Report Compito 07/07/2023

Compito:

Compito di : Mendolia Valerio

Analisi statica e dinamica: Un approccio pratico -

Obiettivo

Con riferimento al file Malware_U3_W2_L5 presente all'interno della cartella

«Esercizio_Pratico_U3_W2_L5» sul desktop della macchina virtuale dedicata per l'analisi dei malware, rispondere ai seguenti quesiti:

1. Quali librerie vengono importate dal file eseguibile?

2. Quali sono le sezioni di cui si compone il file eseguibile del malware?

Con riferimento alla figura in slide 3, risponde ai seguenti quesiti:

1.1. Identificare i costrutti noti (creazione dello stack, eventuali cicli, costrutti)

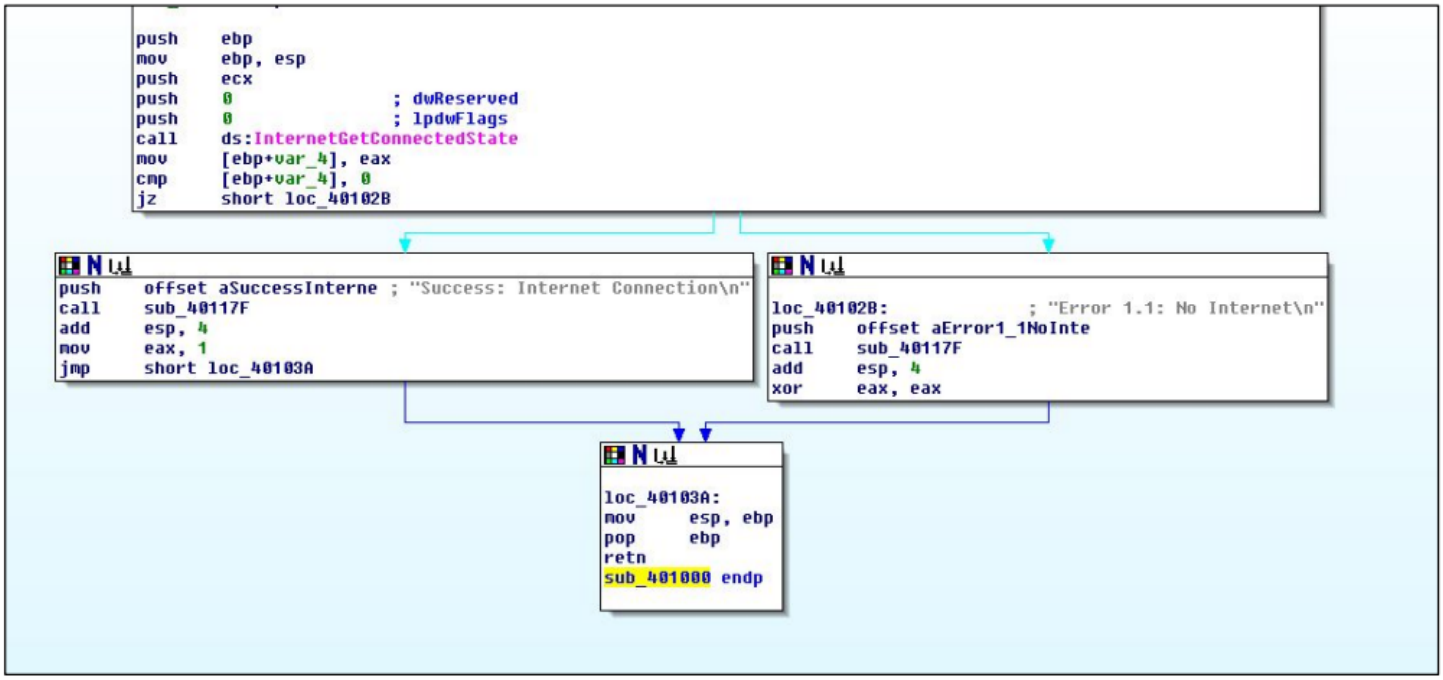
1.3. Ipotizzare il comportamento della funzionalità implementata:

1.4. Come ultimo punto, dopo il bonus, spiegare quale istruzione assembly complessa

3. Come membro senior del SOC ti è richiesto di convincere il dipendente che il file non è maligno.

Immagine originale:

