Guidance on Introduction of Software Bill of Materials (SBOM) for Software Management

Provisional Translation

Background and Purpose

- As software supply chains become more complex and the use of open source software (OSS) becomes more common, vulnerability management and license management in software are becoming increasingly important.
- As a method of software management, a management method using the Software Bill of Materials (SBOM) has been attracting attention.
- Through the Proof-of-Concept (PoC) in several industrial fields, it was confirmed that SBOMs can be used for efficient software management, but it also became clear that various hurdles exist in the actual implementation of SBOMs.
- This Guidance **provides basic information about SBOM, as well as myths and facts about SBOMs**. In order to assist companies in introducing SBOM, it also **presents actions for the SBOM introduction and key points to be aware of when introducing an SBOM**.

Target Audience

- Mainly software suppliers dealing with packaged software and embedded software*.
  - ✓ Software development departments
  - ✓ Product security departments (PSIRT)
  - ✓ Executives/Management
  - ✓ Legal and intellectual property departments

※ In particular,

- Organizations facing challenges in software vulnerability management
- Organizations that have heard of SBOM but do not understand the specifics and benefits of SBOM
- Organizations that understand the need for SBOM but do not know what they need to do to implement it.

Benefits of SBOM

- **Vulnerability management**
  - ✓ Reduce residual vulnerability risk
  - ✓ Reduce vulnerability response time
  - ✓ Reduce cost of vulnerability management
- **License management**
  - ✓ Reduce risk of license violations
  - ✓ Reduce cost of license management
- **Development productivity**
  - ✓ Prevent development delays
  - ✓ Reduce development costs

SBOM Introduction Process

Phase 1 Environment and system development phase

- **1-1. Clarification the scope of the SBOM application**
  - ✓ Organize information (languages, development tools, configuration diagrams, etc.) related to the target software.
  - ✓ Clarify the scope of the SBOM application based on the organized information.
- **1-2. SBOM tools selection**
  - ✓ Organize the viewpoints for the selection of SBOM tools and evaluate and select tools based on the viewpoints. (Examples of viewpoints: functions, performance, analyzable information, analyzable data format, cost, supported formats, component analysis method, support systems, coordination with other tools, etc.)
- **1-3. SBOM tools installation**
  - ✓ Install and configure the SBOM tool by reviewing the tool's instruction manual, README file, etc.
- **1-4. Learning about SBOM tools**
  - ✓ Learn how to use the SBOM tool by reviewing the tool's instruction manual, README file, etc.
  - ✓ Record know-how on using the tool and an overview of each function and share them within the organization.

Phase 2 SBOM production and sharing phase

- **2-1. Component analysis**
  - ✓ Scan the target software by using SBOM tools and analyze component information. Based on the result, check if there are any false positives and false negatives.
  - ✓ By using SBOM tools, component analysis and SBOM production can be performed more efficiently than would be possible manually.
  - ✓ Using a package manager may enable the identification of granular components that cannot be identified by SBOM tools.
- **2-2. SBOM production**
  - ✓ Determine the requirements such as items, format, and output file format to be created and create an SBOM that fulfills the requirements.
- **2-3. SBOM sharing**
  - ✓ Consider the most appropriate means of sharing the SBOM with users and/or suppliers and share the SBOM according to this method if necessary.

phase 3 SBOM use and management phase

- **3-1. Vulnerability management and license management, etc.**
  - ✓ Based on the SBOM tool output on vulnerabilities, conduct vulnerability management such as evaluating severity, assessing impact, fixing vulnerabilities, confirming residual risks, and providing information to relevant organizations.
  - ✓ Based on the SBOM tool output on licenses, confirm whether any OSS license violations have occurred.
- **3-2. SBOM information management**
  - ✓ Appropriately manage information contained in SBOM along with SBOM itself.
    *It is an effective practice to have a department equivalent to a PSIRT to manage SBOM.
  - ✓ SBOM tools that automatically update and provide notification on vulnerability information enable immediate identification of information about new vulnerabilities. If automatic management using the tool is not possible, the operation should be covered by other methods, including designating a person in charge to manage this.

