Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра Обчислювальна техніка та програмування

Звіт з лабораторной роботи з дисциплиі « Реверсне програмування »

Лабораторна робота 2

Виконав ст.гр. КІТ–36

Надірян Г.О.

Перевірив

Челак В.В.

**Харків 2019**

**Лабораторна робота 2**

**Тема**: Впровадження коду в ехе-файл в кінець секції без додавання API-функцій

##### Мета: Набути практичних навичок виправлення помилок в програмному забезпеченні, яке знаходиться в експлуатації; створення нової функціональності, використовуючи неявне в експлуатації програмне забезпечення для платформи х64 в середовищі masm64.

##### Постановка задачі

1.Написати програму в середовищі masm64 відповідно до першої частиною варіанта завдання. Вивести час виконання рівняння. Результати вивести через функцію MessageBox. Програму і результати привести в звіті.

2. Підрахувати вільна кількість байтів для здійснення впровадження.

3. Виправити раніше написану програму в відладчик x64Dbg відповідно з другою частиною варіанта завдання. А також змінити в первісному ехе-файлі виведену інформацію (наприклад, в функції MessageBox). Привести в цій функції своє прізвище. Для цього використовувати таблицю ASCII # 1 251 (наприклад, http://foxtools.ru/ASCII#1251). При виконанні операцій множення і ділення використовувати логічні команди. Вивести час виконання рівняння. Скріншоти виправленої програми та виведене вікно привести в звіті.

**Індивідуальне завдання**

Написати в середовищі masm64 на асемблері програму обчислення виразу:

*22d/b; —> 8d/b – 32d/c;*

**Код програми**

include win64a.inc

.data ; директива начала сегмента данных

a1 dq 1 ; резервирование в памяти 8 байтов для переменной Х

b1 dq 2 ;

c1 dq 64 ;

d1 dq 4

res1 dq ?

titl db "Вывод через функцию MessageBox",0; название упрощенного окна

st1 dq ?,0 ; буфер выведения сообщения.

ifmt db "Дано уравнение:",10,10,"22d/b - > 8d/b-32d/c",10,

"a = %d",10,"b = %d",10,"c = %d",10,"d = %d",10,"res = %d",10,10,

"Время выполнения = %d тиков",10, ;

"Автор программы: Надирян Г.О., КИТ-36",0 ;

.code ; директива начала сегментa команд

WinMain proc

sub rsp,28h; cтек: 28h=32d+8; 8 — возврат

mov rbp,rsp

rdtsc

xchg rax,r15

mov rax,8 ;8

mul d1 ;8\*4=32

div b1 ;32/2=16

mov res1,rax ; res1=16

rdtsc

sub rax,r15

invoke wsprintf,ADDR st1,ADDR ifmt,a1,b1,c1,d1,res1,rax

invoke MessageBox,0,addr st1,addr titl,MB\_ICONINFORMATION;

invoke ExitProcess,0 ;возвращение упр. ОС и освобожд. ресурсов

WinMain endp

end

Результат виконання програми наведено на рис. 1.

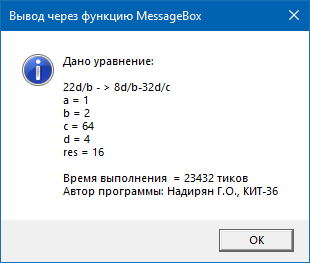


Рисунок 1 - Результат виконання

Дії щодо впровадження найпростішого коду (без додавання API-функцій) в ехе-файл наступні:

Спочатку встановлюємо курсор на першу команду за адресою 00007FF653FC100B і двічі натискаємо на ліву кнопку миші.(рис.2.)

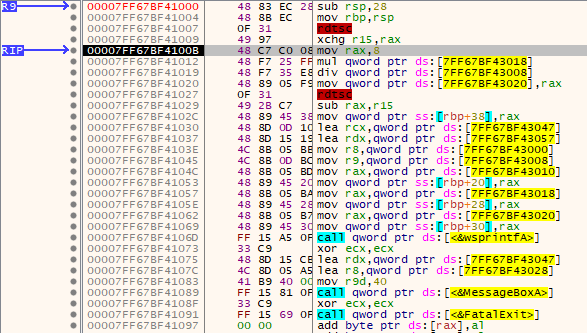


Рисунок 2 – адреса першої команди

Вводимо порожню команду nop Для того, щоб відбулося автоматичне заповнення командами nop всіх байт знищуваної команди необхідно поставити галочку “Зберігати розмір” у спливаючому вікні. (рис.3.)

