Ke, Jessica - Intern

From: Eric Byres <eric.byres@adolus.com>
Sent: Thursday, June 17, 2021 11:25 PM
To: SBOM_RFC
Subject: Comments by aDolus Technology Inc on Software Bill of Materials Elements and Considerations in response to document NTIA-2021-0001

Introduction

aDolus Technology Inc. (aDolus) welcomes the opportunity to respond to the Request for Comment (RFC) by the National Telecommunications and Information Administration (NTIA) on “Software Bill of Materials Elements and Considerations” published in the Federal Register on June 02, 2021 regarding Docket # 210527-0117, document identifier NTIA-2021-0001.

aDolus commends the Department of Commerce (DoC) and the NTIA for leading an open and inclusive stakeholder process on the creation and effective use of Software Bill of Materials (SBOMs) in a way that advances the security of the nation and the world’s software supply chain. The NTIA’s foresight in setting up this process in 2018 is allowing industry and government to respond to the recent global and domestic supply chain developments in a timely manner.

Comments

Section 3(c) Legacy and binary-only software

Comment: SBOMs generated from binary-only software should be considered of equivalent importance as SBOMs generated from source code.

Rationale: As the RFC correctly notes, in many cases source code may not be obtainable, with only the object code available for SBOM generation. This is especially true in Operational Technology (OT), where industrial control system (ICS) equipment may have an expected life span of 25 to 30 years (1). In these cases, SBOMs generated from object code are the only choice for an asset owner wishing to manage their security risks. However, defining this type of SBOM as the "second class citizen of the SBOM world" could lead suppliers to justify not offering SBOMs to their clients, instead demanding costly upgrades to new software or platforms just to get an SBOM. In turn, asset owners, fearing that they could be caught in an upgrade squeeze, could resist the use of SBOMs in security and audit processes.

This is not to say that object code and source code SBOMs will be equivalent. Both have their strengths and weaknesses. While it is likely that source code SBOMs may have additional information available, object code SBOMs show what was actually installed in a device versus what the development build system believed would be installed by the user. Ideally both object code and source code SBOMs should be required, with a comparison process between these two information sources to provide as complete and accurate a picture of the deployed (and thus at-risk) software in a system as possible.

Section 3(i) Vulnerabilities

Comment: Vulnerability data should not be included in the SBOM itself.

Rationale: The core use case for an SBOM is a list of software components at a fixed point in time, such as when the software was built. For regulations that have a defined audit component, such as NERC CIP-013, it must be possible to prove that an SBOM has not been modified if it is to have evidentiary value.

In contrast to the components in a software package, the existence and status of vulnerabilities can change over time, often at a rapid pace. As a result, by including vulnerabilities within the SBOM, multiple versions of a specific product’s SBOM would be generated as new vulnerabilities are discovered. This in turn would require attestation processes that are not provable by simple cryptographic hash matching between the SBOM and the software. Instead some sort of SBOM version management would be required, needlessly complicating the audit process.

Furthermore, while vulnerability management is an important use case for SBOMs, it is by no means the only one. For use cases not related to vulnerability management, the SBOM (which already can be very large for real world software) could contain significant amounts of vulnerability data that has no function. Should the proponents of these other use cases also demand that their data be included in every SBOM, the syntax of the

SBOM is likely to become increasingly complex, multiplying the opportunities for errors and the need for frequent revisions.
Finally, the disclosure policies and timelines for vulnerabilities may not be identical to those for SBOMs. Mixing vulnerability and component information could result in software suppliers resisting timely sharing of core SBOM information so as to align with their vulnerability disclosure timeline. Similarly, customers of certain mission critical systems who might not be privy to detailed vulnerability information may find themselves excluded from SBOM information as a result of the blending of vulnerabilities and SBOMs.

Conclusion

We hope these comments provide useful input into the development of the minimum elements for an SBOM. We look forward to further opportunities to provide input into this process. If you have any questions or would like additional information, please contact Eric Byres at eric.byres@adolus.com or Derek Kruszewski at derek.kruszewski@adolus.com.

References

(1) NIST Special Publication 800-82 Revision 2 - Guide To Industrial Control Systems (ICS) Security, Page 6-6
http://dx.doi.org/10.6028/NIST.SP.800-82r2

Regards,
Eric

Eric Byres, P.Eng, ISA Fellow
CTO, aDolus Technology Inc.
+1-604-897-9980 www.adolus.com
adolus

Need help unravelling the Executive Order on Cybersecurity? Try our handy EO14028 Timeline