**Міністерство освіти і науки України**  
 **Національний технічний університет України**  
 **«Київський політехнічний інститут імені Ігоря Сікорського»**  
 **Факультет інформатики та обчислювальної техніки**  
 **Кафедра обчислювальної техніки**

**Лабораторна робота №3**

з дисципліни  
 «Системне програмування»

Виконав: Перевірив:

Студент 2-го курсу групи ІМ-13 Павлов Валерій Георгійович

Нестеров Дмитро Васильович  
номер у списку групи: 16

Київ 2023

**Мета роботи**

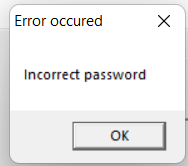
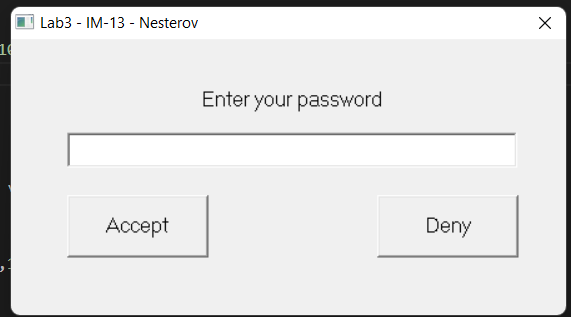
Ознайомлення із специфікацією COFF (Common Object File Format). Вивчення прийомів дослідження структури файлів PE-формату.

**Виконання роботи**

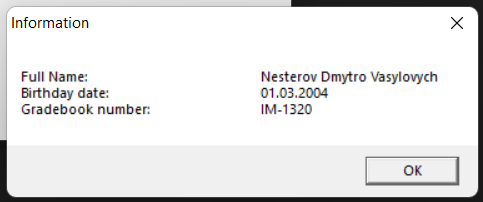
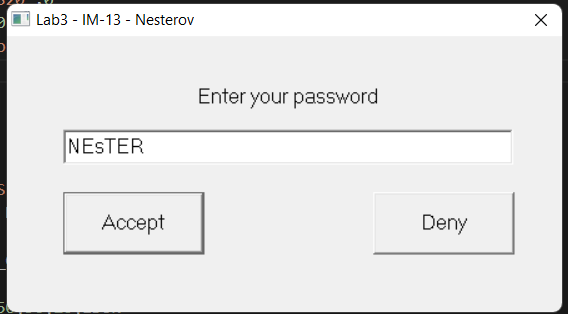
**Перевірка роботи обох програм шляхом введення як правильного, так і**  **невірних паролів.**

**Без шифрування:**

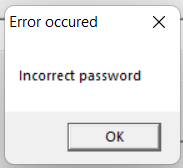
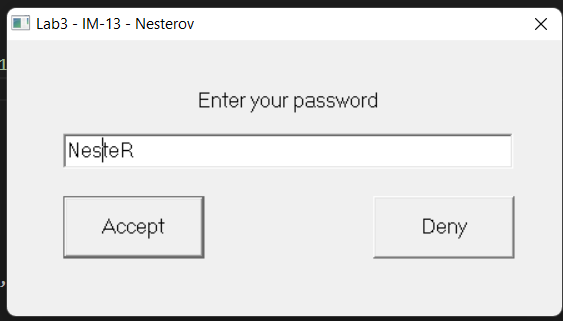
1. **Порожнє поле вводу**



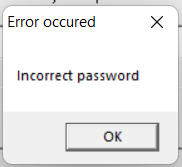
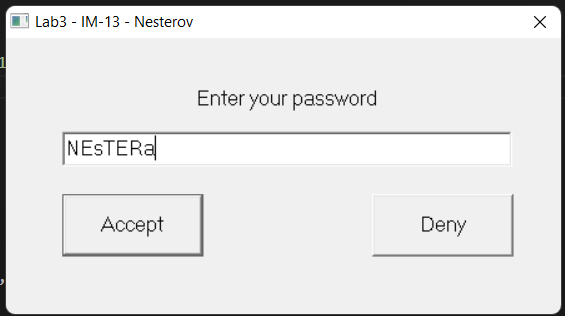
1. **Пароль правильний**



