Laurie Williams
North Carolina State University

Narrowing the Software Supply Chain Attack Vectors

The SSDF Is Wonderful but not Enough

Recent years have shown increased cyberattacks targeting less secure elements in the software supply chain and causing fatal damage to businesses and organizations. Past well-known examples of software supply chain attacks are the SolarWinds or log4j incidents that have affected thousands of customers and businesses. In 2023, Sonatype[1] reported the detection of 245,000 malicious packages, double the number of malicious packages discovered in 2019–2022 combined.

The U.S. government is so concerned by software supply chain security deficiencies that a whole section of Executive Order (EO) 14028[2] (Improving the Nation's Cybersecurity) issued 12 May 2021 is focused on new compliance requirements for government vendors to enhance software supply chain security. The impacts of the EO have spread beyond direct vendors to the U.S. government to vendors who sell to these direct vendors, and to vendors who sell to vendors who sell to the U.S. government, and so on. Other countries and software developers around the world have been influenced as well.

Section 4 of EO 14028 is related to "Enhancing Software Supply Chain Security." In addition to initiating the production of industrial guidelines and definitions, EO Section 4 mandates that additional steps for securing the software supply be completed if an organization wants to sell to the U.S. government. The first major wave of the impact of these mandates was on producing a software bill of materials (SBOM), or a nested inventory, a list of ingredients that makeup software components. After two years and enormous attention from the industry, many tools to produce SBOM though these tools have been shown to produce widely different outputs for the same project, so more attention needs to be placed on SBOM tools. Organizations increasingly use SBOM tools to produce SBOMs and comply with the EO.

The next and current wave of EO-initiated activity is the delivery of a self-attestation document signed by the chief executive officer (CEO) or chief operating officer (COO) of the software producer, attesting that the software produced is developed in conformity with specified secure software development practices. These specified secure software development practices are laid out in the U.S. Department of Homeland Security/Cybersecurity and Infrastructure Security Agency (CISA) Secure Software Development Attestation Form.[3] Specifically, the CEO or CEO attests that their software development organization consistently uses the development practices derived from NIST SP 800-218, the Secure Software Development Framework (SSDF).[4] Each software development practice attested to is mapped to an SSDF task. These practices include the separation of development and build environments, logging and monitoring authorization and access, multifactor authentication, encrypting sensitive data, operational monitoring, the production of provenance, the use of automated tools to check for security vulnerabilities, a process for mitigating vulnerabilities, and operating a vulnerability disclosure program. Anecdotally, having a CEO or COO attest to the use of these practices is concerning and overwhelming for many organizations, both because the organizations must adopt new practices and because producing evidence that the practices have been followed is primarily manual. Opportunities

Digital Object Identifier 10.1109/MSEC.2024.3359798
Date of current version: 15 March 2024

 Copublished by the IEEE Computer and Reliability Societies 1540-7993/24©2024IEEE

abound for tools to automate and standardize the attestation process.

What's Wonderful About the EO 14028, the SSDF, and Self-Attestation?

Certainly, the EO and the resulting pressure to implement and attest to SSDF tasks may not feel wonderful to software organizations. The priority of a company is to deploy new features and products to customers to generate revenue as economically and efficiently as possible. Adding steps to the development process workflow and purchasing new vulnerability detection and SBOM-generation tools may seem detrimental to economics and efficiency.

Sometimes, real progress is made only when we need to overcome challenges, such as complying with the EO. Pre-EO, software organizations were making a tradeoff between dedicating resources to make a product more secure or deploying additional functionality. The tradeoff is related to the risk of an attack when the probability of attack and the impact of an attack is nebulous. With the EO, this same tradeoff relates to the risk of losing all business to the U.S. government with the probability of losing that business at 100%. The need to implement and attest to SSDF-defined software security tasks whereby the CEO or COO signs on the dotted line that the tasks were implemented is real and urgent.