To Management – Toward Introducing SBOMs

Background: Increasing Threats to Software Supply Chain

- **Threats to software have increased in recent years** as software supply chains have become more complex and the use of open source software (OSS) has become more common. The Apache Log4j vulnerability discovered in December 2021 affected the entire world. According to one study, the average annual increase in software supply chain attacks between 2019 and 2022 was 742%[*1].
- **Software security threats have a significant impact on business operations**. For example, companies affected by the SolarWinds cyberattack lost an average of about 11% of their annual revenue[*2]. In other cases, remaining vulnerabilities in products have led to product recalls and sales suspensions.
- To strengthen cybersecurity against software and link it to the trust and safety of the company, it is important to manage software properly.

Overview and Benefits of using SBOM in Software Management

- In response to these threats to the software supply chain, a **management method using the Software Bill of Materials (SBOMs)** is attracting attention as a method of software management to increase software transparency.
- **An SBOM is a machine-readable inventory of software components and dependencies** and is being **introduced by an increasing number of companies worldwide. Regulations are beginning to be considered in some industries**, such as the medical device field.
- While software management requires an enormous amount of information, the introduction of a machine-readable SBOM can **reduce the cost and workload required for software management**, which in turn **leads to higher development productivity**. In fact, in a PoC conducted by the Ministry of Economy, Trade and Industry (METI) in the medical device field, **vulnerability management using an SBOM reduced workloads by approximately 70% compared to manual management**.
- In addition, as a benefit to vulnerability management, the creation and ongoing management of SBOMs is expected to increase software transparency and **reduce the risk of residual vulnerabilities, as well as increase the efficiency of vulnerability response through the supply chain**.
- Furthermore, as an advantage in license management, an **SBOM will help reduce the risk of license violations by managing OSS license information**.
- While the PoC confirmed the benefits of SBOMs, it also revealed various hurdles to overcome in the actual implementation of SBOMs.

Purpose of the Guidance

- This Guidance provides **basic information on an SBOM** and **presents actions for the SBOM introduction and key points to be aware of when introducing an SBOM** in order to support efficient and effective SBOM implementation by companies.

Target Audience

- Management and departments involved in software security at software suppliers, such as a development department, design department, and product security department (e.g., PSIRT)
  *In particular, this Guidance focuses on those at the SBOM beginner level.

*1: Sonatype, "8th Annual State of the Software Supply Chain Report, *2: IronNet, "2021 Cybersecurity Impact Report."

Myths and Facts about SBOM

The Guidance includes the following myths and facts about SBOM, based on a document published by the U.S. NTIA* and the results of the PoC conducted in Japan.

Myth: Only the components directly used by the target software should be subject to SBOM management

[Fact] Vulnerability management may be insufficient if the components recursively used by direct components are excluded. Discussions by experts are ongoing regarding the "depth" of SBOM (i.e., up to what level of components should be included in SBOM).

Myth: No special consideration is needed to select SBOM tools

[Fact] Regarding tools to support SBOM production, several commercial tools and OSS tools provided as OSS are already available. By using OSS tools, the tools themselves can be obtained at no cost, but compared to commercial tools, the manuals and support for introduction and utilization are often limited, which may result in significant costs incurred in learning how to use the tools, and there is a possibility that the purpose of SBOM implementation cannot be achieved. It is necessary to select tools based on the objectives of the company's SBOM implementation.

Myth: SBOM tools can be utilized to fully identify the components contained in the target software

[Fact] Although SBOM tools can be used to efficiently create SBOMs, there may be cases where false positives or false negatives in the production of SBOMs, making it impossible to create accurate SBOMs. Therefore, it is important to consider other ways to reaffirm the accuracy of the SBOM (for example, reviewing the SBOM created by the tool).

Myth: There is a need to respond to all vulnerabilities output by SBOM tools

[Fact] It is necessary to prioritize vulnerabilities when responding to risks based on the output. Prioritization should occur based on the impact of the vulnerability, the results of the risk assessment, and the cost of responding to the vulnerability. In doing so, it should be noted that not all vulnerabilities are available for use, and some vulnerabilities that exist are not affected.

