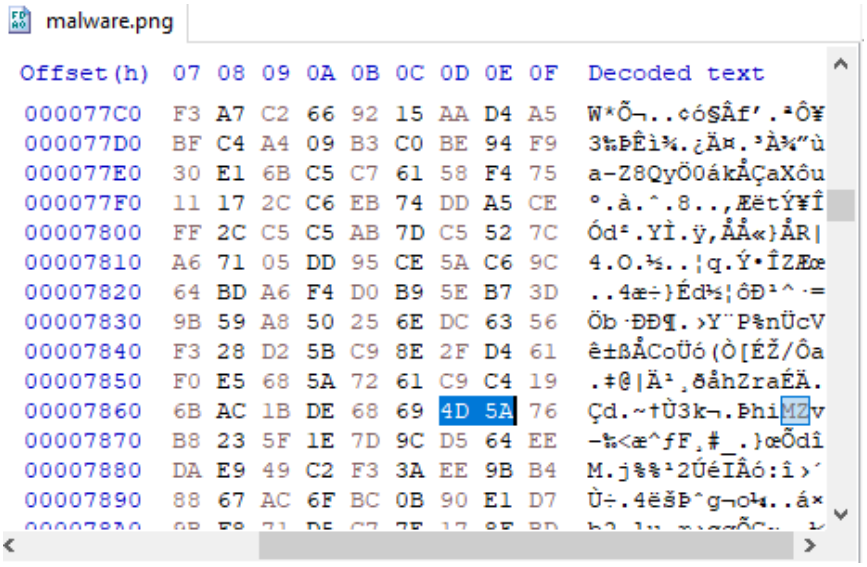
Laborator 7

Exercitiul 1

1. Imaginea se deschide normal si nu apare nicio eroare
2. Imaginea are headerul PNG dar are un executabil ascuns



3. Snippet raport image din VirusTotal:

```
"description": "This signature detects the presence of a number of Windows API function
ality often seen within embedded executables. When this signature alerts on an executab
le, it is not an indication of malicious behavior. However, if seen firing in other fil
e types, deeper investigation may be warranted."
"last_analysis_stats": {
        "harmless": 0,
        "type-unsupported": 14,
        "suspicious": 0,
        "confirmed-timeout": 0,
        "timeout": 1,
        "failure": 0,
        "malicious": 1,
        "undetected": 60
      }
```

4. Snippet raport continut suspicios din VirusTotal:

```
"type_extension": "exe",
"last_analysis_stats": {
        "harmless": 0,
```

```
        "type-unsupported": 12,
        "suspicious": 0,
        "confirmed-timeout": 0,
        "timeout": 0,
        "failure": 0,
        "malicious": 0,
        "undetected": 64
      }
```

5. Fisierele DLL ajuta la rularea executabilului
6. Imaginea este un malware deoarece contine un executabil ce sterge fisierele pdf

Exercitiul 2

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    char pass[7] = "fmiSSI";
    char input[7];
    int passLen = strlen(pass);
    cout << "Introduceti parola: ";
    cin >> input;
    if (strncmp(input, pass, passLen) == 0)
    {
        cout << "Parola introdusa este corecta!\n";
    }
    else
    {
        cout << "Ati introdus o parola gresita\n";
    }
    return 0;
}
```

Daca introducem o parola de 14 caractere orice input va fi consider corect deoarece bufferul va fi suprascris. Aceasta vulnerabilitate exista pentru versiunile de C++ mai vechi de C++ 17 si se numeste **buffer overflow**

Exercitiul 3

Calcularea valorii SHA256 a unui fisier in Python:

```
import hashlib

with open("fisier.txt", "rb") as f:
    bytes = f.read()
    readable_hash = hashlib.sha256(bytes).hexdigest()
    print(f"SHA-256 value: {readable_hash}")
```

```
# Output: SHA-256 value: ea483198f5f978d1c59727f042d62598469897afa658aeb43bef2bb2fe8709cc
```

VirusTotal Api Key:
b3aff7d0c046094bf22521188361a49ba48410ce0fd6cb3e5745a6be42a0ec70

1. Facem un request de upload file pentru a scana fisierul.
2. Facem un request the get file report folosing valoarea SHA256 calculata in programul de Python pentru a vedea raportul despre fisierul incarcat
3. Raspunsul primit de la ultimul request:


```
{
"last_analysis_stats": {
        "harmless": 0,
        "type-unsupported": 15,
        "suspicious": 0,
        "confirmed-timeout": 0,
        "timeout": 0,
        "failure": 0,
        "malicious": 0,
        "undetected": 61
      }
}
```