Software Bill of Materials (SBOM) Approach to IoT Security Vulnerability Assessment

James Bonacci and Reese Martin

Abstract

This paper presents a study of the security vulnerabilities surrounding Internet of Things (IoT) devices, and how these vulnerabilities can be detected and analyzed utilizing the Software Bill of Materials (SBOM). This methodology allows a user to gain more information about a device than what was available before using tools such as an automated vulnerability scanner. Compared to the information available from current popular security vulnerability scanners, the information gathered from the SBOM approach allows a user to have far more insight into a device's vulnerabilities and composition. This study emphasizes the importance of the SBOM and how it can be used to assess such security vulnerabilities on a deeper level than automated scanners. In this study, we compare the security vulnerability assessment capabilities of three different methods: NetRise, Tenable OT Security, and the free National Vulnerability Database (NVD) provided by the National Institute of Standards and Technology. NetRise is the method that will be used to demonstrate the capabilities of SBOM security. Tenable OT Security is a traditional vulnerability scanner. The last method used is referencing the NVD. This is the U.S. government repository of vulnerability management data. Limitations and deficiencies of the SBOM approach to security analysis are also addressed throughout the study.

Keywords

Internet of Things (IoT) · Vulnerabilities · Software Bill of Materials (SBOM) · Automated Scanners · Tenable OT Security · NetRise · National Vulnerability Database (NVD) · Assessment · Cybersecurity · Printers

J. Bonacci(✉) · R. Martin
Robert Morris University, Department of Informatics, Moon Township, PA, USA
e-mail: Jxbst976@mail.rmu.edu; Rgmst170@mail.rmu.edu

1 Introduction

The Internet of Things (IoT) as we know it has transformed the world around us. It has changed the way people live in their homes, the way in which people learn, and it has even changed the way that most businesses operate. As it stands in 2023, there are approximately 15 billion IoT devices worldwide. Studies have predicted that by the year 2030, there will be over 29 billion IoT devices connected to the internet worldwide [1]. Currently, there is not sufficient knowledge on detecting the vulnerabilities and threats on these miscellaneous devices. There is however an urgency for gaining insight into these threats. The Mirai Botnet attack occurred with much ease as all it took was a brute force attack on an IoT device. The attack then was able to collect over 400,000 simultaneously connected devices. The botmaster then could send the devices to the target server to perform a DDoS attack. Some of these security concerns are always running devices, poor maintenance, the ability to attack traffic, and minimally interactive user interfaces [2]. Typically, IoT devices are the weakest security asset within networks due to poor security and maintenance. Without security experts being aware of this area of vulnerability creates an opportunity for more attacks using this methodology to occur.

In May 2021, the President of the United States issued an executive order (EO 14028) to improve the nation's cybersecurity. In summary, his order plans to do this by strengthening the security of federal networks, removing barriers to information sharing between public and private sectors on cyber threat issues, and enhancing software supply chain security [3]. The executive order directed the Secretary of

S. Latifi (ed.), *ITNG 2024: 21st International Conference on Information Technology-New Generations*, Advances in Intelligent Systems and Computing 1456,
https://doi.org/10.1007/978-3-031-56599-1_8

Commerce to carry out many tasks. One of these tasks is to work with the National Telecommunications and Information Administration (NTIA) to publish the minimum elements for the Software Bill of Materials (SBOM). The Department of Commerce considered the SBOM to be the glue between external data, such as vulnerability information, and the products in question [4]. This is exactly what our study is demonstrating through the SBOM approach. Using the SBOM to correlate vulnerabilities to the software in each situation is what gives us insight into the risks of our embedded devices.

Software products are often assembled through very complex supply chains. It is for this exact reason that there has been an increase in Software Supply Chain (SSC) attacks. Oftentimes, these attacks are a result of poorly maintained Open-Source Software (OSS) [5]. OSS is usually unsupported since the creator(s) may have lost interest or moved on to another project. This abandoned software does not go through a regular patching cycle like active software development does. Some of the best-in-class code can have up to 600 defects per million lines of code, while average code can have up to 6000 defects per million lines of code [6]. Along with this, research also shows that up to 5% of software defects are made up of security vulnerabilities. This means that the best-in-class code can have up to 30 vulnerabilities per million lines of code, while the average code can have up to 300 vulnerabilities per million lines of code [6]. Here is where the SBOM comes in to play a security role. An SBOM is a machine-readable list of the components used in a software product. These components often consist of software dependencies, libraries, and other software modules used in a product [5]. This software is then compared to known security vulnerabilities, and this knowledge allows security personnel to make decisions and changes as necessary.