Indeed, industry and government participants of three Software Supply Chain Summits[5] expressed excitement that the EO would force the industry into adopting security practices that should have been done 20 years earlier. And that is wonderful because a more secure software ecosystem and software supply chain is wonderful.

Where the SSDF is not Enough

EO compliance is a huge step toward a more secure software ecosystem and software supply chain. This step currently may feel overwhelming to many or even most organizations. But, closing down the novel software supply chain attack vectors requires additional tasks *not specified in the SSDF* though specified in other frameworks and standards. Some frameworks that specify beneficial tasks to address novel software supply chain attack vectors include the NIST 800-161r1 Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations (800-161),[6] the Cloud Native Computing Foundation Software Supply Chain Security Best Practices paper (SSCP),[7] Supply-chain Levels for Software Artifacts (SLSA),[8] and the Secure Supply Chain Consumption Framework (S3C2F).[9] Tasks references by these frameworks will be referenced next.

With the EO, this same tradeoff relates to the risk of losing all business to the U.S. government with the probability of losing that business at 100%.

Code Dependencies as an Attack Vector

Attackers have long exploited vulnerabilities accidentally injected into a component or product. In the more novel software supply chain attack vector, attackers intentionally inject vulnerabilities into upstream open source components such that they can be leveraged at scale to attack upstream projects. Modern software

IEEE Reliability Society

Executive Committee (Excom) Members: Steven Li, President; Jeffrey Voas, Sr. Past President; Preeti Chauhan, VP Technical Activities; Phil Laplante, VP Publications; Christian Hansen, VP Meetings and Conferences; Janet Lin, VP Membership; Loretta Arellano, Secretary; Jason Rupe, Treasurer

Administrative Committee (AdCom) Members: Loretta Arellano, Preeti Chauhan, Trevor Craney, Joanna F. DeFranco, Pierre Dersin, Ruizhi (Ricky) Gao, Louis J. Gullo, Christian Hansen, Pradeep Lall, Phillip A. Laplante, Janet Lin, George Pallis, Jason W. Rupe, Robert Stoddard, Daniel Sniezek, Scott Tamashiro, Eric Wong, Ruolin Zhou

http://rs.ieee.org

The IEEE Reliability Society (RS) is a technical Society within the IEEE, which is the world's leading professional association for the advancement of technology. The RS is engaged in the engineering disciplines of hardware, software, and human factors. Its focus on the broad aspects of reliability allows the RS to be seen as the IEEE Specialty Engineering organization. The IEEE Reliability Society is concerned with attaining and sustaining these design attributes throughout the total life cycle. The Reliability Society has the management, resources, and administrative and technical structures to develop and to provide technical information via publications, training, conferences, and technical library (IEEE Xplore) data to its members and the Specialty Engineering community. The IEEE Reliability Society has 28 chapters and members in 60 countries worldwide.

The Reliability Society is the IEEE professional society for Reliability Engineering, along with other Specialty Engineering disciplines. These disciplines are design engineering fields that apply scientific knowledge so that their specific attributes are designed into the system/product/device/process to assure that it will perform its intended function for the required duration within a given environment, including the ability to test and support it throughout its total life cycle. This is accomplished concurrently with other design disciplines by contributing to the planning and selection of the system architecture, design implementation, materials, processes, and components; followed by verifying the selections made by thorough analysis and test and then sustainment.

Visit the IEEE Reliability Society website as it is the gateway to the many resources that the RS makes available to its members and others interested in the broad aspects of Reliability and Specialty Engineering.

IEEE

Digital Object Identifier 10.1109/MSEC.2024.3359797

products commonly have tens to hundreds of direct and transitive code dependencies. Malicious code dependencies have become increasingly common due to typo-squatting, dependency confusion, and project take-over attacks.

