2101 Wilson Boulevard, Suite 700, Arlington, VA 22201-3060 • (703) 522-1820 • (703) 522-1885 Fax • NDIA.org

17 June 2021

National Telecommunications and Information Administration
Attn: Evelyn L. Remaley
NTIA, U.S. Department of Commerce, Room 4725
1401 Constitution Avenue NW,
Washington, DC 20230

Re: Request for public comment for NTIA-2021-0001, minimum elements for an SBOM, and what other factors should be considered in the request, production, distribution, and consumption of SBOMs.

Ms. Remaley:

We are pleased to respond to the request for public comment for NTIA-2021-0001, Minimum Elements for an SBOM, and what other factors should be considered in the request, production, distribution, and consumption of SBOMs.

The National Defense Industrial Association (NDIA) represents more than 1,600 corporate members and over 80,000 individual members from small, medium, and large contractors. Our members and their employees feel the impact of any policy change made in how the United States equips and supports its warfighters. Our comments provided below come from this diverse membership and represent a broad range of perspectives across the defense industrial base (DIB).

NDIA is supportive of the overarching policy objectives behind Executive Order 14028, “Improving the Nation's Cybersecurity,” but also encourages all federal agencies to provide adequate time for industry to comment in advance of potential policy, regulatory, and contractual changes to help with effective implementation. In this regard, NDIA and its members welcome the opportunity to comment on the minimum elements for an SBOM, and what other factors should be considered in the request, production, distribution, and consumption of SBOMs. We hope that our comments and questions will help NTIA create and publish a list of “minimum elements for an SBOM” that raises the bar on security while also giving due consideration to costs and benefits of new requirements. Indeed, a rushed rollout of new rules could ultimately lead to setbacks in our shared goal of improving cybersecurity defense in the United States as we have seen in other areas.

In addition to working in the DIB, our membership supports the Federal Civilian Executive Branch (FCEB) agencies, where decisions are often decentralized and have less standardization. We encourage NTIA to ensure that the FCEB perspective is fully incorporated into the SBOM decisions. Further, we would encourage this effort to include requirements for protecting the SBOM, guidance on its maintenance and use, as well guidance for enabling the SBOM supports for DevSecOps, speed-to-delivery, and innovation and agility with enhanced security.

Questions:

1. Are the elements described [in the notice], including data fields, operational considerations, and support for automation, sufficient? What other elements should be considered and why?

The elements and intent described in the Executive Order and resulting notices are a suitable foundation but one item that should be added is a rationale section that identifies the benefits of the SBOM. If you are trying to incentivize firms to incur potentially substantial additional costs, then identifying the benefits will help such as how software quality will be improved. This section should also identify the effort required. With many repair projects, there is a scale, often running from 1 (you can do it in one hour) to 5 (don't do it unless you know what you are doing - hire an expert). Something similar should be done here.

Another recommendation would be the inclusion of a list of resources that someone can draw on if they have further questions. All these suggestions will make the document attractive to potential adopters and especially Small to Medium sized businesses.

The elements in the notices are also a suitable foundation for the metadata requirements. The fields in an SBOM should reflect sufficient coverage of a program's components, while taking into consideration the fact that there is a cost and effort associated with gathering metadata from those components. A balance needs to be struck so as to minimize the effort required by manual human authorship, while still maintaining sufficient coverage of a program's component makeup. Subsequently, we are generally in agreement that the data fields outlined by the NTIA strike this balance; namely, Supplier Name, Component Name, Unique Identifier, Version String, Component Hash, Relationship, and Author Name. While these may be sufficient for a minimum viable SBOM, it does not reflect an easily generated SBOM, however, as some of these fields may be quite difficult to automatically parse given differences in standards and formats across various software platforms and artifacts. For example, production of the hash of a component will not always be possible in the case of secondary SBOM authorship, for example when binary analysis is used to produce the SBOM, and the code being analyzed has been statically linked. We suggest that NTIA should clarify the intended purpose of the hash. We ask

NTIA to further clarify the method for creating the hash (who, when, what, how), and in particular whether the hash can or should be derived from binary or source.

