Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра ОТ

**ЗВІТ**

про виконання лабораторної роботи № 6

з дисципліни

“Системне програмування”

|  |  |  |
| --- | --- | --- |
| Прийняв: |  | Виконав: |
| Павлов Валерій  Георгійович |  | студент 2-го курсу  гр. ІП-53 ФІОТ  Сулима Олександр Сергійович |

Київ – 2016

**Лабораторна робота №6**

**Дослідження структури програм формату COM.**

**Дослідження структури програм формату EXE.**

**Мета роботи**

Ознайомлення із специфікацією **COFF** (Common Object File Format). Вивчення прийомів дослідження структури файлів **PE**-формату.

**Порядок виконання роботи**

1. Вивчити структуру програм формату **EXE** [1].

2. Розробити програму на мові Асемблер, за допомогою якої у віконному інтерфейсі по введеному паролю виводяться персональні дані студента – ПІБ, дата народження, номер залікової книжки тощо (див. лаб. роботу 1).

3. Виконати компіляцію розробленого файлу у формат **EXE**.

4. Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.

5. Отриманий виконавчий файл дослідити за допомогою програми **HEX**-редактора **HIEW32** (див. лаб. роботу 5) або **WinHex** (http://rainbowsky.ru/system/winhex/ - trial версия\*) [2]

6. На скріншоте перших 25 рядків вмісту файлу обвести кольоровим олівцем або фломастером області MS-DOS заголовка (**DOS\_HEADER**), PE заголовка (**PE\_HEADER**) і таблиці секцій (**SECTION\_HEADERS**). Скріншот привести в звіті по лабораторній роботі.

7. Відповідно до опису секцій [1] скласти таблицю, в яку занести параметри свого файлу, вказані в розділах 3.3.1, 3.4.1 і 4 (перша таблиця).

8. У останньому стовпчику таблиці розшифрувати виписані значення полів заголовка файлу. Таблицю привести в звіті по лабораторній роботі.

9. Провести дослідження того ж файлу за допомогою меню "**PE Editor**" безкоштовної програми **PE Tools** (http://soft.mydiv.net/win/download-PE-Tools.html\*). Все скріншоти вікон програми з даними, відповідними раніше побудованій таблиці, привести в звіті по лабораторній роботі.

10. Дослідити таблицю імпорту (**Import Directory**) даного файлу і визначити, які саме функції використовуються з бібліотек, що підключаються. Скріншоти вікон **Import Directory** з функціями, що імпортуються, з кожного бібліотечного файлу привести в звіті по лабораторній роботі.

11. Знайти в тексті файлу по зсуву, узятому з побудованої таблиці, секцію з даними і переконатися, що текст оригінала пароля, що міститься в тексті програми, може бути легко виявлений за допомогою **HEX**-редактора. Привести скріншот цього фрагмента програми у вигляді **HEX** - коду в звіті по лабораторній роботі.

12. Виконати шифрування пароля за допомогою функції **XOR**, знову скомпілювати **EXE** -файл і переконатися, що тепер вони не виявляються явним чином в тексті виконуваного **EXE** -файлу. Привести скріншоти цієї програми в режимах «**Hex**» і «**Text**» в звіті по лабораторній роботі.

**Лістинг:**

.486

.model flat, stdcall

option casemap :none

include \masm32\include\windows.inc

include \masm32\include\masm32.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

include \masm32\include\dialogs.inc

include \masm32\macros\macros.asm

includelib \masm32\lib\masm32.lib

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

DlgProc PROTO :DWORD,:DWORD,:DWORD,:DWORD

EditProc PROTO :DWORD,:DWORD,:DWORD,:DWORD

.data

Access db 0

pass db "masm32",0

passLen=$-pass

InputLen db 0

rig db "Right password",0

no db "Password is false!",0

par db "Parol",0

buf db 0

res db "Result: ", 0

inform db "Sulyma Aleksandr", 10, "5318", 10, "IP-5318", 0

format db "%d", 0

hInstance dd 0

buffer db 50 dup (0)

oldEditProc dd 0

.code

start:

mov hInstance, rv(GetModuleHandle,NULL)

call Main

invoke ExitProcess, eax

Main proc

Dialog "Sulyma-6","Arial",10,WS\_OVERLAPPEDWINDOW, 1,5,5,280,100,1024

DlgEdit WS\_BORDER or ES\_AUTOVSCROLL or ES\_MULTILINE or WS\_TABSTOP,30,20,200,20,1000

