5. Integrity Chains

David A. Wheeler <dwheeler@linuxfoundation.org> (et al)
Director of Open Source Supply Chain Security, The Linux Foundation (*willing to speak*)

> **NIST Request:** *Guidelines for software integrity chains and provenance*. See EO Sections 4(e)(ii, vi, and viii).

Open source software (OSS) is dominant in software supply chains; one study found that on average 70% of codebases are OSS. Closed source software can often be inventoried by looking at contracts and subcontracts. In contrast, OSS procurement is usually frictionless and does not go through traditional procurement practices. Therefore OSS has been a primary driver for many organizations' focus on best practices for supply chain integrity.

The executive order (EO) rightly places a heavy emphasis on software bill of materials (SBOMs). SPDX is an SBOM specification that can provide an inventory of all the components within a larger piece of software, including metadata such as the licenses of those components. The LF has been working with a diverse world wide community in developing and refining SPDX for over ten years; SPDX is used worldwide and the working draft has been made available by ISO at ISO/IEC JTC 1 Draft International Standard (DIS) 5962. SPDX 2.2 already supports the guidance to date from the National Telecommunications and Information Administration (NTIA) for minimum SBOM elements. Some ecosystems have domain-specific conventions for SBOM information, but SPDX can provide information across all arbitrary ecosystems. SPDX is adopted in production use for companies managing global supply chains today. It has also been adopted by open source projects. For example:

- The first NTIA "plugfest" demonstrated ten different producers generating SPDX, and all of the consumer tools were consuming SPDX. SPDX supports acquiring data from different sources and cross referencing multiple SBOM documents.
- A corpus of some LF projects with SPDX source SBOMs is available.
- Various LF projects are working to generate binary SBOMs as part of their builds, including yocto and Zephyr.
- The Automated Compliance Tooling (ACT) Umbrella of OSS projects produce, consume and make it easier for other tools to work with SPDX documents. Current LF projects that are part of ACT are: FOSSology (source code analysis), OSS Review Toolkit (generation of SBOMs in CI & Build infrastructure), Tern (container content analysis), Quartermaster (build extensions), and the SPDX-tools libraries.
- To assist with further SPDX adoption, the LF is developing SPDX plugins for major languages and open source package managers.

SBOMs in SPDX format can enable customers to determine if there are known vulnerabilities in the software they use, at the time of adoption into their systems, or later when a new vulnerability is disclosed. For this to work, however, customers must actually *receive* complete SBOMs in a standard format they can automatically process.

Here are some places to look for best practices & tools for integrity chains:

1. The OpenSSF's CII Best Practices badge project specifically identifies best practices for Open Source Software (OSS), focusing on security and including criteria to evaluate the security practices of developers and suppliers (it has over 3,800 participating projects).
2. OpenChain (ISO 5230) is the International Standard for open source license compliance. Application of OpenChain requires the identification of OSS components, and that information aids other analyses of components (e.g., to look for known vulnerabilities).
3. Supply-chain levels for Software Artifacts (SLSA) is a framework being developed for ensuring software artifacts meet certain minimum end-to-end integrity standards to secure a supply chain.
4. The Linux Foundation (LF) has developed and released its Secure Software Development Fundamentals set of courses available on edX.org to anyone at no cost, and the source material is available in markdown format under an open source license so anyone can reuse it and incorporate it into their own materials.
5. The OpenSSF Best Practices Working Group (WG) actively works to identify and promulgate security best practices. We also provide a number of specific standards, tools, and best practices, as discussed below.
6. The cloud computing industry collaborating within Cloud Native Computing Foundation (CNCF) has also produced a guide for supporting software supply chain best practices for cloud systems and applications.
7. in-toto is a framework developed by the CNCF community to secure the integrity of software supply chains. The original research that produced in-toto was supported by grants from the US National Science Foundation (NSF), the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL).
8. The Update Framework (TUF) helps developers maintain the security of software update systems, and is used in production by various tech companies and open source organizations. Uptane is a variant of TUF to protect software delivered over-the-air to the computerized units of automobiles.
9. sigstore is a project to provide a non-profit service (as a public good) to improve the OSS supply chain. sigstore makes it easy for developers to adopt cryptographic software signing of artifacts (such as release files, SBOMs, and container images) backed by a transparent, tamper-resistant, and public log.
10. The LF is funding work on tools to ease signature and verify origins, e.g., (a) extending git to enable pluggable support for signatures and (b) improve the patatt tool to provide easy end-to-end cryptographic attestation for patches sent via email.
11. The Linux Foundation's Civil Infrastructure Platform community continues to invest in reproducible builds. Being able to reproduce an exact binary from source enables identifying compromised binaries (e.g., like the attack on SolarWinds). It is good to harden the build systems against attack, but it is unwise to assume that attacks can never succeed. The documentation from reproducible-builds.org can be helpful.

In addition, the US government should also modernize the process by which publicly-known vulnerabilities can be identified and tagged. The current CVE process is inadequate to address modern software workflows; it's too slow and cumbersome. The National Vulnerability Database (NVD) should likewise be modernized to include automatically-processable tags for all vulnerable software, including version ids and version ranges.