МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Московский институт электронной техники»
Институт системной и программной инженерии и информационных технологий (СПИНТех)

Уманский Александр Александрович

Магистерская диссертация
по направлению 09.04.04 «Программная инженерия»

Исследование и разработка методики и алгоритма нахождения НДВ ПО с известной моделью нарушителя.

Студент _____
Уманский А.А.
Руководитель,
канд. техн. наук, доц. _____
Кононова А.И.

Москва, г. Зеленоград — 2021

Содержание

Список сокращений и условных обозначений

НДВ	Недекларированные возможности
ПМ	Программный модуль
БД	База Данных
ИСПДН	Информационная система персональных данных
ПО	Программное обеспечение
ЯП	Язык программирования
ЖЦ	Жизненный цикл
GUI	Graphical User Interface
IDE	Интегрированная среда разработки
JSON	Формат описания структур данных в текстовом виде ключ $\rightarrow$ значение
PID	Уникальный идентификатор процесса в ОС
TDD	Test-driven development
	Программный модуль анализа на недекларированные возможности

Словарь терминов

Кроссплатформенный: программа, которая может запускаться на различных операционных системах и/или архитектурах процессоров

Программная закладка: подпрограмма, либо фрагмент исходного кода, скрытно внедренный в исполняемый файл

Динамическая трасса: дерево вызванных программой функций во время конкретного ее исполнения

Статическая трасса: дерево функций программы, которые объявлены для вызова

Отладчик: программа, в контексте которой запускается другая программа для локализации и устранения ошибок в контролируемых условиях

Отладка: процесс локализации и устранения ошибок программы в контролируемых условиях

Удаленная отладка: процесс отладки программы, запущенной вне контекста отладчика

Препроцессор: программа-макропроцессор, обрабатывающая специальные директивы в исходном коде и запускающаяся до компилятора

Препроцессирование: процесс обработки исходного кода препроцессором

Открытое ПО: ПО с открытым исходным кодом, который доступен для просмотра, изучения и изменения

Сериализация: процесс перевода определеного типа данных программы в некоторый формат

Десериализация: процесс перевода данных, находящихся в некотором формате, во внутренний тип данных программы

Скрипт: программа, обычно на интерпретируемом языке программирования, выполняющая конкретное действие

Сигнатура функции: объявление функции, в которое входит имя функции, количество входных параметров и их тип

Сборка: процесс компиляции, линковки и публикации программного обеспечения из исходных кодов

Рефакторинг: процесс улучшения кода без введения новой функциональности. Результатом является чистый код с улучшенным дизайном

Релизная сборка: сборка программы происходит без отладочных символов,

обычно с использованием техник оптимизации кода

Терминал: то же, что и консоль

Антиотладка: набор методов детектирования отлаживаемой программы окружения отладки и препятствование ей

Мультитаскинг: возможность программы или операционной системы обеспечивать возможность параллелльного исполнения задач

Сверхвысокоуровневый ЯП: классификация языков программирования, к данной категории относятся языки программирования, позволяющие описать задачу не на уровне «как нужно сделать», а на уровне «что нужно сделать»

Source-to-source: Компиляция исходного кода некоторого языка в исходный код другого языка. Во время компиляции языка данным способом может происходить несколько итераций преобразования, пока последний язык в цепочке преобразований не будет скомпилирован в машинный код или интерпретирован

Актуальность исследования:

Подтверждение безопасности программного обеспечения – важный этап в продвижении программного продукта.

Сертификация не является универсальным способом решения всех существующих проблем в области информационной безопасности, однако сегодня это единственный реально функционирующий механизм, который обеспечивает независимый контроль качества средств защиты информации. При грамотном применении механизм сертификации позволяет достаточно успешно решать задачу достижения гарантированного уровня защищенности автоматизированных систем.

Отсутствие недекларированных возможностей в скомпилированном объектном файле является ключевым аспектом сертификации ПО. Сертификация программного обеспечения необходима для подтверждения требований заказчика к защите информации, к выполнению функциональных и технических задач и к обеспечению работы ПО в целом.

