Методы поиска дублирующихся недостатков защищённости в исходном коде

Вигура Полина Андреевна

Студент

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: pvigura@gmail.com

Научный руководитель *— Самосадный Кирилл Алексеевич*

Благодаря возможности обнаруживать проблемы на этапе разработки статический анализ исходного кода является одним из распространённых способов автоматического поиска недостатков защищённости в приложении. Однако, данный вид анализа порождает большое количество срабатываний [1], обработка которых — как ручная, так и автоматическая — ресурсозатратна.

Назовём находки, различные с точки зрения процесса анализа, но отражающие один и тот же недостаток программы, **дублирующимися**. Такие находки могут появляться, к примеру, при сканировании одной и той же кодовой базы несколькими инструментами, при итеративных проверках нескольких версий исходного кода или при обнаружении разных признаков присутствия одного и того же недостатка разными правилами одного и того же инструмента.

Для оптимизации работы с результатами статического анализа удобно сгруппировать дублирующиеся срабатывания. Такая группировка позволяет получить более подробную информацию о находке, а также сократить время, затрачиваемое на анализ результатов, за счёт уменьшения их количества.

В работе проводится обзор существующих решений в области поиска дублирующихся результатов статического анализа. Были найдены работы, основанные на сравнении текстовых описаний недостатка [2], и на информации о местоположении находки, её типе и потоке выполнения программы [3]. При обзоре выявлено, что известные методы либо не различают однотипные недостатки в разных местах в коде, либо активно используют номер строки кода с находкой, что делает такое решение неустойчивым при поиске дублирующихся результатов между версиями исходного кода при изменении номеров строк и неприменимым для обнаружения нескольких проявлений одного недостатка в разных местах в коде.

Для поиска дублирующихся срабатываний между двумя версиями исходного кода в работе предлагается решение, основанное на

сравнении типов недостатков в классификации CWE и хешей поддеревьев деревьев абстрактного синтаксиса для фрагментов кода, определённых анализатором как недостаток. Для выявления разных проявлений одного и того же недостатка в разных местах предлагается метод поиска дубликатов, основанный на анализе потоков данных: если поток данных, идущий к одному из недостатков, проходит через местоположение другого недостатка, и типы недостатков совпадают, то недостатки считаются дублирующимися. Методы реализованы и протестированы на демонстрационно уязвимом приложении, проведено сравнение с существующими решениями.

Литература

1. Dencheva L. Comparative analysis of Static application security testing (SAST) and Dynamic application security testing (DAST) by using open-source web application penetration testing tools // Masters thesis, Dublin, National College of Ireland, 2022.
2. Voggenreiter M., Schneider P., Gulraiz A. Aggregating Industrial Security Findings with Semantic Similarity-Based Techniques // Practical Solutions for Diverse Real-World NLP Applications. 2024. P. 121–139.
3. Fry Z. P., Weimer W. Clustering static analysis defect reports to reduce maintenance costs // 20th Working Conference on Reverse Engineering (WCRE), Koblenz, Germany, 2013, P. 282–291.