Myth: Granularity of the SBOM components should be standardized throughout the supply chain and only the necessary component information should be retained

[Fact] Currently, the granularity of "affected software" in vulnerability information databases such as Japan JVN and U.S. NVD is not systematized, and limiting the granularity of components may lead to false negatives in identifying vulnerabilities. Therefore, it is an effective practice to retain component information not only for OSS but also for in-house products.

Myth: SBOM only covers packaged and embedded software

[Fact] Not only software but also IT systems can be covered by an SBOM. In addition, SBOMs for online applications such as SBOMs for container images, SBOMs for SaaS software, and SBOMs for cloud services are also being discussed mainly in the U.S.

Myth: Only three formats of SBOM are allowed: SPDX, CycloneDX, and SWID tags; SBOMs based on proprietary formats are not allowed

[Fact] According to the definition of the U.S. NTIA, an SBOM is "a machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships." Even proprietary formats can be considered SBOMs if they meet this definition. However, since the "automation support" is positioned as the "Minimum Elements" of SBOM, and since automated processing improves efficiency, it is desirable to consider adopting an automatically processable format whenever possible.

* NTIA (National Telecommunications and Information Administration), SBOM Myths vs. Facts https://www.ntia.gov/files/ntia/publications/sbom_myths_vs_facts_nov2021.pdf

[SBOM Introduction Process] Phase 1: Environment and system development phase

- In the environment and system development phase, select an SBOM tool to be utilized after clarifying the scope of the SBOM application with respect to the target software.
- After installing and configuring the SBOM tool, learn about the SBOM tool in preparation for the SBOM production.

Phase 1 Environment and system development phase

Step	Actions for the introduction of SBOM	Points to be aware of for SBOM introduction
1-1: Clarification the scope of the SBOM application	☐ Clarify information about the target software, such as information about development language, component type, development tools, etc. ☐ Create an accurate configuration diagram of the target software and visualize the target of the SBOM application. ☐ Clarify the contractual form and business practices with users and suppliers of the subject software. etc.	● By utilizing the knowledge of developers inside and outside the organization, it is possible to efficiently collect information about the target software. ● The scope of risk management can be clarified by creating an accurate configuration diagram of the target software and by visualizing the target of the SBOM application.
1-2: SBOM tools selection	☐ Organize the viewpoints for the selection of SBOM tools considering the development language of the target software and the constraints within the organization. (Examples of selection viewpoints: functions, performance, analyzable information, analyzable data format, cost, supported formats, component analysis method, support systems, coordination with other tools, form of provision, user interface, operation method, supported software languages, Japanese support, etc.) ☐ Evaluate and select multiple SBOM tools based on the organized viewpoints.	● Since the use of multiple SBOM tools can be inefficient, it is advisable to consider whether the minimum number of SBOM tools should be used for a given purpose. ● Commercial SBOM tools are generally expensive. On the other hand, OSS SBOM tools may require a large number of workloads for implementation and operation due to the lack of information about environmental maintenance and learning. ● Compared to commercial SBOM tools, OSS SBOM tools often have limited functions and performance: recursive use parts cannot be detected, there are limitations on readable SBOM formats, license false negatives occur, or the installation environment is limited. etc.
1-3: SBOM tools installation	☐ Check the requirements of the environment where the SBOM tool can be installed and set up the environment. ☐ Check the instruction manual and README file of the tool and then implement and configure an SBOM tool.	● In the case of commercial SBOM tools for which a support system is in place, the implementation and configuration of a tool can be done efficiently by contacting the sales agent or tool vendor and receiving their assistance. ● OSS SBOM tools may require the burden of trial-and-error configuration because information about tool construction and configuration may be lacking. Effective implementation and configuration of an OSS SBOM tool can be achieved by obtaining assistance from companies that provide support services related to OSS tools, if necessary. etc.
1-4: Learning about SBOM tools	☐ Learn how to use SBOM tools by checking the instruction manual and README file of the tool. ☐ Record know-how on how to use the tool and the outline of each function and share them within the organization.	● With commercial SBOM tools that have a support system, users can learn how to use the tools efficiently by making inquiries to their sales agents or tool vendors. ● By using tools through trial and error by creating sample SBOMs, users can learn how to use their tools efficiently.

