C.4 MATERIALE ȘI METODE ÎN INGINERIA INVERSĂ (I)

PAUL A. GAGNIUC

Academia Tehnică Militară „Ferdinand I”

PRINCIPALELE PĂRȚI ALE PREZENTĂRII

C.4 Materiale și metode în ingineria inversă:

- C.4.1 MEDIUL DE DETONARE
- C.4.2 INSTRUMENTELE DE INGINERIE INVERSĂ (I)

PACHETUL DE FIȘIERE
PENTRU INSTRUMENTE

- kit\Gazda\VirtualBox.exe
- kit\Gazda\win7.iso
- kit\Gazda\IDA.zip

(Gazdă)

- kit\Gazda\PEstudioX86.zip
- kit\Gazda\Cutter.zip
- kit\Gazda\x64dbg.zip
- kit\Masina Virtuala\sysinternals.zip
- kit\Masina Virtuala\wireshark.zip
- kit\Masina Virtuala\procdot.zip
- kit\Masina Virtuala\pestudio.zip
- kit\Masina Virtuala\BCompare.exe

(Masina Virtuală)

C.4.1

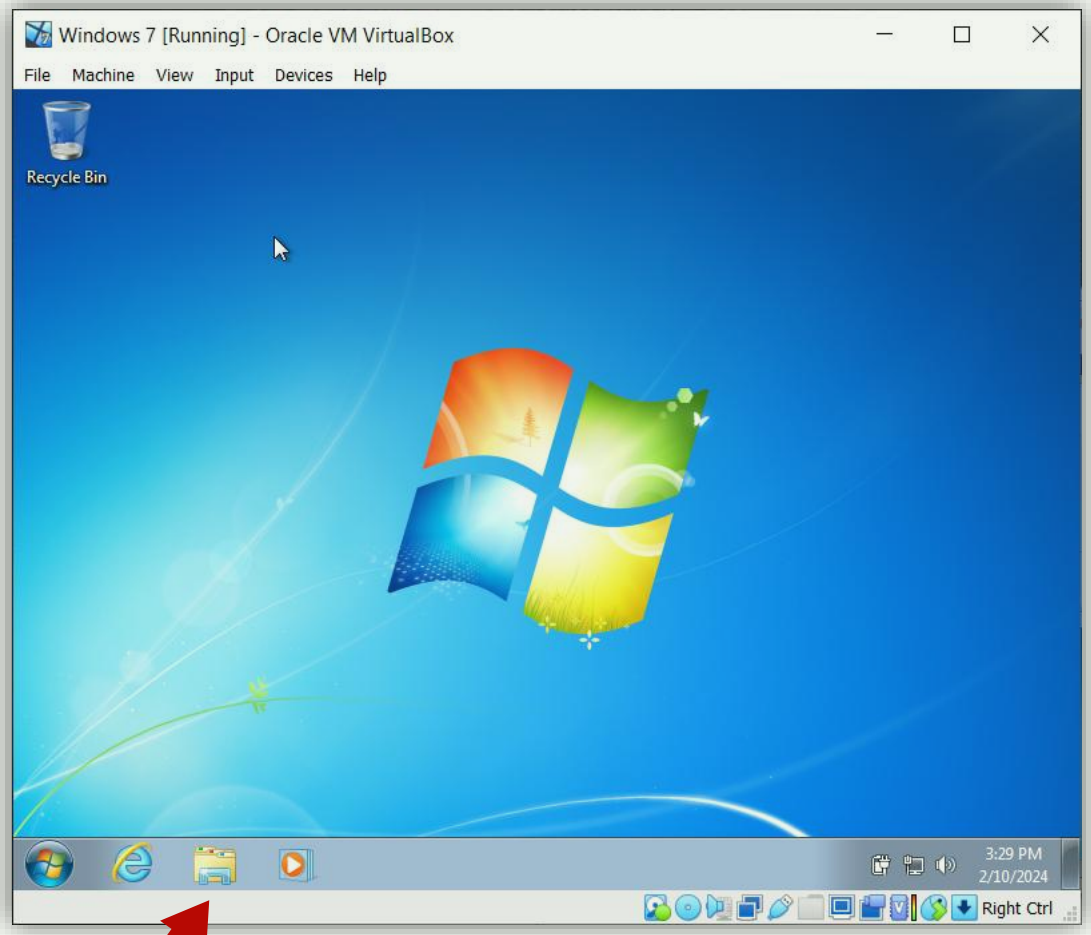
MEDIUL DE DETONARE

Afișarea extensiilor și a fișierelor ascunse !

De ce Windows 7 pe mașina virtuală?

Windows 7 reprezintă echilibrul perfect între dimensiunea kit-ului de instalare și structura modernă a sistemului de operare (ex. Windows 11):

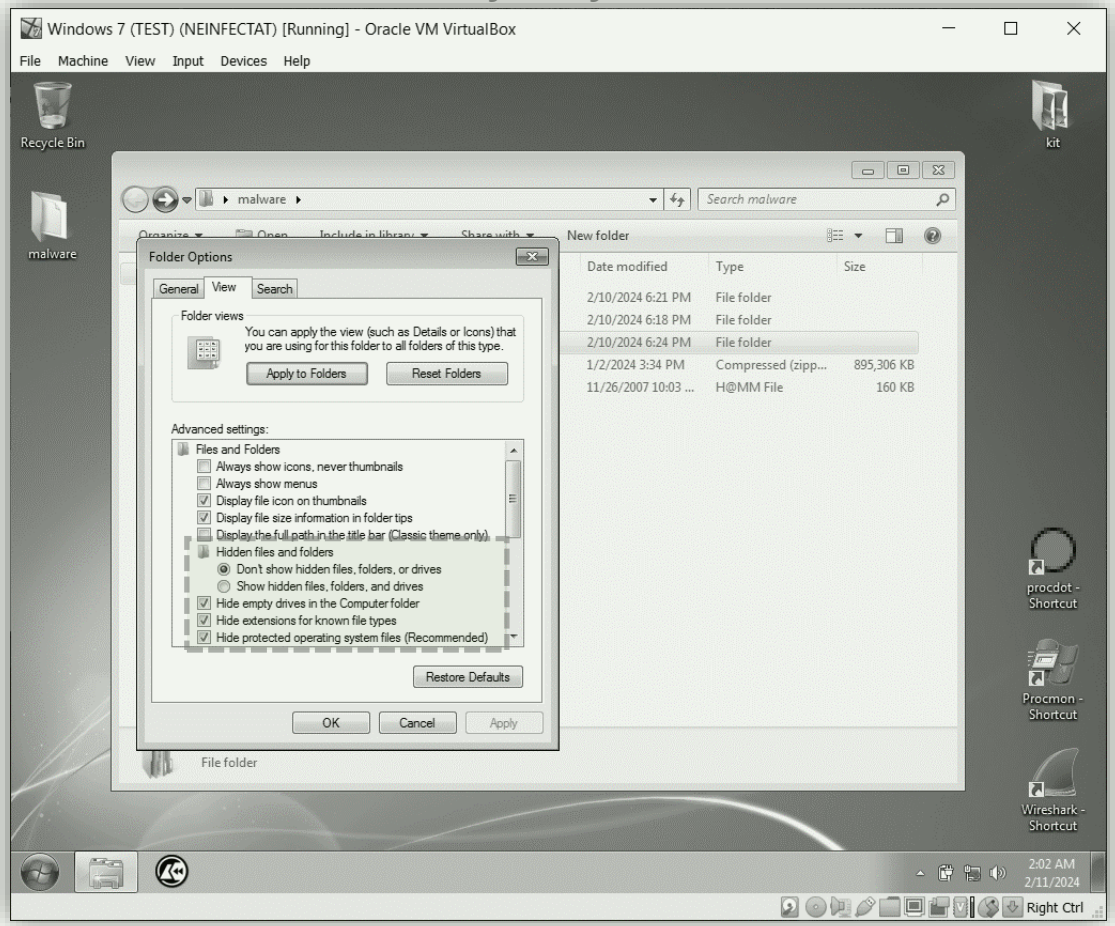
File explorer/Organize/Folder and search options:



1

Deselectare: *Hide OS files*; *Hide extensions*

Selectare: *Show hidden files, folder*, and *drives*



Eliminarea zgomotului de fundal: Oprirea serviciilor

Pentru a reduce activitatea în background („zgomotul”) pe o mașină virtuală cu Windows 7 (valabil și pentru versiunile moderne de Windows), astfel încât în monitorul de procese să fie vizibil doar programul pe care îl executați, puteți dezactiva sau configura următoarele servicii și funcționalități:

- **Indexing Service (Indexare).** Dacă nu aveți nevoie de căutare rapidă prin fișiere, dezactivați acest serviciu pentru a reduce activitatea pe disc și procesor.
- **System Restore (Restaurare Sistem).** Deactivați restaurarea sistemului pentru a preveni crearea punctelor de restaurare în fundal.
- **Windows Update (Actualizări Windows).** Dezactivați actualizările automate Windows pentru a opri verificarea și instalarea actualizărilor în fundal.
- **Windows Defender (sau alte soluții antivirus).** Antivirusul poate efectua scanări periodice care pot interfera cu monitorizarea. Deactivați-l dacă mediul este controlat și sigur.
- **Windows Search (Căutare Windows).** Similar cu serviciul de indexare, acesta poate fi oprit pentru a reduce activitatea discului.
- **Remote Registry (Registrul la Distanță).** Oprește modificările la distanță ale registrului, care de obicei nu sunt necesare.
- **Print Spooler (Listă de așteptare pentru tipărire).** Dacă nu tipăriți de pe mașina virtuală, acest serviciu poate fi oprit.
- **Background Intelligent Transfer Service (BITS).** Oprește transferurile în fundal de date folosind lățimea de bandă neutilizată.
- **Superfetch.** Deși este proiectat să îmbunătățească performanța sistemului, poate fi dezactivat pentru a reduce utilizarea discului și a memoriei.
- **Diagnostic Tracking Service (Serviciul de Urmărire Diagnostic).** Dezactivați acest serviciu pentru a preveni colectarea datelor despre performanța sistemului.
- **Secondary Logon (Autentificare Secundară).** Permite rularea proceselor sub un alt utilizator și poate fi dezactivat dacă nu este necesar.
- **Event Log (Jurnal de Evenimente).** Dacă nu aveți nevoie de jurnale detaliate, puteți să opriți acest serviciu, deși este recomandat să fie lăsat activ pentru depanare și monitorizare de securitate.

OPRIREA SERVICIILOR

PENTRU A DEZACTIVA ACESTE SERVICII, URMAȚI ACEȘTI PAȘI

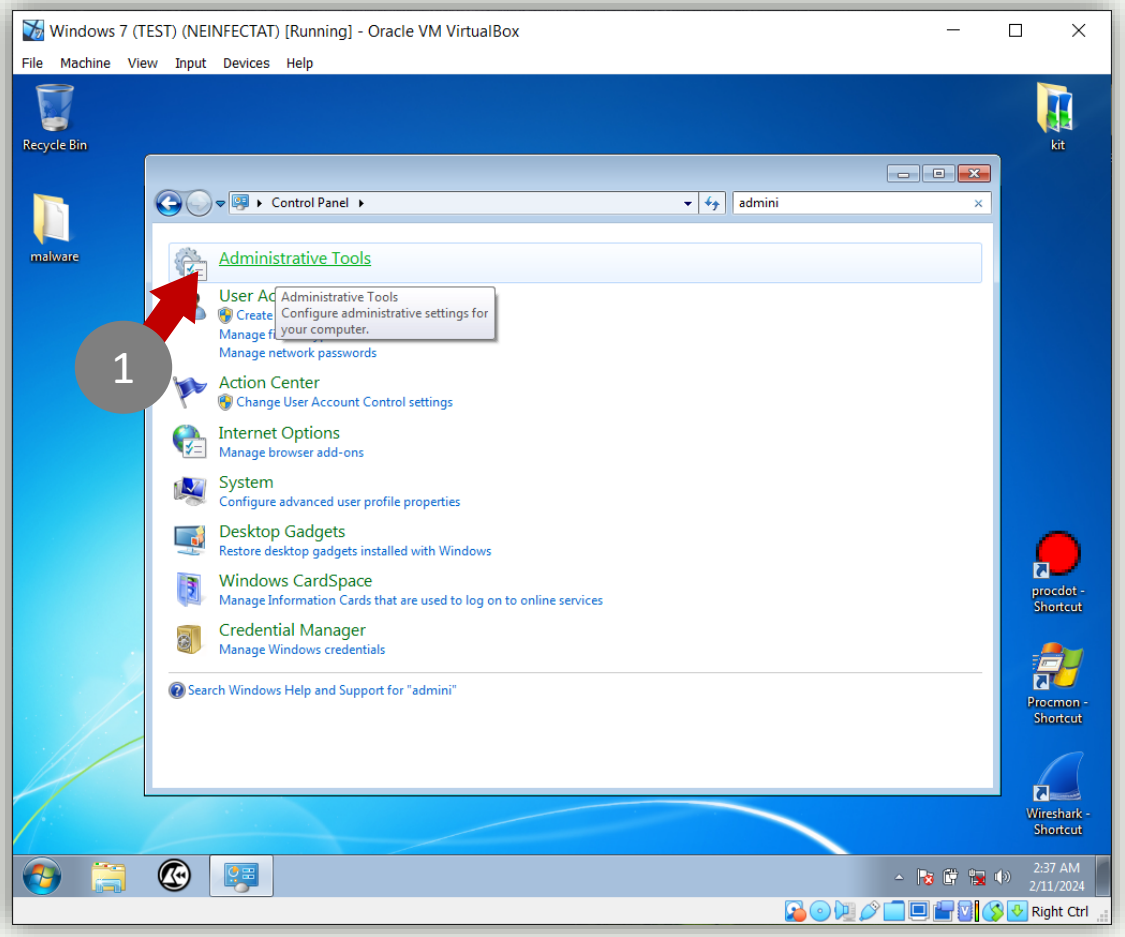
- Deschideți services.msc prin Start > Run/Căutare și introducând **services.msc**.
- Căutați serviciul pe care doriți să-l dezactivați.
- Faceți clic dreapta pe serviciu și selectați **Properties** (Proprietăți).
- În tabul **General**, schimbați **Startup type** (Tipul de pornire) în **Disabled** (Dezactivat).
- Faceți clic pe **Stop** (Oprire) pentru a opri serviciul imediat.
- Aplicați schimbările și închideți fereastra.

Rețineți că dezactivarea unor servicii esențiale poate afecta stabilitatea și funcționalitatea sistemului. Este întotdeauna o practică bună să creați un punct de restaurare înainte de a face astfel de schimbări, astfel încât să puteți reveni la o configurație anterioară dacă apare o problemă.

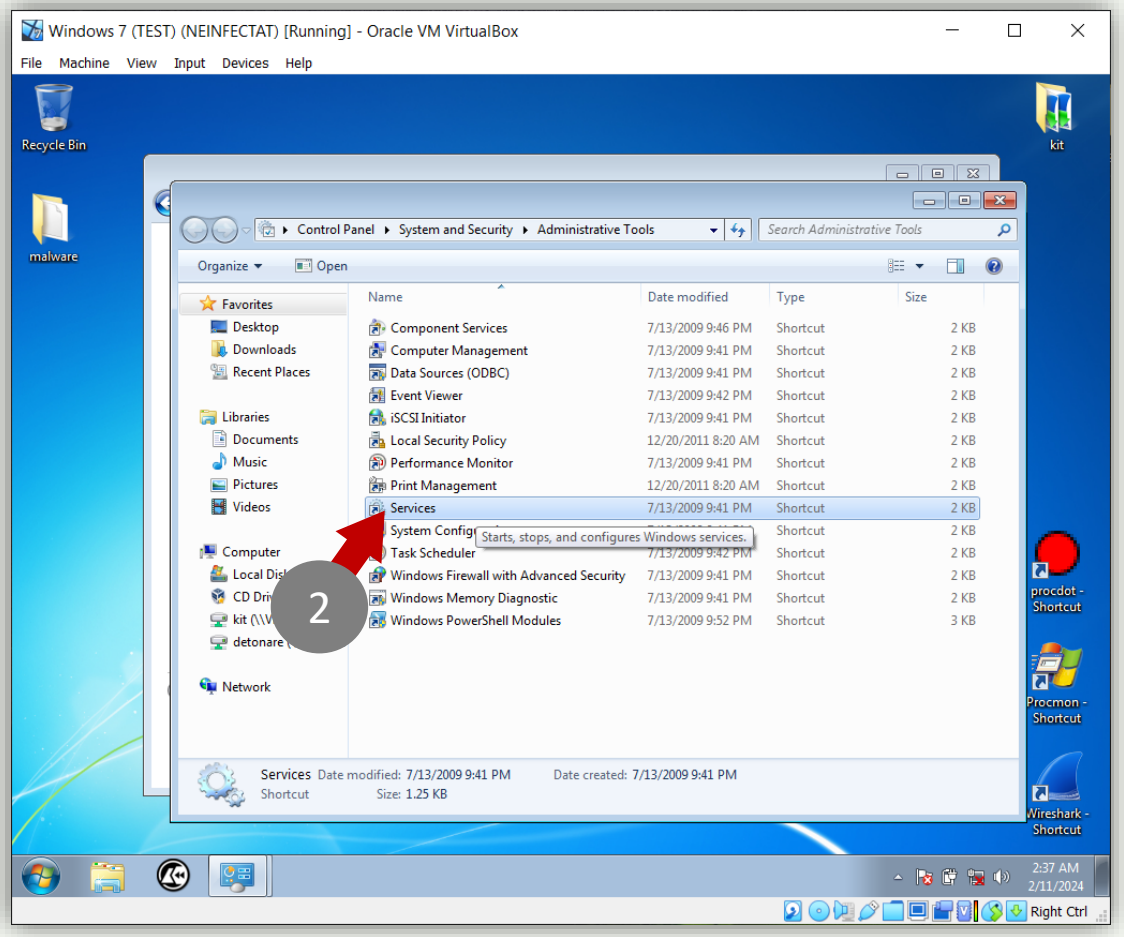
Dezactivarea serviciilor (I)

- Utilizare: Administrative tools
- Selectați opțiunea: Services

Administrative tools



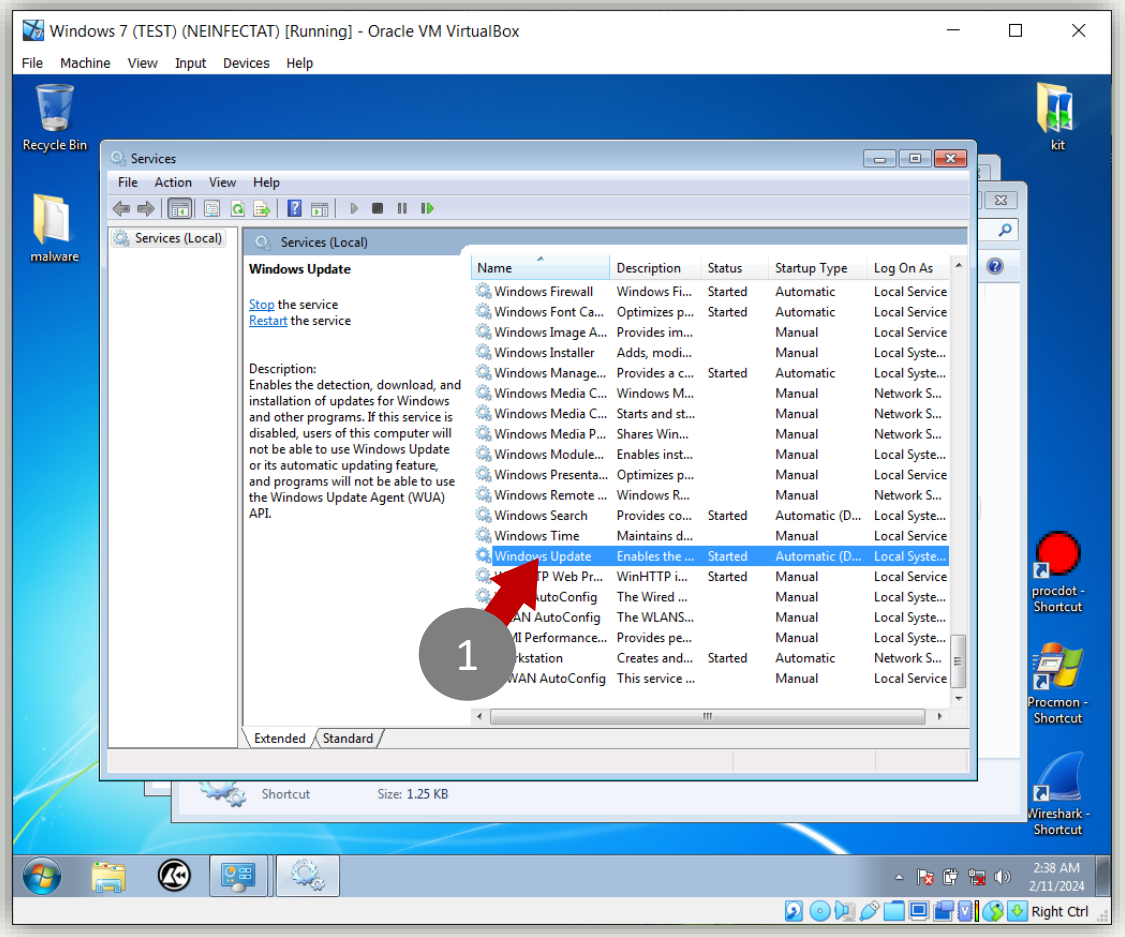
Services

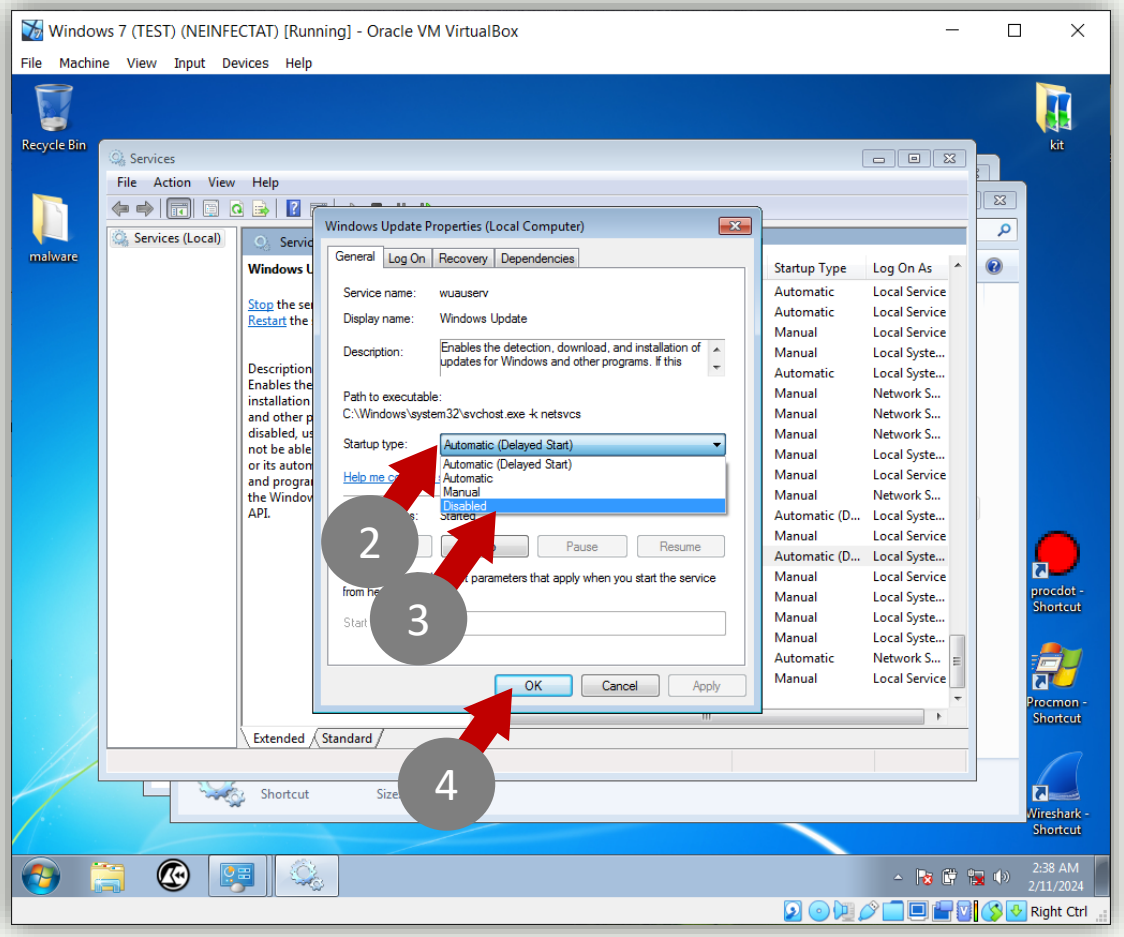


Dezactivarea serviciilor (II)

