Poster: Usability, Acceptability, and Efficacy of SBOM Visualization and AI Assistant

Xinyao Ma, Ankith Veldandi, Zitao Zhang, Peter Caven, Ambarish Gurjar, L. Jean Camp
Indiana University Bloomington
(maxiny, aveldan, zhangzit, pcaven, agurjar, ljcamp)@iu.edu

***Abstract*—The Software Bill of Materials (SBOM) has emerged as a critical tool for mitigating information asymmetry in the software security market. By promoting transparency across the software supply chain, SBOMs provide stakeholders with essential information to support decision-making throughout a product's lifecycle. This transparency has the potential to systematically improve awareness of vulnerabilities and risks in software development. To make SBOM content more accessible and actionable, multiple visualization tools have been developed. We conducted quantitative and qualitative experiments to evaluate the usability, acceptability, and efficacy of popular SBOM visualization tools. Our findings reveal that while visualizations are helpful, they often fall short in scalability and risk communication, particularly for complex dependency graphs. Motivated by these limitations, we propose BomBot, an LLM-based AI assistant, designed to help developers and stakeholders better understand and interact with software supply chain information.**

I. Introduction

The SBOM is a critical tool in software development, providing a nested inventory of the components that make up a software product [1]. SBOMs are essential for identifying vulnerabilities and ensuring software security [2]. However, the complexity of SBOM data often makes it challenging for developers and stakeholders to interpret and act upon the information effectively.

We aim to develop an AI-powered assistant, BomBot, designed to address these challenges by providing visualization, understanding, vulnerability identification, and suggestions capabilities for SBOMs. BomBot leverages Large Language Models (LLMs) to interact with users, answer questions, and provide actionable insights. The motivation behind BomBot stems from the limitations of current SBOM visualization tools, which, while useful, often fall short in terms of usability, accuracy, and scalability, especially when dealing with large and complex dependency graphs.

In our previous research experiment, we compare the straightforward provision of SBOM data with two popular open-source visualization tools, ItDepends and DeepBits, with the traditional JSON text files. Our results illustrate the need for more effective communication within the current security market. While SBOMs may present an effective and viable option, their current instantiation is not suitable for most consumers. This paper seeks to understand the implications of information conveyance of complex code. Specifically, we focus on how visualizing machine-readable SBOMs can be used to increase transparency and mitigate the vulnerability of security and privacy. The results indicate that simply visualizing the interdependencies within nested code creates a 144.94% increase in a more accurate selection and a 162.78% increase in risk evaluation of vulnerabilities.

II. Methodology

The methodology for developing and evaluating BomBot was structured into three main phases: visualization evaluation, chatbot development, and user interaction design. Each phase was designed to address specific challenges in SBOM analysis and vulnerability identification.

A. Visualization Evaluation

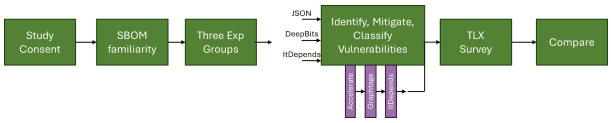

Fig. 1. Experiment design.

Two visualization tools, It-Depends and DeepBits, were chosen for their ability to generate SBOM visualizations from .json files and clear and easy-to-understand structures with nodes connected with lines. The evaluation was conducted using a controlled experiment where participants were asked to interact with SBOM visualizations of varying complexity. The visualizations were assessed based on their usability, acceptability, and accuracy in identifying vulnerabilities and providing mitigation strategies. Usability was measured using the NASA Task Load Index (TLX), which evaluates mental demand, physical demand, temporal demand, performance, effort, and frustration. Acceptability and accuracy were measured through user surveys and task completion rates.

B. Chatbot Development

BomBot was developed as a Retrieval-Augmented Generation (RAG) Bot, leveraging LLMs to answer questions about SBOMs. The chatbot integrates with the Open Source Vulnerability (OSV) database to identify and categorize vulnerabilities by risk level (low, medium, high). The system provides real-time updates on vulnerabilities; however, mitigation information currently requires manual input, as zero-shot training for this task does not consistently yield accurate results.

C. Plan: User Interaction Design

Participants were given tasks to identify vulnerabilities in SBOMs using both visualization tools and the chatbot. The tasks were designed to simulate real-world scenarios where developers need to assess the security of software components.