The goal of our paper is to raise IoT security awareness and add research to IoT vulnerability assessments. There is currently a gap in the IoT industry when it comes to assessing the vulnerabilities in embedded devices. Popular automated vulnerability assessment tools, such as Tenable OT Security, cannot gather much information about the software that makes up an embedded device. We propose a different solution for assessing vulnerabilities by using the SBOM security approach.

2 Literature Review

There are five different types of SBOMs. These five types are design, source, build, analyzed, and deployed. This list can be followed as stages. First, there is a design phase, which is then followed by a sourcing phase, a building phase, an analysis phase, and a deployment phase [7]. In this study, the SBOM type that is focused on is the analyzed type. This is an SBOM that was generated through an analysis of artifacts. This analysis was done by the software tool NetRise. There are benefits and limitations to the analyzed SBOM type. One of the benefits of this type is that it can provide visibility into the software without an active development environment. Another benefit is the ability it has to find hidden dependencies that may have been missed by other SBOM creation tools. A limitation of this SBOM type is it may be susceptible to errors or approximations if the tool is not able to fully analyze the software components correctly, resulting in inaccurate data [7].

In a study of Automatic Vulnerability Detection in Embedded Devices and Firmware Images, Qasem et al. surveyed many different methods of detecting vulnerabilities in firmware images. One of their recommendations was for firmware developers to check if the software libraries or packages they are using contain any pre-existing threats. Another finding of this survey was to test a device firmware image extensively to determine if there is an underlying vulnerability in the firmware's composition [8]. This technique may be difficult in some instances because it requires an emulation approach to test the firmware outside of its original environment. If the firmware needs access to its peripherals, it makes it even more difficult. Once the device is emulated, it then needs to be analyzed by an advanced binary analysis technique, such as fuzzing or symbolic execution.

Another study that was done with the software FirmHunter focused on the improvement of IoT device detection of vulnerability. They concluded that during the implementation process, the software was limited by the CPU architecture and operating system. This made the use of the software only available to devices that run mipsel, mipseb, and armel CPU architecture. Along with the CPU limitation the only protocol that could be analyzed is the HTTP protocol. Devices that are identified as IoT devices can use more than just the HTTP protocol which would make the software unable to detect vulnerabilities [9]. Addressing limitations within detection software can be challenging because of the robustness that the software needs to be. For example, the FirmHunter software can only be run on Linux so running the application on Windows would not be possible. This is a limitation but allows for the application to function within the operating system it is designed for.

The goal of security vulnerability testing and analysis is to assess the security risks and impacts for effective security risk mitigation and management decisions. The SAFER framework, the Security Assessment Framework for Embedded-device Risks, is a comprehensive approach to assessing security risks of IoT devices using network device identification and automated firmware analysis to predict current and future security risks with considerably high success rates [10]. A major strength of SAFER is its significant scalability to assess the security vulnerability levels of large numbers of

IoT devices on large-scale networks. A limitation of the SAFER approach is that it depends on user access to available firmware images for analysis of embedded software. Some vendors may limit access to firmware, which will limit firmware analysis to vendors or trusted users only.

The financial services industry, energy sector, and healthcare industry are three examples of SBOM security being implemented in the wild. In 2015, the financial services industry noticed a slew of software supply chain vulnerabilities, and they decided to adopt SBOM security practices to gain more insight into their vendors' software components. Similarly, the energy sector put more stringent requirements on their vendors to provide documentation on all of their components of the products. The healthcare industry has also acknowledged the strengths of the SBOM, but implementation has been slow due to a lack of data available from the published peer-reviewed literature [11]. The widespread use and adoption of the SBOMs will allow organizations to have more insight into the software that they are purchasing and using to handle sensitive data. Knowledge of software components allows for additional security actions such as pre-positioned risk mitigation measures, quicker incident response times, and potentially reduced disruption effects.

