ATARC

SBOM Challenges and Opportunities

Summary of Roundtable, hosted by ATARC in January 2023

WHITE PAPER

Following watershed security incidents, like SolarWinds, where multiple Federal agencies were compromised by software supply chain attacks, new mandated software development requirements and security guidelines have emerged to help mitigate future risk. One such mitigation requirement is the use of SBOMs, or Software Bill of Materials.

EO 14028 Section 4, OMB M-22-18, and NIST SP 800-218 Secure Software Development Framework (SSDF) highlight how Federal agencies are prioritizing SBOM generational and monitoring across government. An SBOM is a list of all software components and dependencies used in an application or system. Creating an SBOM is critical to secure software development, improve supply chain transparency, and manage open-source software components over time. With such transparency, the risk of security breaches can be minimized.

"Now that we have SBOMs, what do we do with them?"

In a recent roundtable discussion hosted by the Advanced Technology Academic Research Center (ATARC), industry experts from across the Federal government came together to exchange ideas on SBOM compliance.

Using SBOMs

SBOMs are now a mandated component of developing, implementing and managing IT systems in the Federal government. Having a list of all software components and dependencies in an application or system assures government agencies that suppliers know exactly what they are selling, and they have the capabilities to respond when new vulnerabilities are announced.

Over the last few years, SBOMS have matured and are considered a tool to help detect vulnerabilities within new containers, applications, open-source libraries, and across the supply chain. However, roundtable participants caution that SBOMs were never designed to solve all problems, rather they are a helpful tool to identify vulnerabilities so ultimately agencies can focus on high priority risks. Risks identified with the help of SBOMs can vary from a foreign adversary compromising an open source library to other vulnerabilities that could potentially pose a legal risk to the agency.

SBOM Challenges and Opportunities

The technology of a few agencies at the roundtable falls into two paradigms: one built on new platforms with an emphasis on containerization, and one built on legacy platforms that continue to deploy monolithic applications. SBOMs are convenient with containers as they are updated whenever a new vulnerability is discovered, but more tedious with legacy systems as staff must create and update the SBOM as they build and deploy updates.

Agencies operating in these two paradigms may benefit from tools that can consolidate SBOMs from both platforms in one place. This would be particularly helpful if an exploit like Log4J were to happen in the future, agencies could conduct a real-time search to find where vulnerabilities exist in all of their products.

Some roundtable participants question whether agencies could add onto the functionality of an SBOM to enable predictive reasoning, not just reactive reasoning, to better identify compatibility issues and vulnerabilities that may exist when two components interact with one another. Currently, an SBOM is a static representation of a piece of software at a given time without the ability to analyze. Most roundtable participants agree that while analysis is lacking, for the time being, the purpose of an SBOM should not be diluted. Many agencies are still trying to adopt SBOMs and determining the best way to use them in their current form.

One of the biggest challenges is knowing what to do with the information collected in an SBOM. There is currently nothing in an SBOM that tells agencies whether a vulnerable dependency is actually being activated or used. SBOMs do not capture the full risk picture of a given organization, but if agencies augment an SBOM with various tools, a bigger risk picture could emerge. SBOMS are most useful as a static representation of what is in the code from a dependency perspective. Agencies can then layer vulnerability data to identify risk.

Government agencies face many of the same challenges as other industry verticals, especially those that are highly regulated. However, there are unique challenges in government that can complicate security, such as using numerous contractors for software development. A single software system might be developed by a number of different vendors, which is why SBOMs are helpful to provide visibility into the software development process.

Procurement and software acquisition is also a challenging process for most government agencies. SBOMs are helpful for vendors and government agencies alike because they ensure best practices are being followed without being too prescriptive. They provide accountability and transparency in the process without dictating the process. SBOMs are also a good way to ensure that software supplies know exactly what they are shipping. Knowing this information can also help government agencies project the full cost of ownership of a given product.

To improve the usability of SBOM information in various use cases, agencies are working to create an exchange platform for someone to communicate whether a product is or is not affected by a vulnerability. This will allow teams to determine whether a vulnerability is pertinent to their systems. The feedback on the platform may not always come from a supplier, rather they are attestations from actual users. ATARC and other roundtable participants discussed running a pilot to test this sort of platform.

