Introduction

The members of the Cybersecurity Coalition\(^1\) represent several of the largest and most critical producers of software, software assurance, and security services for both the private sector and the United States federal government departments and agencies (USG). As such, the Coalition has a long-standing interest in standards and requirements that can impact how software gets developed, delivered, and maintained.

This paper focuses on the current state of Software Bill of Materials (SBOM). The Coalition wants to stress that SBOM represents only one part of a secure software development strategy and lifecycle. Indeed, an SBOM is a tool (one that is very much a work in progress, as detailed below) in promoting good cyber hygiene. It is not a silver bullet, and certainly not a substitute for the comprehensive approach laid out in Executive Order 14028: Improving the Nation’s Cybersecurity\(^2\) (aka “Cyber EO”) to make mission-critical software (regardless of its proprietary or open-source development model) trustworthy and resilient. Without a focus on open and transparent software management over the life cycle of a software product, as envisioned by the Cyber EO, little progress will be made.

The promise of being able to streamline the vulnerability identification and mitigation process is certainly appealing. An accurate SBOM can play an important role in achieving that goal, by enabling an understanding of the inventory of the used software components. Indeed, component inventories, which are an essential first step to producing an SBOM, can support multiple assurance processes. However, these goals can be achieved most effectively if grounded in a better understanding of the advantages and limitations of SBOMs today, how they fit into essential software lifecycle management and cybersecurity risk management practices, and where SBOMs need to continue to evolve. All of this requires further work before SBOMs can be effectively utilized for procurement, security, or any other use case.

Below, we highlight the key areas that we believe are in need of improvement before procurement requirements can be placed on software producers and before agencies can make effective use of SBOMs in their environments. To address these concerns, we recommend the following:

\(^1\) https://www.cybersecuritycoalition.org/
\(^2\) https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/
● Establish pilot programs involving software suppliers and agencies to demonstrate the effectiveness of SBOMs in improving vulnerability management practices, based on risk metrics, more rapidly and with less effort than existing tools and processes.
● Drive to common standards for sharing, processing, and implementation of SBOMs and infrastructure to reduce potential confusion and inconsistency in outcomes.
● Perform additional research and develop pilot programs to better refine how SBOMs can and should be used in cloud environments.
● Create public/private workshops to discuss both the technical and non-technical aspects of SBOM sharing, including establishment of criteria for ensuring confidentiality where desired, and avoiding liability for software suppliers.

Creation, Sharing, and Consumption

Given that SBOMs are a relatively new and often misunderstood topic, they are not being produced routinely by many software producers, and changing that will require time and effort. As such, the Coalition believes that the U.S. federal government should provide clarity on what software (either by type or specific applications) they are prioritizing so that producers of that software can prioritize their efforts to develop SBOMs with the correct data elements, and ensure that sharing mechanisms are in place. While the National Institute of Standards and Technology (NIST) provided guidance by defining EO-critical software, NIST also noted that the Cybersecurity and Infrastructure Security Agency (CISA) would provide more specifics. As of this writing, these specifics have not been made publicly available. Without this prioritization, efforts to implement the use of SBOM could be hindered by development resources spread too thin and agencies left expecting too much from software producers, or lacking the capabilities to use SBOMs effectively for mission critical applications. Any disconnect on either end of that transaction will result in confusion and disappointment.

Additionally, it is important to recognize that, in the near term, there will be inconsistencies related to software produced before the Cyber EO versus software produced after. This should be accounted for in any requirements that are put into place and ensure that agencies understand the issue well enough to discuss with software producers.

We also note that at this time, it remains unclear how departments and agencies will receive and use SBOMs once they are shared with them. The government has thousands of trained and dedicated Information System Security Officers who already have more information than they can effectively manage in a timely manner. The long-standing processes they follow to identify and mitigate vulnerabilities have been in place largely since the introduction of the Federal Information Security Modernization Act (FISMA), and we are concerned that simply adding SBOMs to the mix, without carefully demonstrating how it will streamline and improve those processes, will result in resistance and failed implementation.

It is necessary to recognize that as government and industry work collectively to mature SBOMs, the ecosystem will remain imperfect and that some protections should be afforded in acknowledgement of this. The nature of those protections will require ongoing discussion as SBOM matures, but our view is that SBOMs are intended to improve communication and
facilitate risk assessment and response. In doing so, they can help foster transparency and trust. The creation, sharing, and consumption of SBOMs must be well understood in the context of software development, lifecycle management, and security practices.

Additionally, overly stringent requirements related to the production and dissemination of SBOMs will require that resources be diverted from other activities, which for many organizations will include meeting other requirements, such as those related to secure software development. The impact of this will vary from one software producer to another, but should be carefully considered so that we don’t take one step forward with SBOM at the expense of slowing or stalling other efforts essential to security and the development of secure software.