III. Evaluation

The evaluation of the SBOM visualization tools was conducted through a combination of quantitative and qualitative measures. The evaluation of BomBot is planned to be conducted as an online AI-assisted vulnerability identify and mitigate human subject experiment, from experts to novices.

A. Visualization Efficacy

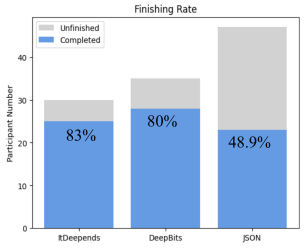

Fig. 2. Acceptability Results: finishing rates for three groups.

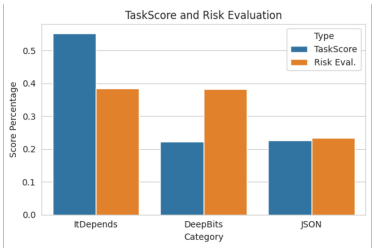

Fig. 3. Accuracy Results: task scores for experiment tasks and risk evaluation of vulnerability packages for three groups.

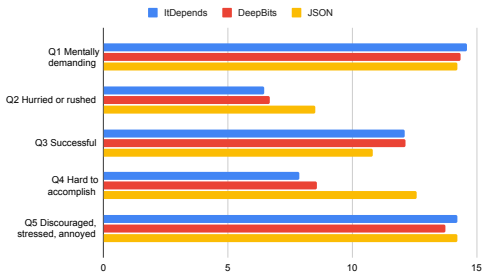

Fig. 4. Usability as Measured by the NASA Task Load Index.

The NASA TLX results indicated that while visualizations were useful, they were also mentally demanding and could be stressful for users. Participants reported high levels of mental effort (Q1) and frustration (Q5) when working on these tasks. Survey results in Fig. 4 showed that 80% to 83% of participants in visualization groups finished all tasks, but only 48.9% in the JSON group finished in Fig. 2. This discrepancy highlights the utility of SBOM visualizations. Participants also identified vulnerabilities with a high degree of accuracy using the visualizations, see Fig. 3. However, the task was perceived as burdensome, even for small packages, and likely infeasible for graphs with thousands of nodes.

B. Future Plan: BomBot Performance

We expect the BomBot to be able to achieve high accuracy in identifying and categorizing vulnerabilities. Participants could query the BomBot for specific vulnerabilities (e.g., in the Jinja2 package) and receive accurate risk assessments (e.g., high-risk vs. low-risk). In the real-world experiment, we expect participants to report higher satisfaction with the BomBot than with traditional visualization tools. The interactive nature of the BomBot allowed users to ask questions and receive immediate feedback, reducing the cognitive load associated with interpreting complex SBOM data. The bot will then compare the new SBOM with the original, analyze the changes, and provide feedback on how those updates have impacted the software's security.

IV. Conclusion

While visualizations were helpful, they were not sufficient for understanding complex SBOM data, especially for large-scale projects. The chatbot, on the other hand, offered a more interactive and scalable solution capable of addressing concerns and answering questions regardless of the complexity of the SBOM. We plan to integrate Microsoft Copilot suggestions with SBOM generation to create a seamless advisory system that provides real-time recommendations during the software development process. Future work will focus on automating risk mitigation information using models like BAAI/bg-m3 and BERT to provide more accurate and consistent mitigation strategies. The team aims to expand BomBot's capabilities to include automated compliance verification, ensuring that software meets regulatory and security standards. Additionally, BomBot will collaborate with GUAC (Graph for Understanding Artifact Composition) for future visualization evaluations and API integration, enhancing the chatbot's ability to process and visualize complex SBOM data.

Acknowledgment

This work was funded by the U.S. Department of Homeland Security under Grant (Award 17STQAC00001-07-00), US Department of Defense (Contract W52P1J2093009), and funding from CTIA. The views and conclusions contained in this poster are those of the authors and should not be interpreted as representing the official policy or opinions expressed or implied by any funder.

References

[1] National Telecommunications and Information Administrator. Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM). 2020.

[2] National Telecommunications and Information Administration. Software Suppliers Playbook: SBOM Production and Provision. 2021. https://www.ntia.doc.gov/report/2021/minimum-elements-software-bill-materials-sbom.