УДК 004.056.53

ТЕСТУВАННЯ БЕЗПЕКИ ПРОГРАМНОГО КОДУ

Казмірчук Є., Ткачук Р.
Львівський державний університет безпеки життєдіяльності, м .Львів

В роботі проведено аналіз існуючих методик розробки програмного забезпечення, методів тестування безпеки програмного коду а також здійснено аналіз платформи Kubernetes та Kubernetes операторів, як засобу автоматизації роботи додатків.

Ключові слова: *безпека програмного забезпечення, програмний код, тестування, вразливість.*

The paper analyzes the existing methods of software development, methods of testing the security of software code, as well as the analysis of the platform Kubernetes and Kubernetes operators as a means of automating applications.

Keywords: *software security, software code, testing, vulnerability.*

Сьогодні сфера розробки програмного забезпечення розвивається дуже швидко. Щодня з'являються нові розробники та компанії, готові запропонувати послуги з розробки програмного забезпечення. З іншого боку, у світі також активно розвивається сфера кіберзлочинності, тому вкрай важливо забезпечити надійний рівень безпеки програм, що розробляються.

Інструменти, що використовуються для підвищення безпеки програмного забезпечення, зазвичай включають пошук, виправлення, попередження та запобігання вразливостям безпеки. Як правило, інструменти та методи використовуються для виявлення слабких місць у життєвому циклі програмного забезпечення, таких як планування, проектування, розробка, розгортання, модифікація та обслуговування. Передбачається, що тести безпеки будуть виконуватися протягом усього життя програми, щоб уразливості були належним чином усунені.

Кіберзлочинці все частіше націлені на веб-додатки, тому організаціям потрібно відавати пріоритет питанням безпеки в SDLC. Це особливо актуально, коли програмне забезпечення є критичним.

Використання сканера безпеки веб-додатків та виконання інших форм тестування безпеки веб-додатків на початку процесу допоможе зменшити ризик, вирішити проблеми, що виникають, і знизити витрати [1].

SDLC є ефективним методом проектування та створення програмного забезпечення

Безпека програмного продукту включає аналіз архітектури під час проектування, перегляд коду під час кодування та конструювання, а також тестування на проникнення перед випуском [2]. Основні переваги безпечного підходу SDLC:

– програмне забезпечення безпечніше, оскільки безпека є постійною складовою при розробці;

– всі причетні до інформаційного продукту обізнані з міркуваннями безпеки;

– недоліки конструкції виявляються до їх кодування;

– економія коштів за рахунок раннього виявлення та усунення дефектів;

– зниження загальних внутрішніх бізнес-ризиків для організації.

Захищений SDLC передбачає інтеграцію тестування безпеки та інших заходів у існуючий процес розробки. Реалізація безпеки SDLC впливає на кожну фазу процесу розробки програмного забезпечення.

Існує багато типів автоматизованих інструментів для виявлення вразливостей програмного забезпечення. Поширені технології для виявлення вразливостей програмного забезпечення включають:

– статичний тест безпеки (SAST);

– динамічна перевірка безпеки (DAST);

– інтерактивне тестування безпеки додатків (IAST).

Автоматизація в кожній галузі дає переваги підвищення продуктивності та зниження витрат. Завдяки гнучкій розробці програмного забезпечення автоматизація стала настільки невід'ємною частиною гнучкого тестування, що одне без іншого важко уявити.

Тестування безпеки розвивається швидше, ніж будь-який інший ринок безпеки, оскільки рішення AST адаптуються до нових методів розробки та збільшують складність додатків.

Таким чином, автоматизовані тести інтегруються в цикл тестування, а модель DevOps залишається в центрі уваги. Це породило широкий спектр інструментів і технологій, які можна використовувати для проведення тестування безпеки з точки зору DevOps. Безперервне тестування та розгортання є основою моделі DevSecOps і роблять процес тестування та розробки спільним [2].

Kubernetes - це портативна, розширювана платформа з відкритим кодом для керування робочими навантаженнями та контейнерними службами, яка полегшує як декларативну конфігурацію, так і автоматизацію. Має велику, швидко зростаючу екосистему. Служби, підтримка та інструменти Kubernetes широко доступні.

Kubernetes забезпечує основу для сталого впровадження розподілених систем. Він дбає про масштабування та відновлення після відмови програми, надає схеми розгортання тощо.

Kubernetes визначає набір будівельних блоків, які разом забезпечують механізми для розгортання, керування та масштабування програм на основі процесора, пам'яті або метричних показників користувача. Kubernetes є вільно з'єднаним і розширюваним, щоб забезпечувати різні навантаження.

Ця розширюваність значною мірою забезпечується Kubernetes API, який використовується внутрішніми компонентами, а також розширеннями та контейнерами, що працюють на Kubernetes [3]. Платформа контролює обчислювальні ресурси та ресурси зберігання та визначає ресурси як об'єкти, якими потім можна керувати.

Додаток на Kubernetes – це програма, яка розгорнута на самому Kubernetes і керується за допомогою Kubernetes API та інструментів kubectl.

Щоб отримати максимальну віддачу від Kubernetes, потрібен доступ до повного набору API, які можна розширити, щоб підтримувати та керувати своєю програмою на основі Kubernetes. У цьому випадку оператори діють як середовище виконання для керування цим типом додатків у Kubernetes.

Оператори Kubernetes можуть виконувати різноманітні операційні завдання, від базової функціональності до логіки. Більш просунуті оператори можуть автоматично реагувати на помилки та виконувати оновлення. Оператори можуть бути адаптовані до потреб конкретної організації, і вони можуть бути повторно використані в різних програмах [4]. Щоб створити власних операторів, ви можете використовувати Operator Framework, який надає інструменти, необхідні для створення, керування та моніторингу ефективності оператора.

Література

1. OWASP Testing Guide 4.0 [Електронний ресурс] – Режим доступу: https://owasp.org/www-pdf-archive/OTGv4.pdf

2. OWASP Software Assurance Maturity Model [Електронний ресурс]: Режим доступу: https://owaspsamm.org

3. Офіційний сайт платформи Kubernetes [Електронний ресурс]: Режим доступу: https://kubernetes.io/docs/home/

4. Офіційна документація Kubernetes Operators [Електронний ресурс]: Режим доступу: https://kubernetes.io/docs/concepts/extend-kubernetes/operator/