3 Methodology

This study aims to compare three different methods of assessing vulnerabilities associated with IoT devices. The first of these methods is firmware image analysis through NetRise. NetRise compares the software in the SBOM to a multitude of different public vulnerability databases. The second method used in our research is Tenable OT Security. Tenable OT Security acts as a typical vulnerability scanner that sends packets to a device and receives information from the network card on the device. This information is compared to Tenable's Plugin Database, which gathers vulnerability information from public domains. The last method used in this comparison is referencing the National Vulnerability Database (NVD) provided by the National Institute of Standards in Technology. Using the search feature, we are able to search for products, and the NVD returns information on product versions that have known vulnerabilities.

This test and data collection took place on a secure segment of an enterprise network at a higher education institution. In this environment, we were limited in the availability of devices that would be relevant to this study. The firmware image of the device was collected in May 2023. On this same day, the image was uploaded to NetRise for analysis. In the same secure environment of the network, this device was first detected by the Tenable OT Security software in December 2022. The software ran a wireless scan of the device, and that is how the data was collected for this study. Lastly, the data from the NVD was collected from the free and publicly available vulnerability database provided by the National Institute of Standards and Technology.

For the sake of this study, we were able to gather security vulnerability information on a Xerox AltaLink C8045, running firmware version 103.002.14100. This very same device was examined using the methodologies previously described. The firmware image was uploaded to the NetRise for analysis. NetRise's SBOM and Vulnerability tabs were utilized to capture data. Tenable OT Security detected this device through a network discovery scan. In Tenable OT Security, the Vulnerabilities tab was accessed to view information on the device's vulnerabilities. Lastly, this device was found through the NVD by using the product search function.

Two firmware security tools were used to gather data for this paper: NetRise and Tenable OT Security. In addition to these two tools, we also referenced the National Vulnerability Database to compare known security vulnerabilities related to the product. The NetRise platform utilizes the firmware image of a device to provide a user with the SBOM, giving insight into what components of the firmware are making up the vulnerabilities on a particular device. This gives its user a list of all the software components that make the device function, along with any vulnerabilities that are associated with them. Tenable OT Security is a much more traditional vulnerability scanner that sends packets to a device and captures what network services and applications are running on it. This is then referenced to a plugin database to determine if the device has any vulnerabilities associated with it.

For the purpose of this paper, we have only collected data from NetRise's vulnerability and SBOM tabs for a particular asset. Similarly, in Tenable OT Security, data has only been collected from the vulnerability tab for the particular asset. Although both of these tools have a plethora of features that give users more information about a device, we limited the features used to show what data was most important to the study. Additionally, we have utilized the free National Vulnerability Database, which is provided by the National Institute of Standards and Technology (NIST). This database allows us to search for particular products and then returns vulnerabilities that are related to the product/device.

4 Findings and Discussion

The NetRise platform SBOM dashboard is displayed in Fig. 1. From the dashboard, we can gather several things about the software components that make up the device, and how many security vulnerabilities are associated with each component. After analyzing the firmware image and relating the findings to public vulnerability databases, NetRise found that the log4j component of the device contained six total vulnerabilities. These six vulnerabilities were categorized

NAME	VERSION	VENDOR	TYPE	LICENSE	CORRELATIONS	VULNERABILITIES
log4j	1.2.17	apache	Package	-	0	3 3
jsoup	1.8.2	jsoup	Package	-	0	2 1
httpclient	4.5	apache	Package	-	0	1

Fig. 1 SBOM Software Components (NetRise)

CVE ID	SEVERITY	COMPONENT	VERSION
CVE-2020-13956	MEDIUM	httpclient	4.5
CVE-2022-36033	MEDIUM	jsoup	1.8.2
CVE-2015-6748	MEDIUM	jsoup	1.8.2
CVE-2021-37714	HIGH	jsoup	1.8.2
CVE-2023-26464	HIGH	log4j	1.2.17
CVE-2022-23302	HIGH	log4j	1.2.17
CVE-2022-23307	HIGH	log4j	1.2.17
CVE-2020-9493	CRITICAL	log4j	1.2.17
CVE-2019-17571	CRITICAL	log4j	1.2.17
CVE-2022-23305	CRITICAL	log4j	1.2.17

