ITI Promoting Innovation Worldwide

June 17, 2021

Ms. Evelyn Remaley
Acting NTIA Administrator
U.S. Department of Commerce
1401 Constitution Ave NW
Washington, DC 20230

RE: ITI Comments Responding to National Telecommunications and Information Administration RFC on Software Bill of Materials Elements and Considerations (RIN 0660-XC051; NTIA 2021-0001)

Dear Acting NTIA Administrator Remaley:

The Information Technology Industry Council (ITI) appreciates the opportunity to provide feedback in response to NTIA's RFC on *Software Bill of Materials Elements and Considerations* as the agency seeks to fulfill the tasks laid out in the recently signed *Executive Order on Improving the Nation's Cybersecurity ("the EO").*

The Information Technology Industry Council (ITI) is the premier global advocate for technology, representing the world's most innovative companies. Founded in 1916, ITI is an international trade association with a team of professionals on four continents. We promote public policies and industry standards that advance competition and innovation worldwide. Our diverse membership and expert staff provide policymakers the broadest perspective and thought leadership from technology, hardware, software, services, manufacturing and related industries.

In addition to the work NTIA is undertaking as a result of this Executive Order, we applaud NTIA's multi-stakeholder efforts to facilitate transparency of, and trust in, software components as these two tenets are foundational to improving cybersecurity.

Our response considers both the explicit questions laid out as well as additional considerations that we believe are important to address as NTIA seeks to develop a set of minimum elements to be included in an SBOM.

Below are some general thoughts, followed by responses to specific questions.

NTIA, and the USG more broadly, should consider that SBOMs could create access to market information not otherwise available. Publicly disclosed SBOMs could subject the federal government to increased cybersecurity risk, as it could reveal how many critical systems use a particular code or IP product. This could reveal insights about sensitive market dynamics or competitive opportunities. We therefore recommend that NTIA avoid requiring public or any other type of unnecessary disclosure of SBOM data.

Relatedly, the scope of SBOM needs to be further clarified to delineate whether an SBOM should include first-party code or whether it covers only third-party code, and what any differences in practice might be if it includes both. SBOM by itself doesn't enforce security, but by including first-party code and linking information back to source code (via Git commit hashes or other identifying information), thereby supporting other security assurance practices (such as vulnerability remediation), it provides a fuller picture of the "ingredients" of the software. However, including first-party code in an SBOM can create challenges. Different companies use different version-control-systems, and each of those systems track information differently. Additionally, not all native code security issues have CVEs assigned to them, not all versions of the code are supported in perpetuity, and not all patch applications result in a version change. NTIA rightly recognizes in the RFC that certain standards in development, like the Cyclone DX SBOM specified PURL (Package URL) may be applicable; however, these standards are still not adopted broadly and thus, difficulties around consistent naming and identification may arise. Finally, publishing first-party metadata about their in-house source code may come with additional concerns from companies about proprietary information. A vendor who has incorporated external IP into a product they provide to the government may be restricted by law or contractual obligations from disclosing it regardless of SBOM requirements. The use of open source modules is also considered proprietary information as such information can be used to reverse engineer the software. Therefore, we recommend that NTIA, and the USG more broadly, avoid mandating complete disclosure as a part of SBOM requirements.

NTIA, and the USG more broadly, should consider who is responsible for providing the SBOM. For example, a business-to-business supplier does not have control over what their customers might do with the software. Indeed, customers could elect to engage or ignore various components of software within their systems or otherwise modify it to meet their needs. If said customer of the software component supplies their end-product to the government using the supplier's components, who is responsible for providing the SBOM? The supplier has no way of knowing which components of their product software or firmware have been utilized, modified, or ignored by the customer, so requiring such a supplier to provide an SBOM for the original product would be a best-guess measure and likely inaccurate. As such, NTIA should clarify that the SBOM is provided by the end-product supplier, and in this case the federal contractor or direct supplier of the finished product or solution. This clarification would further support consistency with current approaches (such as NIST 800-53) and the implementation of the SBOM requirements under section 4 guidelines for federal procurement purposes.

