidance focuses on are software **Risk Management, Vulnerability Management**, and **Incident Management** utilizing SBOMs**.** Within the sphere of software C-SCRM practices, **Risk Management** involves determining the authenticity, completeness, and trustworthiness of software being acquired and used. It includes understanding the applicable risks and comparing that information against the mission's risk profile to determine the software's suitability. **Vulnerability Management** allows IT security teams to adopt a more proactive security posture by identifying and resolving vulnerabilities before they are exploited. The goal of **Incident Management** is to reduce the organization's overall risk exposure, whether that occurs as part of the procurement process or operationally, once an incident has occurred.

Purpose and background

This cybersecurity information sheet (CSI) highlights best practices and provides recommendations for users of National Security Systems (NSS) to help them incorporate SBOM management functions suitable to their C-SCRM needs.

To achieve this outcome, the National Manager for NSS team researched and evaluated a variety of software analysis and SBOM management tools, specifically those that would help NSS users address software Risk Management, Vulnerability Management, and Incident Management tasks. These three management responsibilities are not point-in-time events, but rather, represent a continuous cycle as software is added to or retired from operation, as applicable updates become available or as vulnerabilities are discovered. [1]

Executive Order (EO) 14028: Improving the Nation's Cybersecurity focuses on improving the nation's cybersecurity, in particular for critical software and services used

across the federal government. [2] National Security Memorandum 8 (NSM-8) builds on the requirements established under EO 14028, assigning responsibilities to the National Manager for NSS and pointing to additional requirements needed for National Security Systems. [3] Separately, Executive Order 14017: Executive Order on America's Supply Chains focuses on strengthening U.S. supply chain resiliency, [4] and Department of Defense Instruction 5200.44 focuses DoD efforts on SCRM. [5] Further instruction from Committee on National Security Systems Directive (CNSSD) 505 assigns responsibilities and establishes the minimum criteria for the continued development, deployment, and sustainment of an SCRM program with the intent of protecting NSS. [6]

This CSI communicates the importance of utilizing an appropriate toolset to leverage SBOMs for cybersecurity and SCRM goals and to perform Risk Management, Vulnerability Management, and Incident Management tasks.

The following sections describe best practices and expected SBOM management functionality for NSS owners and operators. The information should guide future actions and decisions in acquiring and implementing an SBOM Management ecosystem with supporting tools for software suppliers and (U.S. Government) software consumers.

Recommendations

NSA recommends the following courses of action to enhance the 'state of the art' for SBOM usage. The recommendations are organized as general guidance for software suppliers and consumers followed by specific guidance for NSS users.

General recommendations for software suppliers

- The software provider community and their consumers should mature the mode of SBOM exchange to protect the intellectual property and product security of the software suppliers while ensuring authenticity, accuracy, timeliness, and efficiency of SBOM information transfer to software consumers.
- Both industry and government entities should expand SBOM research to better understand the minimum requirements for an SBOM to be beneficial, and share best practices to standardize solutions for other technology platforms susceptible to cyber supply chain attacks (for example, Operational Technology, Cloud/SaaS operations, Hardware/Firmware).

- Software developers must take ownership of their customers' security outcomes rather than treating each product as if it carries an implicit caveat emptor. EO 14028 and NSM-8 lay the foundations to enable users of National Security Systems and other critical systems to require software technology providers to make their products "Secure by Design and by Default." [7]

  SBOMs and SBOM management tools play a part in enforcing the requirement to make software secure by design as they provide a mechanism to determine software component risk and establish a level of confidence in the software's freedom from vulnerabilities.

General recommendations for software consumers

- Software consumers should leverage the following resources to ensure their suppliers are designing, developing, and delivering secure software:
  - National Institute of Standards and Technology (NIST) Special Publication (SP) 800-218: Secure Software Development Framework (SSDF), or an alternate secure software development framework that encompasses and expands on it to accommodate NSS-specific considerations. [8]
  - Office of Management and Budget (OMB) Memorandum M-23-16, [9] which updates OMB Memorandum M-22-18, [10] requires all federal agencies to use only software suppliers that comply with NIST SP 800-218.
  - The Software Component Verification Standard (SCVS) provides additional resources that help NSS users drive the requirement for SBOM data upstream to software supplier sources to ensure as complete and accurate as possible information about software component dependencies. [11]
  - Software consumers can also leverage controls identified in the Enduring Security Framework's Securing the Software Supply Chain: Recommended Practices for Software Bill of Materials Consumption [12] and from Open Web Application Security Project (OWASP), [13] in addition to the Cybersecurity and Infrastructure Security Agency's (CISA) draft Secure Software Development Self-Attestation Form. [14]

- Further guidance from collaborations among CISA, the National Telecommunications and Information Administration (NTIA), or the National Manager for NSS, and the cybersecurity community that establishes common practices and standardization for automation of SBOM management.