Fig. 2 CVEs associated with Xerox AltaLink C8045 (NetRise)

by NetRise into three critical, and three high vulnerabilities. There were also three Jsoup security vulnerabilities found in the analysis. One of these vulnerabilities was categorized as high severity, while the other two were labeled as medium severity. The last of the security vulnerabilities detected by NetRise is an HTTP client vulnerability. This was classified as a medium-severity risk. We would like to make a note that NetRise did detect a multitude of software components that make up the SBOM, however, Fig. 1 only shows three. This is because these three components were the only ones to have known security vulnerabilities from public databases. The remaining software components that did not have any publicly known security vulnerabilities were omitted from Fig. 1 to maintain the focus of the study.

Utilizing the NetRise platform to assess the software vulnerabilities in the SBOM. NetRise was able to identify ten different vulnerabilities using several different references from publicly available vulnerability databases. These ten vulnerabilities were grouped by NetRise into three different categories: Critical, High, and Medium. There were a variety of medium vulnerabilities consisting of HttpClient and Jsoup-related vulnerabilities. The high vulnerabilities also included a Jsoup exploit. Jsoup is an open-source Java library for working with HTML. There are several threats of these vulnerabilities, namely potential DOS attacks and Cross-Side-Scripting (XSS) vulnerabilities. Lastly, there are three high and three critical vulnerabilities all relating to Log4j. Log4j is a vulnerability that appeared in late 2019, and it is similar to Jsoup in the sense that it is a Java library. Log4j contains different vulnerabilities that are all affecting this device in particular. Figure 2 displays which vulnerabilities were categorized into which severity ranking as well as the Common Vulnerabilities and Exposures identifier.

Tenable OT Security detected three vulnerabilities through its scan. Tenable OT Security classified these as two “Low” severity and one “High” severity vulnerability. The first of the low vulnerabilities is an SSL 64-bit Block Size Cipher exploit. This vulnerability allows threat actors

Name	S...	Plugin family
SSL 64-bit Block Size Cipher Suites Supported (S...	Low	Generic
SSL RC4 Cipher Suites Supported (Bar Mitzvah)	Low	Generic
Common SNMP Community String Detection	High	SNMP

Fig. 3 Vulnerabilities associated with Xerox Altalink C8405 (Tenable)

to disclose secret text like HTTPS cookies and it could also allow the hijacking of an authentication session [12]. The second of the low vulnerabilities is an SSL RC4 Cipher Suites Supported exploit. This vulnerability could allow an attacker to obtain many ciphertexts, and possibly derive the plaintext from there [12]. The high vulnerability that Tenable discovered is a Common SNMP Community String Detection exploit. This vulnerability could allow an attacker to gain access to the remote SNMP server, and from there change the configuration of the system [12]. Figure 3 displays the vulnerabilities that were captured from Tenable OT Security. The three columns display the name of the vulnerability, the severity (descending), and the plugin family in which the vulnerability is contained.

The third test consisted of consulting the National Vulnerability Database to investigate known vulnerabilities on the Xerox AltaLink C8045. Using the product search feature, we were able to find the product with vulnerabilities pertaining to the firmware version running on our test device. There are three vulnerabilities associated with the Xerox AltaLink C8045 in the NVD. The first vulnerability on the result list is CVE-2021-28669. This vulnerability allows an attacker to potentially configure the device without administrator rights. The second vulnerability, CVE-2021-28668, contains several SQL injection exploit abilities. Lastly, CVE-2021-28670 allows attackers to use the Scan To Mailbox feature to delete files from the disk on the device. The findings of the National Vulnerability Database are shown in Fig. 4.

