MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY

"KHARKIV POLYTECHNICAL INSTITUTE"

Department of Computer Engineering and Programming

«Software Means of Information Protection »

*Laboratory work report No 7*

*Topic: «* **Self-Encrypting Code** *»*

Student:

Group. KH 919 I.E

Ayoub EL-HADDADI

Verified by:

Lecturer Viktor CHELAK

Kharkiv – 2022

***Purpose of work***:

To acquire practical skills in protecting executable files from third-party programs, creation of new functionality using implicit in use software for the x64 platform in the masm64 environment.

***Individual task:***

Variant 8:

Encryption and decryption algorithms: add key – sub key (word)

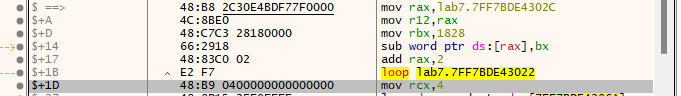
**Algorithm of the program**

First, we declare a section in which the decoder and the code will be in the decrypted state, this section must contain the first letters of the surname and name.

In accordance with the task we implement the calculation module (same as in Laboratory work N4). At the end of the program, display the results through the Message Box function, and before the developed module, it is necessary to write a code decryption block. And need to calculate:

* Address to begin decryption:

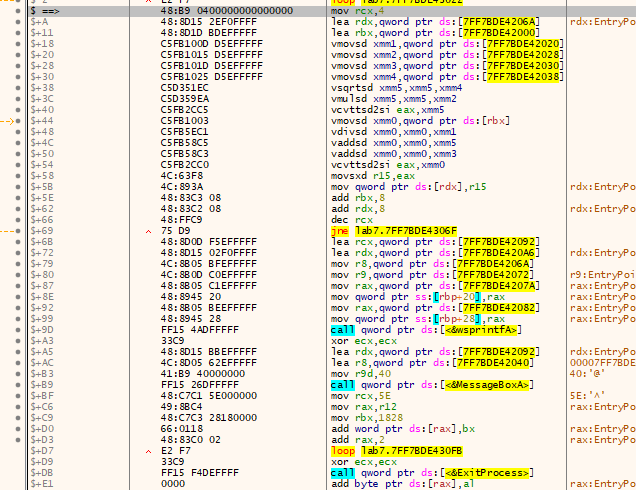
This address is current address plus amount of bytes until first line of code in encrypted state (After loop).



MOV rax, $+1Eh

* Amount of bytes to decrypt:

Amount of bytes from first line of code in encrypted state until last line.



MOV rcx, 0E1h

* Key of decryption: key of decryption can be chosen randomly

MOV rbx, 1828h

Accordance to variant 8, decryption algorithm is to substitute the key from every word (2 bytes), Therefore decryption loop should be like the following:

lp:

SUB word ptr [rax], bx

ADD rax, 2

Loop lp

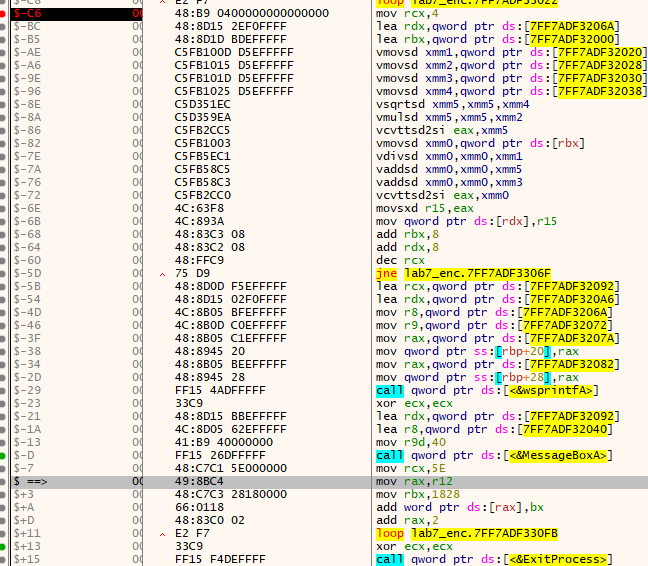
*#Encryption of calculation module:*

Program for encrypting exe file is developed with C# (.NET), and it takes three parameters. First, exe file to encrypt, second bytes number to start from, third is amount of bytes to encrypt. Accordance to variant encryption is to add the key to every word (2 bytes), which means loop must have “i += 2” step.

After encryption of calculation module program should work fine, since the first part of code will decrypt second module and start calculations.

Last part of the program before exit process contain another encryption loop to encrypt calculation module after showing results, we repeat same steps of first decryption part but we change:

* address of beginning encryption



We stored address of beginning of encryption in r12 register so we can use it in this part.