In addition to the SSDF, tasks specified by other frameworks can aid in closing down this component attack vector. In "Securing the Software Supply Chain," CISA[10] recommends a secure repository process flow. This flow begins with a developer selecting a component to download, the component being scanned for vulnerabilities in an intermediate secure repository, and the component being moved to a secure repository if no issues are found. The developer can then download and build with the component from that localized secure repository. The components in the secure repository are *continuously* scanned to detect new vulnerabilities.

Some tasks not explicitly stated in the SSDF that can aid in closing down the dependency attack vector relate to managing component and container choices, managing vulnerable components, and verifying dependencies and the environment during build. Tasks that can aid in making informed third-party component and container choices are 800-161 CM7, S2C2F ING-3, and SSCP SM. Tasks S2C2F ING-1 and SSCP V specify that candidate packages and containers are obtained from trusted ecosystems (such as nmp or Maven) or are rebuilt and that organizations should require signed commits such that legitimate components software that has not been tampered with are used. Additionally, task S2C2F AUD-2 specifies that checks should be in place to ensure developers are not bypassing the component vetting process. The EO and SSDF specify that SBOMs be produced, but not necessarily consumed. Tasks 800-161 SR-4 and SCP SM specify that organizations consume SBOM information to react to security incidents and to identify which components need to be updated or patched.

Malicious code dependencies have become increasingly common due to typo-squatting, dependency confusion, and project take-over attacks.

The Build Infrastructure as an Attack Vector

The process and tooling that turns the code of multiple software projects into the production software product is just as important as the code in the software projects. This importance was highlighted with the December 2020 SolarWinds supply chain attack, where the build process was compromised to inject malicious code into the end product. Unfortunately, build systems have seen relatively little attention compared to software analysis. Tasks from other frameworks can be added to those specified in the SSDF to narrow this novel supply chain attack vector by safeguarding build integrity.

Several tasks can protect from and detect malicious infiltration into software build infrastructure that could lead to the build and deployment of compromised products. Tasks in SLSA, S2C2F AUD-1 through AUD-4, and SSCP V and B ensure the build environment's sources and dependencies come from a secure, trusted source of truth and that provenance exists. Tasks S2C2F REB-1 and SSCP BV specify the defensive use of the compiler and interpreter and build tool features to detect vulnerabilities. CNCF BA specifies that the build pipeline should be a series of hardened build steps implemented through a hardened container image stored within a secured repository and deployed through a hardened orchestration platform. Tasks S2C2F REB-1 and SSCP specify the use of reproducible builds to provide a mechanism to confirm that no malicious backdoor injections have taken place during the build process. Finally, SSCP CD protects the integrity of the build output by specifying that the build output is stored in a different location from the input files.

Compliance with the EO will move the industry forward, resulting in a more secure software environment and software supply chain. However, implementing tasks from other frameworks in addition to the SSDF guide focused efforts toward narrowing novel software supply chain attack vectors. ■