- Selectare serviciu: Windows Update
- Oprire serviciu (Startup type: Disabled)

Dezactivare: Windows Update

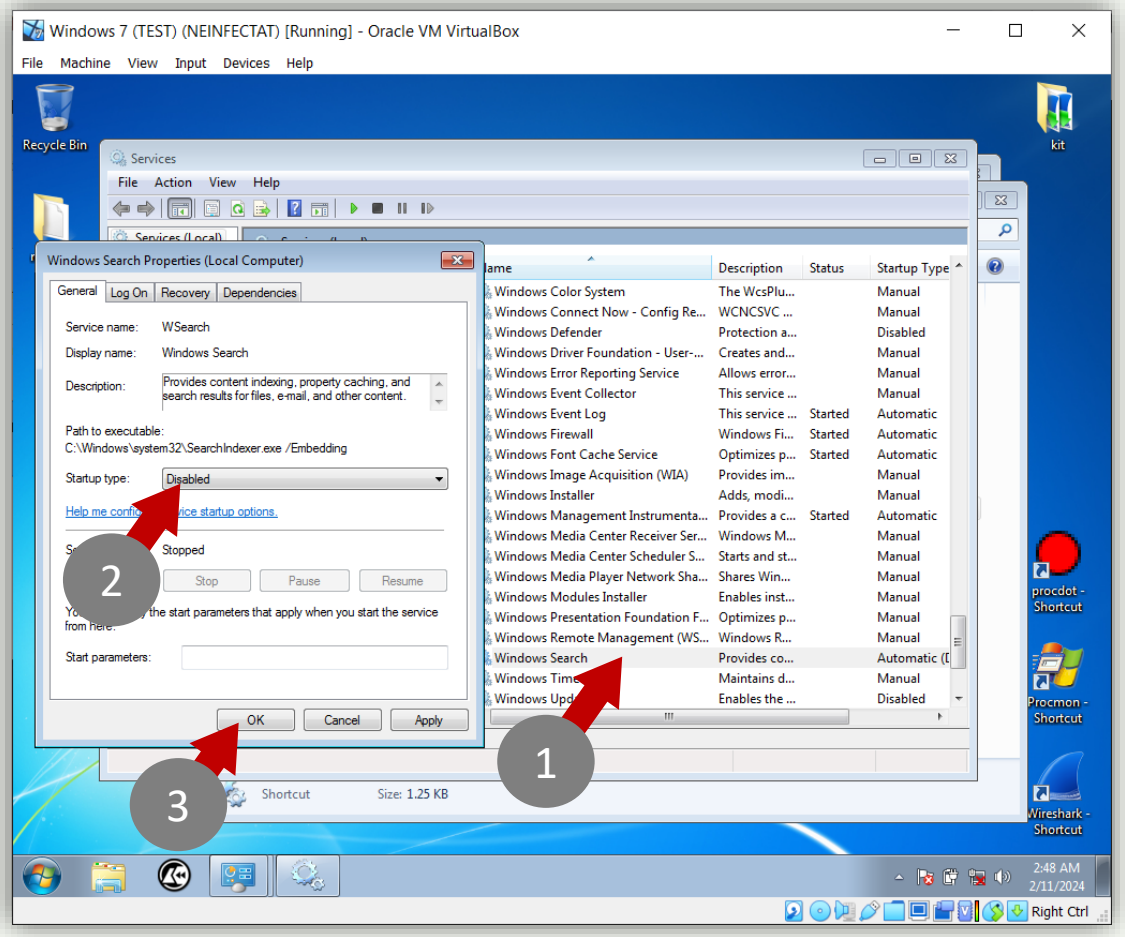




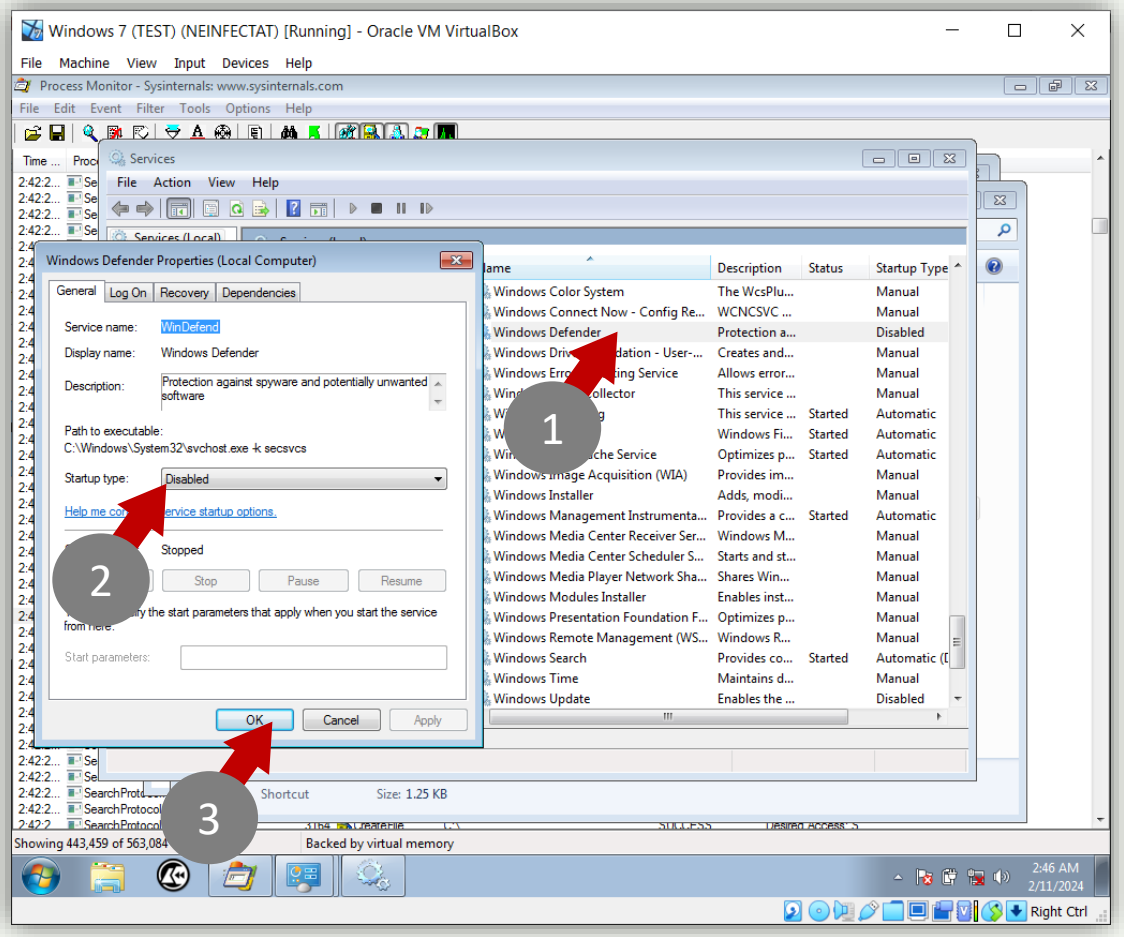
Dezactivarea serviciilor (III)

- Selectare serviciu: Windows Defender
- Oprire serviciu (Startup type: Disabled)

Dezactivare: Windows Search



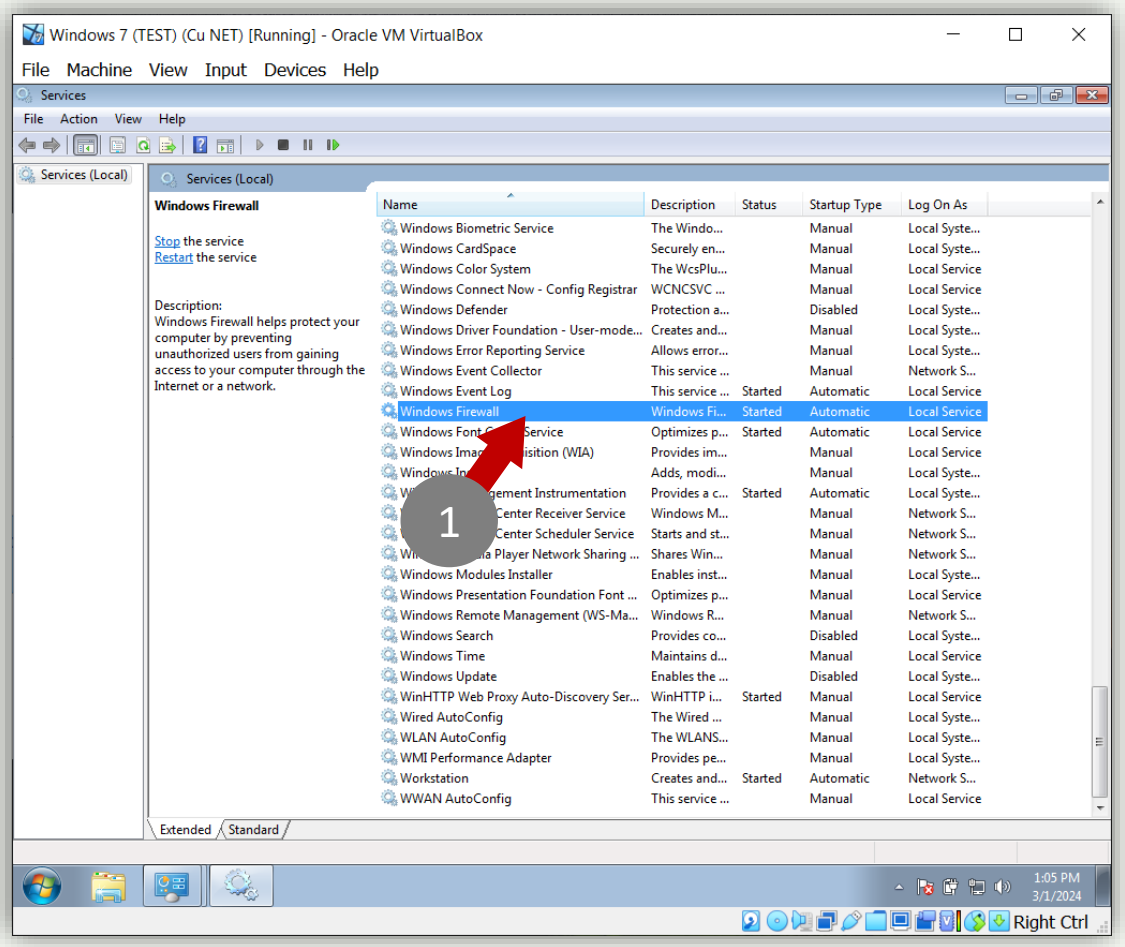
Dezactivare: Windows Defender

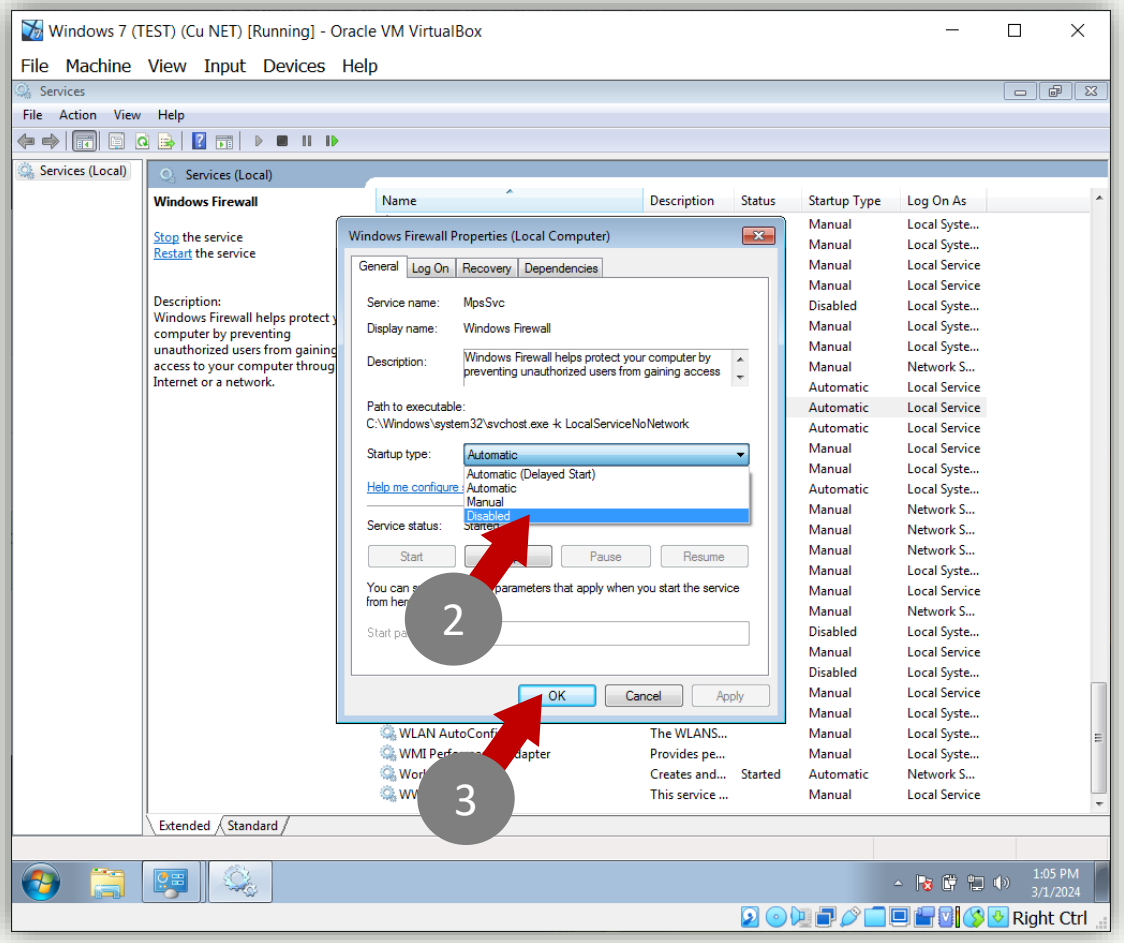


Dezactivarea serviciilor (IV)

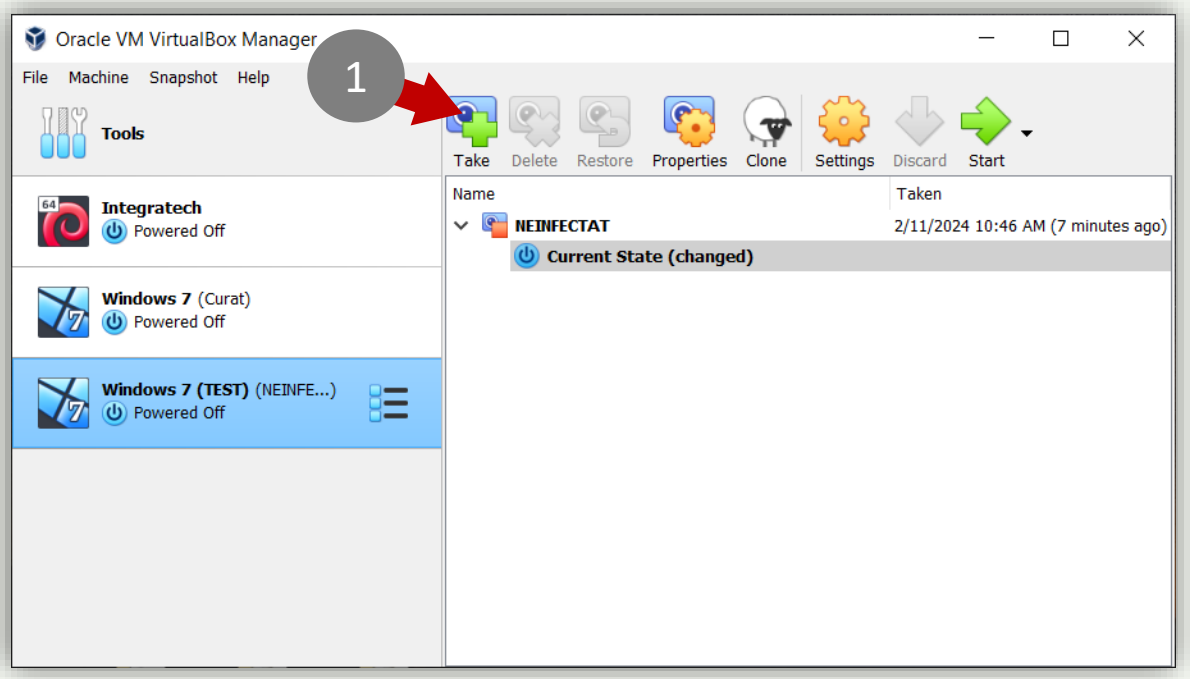
- Selectare serviciu: Windows Firewall
- Oprire serviciu (Startup type: Disabled)

Dezactivare: Windows Firewall





MAȘINA VIRTUALĂ SALVARE STARE:



- Salvare stare: permite păstrarea tuturor setărilor efectuate până în acest moment.
- Până în acest moment avem o mașină virtuală care rulează Windows 7, cu toate setările necesare pentru a detona mostrele malware.

INSTALAREA MAȘINII VIRTUALE PE LINUX

VIRTUALBOX GUEST ADDITIONS

OFERĂ FUNCȚIONALITĂȚI ÎMBUNĂTĂȚITE PENTRU MAȘINILE VIRTUALE PE LINUX

C.4.2

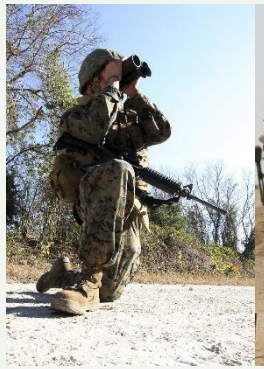
INSTRUMENTELE DE INGINERIE INVERSĂ

Academia Tehnică Militară „Ferdinand I”

Metodologie:

Analiza Statică

Măsoară potențialul



Analiza Dinamică

Observați comportamentul

Analiza Cloud

La distantă:
potenţial şi comportament

ANALIZA STATICĂ:

Inventariere/Inginerie inversă

COD MALIȚIOS: INSTRUMENTE DE ANALIZĂ

- **PEstudio**
- Bcompare
- Cutter & IDA & x64dbg

Analiză Statică

- Sysinternals
- Wireshark
- procDOT

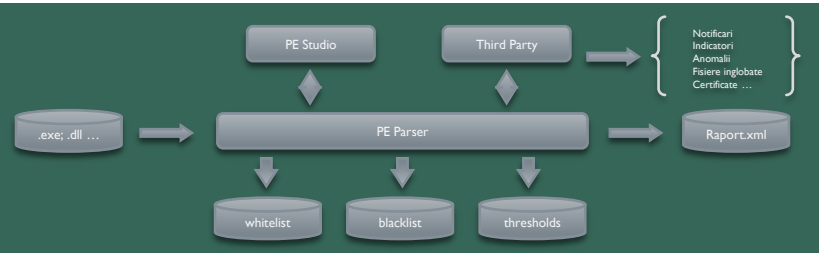
Analiză Dinamică

- Analiza malware in Cloud

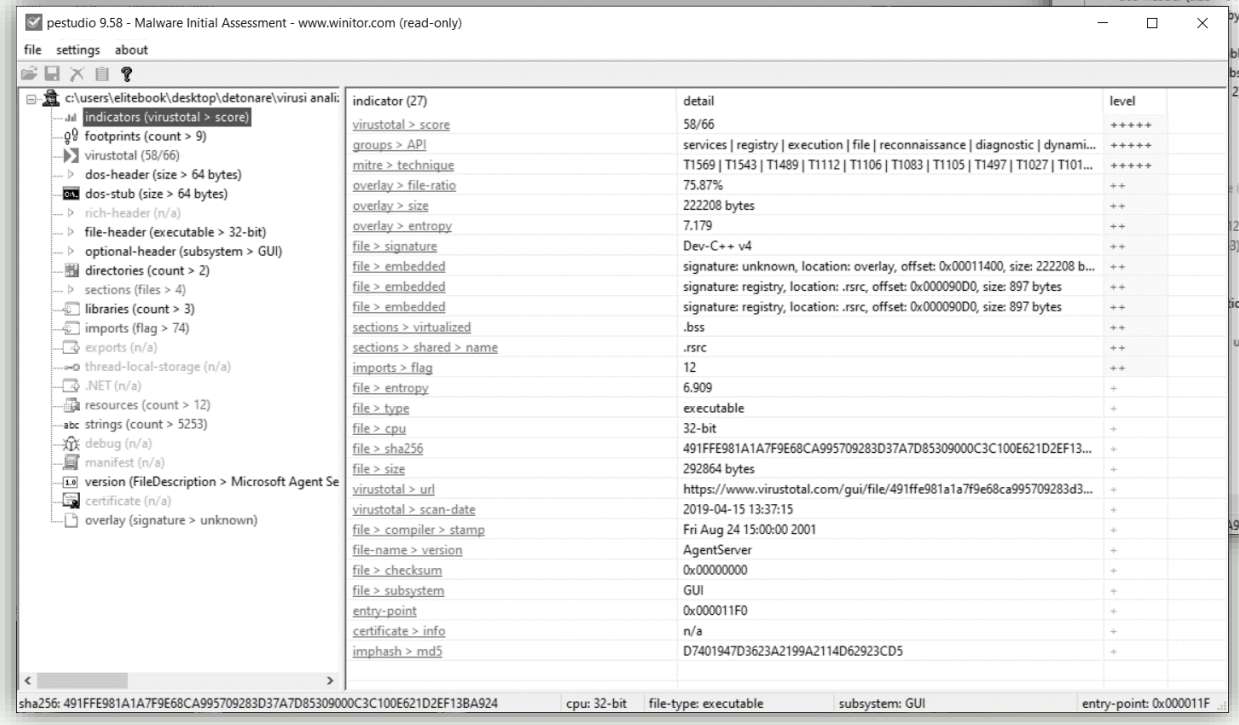
Analiză Cloud

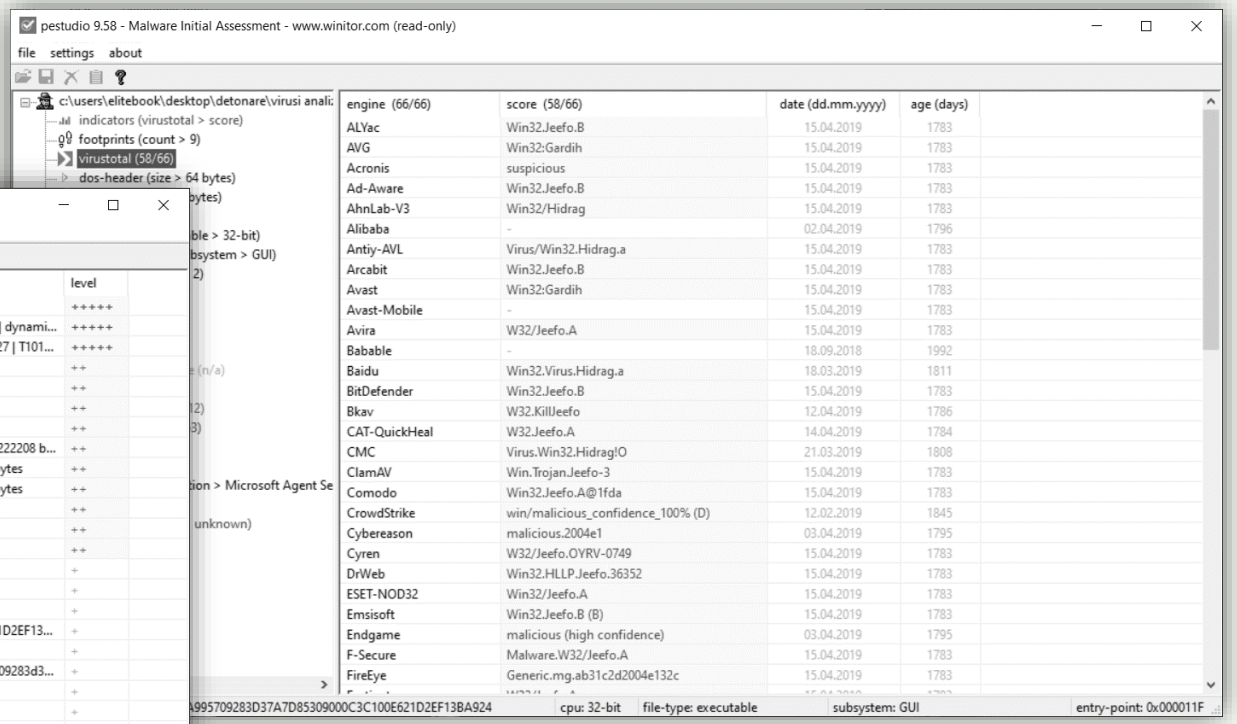
PE STUDIO

INDICATORI & CLOUD (TEST BANCA MALWARE)



Înțelegerea indicatorilor și verificarea în cloud:



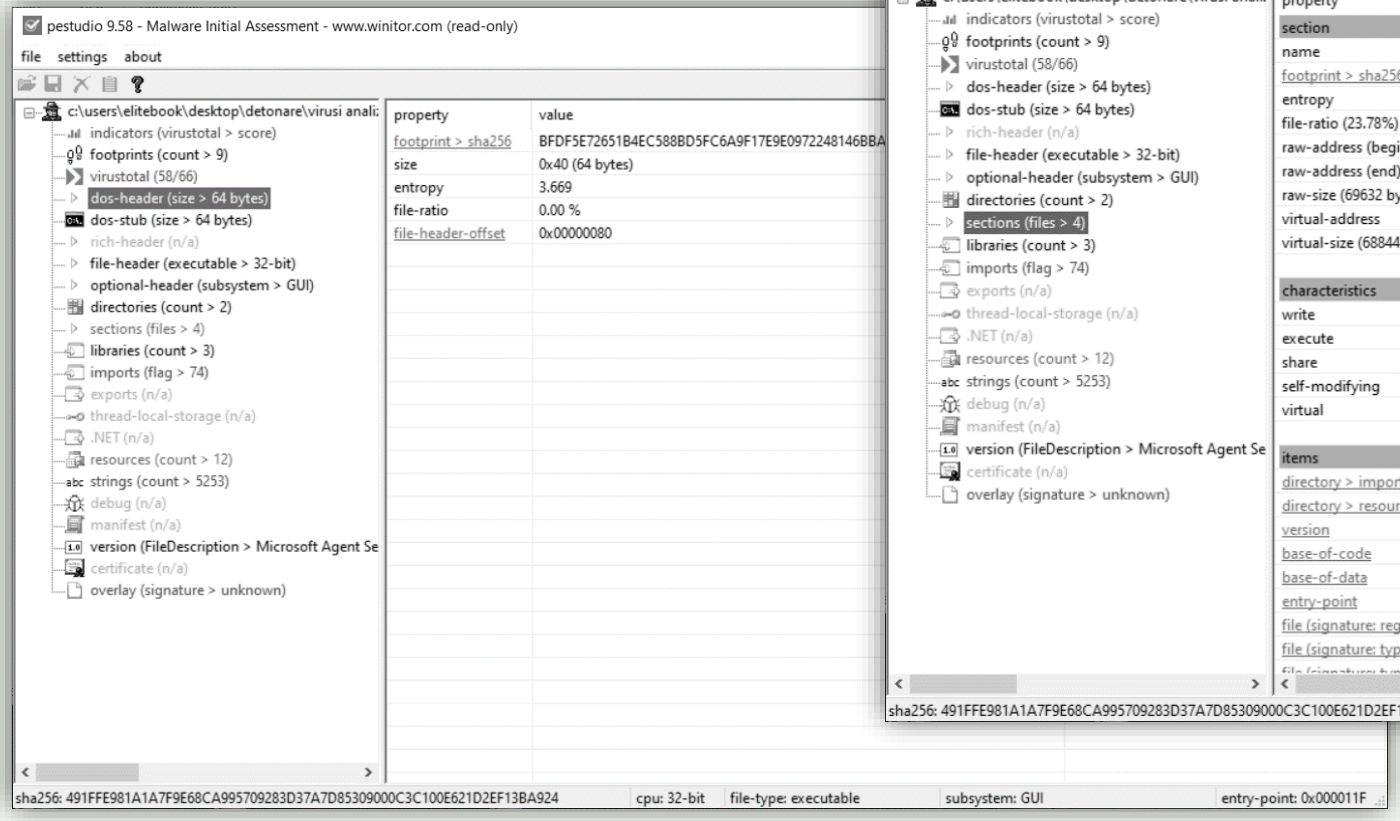


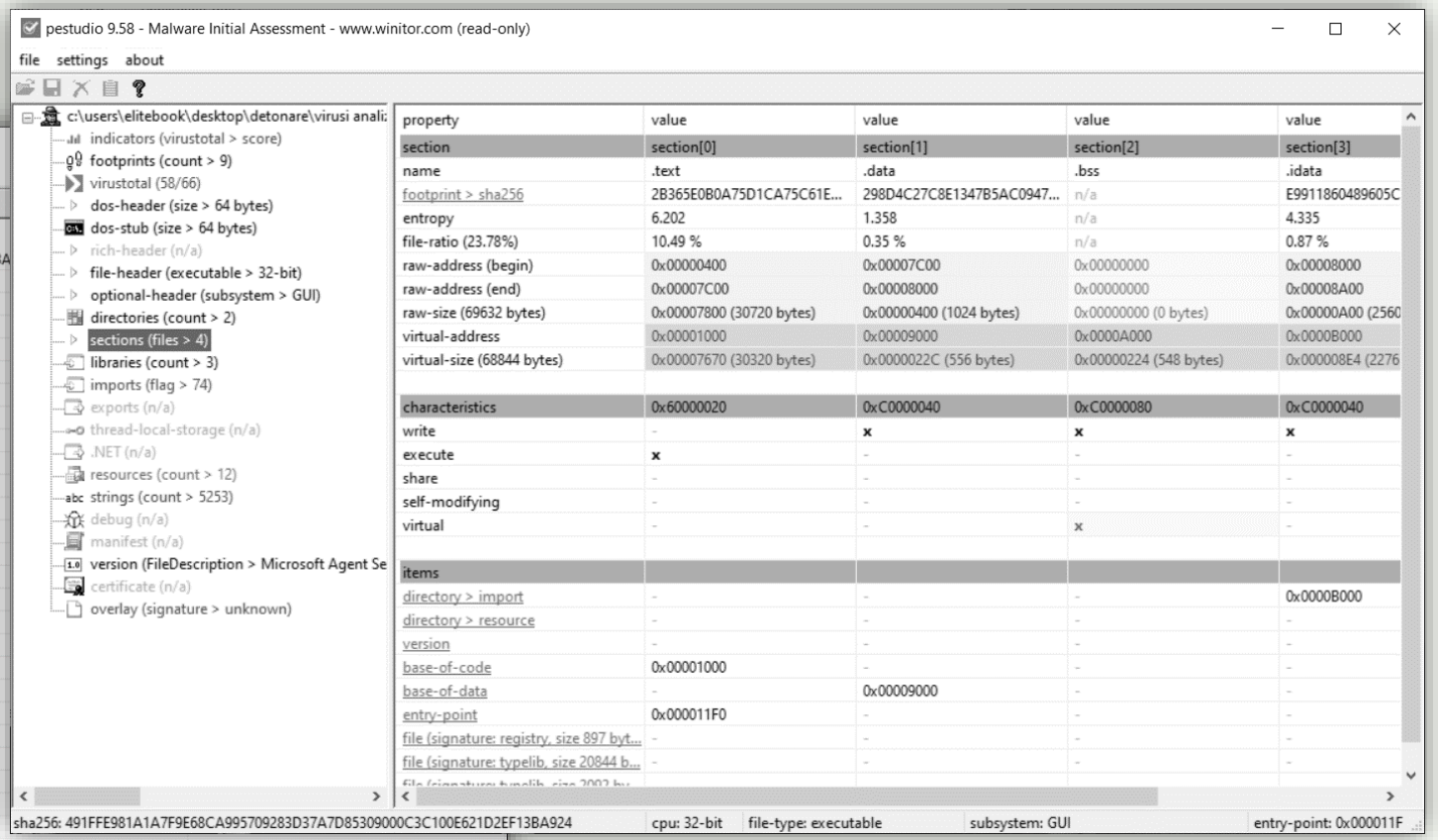
Fișier PE infectat luat din banca de malware !