While some tools do exist that can produce complete SBOMs, they rely on the assumption of complete and available metadata and access to a dependency manager, which one does not necessarily have. We have not yet seen full coverage to support automation across all ecosystems and artifacts that would sufficiently remove manual authorship. That said, the noted basis can be further extended to support the objective use cases of: vulnerability/weakness analysis; vulnerability and incident response; supply chain assessment, pedigree and integrity of the software product and dependent elements; and confidentiality and integrity concerns related to the SBOM itself.

We advise against including a Vulnerability List in the data fields specifically. The Executive Order states that "buyers can use an SBOM to perform vulnerability or license analysis, both of which can be used to evaluate risk in a product," and a vulnerability list is the product of one or more vulnerability analysis efforts, not an SBOM. This is a further security step than what it required in the SBOM. If the SBOM is equivalent to ingredients in food packaging, then including a vulnerability list would equate to adding health risks for each ingredient to food packaging as well. The ability to exchange with a standard format for currently associated vulnerabilities against the SBOM is advisable, though the analysis and reporting of such concerns may be held in a separate location.

We would like to note that different types of software (e.g. Flight safety critical, weapons systems, national security systems, Infrastructure/Ground) may have different cyber security, criticality and safety postures. SBOM items and meta-data may differ for each domain but depending on domain, care must be taken to ensure no undue process or cost is incurred.

Lastly, we believe that the value of an SBOM is only fully realized when operational deployments of a software product or system, are maintained for each deployment to reflect the unique configuration of that deployment. As a result, vulnerabilities, incidents and other concerns such as a supply chain compromise must take into account that an SBOM may vary for a single product across deployments and require adequate configuration control.

We also recommend tracking provenance meta-data with linkage to the SBOM meta-data. This meta-data is likely to be reported on a different cycle than the SBOM and so is treated separately but with intent to maintain linkage to the SBOM.

2. Are there additional use cases that can further inform the elements of SBOM?

The NTIA SBOM Use Cases (Nov 2019) captures the majority of industry technical and procedural use cases. In these use cases, considerations for the SBOM as a living document representing a product, as it is proposed, designed, developed, maintained, and eventually disposed of. The SBOM, as a result must accommodate partial information, baseline updates to include revised licenses, and updated or end-of-life software elements. Further, the SBOM must consider the configuration management/accuracy of deployments or platform configurations; where one platform or deployment may maintain a different SBOM from another platform. This is critical to assess implication of weaknesses, vulnerabilities, or threats against part or all of the SBOM. Further, the threat landscape and evolution of discovered or exposed weaknesses in one or more elements of a software supply chain, and dependencies will result in re-evaluation of future security ratings (i.e. "Energy Star-like" representation).

Work is required to establish standardizations for identifying/characterizing/quantifying risks in partial or incomplete information of an SBOM, and the criticality of that component/element within the SBOM.

3. SBOM creation and use touches on a number of related areas in IT management, cybersecurity, and public policy. We seek comment on how these issues described below should be considered in defining SBOM elements today and in the future.

a. Software Identity: There is no single namespace to easily identify and name every software component. The challenge is not the lack of standards, but multiple standards and practices in different communities.

Mandating a universal or standard software identification method across a domain, such as the DoD is feasible. There are and will continue to be, however, multiple standards and practices used at the product origin due to those products dependencies on commercial and open-source software libraries and products. Any product or program that utilizes components from open-source and/or proprietary sources is likely operating across namespaces in which there can be no assumption of standardization and no imposition of standardization. One potential solution is to use the DoD PlatformOne approach of IronBank, which containerizes many widely used open source and commercially available software products and then uniquely versions and identifies them. These packages have been scanned, catalogued, and vetted for use by PlatformOne and its user community. Reinventing this methodology is unnecessary.

b. Software-as-a-Service and online services: While current, cloud-based software has the advantage of more modern tool chains, the use cases for SBOM may be different for software that is not running on customer premises or maintained by the customer.

In our view, all "as a Service" delivery mechanisms present a different use case. The code base changes at a rapid pace. It is not unusual to update code used to provide the services in cloud environments multiple times a day. This reality makes an SBOM obsolete almost immediately and render it essentially meaningless for assessing risk. A customer would not benefit from a constantly changing document or manifest. We believe that a one size fits all approach has the potential to increase the risk to Federal networks by undermining the benefits of such services.

c. Legacy and binary-only software: Older software often has greater risks, especially if it is not maintained. In some cases, the source may not even be obtainable, with only the object code available for SBOM generation.

