УДК 004.056

А.С. МАМОНТОВ[1], В.Л. ЕВСЕЕВ[1,2], А.С. БУРАКОВ[1], В.С. МАМОНТОВ[3]

[1]Московский государственный лингвистический университет
[2]Национальный исследовательский ядерный университет «МИФИ», Москва
[3]ООО СК «Сбербанк страхование», Москва

АВТОМАТИЗИРОВАННЫЙ КОНТРОЛЬ КОДА С ПОМОЩЬЮ СТАТИЧЕСКОГО АНАЛИЗА В QUALITY GATE: ОБНАРУЖЕНИЕ УЯЗВИМОСТЕЙ И БЛОКИРОВКА НЕБЕЗОПАСНОГО КОДА ДЛЯ ЗАЩИТЫ ПОЛЬЗОВАТЕЛЕЙ И СОЦИУМА

Целью работы является исследование инструментов статического анализа (SAST) в Quality Gates для предотвращения внедрения вредоносного программного обеспечения (ПО), уязвимых библиотек и фишинговых механизмов, угрожающих финансовой безопасности и приватности пользователей. Проанализированы методы выявления скрытых угроз: трояны в зависимостях, поддельные интерфейсы, эксплойты. Особое внимание уделено блокировке кода, нарушающего этические нормы, и интеграции SAST с анализом зависимостей (SCA) для исключения компонентов с критическими уязвимостями [1] Исследованы критерии защиты от социальной инженерии, устаревших библиотек, минимизирующие риски утечек данных и мошеннические схемы.

Введение

Современная разработка ПО требует этической ответственности разработчиков: уязвимости и фишинговые элементы угрожают пользователям и могут привести к потере данных. Интеграция SAST и SCA в Quality Gates позволяет не только выявлять классические уязвимости (XSS, SQL-инъекции), но и блокировать подозрительные алгоритмы и поддельные SDK, сохраняя доверие клиентов и защищая их от мошенничества [2].

Постановка задачи

Система автоматизированного контроля кода с помощью статического анализа в Quality Gate позволяет решать задачи:

– обнаружение уязвимостей в исходном коде: SAST-инструменты анализируют исходный код приложения без его выполнения, выявляя потенциальные уязвимости безопасности, такие как SQL-инъекции, межсайтовый скриптинг (XSS), небезопасная работа с данными и другие.

– автоматическая блокировка небезопасного кода: интеграция SAST в Quality Gates позволяет автоматически блокировать сборку и развертывание приложения, если обнаружены критические уязвимости. Это предотвращает попадание небезопасного кода в продуктивную среду и обеспечивает соблюдение политик безопасности организации;

– оценка рисков и приоритизация уязвимостей: SAST-инструменты предоставляют детальную информацию о типе, критичности и потенциальном влиянии обнаруженных уязвимостей. Это позволяет команде разработчиков эффективно расставлять приоритеты при устранении проблем безопасности и фокусироваться на наиболее критичных угрозах.

Пути решения задачи

Для решения задач автоматизированного контроля кода при разработке ПО целесообразно внедрять инструменты статического анализа безопасности приложений (SAST) в системы Quality Gates. После анализа существующих решений в области автоматизации процесса обнаружения уязвимостей определяются наиболее универсальные и эффективные инструменты. Основное внимание уделяется возможности интеграции с CI/CD, точности обнаружения уязвимостей, скорости анализа, поддержке различных языков программирования и стоимости внедрения.

По итогам анализа предпочтение в использовании целесообразно отдать: 1. SonarQube; 2. PT Application Inspector; 3. Checkmarx SAST.

Заключение

Системы автоматизированного контроля кода с помощью статического анализа, такие как SonarQube, PT Application Inspector и Checkmarx SAST, играют критическую роль в обеспечении кибербезопасности современных программных продуктов. Интеграция SAST-инструментов в Quality Gates позволяет автоматически выявлять уязвимости в исходном коде на ранних этапах разработки и блокировать внедрение небезопасного кода в продуктивную среду.

Список литературы

1. Марков А.С. Важная веха в безопасности открытого программного обеспечения. Вопросы кибербезопасности. 2023, № 1(53), с. 2–12. DOI: http://dx.doi.org/10.21681/2311-3456-2023-1-2-12.

2. Шинкарев А.А. Роль программного обеспечения с открытым исходным кодом в современной разработке корпоративных информационных систем // Компьютерные и информационные науки. – 2021. – № 1. DOI: 10.14529/ctcr210202