PE STUDIO

SECTIUNI EXECUTABIL & ENTROPIE

- Capul de fisier



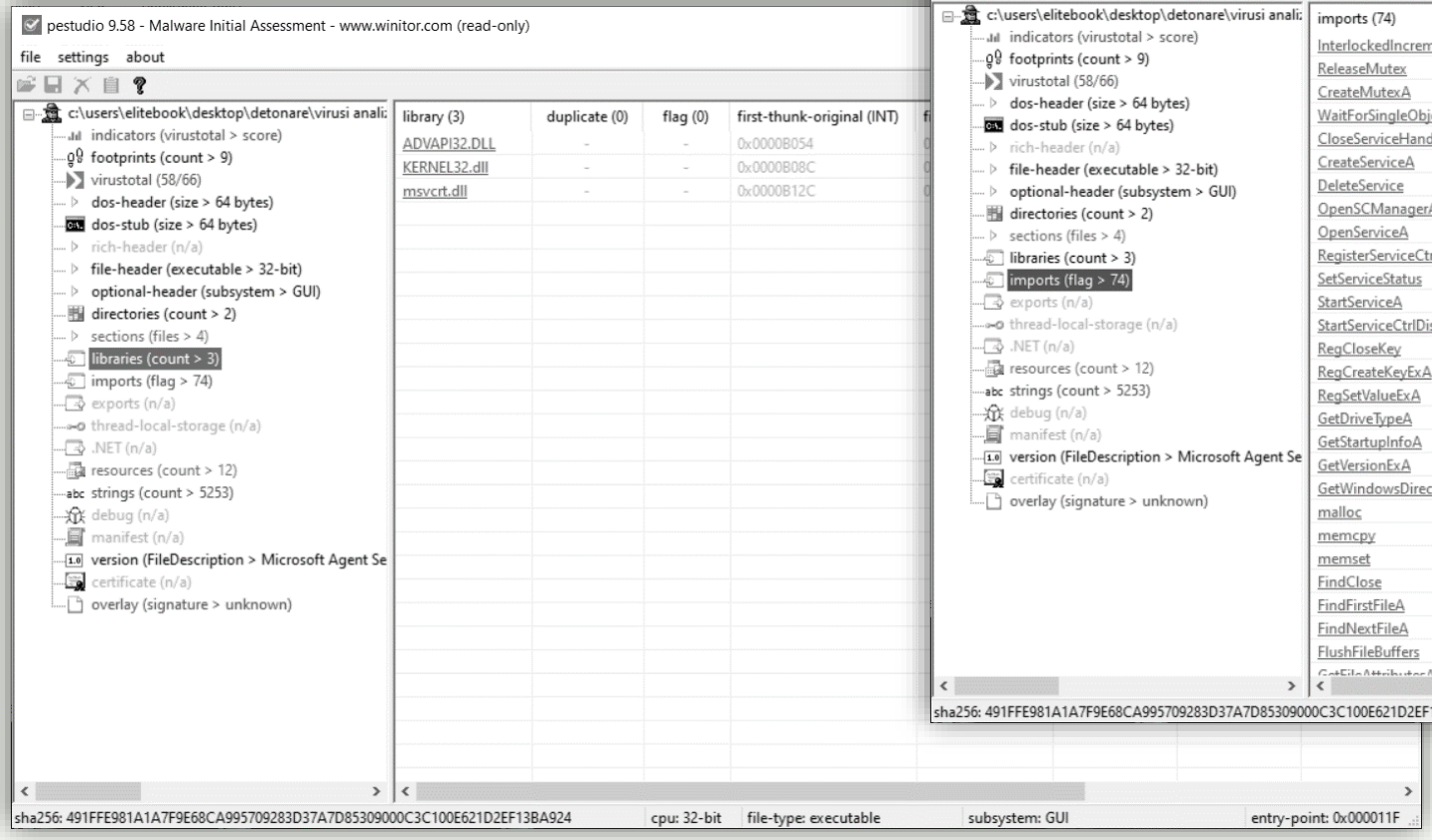


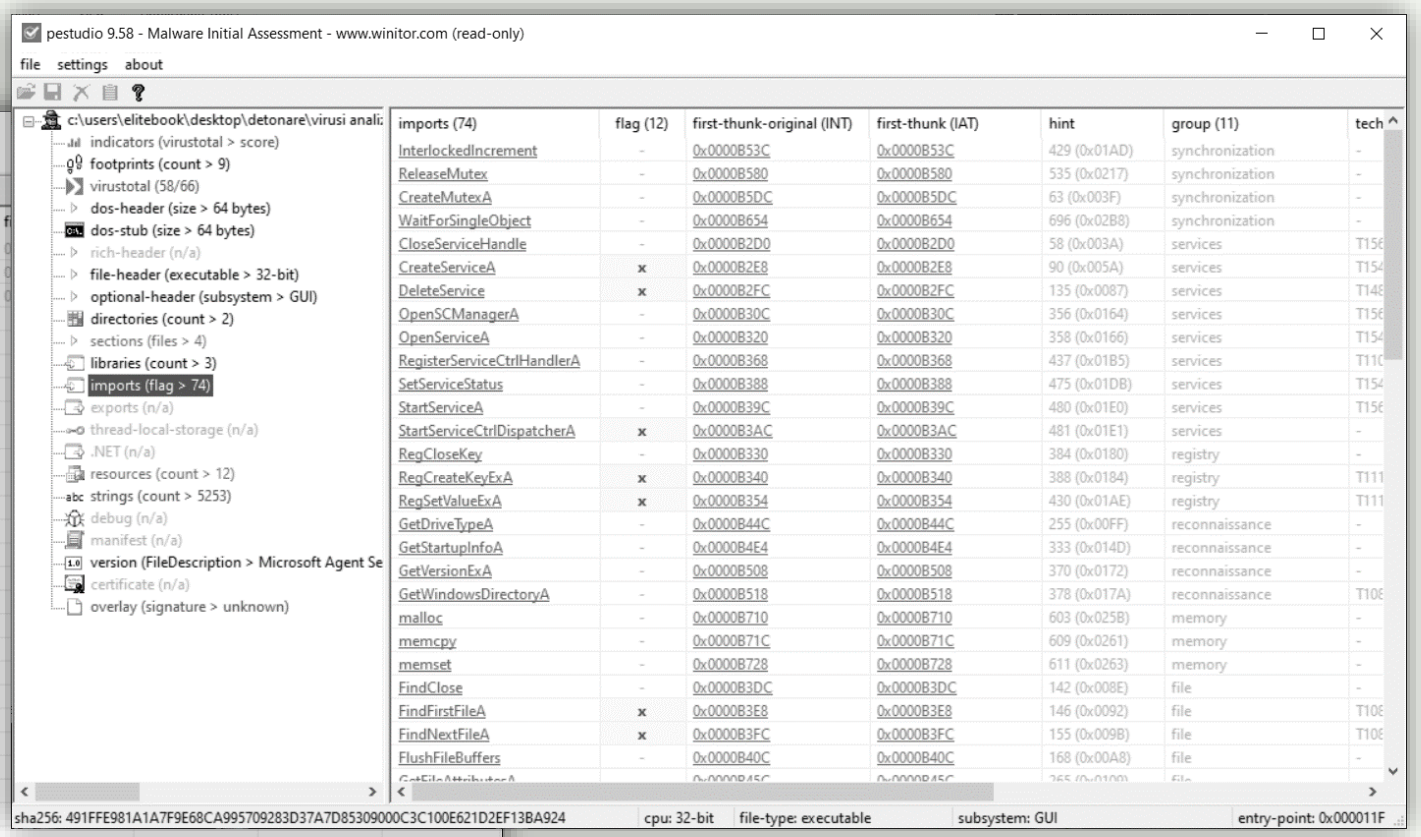
- Secțiuni în fișierele PE

PE STUDIO

LIBRARII & API

- Librării folosite



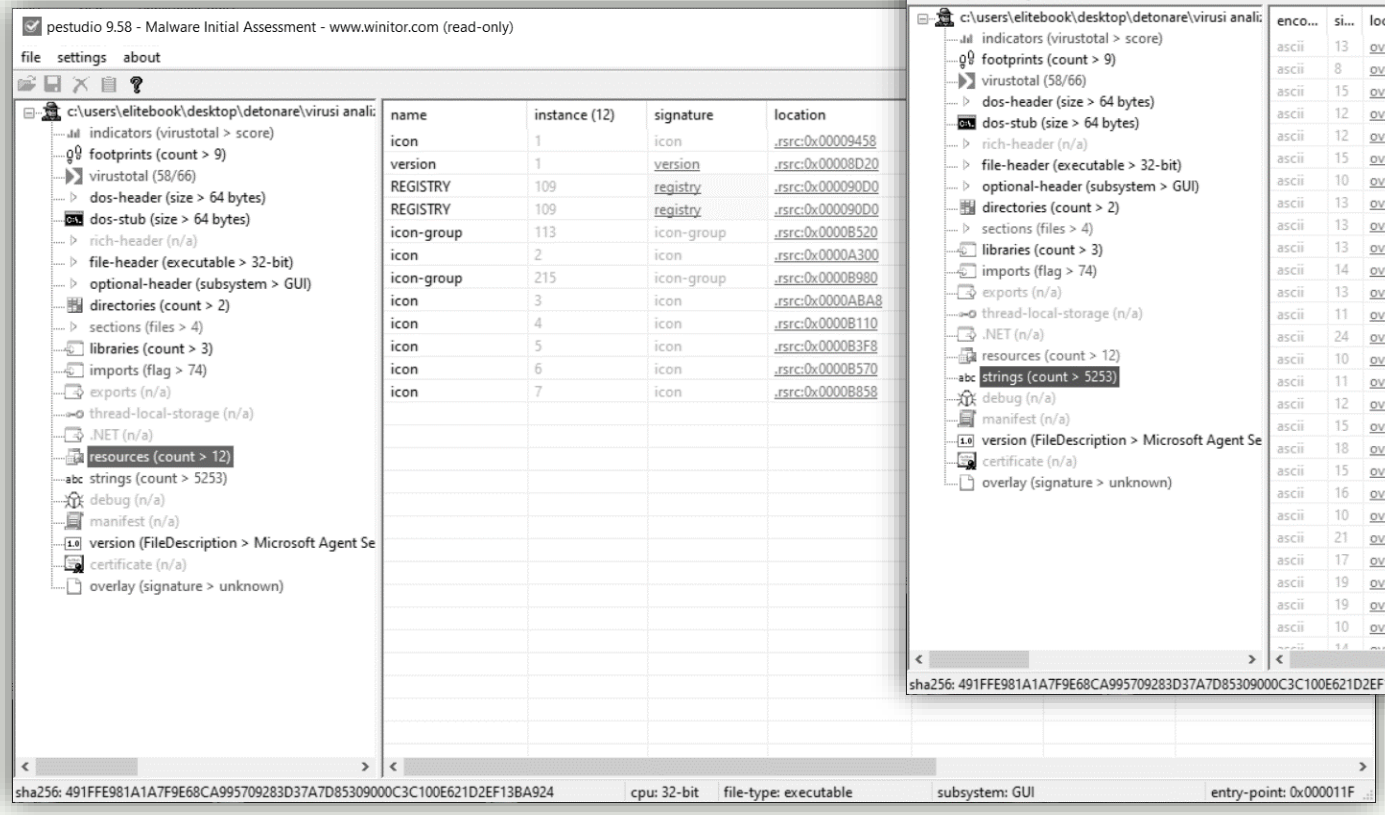


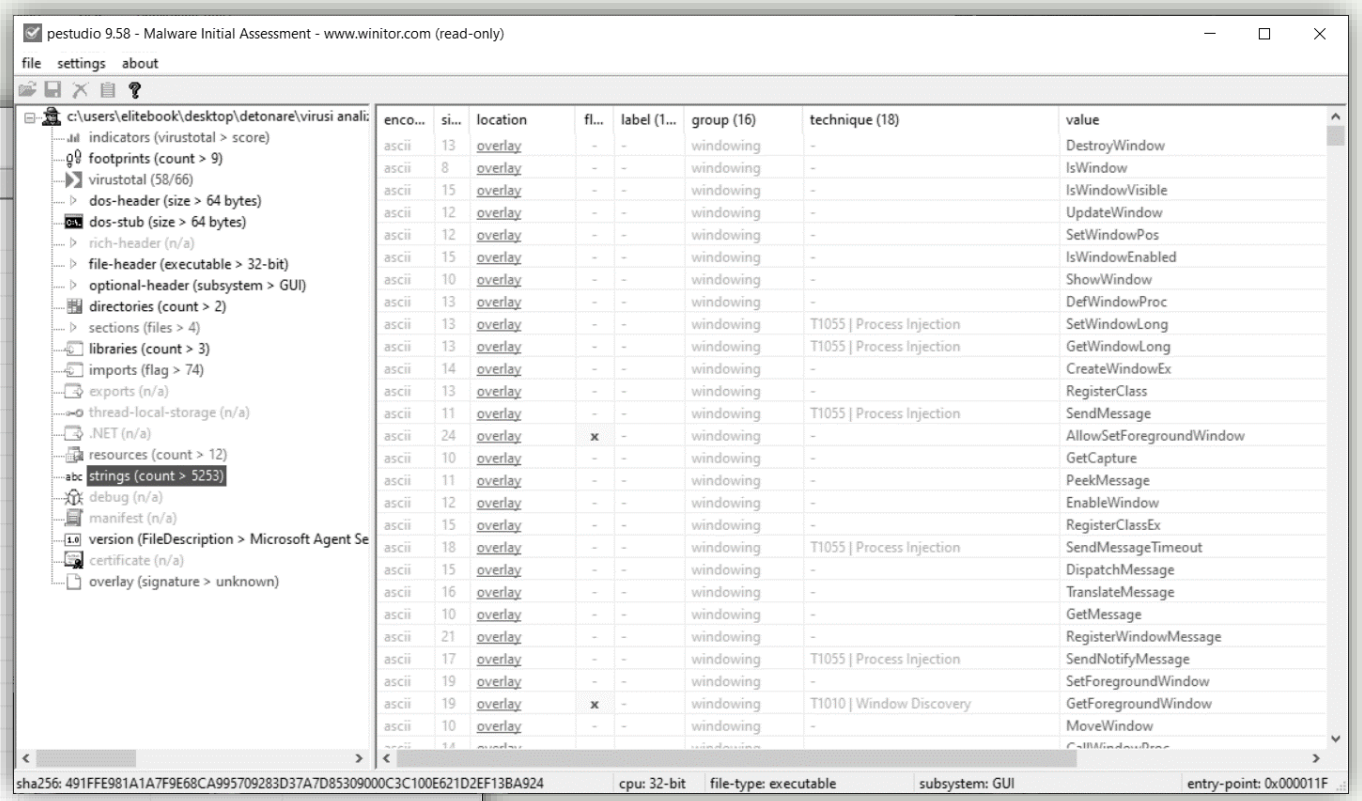
- API folosit

PE STUDIO
RESURSE & STRINGURI

- Resurse
- Strings

- Resursele executabilului





- Șiruri găsite în executabil

COD MALIȚIOS: INSTRUMENTE DE ANALIZĂ

- PEstudio
- **Bcompare**
- Cutter & IDA & x64dbg

Analiză Statică

- Sysinternals & Process Hacker
- Wireshark
- procDOT

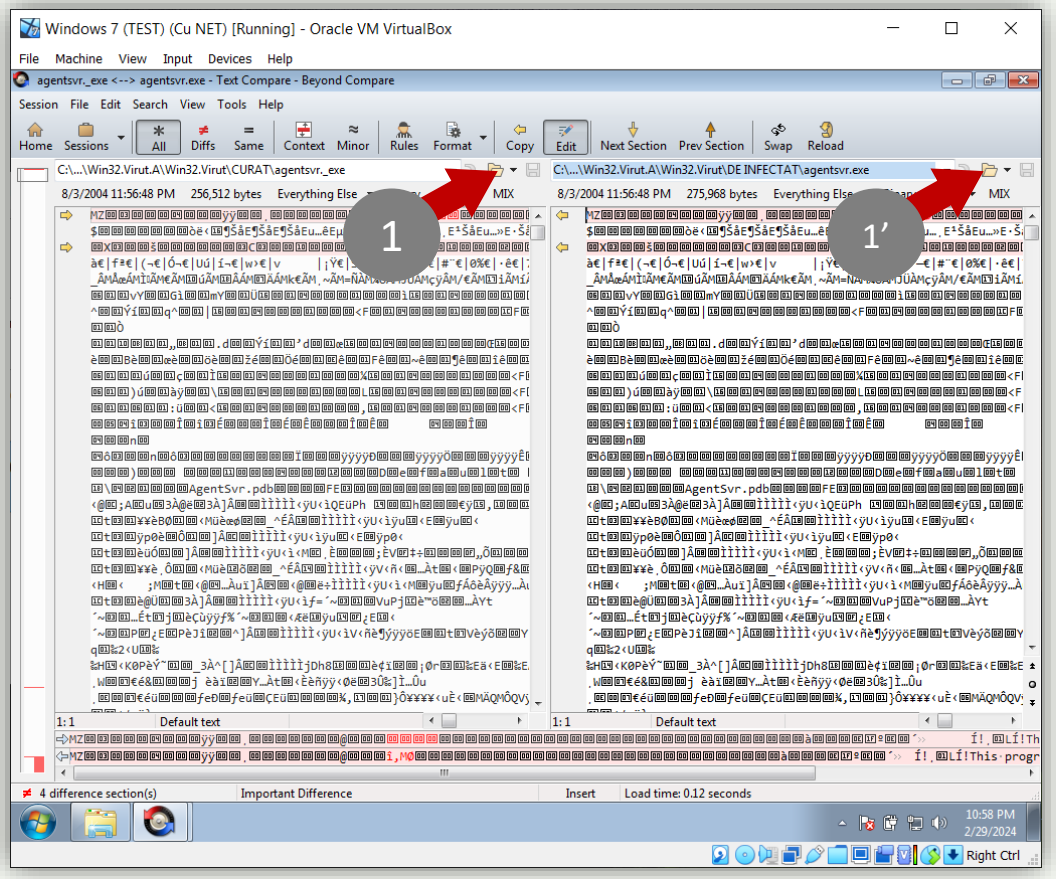
Analiză Dinamică

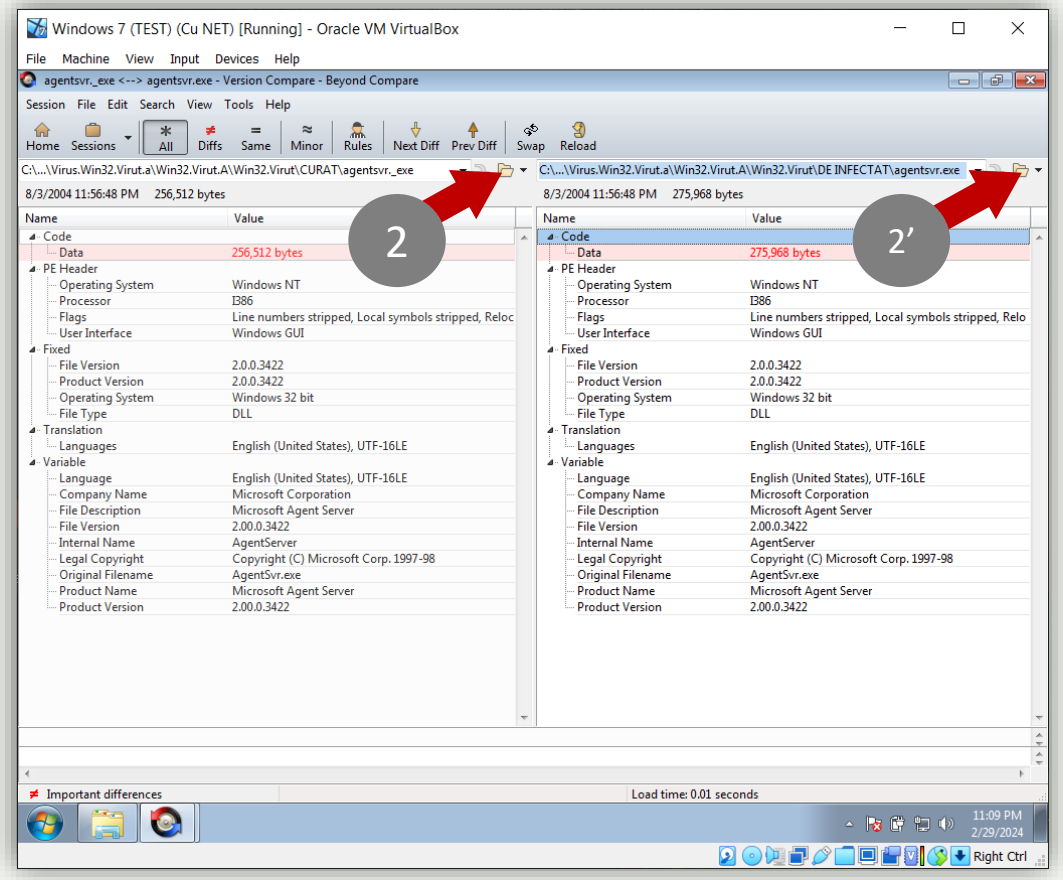
- Analiza malware in Cloud

Analiză Cloud

BCOMPARE

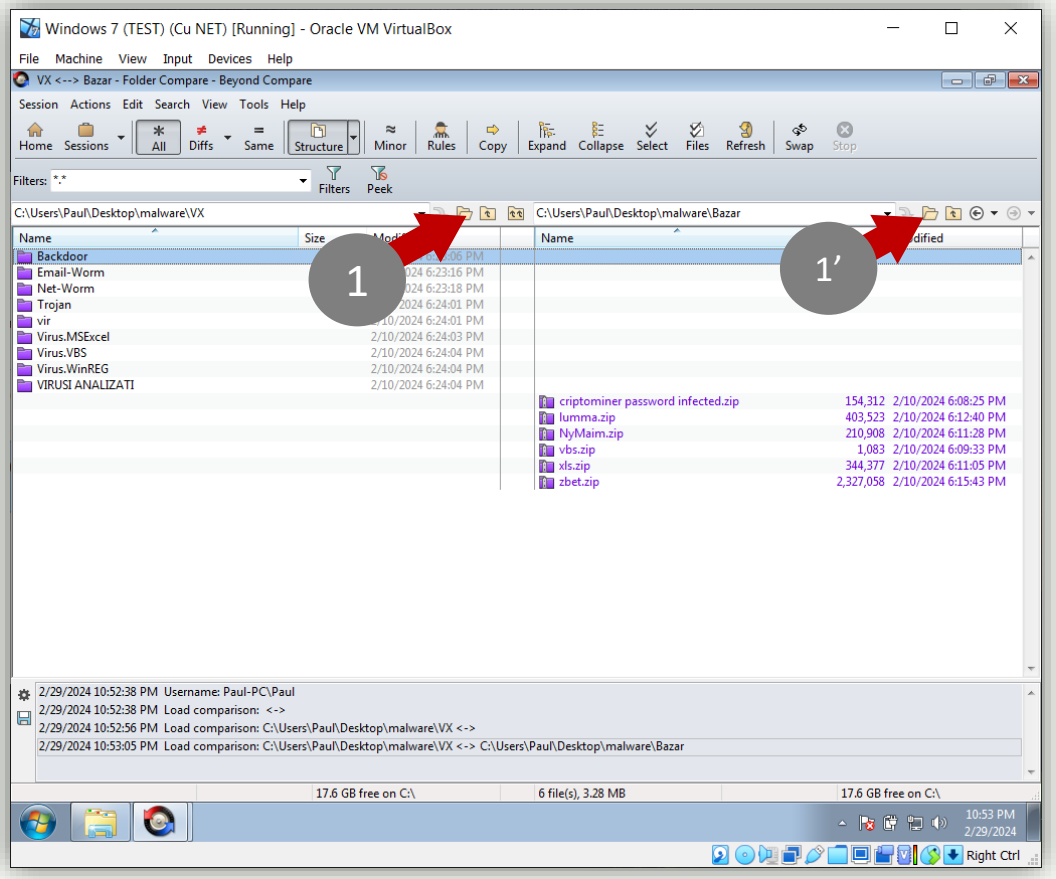
COMPARAREA TEXTULUI ȘI COMPARAREA VERSIUNILOR

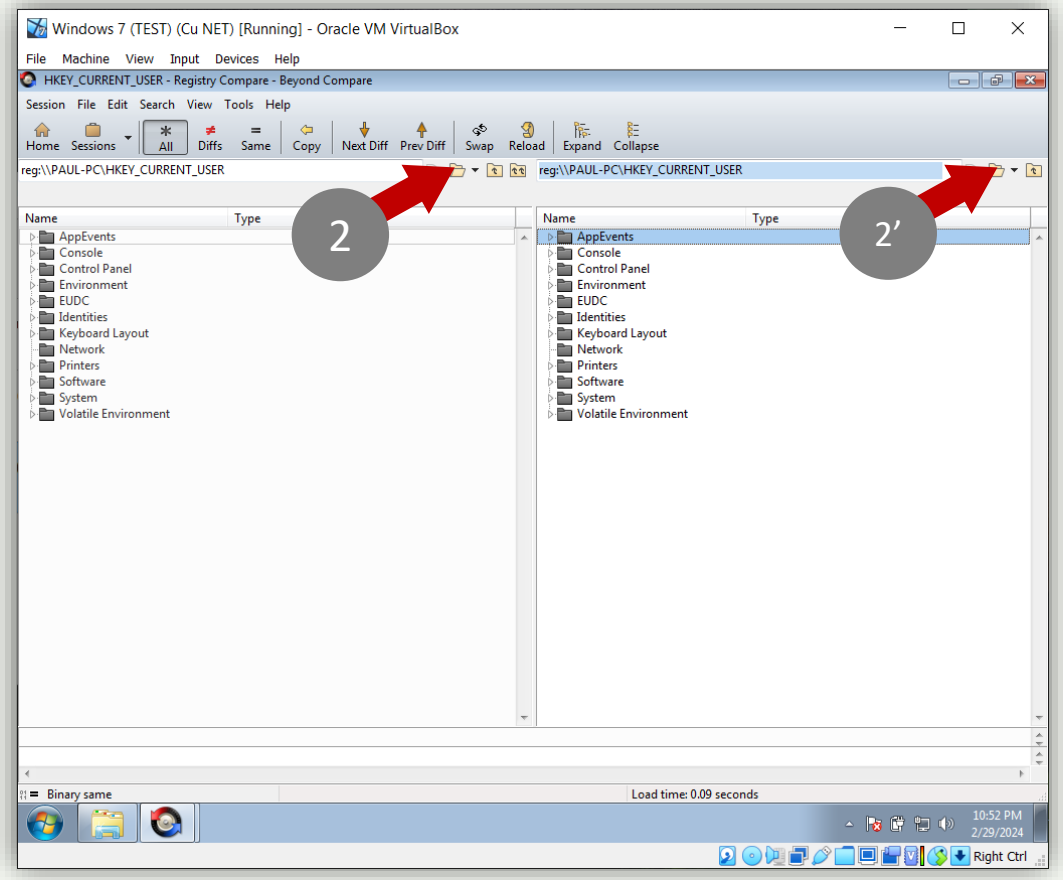




BCOMPARE

COMPARAREA CONȚINUTULUI A DOUĂ FIȘIERE

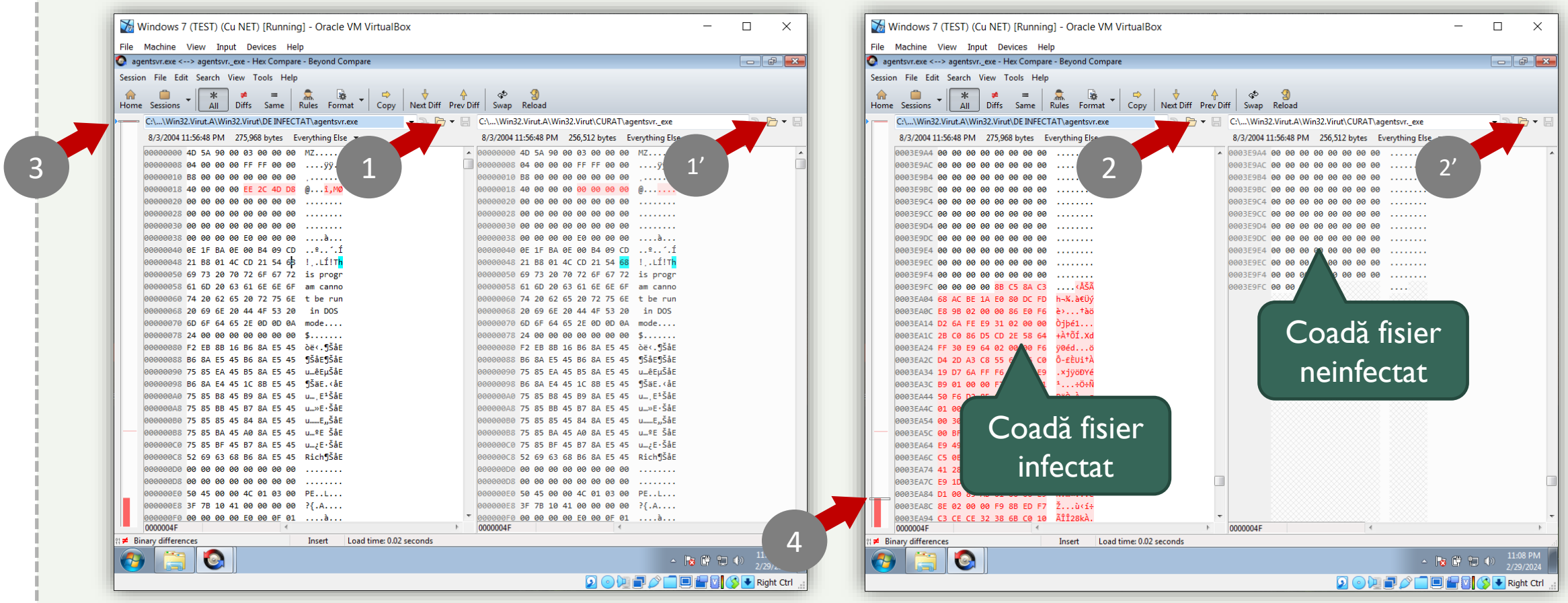




BCOMPARE

COMPARAREA CONȚINUTULUI A DOUĂ FIȘIERE

Observați diferența în cantitatea de informație dintre coada fișierului infectat și cel neinfectat.



COD MALIȚIOS: INSTRUMENTE DE ANALIZĂ

- PEstudio
- Bcompare
- **Cutter & IDA & x64dbg**

Analiză Statică

- Sysinternals
- Wireshark
- procDOT

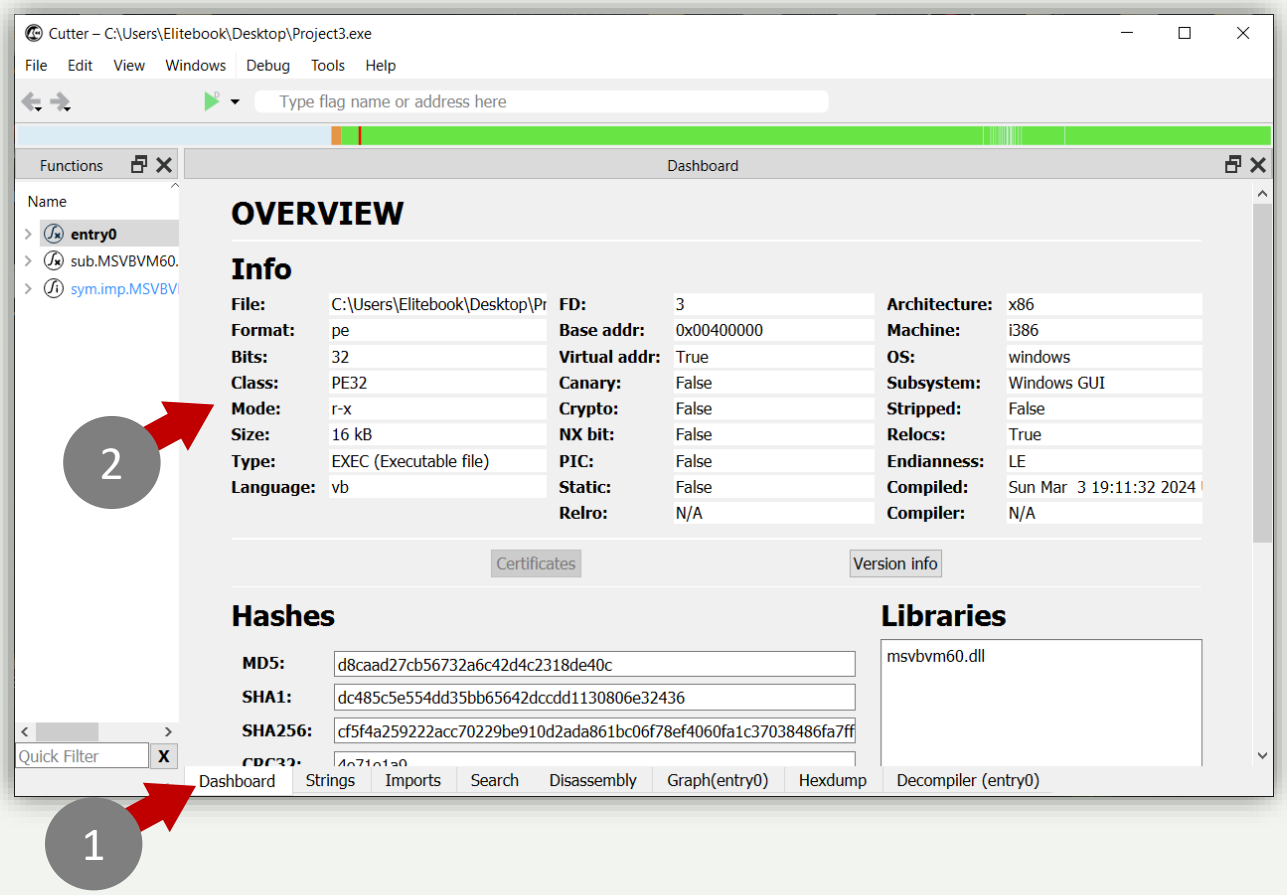
Analiză Dinamică

- Analiza malware in Cloud

Analiză Cloud

CUTTER
DASHBOARD

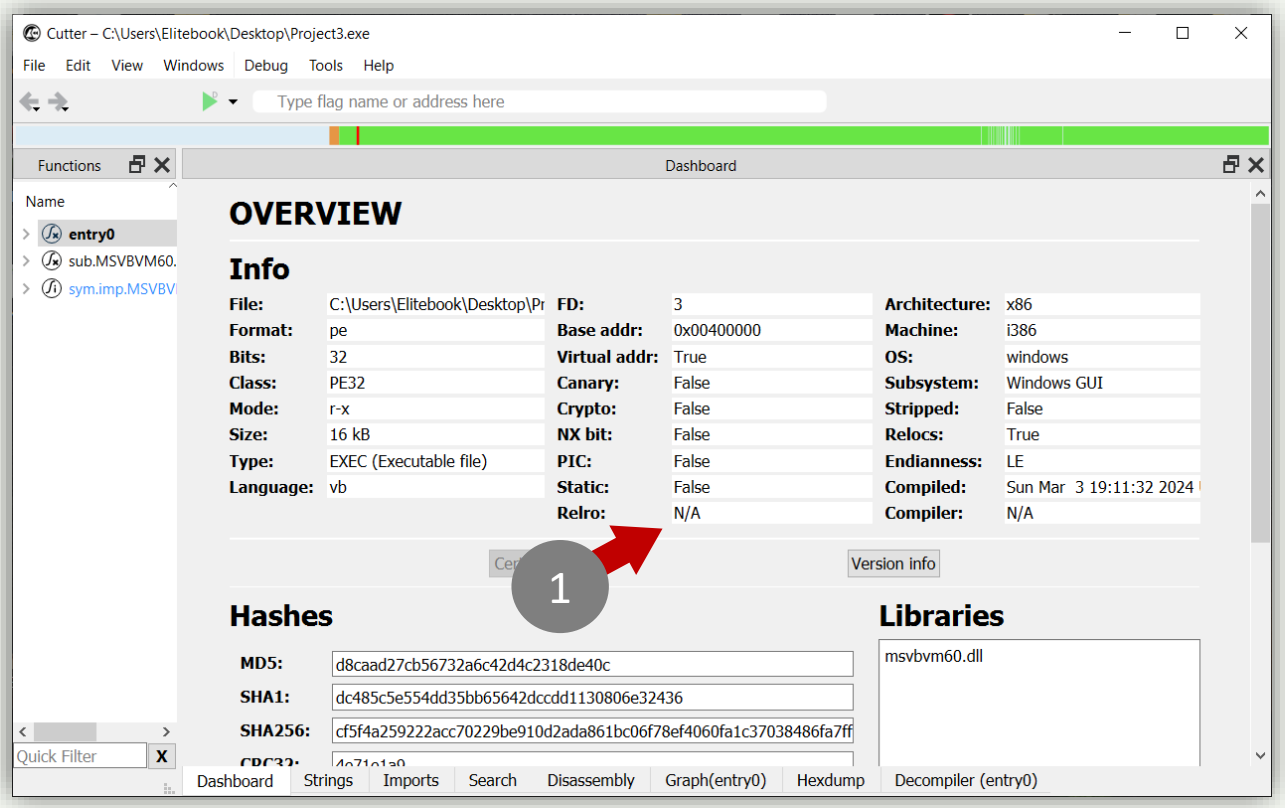
MD5: O valoare hash de 128 de biți, utilizată pentru a verifica integritatea fișierului.
SHA1: O valoare hash de 160 de biți, mai sigură decât MD5.
SHA256: O valoare hash de 256 de biți, parte din familia SHA-2, oferă o securitate și mai puternică.