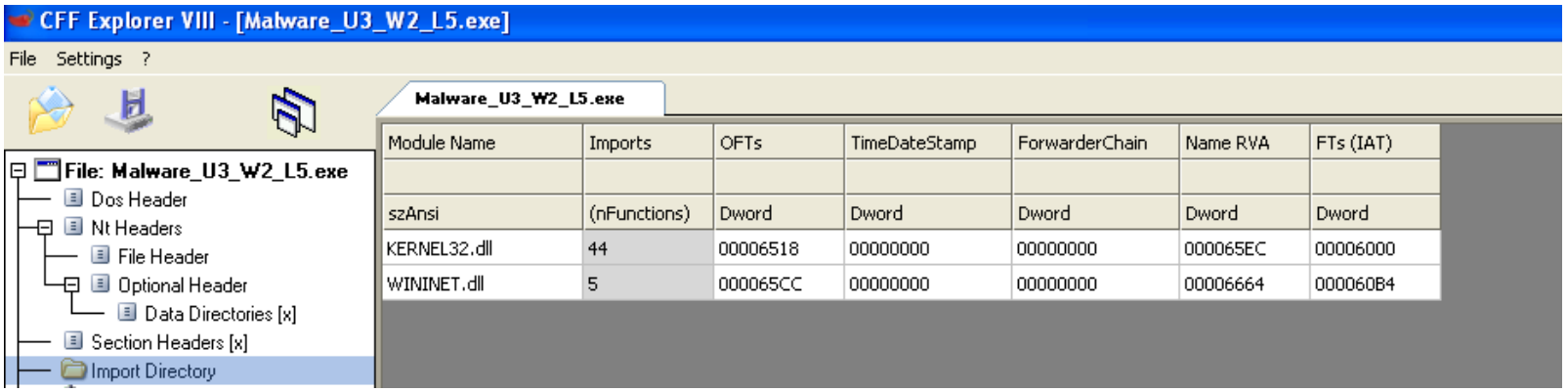
Figura 1



Analisi statica:

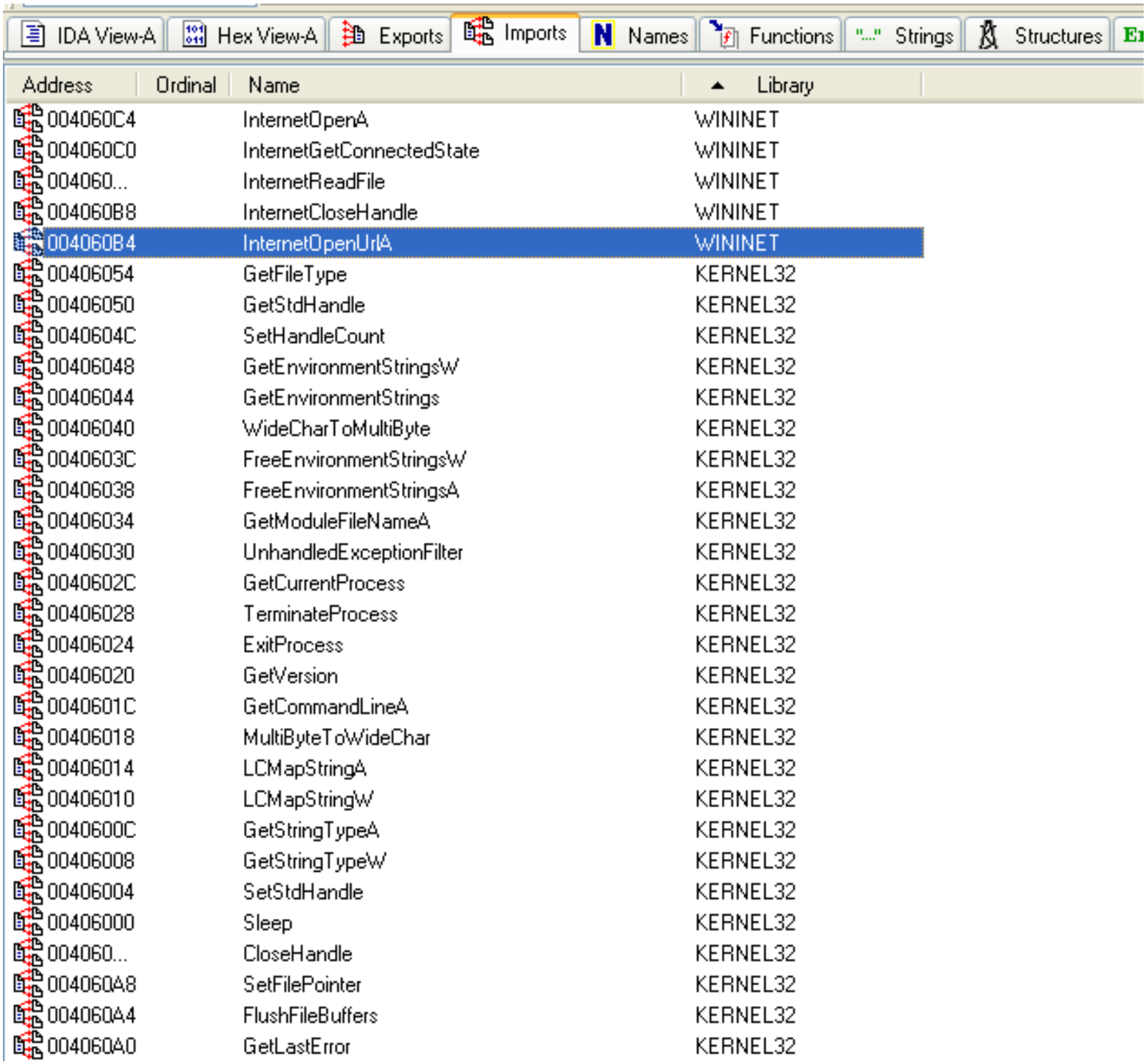
1. Quali librerie vengono importate dal file eseguibile?

