Enterprise Cloud Coalition EC2

June 17, 2021

Ms. Evelyn L. Remaley
Acting Administrator
National Telecommunications and Information Administration
U.S. Department of Commerce
1401 Constitution Avenue NW
Room 4725
Washington, DC 20230
Submitted via email to: SBOM_RFC@ntia.gov

Dear Ms. Remaley:

On behalf of the Enterprise Cloud Coalition (ECC), it is our pleasure to respond to the National Telecommunications and Information Administration (NTIA) Software Bill of Materials (SBOM) Elements and Considerations request for public comment, Docket No. 210527–0117, issued on June 2, 2021.

Background:

By way of background, the ECC is a group of enterprise cloud companies that are united by a common business model and shared policy concerns. Our main objective is to educate policymakers about cloud computing, including the underlying technology, and how cloud computing both promotes innovation and benefits enterprises and their customers. We focus on four key policy areas for our federal government engagement – cross-border data flows, privacy, artificial intelligence, and cybersecurity.

Answers to NTIA Questions 1 and 3

It is ECC's pleasure to provide comments on questions 1 and 3, which are most applicable to the enterprise cloud computing community.

Question 1. Are the elements described above [see the full NTIA post], including data fields, operational considerations, and support for automation, sufficient? What other elements should be considered and why?

ECC response:

In order to most easily and effectively categorize and understand use of SBOM elements for federal agencies, NTIA should consider having a managed database of SBOM objects and versions of those objects. By doing so, software developers could assert, for example, "we use open ssl 1.1.1h," and federal agencies that need to could refer to that database. In turn, the OpenSSL project would host the SBOM file for all of the versions of OpenSSL, including 1.1.1h, providing easy reference for software publishers.

Additionally, as part of its SBOM elements, NTIA should provide additional detail on the use of a cryptographic hash. Examples of, and possibly enumeration of, the different types of hashes would enable them to be compared to one another, so that the vendor would not need to repetitively include all of the components.

Finally, "URL" does not appear to be a predefined identifier that would be included in the SBOM, and ECC members would encourage NTIA to consider this notion. A single, canonical URL for each version of software (like a tagged GitHub branch) would seem to be a benefit for consumers of an SBOM.

Question 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.

ECC response:
A common way software is disambiguated is by linking to the source control as well as the Git commit ID. While this is not ubiquitous, it is fairly common. Having a dependency-tracking database that is specific to SBOM might also be helpful, similar to the National Institute for Standards and Technology (NIST) National Vulnerability Database (NVD).

Question 3
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.

ECC response:
There should be some convention of a "last update" field in SBOM to ensure that the latest version is available, but old revisions should be available, possibly with a "valid from" and "valid to" indication in past versions (there may be some overlap when a software rollout occurs). The history is important to maintain in order to determine the impact of old versions, particularly if a vulnerability is announced for a version of the code that is out of date. Another way to verify the source of the SBOM date and its integrity would be a centralized repository where SBOMs can be submitted by verified parties.

Question 3
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?

ECC response:
ECC members believe that SBOM will be susceptible to compromise via a supply chain attack. At the same time, the incentive for software developers is to ensure the integrity of the software, and forcing otherwise is likely to result in ill-advised investment. ECC members firmly believe that better and more secure software is the result of investing in better build systems and testing systems. NTIA's SBOM efforts, and the efforts of all federal agencies involved in implementation of the software assurance

sections of Executive Order 14028, "Improving the Nation's Cybersecurity," should take this into account.

Question 3
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.

ECC response:
ECC members believe that multiple methods should be provided for SBOM discovery, including standard file location in a Git repository, centralized database, or possibly a URL that is communicated to consumers as part of a federal contract or customer portal. At the same time, too many methods for SBOM discovery will likely result in more fragmentation and less use. Therefore, any methods that are proposed should enable things like API support, which allow end-users to locate the files easily.

Question 3
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.

ECC response:
Software developers should produce SBOMs that list their direct inclusions. It should be an option for developers to include downstream components. And if developers choose to do so, those downstream components should be noted as non-authoritative and just for the convenience of the individual conducting SBOM discovery. Developers could potentially also provide a link to the SBOM for the downstream components, which would either be a direct link to the most recent SBOM or the version that is included. A standardized versioning system would be helpful for API consumers.

Additionally, while there is no specific request for comments on the issue of federal agency user access to the SBOM, ECC members believe that NTIA should support a policy by which access to SBOMs is limited to only those who must have visibility. While SBOMs are generally helpful to improving cybersecurity, tightly controlling access to this information is critical so that this itemized list of components is not then used by attackers to breach components or systems.

Thank you very much for this opportunity to contribute to NTIA's important work in improving the state of federal cybersecurity. We look forward to being part of this dialogue going forward and stand at the ready to answer any questions you might have.

Andrew Howell
Enterprise Cloud Coalition
andrew@enterprisecloudcoalition.org
https://www.enterprisecloudcoalition.org/