* Amount of bytes which is 5E
* Key we keep the same key.
* Operation inside loop to “ADD”

**Source Code**

Full source code of this lab you can find it in:

**<https://github.com/Elh-Ayoub/RP_Labs/tree/main/lab7>**

**Results of the program:**

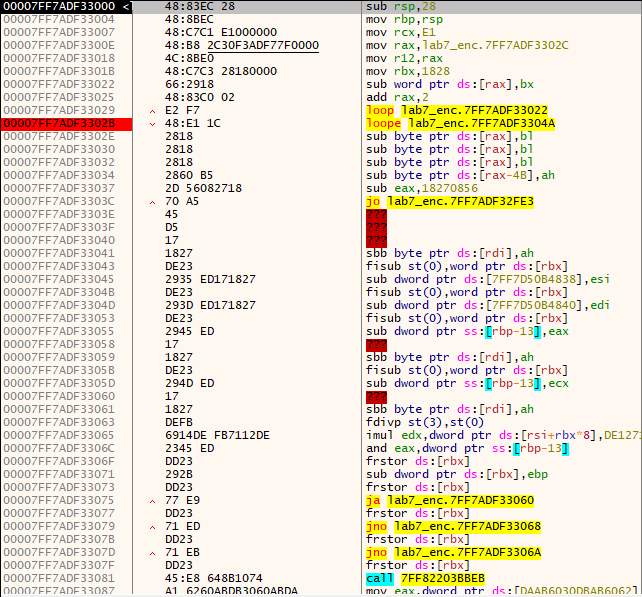


Figure 1 – Program before decryption

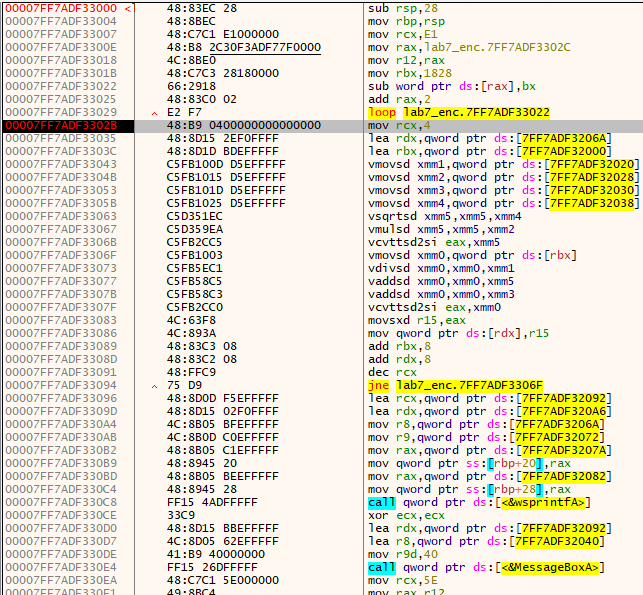


Figure 2 – Program after decryption

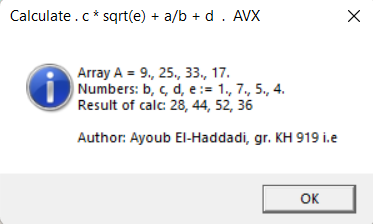


Figure 3 – Program results

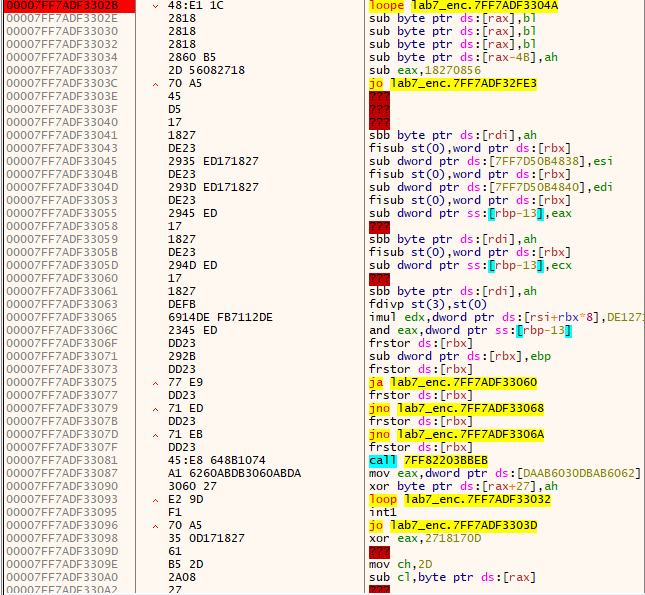


Figure 4 – Program After showing result and self encrypting again

**Conclusions:**

As a result of laboratory work we gained a practical skills in writing and a self-modifying code in masm64 environment, and protecting executable files from third-party programs, creation of new functionality using implicit in use software for the x64 platform in the masm64 environment.

**You can also find this report in:**

<https://github.com/Elh-Ayoub/RP_Labs/tree/main/Docs>