Specific guidance for NSS

The following recommendations build upon information provided by the NTIA Formats and Tooling Working Group in its Software Consumers Playbook. [15] The NTIA's focus is broader, and in general, encompasses systems with a lower level of concern and protection considerations than NSS. The Federal Acquisition Regulation (FAR) [16] and Defense Federal Acquisition Regulation Supplement (DFARS) [17] contract clauses have not caught up with the stipulations regarding the inclusion of SBOMs as a requirement for software development contracted as a service. The National Manager recommends that NSS owners develop and require contract language for the following:

- Inclusion of software component information containing, at a minimum, the NTIA required field for each component in all delivered software. Additional component details should be requested and required as the state of the industry matures. This requirement includes the expectation that a supplier enumerate third-party software dependencies (both open source and proprietary) incorporated into the supplier's product. Where the NTIA required information is not available or verifiable for open source items, derived components must be disclosed. This complete listing of component information helps provide the requisite transparency, not only for software risk management, but also for vulnerability and incident management.
- Identification of runtime or other dependencies for software operation that are not specifically part of the software components listed in the SBOM.
- Inclusion of a container manifest should be required for all software with container components.
- Use of digital signatures or authenticated hashes to validate component (and SBOM) integrity.
- For NSS-related software specifically developed under contract, the SBOM should be generated using source code from the build stage. If not included in

the SBOM itself, runtime dependency information must also be documented and provided as part of the software deliverable.

- For all delivered software, whether the software is acquired or developed under contract, NSS owners should verify the accuracy and completeness of the SBOM utilizing an SBOM management tool as part of its delivery acceptance.
- For delivered software binaries, NSS owners should seek contract agreements that provide the NSS owners limited rights to reverse-engineer the software for the specific purpose of validating SBOMs and resolving any discrepancies in a mutually agreed-upon manner.
- Inclusion of contract metrics that enable tracking and assessment of the software suppliers' "secure by design" performance.

Best practices

In addition to the recommendations above, NSS owners should implement the following best practices:

- Establish a secure, common exchange point between software suppliers and consumers.
- Obtain SBOMs and associated vulnerability and code quality information (e.g., CVEs and CWEs) from analysis of software binaries. While more work is needed to improve the accuracy and timeliness of analysis results, this approach presents a way forward in managing the risks from legacy software that does not have accessible source code/build information for the SBOM.
- Integrate data from SBOMs with NSS acquisition security, asset management, threat intelligence, and vulnerability management.

Recommended tool functionality

This section describes important functionality considerations for consumer-side SBOM management tools. Many current tools do not include all the functionality listed below. An organization should identify the functionality that is appropriate for their environment and then select the tool(s) that supports that functionality.

As part of identifying appropriate SBOM management tool functionality, an organization should assess the tools' ease of understanding (presenting information at the right level

of need and in an easily understood manner) and ease of use (reducing the number of other applications or information required outside the tool to complete the task). Tool functionality that is not easily understood or used is often not a good fit for an organization's environment and processes. SBOM tool providers should continually focus on improving their tools' ease of understanding and ease of use.

The tool should be able to:

SBOM input

- Support and manage all SBOM format versions (e.g., the latest Cyclone DX (CDX) version is v1.4 and the latest SPDX version is 2.1). Ideally, it should import and support both formats.
- Import SBOMs as JSON or XML file types. Ideally, it should import JSON, XML, and CSV file types.
- Check SBOM structure and syntax for compliance with the appropriate format specification (including for a specific version) when importing SBOMs.
- Display information regarding the SBOM's compliance with applicable structure and syntax rules for the SBOM format and version. Ideally, it should alert users with a display indicating the assessed quality level, while giving the option to continue the import or not.
- Include an auto correction option to assist the user in normalizing and correcting an SBOM file being imported.

SBOM output

- Export SBOMs using either the CDX or SPDX format. Ideally, it should include the ability to export both formats.
- Export SBOMs as JSON or XML file types. Ideally, it should include the ability to export JSON, XML, and CSV file types.
- Convert one SBOM format to another.
- Convert one SBOM file type to another.
- Aggregate multiple SBOMs from the SBOM tool's repository into one SBOM.

Generating SBOMs

- Generate SBOMs from various types of software development process outputs (for example, from a software build environment, from analysis of a binary file, from system registry query, etc.).

SBOM component handling

- Display NTIA-minimum SBOM fields (Supplier Name, Component Name, Component Unique Identifier (CPE, PURL)/Hash, Component Version, Component Dependency Relationship, Component Author) for each component.
- Enrich SBOM information using additional reference sources. Ideally, it should provide visual cues indicating external information was used to enrich the SBOM data and references source sites of the enrichment data.
- Include mechanisms to graphically represent component dependencies.
- Display component provenance information, including external enrichments.

