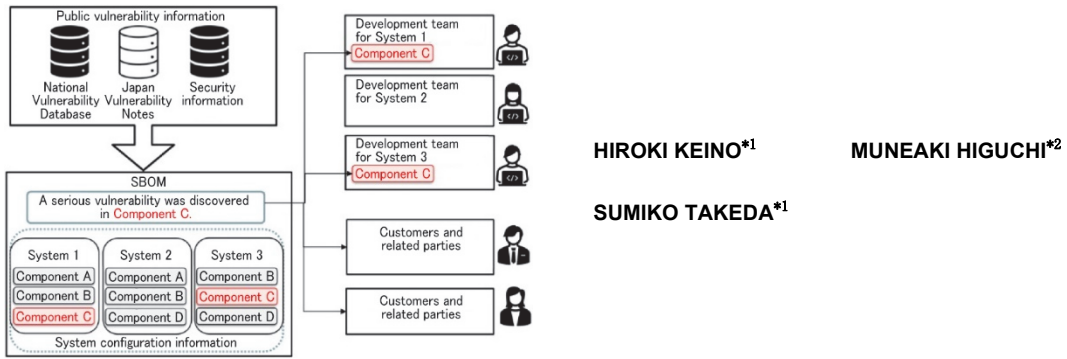


Vulnerability Management with Software Bills of Materials (SBOMs) to Enhance Software Security



As a means of enhancing the security of the software supply chain, vulnerability management using software bills of materials (SBOMs) is coming under the spotlight. Vulnerabilities in software can be detected by checking SBOMs against open vulnerability databases. Vulnerabilities that require early action must be automatically and appropriately identified and addressed from among the more than 50,000 vulnerabilities registered annually. In this report, we introduced a prioritization method based on decision tree analysis, in which information on susceptibility to attacks and damage caused by attacks are used in addition to the severity of vulnerabilities, and confirmed the effectiveness of the method.

[Click here for "Material Issues for MHI Group"](#)

1. Introduction

While the importance of software in products is increasing, the complexity of supply chains and the use of Open Source Software (OSS) are becoming more common. It is therefore required to promptly identify exploited vulnerabilities and serious incidents to ensure product safety and security. For this purpose, the use of Software Bills of Materials (SBOMs) is attracting attention, as they visualize the information on the software's components including dependencies, and appropriate measures can be taken against risks.

This paper describes a vulnerability management method using SBOM, including the background of SBOM utilization, a software vulnerability detection method using SBOM, and then a stratification method using decision tree analysis for the detected vulnerabilities. The effectiveness evaluation results of this stratification are also presented.

2. Background on use of SBOMs in products

When it comes to developing a product, it is not practical to create software from scratch every time a new functionality is needed. A common method is to create a library of widely applicable functions and reuse them across multiple projects. The utilization of third-party libraries or OSS is also becoming common these days, rather than a company creating its own library. Therefore, even for originally developed software, it is difficult to know what libraries are included therein. The Log4Shell issue in 2021 led to a huge and costly investigation by manufacturers around the world.

Having learned a lesson from this incident, people are turning their attention to SBOMs for use in software vulnerability management. In Europe, the Cyber Resilience Act (CRA)⁽¹⁾ was notified with the aim of enhancing cybersecurity throughout the life cycle of digital products. The CRA specifies drawing up of an SBOM in Annex I part II Vulnerability Handling Requirements. Meanwhile, in Japan, the Ministry of Economy, Trade and Industry also issued the Guidance on Introduction of SBOM for Software Management⁽²⁾.

*1 Research Manager, Control Systems Research Department, Research & Innovation Center

*2 Control Systems Research Department, Research & Innovation Center

An SBOM is a list of software components such as software libraries and modules that make up the software and their relationships, as well as supplementary information. Specifically, it includes information such as names, versions and developers, as well as the software's information concerning libraries that are unintentionally incorporated to establish dependencies. SBOMs are generated in machine-readable formats such as XML. The use of SBOMs is therefore expected to shorten the time for identifying vulnerabilities and verifying the effects of the patches (**Figure 1**).

Moreover, sharing SBOMs between different actors across the supply chain, from upstream to downstream, can achieve better transparency of software. SBOMs are therefore expected to serve as a solution to the issues in the vulnerability management of software in which third-party libraries and OSS are used.