- **File.** Calea către fișierul executabil analizat.
- **Format.** Indică formatul fișierului; în acest caz, "pe" pentru Portable Executable.
- **Bits.** Arată lățimea arhitecturii procesorului în biți; aici este "32", indicând un executabil pe 32 de biți.
- **Class.** Se referă la clasa specifică a formatului fișierului; "PE32" indică un Portable Executable pe 32 de biți.
- **Mode.** Permisiunile cu care fișierul este mappat; "r-x" înseamnă citire și executare.
- **Size.** Dimensiunea fișierului în kiloocteți.
- **Type.** Tipul fișierului, în acest caz "EXEC" (Executable file).
- **Language.** Limbajul de programare folosit, aici este "vb" pentru Visual Basic.

CUTTER
DASHBOARD

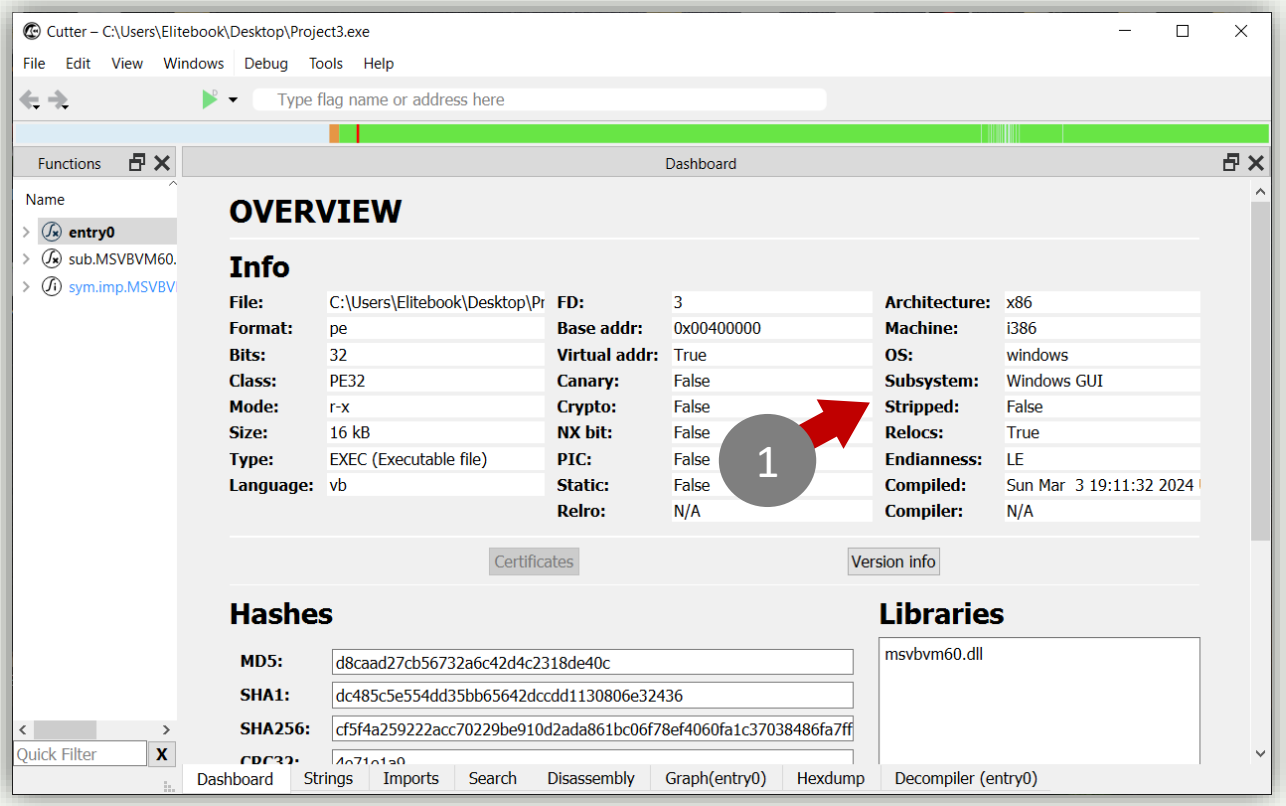
MD5: O valoare hash de 128 de biți, utilizată pentru a verifica integritatea fișierului.
SHA1: O valoare hash de 160 de biți, mai sigură decât MD5.
SHA256: O valoare hash de 256 de biți, parte din familia SHA-2, oferă o securitate și mai puternică.



- **FD.** Numărul descriptorului de fișier.
- **Base addr.** Adresa de bază unde executabilul este mapat în memorie.
- **Virtual addr.** Indică dacă fișierul are o adresă virtuală; aici este "True".
- **Canary.** O caracteristică de securitate pentru a preveni atacurile de tip buffer overflow; "False" indică faptul că nu este prezent.
- **Crypto.** Indică dacă sunt detectate funcții criptografice; "False" sugerează că nu sunt.
- **NX bit.** No-Execute bit, care dacă este "False" permite executarea codului în toate regiunile de memorie.
- **PIC.** Position Independent Code, care dacă este "False" înseamnă că codul nu este independent de poziție.
- **Static.** Indică dacă binarul este legat static; "False" înseamnă că nu este.
- **Relro.** Relocation Read-Only, o caracteristică de securitate care face anumite secțiuni ale binarului să fie doar de citire după ce relocațiile au fost procesate; "N/A" indică că nu este aplicabil sau nu este prezent.

CUTTER
DASHBOARD

MD5: O valoare hash de 128 de biți, utilizată pentru a verifica integritatea fișierului.
SHA1: O valoare hash de 160 de biți, mai sigură decât MD5.
SHA256: O valoare hash de 256 de biți, parte din familia SHA-2, oferă o securitate și mai puternică.

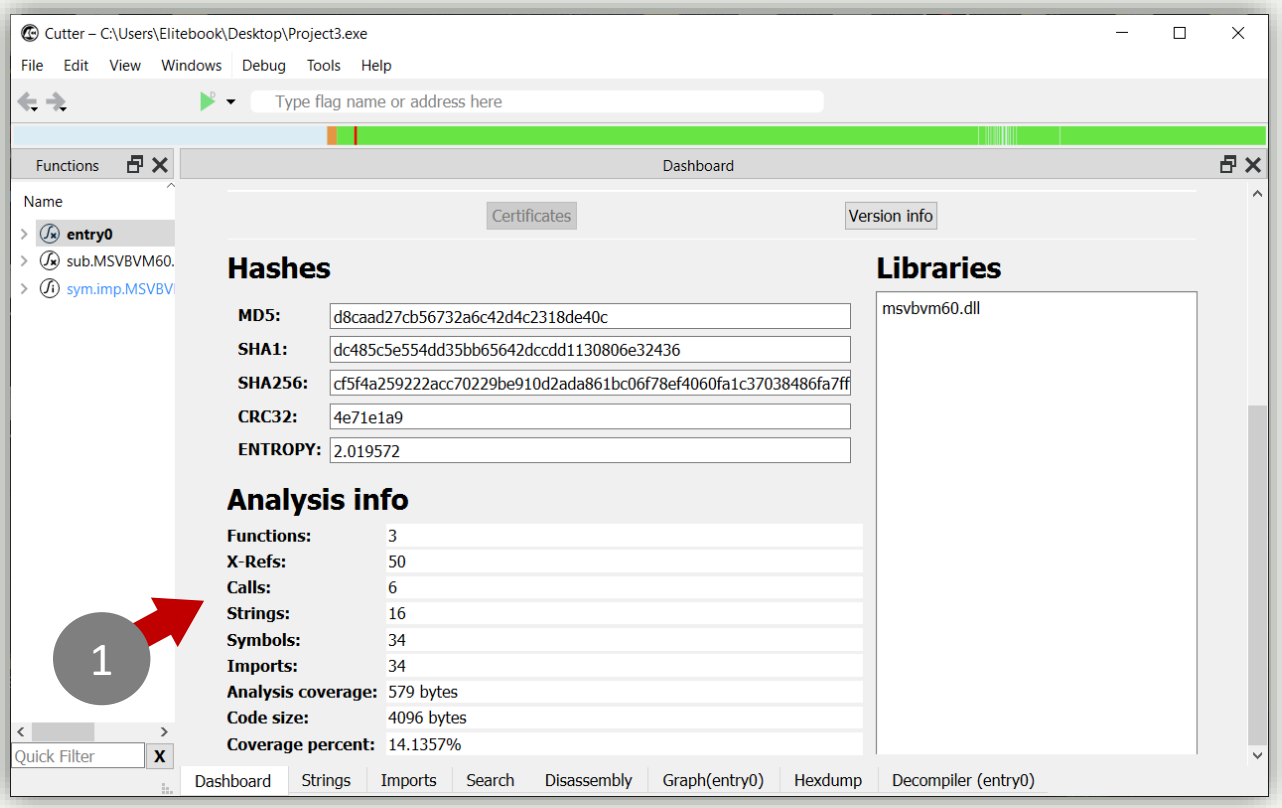


- **Architecture.** Indică arhitectura CPU, care este x86.
- **Machine.** Tipul specific de mașină sau procesor, i386, compatibil cu arhitectura x86 de 32 de biți.
- **OS.** Sistemul de operare, care este Windows.
- **Subsystem.** Subsistemul sub care executabilul funcționează, aici este "Windows GUI".
- **Stripped.** Indică dacă informațiile despre simboluri sunt eliminate din executabil; "False" înseamnă că nu sunt.
- **Relocs.** Relocații, indică dacă binarul poate fi relocat; "True" înseamnă că poate fi.
- **Endianness.** Ordinea octeților utilizată de sistem; "LE" înseamnă Little Endian.
- **Compiled.** Data și ora când executabilul a fost compilat.
- **Compiler.** Compilatorul folosit pentru a construi executabilul, în acest caz, "N/A" indică că nu este disponibil.

CUTTER

DASHBOARD - ANALYSIS INFO

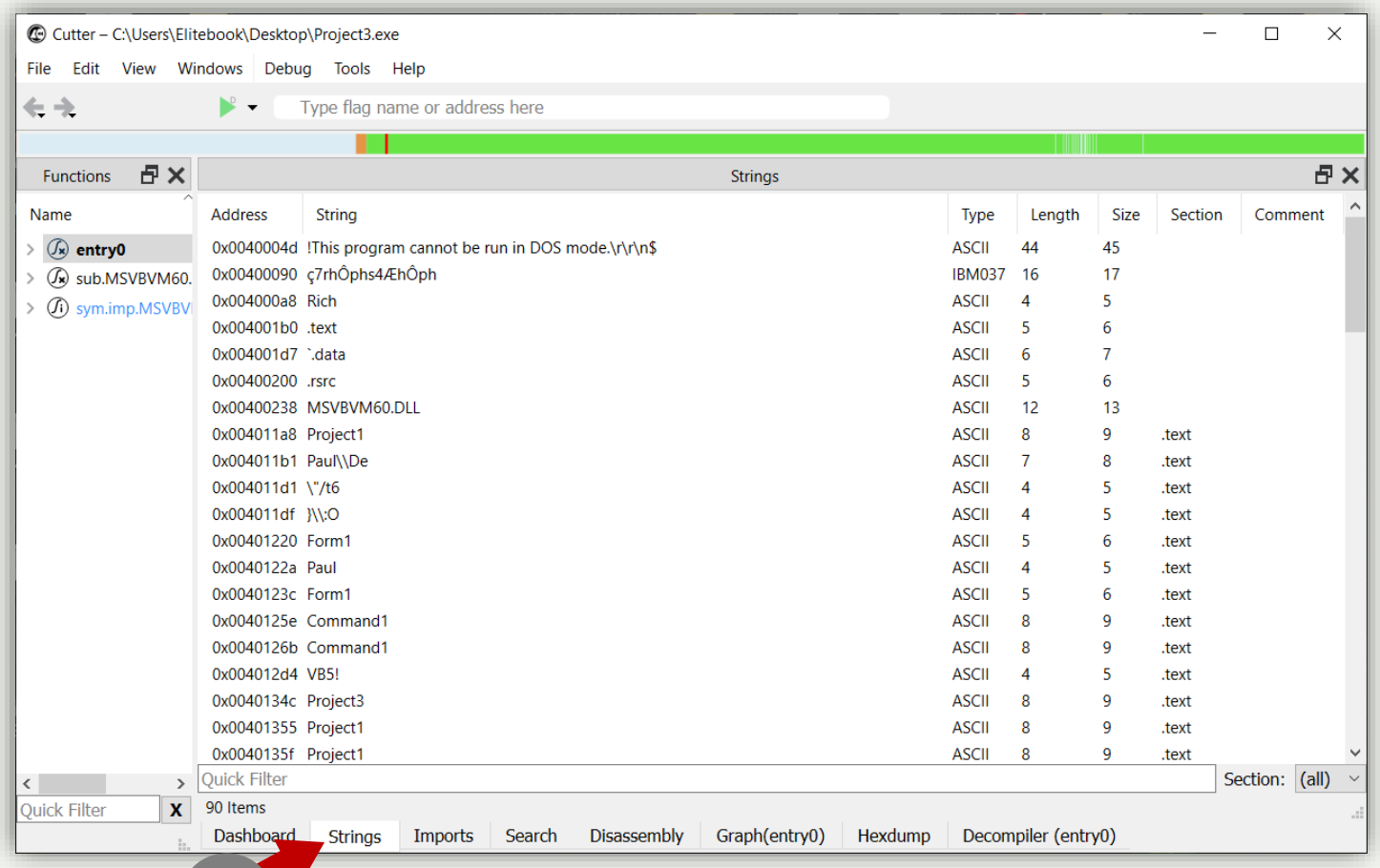
MD5: O valoare hash de 128 de biți, utilizată pentru a verifica integritatea fișierului.
SHA1: O valoare hash de 160 de biți, mai sigură decât MD5.
SHA256: O valoare hash de 256 de biți, parte din familia SHA-2, oferă o securitate și mai puternică.



- **Functions.** Numărul de funcții identificate în fișierul executabil.
- **X-Refs.** Numărul de referințe încrucișate care arată cum sunt legate între ele diversele secțiuni de cod sau date.
- **Calls.** Numărul de apeluri de funcții din cadrul codului.
- **Strings.** Numărul de șiruri de caractere detectate în fișierul executabil.
- **Symbols.** Numărul de simboluri detectate, care pot include funcții și variabile.
- **Imports.** Numărul de funcții importate, care sunt de obicei funcții din biblioteci externe folosite în cod.
- **Analysis coverage.** Cantitatea de cod analizată în octeți.
- **Code size.** Dimensiunea totală a secțiunii de cod din fișierul executabil în octeți.
- **Coverage percent.** Procentul de cod care a fost analizat în raport cu dimensiunea totală a codului.

CUTTER
STRINGS

Această funcție este utilă pentru a identifica mesaje, căi de fișiere, URL-uri și alte date posibil interesante pentru cei care fac inginerie inversă sau analize de securitate.



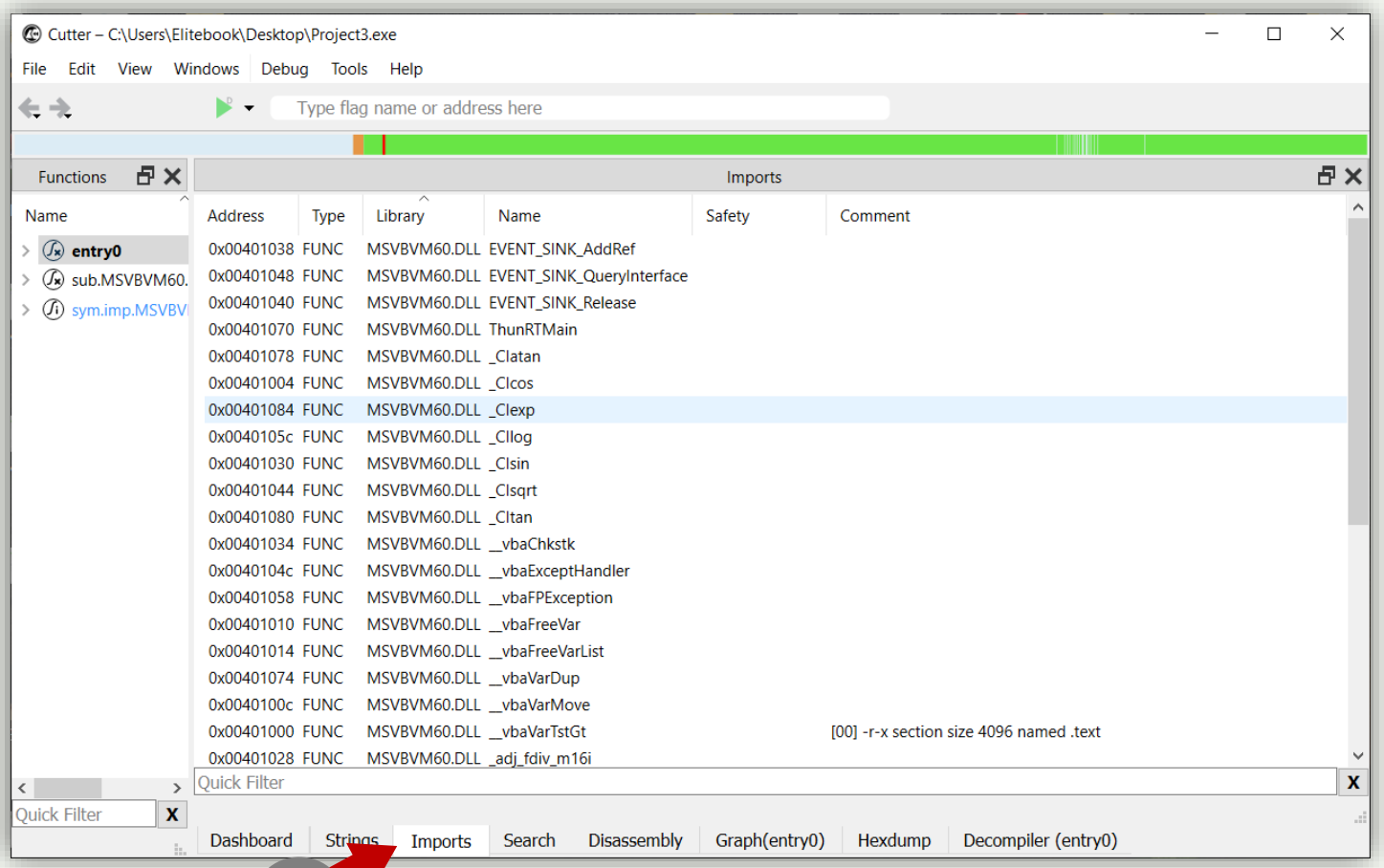
- **Name.** Numele sau eticheta asociată cu șirul în cadrul fișierului binar.
- **Address.** Adresa de memorie unde este localizat șirul.
- **String.** Textul efectiv al șirului.
- **Type.** Tipul de codificare a șirului, de exemplu ASCII sau un alt tip de codificare.
- **Length.** Lungimea șirului, fără a include caracterul terminator null.
- **Size.** Mărimea șirului în octeți, care poate include terminatorul null.
- **Section.** Secțiunea fișierului binar unde se află șirul. De exemplu, secțiunea ".text" conține cod executabil, în timp ce ".data" și ".rsrc" conțin de obicei date.

CUTTER

IMPORTURI (EXECUTABILE IMPURE & DEPENDENTE DE MV SAU PSEUDO-MV)

- Executabilele create cu Visual Basic 6 (VB6) nu rulează pe o mașină virtuală în modul în care înțelegem astăzi termenul de "mașină virtuală" pentru limbaje de programare cum ar fi Java (care utilizează Java Virtual Machine) sau C# (care rulează pe Common Language Runtime - CLR, parte a .NET Framework).

