МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

УЧЕБНО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Для выполнения курсовой работы по дисциплине «Безопасность программного обеспечения»

для студентов специальности 10.03.01 – Информационная безопасность,
10.05.03 – Информационная безопасность автоматизированных систем,
10.05.04 – Информационно-аналитические системы безопасности

Факультет - Безопасности
Кафедра - Комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

Разработчик:
доцент каф. КИБЭВС К.С. Сарин

Томск 2022

Цель курсовой работы: Обрести навыки использования методов обфускации программ для противодействия от исследования и нахождения уязвимых мест.

Теоретический материал.

Обфускация - защита программ от воссоздания исходного кода (декомпиляции) и незаконного использования, нарушения авторских прав программистов.

С помощью запутывания можно перемешать в программе куски кода или действия так, что логика работы становится совершенно непонятной. Кроме того, при запутывании могут вставляться новые куски неисполняемого (неиспользуемого) кода, а существующие блоки кода могут быть модифицированы таким образом, чтобы они использовались в нескольких частях программы одновременно.

Методы запутывания программ. Запутывающие преобразования можно разделить на несколько групп в зависимости от того, на трансформацию какой из компонент программы они нацелены

1. Преобразования *форматирования*, которые изменяют только внешний вид программы. К этой группе относятся преобразования, удаляющие комментарии, отступы в тексте программы или переименовывающие идентификаторы.
2. Преобразования *структур данных*, изменяющие структуры данных, с которыми работает программа. К этой группе относятся, например, преобразование, изменяющее иерархию наследования классов в программе, или преобразование, объединяющее скалярные переменные одного типа в массив.
3. Преобразования *потока управления* программы, которые изменяют структуру её графа потока управления, такие как развёртка циклов, выделение фрагментов кода в процедуры, и другие.

Преобразования форматирования.

- удаление всех комментариев в коде программы или изменение их на дезинформирующие;
- удаление различных пробелов, отступов, которые обычно используют для лучшего визуального восприятия кода программы;
- замена имен идентификаторов (имен переменных, массивов, структур, хешей, функций, процедур и т.д.) на произвольные длинные наборы символов, которые трудно воспринимать человеку;
- добавление различных лишних (мусорных) операций;
- изменение расположения блоков (функций, процедур) программы, таким образом, чтобы это никак не повлияло на ее работоспособность.

Преобразования структур данных.

1. Обфускация хранения, заключается в трансформации хранилищ данных, а

также самих типов данных (например, создание и использование необычных типов данных, изменение представления существующих и т.д.);
2. Обфускация соединения - это соединение независимых данных или разделение зависимых. Например, разбивка массивов на несколько частей, склеивание нескольких массивов, уменьшение и увеличение размерности массивов (приведение массивов к плоскому или наоборот многомерному представлению);
3. Обфускация переупорядочивания, заключается в изменении последовательности объявления переменных, внутреннего расположения хранилищ данных, а также переупорядочивании методов классов, массивов (использование нетривиального представления многомерных массивов), определенных полей в структурах и т.д.

Преобразование потока управления.
1. Внесение недостижимого кода. Недостижимый код никогда не выполняется, данное преобразование влияет только на размер запутанной программы, но не на скорость её выполнения.
2. Внесение мертвого кода. Мертвый код в программе выполняется, но его выполнение никак не влияет на результат работы программы.
3. Внесение избыточного когда. Избыточный код, в отличие от мёртвого кода выполняется, и результат его выполнения используется в дальнейшем в программе, но такой код можно упростить или совсем удалить, так как вычисляется либо константное значение, либо значение, уже вычисленное ранее.
4. Переплетение функций. Идея этого запутывающего преобразования в том, что две или более функций объединяются в одну функцию. Списки параметров исходных функций объединяются, и к ним добавляется ещё один параметр, который позволяет определить, какая функция в действительности выполняется.
5. Развертка циклов. Развёртка циклов заключается в том, что тело цикла размножается два или более раз, условие выхода из цикла и оператор приращения счётчика соответствующим образом модифицируются. Если количество повторений цикла известно в момент компиляции, цикл может быть развёрнут полностью.

Задание на курсовую работу.

1. Выбрать три метода обфускации программ, по одному каждого типа: преобразования форматирования, преобразования структур данных, преобразования потока управления. Согласовать методы с преподавателем.
2. Разработать простейшую программу на языке высокого уровня (C#, C++) которая, например, проводить сортировку массива, решение квадратного уравнения и т.д.
3. Перед запуском программы вставьте проверку на правильно введённый

серийный номер или пароль. Если пароль введен правильно, то программа выполняет свои задачи, если нет – то программа выдает сообщение об ошибке и закрывается.

4. К коду программы, отвечающему за проверку серийного номера, примените методы обфускации, выбранные в п.1.
5. Откомпилируйте полученную программу и передайте исполняемый модуль для исследования одногруппнику. В свою очередь, возьмите у одногруппника его исполнимый модуль для исследования.
6. С помощью дизассемблера-отладчика (например, IDA-PRO, Win32 DASM), попробуйте провести анализ исполняемого модуля и обойти ввод пароля (серийного номера).
7. Исправьте исполняемый модуль одногруппника с помощью Hex редактора, таким образом, чтобы программа запускалась в обход серийного номера.
8. Зафиксируйте в отчете курсовой работы примененные вами методы обфускации, а также методы, которые применил одногруппник.
9. Опишите в отчете ваши действия по анализу программы для обхода серийного номера.
10. Составьте доклад по проделанной работе и защитите у преподавателя.

Литература

1. Макарова Н.В., Волков В.Б. Информатика: учебник для вузов. - СПб. : ПИТЕР, 2012. - 576 с.
2. Пирогов В. Ю. Ассемблер и дизассемблирование. - СПб. : БХВ-Петербург, 2006. - 447 с.
3. Градский Д. Ю. Методы обфускации кода // Научный электронный журнал «Оригинальные исследования». – 2020 – №5. – С. 177-180.
4. Носов А. С. Анализ методов защиты программных кодов с помощью обфускации / А. С. Носов, Н. Е. Губенко. – Донецк: ДонНТУ, 2013
5. Ивченкова Ю. С. Виды и способы обфускации / Ю. С. Ивченкова, М. К. Савкин// Электронный журнал: наука, техника и образование – 2016 – №2(6). – С. 60-66.
6. Казанцева А. И. Методы обфускации программного кода / А. И. Казанцева, Т. А. Маркина // Сборник трудов viii научно-практической Конференции молодых ученых «вычислительные системы и сети (майоровские чтения)». – Санкт-Петербург: Университет ИТМО, 2017 – С. 39-42