Per verificare le librerie importate si possono utilizzare dei decompilatori come CFF Explorer oppure Ida per facilitare i tempi:
Andando su CFF Explorer su import directory possiamo vedere le librerie che usa il programma:



Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

Esempio con Ida. Come si vede su Ida vengono divise in base alle funzioni delle librerie:



Address	Ordinal	Name	Library
004060C4		InternetOpenA	WININET
004060C0		InternetGetConnectedState	WININET
004060...		InternetReadFile	WININET
004060B8		InternetCloseHandle	WININET
004060B4		InternetOpenUrlA	WININET
00406054		GetFileType	KERNEL32
00406050		GetStdHandle	KERNEL32
0040604C		SetHandleCount	KERNEL32
00406048		GetEnvironmentStringsW	KERNEL32
00406044		GetEnvironmentStrings	KERNEL32
00406040		WideCharToMultiByte	KERNEL32
0040603C		FreeEnvironmentStringsW	KERNEL32
00406038		FreeEnvironmentStringsA	KERNEL32
00406034		GetModuleFileNameA	KERNEL32
00406030		UnhandledExceptionFilter	KERNEL32
0040602C		GetCurrentProcess	KERNEL32
00406028		TerminateProcess	KERNEL32
00406024		ExitProcess	KERNEL32
00406020		GetVersion	KERNEL32
0040601C		GetCommandLineA	KERNEL32
00406018		MultiByteToWideChar	KERNEL32
00406014		LCMapStringA	KERNEL32
00406010		LCMapStringW	KERNEL32
0040600C		GetStringTypeA	KERNEL32
00406008		GetStringTypeW	KERNEL32
00406004		SetStdHandle	KERNEL32
00406000		Sleep	KERNEL32
004060...		CloseHandle	KERNEL32
004060A8		SetFilePointer	KERNEL32
004060A4		FlushFileBuffers	KERNEL32
004060A0		GetLastError	KERNEL32

Come si vede dai due esempi vengono utilizzate:
KERNEL32.dll: Che è una delle librerie fondamentali di Windows che consentono al sistema operativo di gestire risorse hardware,memoria,processi e file.

WININET.dll: Fornisce funzionalità per creare connessioni, inviare richieste HTTP, scaricare file, eseguire operazioni FTP e molto altro. Insomma viene utilizzata dal malware per inviare o ricevere informazioni per conto dell'attaccante.