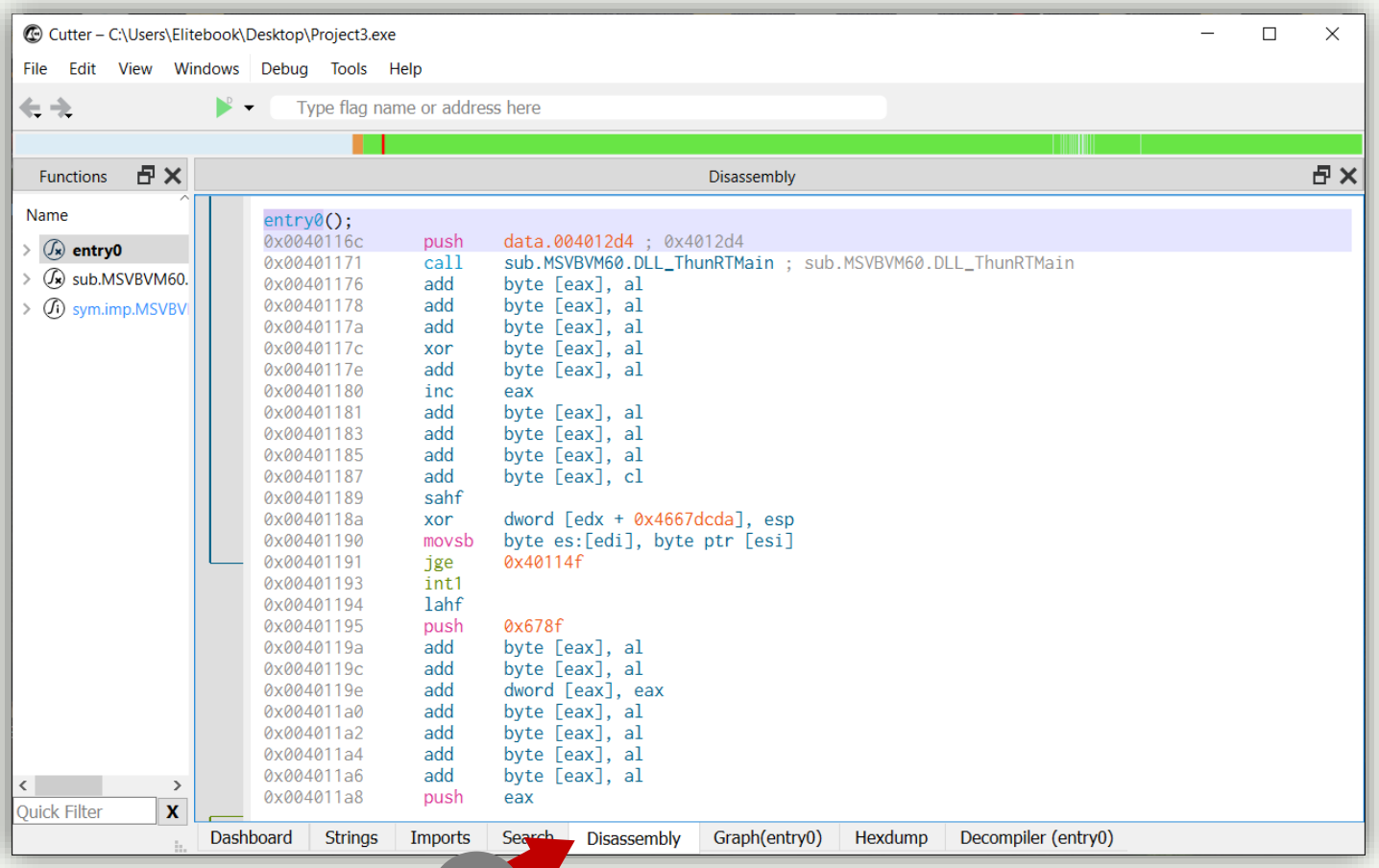
- În schimb, VB6 compilează codul sursă în cod de mașină nativ pentru procesorul sistemului de operare pe care rulează, de obicei Windows. Cu toate acestea, VB6 se bazează pe un set mare de componente runtime, cunoscute sub numele de VB6 Runtime, pentru a funcționa corect.
- Aceste biblioteci runtime trebuie să fie prezente pe sistemul pe care rulează executabilul pentru a furniza funcționalitățile necesare, cum ar fi gestionarea ferestrelor, controlul interfețelor utilizator și alte funcții de nivel înalt.
- Aceste componente runtime acționează ca un strat intermediar între aplicația VB6 și sistemul de operare.



CUTTER
DEZASAMBLARE

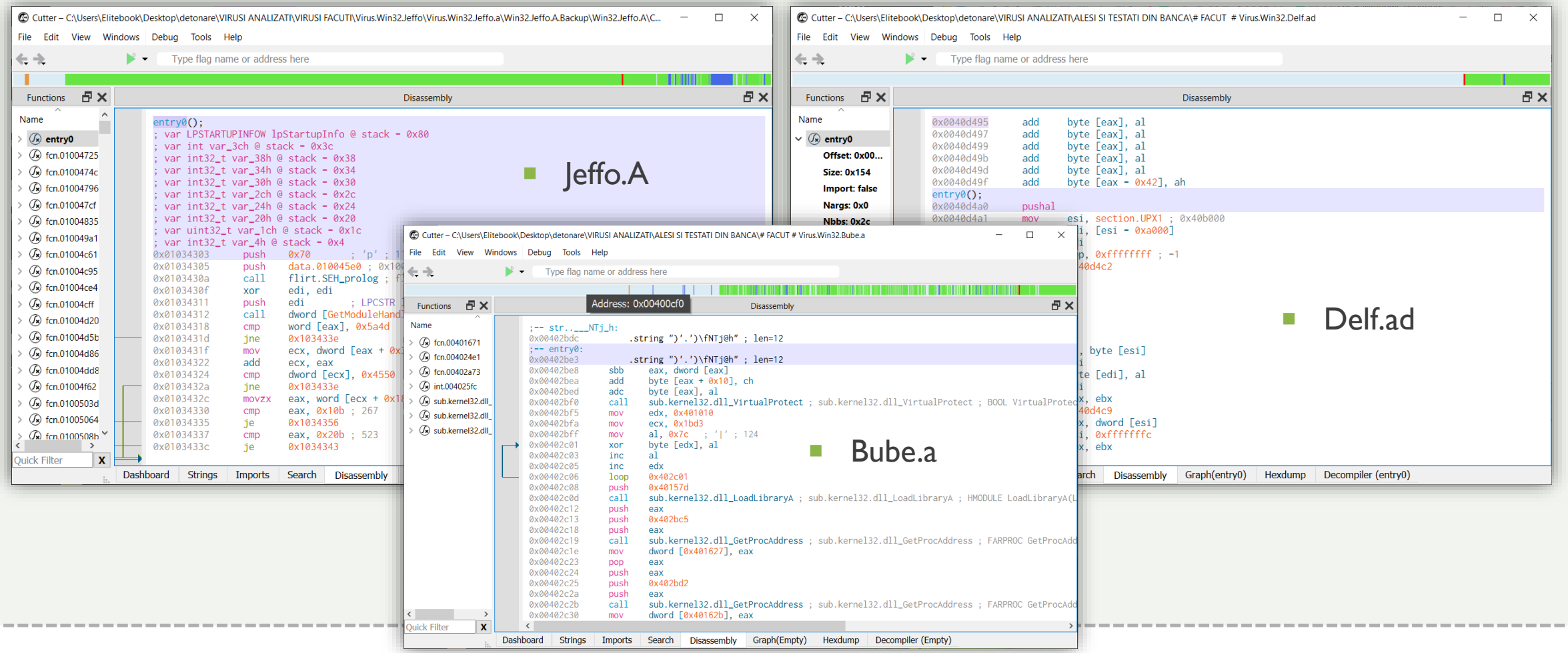
- Cum arată un punct de intrare (entry point) al unui executabil impur care își cere „mașina virtuală”?

- „adresele virtuale” sunt adresele la care codul și datele ar fi încărcate în memoria virtuală a unui proces dacă executabilul ar rula efectiv.



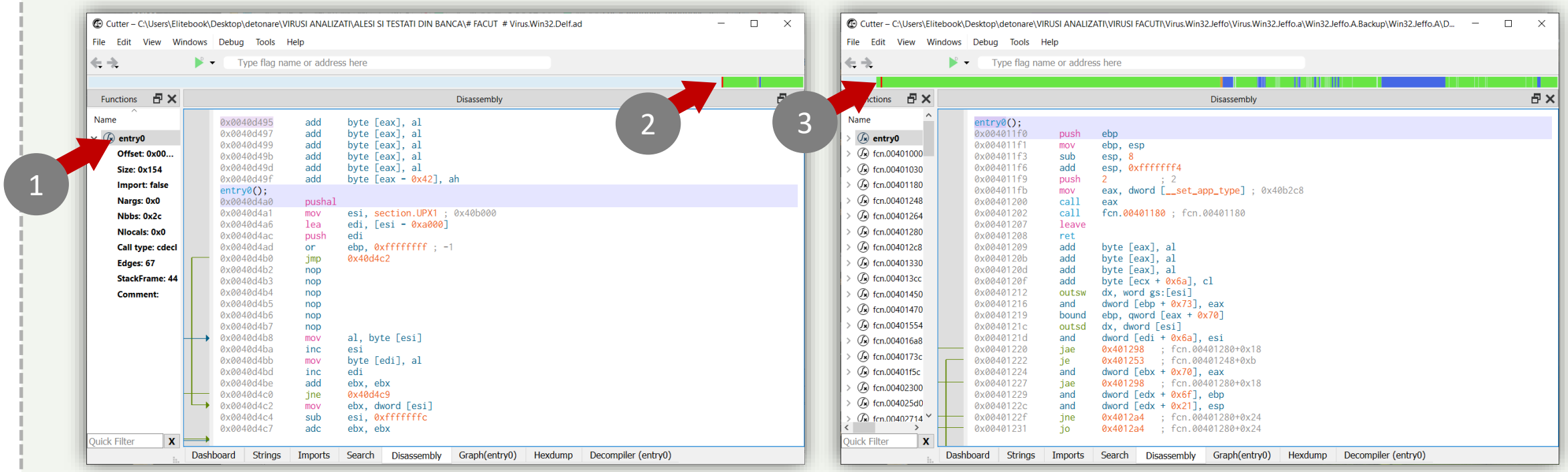
INFECTIE

VIZIBIL CU UN CUTTER - ANALIZA A TREI FIȘIERE INFECTATE !



CUTTER
DESCHIDERE ȘI DEZASAMBLARE

Exemple de fișiere executabile infectate. Observați poziția relativă a punctului de intrare în execuție:



Dezasamblare (fisier infectat & ne-infectat)

CUTTER

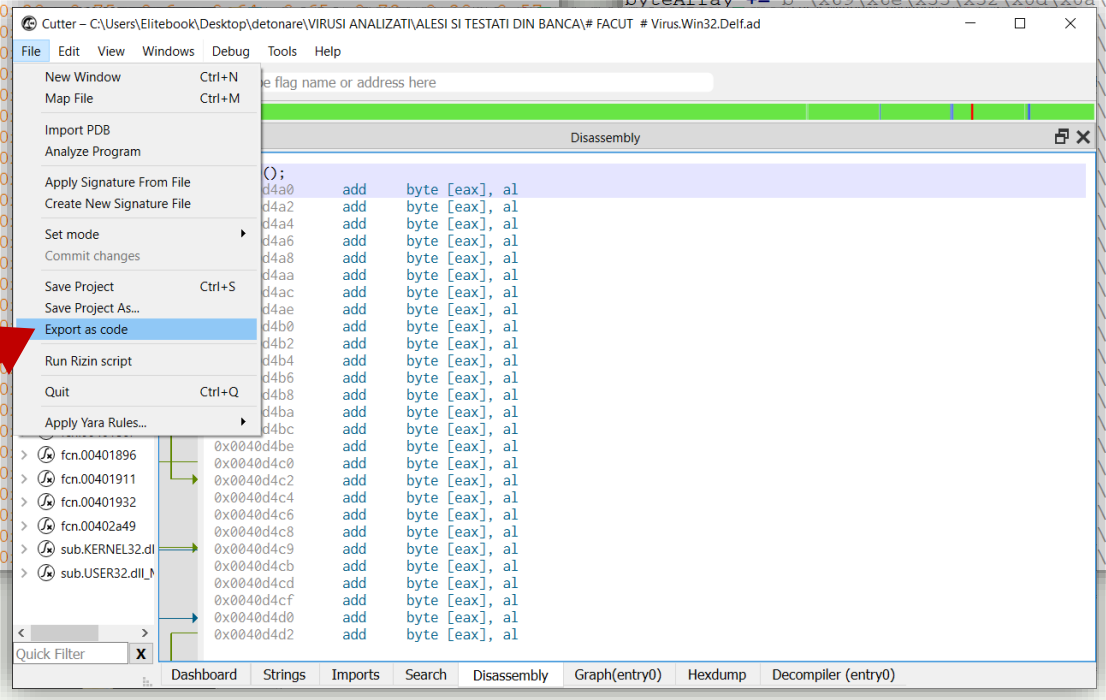
MENIURI UTILE – EXPORTAȚI FIȘIERUL ANALIZAT CA SHELLCODE

Shellcode in C++

```
#define ARRAY_SIZE 40960
const uint8_t array[ARRAY_SIZE] = {
  0x4d, 0x5a, 0x50, 0x00, 0x02, 0x00, 0x00, 0x00, 0x04, 0x00, 0x0f, 0x00, 0xff, 0xff, 0x00, 0x00,
  0xb8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x1a, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00,
  0xba, 0x10, 0x00, 0x0e, 0x1f, 0xb4, 0x09, 0xcd, 0x21, 0xb8, 0x01, 0x4c, 0xcd, 0x21, 0x90, 0x90,
  0x54, 0x68, 0x69, 0x73, 0x20, 0x70, 0x72, 0x6f, 0x67, 0x72, 0x61, 0x6d, 0x20, 0x6d, 0x75, 0x73,
  0x74, 0x20, 0x62, 0x65, 0x20, 0x72, 0x75, 0x6e,
  0x69, 0x6e, 0x33, 0x32, 0x0d, 0x0a, 0x24, 0x37,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x50, 0x45, 0x00, 0x00, 0x4c, 0x01, 0x04, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x8e, 0x81,
  0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xb0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xe0, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00,
  0x00, 0x00, 0x10, 0x00, 0x00, 0x20, 0x00, 0x00
  0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00,
  0x00, 0xc0, 0x00, 0x00, 0x90, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xd0, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0xa0, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00,
```

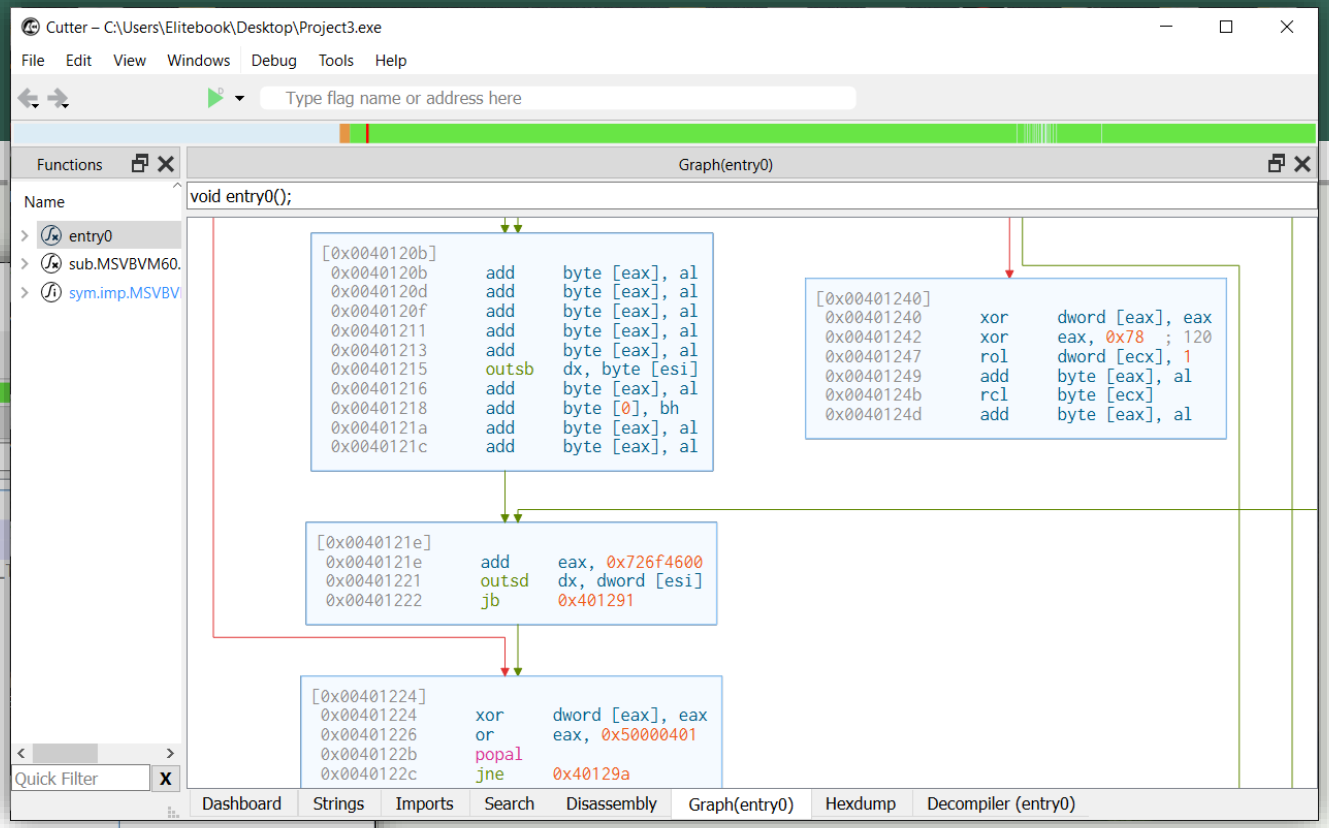
Shellcode in Python

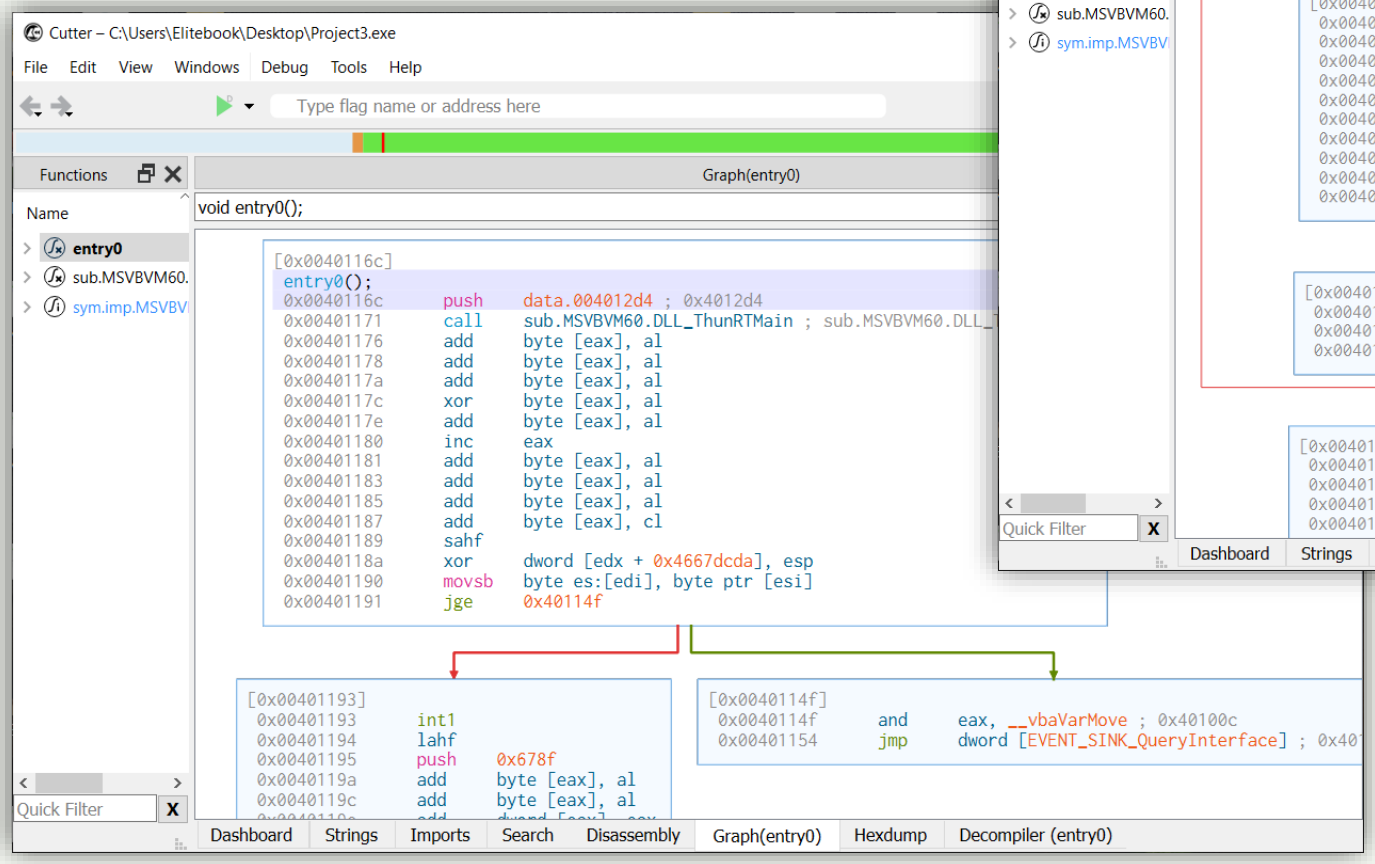
```
byteArray = b'\x4d\x5a\x50\x00\x02\x00\x00\x00\x04\x00\x0f\x00\xff\xff\x00\x00'
byteArray += b'\xb8\x00\x00\x00\x00\x00\x00\x00\x40\x00\x1a\x00\x00\x00\x00\x00'
byteArray += b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
byteArray += b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00'
byteArray += b'\xba\x10\x00\x0e\x1f\xb4\x09\xcd\x21\xb8\x01\x4c\xcd\x21\x90\x90'
byteArray += b'\x54\x68\x69\x73\x20\x70\x72\x6f\x67\x72\x61\x6d\x20\x6d\x75\x73'
byteArray += b'\x74\x20\x62\x65\x20\x72\x75\x6e\x20\x75\x6e\x64\x65\x72\x20\x57'
byteArray += b'\x69\x6e\x33\x32\x0d\x0a\x24\x37\x00\x00\x00\x00\x00\x00\x00'
```



Aplicații metamorfice

CUTTER
GRAPH (VISUALIZARE)

