УДК 004.4'23

РАЗРАБОТКА МОДУЛЕЙ РАСПОЗНАВАНИЯ ИСХОДНОГО КОДА И ВЫЯВЛЕНИЯ ЕГО НЕДОСТАТКОВ ДЛЯ СТАТИЧЕСКОГО АНАЛИЗАТОРА

Погорелов С. А.[1]
Научный руководитель – кандид. техн. наук, доцент Коцюба И. Ю.[1]
[1]Университет ИТМО
nerower.shp@gmail.com

Введение

При разработке программных продуктов особое внимание уделяется безопасности программных решений. Развитие методологии DevSecOps позволяет в значительной мере автоматизировать процессы выявления и устранения угроз безопасности на ранних этапах разработки ПО. Эта методология предполагает внедрение инструментов анализа безопасности приложений *(AST – application security testing)* в цепочку CI/CD [1].

В наиболее грубом приближении эти инструменты можно разделить на 2 группы: статического *(SAST)* и динамического *(DAST)* анализа. Их основное различие заключается в том, что при статическом анализе не требуется запускать тестируемое приложение - проверяется его исходный код и/или артефакты, полученные в результате сборки. А при динамическом анализе происходит тестирование непосредственно запущенного приложения.

В этой работе будет рассмотрен процесс разработки парсера - важного модуля статического анализатора, отвечающего за преобразование исходного кода в удобную для последующего анализа структуру.

Основная часть

Разработать модули распознавания исходного кода и шаблонных алгоритмов для анализа AST-дерева [2] в составе статического анализатора кода, обеспечивающие выявление шаблонных структур и потенциальных ошибок на этапе компиляции [3].

В работе рассматривается практическая сторона решения поставленных задач:

1. Проанализировать существующие подходы к построению и анализу AST-деревьев в статических анализаторах кода, выявив сильные и слабые стороны для выбранного языка программирования.
2. Разработать модуль парсинга и распознавания исходного кода, преобразующий текст программы в AST-дерево с учетом синтаксических конструкций целевого языка.
3. Реализовать шаблонные алгоритмы анализа AST-дерева, на основе которых будут создаваться проверки кода для выявления недостатков кода, таких как неэффективные конструкции, потенциальные утечки ресурсов или нарушения стиля кодирования.
4. Интегрировать разработанные модули в коммерческий инструмент статического анализатора, обеспечив модульность, расширяемость и поддержку пользовательских правил анализа.
5. Провести тестирование модулей на тестовых наборах кода, оценить точность распознавания, производительность и провести сравнение с аналогами.

Выводы

Модули распознавания исходного кода и шаблонных алгоритмов для анализа построенного AST-дерева разработаны и внедрены в коммерческий статический анализатор кода.

Литература

1. David N. Kleidermacher. Integrating Static Analysis into a Secure Software Development Process. 2008 IEEE Conference on Technologies for Homeland Security // IEEE URL: https://ieeexplore.ieee.org/document/4534479
2. Игнатьев В.Н. Организация статического анализа на абстрактных синтаксических деревьях с помощью конечных автоматов // Труды института системного программирования РАН. - 2025. - Том 37, № 1. - С. 7-40.
3. V. Gorbunov, D. Kochkin, D. Sorokin. Development of an API architecture for static application security analysis // Physical basis of science-intensive technologies in the modern world. - 2025 // ВГЛТУ URL: https://bibl.vgltu.ru/en/nauka/conference_article/16854/view (дата обращения: 18.11.2025).