;DlgButton "OK",WS\_TABSTOP,150,50,60,20,IDOK

CallModalDialog hInstance,0,DlgProc,NULL

ret

Main endp

DlgProc proc hWin:DWORD,uMsg:DWORD,wParam:DWORD,lParam:DWORD

.IF uMsg == WM\_INITDIALOG

invoke GetDlgItem,hWin,1000

invoke SetWindowLong,eax,GWL\_WNDPROC,EditProc

mov oldEditProc, eax

.ELSEIF uMsg == WM\_COMMAND

.ELSEIF uMsg == WM\_CLOSE

invoke EndDialog, hWin, 0

.ENDIF

return 0

DlgProc endp

EditProc proc hCtl:DWORD,uMsg:DWORD,wParam:DWORD,lParam:DWORD

.IF uMsg == WM\_KEYDOWN

.IF wParam == VK\_RETURN

invoke GetWindowText,hCtl,ADDR buffer,512

call comp

cmp Access, 1

invoke MessageBoxA, 0 , addr no, addr par, 0

.ENDIF

.ENDIF

invoke CallWindowProc,oldEditProc,hCtl,uMsg,wParam,lParam

ret

EditProc endp

good proc

invoke MessageBoxA, 0, offset inform, offset res, 0

invoke ExitProcess, 0

good endp

comp proc

mov esi, offset pass

mov edi, offset buffer

mov ecx, 6 ; длина строки

cld ; сравнение пока не конец или пока элементы равны

repe cmpsb

jnz equal\_words

mov Access, 0

call good

equal\_words:

