Отзыв

официального оппонента Кознова Дмитрия Владимировича на диссертацию Саргсяна Севака Сениковича «Методы оптимизации алгоритмов статического и динамического анализа программ», представленную на соискание ученой степени доктора технических наук по специальности 2.3.5 – математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей

Актуальность темы исследования

В современных условиях разрабатывается и используется всё более сложное программное обеспечение (ПО), и вопросы обеспечения его безопасности и выявления ошибок становятся все более значимыми. ПО разрабатывается для управления различными критическим инфраструктурами, активно развиваются телекоммуникационные системы, включающие в свой состав миллионы строк кода и оказывающиеся крайне чувствительными к наличию ошибок, создаются информационные системы для крупных предприятий, требующие целостности данных, широкое распространение Интернета остро ставит вопрос безопасности различных приложений и систем и т.д. Соответственно, требуются всё более результативные алгоритмы и методы обеспечения качества. При этом важную роль начинают играть различные интеграционные подходы, а также ориентация методов на конкретные бизнес-сценарии и удовлетворение различным технологическим требованиям (их программного окружения, целевого ПО и пр.). Таким образом, актуальность представленной диссертации, нацеленной на интеграцию статического и динамического анализа, а также на интеграцию различных методов обеспечения качества в рамках единой платформы, не вызывает сомнений.

Краткий обзор диссертационной работы

Представленная диссертационная работа включает введение, семь глав, заключение, список литературы из 271 наименований и девять приложений. Общий объем работы составляет 268 страниц, включая 56 рисунка 35 таблиц и 2 приложения (патенты, защищённые по результатам работы, и акты о внедрении). Автореферат диссертации, представленный на 43 страницах, адекватно отражает основное содержание диссертации.

Во введении обоснована актуальность исследования, определены цель и задачи работы, объяснена теоретическая и практическая значимость полученных результатов, а также приведены основные положения, выносимые на защиту.

В первой главе обосновывается актуальность исследований по обеспечению безопасности программного обеспечения. Рассматриваются существующие методы, их применение и ограничения, что позволяет выделить ключевые направления исследования: создание платформы для интеграции различных инструментов анализа программ; разработка средств поиска клонов и копий известных уязвимостей, сопоставление исходного и бинарного кода; выявление ошибок, связанных с форматными строками и

динамической памятью; оптимизация методов фазинга. Также анализируются требования к платформе анализа программ, формулируется её архитектура.

Во второй главе проведен обзор методов, усовершенствованию которых посвящен данное исследование – поиск клонов, анализ разницы двух версий кода для поиска ошибок, сопоставление исходного и бинарного кода ПО, утечки памяти и др.

В третьей глава содержится описание разработанных методов поиска клонов кода в исходных и бинарных файлах, включая основанные на них инструменты для поиска неисправленных ошибок и сопоставления исходного и бинарного кода.

В четвертой главе описываются методы обнаружения утечек памяти, проблемы некорректного использования динамической памяти и способы анализа помеченных данных.

В пятой главе описываются разработанные методам фазинга, в том числе основанные на комбинации фазинга с символьным выполнением и статическим анализом. Рассмотрены также методы фазинга для программ, работающих со структурированными данными и интерфейсными функциями.

В шестой главе дается описание платформы, интегрирующей представленные выше методы. Рассмотрены функциональные возможности платформы и приведены примеры ее применения.

В седьмой главе раскрывается практическая значимость работы, в том числе обобщены ошибки, найденные в различном ПО с помощью представленных методов.

В заключении сформулированы основные результаты работы, а также предложены возможные направления для дальнейших исследований.

Степень обоснованности научных положений

Автором диссертации проведен глубокий обзор существующих методов анализа кода, рассмотрены их недостатки и предложены пути их преодоления. В работе используются современные методы, включая теорию графов и символьное выполнение, что позволяет достичь высокого уровня точности при анализе программ. Методология исследования включает как статический анализ исходного кода, так и динамическое выполнение программ, что значительно расширяет возможности поиска ошибок и уязвимостей.