Validation of SBOM and SBOM component integrity

- Capture and display hash information for each component. Ideally, this validation should provide a digital signature for the SBOM and provenance information for each component, as well as the Component Hash, and a PURL or CPE pointer.
- Include links to information sources where provenance data was gathered (supports ability to verify integrity and assess risk).

Vulnerability tracking and analysis

- Provide daily updates from the National Vulnerability Database (NVD) and other vulnerability data. Ideally, these updates should provide continuous extracts and analysis from associated cyber threat intelligence (CTI) and SBOM data enhancement services.
- Notify users of new vulnerabilities and updates, including alerts of emergent critical vulnerabilities and their severity. Ideally, these notifications should clearly distinguish between a new vulnerability and an update to an existing vulnerability, and provide additional information to prioritize vulnerability responses along with risk remediation guidance.

- Integrate various sources of threat intelligence in addition to the various software vulnerability/weakness databases.
- Provide a flexible policy engine, including the ability to apply and update organization-specific policy rules. Ideally, this customization should enable integration of threat intelligence as policy rules.
- Provide multiple ways to identify and research an emergent vulnerability's existence in the user's SBOM repository/asset inventory. Ideally, it should quickly identify specific networks or endpoints containing the software and configurations affected by a newly discovered vulnerability.
- Support and track the timeliness of vulnerability remediation (including configuration management/traceability to a new SBOM to distinguish the vulnerable, replaced software from the remediated/hardened replacement software).

Distinguishing identified vs. exploitable vulnerabilities

- Indicate whether a vulnerability is actually exploitable and support accompanying evidence and justification for non-exploitable claims. Ideally, it should annotate and update information about the exploitability of a component vulnerability using Vulnerability Exploitability eXchange (VEX) format.

User interface

- Follow Human Computer Interface (HCI) standards.
- Incorporate accessibility features.
- Provide mechanisms that make the information easy to assess and, if desired, enable the user to easily delve further (often by hovering the cursor over icons or clicking on icons with links) to view the next level of detail.
- Provide easily understandable graphic representation methods and formats to convey information attributes about software components, vulnerabilities, licenses, supplier organizations, users, and user organizations.
- Provide multiple ways to 'drill down' and obtain additional information for software component provenance, vulnerability, license, and risk status.

- Provide means to create structured groupings or categories of SBOMs to facilitate asset tracking, vulnerability management, incident management, etc.
- Provide the ability to filter/sort/group SBOM information according to user-selectable attributes (such as, by software/BOM type, software/BOM source, software/BOM PoC; component type, component package, component age, component versions, security trend; vulnerability severity, vulnerability count; and organization level, license type, violation).

Output forms and methods

- Output standardized reports regarding component attributes, vulnerability information, license information, and component supplier information.
- Export dependency information in graphic and/or text format.
- Output graphic representations of analysis results.
- Ideally, provide a modular means to export specific text and graphics (whether from the SBOM itself or derived from analysis and enhancement processes) for use in external communications.

SBOM versioning and configuration management support

- Include a scalable configuration management capability for SBOMs. Minimally, it should include mechanisms to organize SBOMs, maintain version history, and track changes of SBOMs/software.
- Include user-tailorable mechanisms to organize SBOMs on multiple information attributes (such as by organization, software supplier, type of software, type of BOM, license type, etc.).
- Include a trend graphic showing the number of vulnerabilities for each severity level across each component version and report whether the numbers of component vulnerabilities are increasing or decreasing with each version release.
- Compare SBOM versions for the same software and highlights differences (such as by different components or different component versions, different sources, etc.).

Integration and workflow with other systems

- Employ "API First" design to facilitate import and export of information with other systems. Ideally, information elements within the tool should be individually exportable/downloadable.
- Integrate with multiple types of SBOM sources and other data that can be combined together for analysis.
- Leverage format-agnostic, independent, stateless, and scalable API capabilities (such as REST) to automate processes/workflow.
- Support a secure, integrated Producer/Consumer exchange ecosystem.

Supporting access to data sources

- Integrate AI/ML engines and associated ‘data lakes’ that analyze SBOM component information against diverse types of threat signatures and patterns.
- Include an updatable library of open source software licenses that the SBOM management tool identifies and tracks where applicable.

Scalable architecture

- Include mechanisms to support distinct sub-organizations within an enterprise that may have different risk tolerance rules or policies.
- Handle other types of BOMs.
- Be part of, or support, a suite of tools that work together to accomplish Risk Management, Vulnerability Management, and Incident Management activities.