[SBOM Introduction Process] Phase 2: SBOM production and sharing phase

- In the SBOM production and sharing phase, use the SBOM tool to analyze the components and then produce an SBOM. Since the component analysis results may contain false positives or false negatives, the contents must be reviewed.
- In addition, consider sharing the SBOM with users and/or suppliers of the target software.

Phase 2 SBOM production and sharing phase

Step	Actions for the introduction of SBOM	Points to be aware of for SBOM introduction
2-1: Component analysis	☐ Scan the target software and analyze the component information using an SBOM tool. ☐ Examine the analysis log of the SBOM tool and check whether the analysis has been correctly executed without any false positives or false negatives caused by errors or lack of information. ☐ Check the component analysis results to see if there are any false positives and false negatives.	● SBOM tools can be used to analyze components and create SBOMs more efficiently than the manual method. The effect of using an SBOM tool is greater when the number of components is larger. ● In some cases, it is effective to use the configuration information of a package manager. In some cases, the package manager may also be used to identify granular components that cannot be identified by an SBOM tool. ● False positives and false negatives of components may occur. For example, components such as symbolic links and runtime libraries, deep hierarchical components, and components used only in specific fields may not be detected. Even if components are identified, their version information may be wrong. ● The output results differ, depending on the component analysis method in the SBOM tool. In the case of analysis based on dependencies, the possibility of false detection is extremely low, but in the case of other analysis methods, there is a possibility of false positives and false negatives. etc.
2-2: SBOM production	☐ Determine the requirements for the SBOM to be produced, such as items, format, and output file format. ☐ Produce an SBOM that satisfies the requirements, by using the SBOM tool.	● Considering the purpose of creating and sharing an SBOM, full accurate information should be included in the SBOM. ● When a component is used that is provided by a third party, such as an OSS community, it may be able to receive an SBOM of the component. However, if the component is used after being modified within the organization, it will not be able to use the provided SBOM as it is. ● By setting the names in the SBOM from the viewpoint of SBOM users, it is possible to eliminate rework after the SBOM is shared.
2-3: SBOM sharing	☐ Share an SBOM with the users and/or suppliers of the target software as necessary after determining the method of sharing the SBOM. ☐ Consider using electronic signature technology or other technologies to prevent falsification of the sharing of SBOM data.	● Different SBOM sharing methods may be adopted, depending on the SBOM tool used by the supplier. ● Various SBOM sharing methods will be available to different users. When sharing an SBOM with users, it is necessary to examine the advantages and disadvantages of each SBOM sharing method.

[SBOM Introduction Process] Phase 3: SBOM use and management phase

- In the SBOM use and management phase, implement vulnerability management, license management, and other operations based on the produced SBOM.
- In addition, even after the SBOM is produced, properly manage the information contained in the SBOM and the SBOM itself.

Phase 3 SBOM use and management phase

Step	Actions for the introduction of SBOM	Points to be aware of for SBOM introduction
3-1: Vulnerability management and license management, etc.	☐ Based on the output of the SBOM tool, assess the severity, evaluate the impact, fix the vulnerabilities, check the residual risk, and provide information to the relevant organizations. ☐ Based on the output of the SBOM tool, check whether there is any violation of the OSS license.	● The vulnerability information and license information outputted by the SBOM tool may be incorrect, so it is necessary to check the output results. ● If the EOL of a component cannot be identified by the SBOM tool, it is necessary to investigate it separately.
3-2: SBOM information management	☐ Keep the created SBOM for a certain period of time, including the change history, so that it can be referred to in case of inquiries from outside the company. ☐ Manage the information contained in the SBOM and the SBOM itself appropriately.	● Information about new vulnerabilities can be immediately obtained by using an SBOM tool that automatically updates and notifies vulnerability information. If automatic management using a tool is not possible, it is necessary to cover the situation in terms of operation by appointing a separate person in charge, but this requires more workload. ● SBOMs can be most effectively managed by the department corresponding to PSIRT in the organization, or by the quality control department if there is no department corresponding to PSIRT.