Software Supply Chain Security

for root, subdirs, files in os.walk(file_name):
for file_name in files:
if file_name.endswith(".java"):
path = os.path.join(root, file_name)
SAST/DAST analyzers (SonarQube, SporBugs, etc.)
Delta evaluation

Fabio Massacci | University of Trento and Vrije Universiteit Amsterdam
Laurie Williams | North Carolina State University

Today, the Ancient Mariner would rhyme "Code, code every where, Not any drop to trust." This special issue of *IEEE Security & Privacy* highlights software supply chain security research and experiences of value to practitioners and researchers alike.

The modern world relies on digital innovation in almost every human endeavor and for our critical infrastructures. Among the key enablers, software libraries stand apart. It is fair to say that modern software cannot be built without the layers of reusable abstractions, including libraries, frameworks, and cloud infrastructures. Yet, these very libraries often lie outside an organization's trust boundary.[1] Leveraging these reusable abstractions gives rise to software supply chains where software products include "upstream" components, as well as dependencies, created and modified by others, which again often include their own transitive dependencies. Most of these dependencies are open source projects. However, with all the power that software supply chains and open source infrastructure provide also come risks.[2] They are the giant built on the giant, built on the giant ... and fifty stacked giants below, almost invisible, lay the feet of clay.

Modern software cannot be built without the layers of reusable abstractions, including libraries, frameworks, and cloud infrastructures.

The 2022 annual report from Sonatype shows an average 742% annual increase in software supply chain attacks over the past three years. The impact of these attacks has been widespread, as shown by the Solarwinds, Codecov, and the log4j attacks. The software industry has experienced a tectonic shift from passive adversaries finding and exploiting vulnerabilities to a new generation of supply chain attacks where they aggressively implant malware directly into open source projects and find their way into build and deployment pipelines.[3]

Particularly since the 12 May 2021 Executive Order 14028 in the United States, which set forth that vendors must produce a software bill of materials (SBOM) for all the products they sell to the U.S. government, SBOM production has been a focal point of software supply chain hardening. Balliu et al.[A1] share the results of a comparison of six SBOM generation tools run on 26 Java software products. Their study reveals some hard challenges for the accurate production and usage of SBOMs.

Digital Object Identifier 10.1109/MSEC.2023.3321189
Date of current version: 10 November 2023

The European regulation on cybersecurity mandated consistent software certification. In the era of continuous development and continuous integration, having to repeat certification processes for each change could cripple the software development industry. Milánkovich and Tuma[A2] propose focusing the recertification only on actual changes and their effects. This could help to shorten the recertification process. They describe and validate an industry-level tool to support practitioners in analyzing and visualizing delta certification (i.e., analyzing the introduced changes in a new project version). They show the tool could help analysts to quickly check whether currently used software packages are vulnerable to known vulnerabilities.

Ladisa et al.[A3] propose a general taxonomy of open source software supply chain attacks based upon a systematic literature review of both scientific and gray literature. The taxonomy describes how attackers conduct the attacks. To mitigate such attacks, the authors provide a mapping from attack tree nodes to safeguards, which helps to determine the final exposure to supply chain attacks.

Torres-Arias et al.[A4] discuss in their viewpoint what is really behind an SBOM. By using two open source SBOM quality tools and digging into a new publicly available SBOM "bom-shelter" dataset, they show that SBOMs have a long way to go to be really informative: only one percent of the analyzed SBOMs contain the U.S. National Telecommunications and Information Administration's "minimum elements" data for all reported components.

Melara and Torres-Arias[A5] share their viewpoint that the current adoption challenges being faced to secure the software supply chain are largely caused by the language we use to describe risk and defenses, and other sociocultural gaps, rather than technological problems. They argue that by shedding light on these gaps, we can find great opportunities to build practical defenses and to support software vendors' unique needs for software supply chain security and compliance.

Fourné et al.[A6] share a viewpoint that securing the software supply chain involves securing not just dependencies and build systems but also what may be the weakest link in the software supply chain, the individual developers. They provide a comprehensive approach that shows how human factors are crucial for effective software supply chain security. The usability of supply chain security measures by the people who will use them, especially developers, is arguably a critically underinvestigated area of research.