2. Quali sono le sezioni di cui si compone il file eseguibile del malware?

Le sezioni sono parti di un file eseguibile o di una libreria che organizzano e memorizzano diversi tipi di dati e codice, compongono il malware e si possono trovare facilmente con l'uso di un decompilatore di nome CFF Explorer nel esempio sottostante possiamo vederle:

Malware_U3_W2_L5.exe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00004A78	00001000	00005000	00001000	00000000	00000000	0000	0000	60000020
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000	0000	0000	40000040
.data	00003F08	00007000	00003000	00007000	00000000	00000000	0000	0000	C0000040

Come possiamo vedere ci sono tre sezioni chiamate .text, .rdata , .data
La prima sezione .text contiene il codice eseguibile del programma, come istruzioni e funzioni.
La seconda sezione .rdata contiene dati di sola lettura, come costanti o stringhe, che il programma utilizza durante l'esecuzione.
La terza sezione .data memorizza dati inizializzati che possono essere modificati durante l'esecuzione del programma, come variabili globali.

Un altro esempio con Ghidra :

Memory Map [CodeBrowser: test:/Malware_U3_W2_L5.exe]

File Edit Help

Memory Map - Image Base: 00400000

Memory Blocks

Name	Start	End	Length	R	W	X	Volatile	Overlay	Type	Initialized	Byte Source	Source
Headers	00400000	00400fff	0x1000	☑	☐	☐	☐	☐	Default	☑	File: Malware_U3_W2...	
.text	00401000	00405fff	0x5000	☑	☐	☑	☐	☐	Default	☑	File: Malware_U3_W2...	
.rdata	00406000	00406fff	0x1000	☑	☐	☐	☐	☐	Default	☑	File: Malware_U3_W2...	
.data	00407000	00409fff	0x3000	☑	☑	☐	☐	☐	Default	☑	File: Malware_U3_W2...	
.data	0040a000	0040af07	0xf08	☑	☑	☐	☐	☐	Default	☐		
tdb	ffdff000	ffdfffff	0x1000	☑	☑	☐	☐	☐	Default	☑		

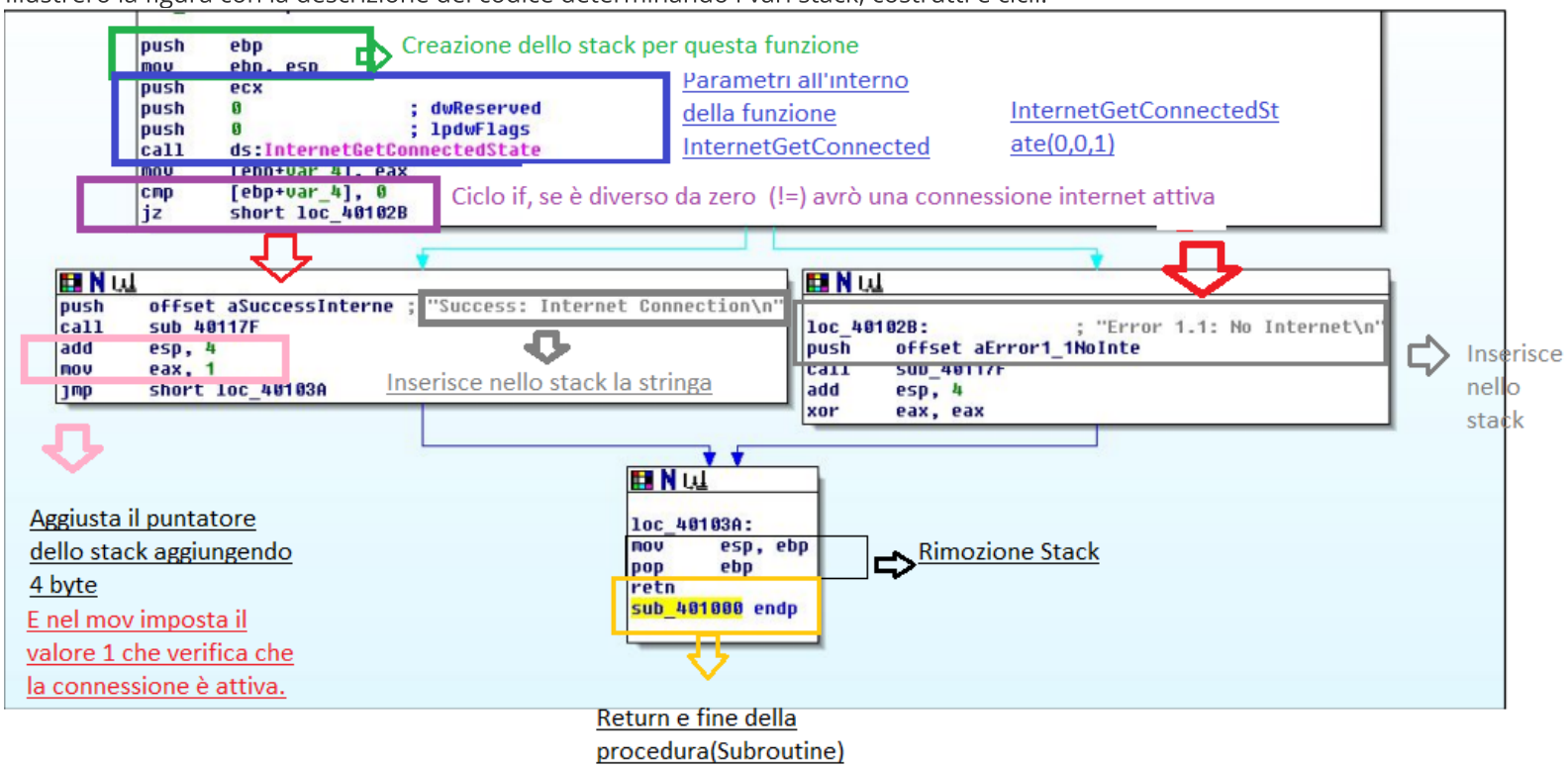
Visualizzazione sul codice delle determinate sezioni all'inizio del programma:

