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 6

Виконав ст.гр. КІТ–36

Надірян Г.О.

Перевірив

Челак В.В.

Харків 2019

**Лабораторна робота 6**

**Тема**: «Шифрування коду.»

##### Мета: Набути практичних навичок у захисті виконуваних файлів від сторонніх програм; створення нової функціональності, використовуючи неявне в експлуатації програмне забезпечення для платформи х64 в середовищі masm64

##### Постановка задачі

##### Виконати завдання в кілька етапів:

##### 1. Оголосити свою секцію, в якій буде перебувати дешифровщик і код в розшифрованому стані. Назва секції (НЕ сегмента) має містити перші літери прізвища, імені та по батькові. Наприклад, «CVV» або «ChelakVV».

##### 2. Написати програму в середовищі masm64 відповідно до завдання. Спочатку необхідно написати модуль обчислень, перевірити його коректність. В кінці програми вивести отримані результати через функцію MessageBox, а також номер варіанта, назва лабораторної роботи, шифр групи, прізвище виконавця. Перед розробленим модулем необхідно написати блок дешифрування коду. (Необхідно розрахувати адреса, кількість ділянок на шифрування (Byte, word, dword, qword - згідно варіанту)).

##### 3. Реалізувати програму шифрування тексту на будь-якій мові. Програма (консольна) повинна приймати 3 параметра - шлях до файлу, який потрібно зашифрувати, адреса першого байта шифрування (можна використовувати будь-яку систему числення для адресації) і кількість байт, які шифрують. Програма виконує шифрування заданого файлу в заданому діапазоні (від S байта до S + N байта). Механізм зміни файлу - на розсуд студента.

##### 4. Виконати розрахунок потрібних адрес для програми-шифрування.

##### 5. Зашифровать розроблену в masm64 програму за допомогою програми для шифрування.

##### 6. Запустити програму під отладчиком і звичайним запуском - програма повинна працювати коректно в обох випадках

##### Індивідуальна варіація взята з ЛР4

##### Відповідно до номером студента в групі вибрати варіант завдання і написати в середовищі masm64

##### на асемблері програму. Числа а є {a1, a2, a3, a4} задані масивом.

##### 12. a/cd – √e + b;

##### Індивідуальна варіація щодо шифрування

##### Нижче представлені найпростіші алгоритми шифрування і алгоритми дешифрування.

##### 12. sub key - add key (word)

##### *key – 15h*

**КОД ПРОГРАМ**

Програма на на асемблері

include win64a.inc ; Подключаемые библиотеки

\_NHAMLET segment READ WRITE EXECUTE alias**(**"NHO"**)**

mas1 **real8** 1.**,**2.**,**3.**,**4. ; массив чисел B

len1 equ **($-**mas1**)/**type mas1

mas2 **real8** 100.**,**2.**,**2.**,**4. ; a, c, d,e

tit1 db "Результат вычисления уравнения a/cd – sqart(e) + b ."**,**0

res dq len1 DUP**(**0**),**0 ;