Результаты, выносимые на защиту

В диссертации на защиту представлены следующие результаты.

- Архитектура и прототип платформы анализа программ, обеспечивающие сбор артефактов для большого объема открытого ПО и информации об известных уязвимостях; единообразный подход комбинирования различных методов анализа кода в зависимости от конкретной задачи.
- Масштабируемые и точные методы нахождения клонов кода, основанные на поиске схожих подграфов максимального размера для графов зависимостей программ, построенных на основе промежуточных представлений исходного и бинарного кода.

- Метод сопоставления исходного кода и бинарных файлов на основе отладочной информации.
- Метод поиска утечек памяти для программ на C/C++ методом направленного символьного исполнения.
- Метод фазинга программ, генерирующий структурированные данные на основе специализированных автоматов БНФ грамматик, адаптируя шаблоны генерируемых программ в зависимости от их эффективности для увеличения покрытия кода.
- Метод фазинга интерфейсных функций, обеспечивающий возможность подготовки необходимых ресурсов для тестирования сложных сценариев использования нескольких функций в среде исполнения.
- Метод направленного фазинга для быстрой генерации входных данных с целью выполнения конкретных инструкций или фрагментов целевой программы, содержащий потенциальные уязвимости или дефекты.
- Метод интеграции статического анализа с фазингом, который применяет статический анализ для получения константных значений для генерации входных данных, покрывающих соответствующие ветви кода.

Оценка научной новизны

Основной новизной предлагаемых результатов является настраиваемая и конфигурируемая среда поиска уязвимостей. Фактически, имеющиеся на рынке средства предлагают «решения из коробки», что является весьма востребованным в индустрии подходом (часто разработчики не хотят возиться с незнакомым инструментом, а просто его запустить и получить пользу). Однако в данной предметной области, когда многое зависит от целевых проектов, платформ их разработки, приоритетов ошибок и пр., важно иметь настраиваемую, конфигурируемую среду поиска уязвимостей. Возможно, её будет труднее коммерциализировать, но польза от её реального внедрения может оказаться существенно больше, чем от «решений из коробки».

Соответственно, автор предлагает средства для различных сочетаний большого числа базовых алгоритмов, а также возможности по созданию различных итоговых сервисов по поиску уязвимостей на основе базовых. Такой подход (гибкая настраиваемая платформа тестирования) является традиционным для методов и инструментов ИСП РАН, под эгидой которого было выполнено это исследование. Но впервые была создана платформа, объединяющая большое количество разнообразных методов и, в частности, статические и динамические подходы к анализу программного кода.

Также в рамках работы предложен интересный набор целевых сервисов (бизнес-сценариев) использования базовых алгоритмов, например, применение поиска клонов для поиска копий уязвимостей, что весьма актуально для больших проектов (десятки и сотни миллионов строк кода и активное использование cut&paste).

Также новым являются средства увеличения производительности (в частности, распараллеливание) для увеличения производительности алгоритмов анализа кода, что позволяет, сохраняя точность, применять их для больших проектов.

Степень достоверности результатов

Результаты диссертационного исследования основаны на детально проработанных экспериментах, которые проводились на большом объеме синтетических тестов и реальных программных проектах. Полученные данные были сопоставлены с результатами существующих инструментов и выявлена более высокая точность и производительность. Достоверность выводов также подтверждена успешным внедрением предложенных методов в разные компании. В целом, важно подчеркнуть, что предложенная платформа способна работать с проектами разных видов – информационными системами, Интернет-приложениями, телекоммуникационным ПО, операционными системами и др., – что говорит об устойчивости предложенной концепции, удачной архитектуре платформы и эффективности предложенных методов.