The data from each of the methods we used show that there is a wide range of known vulnerabilities related to this product. The NetRise platform was able to discover ten different vulnerabilities related to the software components of the SBOM. Tenable OT Security discovered three vulnerabilities through its traditional scan. After referencing the National Vulnerability Database, three vulnerabilities were identified relating to the particular device of this study. Each of these methods gathered different information about the same device. This study shows that there are many different methods of testing devices for security vulnerabilities, but they are not all on the same page. The difference in the data proves that there is still a need for a centralized vulnerability database that is specialized.

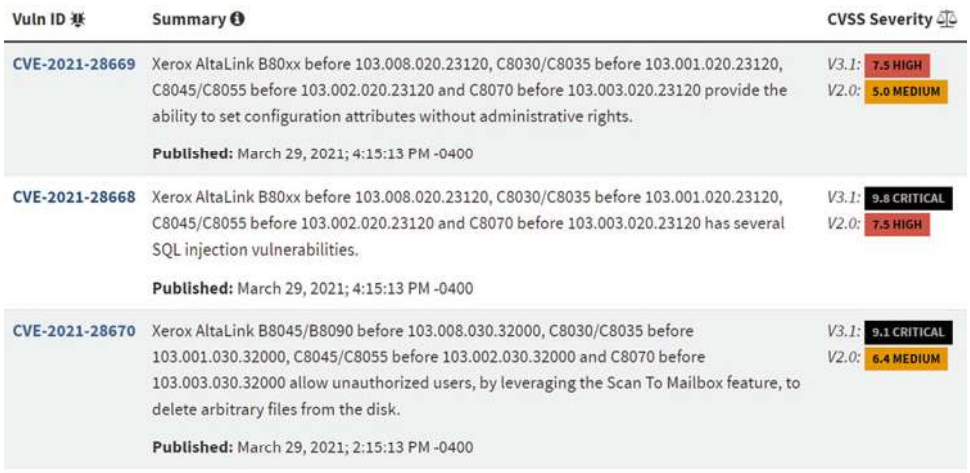
The databases used to reference vulnerabilities of IoT devices are very broadly defined. Currently, there is no widely available IoT dedicated vulnerability database. There are however efforts in place to create such a resource [13]. Having a dedicated IoT vulnerability database would strengthen the robustness of the SBOM.

5 Conclusion

As the number of IoT devices worldwide continues to rapidly increase, there must be more of an attempt to secure these devices. The SBOM is a great vessel to provide insight into software security vulnerabilities that are present in the software that makes IoT devices run. The SBOM provides a form of transparency between the product manufacturers and the users. This can also lead to more trustworthy interactions between the providers and consumers. The SBOM collects more information on a device than most typical vulnerability scanners are capable of. With the addition of an SBOM compilation and analysis tool, analysts can have more knowledge about a device than ever before. There are not many well-known SBOM analysis tools out there, but there are emerging companies like NetRise that are making efforts to fill the void in IoT security. It is projected that by 2022 and 2023, there will be a major increase in SBOM security tools [5].

There are, however, limitations to the SBOM. The major limitation of the SBOM is its availability. The vendor who is making the product needs to be on board to share their SBOMs, otherwise this is not a viable solution. Some software companies do not want to make their SBOMs publicly available [14]. This is where the power of the SBOM immediately comes to a halt. There are ongoing efforts dedicated to enhancing the availability of SBOMs [4]. Additionally, some vulnerabilities in the SBOM may overstate the actual risk that a product contains [14].

There is still more research that needs to be done on this topic before it can become the standard approach to assessing security vulnerabilities in IoT devices. SBOM data needs to be more readily available from vendors. Without this availability, there cannot be much transparency into a product's software composition. There is also a need for