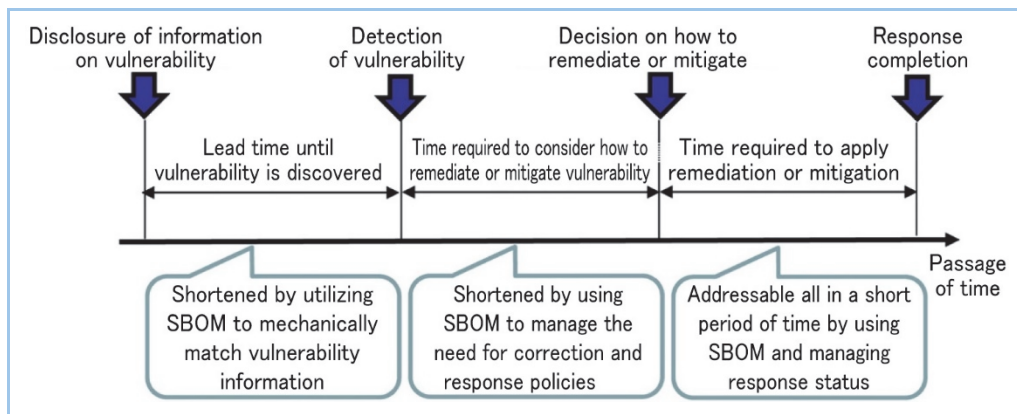


Figure 1 Advantages of using SBOMs to reduce vulnerability response time

3. Overview of vulnerability detection methods utilizing SBOM

The information on the severity of discovered vulnerabilities, the preventive measures, etc., is collected and analyzed by semiconductor makers such as Qualcomm (in reference to vulnerabilities latent in firmware), OS vendors such as Google and RedHat and others on a daily basis. It is then made available in detail to the public as vulnerability databases⁽³⁾. Among many examples, the National Vulnerability Database (NVD) managed by the U.S. National Institute of Standards and Technology⁽⁴⁾, and the GitHub Advisory Database are widely known. In 2023, the NVD received more than 50,000 registrations, totaling over 200,000 entries in the database. When SBOMs are used to detect software vulnerabilities, the software information contained in the SBOM and the software information registered in the vulnerability database are used to check for the presence or absence of vulnerabilities using special software and tools.

4. Utilization of SBOMs for vulnerability management system

In developing a vulnerability management technique utilizing SBOMs, we confirmed that manual registration and operation in a local environment could also occur based on use cases assuming in-house operation, so we summarized the necessary requirements as follows and constructed a vulnerability management system:

1. Able to automate creation/registration of SBOM
2. Able to register SBOM manually
3. Able to import vulnerability information from external vulnerability databases
4. Able to manage response status of detected vulnerability

For system construction, our first task was to select a format for SBOM. The possible format is either SPDX or CycloneDX. It is said that the former is effective in managing licenses, whereas the latter is effective in managing vulnerabilities. In this report, it has been confirmed that the system requirements listed above can be satisfied by combining an SBOM generation tool available at Tool Center of CycloneDX's official website, with the software component analysis tool Dependency-Track.

5. Problems in vulnerability management, our solution and its effectiveness

5.1 Problems about conventional vulnerability management

Conventionally used for vulnerability management are the base scores of the Common Vulnerability Scoring System (CVSS), which signify characteristics intrinsic to vulnerabilities. Being quantitatively assessed, the CVSS base score is a numerical representation of the severity of a vulnerability on a scale of 0.0-10.0, which is based on the attributes inherent in the software or system. Further classified into 5 severity levels according to the base score (**Table 1**). Vulnerability management involves deciding how to respond to vulnerabilities based on the severity levels and prioritizing action. For example, prompt responses are required for the levels of “High” or above.

However, the decision should not depend solely on CVSS base scores, as this framework of prioritization has the following problems.

- The CVSS base score is often evaluated by the developer of the target software, but the results tend to be higher, so early action is required for many vulnerabilities. Such a management system lacks practicality because it is even required to respond to vulnerabilities that do not need to be addressed early enough to halt the application, depending on the mode of operation.
- The CVSS base scores reflect only the results of evaluation from a technical point of view. The implementation environment is not taken into account (e.g., with or without a direct connection to the internet).