Listing: Malware_U3_W2_L5.exe

```
004000df 00              ??         00h
004000e0 00              ??         00h
004000e1 00              ??         00h
004000e2 00              ??         00h
004000e3 00              ??         00h
004000e4 00              ??         00h
004000e5 00              ??         00h
004000e6 00              ??         00h
004000e7 00              ??         00h
004000e8 50 45 00        IMAGE_NT...                                  = 000065e4
         00 4c 01                                                     = 6518h
         03 00 a1 ...
004001e0 2e 74 65        IMAGE_SE...                                  .text
         78 74 00
         00 00 78 ...
00400208 2e 72 64        IMAGE_SE...                                  .rdata
         61 74 61
         00 00 5e ...
00400230 2e 64 61        IMAGE_SE...                                  .data
         74 61 00
         00 00 08 ...
00400258 00              ??         00h
```

1.1. Identificare i costrutti noti (creazione dello stack, eventuali cicli, costrutti):

Illustrerò la figura con la descrizione del codice determinando i vari stack, costrutti e cicli:



Quindi posso constatare un paio di cose, elencherò quelle più importanti:
1. Creazione dello stack di memoria nel blocco di istruzioni all'interno del rettangolo verde per questa funzione.
2. I parametri che andranno a comporre la funzione dichiarati con il push, tra cui un parametro sconosciuto richiamato sopra la funzione nel rettangolo blu e richiama la libreria WININET.ddl per utilizzare questa funzione

```
51          PUSH    param_1
6a 00       PUSH    0x0
6a 00       PUSH    0x0
ff 15 c0    CALL    dword ptr [->WININET.DLL::InternetGetConnected... = 000065fa
```

3. Costrutto condizionale IF, identificato dalla coppia di istruzioni nel rettangolo viola, dove sicuramente viene utilizzato un diverso da zero(!= 0) .

4. Aggiusta il puntatore dello stack e lo allarga aggiungendo 4 byte, e nel move possiamo vedere che c'è il valore 1 che indica che la connessione è attiva. Si trova nel riquadro rosa.

5. Infine possiamo vedere la rimozione dello stack all'interno della figura nera, è un return nella figura gialla.

Quindi ho ipotizzato questo tipo di codice per la funzione:

```c

void funzione_generica_controllo_connessione(int param1) {

int statointernet; // Qua sono sicuro perchè si tratta di numeri come inserito nell'assembly
int dwReserved = 0;
int lpdwFlags = 0;
statointernet = internetgetconnectedstate(dwReserved,lpdwFlags,param1); // 0, 0 , parametro sconosciuto


if (statointernet != 0) {
printf ("Success: Internet Connection\n");
} else {
printf ("Error 1.1: No Internet\n");
}
return;// fine funzione e return
}
```

1.3 Ipotesi funzionalità:

Ipotesi funzionalità :
La funzione getInternetConnectState viene utilizzata per verificare la presenza di una connessione Internet su un determinato dispositivo.

