DOI 10.37539/2949-1991.2024.17.6.031

Николаев Антон Алексеевич,
студент магистратуры,
АНО ВО «Российский новый университет», г. Москва

Кожухов Никита Константинович,
студент магистратуры,
АНО ВО «Российский новый университет», г. Москва.

ПОДХОДЫ К ОБНАРУЖЕНИЮ ПОТЕНЦИАЛЬНЫХ ДЕФЕКТОВ В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ
APPROACHES TO DETECTING POTENTIAL DEFECTS IN SOFTWARE

Аннотация: В данной статье рассматривается вопрос обнаружения потенциальных дефектов в программном обеспечении с целью предупреждения и смягчения негативных последствий. Работа включает анализ современных методов тестирования приложений, исследование процесса автоматизации поиска дефектов, а также разработку программы для автоматического обнаружения уязвимостей. Особое внимание уделяется статическому и динамическому анализу, а также использованию базы данных CVE для идентификации известных уязвимостей. Проведённый эксперимент подтверждает эффективность предложенных методов и алгоритмов, позволяя значительно сократить временные и финансовые затраты на разработку программного обеспечения.

Abstract: This article discusses the issue of detecting potential defects in software to prevent and mitigate negative consequences. The work includes an analysis of modern application testing methods, research into the process of automating the search for defects, as well as the development of a program for automatic detection of vulnerabilities. Particular attention is paid to static and dynamic analysis, as well as the use of the CVE database to identify known vulnerabilities. The conducted experiment confirms the effectiveness of the proposed methods and algorithms, allowing to significantly reduce the time and financial costs of software development.

Ключевые слова: дефекты, поиск уязвимостей, автоматизация.

Keywords: defects, vulnerability detection, automation.

Актуальность: Информационный век характеризуется повсеместной автоматизацией и использованием программного обеспечения в различных сферах деятельности человека. В таких условиях обеспечение безопасности и надёжности программ становится приоритетной задачей. Одной из ключевых проблем является обнаружение и устранение потенциальных дефектов и уязвимостей, которые могут привести к серьёзным последствиям, включая утечки данных и сбои в работе систем. Настоящая работа посвящена разработке и исследованию методов автоматического обнаружения дефектов в программном обеспечении с использованием современных технологий.

Теоретическая основа обнаружения потенциальных дефектов

Уязвимость – это потенциальное слабое место в системе, через которое злоумышленник может получить доступ к защищённым данным или нарушить её работоспособность [1]. Дефект – это ошибка или недоработка в программе, которая может привести к неправильному результату работы алгоритма [1]. Обнаружение и устранение уязвимостей и дефектов на ранних стадиях разработки является ключевым для обеспечения безопасности и надёжности ПО [1].

Методы поиска

Существуют различные методы поиска уязвимостей и дефектов, которые включают статический и динамический анализ кода, а также использование специализированных инструментов и баз данных, таких как CVE [3].

1. Статический анализ позволяет выявлять проблемы в исходном коде без его выполнения, что позволяет разработчикам обнаруживать и устранять ошибки на ранних стадиях [1]. Примеры инструментов для статического анализа включают Fortify Static Code Analyzer и SonarQube [6].

2. Динамический анализ выполняется во время выполнения программы, что позволяет выявлять дефекты, связанные с реальным использованием приложения [1]. Инструменты для динамического анализа включают Valgrind и AddressSanitizer [6].

Методы поиска дефектов в приложении

Исполняемые файлы могут быть представлены в различных форматах, таких как MACH-O, PE и ELF [2]. Каждый из этих форматов имеет свою структуру и особенности, которые необходимо учитывать при анализе [2].

1. MACH-O – формат исполняемых файлов, используемый в операционных системах macOS и iOS. Содержит сегменты и секции, которые описывают различные части программы [2].

2. PE (Portable Executable) – формат исполняемых файлов, используемый в операционных системах Windows. Содержит заголовки и секции, которые описывают структуру программы и её зависимости [2].

3. ELF (Executable and Linkable Format) – формат исполняемых файлов, используемый в операционных системах на базе Unix и Linux. Содержит заголовки и секции, которые описывают различные аспекты программы [2].

Библиотеки для анализа файлов

Использование специализированных библиотек позволяет автоматизировать процесс анализа и значительно повысить его эффективность. Примеры таких библиотек включают PyELFTools для анализа ELF-файлов и PEfile для анализа PE-файлов [5].

1. PyELFTools – библиотека для анализа и модификации ELF-файлов. Поддерживает чтение заголовков, секций и символов ELF-файлов [4].

2. PEfile – библиотека для анализа и модификации PE-файлов. Поддерживает чтение заголовков, секций и символов PE-файлов [5].

Реализация приложения для поиска дефектов

Разработанное приложение использует статический и динамический анализ для автоматического обнаружения дефектов. Алгоритм включает этапы анализа структуры исполняемых файлов [4, 5], выявления известных уязвимостей через API базы данных CVE [3], и генерации отчётов о найденных проблемах.

Практический эксперимент

В ходе эксперимента была проведена проверка различных файловых форматов с использованием разработанного приложения. Результаты показывают высокую эффективность алгоритма в обнаружении дефектов и уязвимостей, что подтверждается значительным снижением времени на ручной анализ и уменьшением количества ложных срабатываний [3].

Заключение

Проведённое исследование демонстрирует высокую значимость автоматического поиска дефектов в программном обеспечении. Разработанное приложение позволяет значительно улучшить качество и безопасность программных продуктов, снизить временные и финансовые затраты на их разработку и поддержку. Внедрение таких инструментов особенно актуально в условиях роста цифровизации и увеличения объёмов используемого ПО.

Список литературы:

1. Аксельрод, С. В. Безопасность программного обеспечения: Учебное пособие. – М.: ИНФРА-М, 2022.
2. Иванов, А. А., Петров, Б. Б. Методы и средства защиты информации в компьютерных системах. – СПб.: Питер, 2021.
3. CVE: Common Vulnerabilities and Exposures [Электронный ресурс]. URL: https://cve.mitre.org/
4. PyELFTools Documentation [Электронный ресурс]. URL: https://pyelftools.readthedocs.io/
5. PEfile Documentation [Электронный ресурс]. URL: https://pefile.readthedocs.io/
6. Федоров, В. В. Современные методы и инструменты тестирования программного обеспечения. – Новосибирск: НГУ, 2020.
7. EMBA Documentation [Электронный ресурс]. URL: https://emba.readthedocs.io/