In this report, these problems are handled by adopting decision trees for prioritization, in addition to the uniform ratings based on the severity of vulnerability. The judgement conditions in decision tree analysis are based on the information on how to assess the likelihood of being attacked and the information on the likely impact of the attack according to a given operating configuration. Then the effectiveness of this prioritization framework was evaluated.

Table 1 CVSS base scores and definitions of severity levels

Severity level	Base score	Description
Critical	9.0 - 10.0	The impact of the vulnerability on the system is critical, requiring immediate response.
High	7.0 - 8.9	The impact of the vulnerability on the system is high from the perspectives of exploitability, confidentiality and integrity.
Medium	4.0 - 6.9	The vulnerability may affect the system, but the probability is not so high.
Low	0.1 - 3.9	The impact of the vulnerability on the system is limited.
None	0.0	There is hardly any impact of the vulnerability on the system.

5.2 Application of prioritization framework using decision tree analysis

As many of the products of Mitsubishi Heavy Industries, Ltd. are designed for continuous operation, such as thermal power plants, the selection is needed for software updates that entail system shutdown. When it comes to shutting down the system, the schedule also needs to be aligned with other relevant parties. In the decision tree analysis, therefore, four aspects of judgement (i.e., Exploitation, Exposure, Utility, and Well-being and Mission Impact) with explainable criteria have been determined, leading to the setting up of the following four stages in line with the order of priority⁽⁵⁾.

- Immediate: Emergency response
Act immediately, which includes, if necessary, temporary suspension of the application. The probability of being attacked is high.
- Out-of-Cycle: Unplanned response
Act more quickly than usual, although an immediate response is not needed.
- Scheduled: Planned response
Act on occasions such as performing regular maintenance, as the imminent threat of being attacked is low.
- Defer: Watchful waiting
Observe for a while, because the risk is tolerable (e.g., the potential impact of the attack is low).

An explanation of decision factors is described in **Table 2**. **Table 3** gives a list of information used as reference for the criteria in this regard.

Table 2 Explanation of decision factors in decision tree analysis

Decision Factors	Description
Exploitation	Vulnerability exploitation status at this time
Exposure	Assess the extent to which the application under attack is exposed to the outside world
Utility	Usefulness of the vulnerability to the adversary
Well-being and Mission Impact	Assessment of the impact of the attack

Table 3 Sources of information on criteria for each aspect of judgement

Information	Description	Source	Linked aspect of decision factors
KEV catalog	A list of known exploited vulnerabilities	CISA	Exploitation
EPSS score	A probability of exploitation activity against a vulnerability of concern in the next 30 days	FIRST	Exploitation
Application implementation environment (with or without a direct connection to the internet)	Presence or absence of a direct connection to the internet in the server on which an application of concern runs	In-house system operation division	Exposure
CVSS vector string	Information on vulnerability characteristics	NIST	Utility Well-being and Mission Impact

5.3 Evaluation of prioritization framework using decision tree analysis

To verify the effectiveness of decision tree analysis, approximately 7,000 vulnerabilities, which were disclosed between April and June 2023, were used to compare the following two prioritization frameworks: the CVSS base score (i.e., severity of vulnerability) and the decision tree. The results are shown in **Table 4**.

Table 4 Verification results of prioritization framework using decision tree analysis (with direct connection to internet)

		Severity level (CVSS)				Total
		Critical	High	Medium	Low	
Decision tree analysis	Immediate	14 (0.2%)	4 (0.1%)	0 (0.0%)	0 (0.0%)	18 (0.3%)
	Out-of-Cycle	80 (1.1%)	40 (0.6%)	4 (0.1%)	0 (0.0%)	123 (1.8%)
	Scheduled	925 (13.0%)	2,619 (36.7%)	1,392 (19.4%)	2 (0.0%)	4,939 (69.1%)
	Defer	0 (0.0%)	13 (0.2%)	1,944 (27.2%)	100 (1.4%)	2,057 (28.8%)
	Total	1,019 (14.3%)	2,676 (37.5%)	3,340 (46.8%)	102 (1.4%)	7,137 (100.0%)