buf1 dd len1 DUP**(**0**),**0 ; буфер вывода сообщения

ifmt db "Masm64 Массив bi = 1., 2., 3., 4."**,**10**,**

9**,**"Числа: a, c, d, e := 100., 2., 2.,4."**,**10**,**

"Результаты : %d ,%d ,%d ,%d "**,**10**,**10**,**

"Автор: HamletOs,фак. КИТ-36, НТУ ХПИ"**,**10**,**

9**,**0

WinMain proc

**sub** **rsp,**28h ; Выравнивание стека

**mov** **rbp,rsp**

**xor** **rdx,rdx**

;mov rcx,0EAh

**mov** **rcx,**06Ah

**mov** **rax,$+**1Fh ; Запись в rax адрес первого байта для шифра

**mov** **bx,**01515h ; Запись ключа в rbx

lp**:**

**mov** **dx,word** ptr **[rax]**

**add** **dx,bx**

**mov** **word** ptr **[rax],dx** ; Производим операцию xor с ключем

**inc** **rax** ; Переходим на следующий байт

**inc** **rax**

**loop** lp

; Начиная со следующей команды – код будет в

; зашифрованном/расшифрованном виде

**xor** **rcx,rcx**

**xor** **rax,rax**

**xor** **rbx,rbx**

**mov** **rcx,**len1

**lea** **rdx,**res

**lea** **rbx,**mas1

vmovsd **xmm1,**mas2**[**0**]** ; xmm1 — a — переслать real8

vmovsd **xmm2,**mas2**[**8**]** ; xmm2 — c

vmovsd **xmm3,**mas2**[**16**]** ; xmm3 — d

vmovsd **xmm5,**mas2**[**24**]**; ; xmm4 - e

vmulsd **xmm3,xmm2,xmm3**; ;xmm3 - c\*d

@@**:**

vmovsd **xmm0,qword** ptr**[rbx]**; - xmm0 - b

vmovsd **xmm4,xmm4,xmm1** ;xmm4 - a

vdivsd **xmm4,xmm4,xmm3** ;xmm4 - a/cd

;vmovsd xmm6,xmm6,xmm5 ;xmm6 - e

vsqrtsd **xmm6,xmm6,xmm5** ;xmm6 - sqrt(e)

;vaddsd xmm6,xmm6,xmm0; ;xmm6 - (sqrt(e)+b)

vsubsd **xmm4,xmm4,xmm6**; a/cd-sqrt(e)

vaddsd **xmm4,xmm4,xmm0**;a/cd-sqrt(e)+b

vcvttsd2si **eax,xmm4** ; от

vcvttsd2si **eax,xmm4** ; для отладки

**movsxd** **r15,eax** ;

**mov** **[rdx],eax** ; сохранение результата

**add** **rbx,**8

**add** **rdx,**8

**dec** **rcx**

**jnz** **@b** ; ссылка на предыдущую метку @@ (наверх)

invoke wsprintf**,addr** buf1**,addr** ifmt**,**res**,**res**[**8**],**res**[**16**],**res**[**24**]**

invoke MessageBox**,**0**,addr** buf1**,addr** tit1**,**MB\_ICONINFORMATION

invoke ExitProcess**,**0

WinMain endp

\_NHAMLET ends

end

**Програма на c#**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace ReverseProgramming

{

class Program

{

static void Main(string[] args)

{

if (args.Length != 3)

{

Console.WriteLine("Usage: RP.exe DecryptedExe.exe startbytenumber size");

return;

}

byte[] exe = File.ReadAllBytes(args[0]);

ushort key = 0x1515;

int start = int.Parse(args[1]);

int stop = int.Parse(args[2]);

for (int i = start, k = 0; k < stop; i += 2, k++) // к увеличиваем на 2 т.к. изменяем слова.

{

ushort res = (ushort)((exe[i + 1] << 8) | exe[i]);

res -= key;

exe[i + 1] = (byte)(res >> 8);

exe[i] = (byte)res;

}

File.WriteAllBytes("Encrypt.exe", exe);

}

}

##### }

6f – 50 =1Fh адреса першого байту

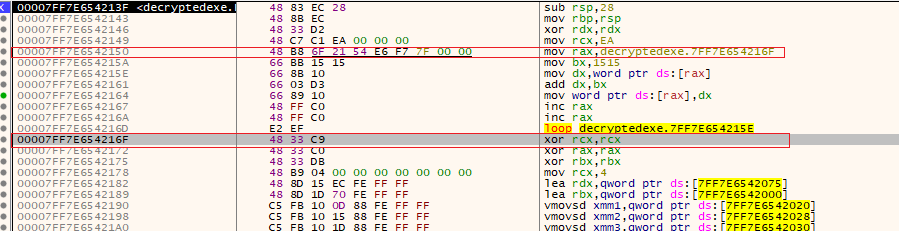
*mov rax,$+1Fh ;* Запись в rax адрес первого байта для шифра

Рисунок 1 – адреса першого байту

2242-216F = D3/2=6A (округлили)

