УДК 004.056

Метод автоматизированной фильтрации ложных срабатываний статического анализа исходного кода в процессах непрерывной интеграции и доставки программного обеспечения для веб-приложений с использованием больших языковых моделей

Ефимов М. Д. (Университет ИТМО)

Научный руководитель – аспирант, научный сотрудник Еритенко Н. А.
Университет ИТМО

Введение

В настоящее время в индустрии разработки веб-приложений характерна высокая частота релизов и применение практик непрерывной интеграции и доставки программного обеспечения. В рамках DevOps- и DevSecOps-подходов инструменты статического анализа исходного кода являются обязательным этапом автоматизированной проверки безопасности [1].

Однако при практическом применении статического анализа остаётся проблема высокой доли ложных срабатываний. Наличие большого количества ложных предупреждений увеличивает время обработки результатов анализа и снижает доверие разработчиков к инструментам контроля безопасности [1]. Что приводит к снижению эффективности встроенных механизмов обеспечения качества и безопасности.

Существующие методы уменьшения количества ложных срабатываний преимущественно основаны на ручной настройке правил и использовании эвристических фильтров. Данные подходы не учитывают семантический контекст приложения, особенности архитектуры и бизнес-логики конкретного проекта.

В настоящее время развитие больших языковых моделей, которые могут осуществлять семантический анализ программного кода и учитывать контекст, позволяет интеллектуализировать процедуру статического анализа [2, 3]. В работе предлагается метод автоматизированной фильтрации ложных срабатываний с применением больших языковых моделей, интегрируемый в CI/CD-конвейер веб-приложений.

Основная часть

Предлагаемый метод реализует дополнительный этап обработки результатов статического анализа перед формированием итогового отчёта в CI/CD-пайплайне. После завершения анализа формируется перечень предупреждений, для каждого из которых извлекается контекст, включающий фрагмент исходного кода. Эти данные передаются в модуль обработки на основе большой языковой модели.

Модель выполняет анализ кода и предупреждения, оценивая вероятность его ложного характера. По результатам оценки принимается решение о фильтрации предупреждения либо его передаче разработчику. На основе предложенного метода реализуется решение, которое интегрируется в CI/CD-инфраструктуру без модификации существующих средств анализа и демонстрирует снижение доли ложных срабатываний при сохранении чувствительности к реальным уязвимостям.

Выводы

Разработан метод автоматизированной фильтрации ложных срабатываний статического анализа исходного кода на основе больших языковых моделей. Предложенный метод интегрируется в процессы непрерывной интеграции и доставки

веб-приложений и служит основой для построения алгоритмов и программных решений, выполняющих интеллектуальную обработку предупреждений статического анализа. Применение метода нацелено на сокращение количества ложных срабатываний, тем самым повышая точность анализа исходного кода.

Литература

1. Li H., Hao Y., Zhai Y., Qian Z. Assisting Static Analysis with Large Language Models: A ChatGPT Experiment // Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2023). – San Francisco, USA, 2023. – С. 1463–1475. – DOI: 10.1145/3611643.3613897.
2. Mohajer M. M., Aleithan R., Harzevili N. S., Wei M., Belle A. B., Wang S. SkipAnalyzer: A Tool for Static Code Analysis with Large Language Models [Электронный ресурс] // arXiv preprint arXiv:2310.18532. – 2023. – 14 с. – Режим доступа: https://arxiv.org/abs/2310.18532 (дата обращения: 20.01.2026).
3. Shafiei N., Shajari M., Khadivi S. Using ChatGPT as a Static Application Security Testing Tool [Электронный ресурс] // arXiv preprint arXiv:2308.14434. – 2023. – Режим доступа: https://arxiv.org/abs/2308.14434 (дата обращения: 22.01.2026).