The “Critical” and “High” severity levels require action, because there is a concern about the potential impact on the system. Of all the vulnerabilities, those assigned to these two levels of severity account for 51%, which confirms the tendency toward overestimation of early response as mentioned in Section 5.1. On the other hand, the counterpart in the decision tree analysis (i.e., “Immediate” and “Out-of-Cycle”) makes up 2.1%. These identified vulnerabilities are characteristically either ones that may cause long-term disruption of critical functions or continuously exploitable ones that may lead to disruption of functions. It has thus been confirmed that this prioritization framework using decision trees identifies vulnerabilities that should be promptly taken care of. The vulnerabilities assigned to “Scheduled” were also examined to see if there were any vulnerabilities whose priorities should have been considered higher; a vulnerability alerted in the KEV catalog named “CVE-2023-20867” was included therein. This vulnerability was regarded as needing no early response because of the complexity of the attack and the degree of the impact on the system. However, if the KEV catalog is to be weighted, reviewing is required. Meanwhile, there were four vulnerabilities that were identified as “Out-of-Cycle” in the decision tree analysis, despite them being assigned to “Medium” according to the severity level. Although their severity levels were judged to be low owing to the complexity of the attack required to exploit them, the fact of being exploited or having a high possibility of being exploited was reflected by applying the decision factor for Exploitation and Utility. In this way, we succeeded in estimating

their priorities as needing an early response. Therefore, it can also be said that the prioritization framework using decision trees is effective in detecting the vulnerabilities in need of action.

The results in Table 4 pertain to the case in which there is a direct connection to the internet. For comparison, the results with no internet connection are given in Table 5. The differences between Table 4 and Table 5 indicate how the presence/absence of an internet connection can affect the classification by decision trees. It has thus been confirmed that the problems with the CVSS base score-based prioritization, which were mentioned in Section 5.1, can also be resolved.

Table 5 Verification results of prioritization framework using decision tree analysis (without direct connection to internet)

		Severity level (CVSS)				Total
		Critical	High	Medium	Low	
Decision tree analysis	Immediate	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Out-of-Cycle	15 (0.2%)	15 (0.2%)	2 (0.0%)	0 (0.0%)	32 (0.4%)
	Scheduled	1,004 (14.1%)	2,648 (37.1%)	1,394 (19.5%)	2 (0.0%)	5,048 (70.7%)
	Defer	0 (0.0%)	13 (0.2%)	1,944 (27.2%)	100 (1.4%)	2,057 (28.8%)
Total		1,019 (14.3%)	2,676 (37.5%)	3,340 (46.8%)	102 (1.4%)	7,137 (100.0%)

6. Conclusion

This report pertains to vulnerability management with SBOMs. Specifically, it involves using SBOMs to detect software vulnerabilities and using decision tree analysis to stratify the detected vulnerabilities. In decision tree analysis, vulnerabilities can be prioritized based on the assessment criteria we have set up, and extract vulnerabilities that require early response from among the more than 50,000 vulnerabilities registered annually. The CRA prescribes that vulnerabilities should be reported within 24 hours (Article 14). Our vulnerability management method in which SBOMs and decision trees are used in a combined manner can be of help in handling the matter in accordance with laws and regulations.

Vulnerability management with SBOMs is currently in trial use with a view to practical application to multiple products.

References

- (1) European Parliament, Cyber Resilience Act, https://www.europarl.europa.eu/doceo/document/TA-9-2024-0130_EN.pdf, (2024-3)
- (2) Ministry of Economy, Trade and Industry, Introduction to Software Bill of Materials (SBOM) for Software Management Ver. 1.0, (in Japanese) <https://www.meti.go.jp/press/2023/07/20230728004/20230728004-1-2.pdf>, (2023-7)
- (3) Information-technology Promotion Agency, Japan, Vulnerability Countermeasure Information Database JVN iPedia, (in Japanese) <https://www.ipa.go.jp/security/reports/vuln/jvn/ipedia2024q1.html>
- (4) NIST, NATIONAL VULNERABILITY DATABASE, <https://nvd.nist.gov/>
- (5) Spring, J. et al., PRIORITIZING VULNERABILITY RESPONSE: A STAKEHOLDER-SPECIFIC VULNERABILITY CATEGORIZATION (VERSION 2.0), https://insights.sei.cmu.edu/documents/606/2021_019_001_653461.pdf, (2021-4)