The Coalition encourages the federal government to seek additional input and collaboration with software producers to continue capturing best practices and ensure that the use of SBOM across departments and agencies is successful. We recognize that CISA is planning working groups designed to help accomplish this goal, and the Coalition and its members look forward to participating.

Automation

The amount of software in most organizations, combined with how often it gets updated or replaced, creates a dynamic environment that is far greater than humans can process manually.

For software consumers, introducing SBOMs, which track the tens, hundreds, or thousands of components in each piece of software an organization manages, without improving automation will only make this problem worse. For SBOMs to be useful to consumers, technology and automated processes need to be standardized, validated, and in place to receive and/or review SBOMs, integrate their information into existing vulnerability and asset management systems, and enable those systems to interact with SBOMs quickly and efficiently. Robust SBOM data integrity and authenticity solutions (e.g., signatures) should be developed and adopted for organizations to automatically validate and trust the contents of SBOMs they receive.

For software producers, a lack of automation could be especially problematic for complex systems that have dependencies on multiple open source and commercial components. For example, most software producers are somewhere in the middle of the supply chain and automation is required to combine upstream SBOMs with other providers with their own. That same process could occur many times, by various software producers before a final product and associated SBOM can be delivered. In order for consumers to make effective use of SBOMs to mitigate security risks, producers must be able to generate accurate, trustworthy SBOMs as a natural byproduct of the software build and maintenance processes. This requires automation to integrate SBOM creation and signing into many software programming languages, build tools and package managers.

While good progress is being made, the necessary technology and processes are not in place at scale today, particularly between organizations (both within the supply chain and between vendors and consumers of software) where sharing of SBOMs is necessary for them to be
effective. The utility of SBOMs is impacted by the SBOM consumer’s ability to receive them, validate their integrity, organize them, associate the correct SBOM with the installed product version, and conduct queries as needed automatically. There is currently a lack of standardized SBOM adoption which makes sharing difficult. This is especially difficult for certain environments in which the underlying software builds and dependencies are rapidly evolving, and where development lifecycles are happening faster.

The Coalition recognizes that this is a classic “chicken and egg” problem. We believe that a series of pilots and test cases could be beneficial in encouraging the creation, sharing, and consumption of SBOMs, as well as create a better understanding of the automation that is necessary to use them effectively.

**Challenges in Cloud Environments**

Software is continuously being altered during its lifecycle. Even relatively stable software products, such as operating systems, receive patches or new features routinely. This is only more challenging when we take Software as a Service (SaaS) into consideration, where updates may be occurring on a nearly continuous basis. In practice, this generally results in a net positive for users, since it standardizes the patching process and timeline, leading to faster resolution of vulnerabilities when they are discovered across all users of that software. Given the ongoing and interactive nature of changes in cloud environments, effectively generating and sharing SBOMs in a cloud environment poses challenges. For starters, there is a real risk of SBOMs becoming outdated so quickly, that a false sense of security could be created if SBOMs aren’t updated at the same pace as the code itself.

Further, clouds also pose the problem of layered deployments. Services depend on other services, some of which are not publicly available. How can we determine the appropriate place for the SBOM to “start” and “end”? For example, how far down the infrastructure stack should an SBOM go? Should it include the dependencies used to deploy and operate a cloud service and/or the infrastructure needed to run it? Further, cloud services are highly dynamic and requests can be routed based on all kinds of inputs (e.g. who the user is, location, cohort for A/B testing, cohort for blue/green deployment). This is true for both the initial service call as well as all subsequent calls. During a service rollout, it’s also likely that different versions of a service may be in use at the same time.

These challenges are not insurmountable, but they do suggest that real-world test cases and pilots could prove valuable in addressing them. The General Services Administration (GSA) and Department of Homeland Security (DHS) are well-positioned to be able to develop and run meaningful pilots with industry and we also believe that existing cloud certification processes, such as FedRAMP, could be leveraged in establishing SBOM related requirements and validation, rather than through general procurement contract clauses or regulations.
Conclusion

The concept of an SBOM is rooted in a traditional software publisher scenario. However, increasingly, that assumption is misplaced. Certainly, as suggested by some, having a vendor-produced SBOM could have assisted in discovery of the use of #Log4j. Indeed, this particular use case has energized many around the idea of SBOM and has brought increased attention to implementation. However, digital transformation in the enterprise (and in government) means that all companies and organizations are becoming software developers themselves. Thus, it is likely that a significant share of the deployment of Log4j was outside traditional vendor-provided software (or in products that were no longer supported by vendors). As such, it is likely that some portion of future vulnerabilities will be similarly “hidden”, even with a robust SBOM ecosystem in place. This is an important reminder that even with the challenges discussed and addressed here, SBOM remains only a part of the overall solution to procuring, implementing, and managing secure software.

The discipline needed to produce robust, resilient software-based solutions requires everyone to improve their software use and production processes as envisioned by the Cyber EO’s focus on open and transparent software management over the life cycle of a software product.

May 2022