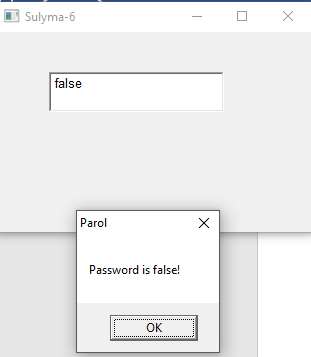
ret

comp endp

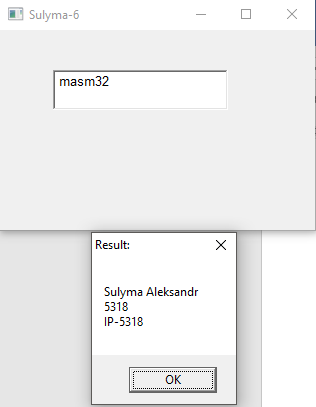
end start

**Приклад виконання:**

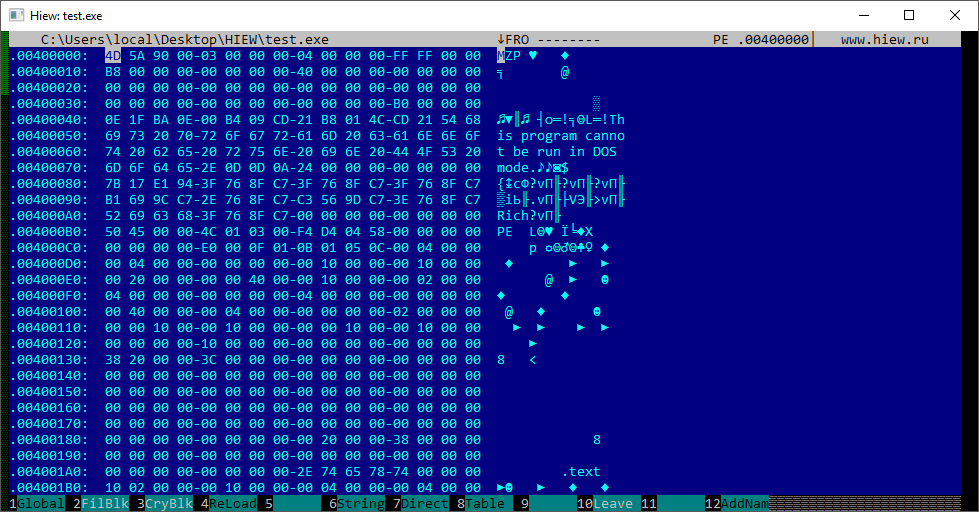
**Неправильний пароль:**

****

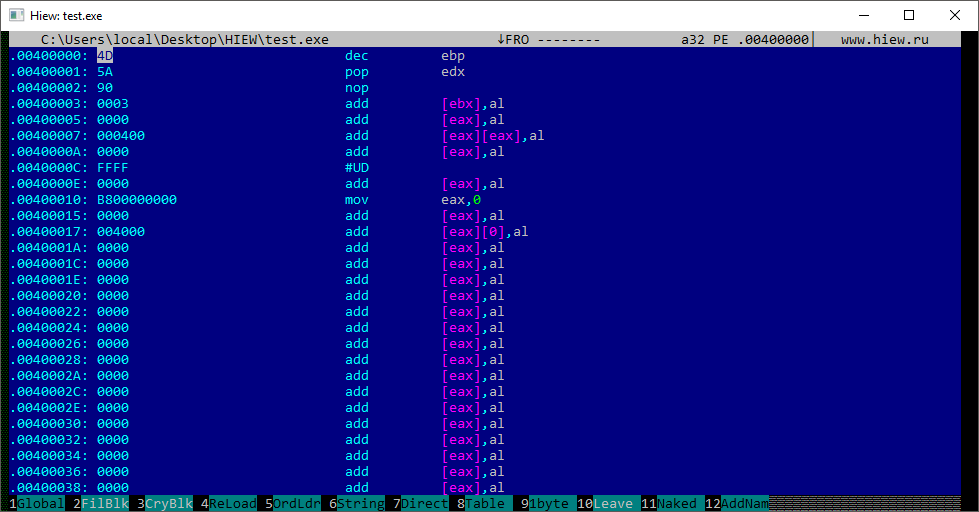
**Правильний пароль:**

****

Hex:



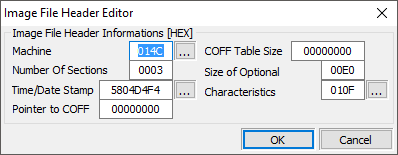
Decode:

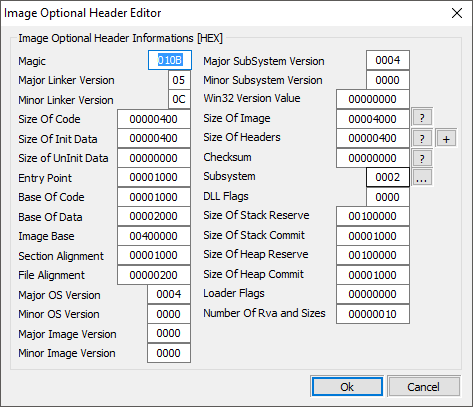
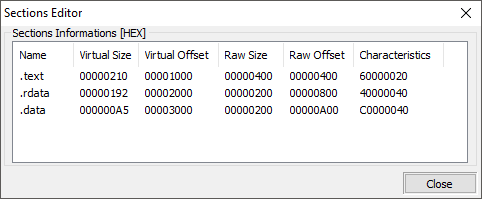