> *"One of the key thoughts going forward is* WHAT *an SBOM might be."*

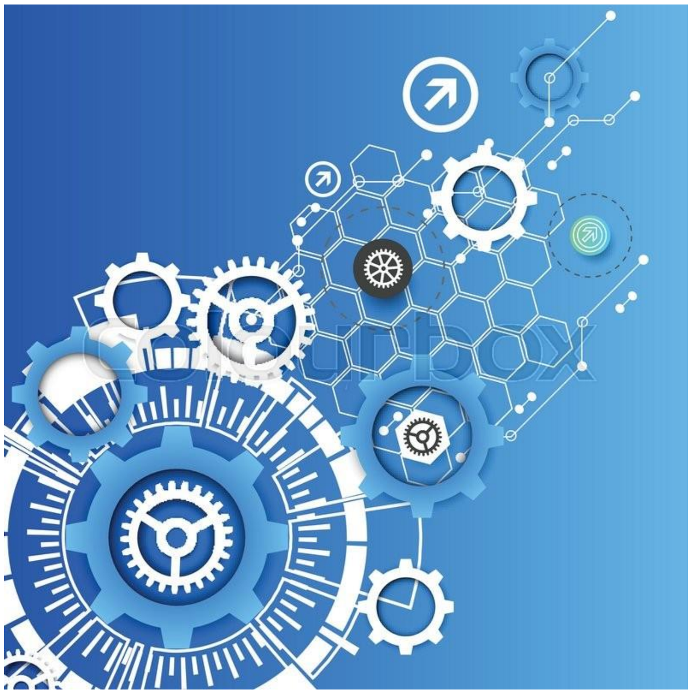
Another challenge indicated by the roundtable is how SBOMs work with cloud-based applications, SaaS connectors and other app add-ins. With so many connectors, app add-ins, and other considered software, the question becomes where is the line drawn when it comes to SBOMs? There are commercial tools that are able to consume third-party SBOMs and monitor them the same way as other internal applications. But agencies may not want their SBOMs to include cloud connections because their vulnerability information is dynamic. The utility of capturing dynamic information in a static format may not be helpful.

As agencies begin to think about how to adapt infrastructure for AI and other dynamic technology, the conversations shift from focusing on software transparency to service transparency. This is proving to be a very hard problem to solve. Figuring out how to globally reference an API is challenging and remains a work in progress.

Questions & Answers

What is the process for producing SBOMs for contracted software developers?

- Most contractors follow the direction of the agency, especially if they are developing on agency platforms.
- Agency includes SBOMs as part of the building process to ensure they are created every time. The Agency will request the SBOM up front and then check the final build product to ensure they match.
- Businesses that are able to produce SBOMs, it gives them an implicit advantage over companies that do not have SBOMs built into their processes.
- One agency shared that while SBOMs are in theory a good way to plan ahead and identify vulnerabilities, that is not reality. Most of the time, the agency ends up trying to connect the puzzle pieces together to determine what they are missing.



What process do you have in place for responding to security issues that you discover via SBOM monitoring? What happens if you identify something, but it does not apply to your organization?

- Having a centralized repository to track SBOMs is critical to knowing what applications an agency has.
- Agencies should verify and validate the vulnerability. Starting with what the threat is, whether it is someone using a bad SBOM tool, if a vendor is lying, or if an attacker is compromising the actual SBOM creation process. Each of these scenarios require different ways to verify and validate, as well as solve.

What are some concrete things agencies should do before starting a new project?

- Default to a cloud native application architecture. Cloud native doesn't necessarily mean that it needs to be hosted in the cloud, rather posted on a platform that is cloud native.
- Give consideration to what dependencies are included at the beginning. What is the project quality? What might happen if a vulnerability is discovered? Is this component responsive to those vulnerabilities?

For events on this and other topics, please see ATARC **Events calendar**.

For on-going conversations on emerging technology topics, please visit ATARC **Working Groups**.