УДК 004.056

А.В. Семенченко
Л.М. Куперштейн

СИСТЕМА ІНТЕГРАЦІЇ ЗАСОБІВ ТЕСТУВАННЯ БЕЗПЕКИ ДОДАТКУ В КОНВЕЄР РОЗРОБКИ

Вінницький національний технічний університет

Анотація

Визначено поняття конвеєру безперервної інтеграції та доставки. Проаналізовано компоненти конвеєру. Розглянуто питання безпеки при впровадженні та побудові CI/CD. Наведено конкретні приклади використання

Ключові слова: тестування безпеки, інтеграція, *безперервна інтеграція, безперервна доставка*, DevOps, DevSecOps

Abstract

The concept of continuous integration and delivery pipeline is defined. The components of the conveyor were analyzed. The issue of security during the implementation and construction of CI/CD is considered. Specific examples of use are provided

Keywords: security testing, continuous integration, continuous delivery , DevOps, DevSecOps.

Вступ

На початку розвитку комп'ютерної техніки, розробка програмного забезпечення зводилась до написання ПЗ, запису його на носій, копіювання на цільову систему та подальшого запуску. З часом персональні комп'ютери почали з'являтись у все більшої кількості людей, що спричинило більший запит на різного роду програмного забезпечення, якість та безпеку. Для того щоб задовільнити ринок, технологічним компанія довелось покращувати процес розробки ПЗ. В процесі розробки почали активно використовувати системи контролю версій, віртуалізацію, контейнеризацію.

Зараз програмне забезпечення розвивається неперервно перш за все заради покращення функціоналу та оптимізації ПЗ, втім частіше всього оновлення роблять за для виправлення помилок та виправлення вразливостей.

Continuous Integration (CI) - перша практика, що виникла у середині 2000-х років. Вона полягає у постійній інтеграції коду розробників у спільний репозиторій з метою виявлення можливих конфліктів між кодом різних розробників і відшкодування їх до того, як код відправиться на експлуатацію [1].

Continuous Delivery (CD) - наступний етап в еволюції процесів розробки, який з'явився наприкінці 2000-х років. Він передбачає автоматизацію процесу випуску програмного забезпечення в експлуатацію. За допомогою CD розробники можуть автоматично збирати, тестувати та випускати свій код у експлуатацію, що зменшує час випуску програмного забезпечення на ринок [2].

DevOps - термін, який виник на початку 2010-х років та поєднує практики CI та CD з метою поліпшення співпраці між розробниками, тестувальниками та операторами. DevOps передбачає налагодження процесів співпраці та взаємодії між цими групами фахівців, а також використання інструментів автоматизації для підвищення ефективності та швидкості розробки та випуску програмного забезпечення [3].

DevSecOps - це розвиток DevOps, що виник наприкінці 2010-х років, і полягає у включенні практик безпеки у процеси розробки програмного забезпечення. DevSecOps сприяє постійному включенню аналізу потенційних загроз безпеці у процеси розробки, щоб забезпечити, що програмне забезпечення випускається з необхідним рівнем безпеки та не містить вразливостей, які можуть бути використані зловмисниками. DevSecOps передбачає взаємодію між командами безпеки, розробниками та операторами, використання інструментів аналізу безпеки коду та автоматизацію процесів забезпечення безпеки програмного забезпечення.[4]

У багатьох компаній почались проблеми з розробкою, доставкою, розгортанням та контролем всього процесу. Для вирішення цих проблем була створена методика CI/CD.

CI/CD — це метод швидкої доставки додатків клієнтам шляхом впровадження автоматизації на етапах розробки додатків. Основні концепції CI/CD — безперервна інтеграція, безперервна доставка та безперервне розгортання [2]. CI/CD — це рішення проблем, які інтеграція нового коду може спричинити для команд розробки та операцій (відомих також як «інтеграційне пекло»).

