Research Proposals

Yao-Wen Chang
Student ID: 1346258

Supervisor: Christoph Treude
Industry Mentor: Behnaz Hassanshahi (Oracle Labs, Australia)

Word Count: X words

Submission Date: September 17, 2023

Abstract

This literature review will introduce Continuous Integration/Continuous Delivery (CI/CD). Also, some attack surface and how the malicious attacker exploit these attack surface will be covered in this review. From the developers' and maintainers' point of view, the methods and frameworks are going to be introduced to counter the attack. In this literature review, the framework will focus on Supply Chain Level Security Artifacts (SLSA) which is adopted by Google.

1 Introduction

Continuous IntegrationContinuous Delivery (CI/CD) is a development process for quickly building and testing code changes that helps organizations maintain a consistent code base for their applications while dynamically integrating code changes. Therefore, CI/CD environments are attractive targets for malicious cyber actors (MCAs) whose goals are to compromise information by introducing malicious code into CI/CD applications, gaining access to intellectual property/trade secrets through code theft, or causing denial of service effects against applications.

Recent incidence like the infection of SolarWind's Orien platform [1, 6] which is used to monitor and manage the network is downloaded by thousands customers, including U.S. government agencies, critical infrastrure providers, and private companies.

Section 2 will briefly introduce CI/CD, and section 3 would target the attack surface within the process of CI/CD. Then, section 4, we are going to map the defense method to the attack method, and providing real world attack event. In section 5, our literature review would introduce SLSA framework from Google, and explain how SLSA can patch the vulnerable CI/CD process. In section 6, the aim and objects of the research will be explained. And the research plan will be introduced in section 7.

2 Definition of CI/CD

CI/CD is a development process for automatically building and testing code changes that support orgamizations maintain a consistent code base. CI involves developers frequently merging code changes into a central repository where automated builds and tests run. Build is the process of converting source code into executable code. Then, running the automated tests against the build. These process will avoid integration challenges that can happen when waiting for release day to merge changes into the release branch [5].

The convenience and capabilities of the third-party source code usually brings cybersecurity risks [3]. Software supply chain attacks aim at injecting code into software components to compromise downstream users [1]. Some vulnerabilities will be introduced in the next section.

3 Attack Surface and Impact

3.1 Source Code Repository

Obtain Git Repository credentials by dumping the environment Variables. Then, user the stolen secrets to access the Repository and modify the CI/CD configuration. Finally, the code will successfully be injected into the code base, and affects downstream users.

3.2 Build/Test

3.3 Deploy

Java Virtual Machine (JVM) executes Java bytecode and provides strong safety guarantees. However, the unsafe API, "sun.misc.Unsafe", will cause serious security issue if it is misused by the developers. The research [3] studied a large repository, Maven, and analyzed the compiled Java code. The security issues include violating type safety, crashing the virtual machine (VM) , uninitialized objects and so on. These misuse might impact third-party package management service.

4 Defense Method

These goals are seek to establish the trust in the software supply chain by verifying in- formation about the participants or processes [4]. The complex CI/CD has the potential to catch the attackers more quickly via peer review, because the developers and distributors get a chance to review the code, increasing the chances of malicious code being discovered [2]. Providing cryptographic hashes if the software packages is of significance to verify the software's integrity [2]. Signing the releases of the packages by the providers with the public-key enable the users to verify the

Some of the projects aims at providing single solution that conflates multiple objectives [4].

Despite the previously introduced methods seems to address all the security issue existed in the code base and within the CI/CD, some of them may overemphasize one particular approach to address software supply chain security. without considering compounding factors that impact risk [4].

5 Supply Chain Level Security Artifacts

5.1 What is Software Supply Chain

Software supply chain is composed multiple components, first-party or third- party libraries, and processes used to develop, build, test, and publish a software artifact [5].

6 Research Aims and Objectives

7 Research Plan

References

[1] Piergiorgio Ladisa, Henrik Plate, Matias Martinez, and Olivier Barais. Sok: Taxonomy of attacks on open-source software supply chains. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1509–1526. IEEE, 2023.

[2] Elias Levy. Poisoning the software supply chain. *IEEE Security & Privacy*, 1(3):70–73, 2003.

[3] Luis Mastrangelo, Luca Ponzanelli, Andrea Mocci, Michele Lanza, Matthias Hauswirth, and Nathaniel Nystrom. Use at your own risk: The java unsafe api in the wild. *ACM Sigplan Notices*, 50(10):695–710, 2015.

[4] Marcela S. Melara and Mic Bowman. What is software supply chain security?, 2022.

[5] U.S. Department of Defense. Defending continuous integration/continuous delivery (ci/cd) environments. PDF document, 2023.

[6] Sean Peisert, Bruce Schneier, Hamed Okhravi, Fabio Massacci, Terry Benzel, Carl Landwehr, Mohammad Mannan, Jelena Mirkovic, Atul Prakash, and James Bret Michael. Perspectives on the solarwinds incident. *IEEE Security & Privacy*, 19(2):7–13, 2021.