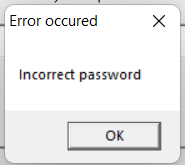
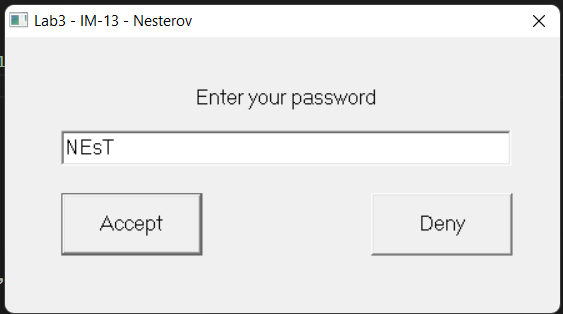
1. **Є літери у неправильному регістрі**



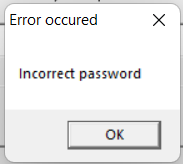
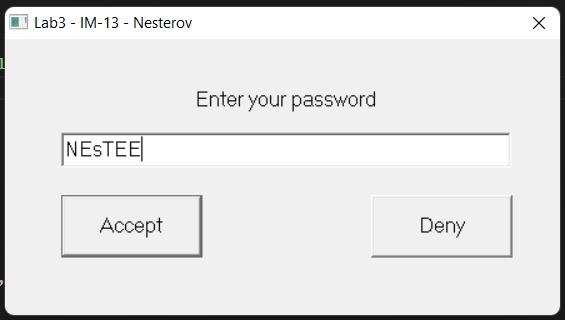
1. **Є зайві літери**



1. **Недостатньо літер**

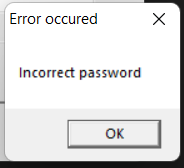
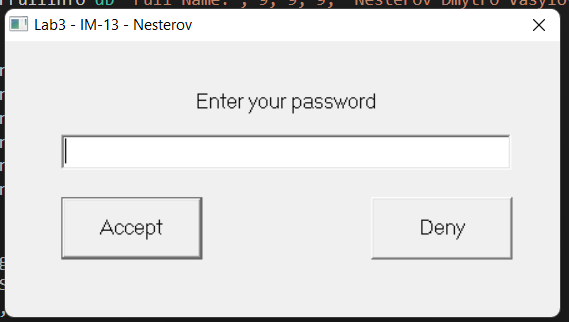


1. **Неправильний пароль**

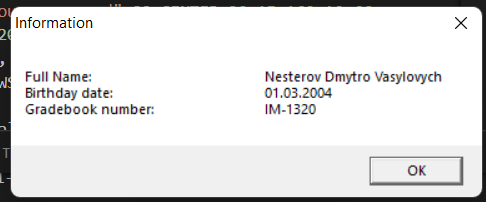
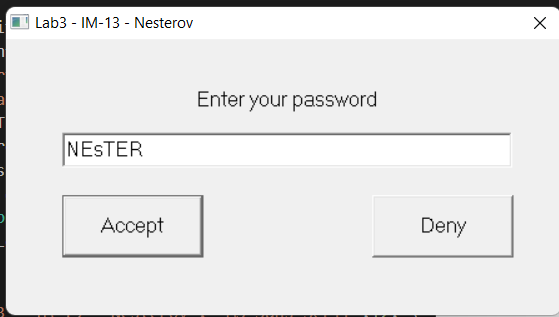


**З шифруванням**

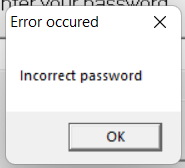
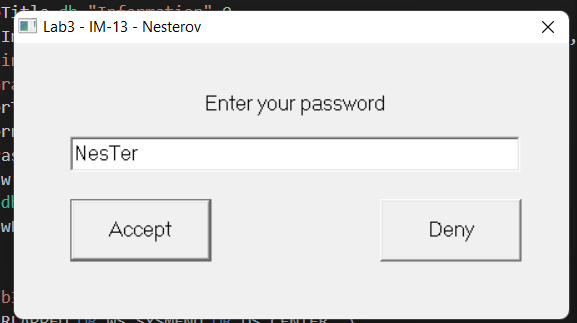
**1) Порожнє поле вводу**