NTIA, and the USG more broadly, should seek to minimize overlap with existing standards and frameworks. In order to appropriately respond to the objectives of the EO, we recommend that NTIA refine their focus to the minimal elements of the BOM, leveraging existing standards and frameworks (NIST SP 800-53, NIST 800-161, SPRIS, and others), to ensure that the elements proposed do not duplicate or contradict existing guidance for federal contractors. If there are existing controls in place, we encourage the USG more broadly to look to those frameworks instead of laying down an additional SBOM requirement. For example, there are already existing regimes for direct, protected disclosure of information of component inventory that consider the applicable access controls, given the sensitivity of the information disclosed and federal use case. NTIA should consider how to maintain consistency with such regimes and focus on the minimum technical elements of the SBOM deliverable. While the RFC includes a discussion of SBOM delivery, including tooling/attestation infrastructure, as in scope, these elements should be distinguished from minimal SBOM elements for the purposes of the EO. Moving forward, NIST and NTIA should invite additional stakeholder input on the appropriate communication and delivery of the SBOM for other use cases. We encourage NIST to further prioritize broader discussion of whether and how to include these topics in any guidance it ultimately issues regarding SBOM standards, practices, or procedures pursuant to Sec. 4 of the EO.

Although an SBOM can provide value, we encourage NTIA and the USG more broadly to also consider that an SBOM as currently construed may not provide software users with actionable vulnerability information in certain contexts. This is the case for several reasons, some of which are raised as issues for consideration in the RFC. For example, under currently established practice a developer that uses a third-party package may incorporate it in its entirety, or select individual files, or functions, or even lines of code. These fractional pieces do not have their own identifiers or versions and are therefore not possible to meaningfully reflect with an SBOM at this time. Such practices highlight the complexity around vulnerability tracking in third-party code. While it may be a legitimate aspirational goal to improve cybersecurity by developing new practices and standards for secure software development and coding, we suggest that such work must first be advanced (most appropriately, through processes run by NIST) prior to determining that such elements are ripe for inclusion in an SBOM. A further complication is that vulnerability information is very context-specific. Without the ability to identity and reflect the appropriate context, it is challenging to act on certain types of information that may be provided in an SBOM. That being said, we also appreciate that NTIA recognizes this challenge and has proposed that one way to address it is to indicate that the software is "not affected" by a specific vulnerability by tying it to a Vulnerability Exploitability Exchange (VEX). We further elaborate on this point in response to question 3(i). We also note that ongoing industry efforts in this space are needed. For example, SPDX has been considering ways to share responses to known vulnerabilities as part of an SPDX document - if a provider knows a vulnerability does not apply because it is not using the part of the component that is vulnerable, then that contextual information could be disclosed.

Any SBOM requirement that may result from this EO could be premature. Although not explicitly in scope of this RFC, we think it important to emphasize that further down the road, as NIST consults with stakeholders for the purpose of identifying practices that enhance supply chain security pursuant to sec. 4 of the EO (including to consider whether standards, procedures, or guidelines regarding SBOM are sufficiently mature), it should also take into account that SBOM is not yet widely practiced and therefore, it may be too soon to actually identify it as a best practice sufficient to potentially form the basis of a requirement. The elements of an SBOM should reflect currently established best practices; we shouldn't add requirements to an SBOM first and then develop consensus practices and standards to match those requirements later. We are not suggesting SBOM should not be explored further – this RFC is a helpful first step to begin the process of identifying and driving consensus, as is the work that is being undertaken by NTIA more broadly.

Responses to Questions

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

At this stage of adoption of SBOMs in the market, and the lack of harmonized approaches to them, we strongly urge NTIA to keep its recommended minimum elements simple and not try to solve the entire supply chain challenge via SBOM. We also recommend that NITA focus specifically on determining the minimum data field elements and to consider the operational aspects separately through a NIST-driven process.

Data fields

We believe that the elements included as minimum fields in the SBOM for software transparency purposes should be Component Name, Supplier Name, Version of the software component, and Origin (which could be different than Supplier Name).