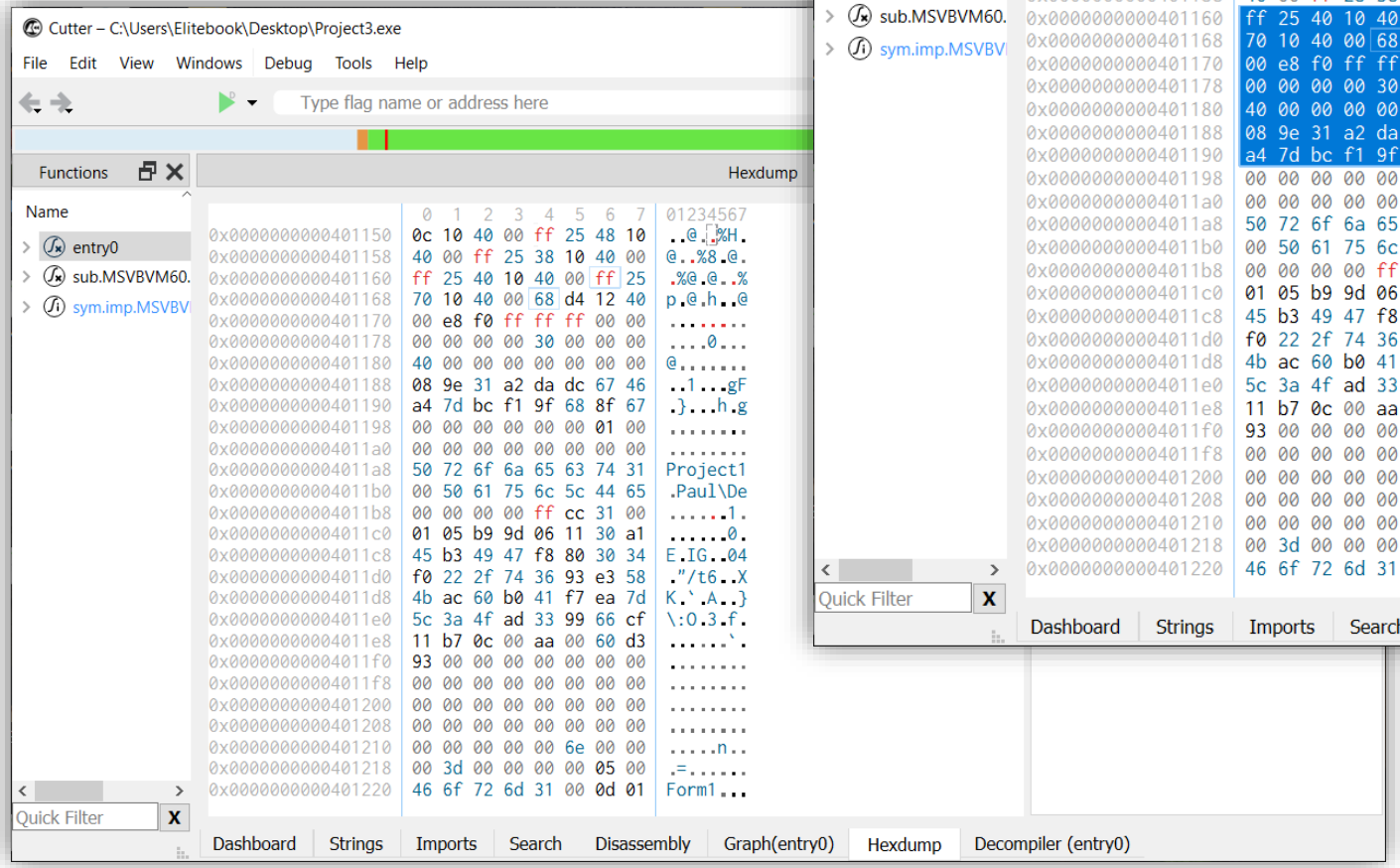


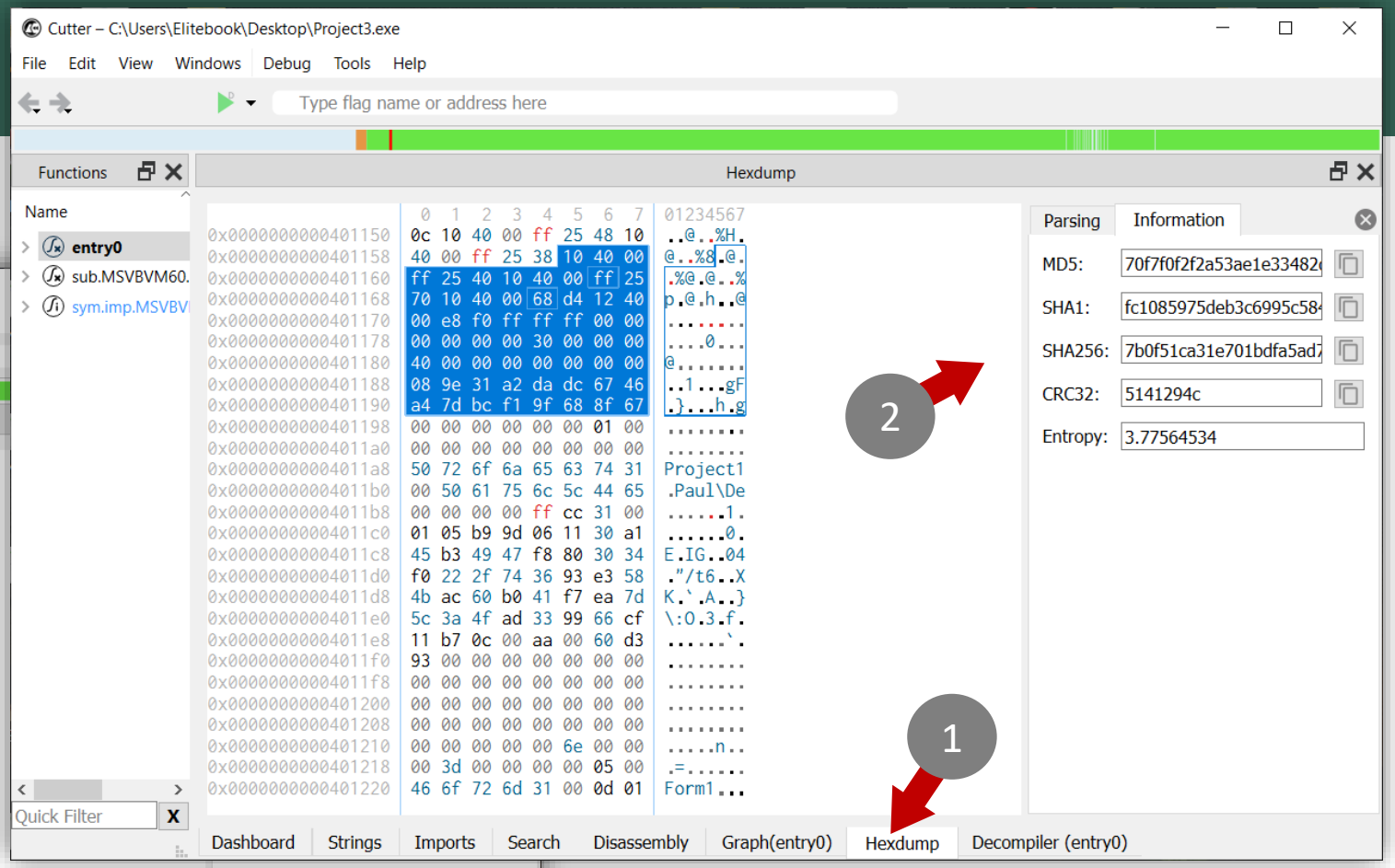


- Fiecare panou poate fi văzut ca un mic modul.

CUTTER
HEXDUMP

- Vizualizare hex

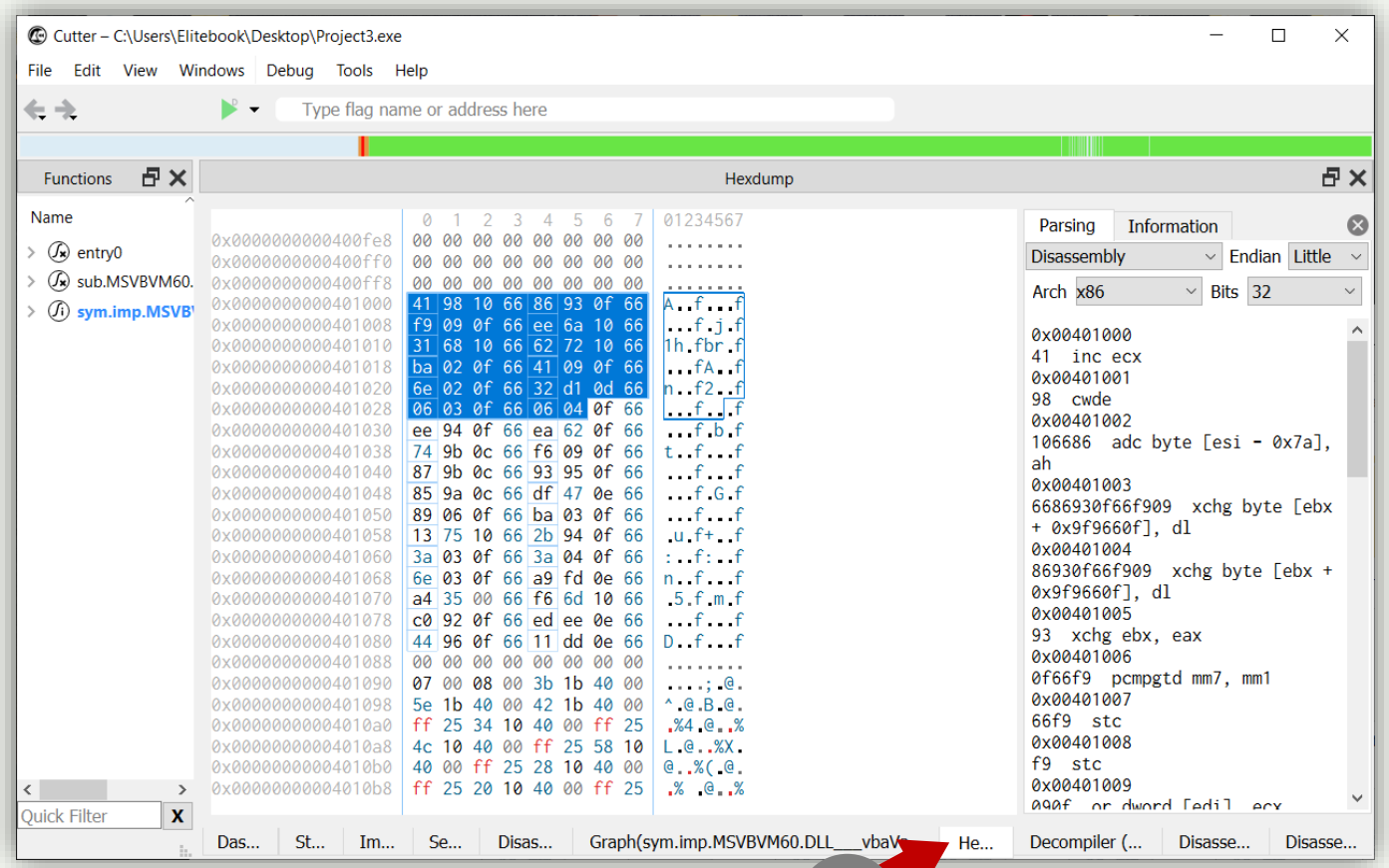




- În timpul selecției se calculează o serie de indicatori (semnaturi hash)!

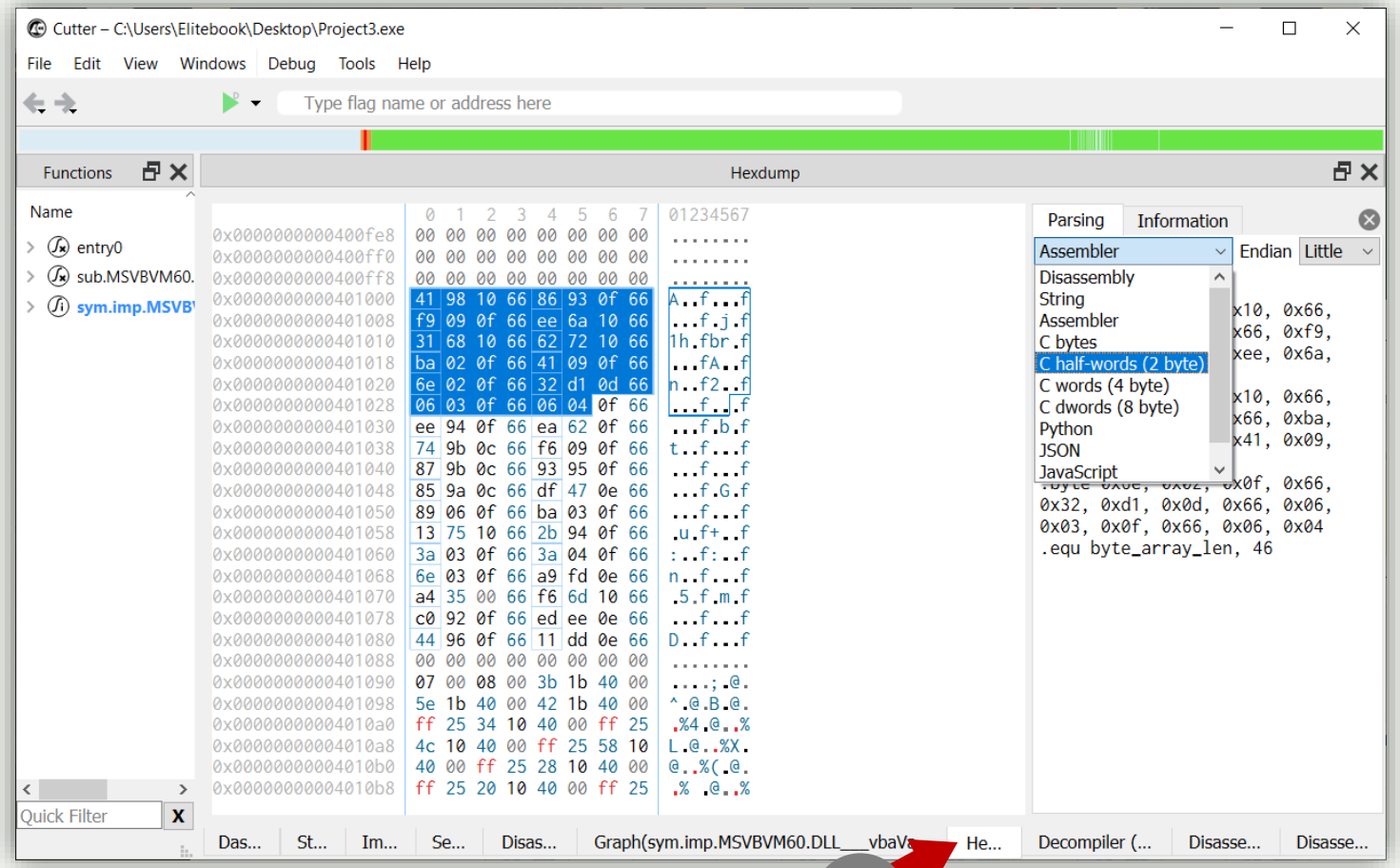
CUTTER

HEXDUMP – DEZASAMBLARE LA SELECTIE



- Oferă posibilitatea de demontare la selectarea șirului hexazecimal cu mouse-ul.

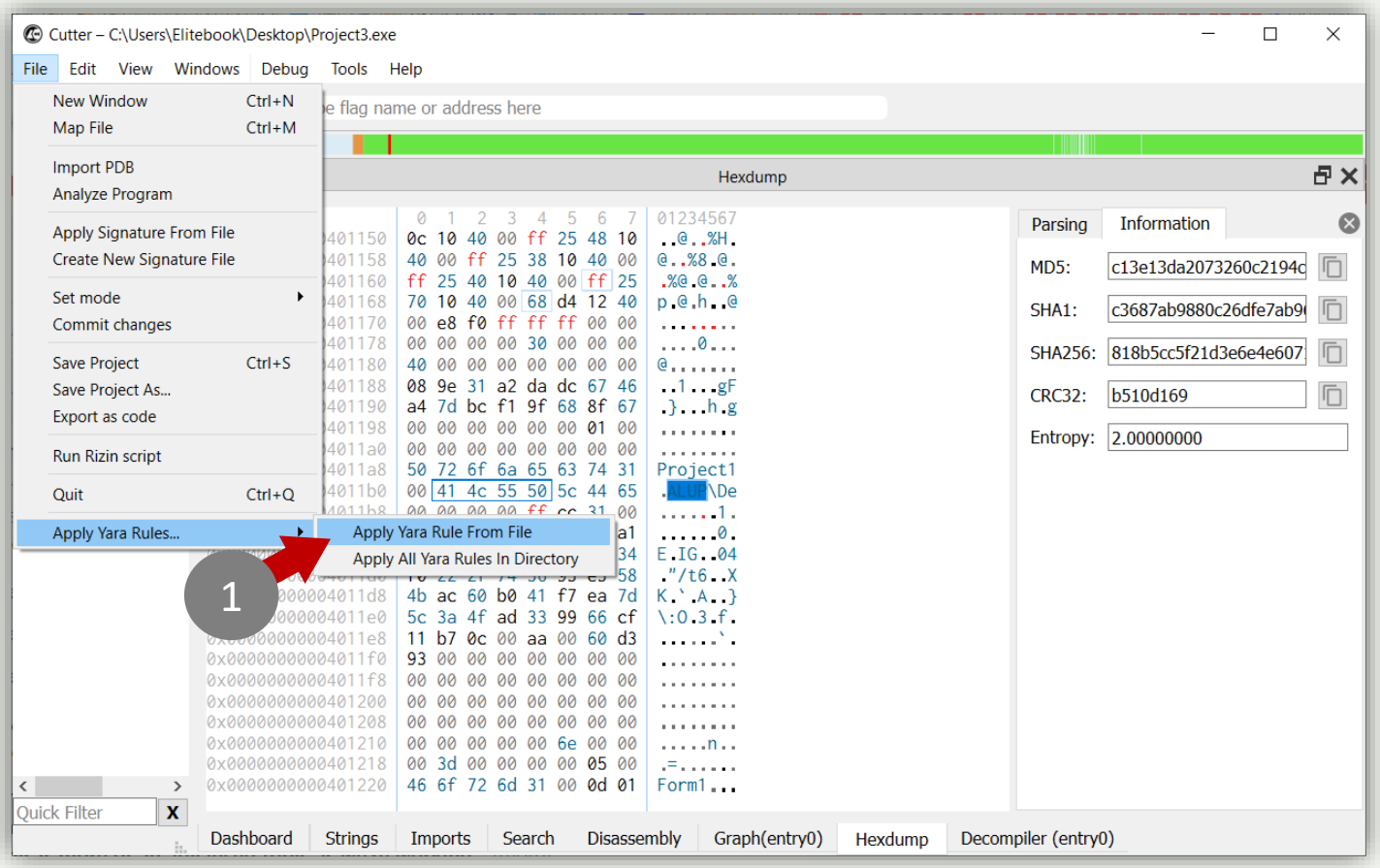
CUTTER
HEXDUMP – EXPORTARE IN COD SURSA



- Poate exporta selecția în codul sursă pentru diferite limbaje de programare.

CUTTER
EXPORT YARA

- YARA este un instrument de securitate a sistemelor informatice, folosit pentru identificarea și clasificarea malware-ului.
- Este bazat pe reguli care utilizează pattern-uri de șiruri de caractere și un set de expresii logice pentru a detecta anumite caracteristici sau comportamente ale fișierelor suspecte.
- YARA este folosit adesea de cercetătorii în securitatea cibernetică pentru a crea semnături care ajută la identificarea rapidă a amenințărilor specifice, a variantelor de malware sau a documentelor malițioase. Este extrem de versatil și poate fi integrat în diverse fluxuri de lucru de securitate și sisteme de analiză automată.



CUTTER
DECOMPILARE

Cutter – C:\Users\Elitebook\Desktop\Project3.exe

File Edit View Windows Debug Tools Help

Type flag name or address here

Functions

Name

- entry0
- sub.MSVBVM60.
- sym.imp.MSVBV

Decompiler (entry0)

```
void entry0(void)
{
    uint32_t *puVar1;
    uint32_t uVar2;
    code *pcVar3;
    uint8_t uVar4;
    char cVar5;
    uint8_t *puVar6;
    char extraout_CL;
    undefined *unaff_ESI;
    undefined *unaff_EDI;
    undefined8 uVar7;
    code *pcStack_4;

    pcStack_4 = data.004012d4;
    uVar7 = sub.MSVBVM60.DLL_ThunRTMain();
    puVar6 = (uint8_t *)uVar7;
    uVar4 = (uint8_t)uVar7;
    *puVar6 = *puVar6 + uVar4;
    *puVar6 = *puVar6 + uVar4;
    *puVar6 = *puVar6 + uVar4;
    *puVar6 = *puVar6 ^ uVar4;
    *puVar6 = *puVar6 + uVar4;
    puVar6 = puVar6 + 1;
    cVar5 = (char)puVar6;
    *puVar6 = *puVar6 + cVar5;
```

Quick Filter

Decompiler: Ghidra

Dashboard | Strings | Imports | Search | Disassembly | Graph(entry0) | Hexdump | Decompiler (entry0)

Cutter – C:\Users\Elitebook\Desktop\Project3.exe

File Edit View Windows Debug Tools Help

Type flag name or address here

Functions

Name

- entry0
- sub.MSVBVM60.
- sym.imp.MSVBV

Decompiler

```
/* jsdec pseudo code output */
/* C:\Users\Elitebook\Desktop\Project3.exe @ 0x401198 */
#include <stdint.h>

uint32_t rotate_left32 (uint32_t value, uint32_t count) {
    const uint32_t mask = (CHAR_BIT * sizeof (value)) - 1
    count &= mask;
    return (value << count) | (value >> (-count & mask));
}
uint8_t rotate_left8 (uint8_t value, uint32_t count) {
    const uint8_t mask = (CHAR_BIT * sizeof (value)) - 1;
    count &= mask;
    return (value << count) | (value >> (-count & mask));
}

int32_t entry0 (void) {
    do {
        eax &= __vbaVarMove;
        al = MSVBVM60_DLL_ThunRTMain (data.004012d4);
        *(eax) += al;
        *(eax) += al;
        *(eax) += al;
        *(eax) ^= al;
        *(eax) += al;
        eax++;
        *(eax) += al;
        *(eax) += al;
```

Quick Filter

Decompiler: jsdec

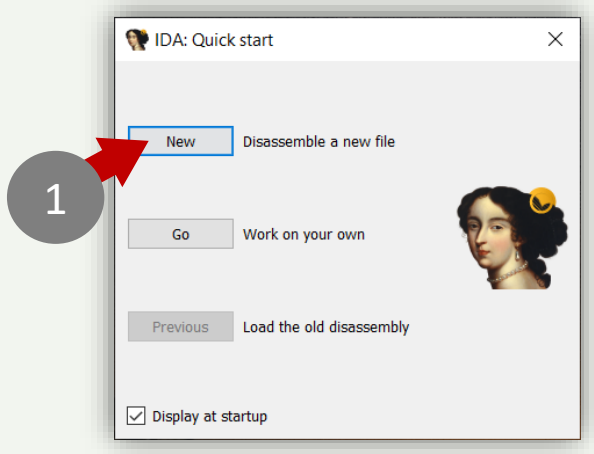
Dashboard | Strings | Imports | Search | Disassembly | Graph(entry0) | Hexdump | Decompiler (entry0)