We hope this selection of peer-reviewed and viewpoint articles will be of interest to both practitioners and researchers. As in the poetry of the Ancient Mariner, becoming aware of the current state of practice and the challenges faced by research makes "a sadder and wiser coder, one rose the morrow morn." We look forward to our readers submitting their opinions and their research to further improve one of the pillars of the modern economy. ■

Acknowledgments

This work was partially supported by the European Union H2020 AssureMOSS project (Project 952647).

Appendix: Related Articles

A1. M. Balliu et al., "Challenges of producing software bill of materials for Java," *IEEE Security Privacy*, vol. 21, no. 6, pp. 12–23, Nov./Dec. 2023, doi: 10.1109/MSEC.2023.3302956.

A2. Á. Milánkovich and K. Tuma, "Delta security certification for software supply chains," *IEEE Security Privacy*, vol. 21, no. 6, pp. 24–33, Nov./Dec. 2023, doi: 10.1109/MSEC.2023.3311464.

A3. P. Ladisa, S. E. Ponta, A. Sabetta, M. Martinez, and O. Barais, "Journey to the center of software supply chain attacks," *IEEE Security Privacy*, vol. 21, no. 6, pp. 34–49, Nov./Dec. 2023, doi: 10.1109/MSEC.2023.3302066.

A4. S. Torres-Arias, D. Geer, and J. S. Meyers, "A viewpoint on knowing software bill of materials quality when you see it," *IEEE Security Privacy*, vol. 21, no. 6, pp. 50–54, Nov./Dec. 2023, doi: 10.1109/MSEC.2023.3315887.

A5. M. S. Melara and S. Torres-Arias, "A viewpoint on software supply chain security: Are we getting lost in translation?" *IEEE Security Privacy*, vol. 21, no. 6, pp. 55–58, Nov./Dec. 2023, doi: 10.1109/MSEC.2023.3316568.

A6. M. Fourné, D. Wermke, S. Fahl, and Y. Acar, "A viewpoint on human factors in software supply chain security: A research agenda," *IEEE Security Privacy*, vol. 21, no. 6, pp. 59–63, Nov./Dec. 2023, doi: 10.1109/MSEC.2023.3316569.

References

1. H. Mack and T. Schroer, "Security midlife crisis: Building security in a new world," *IEEE Security Privacy*, vol. 18, no. 4, pp. 72–74, Jul./Aug. 2020, doi: 10.1109/MSEC.2020.2989643.
2. F. Massacci and I. Pashchenko, "Technical leverage: Dependencies are a mixed blessing," *IEEE Security Privacy*, vol. 19, no. 3, pp. 58–62, May/Jun. 2021, doi: 10.1109/MSEC.2021.3065627.
3. 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.

Fabio Massacci is a Professor at the University of Trento, 38123 Trento, Italy, and chair of foundational security at Vrije Universiteit of Amsterdam, 1081 HV Amsterdam, The Netherlands. His research interests include foundational and experimental approaches to security. Massacci received a Ph.D. in computer science and engineering from the University of Rome La Sapienza. In 2015, he received the 10 Year Most Influential Paper Award at the IEEE Requirements Engineering Conference for his work on security in sociotechnical systems. He is the chair of the security and defense specialty group of the Society for Risk Analysis and coordinates the H2020 AssureMOSS project on the security of multiparty open source software and the upcoming HE Sec4AI4Sec project on security for AI and AI for security. He is a Member of IEEE. Contact him at fabio.massacci@ieee.org.

Laurie Williams is a distinguished university professor, director of the Secure Software Supply Chain Center, and codirector of the Secure Computing Institute at North Carolina State University, Raleigh, NC 27695 USA. Her research interests include software security. Williams received a Ph.D. in computer science from the University of Utah. She is a Fellow of IEEE and a Fellow of the Association for Computing Machinery. Contact her at lawilli3@ncsu.edu.

Digital Object Identifier 10.1109/MSEC.2023.3324513