The SBOM generators that are available typically do rely on access to a dependency management file, such as a POM, to parse library information. That said, there are open-source tools that do perform binary analysis and other built-in commands to examine artifacts. These tools pose limitations in how much information can be parsed from the raw artifacts. There are also commercial programs, such as Synopsis's Black Duck Binary Analysis and NetRise program, which perform SCA operations on these artifacts. While SCA tools have traditionally focused more on source code, the maturity and capability of binary SCA tools are improving quickly, making them a desirable choice to complement source code SCA tools in generating accurate and reliable SBOM. In the case of legacy and binary-only software, even a partial SBOM generated by a binary SCA tool will contribute greatly in uncovering security vulnerabilities compared to accepting or rejecting the legacy and binary-only software as a black box.

d. Integrity and authenticity: An SBOM consumer may be concerned about verifying the source of the SBOM data and confirming that it was not tampered with. Some existing measures for integrity and authenticity of both software and metadata can be leveraged.

Some considerations must be made with respect to verifying the integrity and authenticity of the SBOM. First, the SBOM likely is a composite of multiple SBOM from one or more producers. As previously noted, this means that production of the hash of a component will not always be possible in the case of secondary SBOM authorship.

e. Threat model: While many anticipated use cases may rely on the SBOM as an authoritative reference when evaluating external information (such as vulnerability reports), other use cases may rely on the SBOM as a foundation in detecting more sophisticated supply chain attacks. These attacks could include compromising the integrity of not only the systems used to build the software component, but also the systems used to create the SBOM or even the SBOM itself. How can SBOM position itself to support the detection of internal compromise? How can these more advanced data collection and management efforts best be integrated into the basic SBOM structure? What further costs and complexities would this impose?

Chain of custody for compile/link tool chains is not a widespread practice, even for commercial compiler companies that produce tools that generate binaries for the safety and security critical U.S infrastructure and defense sector software. Secure supply chain best practices help mitigate these risks but a comprehensive repeatable process for detection of internal or external compromise is still needed. In some situations, we use development tool/environment diversity to guard against common mode errors in high assurance safety critical software. While these techniques are higher in cost, they should be considered in the trade space of “critical software” to mitigate insider threat and external supply chain compromise.

f. High assurance use cases: Some SBOM use cases require additional data about aspects of the software development and build environment, including those aspects that are enumerated in Executive Order 14028. How can SBOM data be integrated with this additional data in a modular fashion?

As part of high assurance use cases, an example is DO-330 tool qualification criteria. Those qualification criteria are safety-centric and are intended to provide more confidence in error free software. For example, if a code generator tool's output is part of an airborne software and could insert an error, that tool is subject to higher scrutiny. In that context, for an SBOM, that situation and tool could equate to a separate meta-data that designates the criticality of the tool and associated product. In DO-330 terms, it would be Tool Qualification Level (TQL), 1 through 5. These criteria could be easily added as extensions to the SBOM, and in how the SBOM is utilized with respect to analysis and reporting.

g. Delivery. As noted above, multiple mechanisms exist to aid in SBOM discovery, as well as to enable access to SBOMs. Further mechanisms and standards may be needed, yet too many options may impose higher costs on either SBOM producers or consumers.

We recommend standardizing upon a "core" set of SBOM meta-data fields for "critical software" as recommended above and by working with industry to select and mandate a particular format standard. The format standard needs to support optional extensions for high assurance. We generally advocate adopting the standardized format that the NTIA has adopted, namely SPDX. Increasing the number of standardized formats brings risk, as the choice likely will confuse many who don't know how or why to choose one format over another. Priority should be given to the automated production of SBOMs through dependency discovery and supporting more functionalities that work with the existing formats. Adoption and authorship of SBOMs will increase if less manual work is required.

h. Depth. As noted above, while ideal SBOMs have the complete graph of the assembled software, not every software producer will be able or ready to share the entire graph.

