УДК 004.056

ОПРЕДЕЛЕНИЕ УЯЗВИМЫХ МЕТОДОВ ДЛЯ ВЫПОЛНЕНИЯ ПРОВЕРОК ДОСТИЖИМОСТИ В РАМКАХ КОМПОЗИЦИОННОГО АНАЛИЗА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Гупало А. В.[1]
Научный руководитель – Ищенко А. П.[1]
[1]Университет ИТМО
anastalucy@yandex.ru

Работа выполнена в рамках темы НИР №3 «Разметка уязвимых функций для выполнения проверки достижимости в рамках композиционного анализа программного обеспечения».

Введение

Несмотря на удобство и эффективность использования композиционного анализа, он имеет проблемы, которые могут существенно влиять на безопасность разрабатываемого программного обеспечения и загруженность специалистов, основные из них – большое количество ложноположительных срабатываний и присутствие ложноотрицательных срабатываний [1]. Интеграция в композиционный анализ программного обеспечения проверок достижимости уязвимых методов позволяет сократить ложные срабатывания, однако зачастую уязвимый метод отсутствует в описании уязвимости, также он может быть указан неверно, из-за чего возникает необходимость в дополнительном сборе и анализе информации по нему [2].

Некоторые компании решают данную проблему путем ведения собственной базы данных уязвимостей. Однако обогащение описаний уязвимостей с указанием уязвимых методов выполняется вручную, из-за чего при появлении новых уязвимостей проверка достижимости может быть не выполнена по причине отсутствия размеченного уязвимого метода. Также ручное определение уязвимых методов может занимать значительную часть рабочего времени, учитывая постоянную тенденцию роста количества новых уязвимостей. По этим причинам возникает необходимость в анализе проблем и выявлении путей и способов определения уязвимых методов для выполнения проверок достижимости, в том числе разработке алгоритма определения таких методов для последующей его автоматизации.

Основная часть

При проведении исследования были определены способы выявления уязвимых методов: использование описания уязвимости, исправляющего коммита, уязвимого коммита, других источников, например, отчета от вендора проверяемого пакета, issue на Github, PoC, эксплоитов, информации из баг-трекеров.

Для каждого способа были описаны проблемы, возникающие при определении уязвимых методов, и предложены следующие пути их решения: при определении уязвимого метода по описанию уязвимости использовать несколько источников информации; при определении метода по исправляющему или уязвимому коммиту выполнять проверку коммитов с различных сайтов, так как может быть указан неверный коммит; при определении уязвимого метода по исправляющему коммиту проверять, была ли ранее переименована функция, которую затрагивают изменения из коммита; комбинировать способы определения уязвимой функции, то есть использовать и описание уязвимости, и исправляющий коммит, и уязвимый коммит.

На основе предложенных решений выявленных проблем, была выполнена разработка алгоритма определения уязвимых методов. По результатам анализа возможности его автоматизации было определено, что данный алгоритм возможно автоматизировать, однако разработанная программа будет иметь некоторые

ограничения, связанные с трудностями парсинга сайтов с различной структурой. Несмотря на это, автоматизация позволит выполнять большее количество проверок достижимости уязвимых методов и сократить ручной труд специалистов.

Выводы

Полученные результаты будут использованы в исследовании по модернизации композиционного анализа программного обеспечения с целью повышения точности выявления уязвимостей. Разработанный алгоритм определения уязвимых методов будет автоматизирован и протестирован в рамках оценки точности выявления уязвимостей при использовании прототипа программного обеспечения, который будет разработан в рамках дальнейшего исследования.

Литература

1. Reachability analysis. Новый подход к SCA [Электронный ресурс]. – Режим доступа: https://pt-phdays.storage.yandexcloud.net/Puzankov_Artem_Reachability_Analysis_novyj_podhod_k_SCA_32aac85626.pdf (Дата обращения: 15.06.2025).
2. Zhang L. Vulnerability root cause function locating for java vulnerabilities //Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings. – 2024. – С. 444-446.