Практическая значимость работы

Представленные методы и инструменты обладают высокой практической значимостью, поскольку могут быть использованы для анализа большого объема кода (правда, в серверном режиме, на высокопроизводительных ресурсах, в backend-режиме). Разработанные подходы осуществляют достаточно точный анализ кода и обладают приемлемой производительностью. Разработанная в диссертации платформа покрывает многие из требований ГОСТ Р 56939-2016 и «Методики выявления уязвимостей и недекларированных возможностей в программном обеспечении» ФСТЭК Российской Федерации. Разработанные методы были внедрены в несколько компаний и используются в процессе разработки ПО.

Общая характеристика публикаций автора

В рамках подготовки диссертации было опубликовано достаточное количество статей в журналах из списка ВАК по специальности 2.3.5: журнал «Труды ИСП РАН» (8 публикаций), журнал «Программирование» (3 публикации), 1 статья из журнала «IEEE Accesss» (Q1 по SJR), а также патенты и свидетельства о регистрации программ для ЭВМ (7 штук).

Недостатки и замечания

Несмотря на высокую научную значимость работы, можно отметить следующие замечания.

1) Некоторые разделы работы требуют более детального описания технической реализации предложенных методов. Также не хватает описания применимости представленных методов для анализа ПО, созданного на других языках программирования, отличных от C/C++.

2) В разделе 3.1.1 при сравнении инструментов поиска клонов кода на реальных проектах (таблица 18) представлено только сравнение времени работы этих инструментов, но не сравниваются найденные клоны, что не позволяет получить полное представление о качестве разработанного инструмента.
3) В разделе 3.2.5 отсутствует оценка времени работы разработанного инструмента binCCD на реальных проектах.
4) Алгоритм поиска ошибок использования памяти после её освобождения (раздел 4.1.4) способен обнаруживать ошибки в однопоточных программах, однако его применение для многопоточных программ не может обеспечить нахождение всех ошибок такого типа.
5) Мне кажется, что как в самой платформе, так и в тексте работы следовало бы чётче разграничить базовый уровень – основные алгоритмы статического и динамического анализа кода, и целевые сервисы по поиску уязвимостей, созданные и создаваемые на их основе. Дело в том, что последние часто сильно зависят от специфических бизнес-сценариев, актуальных для целевых систем, которые могут сильно варьироваться от одного большого проекта к другому. И все такие бизнес-сценарии в рамках универсальной платформы не предусмотреть, а если к этому стремиться, то в это будет методологической ошибкой. Кроме того, реализация различных бизнес-сценариев вовсе не ограничивается лишь комбинированием базовых методов.
6) Использование в работе средств, работающих с деревом программы, хоть и является гарантом точности, но очень сильно влияет на производительность алгоритмов на больших проектах (даже для относительно небольшого индустриального ПО с объёмом исходного кода в 500 Мб). Средства распараллеливания способны тут лишь частично решить проблему. С другой стороны, высокая точность не всегда нужна в реальных контекстах. В общем, интерес представляют альтернативные подходы в этом направлении.
7) В исследовании крайне незначительно затронуты нейросети для решения выбранных задач: почти не исследуются существующие методы, нет собственных исследований в этом направлении. А между тем этот подход может оказаться весьма полезным для уменьшения вычислительной сложности алгоритмов анализа кода, то есть это перспективное направление продолжения представленных исследований.

Приведенные замечания не влияют на общую положительную оценку работы.

Заключение

Диссертация Саргсяна Севака Сениковича представляет собой завершенное научное исследование, содержащее существенные научные и практические результаты в области

оптимизации алгоритмов анализа программ. Основные результаты диссертации полностью и своевременно опубликованы.

Диссертация удовлетворяет всем требованиям ВАК, предъявляемым к диссертационным работам на соискание ученой степени доктора технических наук по специальности 2.3.5 – математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей, а Саргсян Севак Сеникович заслуживает присуждения ему указанной степени.

Профессор кафедры системного программирования Санкт-Петербургского государственного университета, доктор технических наук _____ /Кознов Дмитрий Владимирович/

24.10.2024