Position Statement on

Area 5: Guidelines for software integrity chains and provenance and

Area 4: Initial minimum requirements for testing software source code

Requirements for **Standardizing Software Bills of Materials**

Author: Brian Fox (brianf@sonatype.com)
Title: CTO and Co-founder, Sonatype; member of the Apache Software Foundation and former Chair of the Apache Maven project
Interested in speaking: Yes

Background

The Executive Order on Improving the Nation's Cybersecurity acknowledges and encourages the acceleration of the federal government's adoption of cloud computing services and simultaneously directs that suppliers of software to the federal government should improve transparency by providing government customers with a software bill of materials.

Whether "on premise" or "in the cloud", the process by which applications are developed, secured, and operated closely resembles a digital supply chain that will benefit from improved transparency. Such transparency is easily achieved when vendors and customers are aligned with respect to a standard process for generating and sharing a software bill of materials (SBOM).

A SBOM in the context of a **traditional on-premises software** is static. It need only be updated when new software versions are delivered and deployed, which is an infrequent process due to the overhead of upgrading on-premises software.

A SBOM in the **context of cloud native software** which is constantly evolving due to many frequent releases and updates, presents audit, compliance, and risk management challenges. However, modern software composition analysis tools have largely solved these challenges.

In this paper I describe recommended minimum requirements for tools, tool usage, and data formatting so that vendors selling both on-premise and cloud-native applications to federal government customers can also deliver a standard SBOM in compliance with the Executive Order.

Standardized SBOM Formats

Even in on-premise deployment scenarios, machine-readable SBOM formats documenting third-party dependencies provide substantial value. They can be fed into databases, incorporated into audit logs, and compared against the ever-evolving set of known software vulnerabilities. With the increased pace of development and delivery in cloud environments, standardized machine-readable SBOM formats are critical.

There are several SBOM formats available, including SWID, SPDX, and CycloneDX. Among these, CycloneDX offers distinct advantages that are important for meeting the goals of the Executive Order. These include:

- Standard inclusion of metadata, components, services, dependencies, and compositions.
- Metadata that consists of the supplier, manufacturer, the target component the SBOM describes, the tools used to create the BOM, and license information
- The ability to describe components and their dependency on other components
- Ability to be represented as XML, JSON and Protocol Buffers, making it easy to use
- Can be made more complex as needed, but as a standard is quite simple and provides easy interoperability

SBOM Scope

A software bill of materials traditionally focuses on a particular application, listing the third-party components that the application makes use of. However, cloud applications as well as modern on-premise applications are increasingly being delivered using containers or virtualization technology, which packages an application together with its execution environment. In such cases, knowledge of this environment is an important part of understanding the risk surface of the application. Even with non-containerized solutions, cloud-based SaaS applications generally do not disclose the infrastructure, environment, and configurations involved in their deployment. While there is no standardized method of reporting these aspects of software composition, and so it would be difficult to define minimal standards, this should be an area of investigation and research for the community going forward.

Continuous Compliance

Cloud-based deployments often go hand-in-hand with software development processes based around continuous deployment. When new versions of software are being deployed every day or multiple times per day, automated audit and compliance processes become a necessity. With continuous deployment, the software development and deployment process has to include automated software composition analysis. We recommend that teams move to a *continuous compliance* process, whereby automated tools that are integrated into development processes warn developers when vulnerable dependencies are added or when existing dependencies become subject to newly-discovered vulnerabilities. This aligns with the trend toward automated testing, packaging, and deployment and it solves the problem of how the government can have trust in continuously-evolving software. It is not too strong a statement to say that such automation is the only way the goals of the Executive Order can be realized in a fully cloud-based and continuously-deployed setting.

Conclusion

The Executive Order says, “Understanding the supply chain of software, obtaining an SBOM, and using it to analyze known vulnerabilities are crucial in managing risk.” An SBOM is only half the battle. Standardizing on a machine-readable SBOM format that includes compositions and dependency relationships is vital to success of the Executive Order actually aiding in risk management. When a vendor gives the government a software bill of materials, they will have to trust it because they can’t verify that it is accurate. To create trust, we need transparency, standardization and automation.