**Отзыв**
**официального оппонента Кореньков Владимир Васильевич**
**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**

Актуальность темы работы С.С. Саргсяна не вызывает сомнений. В условиях роста сложности программных систем и увеличения количества уязвимостей вопросы обеспечения безопасности ПО приобретают особую важность. Разработка методов оптимизации статического и динамического анализа программ является актуальной задачей не только для научных исследований, но и для практической разработки ПО. А предлагаемая автором платформа для сбора большого объема артефактов открытого ПО и возможность универсальной комбинации нескольких методов анализа в зависимости от задачи могут стать ключом к решению множества проблем безопасности.

**Содержание диссертационной работы**

Диссертация С.С. Саргсяна состоит из введения, семи глав, заключения и списка литературы, включающего 271 наименование, а также девяти приложений. Работа изложена на 268 страницах и содержит 56 рисунков и 35 таблиц.

**Во введении** обоснована актуальность исследования, сформулированы цель и задачи работы, выделены научная новизна и практическая значимость полученных результатов, а также представлены ключевые положения, выносимые на защиту.

**Первая глава** посвящена исследованию методов обеспечения безопасности ПО. Автор проводит сравнительный анализ существующих методов анализа ПО и их ограничений для выделения основных направлений исследований. В результате анализа выделяются следующие основные направления исследований: создание интеграционной платформы для анализа программ, средства поиска клонов кода и копий известных уязвимостей, сопоставление исходного и бинарного кода, а также оптимизация методов фаззинга. Для предложенной платформы также рассматриваются требования, формулируется концепция и описываются ключевые элементы архитектуры, которые обеспечивают обработку большого объема ПО и позволяют комбинировать методы анализа кода.

**Во второй главе** представлен обзор существующих методов статического и динамического анализа программ. Рассмотрены подходы к поиску клонов кода, сравнению исполняемых файлов и анализу изменений между версиями ПО. Также описаны методы обнаружения ошибок, связанных с использованием динамической памяти, и методы фаззинг-тестирования, которые позволяют выявить входные данные, приводящие к сбою программы.

**Третья глава** посвящена разработанным методам поиска клонов кода в исходных и бинарных файлах. Поиск осуществляется на основе выявления схожих подграфов в графах потока управления и данных. В этой главе также описаны инструменты, созданные с использованием технологии поиска клонов. В частности, рассматриваются инструменты для обнаружения неисправленных ошибок и сопоставления исходного и бинарного кода.

**В четвертой главе** описываются методы обнаружения проблем, связанных с некорректным использованием динамической памяти. Представлен двухэтапный метод поиска утечек памяти, который сочетает статический анализ с направленным символьным выполнением.

**В пятой главе** представлены разработанные методы фаззинга для различных сценариев. Подробно описаны реализованные автором методы комбинирования фаззинга с символьным выполнением, а также его сочетание со статическим анализом. Приведено описание метода генерации структурированных данных и его интеграции с инструментом фаззинга.

**В шестой главе** представлено детальное описание разработанной интеграционной платформы. Она позволяет объединять доступные методы анализа программ в зависимости от конкретной задачи. Приведены примеры использования различных функциональных возможностей платформы для решения разнообразных задач.

**В седьмой главе** обобщены ошибки, выявленные с помощью разработанных методов анализа. Описана критичность найденных уязвимостей, обнаруженных в значимых проектах и потенциально затрагивающих всех пользователей интернета.

**В заключении** сформулированы основные результаты работы и предложены направления для дальнейших исследований.

Содержание диссертации полностью отражено в автореферате объемом 43 страницы. Диссертационную работу автор выполнил в Государственном образовательном учреждении высшего профессионального образования Российско-Армянский (Славянский) университет.

**Новизна исследования**

Новизна диссертационной работы заключается в разработанной архитектуре и экспериментальном прототипе платформы анализа программ, позволяющей собирать артефакты и информацию об известных уязвимостях в открытом ПО. Кроме того, разработаны и реализованы новые методы анализа программ, которые интегрированы в платформу и могут быть комбинированы в зависимости от задачи. Разработаны методы поиска клонов кода, основанные на поиске схожих подграфов в графе зависимостей программ. Разработаны методы поиска ошибок, связанных с неправильным использованием динамической памяти, в частности, поиска утечек памяти для языков C/C++, который анализирует потоки управления и данных программы, после чего производит проверку выполнимости путей ошибок с помощью символьного выполнения. Разработан метод, позволяющий производить сопоставление исходного кода с бинарным. Также разработано множество методов фаззинга программ для разных сценариев, включая:

генерацию сложно структурированных данных на основе БНФ-грамматик; фаззинг интерфейсных функций для тестирования сложных сценариев комбинирования; направленный фаззинг для генерации данных, позволяющих выполнять конкретные (уязвимые) фрагменты программы; интеграцию статического анализа с фаззингом для генерации входных данных, покрывающих невыполненные ветви кода.