Проблемная ситуация в области объекта исследований:

Большое количество ложноположительных и ложноотрицательных срабатываний при проведении анализа ПО на НДВ не отвечает современным требованиям безопасности.

Причины сложившейся ситуации:

1) у современных интерфейсов, как программных, так и пользовательских, большая «поверхность» для атаки;
2) современные компиляторы производят большое количество изменений кода, таких как: встраивание тел функций, разворачивание циклов, объединение функций;
3) требования и методики нахождения НДВ в ПО разрабатывались во времена с другим уровнем и сложностью технологий;
4) новые атаки на ПО появляются постоянно, нет гибкого механизма их диагностирования, который соответствовал бы текущему уровню развития технологий.

Объект исследования:

Существующие методики поиска НДВ в программном обеспечении.

Предмет исследования:

Программное обеспечение.

Цель исследования: уменьшение количества ложноположительных и ложноотрицательных срабатываний поиска НДВ.

Задачи исследования:

1) анализ возможностей нарушителя;
2) анализ критичности и применимости воздействия нарушителя на ПО;
3) создание методики выбора инструментов для проверки воздействия нарушителя на ПО;
4) разработка алгоритма поиска НДВ;

Раздел 1. Аналитический обзор существующих методов нахождения НДВ

1.1 Анализ существующих подходов к нахождению НДВ

Процедура нахождения НДВ состоит из следующих этапов:

1) готовность документации ПО, доступность исходных текстов;
2) опредление объема исходных текстов, подлежащих анализу;
3) обращение заявителя в испытательную лабораторию с собранной информацией;
4) анализ документации;
5) разработка «Программы и методик проведения сертификационных испытаний»;
6) проведение испытаний;
7) экспертиза результатов.

Сертификация должна выявить присутствие в исполняемом файле недекларированных возможностей, которые могут являться как злым умыслом разработчиков компилятора, линкера и других вспомогательных программ, так и методами оптимизации ПО, которые применяются для более рационального потребления ресурсов программой.

1.2 Анализ актуальности существующих нормативных документов для исследования ПО на НДВ

1.3 Анализ имеющегося программного обеспечения для исследования ПО на НДВ

1.4 Постановка задач диссертации

1.5 Выводы по главе 1

Раздел 2. Формализованное представление процесса разработки

2.1 Декомпозиция поставленной задачи для создания методики и алгоритма нахождения НДВ

2.2 Этапы нахождения НДВ с применением автоматизации

2.2.1 Анализ условий, в которых оперирует нарушитель

2.2.2 Анализ возможностей нарушителя в заданных условиях

2.2.3 Создание модели нарушителя

2.2.4 Определение возможных мест проявления НДВ

2.2.5 Проверка на эксплуатируемость возможных мест проявления НДВ

Раздел 3. Программная реализация и экспериментальное подтверждение результатов исследования

Раздел 4. Экспериментальные исследования разработанной методики и алгоритма нахождения НДВ ПО с известной моделью нарушителя

Заключение

Результатом выпускной квалификационной работы стала рабочая версия программного модуля анализа программ на языках C/C++ на недекларированные возможности. позволил унифицировать и ускорил процесс исследования программного обеспечения на НДВ. Уменьшение фрагментации программ по анализу ПО на наличие НДВ позволяет не тратить время программистов на написание анализатора под конкретный продукт, что положительно сказывается на продуктивности всей команды разработчиков.

В рамках выпускной квалификационной работы были решены задачи:

1) исследование предметной области ;
2) сравнительный анализ существующих программных решений;
3) выбор языка и среды разработки;
4) разработка схемы данных ;
5) разработка схемы алгоритма ;
6) программирование ;
7) отладка и тестирование ;
8) разработка документации к .

В заключение автор выражает благодарность и большую признательность научному руководителю Кононовой Александре Игоревне за поддержку, помощь, обсуждение результатов и научное руководство.