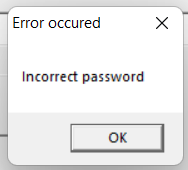
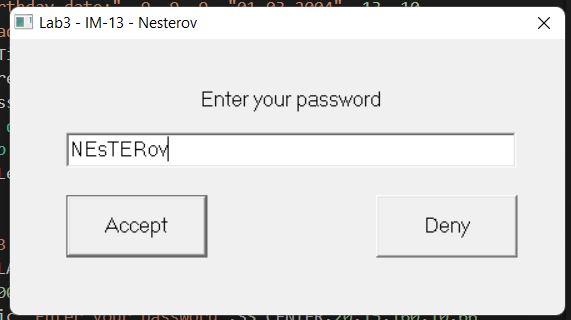
**2) Пароль правильний**



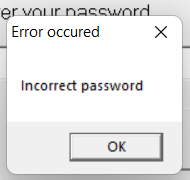
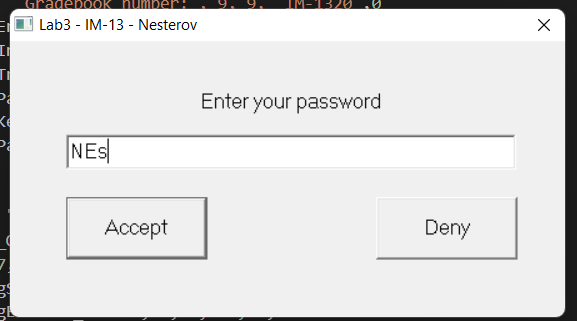
**3) Є літери у неправильному регістрі**



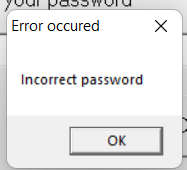
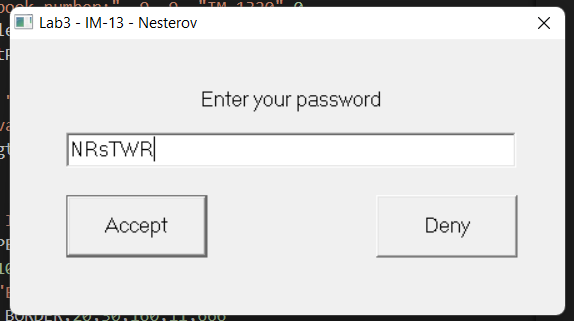
**4) Є зайві літери**

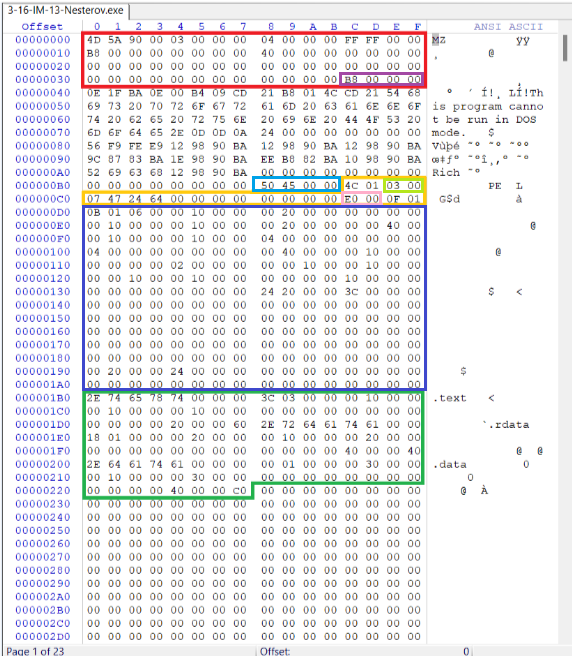


**5) Недостатньо літер**



**6) Неправильний пароль**



**5-6. Дослідження файлу в WinHex.** 

**Пояснення до виділених фрагментів:**

DOS Header починається з початку файлу і має сталу довжину 64 байти (**на скріншоті виділено червоним**), в останніх 4 байтах DOS Header прописаний вказівник на PE Header (**на скріншоті виділено фіолетовим**). Оскільки байти написані у зворотньому порядку, початок PE Header знаходиться за зсувом 00 00 00 B8. Це 184 байти.