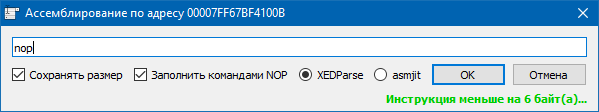


Рисунок 3 - Вводимо порожню команду nop

##### Вставляємо команду *jmp* на адресу початку нового фрагмента в кінці коду. Ставимо курсор після останньої команди *call qword ptr ds: [<& FatalExit>]*, яка знаходится за адресою *00007FF67BF41091*. (рис.4.) Копіюємо адресу вільної комірки (адреса *00007FF67BF41097*) Після цього, ставимо курсор в початок кода після команди *xchg rcx, rax* тобто на першу порожню команду *nop*. (рис.5.). Двічі натискаємо праву кнопку миші і у вікні вводимо слово *jmp*, потім пробіл і вставляємо адресу (рис.5.). Дописуємо необхідний код (рис.5.) на який будемо переходити командою *jmp*.

|  |  |  |
| --- | --- | --- |
| 00007FF67BF41027 | 0F 31 | rdtsc |

##### Останньою командою в кінці нового фрагмента повинна бути команда jmp з адрессою повернення - адресною першої незміненій на початку програми команди. (рис.5.).

В результаті, новий блок коду буде мати вигляд (рис.5.)

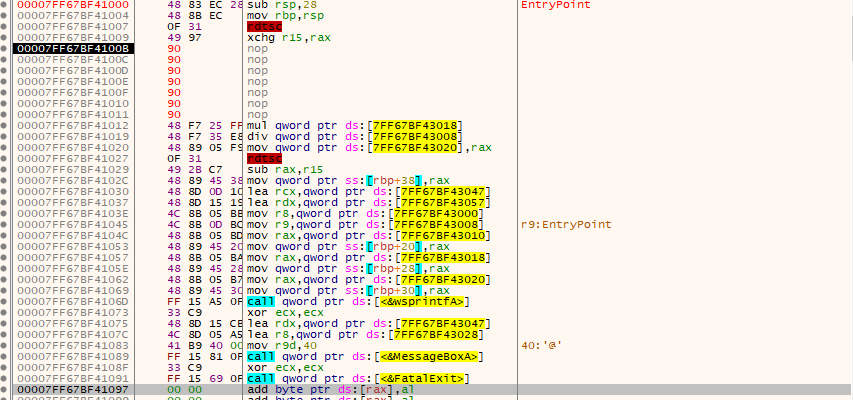


Рисунок 4 - Копіюємо адресу вільної комірки

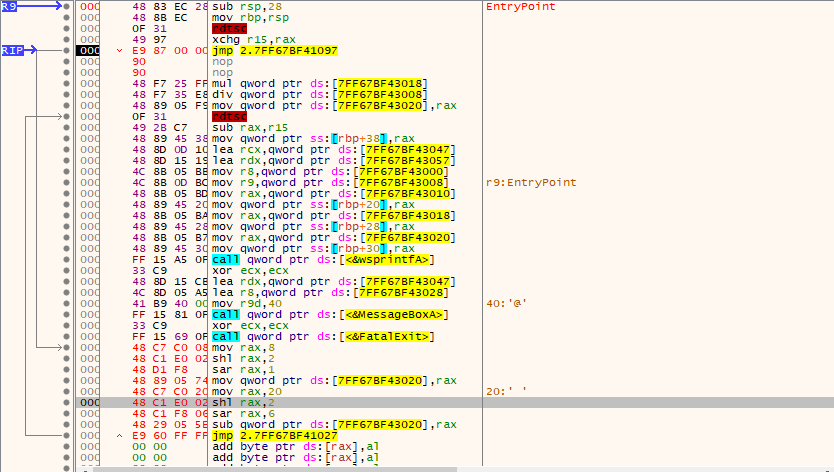


Рисунок 5 - Новий блок коду

Міняймо параметр функції *wsprintf*, який передається через регістр *RDX*.

