УДК 004.056

РАЗРАБОТКА МЕТОДИКИ ВЫЯВЛЕНИЯ ЛОЖНОПОЛОЖИТЕЛЬНЫХ СРАБАТЫВАНИЙ В РЕЗУЛЬТАТАХ СТАТИЧЕСКОГО АНАЛИЗА БЕЗОПАСНОСТИ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

Нгуен К.Т. (ИТМО)
Научный руководитель – кандидат технических наук, доцент Менщиков А.А.
(ИТМО)

Введение. Статический анализ безопасности приложений (SAST) является одним из базовых инструментов выявления уязвимостей на ранних этапах жизненного цикла программного обеспечения. Однако практическое применение SAST ограничивается высокой долей ложноположительных срабатываний, что приводит к росту трудозатрат на ручной триаж, снижению доверия к отчётам и ухудшению эффективности процессов DevSecOps. В результате возникает необходимость в методах постобработки, позволяющих повышать практическую ценность результатов SAST без существенного снижения полноты выявления слабостей. Перспективным направлением является использование больших языковых моделей (LLM) для контекстной интерпретации результатов анализа. При этом ключевыми остаются вопросы воспроизводимости решений LLM и корректной подготовки входных данных: модель должна опираться на наблюдаемые доказательства в коде и в отчёте анализа, а результат классификации должен быть формализован и пригоден для автоматизированной интеграции [1].

Основная часть. Предлагается методика выявления ложноположительных срабатываний в результатах SAST с использованием больших языковых моделей. Методика реализуется как слой постобработки и включает последовательность этапов:

1) Нормализация и каноникализация результатов SAST. В качестве входного формата используется SARIF [2], обеспечивающий унифицированное представление срабатываний (правило, координаты, описание) и, при наличии, трасс потока данных «источник → шаги распространения → приёмник». Срабатывания приводятся к внутреннему представлению, что позволяет одинаково обрабатывать результаты разных правил и обеспечивать воспроизводимость экспериментов.
2) Контекстуальное обогащение доказательствами из кода и трасс. Для каждой обнаруженной локации извлекаются релевантные фрагменты исходного кода: строка срабатывания и окружение, границы метода/функции, а также места вызова при переходе между методами. Если в отчёте присутствуют трассы потока данных, дополнительно фиксируются элементы прохождения данных через шаги распространения, что повышает объяснимость решения и снижает риск неверных допущений.
3) Формирование запроса к LLM по CWE-ориентированному шаблону. Для каждого класса слабости [3] (CWE) применяется единый шаблон запроса, включающий: (а) роль и ограничения (использовать только предоставленные фрагменты и трассы), (б) краткие критерии уязвимости (микро-правила), (в) контрольный перечень проверки, (г) строгий формат ответа. Такой шаблон снижает вариативность генерации и повышает сопоставимость результатов между экспериментами.
4) Классификация срабатываний и выходной формат. На основании предоставленного контекста LLM выполняет бинарную классификацию: истинноположительное срабатывание (уязвимость подтверждается) либо ложноположительное (уязвимость не подтверждается). Результат сохраняется в структурированном виде для последующей агрегации статистики и интеграции в конвейер разработки.
5) Экспериментальная оценка. Эффективность методики оценивается на OWASP Benchmark

[4], а также дополнительно проверяется на реальных проектах, приближённом к промышленным условиям. Используются показатели качества классификации (точность, полнота, F1-мера), а также снижение доли ложноположительных срабатываний по сравнению с исходными результатами SAST.

Выводы. Предложена методика выявления ложноположительных срабатываний в результатах статического анализа безопасности приложений с использованием больших языковых моделей. Методика опирается на каноникализацию SARIF-данных, извлечение доказательств из исходного кода и трасс потока данных, а также на CWE-ориентированное формирование запросов к LLM со строгим форматом ответа. Практическая значимость подхода заключается в снижении нагрузки на ручной триаж и повышении применимости результатов SAST в DevSecOps-конвейерах. Экспериментальная проверка на OWASP Benchmark и реальных проектах позволяет количественно оценить выигрыш по доле ложноположительных срабатываний и метрикам качества классификации.

Список использованных источников:

1. IRIS: LLM-Assisted Static Analysis for Detecting Security Vulnerabilities – 2025. – С.2 – 10.
2. OASIS. **S**ARIF Version 2.1.0 (Static Analysis Results Interchange Format): спецификация формата представления результатов статического анализа [Электронный ресурс]. – URL: https://docs.oasis-open.org/sarif/sarif/v2.1.0/sarif-v2.1.0.html (дата обращения: 25.11.2025).
3. MITRE. Common Weakness Enumeration (CWE): классификация типовых слабостей программного обеспечения [Электронный ресурс]. – URL: https://cwe.mitre.org/top25/ (дата обращения: 15.12.2025).
4. OWASP Foundation. OWASP Benchmark Project: набор тестов для оценки инструментов анализа безопасности приложений [Электронный ресурс]. – URL: https://owasp.org/www-project-benchmark/ (дата обращения: 05.01.2026).
5. Comparison and evaluation on Static Application Security Testing (SAST) tools for Java – 2023. – С.6 – 9.
6. Harnessing Large Language Models for Software Vulnerability Detection: A Comprehensive Benchmarking Study – 2024. – С.6 – 11.
7. Understanding the Effectiveness of Large Language Models in Detecting Security Vulnerabilities – 2024. – С.3 – 11.

Автор _____ Нгуен К.Т.

Научный руководитель _____ Менщиков А.А.