NTIA sets out "dependency relationship" as one of the "baseline component information" pieces to be included in an SBOM. We are concerned with the inclusion of this element and can foresee challenges emerging. As a general matter, dependency is relevant for functionality (the author of software) not the consumer of the SBOM (integrity and chain of trust care). Therefore we recommend excluding "dependency relationship" as a minimum element.

We also believe that the "cryptographic hash of the component" be included only as an optional element and not a required element. The RFC is vague on what the cryptographic hash involved should be and several outstanding questions remain, including what hash should be used for products that require multiple downloaded files to install and what hash is used when compiling from source versus using the binary. NTIA should not overly complicate the minimum elements by requiring an SBOM to contain a cryptographic hash and a hash especially should not be required for the sub-components. A cryptographic hash alone is not useful because there is nothing to search for. There is no database that can be used to look up a hash. In addition to the vagueness of this requirement, NTIA should not require a supplier to independently compute a cryptographic hash of a component in their software code. "URL" does not appear to be a predefined identifier that would be included in the SBOM. If there is a canonical URL for each version of software (like a tagged GitHub branch) that would make it significantly easier on some consumers of SBOM.

An additional data field that may be helpful to consider as an additional minimum element is Wrapper Data – this would include information on who wrapped the data (entity), the date it was wrapped, and the signing certificate/key of author – all of which would help to clarify the chain custody.

Operational considerations

Delivery. One of our concerns with the draft, and the minimum elements as currently identified, is how access control is construed. Although "delivery" is one of the areas outlined under operational considerations, it is only loosely defined, stating that "SBOMS should be available in a timely fashion and have proper access permission and roles in place." It is difficult to opine on the sorts of data that should be collected and shared via an SBOM without understanding the scope of access. As such, we encourage the delivery of the BOM minimal elements to be determined by NIST under Section 4 of the EO, consistent with current access controls/federal supplier delivery methods under NIST SP (including 800-53), FARs, etc. NTIA should convene stakeholders on the broader delivery of the SBOM question (outside of the EO), ideally in collaboration with NIST. This should be followed by international standardization. However, to the extent that NTIA does not exclude delivery from the scope of this RFC, we believe that access to the SBOM be limited to a need-to-know basis and that roles should be specified in the applicable contract. When externally sharing sensitive information in the SBOM, consideration needs to be given for authorization for privileged access. Secure, controlled distribution avoids potential issues for the vendor and operator that would result from public disclosure.

Depth. The RFC rightly notes that "complete depth may not be feasible." Indeed, we agree that mapping the entire graph is misguided and likely not possible in some cases. We therefore encourage

NTIA to focus on building the syntactic tools to help lead to a standards-based process. This will help industry to figure out what a useful level of depth might be. NTIA should also consider developing a centralized managed database of SBOM objects to versions, which would allow software developers to indicate, for example, that they use open ssl 1.1.1h. NTIA's database would host the SBOM file for all of the versions of Open SSL including 1.1.1h and companies could simply reference that. However, we note such a database could grow quickly and will require significant resources to maintain, update and secure.

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

The usage scenarios of the SBOM might differ from one software to another. There might be cases where a software component that is used by a product and included in its SBOM is identified as vulnerable, but this vulnerability may not be relevant to the customer's use case. In such scenarios, SBOM might give a false signal about the vulnerability of the product. Vulnerability ratings can be use case dependent. A vulnerability must be exploitable to be a risk. Thus, the assignment of the criticality level of the vulnerability and the control of SBOM data should be managed by the organization developing the final product.

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.

We encourage NTIA to convene stakeholders to explore methods to address questions related to software identity, including how to handle inventorying beyond the version number. Although the RFC contends that the "challenge is not the lack of standards," further standardization work is needed. As such, NTIA should maintain flexibility to allow for adoption of such standards absent a unified convention. In order to do this effectively, there need to be explicit definitions for version numbers, especially in open source projects with multiple supported branches at once (e.g., Python). A solution is also needed when a project changes their versioning systems.