Розмір PE Signature завжи 4 байти (**на скріншоті виділено голубим**), значенням в ASCII-форматі є ‘PE\0\0’. Після PE Signature іде заголовок PE-файлу (**на скріншоті виділено жовтим**), який теж має сталу довжину, а саме 20 байтів. Потім починається Optional Header (**на скріншоті виділено синім**). Розмір Optional Header змінний і його можна дізнатися з двобайтового поля заголовку PE-файлу, яке починається після перших 16 байтів заголовку (**На скріншоті виділено рожевим)**. Маємо «**Е0 00**», отже розмір необов’язкового хедера дорівнює 00Е0h, а це 224 байти.

Section Headers починається одразу після Optional Header (**на скріншоті виділено зеленим**). Кількість секцій можна дізнатися з двобайтового поля заголовку PE-файлу, що починається після перших 2 байтів, (**на скріншоті виділено салатовим**), маємо «**03 00**», тобто 3 секції, кожна з яких має розмір 40 байтів, що є сталим.

**7-8. Таблиця з параметрами файлу**

Розділ 3.3.1: Machine Types

|  |  |  |
| --- | --- | --- |
| Поле | Значення | Опис |
| Machine type | 014С | Тип машини: *Intel 386* або пізніше або інші схожі процесори |

Розділ 3.4.1: Optional Header Standard Fields (Image Only)

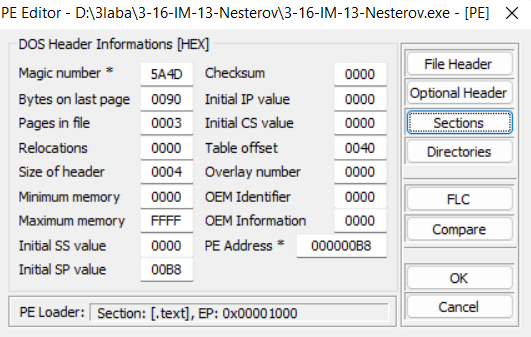
|  |  |  |
| --- | --- | --- |
| Поле | Значення | Опис |
| Magic | 01 0В | Беззнакове ціле число (unsigned int), що ідентифікує стан файлу. *0x10B* – ідентифікатор звичайного виконуваного файлу (.exe). |
| MajorLinkerVersion | 06 | Старша цифра номера версії компонувальника |
| MinorLinkerVersion | 00 | Молодша цифра номера версії компонувальника. |
| SizeOfCode | 00 00 10 00 | Сума розмірів всіх секцій, що містять програмний код. |
| SizeOfInitializedData | 00 00 20 00 | Сума розмірів всіх секцій, що містять ініціалізовані дані. |
| SizeOfUninitializedData | 00 00 00 00 | Сума розмірів всіх секцій, що містять неініціалізовані дані. |
| AddressOfEntryPoint | 00 00 10 00 | Адреса точки запуску програми |
| BaseOfCode | 00 00 10 00 | Адреса початку розділу коду |
| BaseOfData | 00 00 20 00 | Адреса початку розділу змінних |

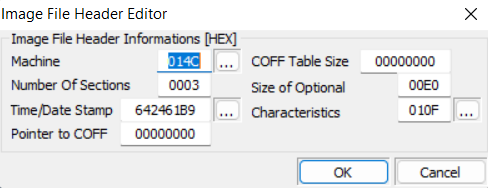
Розділ 4: Section Table (Section Headers)

