# Лабораторная работа №8 Дизассемблирование приложений

1. Цель работы
   1. Научиться применять дизассемблеры для изучения и модификации ПО.
2. Литература
   1. Касперски К. Учимся анализировать программы для x86 с нуля / К. Касперски, Ю. Язев. – Текст: электронный // xakep.ru : [сайт] – 2022. – 7 декабря – URL: https://xakep.ru/2022/12/07/nezumi-book-excerpt/ – Режим доступа: свободный.
3. Подготовка к работе
   1. Повторить теоретический материал (см. п.2).
   2. Изучить описание лабораторной работы.
4. Основное оборудование
   1. Персональный компьютер.
5. Задание
   1. Изучение дизассемблированного кода в Visual Studio
      1. Создать в Visual Studio два консольных приложения на языке С++, вычисляющих в цикле со счетчиком сумму чисел от 1 до n (вводится пользователем). В первом приложении переменная-счетчик должна быть объявлена в разделе for, во втором — перед циклом. В режиме отладки включить окно «Дизассемблированный код» (меню Отладка — Окна — Дизассемблированный код), в контекстном меню которого отметить галочками все пункты «Показать ...» кроме «Показать байты кода». Сравнить дизассемблированный код разработанных приложений.
   2. Применение анализаторов raw-данных

Приложение запрашивает у пользователя пароль для того, чтобы можно было работать с программой в определенном режиме. В случае, если пароль корректен, программа выводит сообщение, что пароль корректен, иначе сообщает о некорректных данных. Требуется подобрать пароль для приложения, проанализировав его байт-код с помощью онлайн-редактора hexed.it.

* + 1. Загрузите файл приложения: https://github.com/ReyRom-Edu/SYSPR/blob/main/Resources/TryToCrackMe.zip
    2. Откройте сайт hexed.it. Загрузите exe-файл при помощи кнопки «Открыть файл». В главной рабочей области окна отобразится содержимое файла в шестнадцатеричном виде
    3. Для решения задачи следует при помощи панели поиска анализировать уже известные строковые данные, и изучать информацию, расположенную рядом с ними.
  1. Анализ дизассемблированного кода
     1. Откройте Visual Studio. Откройте Средства – Командная строка – Командная строка разработчика
     2. Перейдите в каталог, в котором располагается exe-файл анализируемого приложения при помощи команды cd путь.
     3. С помощью утилиты disasm извлеките ассемблерный код программы в файл. Воспользуйтесь командой:

dumpbin /disasm TryToCrackMe.exe > disasm.txt

* + 1. Открыть полученный файл в текстовом редакторе. Проанализировать код дизассемблированного приложения.
    2. С помощью поиска определить фрагмент кода, в котором выполняется проверка корректности введенного пароля. Для этого выполните поиск обнаруженной ранее строки в дизассемблированном коде. Иногда при дизассемблировании явных значений строк найти невозможно, в таком случае используйте команду для извлечения данных программы.

dumpbin /rawdata TryToCrackMe.exe > data.txt

В полученном файле найдите целевую строку как в п.5.2.3, определите ее адрес на основе начального байта и значений слева от шестнадцатеричных данных. В файле disasm.txt ищите определенный вами адрес.

* + 1. Восстановить алгоритм работы найденного фрагмента и определить, что нужно изменить в ассемблерном коде, чтобы при вводе любого пароля пользователю выводилось приветствие.
  1. Изменение кода приложения

Проверка результата сравнения строк выполняется при помощи команды test eax, eax, но если команду test заменить на команду xor, результат сравнения всегда будет положительным, вне зависимости от введенного пароля.

Обратите внимание на двухзначные шестнадцатеричные значения слева от команд в дизассемаблированном коде. Каждое из этих значений соответствует байту кода команды.

* + 1. Определите код команды xor при помощи поиска по файлу
    2. В редакторе hexed.it найдите последовательность байт, соответствующую необходимому вызову команды test, и замените байт команды test на значение байта для команды xor.
    3. Сохраните файл с измененным исходным кодом и протестируйте работу измененной программы.

1. Порядок выполнения работы
   1. Выполнить все задания из п.5.
   2. Ответить на контрольные вопросы.
2. Содержание отчета
   1. Титульный лист
   2. Цель работы
   3. Ответы на контрольные вопросы
   4. Вывод
3. Контрольные вопросы
   1. Что такое «дизассемблирование»?
   2. Для чего выполняется дизассемблирование программного кода?
   3. Что такое «дизассемблер»?
   4. Какие существуют программы-дизассемблеры?
   5. Как открыть окно дизассемблированного программного кода в Visual Studio?
   6. Для чего выполняется обфускация кода?