Poster: HatBOM, An Automated Analysis Platform for SBOM and VEX Generation

Kyeongseok Yang[1], Sungyu Kim[2], Jeongwoo Lee[1], Yoongjong Na[1], Duyeong Kim[1], and Heejo Lee[1]

[1] Korea University, Seoul, Republic of Korea
[2] Center for Software Security & Assurance (CSSA), Seoul, Republic of Korea

Abstract. As the software ecosystem continues to grow in complexity, software supply chain attacks have emerged as a critical concern. To mitigate these threats, Software Bill of Materials (SBOM) and Vulnerability Exploitability eXchange (VEX) documents have gained increasing attention as essential tools for enhancing transparency and security. Despite their growing importance across industries, challenges remain—most notably, the lack of centralized platforms and the absence of precise, automated generation techniques.
In this work, we introduce HatBOM, a platform designed to generate accurate SBOM and VEX documents. HatBOM integrates state-of-the-art analysis techniques to provide high-fidelity results and is designed with usability in mind, enabling even non-experts to perform comprehensive software composition and vulnerability analysis with ease.

Keywords: SBOM · VEX · Verification.

1 Introduction

As the complexity of modern software ecosystems increases, vulnerabilities related to supply chain management have become a critical concern. One notable example is the remote code execution vulnerability discovered in the widely used Java logging library Log4j (CVE-2021-44228). Although the vulnerability impacted a wide range of Java-based web services, many developers faced difficulties in identifying whether their applications had a dependency on Log4j. This challenge led to delayed responses and subsequently resulted in secondary damages [5].

In response to such global supply chain security incidents, the Biden administration issued Executive Order 14028 in 2021 [6], mandating that all software products procured by the U.S. federal government include a Software Bill of Materials (SBOM) to improve national cybersecurity. In addition to SBOM, active discussions have emerged around the Vulnerability Exploitability eXchange (VEX), a complementary standard that provides contextual information on whether known vulnerabilities are exploitable in a specific software environment. As of 2025, regulatory and legislative initiatives concerning SBOM and

VEX are actively progressing across key sectors—including healthcare, automotive, and IoT—in the United States, Europe, and Japan.

Due to increasing software supply chain risks, the importance of SBOM and VEX is increasing. However, the absence of an integrated platform necessitates significant manual effort, making the generation of these documents particularly challenging.

To address these challenges, we propose HatBOM, an integrated analysis platform for SBOM and VEX generation. HatBOM leverages and consolidates existing open-source analysis techniques to deliver high-accuracy results. Furthermore, it facilitates automated and user-friendly document generation, thereby contributing to the mitigation of software supply chain risks.

1.1 Principal Strategy

HatBOM aims to provide an easy-to-use platform for automatically generating SBOM and VEX documents. To achieve this goal, we adopt the following three core strategies:

Building a web-based platform To facilitate broader adoption of SBOM and VEX generation, HatBOM focuses on accessibility. Existing tools often require users to build and configure commercial or open-source software environments manually, posing significant usability challenges. To address this, we designed a streamlined analysis process that supports SBOM and VEX generation through a simple drag-and-drop interface, which is publicly accessible via the web.

Conducting international joint research Achieving scalable and accurate generation of SBOM and VEX documents requires extensive research efforts. So, we have collaborated with leading research groups from various institutions, including Korea University, KAIST, Georgia Institute of Technology, Northeastern University, and ETH Zurich, which possess deep expertise in open-source analysis, vulnerability assessment, and software verification. Furthermore, we worked with the Center for Software Security and Assurance[3] (CSSA) to build an accessible and user-friendly platform.

Community connection To contribute to the broader open-source ecosystem, we are in ongoing discussions with the Open Source Security Foundation (OpenSSF) regarding the integration and dissemination of our technologies. In addition, we are engaging with communities in the medical software to explore potential improvements to the HatBOM platform.

2 HatBOM

The following figure 1 presents an overview of the HatBOM platform. After receiving the source code of the target project, the system performs a three-step process: SBOM generation, vulnerability analysis, and VEX generation.

[3] Led by professor Heejo Lee, located in Korea University, available at http://cssa.korea.ac.kr

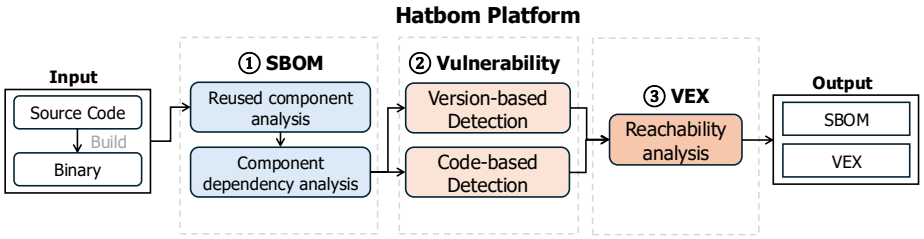


Fig. 1. High-level overview of HatBOM.

2.1 SBOM Generation

To generate an SBOM document, it is essential to identify the components used and analyze the dependencies among them. HatBOM achieves this through a two-step process: Reused Component Analysis and Component Dependency Analysis.

