Autori: Marian Gușatu, Ruxandra F. Olimid
Departamentul de Informatică, Universitatea din București

- Laboratorul 7 -

Introducere în inginerie inversă

Disclaimer: Pe parcursul acestui curs/laborator vi se vor prezenta diverse noțiuni de securitate informatică, cu scopul de a învăța cum să securizați sistemele. Toate noțiunile și exercițiile sunt prezentate în scop didactic, chiar dacă uneori se presupune să gândiți ca un adversar. Nu folosiți aceste tehnici în scopuri malițioase! Acestea pot avea consecințe legale în cazul comiterii unor infracțiuni, pentru care **deveniți pe deplin răspunzători**!

1. *Tipuri de fișiere, fișiere Portable Executable (PE)*

Instalați pe mașina virtuală un editor hex [1]. Citiți despre formatul *Portable Executable (PE)* [2] și despre lista semnăturilor fișierelor [3].

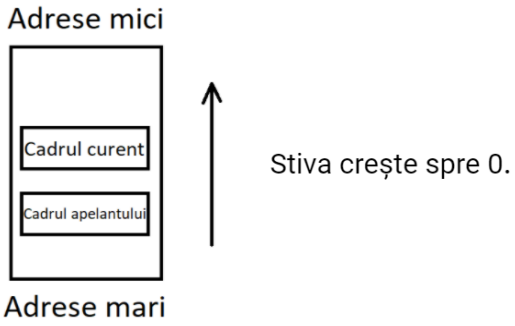
Dezarhivați fișierul ***sample1.zip*** (parola: *infected*). Dezarhivați fișierul ***DLLs.rar*** (parola: *infected*). În fisierul *sample1.zip* veți găsi o imagine. Deschideți imaginea folosind *Windows Photos* (dublu click pe imagine), apoi deschideți imaginea in *HxD*.

Răspundeți la următoarele cerințe:

a) Când ați deschis imaginea prin dublu click ce puteți observa? Este ceva suspicios? A apărut vreo eroare?
b) Când ați deschis imaginea in *HxD* ați sesizat ceva suspicios? Daca da, ce?
c) Încărcați imaginea în *VirusTotal* [4]. Ce rezultate ati obținut?
d) Folosiți *HxD* și extrageți conținutul suspicios, apoi încărcați ceea ce ați extras din imagine în *VirusTotal*. Ce rezultate ați obținut?
e) Fișierele din ***DLLs.zip*** sunt de folos? Argumentați.
f) Este/conține imaginea un malware? Care a fost raționamentul pe care l-ați urmărit?

2. *Vulnerabilități introduse prin programare*

Citiți *p4-511* (pg.1163) din [5] pentru o privire tehnică asupra modului de funcționare al stivei. Structura stivei în arhitectura *Intel 64* si *IA-32* [5] este evidențiată în imaginea următoare:



Reproduceți codul de mai jos in *C++*. Rulați programul și încercați diverse inputuri diferite de parola corectă (i.e., orice input diferit de string-ul „*fmiSSI”*).

- Ce observați?
- Folosind modul în care este aranjată stiva pe *IA-32*, găsiți un input pentru care veți avea afișat mesajul *“Parola introdusă este corectă”*, chiar dacă aceasta este complet diferită de string-ul *“fmiSSI”*.
- Cum se numește aceasta vulnerabilitate/acest atac?

```
#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;
                                }
```

3. Detecția fișierelor pe baza valorii hash

Realizați un script în *Python* care calculează valoarea *SHA256* pentru un fișier de pe disk. Realizați un request către *VirusTotal* folosind *VirusTotal API v3* [6] pentru hash-ul unui fișier și afișați numărul de vendori anti-virus care detectează acel fișier, interpretând răspunsul *JSON* primit.

4. Timestamps

Descărcați *pestudio* [7]. Compilați un program default (*„hello world”*) în *Code::Blocks*[8] (în C/C++). Deschideți binarul obținut (aflat în *numeproiect/bin/Debug*) în *pestudio*.

- Găsiți data la care a fost compilat binarul respectiv (faceți o poză ecranului cu informația din *pestudio*).
- Deschideți binarul și în *HxD* și identificați zona în care este prezentă data de compilare a binarului respectiv. Puteți folosi un *timestamp converter* [9] pentru a verifica dacă data este corectă.

Referințe bibliografice

1. HxD – Freeware Hex Editor and Disk Editor. Accesibil la: https://mh-nexus.de/en/hxd/
2. Medium Ax1al. *A brief introduction to PE format*. Accesibil la: https://medium.com/ax1al/a-brief-introduction-to-pe-format-6052914cc8dd
3. Wikipedia. *List of file signatures*. Accesibil la: https://en.wikipedia.org/wiki/List_of_file_signatures
4. VirusTotal. Accesibil la: https://www.virustotal.com/gui/home/upload
5. Intel. *Intel® 64 and IA-32 Architectures Software Developer's Manual - Volume 2 (2A, 2B, 2C & 2D): Instruction Set Reference, A-Z* Accesibil la: https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-instruction-set-reference-manual-325383.pdf
6. VirusTotal. *VirusTotal API v3 Overview*. Accesibil la: https://developers.virustotal.com/reference/overview
7. pestudio. Accesibil la: https://www.winitor.com/download
8. Code::Blocks. Accesibil la: https://www.codeblocks.org/
9. Unix Hex Timestamp Converter. Accesibil la: https://www.epochconverter.com/hex