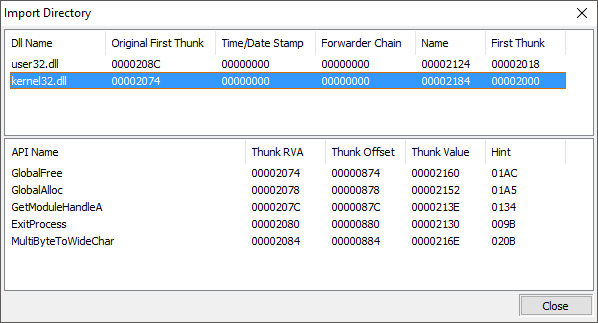
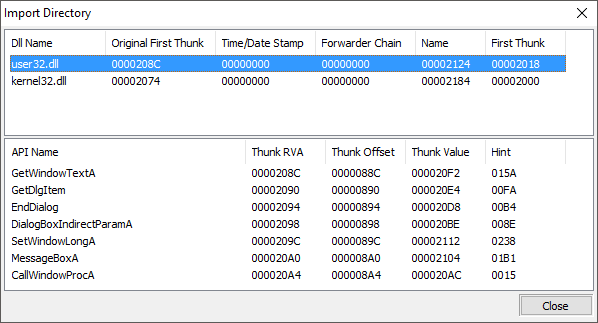
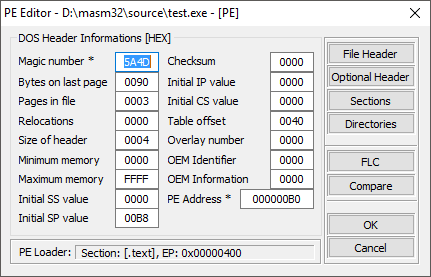
**Таблиця програмних полів**

|  |  |
| --- | --- |
| **Field** | **Value** |
| Machine Type | 014C |
| Magic | 010B |
| MajorLinkerVersion | 05 |
| MinorLinkerVersion | 0C |
| SizeOfCode | 00000200 |
| SizeOfInitializedData | 0000 0400 |
| SizeOfUninitializedData | 0000 0000 |
| AddressOfEntryPoint | 0000 1000 |
| BaseOfCode | 0000 1000 |
| BaseOfData | 0000 2000 |
| Name | .text |
| VirtualSize | 000000D6 |
| VirtualAddress | 00001000 |
| SizeOfRawData | 00000200 |
| PointerToRawData | 00000400 |
| PointerToRelocations | 00000000 |
| PointerToLinenumbers | 00000000 |
| NumberOfRelocations | 0000 |
| NumberOfLinenumbers | 0000 |
| Characteristics | 60000020 |
| Name | .rdata |
| VirtualSize | 000000DA |
| VirtualAddress | 00002000 |
| SizeOfRawData | 00000200 |
| PointerToRawData | 00000600 |
| PointerToRelocations | 00000000 |
| PointerToLinenumbers | 00000000 |
| NumberOfRelocations | 0000 |
| NumberOfLinenumbers | 0000 |
| Characteristics | 40000040 |
| Name | .data |
| VirtualSize | 000000F0 |
| VirtualAddress | 00003000 |
| SizeOfRawDatac | 00000200 |
| PointerToRawData | 00000800 |
| PointerToRelocations | 00000000 |
| PointerToLinenumbers | 00000000 |
| NumberOfRelocations | 0000 |
| NumberOfLinenumbers | 0000 |
| Characteristics | C0000040 |

**Серія скріншотів з програми PE Tools v1.5**







**Висновок:**

Виконавши дану лабораторну роботу, я дослідив роботу типових команд роботи з консоллю та вікнами. В данній роботі була реалізована перевірка паролю та його шифрування, для того, щоб не можна було побачити цей пароль у програмах типу Hiew32 або Win-HEX. Також була вивчена компіляція програм зі специфікацією PE/COFF, що дозволяє продивлятися внутрішні поля програми, загальні дані про програму: використані бібліотеки, використані шаблонні класи цих бібліотек і таке інше.