SBOM tool setup and configuration

- Provide mechanisms and supporting materials to easily download, setup, and integrate in Linux or Microsoft environments. Ideally, it should support both environments.

Works cited

[1] National Institute of Standards and Technology. Special Publication 800-161 Rev. 1 Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations. 2022. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161r1.pdf

[2] Executive Office of the President. Executive Order 14028: Improving the Nation's Cybersecurity. 2021. https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nationscybersecurity

[3] White House. National Security Memorandum 8: Memorandum on Improving the Cybersecurity of National Security, Department of Defense, and Intelligence Community Systems. 2022. https://www.whitehouse.gov/briefing-room/presidential-actions/2022/01/19/memorandum-on-improving-the-cybersecurity-of-national-security-department-of-defense-and-intelligence-community-systems

[4] Executive Office of the President. Executive Order 14017: Executive Order on America's Supply Chains. 2021. https://www.whitehouse.gov/briefing-room/presidential-actions/2021/02/24/executive-order-on-americas-supply-chains/

[5] Department of Defense (DoD). DoD Instruction (DODI) 5200.44: Protection of Mission Critical Functions to Achieve Trusted Systems and Networks (TSN). 2018. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/520044p.pdf

[6] Committee on National Security Systems (CNSS). CNSS Directive 505: Supply Chain Risk Management (SCRM). 2021. https://www.cnss.gov/CNSS/issuances/Directives.cfm

[7] Cybersecurity and Infrastructure Security Agency (CISA) and partners. Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software. 2023. https://media.defense.gov/2023/Apr/13/2003198917/-1/-1/0/Shifting-the-Balance-of-Cybersecurity-Risk-Principles-and-Approaches-for-Secure-by-Design-Software.PDF

[8] National Institute of Standards and Technology. Special Publication 800-218: Secure Software Development Framework (SSDF): Recommendations for Mitigating the Risk of Software Vulnerabilities. 2022. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf

[9] Office of Management and Budget (OMB). OMB Memorandum M-23-16: Update to Memorandum M-22-18, Enhancing the Security of the Software Supply Chain through Secure Software Development Practices. 2023. https://www.whitehouse.gov/wp-content/uploads/2023/06/M-23-16-Update-to-M-22-18-Enhancing-Software-Security.pdf

[10] Office of Management and Budget (OMB). OMB Memorandum M-22-18: Enhancing the Security of the Software Supply Chain through Secure Software Development Practices. 2022. https://www.whitehouse.gov/wp-content/uploads/2022/09/M-22-18.pdf

[11] Open Worldwide Application Security Project (OWASP). Software Component Verification Standard (SCVS). 2023. https://scvs.owasp.org/

[12] Enduring Security Framework. Securing the Software Supply Chain: Recommended Practices for Software Bill of Materials Consumption. 2023. https://media.defense.gov/2023/Nov/09/2003338086/-1/-1/0/SECURING%20THE%20SOFTWARE%20SUPPLY%20CHAIN%20RECOMMENDED%20PRACTICES%20FOR%20SOFTWARE%20BILL%20OF%20MATERIALS%20CONSUMPTION.PDF

[13] Open Worldwide Application Security Project (OWASP). OWASP. 2023. https://owasp.org/

[14] Cybersecurity and Infrastructure Security Agency (CISA). Request for Comment on Secure Software Development Attestation Common Form. 2023. https://www.cisa.gov/secure-software-attestation-form

[15] National Telecommunications and Information Administration (NTIA). Software Consumers Playbook: SBOM Acquisition, Management, and Use. 2021. https://www.ntia.gov/files/ntia/publications/software_consumers_sbom_acquisition_management_and_use_-_final.pdf

[16] General Services Administration. Federal Acquisition Regulation. 2023. https://www.acquisition.gov/browse/index/far

[17] Department of Defense. Defense Federal Acquisition Regulation Supplement (DFARS) and Procedures, Guidance, and Information (PGI). 2023. https://www.acq.osd.mil/dpap/dars/dfarspgi/current/index.html

Disclaimer of endorsement

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government, and this guidance shall not be used for advertising or product endorsement purposes.

Purpose

This document was developed in furtherance of NSA's cybersecurity missions, including its responsibilities to identify and disseminate threats to National Security Systems, Department of Defense, and Defense Industrial Base information systems, and to develop and issue cybersecurity specifications and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

Contact

Cybersecurity Report Feedback: CybersecurityReports@nsa.gov
General Cybersecurity Inquiries or Customer Requests: Cybersecurity_Requests@nsa.gov
Defense Industrial Base Inquiries and Cybersecurity Services: DIB_Defense@cyber.nsa.gov
Media Inquiries / Press Desk: NSA Media Relations: 443-634-0721, MediaRelations@nsa.gov