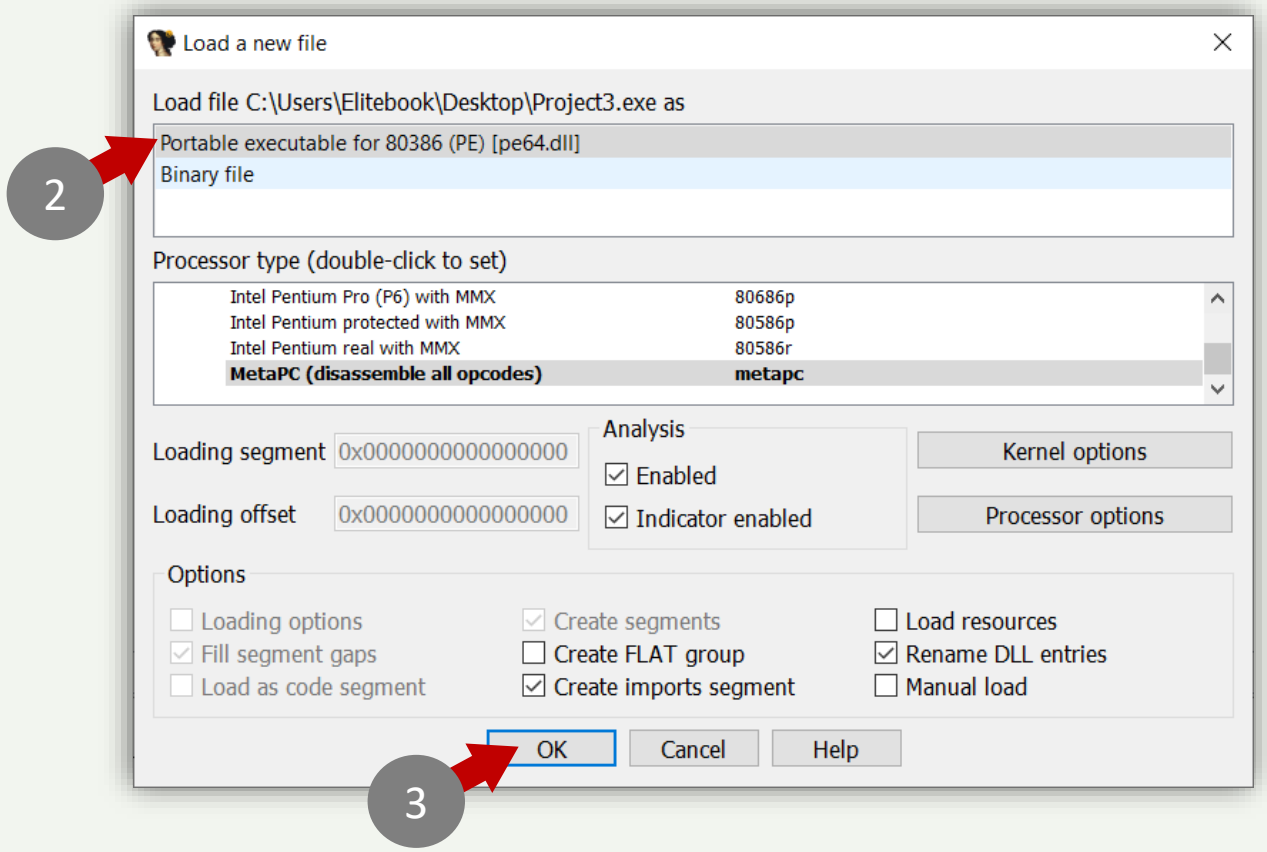
1

- Ghidra
- Jsdec

IDA FREE

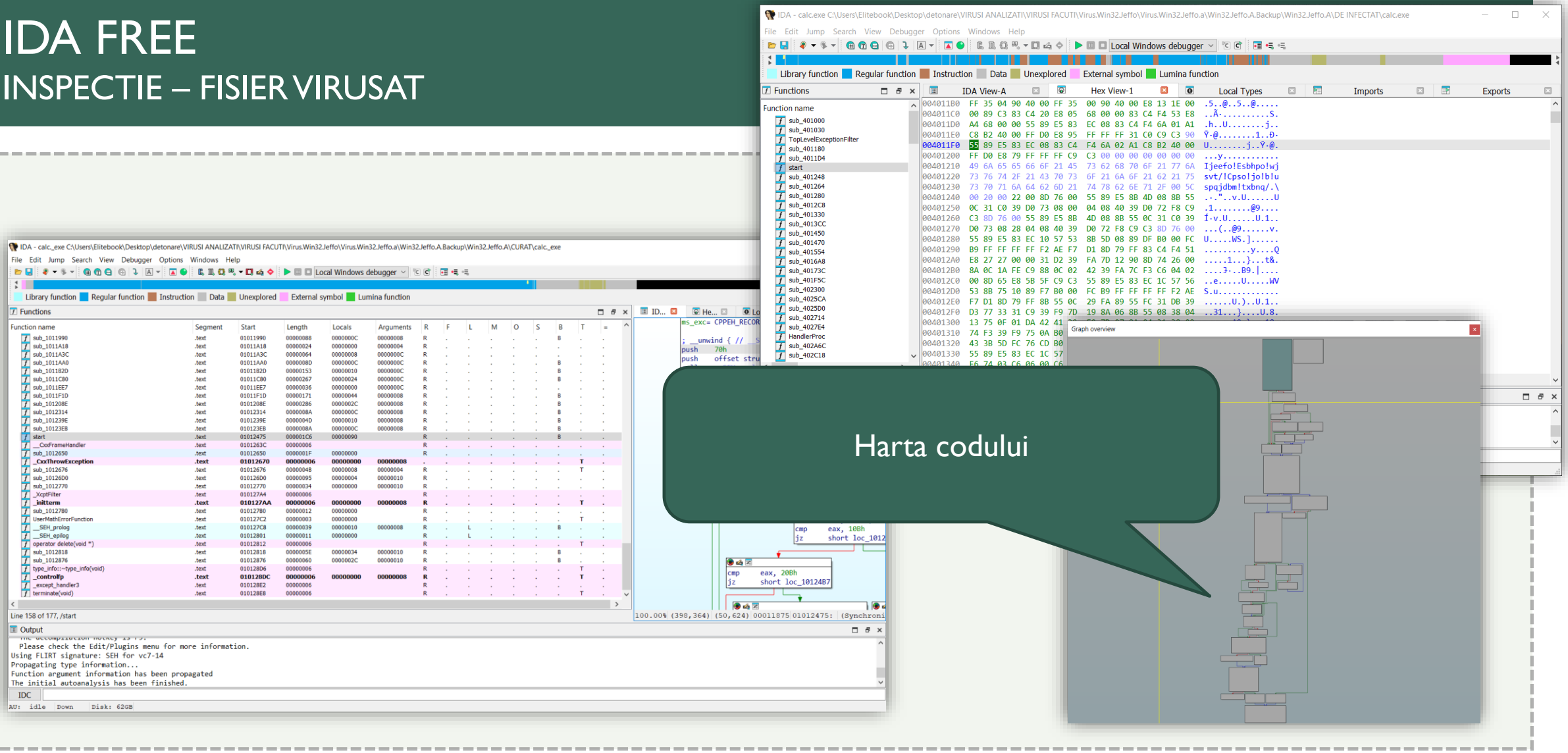
DESCHIDERE SI DEZASAMBLARE





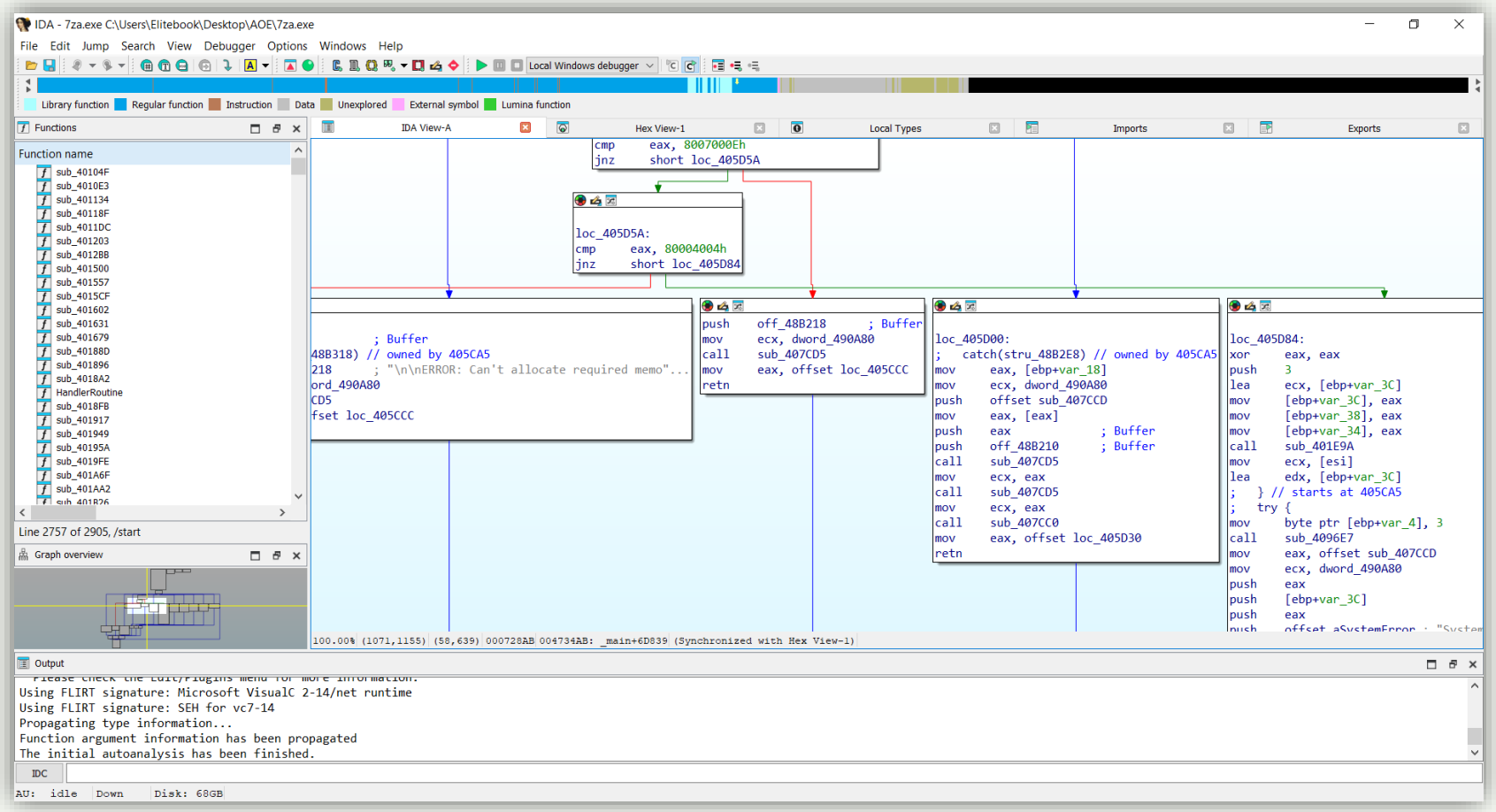
IDA FREE

INSPECTIE – FISIER VIRUSAT



IDA FREE

DEZASAMBLARE – EXECUTABILE PURE

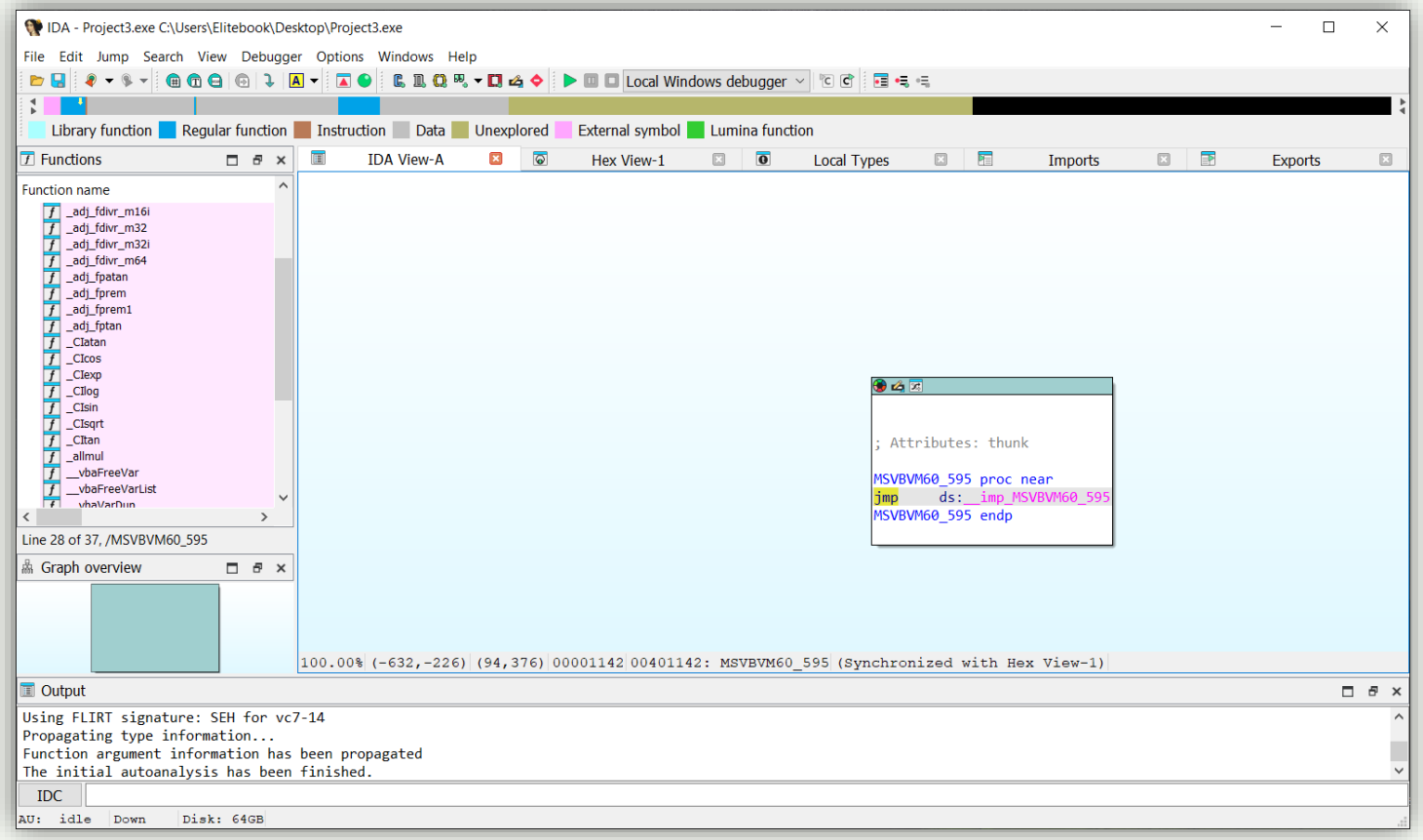


- Relativ independent cu utilizarea importurilor.
- Permite IDA să mapeze codul.

IDA FREE

DEZASAMBLARE – EXECUTABILE INPURE (MSVBVM60.DLL)

- Executabile dependente de mașini virtuale sau de un sistem intermediar între OS și executabil.
- IDA nu știe să mapeze codul fără suportul intermediar.



PETICIRI DE EXECUTABILE

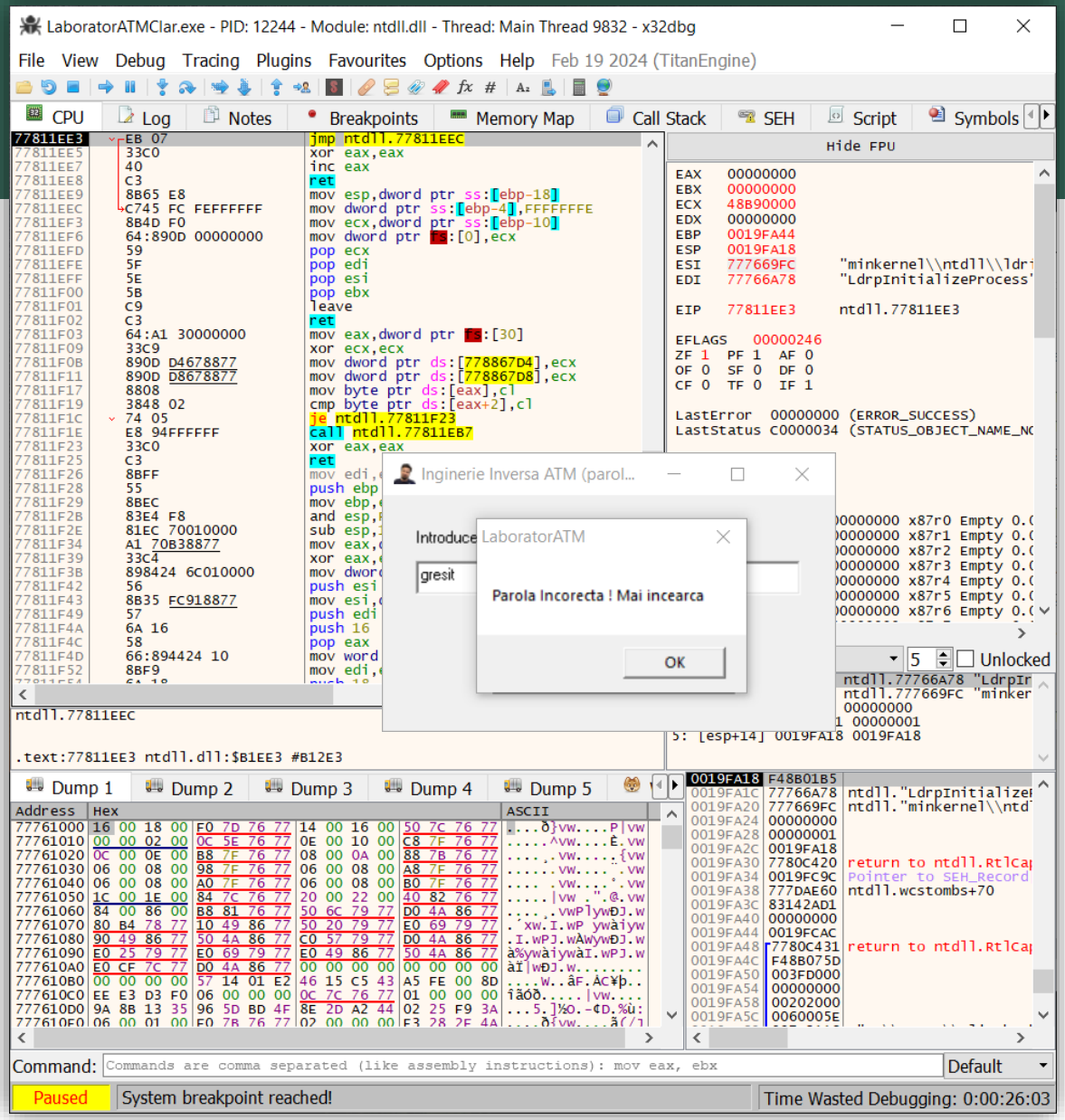
IDA & CUTTER & X32DBG

- CPU - Panoul CPU este probabil cel mai important și afișează instrucțiunile mașină ale programului (codul assembly), permițând utilizatorului să navigheze prin ele, să pună puncte de oprire (breakpoints) și să urmărească execuția pas cu pas. Acest panou include de obicei și vizualizarea registrelor procesorului, arătând starea curentă a acestora.

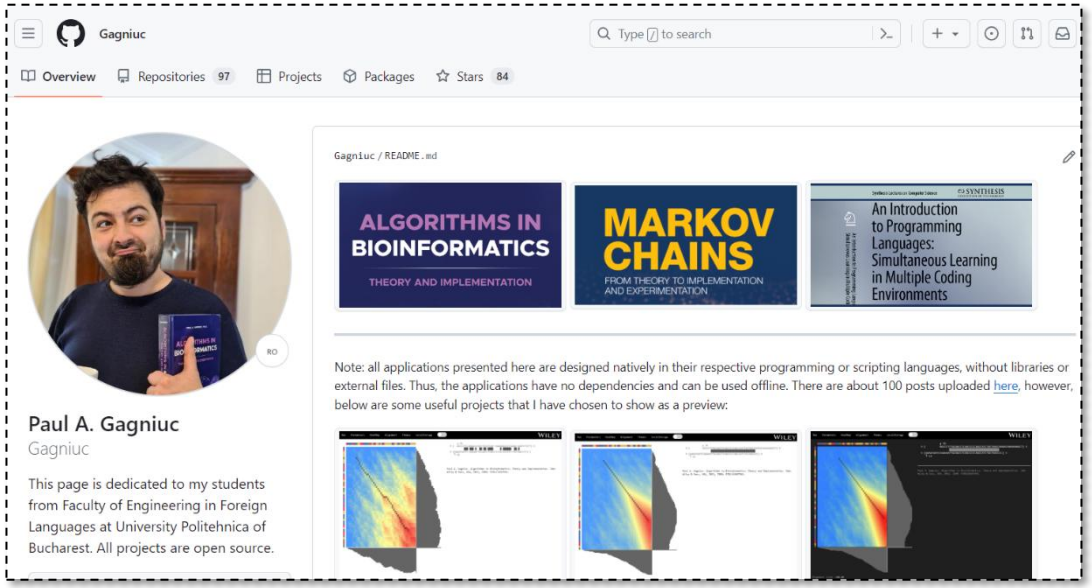
- Registru - Afișează starea curentă a registrelor CPU. Acestea includ registre generale (cum ar fi EAX, EBX pentru arhitectura x86), registre de index și de bază, registre de segment, pointerul de instrucțiuni (EIP), și flags-urile. Modificările în registre sunt adesea evidențiate pentru a facilita urmărirea modificărilor de stare.

- Dump - Panoul de dump arată conținutul memoriei la o anumită adresă. Este util pentru examinarea și modificarea datelor în memorie. Utilizatorii pot căuta anumite valori sau stringuri în acest panou.

- Stack - Afișează conținutul stack-ului curent al programului, oferind o vedere asupra apelurilor de funcții și a datelor locale. Acest panou este esențial pentru a urmări cum sunt gestionate apelurile de funcții și pentru a identifica posibile probleme legate de stack.



BIBLIOGRAFIE / RESURSE

- Paul A. Gagniuc. *Antivirus Engines: From Methods to Innovations, Design, and Applications*. Cambridge, MA: Elsevier Syngress, 2024. pp. 1-656.
- Paul A. Gagniuc. *An Introduction to Programming Languages: Simultaneous Learning in Multiple Coding Environments. Synthesis Lectures on Computer Science*. Springer International Publishing, 2023, pp. 1-280.
- Paul A. Gagniuc. *Coding Examples from Simple to Complex - Applications in MATLAB*, Springer, 2024, pp. 1-255.
- Paul A. Gagniuc. *Coding Examples from Simple to Complex - Applications in Python*, Springer, 2024, pp. 1-245.
- Paul A. Gagniuc. *Coding Examples from Simple to Complex - Applications in Javascript*, Springer, 2024, pp. 1-240.
- Paul A. Gagniuc. *Markov chains: from theory to implementation and experimentation*. Hoboken, NJ, John Wiley & Sons, USA, 2017, ISBN: 978-1-119-38755-8.

https://github.com/gagniuc