Reused component analysis In this step, we detect reused components within the input source code project by performing code clone detection, leveraging a pre-collected and preprocessed database. This process is implemented using CENTRIS [7], a technique designed to accurately identify reused components.

Component dependency analysis In this step, we analyze the dependencies among the identified components. This information is essential for SBOM document. For this purpose, HatBOM utilizes CNEPS [3], a technique that infers component-level dependencies from module-level interactions.

2.2 Vulnerability Analysis

Based on the generated SBOM document, we perform vulnerability detection. Currently, our platform provides two primary methods for identifying vulnerabilities:

Version-based detection This method detects vulnerabilities by comparing version information listed in the generated SBOM with known vulnerable versions from the National Vulnerability Database (NVD) [4]. Each CVE entry in the NVD includes affected version ranges, vendor, and product information. HatBOM stores this CVE data in a structured database and analyzes vulnerabilities by matching it against the SBOM content.

Source Code-based detection This method detects vulnerabilities using the source code. We apply a code clone detection technique designed to identify vulnerable code patterns. This approach is optimized for large-scale codebases and achieves high accuracy [2]. Our vulnerability database is built on xVDB, a

technique for discovering vulnerable code snippets across diverse sources [1]. Currently, HatBOM supports detection of vulnerabilities related to 39,622 unique CVEs and 334,498 vulnerable functions.

2.3 VEX Generation

The vulnerabilities identified in the previous step are output in the VEX (Vulnerability Exploitability eXchange) document format. To assign higher confidence scores to vulnerabilities that are likely to be exploitable, we perform verification process.

This process involves generating function call graphs for each module of the target open-source software. To ensure accurate and efficient graph construction, we resolve indirect call targets using type analysis. Based on the generated call graphs, we verify whether there are feasible paths from entry points to vulnerable functions. The detected vulnerabilities are classified for VEX generation as follows: if they are reachable from the function entry point, they are labeled as "Affected"; if they are not reachable, they are labeled as "Not Affected"; and if the vulnerable function does not exist, they are labeled as "Not Exist".

3 Result and Futurework

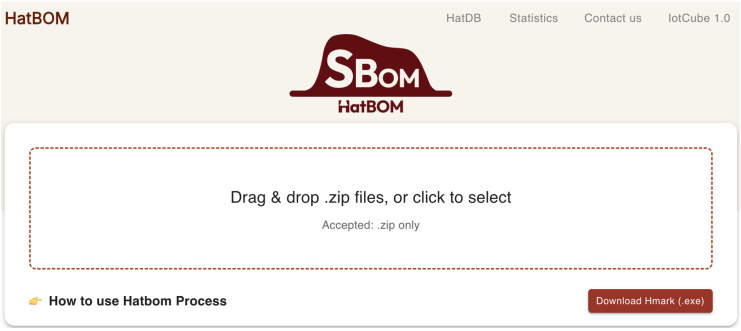


Fig. 2. The main page of HatBOM.

HatBOM was released to the public on August 28, 2025 via web platform [4] (figure 2).

We also plan to extend our system to support SBOM and VEX generation for various input types, such as binaries and containers.

4 Conclusion

HatBOM is an effective solution for SBOM and VEX generation. Users are able to examine a target source code through a few clicks through web platform, and get a reliable analysis result.

[4] https://iotqv.korea.ac.kr/

Acknowledgments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00440780, Development of Automated SBOM and VEX Verification Technologies for Securing Software Supply Chains), (No.RS-2022-II220277, Development of SBOM Technologies for Securing Software Supply Chains). In addition, this work was supported by ICT Creative Consilience Program through the Institute of Information Communications Technology Planning Evaluation(IITP) grant funded by the Korea government (MSIT) (IITP-2025-RS-2020-II201819)

References

1. Hong, H., Woo, S., Choi, E., Choi, J., Lee, H.: xvdb: A high-coverage approach for constructing a vulnerability database. IEEE Access **10**, 85050–85063 (2022)
2. Kim, S., Woo, S., Lee, H., Oh, H.: Vuddy: A scalable approach for vulnerable code clone discovery. In: 2017 IEEE symposium on security and privacy (SP). pp. 595–614. IEEE (2017)
3. Na, Y., Woo, S., Lee, J., Lee, H.: Cneps: A precise approach for examining dependencies among third-party c/c++ open-source components. In: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. pp. 1–12 (2024)
4. National Institute of Standards and Technology: National Vulnerability Database (NVD). https://nvd.nist.gov (2025), last accessed: 2025-07-25
5. Snyk Security Team: Log4j vulnerability and software supply chain security (log4shell) (2021), https://snyk.io/blog/log4j-vulnerability-software-supply-chain-security-log4shell/, accessed: 2025-07-25
6. The White House: Improving the nation's cybersecurity, executive order 14028 (2021), https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity, 86 Fed. Reg. 26633
7. Woo, S., Park, S., Kim, S., Lee, H., Oh, H.: Centris: A precise and scalable approach for identifying modified open-source software reuse. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). pp. 860–872. IEEE (2021)