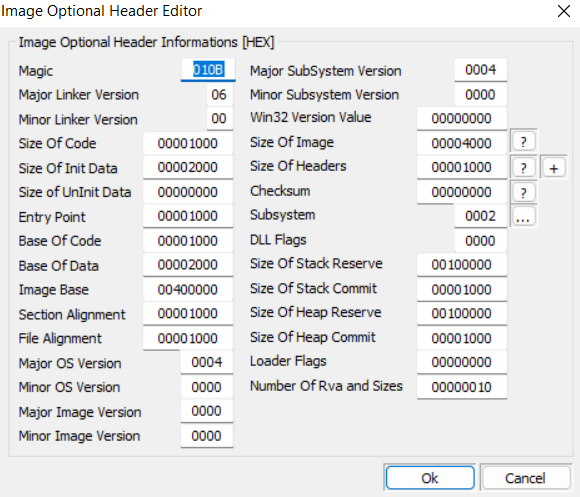
Як приклад, розібрав .rdata

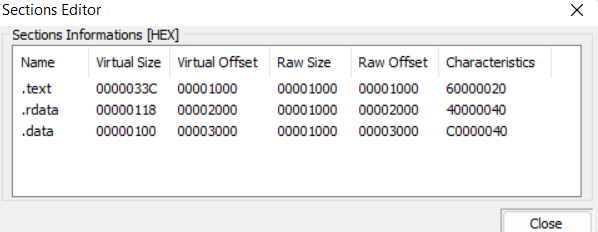
|  |  |  |
| --- | --- | --- |
| Поле | Значення | Опис |
| Name | **.rdata\0\0** | 8-байтовий рядок ASCII із доповненням нульовими значеннями, це назва секції |
| VirtualSize | **00 00 01 18** | Розмір секції при в пам'яті |
| VirtualAdress | **00 00 20 00** | Адреса першого байту розділу |
| SizeOfRawData | **00 00 10 00** | Розмір розділу у файлі |
| PointerToRawData | **00 00 20 00** | Вказівник на першу сторінку розділу в COFF файл |
| PointerToRelocations | **00 00 00 00** | Вказівник файлу на початок записів переміщення для розділу (0 для .exe) |
| PointerToLinenumbers | **00 00 00 00** | Вказівник файлу на початок записів номерів рядків для розділу |
| NumberOfRelocations | **00 00** | Кількість релокаційних входжень в розділ (0 для .ехе) |
| NumberOfLinenumbers | **00 00** | Кількість номерних записів розділу |
| Characteristics | **40 00 00 40** | Прапори, що описують характеристики розділу (атрибути секції). |