A minimum depth cannot be explicitly mandated because not every software package (e.g. open source) will be accompanied by desired or complete data. Empty or suboptimal graphs are to be used as indicators that risks, gaps in relevant information and mitigation plans needs to be identified depending upon the context of how the package or dependent element is used, and its upstream and downstream impacts. These indicators should be prominently visible to development and product security engineering teams using decision aids through commercial tools like Sonatype Nexus that include push notifications. Fundamentally, the depth, completeness, and accuracy of information may vary from element to element. It is recommended that the metadata characterizing part or all of the SBOM include a characterization of completeness, pedigree/provenance or other quality factor, to support analysis of partial/incomplete/complete metadata which can lead to a risk determination.

Data logs, especially automatically generated ones, can create volumes of data that is not part of the production system. Retaining logs for all tools and all builds should have a defined retention period. Provenance artifacts for a particular release artifact are already defined and include the relevant security artifacts.

i. Vulnerabilities. Many of the use cases around SBOMs focus on known vulnerabilities. Some build on this by including vulnerability data in the SBOM itself. Others note that the existence and status of vulnerabilities can change over time, and there is no general guarantee or signal about whether the SBOM data is up-to-date relative to all relevant and applicable vulnerability data sources.

Vulnerability analysis should be a separate functional activity, that leverages a robust SBOM and interface standard (SPDX and extensions, as example) to develop an integrated repository that associates and derives potential weakness, vulnerability, threat, or unknown concerns. Unknown concerns would be associated with risk due to a lack of clear enumeration and characterization within the BOM. The lack of information, in and of itself presents a gray risk that could be used to indicate and focus assessment by the developer/program. This risk categorization for unknown or missing information requires standardization.

Regarding the existence of and change of vulnerabilities over time, Automation tools built within DevSecOps software factories allows one to maintain configuration control over SBOMs, and accommodate frequent (e.g. daily) vulnerability status changes to part or all of a package and associated SBOM. This is accomplished and supported using already-available tools like Artifactory, XRay, and Gitlab.

j. Risk Management. Not all vulnerabilities in software code put operators or users at real risk from software built using those vulnerable components, as the risk could be mitigated elsewhere or deemed to be negligible. One approach to managing this might be to communicate that software is "not affected" by a specific vulnerability through a Vulnerability Exploitability eXchange (or "VEX"), but other solutions may exist.

We would concur that a measure of impact or the affect as a result of a vulnerability is important. Perhaps more important, is the determination of what the product or mission performance consequence of one or more weaknesses/vulnerabilities or threats should they be exploited. Following this resiliency and mission impact perspective weakness, vulnerability, and threat must be assessed with respect to the adversity to the product and its performance. For example, a critical vulnerability or exploit may exist against a specific software element or library, the likelihood and more so consequence to the overarching product, may be such that it does not warrant elevation or mitigation, when compared to cost or other factors. Conversely, a relatively simple weakness may expose a greater product or mission impact that would warrant mitigating a relatively benign concern.

The vulnerability exchange format, must align with the SBOM unique identification and characterize both individual and product/mission level criticality to properly prioritize, plan, and mitigate.

4. Flexibility of implementation and potential requirements. If there are legitimate reasons why the above elements might be difficult to adopt or use for certain technologies, industries, or communities, how might the goals and use cases described above be fulfilled through alternate means? What accommodations and alternate approaches can deliver benefits while allowing for flexibility?

Just like the internet economy that incentivizes producers and consumers of IP packets like content delivery networks, internet exchanges, and internet service providers to deliver/route traffic reliably and in a timely fashion, the DoD and its broad supply base can incentivize commercial software suppliers to comply with SBOM best practices through supplier management relationships and additional compensation models. Even so the ultimate burden of proof is on the end product developer that relies on those software packages. Making the process too arduous (update frequency, degree of specificity, depth of supply chain, etc.) will likely reduce the effectiveness and drive innovators away from the USG market or result in low quality deliverables that will just lead to a whole lot of auditing to validate by the USG.

Either additional costs must be incurred to “verify but trust” or rely completely on the “zero trust architecture” to isolate “critical software” from potential bad actors. In some instances, where national security and human safety are at risk, the cost to “verify but trust” must be paid. Until the DoD and its supply base complete Dr. Roper’s transformation to the “digital trinity”, these issues will remain.

NDIA appreciates the opportunity to comment on the Interim Rule. The point of contact for this comment is Robbie Van Steenburg, NDIA’s Regulatory Policy Associate, who may be reached at (703) 247-2562 or at rvansteenburg@ndia.org.

Very respectfully,

National Defense Industrial Association