# Ke, Jessica - Intern

**From:** Scott Snowden <scott.snowden@whitesourcesoftware.com>
**Sent:** Thursday, June 17, 2021 4:00 PM
**To:** SBOM_RFC
**Cc:** Rhys Arkins; Friedman, Allan
**Subject:** Comments on Software Bill of Materials Elements and Considerations

Allan,

The comments below are WhiteSource Software's response to the request for comments published in Federal Register / Vol. 86, No. 104 / Wednesday, June 2, 2021 related to Software Bill of Materials Elements and Considerations. They are organized by question and sub-section as appears on page 29570.

Point of contact for further communications

Scott Snowden – Enterprise Sales Manager (scott.snowden@whitesourcesoftware.com)
Rhys Arkins – Director of Product (rhys.arkins@whitesourcesoftware.com)

Comments

1. While the current text is pragmatic, we are concerned that the importance of "transitive" dependencies is underestimated. As it pertains to risk and composition, there is no difference between a direct dependency and a transitive dependency. Both are 3rd party components that exist in the application. Transitive dependencies - particularly open source ones - are often 60-80% of software lines of code, and therefore cannot be demoted to ideal or nice-to-have. We suggest that the guidelines require manufacturers to include transitive dependencies. If transitive dependencies are unable to be included the SBOM should explicitly declare this gap and outline what risk additional risk exists as a result.

Regarding data fields, the license type of 3rd party components should be required. As an SBOM is intended to illustrate the "relationships of various components," knowing the license type of components would allow a purchaser to know if the software can be used as a component of another application without creating legal risk.

2. None

3.(i) The vulnerabilities associated with an SBOM are dynamic and can change (for the worse) day by day. Unless the SBOM exchange is somehow itself a living document, it would be unwise to embed any vulnerability list as it would create a false sense of security and would become inaccurate over time. A better approach would be to ensure that the components are identified in a consistent, well-defined manner. This will facilitate easily linking to public vulnerability data for monitoring. (This challenge is briefly touched on in section 3.a)

3.(j) Similar to (i) above, care should be taken not to produce static risk assessments that disguise risks that develop later. If a mechanism is added to allow suppliers to identify CVEs as "not effective," a standard list of options should exist to indicate why the supplier believes the CVE is not relevant. Options for this list might include: vulnerable component not called by software, internal compensating control, external compensating control, etc. A separate notes field should also be available to provide additional detail.

V/r

**Scott Snowden – CISSP, CSSLP, CEH**

Enterprise Sales Manager
WhiteSourceSoftware.com

**Use Open Source Freely and Fearlessly.**