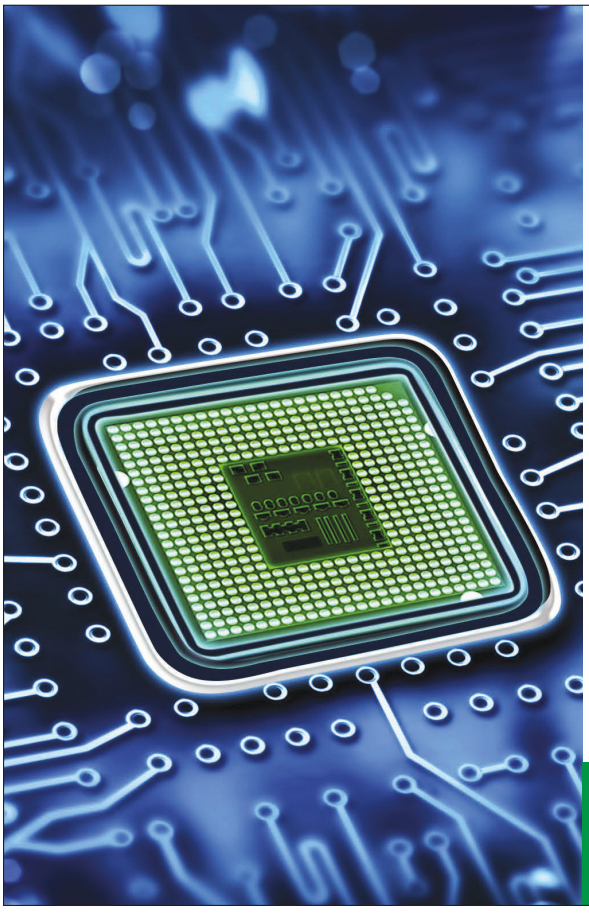
Acknowledgment

The detailed work to formulate these views was done during my sabbatical at Synopsys. My primary collaborators during my sabbatical project were Jamie Boote and Sammy Migues as well as Chris Madden, DJ Schleen, and Robert Hines from Yahoo. I also appreciate the guidance I received from Karen Scarfone from NIST. This work was partially supported and funded by the National Science Foundation under Grant 2207008. Any opinions expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. "Sonatype's 9th annual state of the software supply chain report reveals ways to improve developer, DevSecOps

efficiency." Sonatype. Accessed: Jan. 15, 2024. [Online]. Available: https://www.sonatype.com/press-releases/sonatype-9th-annual-state-of-the-software-supply-chain-report
2. "Executive order 14028: Improving the nation's cybersecurity." Federal Register. Accessed: Jan. 15, 2024. [Online]. Available: https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity
3. "Secure software development attestation form (Draft)," Cybersecurity and Infrastructure Security Agency, Arlington, VA, USA, 2023. [Online]. Available: https://www.cisa.gov/sites/default/files/2023-11/Secure%20Software%20Development%20Attestation%20Form_508c.pdf
4. "Secure software development framework (SSDF) version 1.1: Recommendations for mitigating the risk of software vulnerabilities," NIST, Gaithersburg, MD, USA, NIST SP 800-218, Feb. 2022. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/218/final
5. W. Enck and L. Williams, "Top five challenges in software supply chain security: Observations from 30 industry and government organizations," *IEEE Security Privacy*, vol. 20, no. 2, pp. 96–100, Mar./Apr. 2022, doi: 10.1109/MSEC.2022.3142338.
6. "Cybersecurity supply chain risk management practices for systems and organizations," NIST, Gaithersburg, MD, USA, NIST SP 800-161 Rev. 1, May 2022. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/161/r1/final
7. "Cloud native computing foundation: Software supply chain best practices." GitHub. Accessed: Jan. 15, 2024. [Online]. Available: https://github.com/cncf/tag-security/blob/main/supply-chain-security/supply-chain-security-paper/CNCF_SSCP_v1.pdf
8. OpenSSF. "Supply-chain levels for software artifacts v1.0." SLSA. Accessed: Jan. 15, 2024. [Online]. Available: https://slsa.dev/
9. OpenSSF. "Secure supply chain consumption framework." GitHub. Accessed: Jan. 15, 2024. [Online]. Available: https://github.com/ossf/s2c2f
10. "Securing the software supply chain: Recommended practices for developers," Cybersecurity and Infrastructure Security Agency, Arlington, VA, USA, Aug. 2022. [Online]. Available: https://www.cisa.gov/sites/default/files/publications/ESF_SECURING_THE_SOFTWARE_SUPPLY_CHAIN_DEVELOPERS.PDF



IEEE TRANSACTIONS ON

COMPUTERS

Call for Papers:
IEEE Transactions on Computers

Publish your work in the IEEE Computer Society's flagship journal, *IEEE Transactions on Computers*. The journal seeks papers on everything from computer architecture and software systems to machine learning and quantum computing.

Learn about calls for papers and submission details at
www.computer.org/tc.

IEEE COMPUTER SOCIETY

IEEE

Digital Object Identifier 10.1109/MSEC.2024.3369425