Next, open source is a key example of quality code that can be created, allowing for rapid innovation and transparency aligned with the principles of the EO. However, this type of code can be overweighted in an SBOM process as it is created openly and available to be integrated into any solution – known and unknown. It would therefore to be helpful to take this into account when determining how to achieve the goals of the EO.

Finally, since many projects do not update to the latest version of an open source project and the same open source projects are often built in different ways, it is crucial that the BOM is not limited to a numerical value.

It may also be helpful for NTIA to consider another common way software is disambiguated - by linking to the source control as well as the commit ID. This is not ubiquitous, but fairly common.

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.

Software-as-a-Service is indeed a far different use case. Indeed, the code base in some cloud offerings may change daily – or even more frequently. In such a case, as it is developing minimum elements for an SBOM and pursuing additional guidance, NTIA should consider whether it makes sense to require that manufacturers produce a software bill of materials that would become almost immediately obsolete.

Additionally, in cases where the service provider is responsible for all patching and upgrades, it is unclear who would benefit from a resource that is used to provide out-of-date material that is not actionable by the customer base. Note that even in cases where a customer is responsible for patching (such as components a customer may run on infrastructure as a service (IaaS)), a software bill of materials would not per se indicate the presence of vulnerabilities, how severe they are (e.g., via a Common Vulnerability Scoring System (CVSS) Base Score), or where to obtain necessary patches. These considerations are important to customers in ascertaining their ability to determine a problem, how bad it is, and what action needs to be taken to mitigate and/or fix it.

We recommend that NIST, in conjunction with NTIA, launch a discrete stakeholder consultation and/or working group effort to focus on services. Much of the work undertaken on SBOM to date has not focused on services and as evidenced by the above, there is a need to consider elements and operational issues specific to services.

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.

One way to address considerations around integrity and authenticity is to include some version of a “last update” field in an SBOM. This would help to ensure that the latest version is available, but that old revisions are also available, possibly with a “valid from” and “valid to” indication in past versions (there may be some overlap when a software rollout occurs.) It is important to maintain the history of older software to determine possible impacts if a vulnerability is announced for a version of the code that is out of date.

Another way to potentially address the integrity and authenticity question is establishing a requirement for organizations to digitally sign the entire SBOM. This ensures the reliability and integrity of a completed SBOM upon receipt by a consumer.

Finally, we believe that third-parties in receipt of SBOM should ensure that they securely store the SBOM with modern ciphers for confidentiality and integrity protection and that third-parties in receipt of SBOM should maintain secure, controlled access to it according to contractual agreement.

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?*

It is our view that SBOM will be susceptible to compromise due to a supply chain attack. The incentive for developers is to ensure the integrity of the software over SBOM and forcing otherwise is likely to result in ill-advised investment. It is far preferable that a company invests in building and testing better systems rather than on establishing SBOM chains.

NTIA, and the USG more broadly, should be careful not to make SBOM the only mechanism for assurance, supply chain security, incident management, etc. Indeed, while an SBOM may provide helpful data, it will not describe who, how, and when a particular component was compromised, nor will it provide the precise exploit code (in a standardized format) that the attacker is embedding in the corrupted component.

The question that NTIA and the USG should consider instead is who benefits from more information, and what is the cost to obtain it? Cost considerations are essential. The level of effort to implement robust SBOM capabilties, especially for continuous deployment models common in modern software development, will not be small and may distract software providers from implementing other software and build integrity practices that will more directly address the core issues that necessitiated the EO in the first place.

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 14028.13 How can SBOM data be integrated with this additional data in a modular fashion?

Trying to make the SBOM the "one stop shop" for all security questions is not appropriate or cost effective, especially as different elements of an SBOM may have differing security attention (or security assurance efforts) paid to them. For example, will a cryptographic library, which is an important software component, benefit from a FIPS-140 certification or an SBOM? A FIPS-140 certification will include verification of the software sub-components of the cryptographic libraries in addition to their security design and implementation, while an SBOM won't provide any such information.

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.