Vuln ID	Summary	CVSS Severity
CVE-2021-28669	Xerox AltaLink B80xx before 103.008.020.23120, C8030/C8035 before 103.001.020.23120, C8045/C8055 before 103.002.020.23120 and C8070 before 103.003.020.23120 provide the ability to set configuration attributes without administrative rights. **Published:** March 29, 2021; 4:15:13 PM -0400	*V3.1:* 7.5 HIGH *V2.0:* 5.0 MEDIUM
CVE-2021-28668	Xerox AltaLink B80xx before 103.008.020.23120, C8030/C8035 before 103.001.020.23120, C8045/C8055 before 103.002.020.23120 and C8070 before 103.003.020.23120 has several SQL injection vulnerabilities. **Published:** March 29, 2021; 4:15:13 PM -0400	*V3.1:* 9.8 CRITICAL *V2.0:* 7.5 HIGH
CVE-2021-28670	Xerox AltaLink B8045/B8090 before 103.008.030.32000, C8030/C8035 before 103.001.030.32000, C8045/C8055 before 103.002.030.32000 and C8070 before 103.003.030.32000 allow unauthorized users, by leveraging the Scan To Mailbox feature, to delete arbitrary files from the disk. **Published:** March 29, 2021; 2:15:13 PM -0400	*V3.1:* 9.1 CRITICAL *V2.0:* 6.4 MEDIUM

Fig. 4 Vulnerabilities associated with Xerox AltaLink C8045 (NVD)

tool integrations that can help an organization consume the information from the SBOM in a way that allows for actionable responses to security vulnerabilities. A specialized database of IoT devices and vulnerabilities would also be a tremendous advancement in learning more about the risks of the IoT. Lastly, SBOM investigations for security vulnerabilities may lead to overstated risks. This leaves room for further research on how this happens and how this can be reduced.

Acknowledgments We acknowledge the advice on this research received from Dr. Ping Wang and Jim Mahony. This research is funded by NSF SFS grant # 2234554 awarded to Robert Morris University.

References

1. *Number of Internet of Things (IoT) Connected Devices Worldwide from 2019 to 2023, with Forecasts from 2022 to 2030* (Statista). https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/. Accessed 11 Oct 2023
2. C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, DDoS in the IoT: Mirai and other botnets. Computer **50**(7), 80–84 (2017). https://doi.org/10.1109/MC.2017.201
3. J. Biden, *Executive Order on Improving the Nation's Cybersecurity* (Whitehouse). https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/. Accessed 23 Oct 2023
4. *The Minimum Elements for a Software Bill of Materials (SBOM)* (U.S. Department of Commerce, USA, 2021)
5. B. Xia, et al., An empirical study on software bill of materials: Where we stand and the road ahead (2023). https://doi.org/10.1109/ICSE48619.2023.00219
6. C.M. Wallen, C.J. Alberts, M.S. Bandor, C. Woody, *Software Bill of Materials Framework: Leveraging SBOMS for Risk Reduction* (Software Engineering Institute, 2023)
7. Cybersecurity Infrastructure and Security Agency, *Types of Software Bill of Material (SBOM) Documents*. https://www.cisa.gov/sites/default/files/2023-04/sbom-types-document-508c.pdf. Accessed 14 Nov 2023
8. A. Qasem, P. Shirani, M. Debbabi, L. Wang, B. Lebel, B.L. Agba, Automatic vulnerability detection in embedded devices and firmware: Survey and layered taxonomies. ACM Comput. Surv. **54**(2), 1–42 (2022). https://doi.org/10.1145/3432893
9. Q. Yin, X. Zhou, H. Zhang, FirmHunter: State-aware and introspection-driven grey-box fuzzing towards IoT firmware. Appl. Sci. **11**(19), 9094 (2021). https://doi.org/10.3390/app11199094
10. P. Oser, R.W. van der Heijden, S. Lüders, F. Kargl, Risk prediction of IoT devices based on vulnerability analysis. ACM Trans. Priv. Secur. **25**(2), 1–36 (2022). https://doi.org/10.1145/3510360
11. S. Carmody et al., Building resilient medical technology supply chains with a software bill of materials. NPJ Digit. Med. **4**(34) (2021). https://doi.org/10.1038/s41746-021-00403-w
12. "Plugins", *Tenable*. https://www.tenable.com/plugins. Accessed 25 Oct 2023
13. M. Janiszewski, M. Rytel, P. Lewandowski, H. Romanowski, Creating vulnerabilities and exploits database of IoT devices, in *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference (EICC '22)*, (Association for Computing Machinery, New York), pp. 91–92. https://doi.org/10.1145/3528580.3532990
14. U.S. Department of Defense, *Securing the Software Supply Chain: Recommended Practices for Software Bill of Materials Consumption* (National Security Agency, USA, 2023)