Результати дослідження

При впровадженні CI/CD важливо забезпечити високий рівень безпеки. Ось деякі з найбільш важливих питань безпеки в конвеєрі CI/CD :

1. Захист від несанкціонованого доступу: необхідно забезпечити, щоб тільки авторизовані користувачі мали доступ до конвеєра CI/CD. Для цього можна використовувати механізми аутентифікації та авторизації, такі як OAuth2 або LDAP.

2. Захист від вразливостей в програмному забезпеченні: під час використання CI/CD важливо забезпечити, щоб весь код, який проходить крізь конвеєр, був безпечним. Це означає, що код повинен бути перевірений на наявність вразливостей перед тим, як він буде дозволений на наступний етап конвеєра.

3. Захист від зловмисного програмного забезпечення: під час використання конвеєра CI/CD важливо забезпечити, щоб весь код, який проходить крізь конвеєр, був перевірений на наявність шкідливого програмного забезпечення. Для цього можна використовувати засоби, такі як антивіруси та інші інструменти з безпеки.

4. Захист від проблем з конфігурацією: під час використання CI/CD важливо забезпечити, щоб конфігурація конвеєра була безпечною. Наприклад, важливо забезпечити, щоб віддалені сервери, на яких запускається код, були налаштовані правильно та захищені.

5. Захист від проблем зі сховищами даних: під час використання CI/CD важливо забезпечити, щоб всі дані, що зберігаються в ході проходження конвеєра, були безпечними. Наприклад, важливо забезпечити, щоб дані авторизації та конфігураційні файли були захищені від несанкціонованого доступу.

6. Захист від атак DDoS: під час використання CI/CD важливо забезпечити, щоб сервери, на яких запускається код, були захищені від атак типу DDoS. Для цього можна використовувати файєрволи.

7. Захист від втручання в процес CI/CD: під час використання CI/CD важливо забезпечити, щоб процес збірки, тестування та доставки програмного забезпечення не можна було втрутитися в ручному режимі. Для цього можна використовувати механізми контролю доступу та автоматизації процесу.

8. Захист від втрати даних: під час використання CI/CD важливо забезпечити, щоб всі дані, що зберігаються в процесі конвеєра, були збережені в безпечному місці та були відновлюваними в разі втрати даних.

Загалом, безпека є надзвичайно важливою під час використання конвеєра CI/CD. Необхідно забезпечити всі можливі заходи забезпечення для захисту коду та даних в ході проходження конвеєра та забезпечити, щоб весь процес збірки, тестування та доставки програмного забезпечення був безпечним та автоматизованим.

Система інтеграції тестування безпеки додатку в конвеєр CI/CD дозволяє забезпечити автоматизацію процесу тестування та виявлення вразливостей, що дозволяє швидко виправляти помилки та забезпечувати безпеку продукту. Для цього можуть використовуватися різноманітні тестові сканери та інструменти, які дозволяють проводити тестування на різних рівнях, таких як сканування вразливостей, тестування на проникнення тощо.

Однією з ключових переваг інтеграції тестування безпеки в конвеєр CI/CD є можливість виявляти потенційні вразливості на ранніх етапах розробки, що зменшує ризик їх появи в продукті в майбутньому. Також, інтеграція тестування безпеки може допомогти забезпечити виконання вимог безпеки та стандартів, які обов'язково повинні виконуватися в деяких галузях.

Для успішної інтеграції тестування безпеки в конвеєр CI/CD, слід використовувати інструменти, які забезпечують автоматизоване тестування безпеки та надійне виявлення вразливостей. Одним з таких інструментів може бути програмний комплекс OWASP ZAP [5], який дозволяє проводити тестування на різних рівнях та забезпечує автоматичну інтеграцію в конвеєр CI/CD [5].

Також, для успішної інтеграції тестування безпеки в конвеєр CI/CD необхідно дотримуватися певних принципів, таких як забезпечення постійного тестування, моніторингу та звітування про виявлені вразливості, забезпечення безпеки на всіх етапах розробки та впровадження продукту.