It is important to balance the above considerations. While too many options may inundate SBOM producers or consumers, it may make sense to provide a few accepted methods, while preserving flexibility for adoption of future approaches. We agree that too many methods will result in more fragmentation and decrease usage. We discussed delivery at the outset of our paper and continue to encourage further consideration of this element via a discrete stakeholder feedback process. However, to reiterate some of the points made above, we believe that the SBOM should be delivered in a controlled manner with limited distribution to only trusted third-parties according to contractual agreement, including confidentiality and integrity protection and mutual authentication during the transfer process. Third-parties in receipt of the SBOM should be required to securely store the SBOM with modern ciphers for confidentiality and integrity protection. It should not be publicly disclosed.

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.

We recommend that software developers be allowed to produce an SBOM that lists their direct inclusions and not be required to use tools they are not already using. If they are so inclined, they can also include downstream components, but those should be listed as non-authoritative and for convenience only. Developers could potentially also link to the SBOM for the downstream components, which could either take the form of a direct link to the most recent SBOM or the version that is included.

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.

The RFC notes that the proposed use cases for SBOM focus on known vulnerabilities, which rely on frameworks such as the Common Vulnerabilities and Exposures (CVE) program to enable organizations to identify and report vulnerabilities in a consistent way. While some value may come from providing vulnerability information, context is important to consider. Vendors need to be able to identify that components containing known CVEs are not vulnerable in the context they are used. How a specific component is used, which features are enabled, and how the component interacts with the rest of the system is often essential to assessing the impact of specific vulnerabilities. Yet this information will not be and cannot be expressed through SBOM. Users will thus have to continue to rely on and trust the product vendor to provide an accurate vulnerability impact assessment for components.

Second, not all vendors have the same business model or the same mechanisms to provide information about vulnerabilities in software. It thus makes little sense to "standardize" this information by mandating that CVEs are referenced as a minimum element in an SBOM in all instances. This is especially the case given that vendors do not necessarily fix all vulnerabilities in all versions of code that they produce (meaning code written in-house as opposed to third party inclusions) - though in some contexts referencing CVEs may be appropriate. In some cases, this is because issues cannot be backported, as the "fix" requires an architectural change or a fix (e.g., in a hardware device) exceeds the memory capacity of the device and upgrading would brick the device. Such a result could be exacerbated in cloud services contexts where there may be even faster fixing, so the effort to include "vulnerability information" is quickly out of date as the remediation is rapidly pushed out.

Third, NTIA should consider that including CVEs as a minimum element may serve to drown users with information about low-risk vulnerabilities, or vulnerabilities that are not vulnerable in the context they are used, flooding out areas of actual critical concern and diverting resources away from addressing those critical areas.

An internal-only SBOM is an industry best practice and can increase the efficiency of a Product Security Incident Response Team (PSIRT) function within an organization. A minimum set of SBOM elements could enable faster identification of potentially impacted software components when handling vulnerabilities, particularly when used in conjunction with other processes.

In sum, SBOM may be used to track down a potentially vulnerable component when a new vulnerability is discovered, which will aid in the patching process. However, SBOM cannot be the source of truth for vulnerability information relating to a software product since (1) not all known vulnerabilities are relevant even if a vulnerable component is used; and (2) an SBOM is unlikely to be complete in capturing everything, including the level of dependencies known to the product developer and third-party suppliers.

***

We appreciate the opportunity to provide comments to NTIA on minimum elements and additional considerations related to SBOM. While there is some value that may be derived from SBOM, we also urge NTIA to consider our inputs related to depth, the limitations of SBOM due to versioning and software identification issues, the importance of context when it comes to vulnerability information, and the level of effort and resourcing that will be necessary for companies to prepare SBOMs. We welcome the opportunity to engage in additional conversations with NTIA and look forward to continuing to provide input and feedback on this issue as NIST leverages NTIA's work to develop related guidelines pursuant to section 4 of the EO.

Sincerely,

John S. Miller
Senior Vice President of Policy
and General Counsel

Courtney Lang
Director of Policy