**Степень обоснованности научных положений и достоверность результатов**

Для выявления основных направлений исследования автором проведен глубокий анализ существующих методов анализа кода. Проведен сравнительный анализ их недостатков, на основе которого выбраны пути их преодоления. Для решения существующих проблем используются методы теории графов, современные методы символьного выполнения и фаззинга. В ходе исследования применяются методы статического и динамического анализа кода. Такой подход расширяет возможности поиска ошибок и уязвимостей, что на практике подтверждается множеством найденных ошибок в открытом ПО.

Достоверность результатов обоснована детально проработанными экспериментами. Эксперименты проводились на реальных проектах, и полученные результаты были сопоставлены с результатами существующих инструментов. Результаты, полученные автором, демонстрируют высокую точность и эффективность разработанных методов, а достоверность выводов подтверждается внедрением реализованных инструментов в цикл разработки ПО в различных компаниях.

**Практическая значимость работы**

Объединяющая платформа обладает высокой практической значимостью, поскольку позволяет собрать большой объем открытого ПО и известных уязвимостей для улучшения безопасности широко применяемых систем. Благодаря разработанной платформе и методам возможен точный и масштабируемый анализ открытого ПО для своевременного обнаружения критических ошибок и уязвимостей. Это, в свою очередь, повышает надежность и устойчивость повседневно используемых программных систем.

Разработанные методы и инструменты внедрены в жизненный цикл разработки ПО в различных компаниях для обеспечения выполнения многих требований ГОСТ Р 56939-2016 и "Методики выявления уязвимостей и недекларированных возможностей в программном обеспечении" ФСТЭК Российской Федерации.

**Недостатки и замечания**

Несмотря на высокую научную значимость и качество работы, можно выделить следующие замечания.

1) Методы машинного обучения все чаще применяются в задачах анализа кода; однако из предлагаемой архитектуры платформы не очевидно, рассчитана ли она на будущую интеграцию этих методов для анализа кода.

2) В разделе 3.3 рассматривается задача сопоставления исходного и бинарного кода. Рассматривается только случай, когда предоставленный исходный код компилируется, тогда как существует много случаев, когда исходный код доступен частично или не компилируется. Было бы полезно предложить метод решения этой задачи для всех случаев.
3) Не всегда обосновывается выбор операционных систем, программ и уязвимостей, участвующих в тестировании.
4) Замечание по оформлению: положения, выносимые на защиту, и научная новизна внесена в общий пункт, хотя они имеют разный смысл. В этом пункте перечислены важные результаты и методы без уточнения, где предложен новый метод, где модифицированы или развиты существующие методы или подходы.
5) В автореферате на стр. 8 в разделе "Личный вклад" указано, что статьи под номерами [1-16] в списке литературы "полностью написаны автором лично". Это требует уточнений, поскольку не совсем понятно, что имеет в виду автор, так как 14 из 16 указанных статей опубликованы в соавторстве.

Приведенные замечания не снижают общую положительную оценку работы.

**Заключение**

Диссертация С.С. Саргсяна представляет собой самостоятельное и завершенное научное исследование, выполненное на высоком научном уровне. Сформулированная в диссертации цель и поставленные задачи обладают высокой научной актуальностью, а полученные практические результаты решают широкий спектр задач в области оптимизации алгоритмов анализа программ. Основные результаты диссертации полностью и своевременно опубликованы.

Диссертация С.С. Саргсяна полностью удовлетворяет всем требованиям ВАК, предъявляемым к 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.

Д-р техн. наук,
Научный руководитель
Лаборатории информационных
Технологий имени М.Г.Мещерякова,
Международная межправительственная
научно-исследовательская организация
Объединенный институт
Ядерных исследований

В.В. Кореньков

28.11.2024