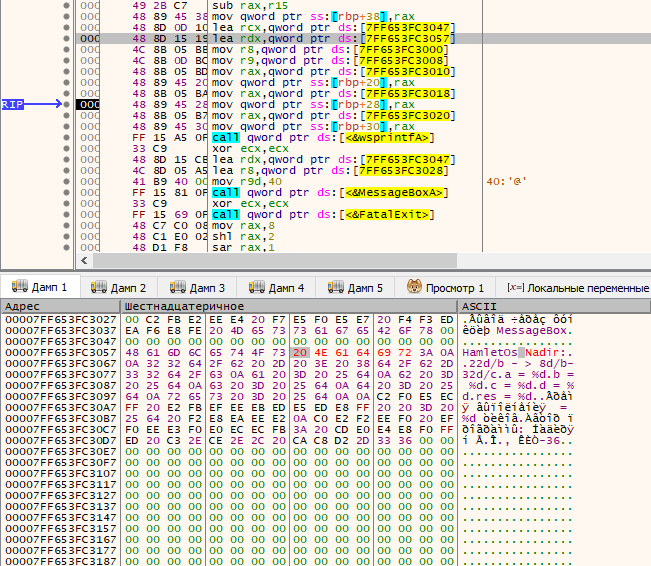


Рисунок 6 – Міняємо параметр функції *wsprintf*

Міняємо параметр функції *MessageBox*, який передається через регістр *R8.*

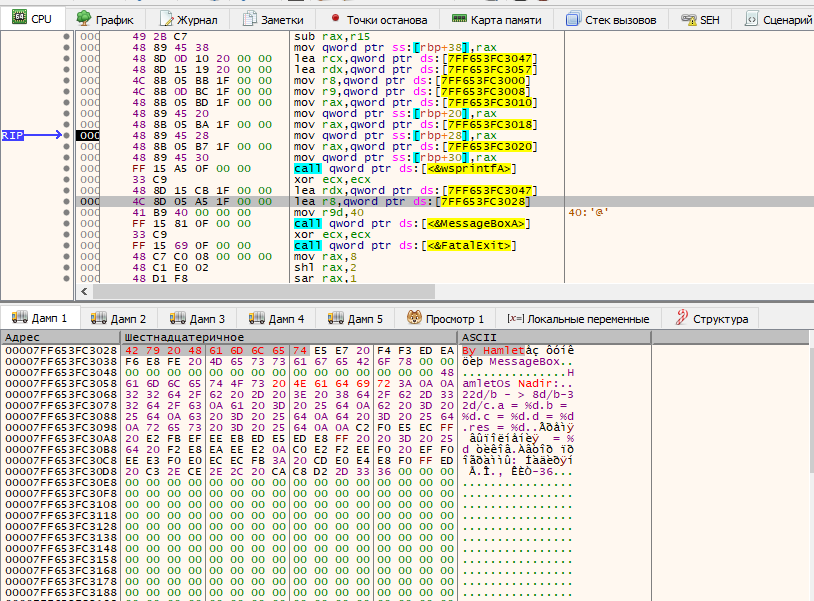


Рисунок 7 – Міняємо параметр функції *MessageBox*

Результат роботи програми з впровадженням наведено на рис. 8.

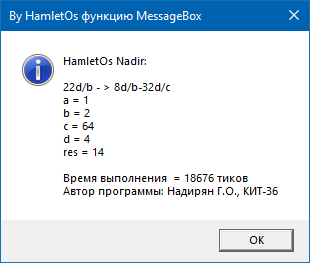


Рисунок 8 - Результат виконання

##### Алгоритм програми наведено на рис.9. (Клавіша «G» у вікні відладчика або можна перейти на вкладку з назвою "Графік");

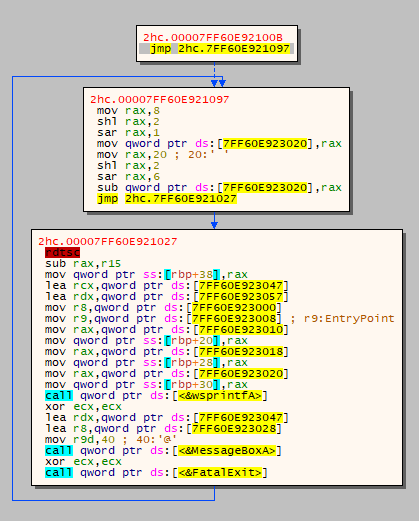
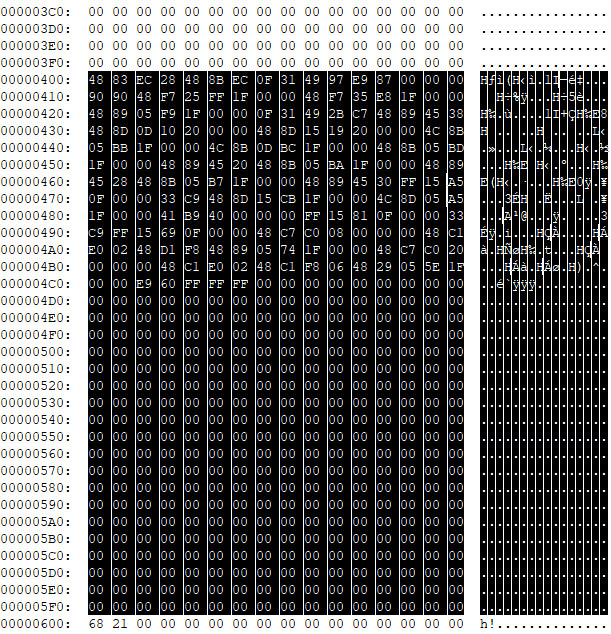


Рисунок 9 - Алгоритм програми

**Визначення вільного місця в секції**

Для підрахунку кількості байтів вільних осередків пам'яті необхідно з початкової адреси наступної секції відняти першу вільну комірку після закінчення коду 600h– 4C7h =139h



##### Рисунок 10 - **Визначення вільного місця в секції**

**Висновок**

Написали програму в середовищі masm64 відповідно до варіанта завдання відповідно до першої частиною варіанта завдання. Вивели час виконання рівняння. Результати вивели через функцію MessageBox. Підрахували вільну кількість байтів для здійснення впровадження. Виправили раніше написану програму в відладчик x64Dbg відповідно з другою частиною варіанта завдання. А також змінили в первісному ехе-файлі виведену інформацію (в функції *wsprintf* та функції MessageBox). Привели в цій функції своє прізвище. При виконанні операцій множення і ділення використовували логічні команди. Вивели час виконання рівняння.