

# **Software Assurance Tips**

A product of the Software Assurance Tips Team[2]

Jon Hood

Monday 5<sup>th</sup> July, 2021

# 1 A Pedigree of S-BOMs

Updated Wednesday 11<sup>th</sup> February, 2026

**NOTE:** As of April, 2022, the DoD has released CCIs for NIST 800-53rev5, including SR-4(4) CCI-5110: “Conduct organization-defined analysis to ensure the integrity of the system and system components by validating the internal composition and provenance of critical or mission essential technologies, products, and services.” I plan to release a new version of this article after SPDX 3.0 is released.

Recently, a large software project encountered a serious vulnerability. The project has been in existence since the early 1990s and consists of millions of lines of code. On May 20<sup>th</sup>, CISA issued an advisory that resulted in several CVEs being created for some of the older RTOSes.[1] When we marked this as a critical finding against the legacy system, the response we received was, “I didn’t even know that was in there!”

## 1.1 Establishing a Pedigree

While the DoD is still proceeding with its RMF implementation of NIST 800-53 Revision 4, Revision 5 of the RMF controls creates a new family: SR - Supply Chain Risk Management. One of the new controls is SR-4, related to the provenance of the supply chain. To date, no categorization baseline requires the implementation of SR-4; however, it should be anticipated that this control will be tailored in for high-importance and tactical systems in the future.

Part of SR-4 is enhancement SR-4(4) which requires the establishment of provenance and pedigree by keeping up with the internal composition of software and hardware components. “For software this includes the composition of open-source and proprietary code, including the version of the component at a given point in time.”[3, p. 66] This is a step above the hardware and software lists currently implemented in RMF; the program must manage the composition at a more granular level. Claiming not to know that a component includes a dependency would be a failure against this control.

## 1.2 Compliance

For software, I recommend implementing Software Package Data Exchange (SPDX) and a Software Bill of Materials (S-BOM). A compliant policy would include a statement like the following:

Our organization requires that each main software delivery must define a Software Package Data Exchange (SPDX) file and Software Bill of Materials (S-BOM) detailing all first-order dependencies.[4] The SPDX file, at a minimum, must include the `PackageName` and `PackageLicenseDeclared`. If they are available, `PackageOriginator` and `PackageHomePage` must also be provided. Each main delivery product will also provide an S-BOM consisting of an SPDX file for each first-order dependency when that first order dependency properly manages its own dependencies. For example, if Product A is the deliverable and it depends on Product B, Product B is a first-order dependency. If Product B depends on Product C, Product C is a second-order dependency and only requires documentation in the S-BOM if it is also a first-order dependency or if Product B requires Product A to manage its dependencies.

For every major release of the deliverable or every three years (whichever occurs first), a Software Composition Analysis is performed, and the S-BOM’s SPDX files are compared to the results. The software composition analysis may be conducted automatically as part of the CI/CD pipeline or in our Software Assurance assessments. Undocumented dependencies are triaged as security concerns in our issue tracking system.

Such a policy requires listings of dependencies with their version numbers, encourages developers to automate composition analysis, documents the POCs for each dependency, and records the license restrictions of each component.

### **1.3 Recommendations**

The Application Security and Development Security Technical Implementation Guide (STIG) should be updated to check that dependencies are documented appropriately. There is not currently a STIG requirement to document and manage dependencies.

When issuing Control Correlation Identifiers (CCI) against SR-4(4), DISA should divide this into at least two checks: one for documenting the composition correctly, and the other for verifying the integrity and correctness of that composition.

## References

- [1] US-CERT. ICS Advisory (ICSA-21-119-04): Multiple RTOS (Update B). CISA. May 20, 2021. URL: <https://us-cert.cisa.gov/ics/advisories/icsa-21-119-04> (visited on 06/23/2021).
- [2] Jon Hood, ed. SwATips. <https://www.SwATips.com/>.
- [3] National Institute of Standards and Technology. Security and Privacy Controls for Information Systems and Organizations. Tech. rep. Special Publication (SP) 800-53 Revision 5. Washington, D.C.: U.S. Department of Commerce, 2020. DOI: 10.6028/NIST.SP.800-53r5. URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf>.
- [4] David A. Wheeler. SPDX Tutorial. URL: <https://github.com/david-a-wheeler/spdx-tutorial/blob/cee3cbe7ae5f83ec478e2acf2c9282eafd42ff0f/README.md> (visited on 06/23/2021).