**9. Аналіз того ж самого файлу в PE Tools, дані співпали.**

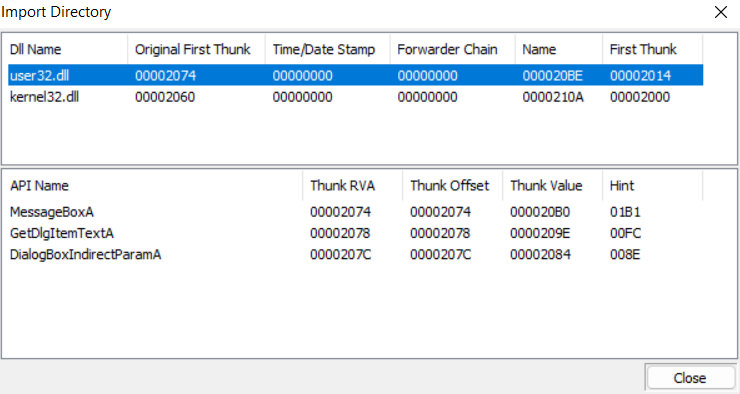


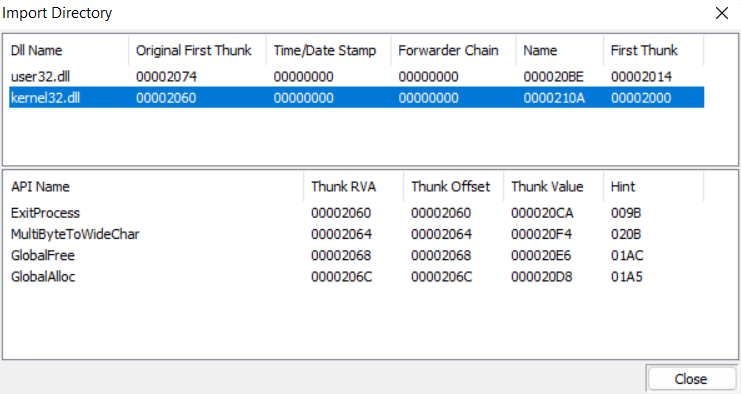






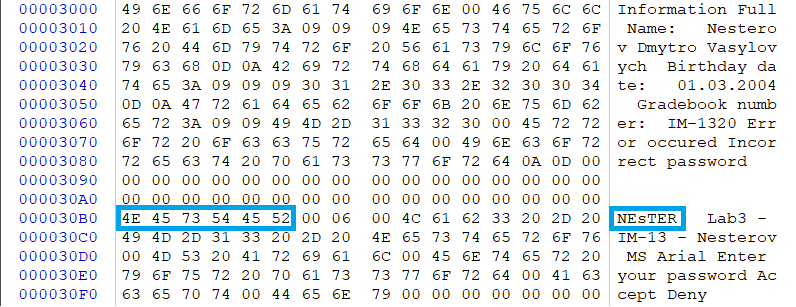
**10. Таблиця імпорту**





На приведених вище скріншотах видно які бібліотечні файли імпортуються (user32.dll, kernel32.dll), та які саме функції з них використовуються (3 і 4 функції відповідно).

**11.** За значенням зсуву VirtualAdress = 00 00 30 00 для .data знаходимо початок секції з даними, а потім знайдемо і сам пароль. Бачимо, що його ніяк не зашифровано і зберігається він явно.

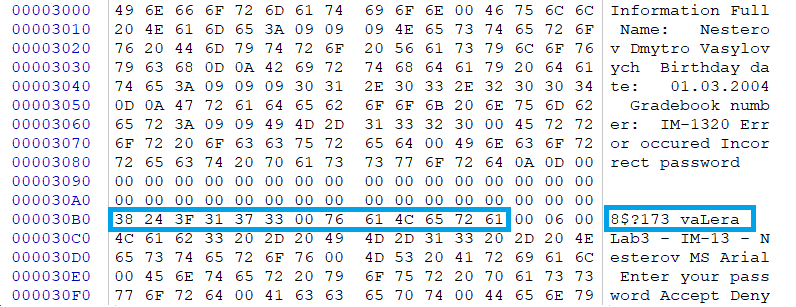


**12. Виконуємо шифрування паролю використовуючи метод XOR**

Зашифруємо вручну пароль “NEsTER” з ключем “vaLera”

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ASCII** | **Binary** | | | | | |
| NEsTER | 01001110 | 01000101 | 01110011 | 01010100 | 01000101 | 01010010 |
| vaLera | 01110110 | 01100001 | 01001100 | 01100101 | 01110010 | 01100001 |
| **XOR** | | | | | | |
| 8$?173 | 00111000 | 00100100 | 00111111 | 00110001 | 00110111 | 00110011 |

Відкриваємо новий exe-файл в WinHex і знаходимо місце зашифрованого паролю. Як бачимо, тепер явно він не зберігається.



***Висновки:***

Під час виконання даної лабораторної я дізнався як можна частково дослідити exe-файл, а саме:

* де знаходиться заголовок DOS та який його розмір
* як знайти розташування початку заголовка PE, які складові входять до нього та їх розміри, якщо вони сталі, або де ці розміри дізнатися, якщо ні, а ще розташування Section Headers та його розмір секції і де дізнатись їх кількість
* як знайти початок самих секцій у пам’яті, щоб наприклад виявити пароль використовуючи певні програми (у моєму випадку **WinHex** та **PE Editor**).

Також я навчився шифрувати дані методом XOR. Хоча цей метод є досить не надійним, бо і зашифрований пароль і ключ можна знайти в файлі і розшифрувати пароль, але на це треба буде витратити певний час і пароль хоча б не прописаний явно. В реальному світі шифрування – дуже корисна річ і там використовують набагато складніші методи, при яких маючи публічний ключ практично неможливо визначити приватний. Можна лише вгадувати.