Окрім того, слід забезпечити взаємодію розробників, тестувальників та команди з безпеки продукту, щоб забезпечити високий рівень безпеки продукту та відповідність вимогам безпеки.

Компанія Amazon Web Services (AWS) надає безкоштовні інструменти для автоматизованої перевірки безпеки в CI/CD конвеєрі. Один з таких інструментів - AWS CodePipeline, дозволяє автоматично запускати та виконувати тестування на безпеку під час кожного етапу CI/CD конвеєра [6]. Інструмент SonarQube - це інструмент, який дозволяє проводити автоматичну перевірку безпеки та якості коду. SonarQube можна інтегрувати в CI/CD конвеєр та автоматично виконувати тестування під час кожного етапу [7].

Загалом, CI/CD та тестування безпеки при розробці масштабних продуктів є необхідними речами, які до того ж гарно поєднуються.

Висновки

Важливо розуміти, що система тестування безпеки додатку в конвеєрі CI/CD - це не єдиний метод забезпечення безпеки програмного забезпечення. Вона повинна бути частиною загальної стратегії безпеки, яка включає в себе такі елементи, як аналіз ризиків, використання стандартів та нормативів безпеки, кібербезпеку відповідно до законодавства та інші практики.

У підсумку, система тестування безпеки додатку в конвеєрі CI/CD дозволяє забезпечити постійний моніторинг та звітування про вразливості безпеки, що дозволяє розробникам та команді безпеки приймати швидкі та ефективні рішення для захисту додатку від загроз безпеки. Така система є необхідною частиною будь-якої стратегії безпеки

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Qentelli: An Introduction to Continuous Integration. URL: https://www.qentelli.com/thought-leadership/insights/continuous-integration#:~:text=The%20term%20Continuous%20Integration%20was,Chrysler%20Comprehensive%20Compensation%20System%20Project (Access date 23.05.23).
2. Circleci: A Brief History of DevOps, Part III: Automated Testing and Continuous Integration. URL: https://circleci.com/blog/a-brief-history-of-devops-part-iii-automated-testing-and-continuous-integration/#:~:text=The%20phrase%20continuous%20integration%20(CI,Ivar%20Jacobson%20and%20James%20Rumbaugh (Access dat e23.05.23).
3. Itrevolution: The History Of DevOps. URL: https://itrevolution.com/articles/the-history-of-devops (Access date 23.05.23).
4. Devops Institute: "The History of DevSecOps and 10 Ways to Advance DevSecOps" URL: https://www.devopsinstitute.com/the-history-of-devsecops/#:~:text=The%20history%20of%20DevSecOps%20starts,describe%20the%20attributes%20of%20quality.&text=Many%20of%20these%20papers%20described,to%20be%20next%2Dgeneration%20quality (Access date 23.05.23).
5. Zed Attack Proxy (ZAP). URL: https://www.zaproxy.org (Access date 23.05.23).
6. AWS: "AWS CodePipeline". URL: https://aws.amazon.com/codepipeline (Access date 23.05.23)
7. SonarQube: "About SonarQube" . URL: https://www.sonarqube.org/ (Access date 23.05.23)

Семенченко Антон Валерійович – студент групи 1БС-19Б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: sav.miner@gmail.com

Куперштейн Леонід Михайлович – кандидат технічних наук, доцент кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця, e-mail: kupershtein@vntu.edu.ua.

Semenchenko Anton Valeriyovych - student of group 1BS-19B, faculty of information technologies and of computer engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: sav.miner@gmail.com

Kupershtein Leonid Mykhailovych – Candidate of Technical Sciences, Associate Professor of the Department of Information Protection, Vinnytsia National Technical University, Vinnytsia, , e-mail: kupershtein@vntu.edu.ua