*mov rcx,06Ah*

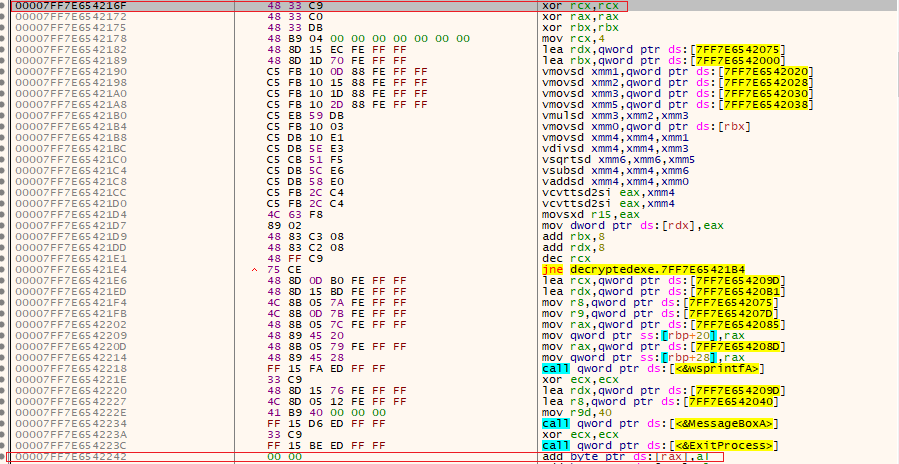
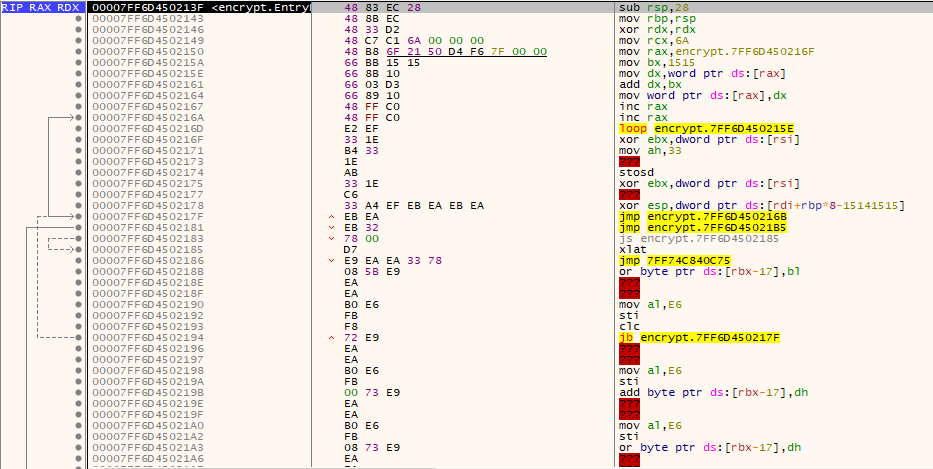


Рисунок 2 - рахуємо скільки байт будемо розглядати



Програма 3 – Зашифрована програма

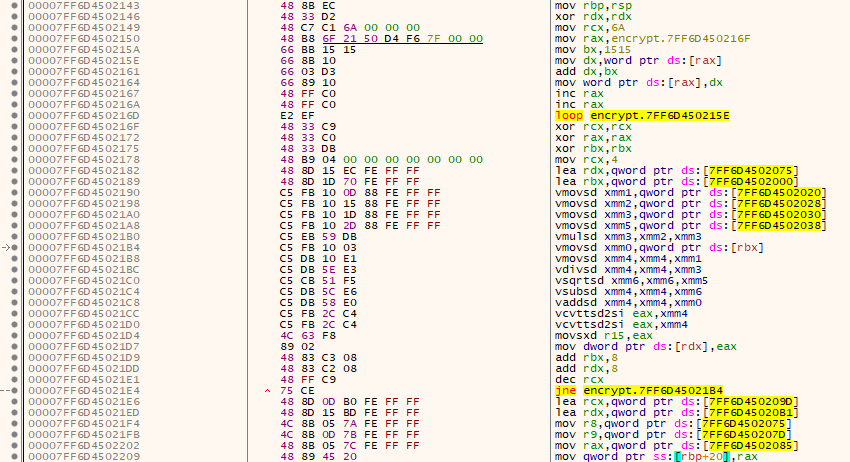


Рисунок 4 - Програма після розшифрування

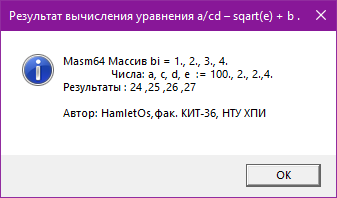


Рисунок 5 - Робота програми

**Висновок**

Набули практичні навички у захисті виконуваних файлів від сторонніх програм; створили новий функціональності, використовуючи неявне в експлуатації програмне забезпечення для платформи х64 в середовищі masm64.