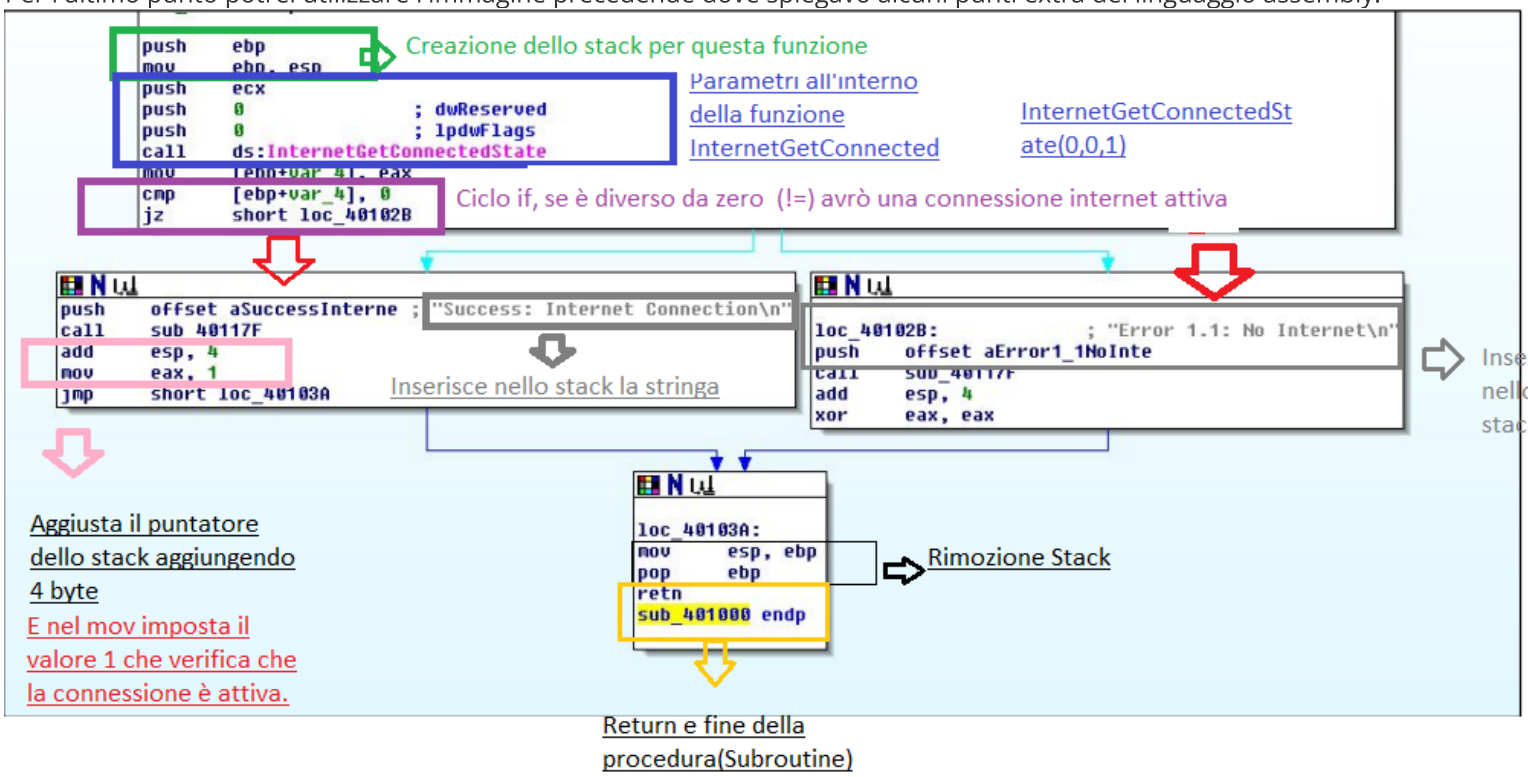
La logica di controllo è implementata attraverso un'istruzione condizionale "if". La condizione dell'istruzione "if" verifica se il valore restituito dalla funzione getInternetConnectState è uguale a 0.

Se il valore restituito è uguale a 0, significa che la connessione Internet non è presente. In questo caso, la funzione mostra il messaggio "no internet" a schermo e completa la sua esecuzione.

Se il valore restituito dalla funzione getInternetConnectState è diverso da 0, indica che la connessione Internet è presente. In questo caso, la funzione mostra il messaggio "Success: Internet Connection" a schermo prima di terminare il suo compito.

1.4 Come ultimo punto, dopo il bonus, spiegare quale istruzione assembly complessa:

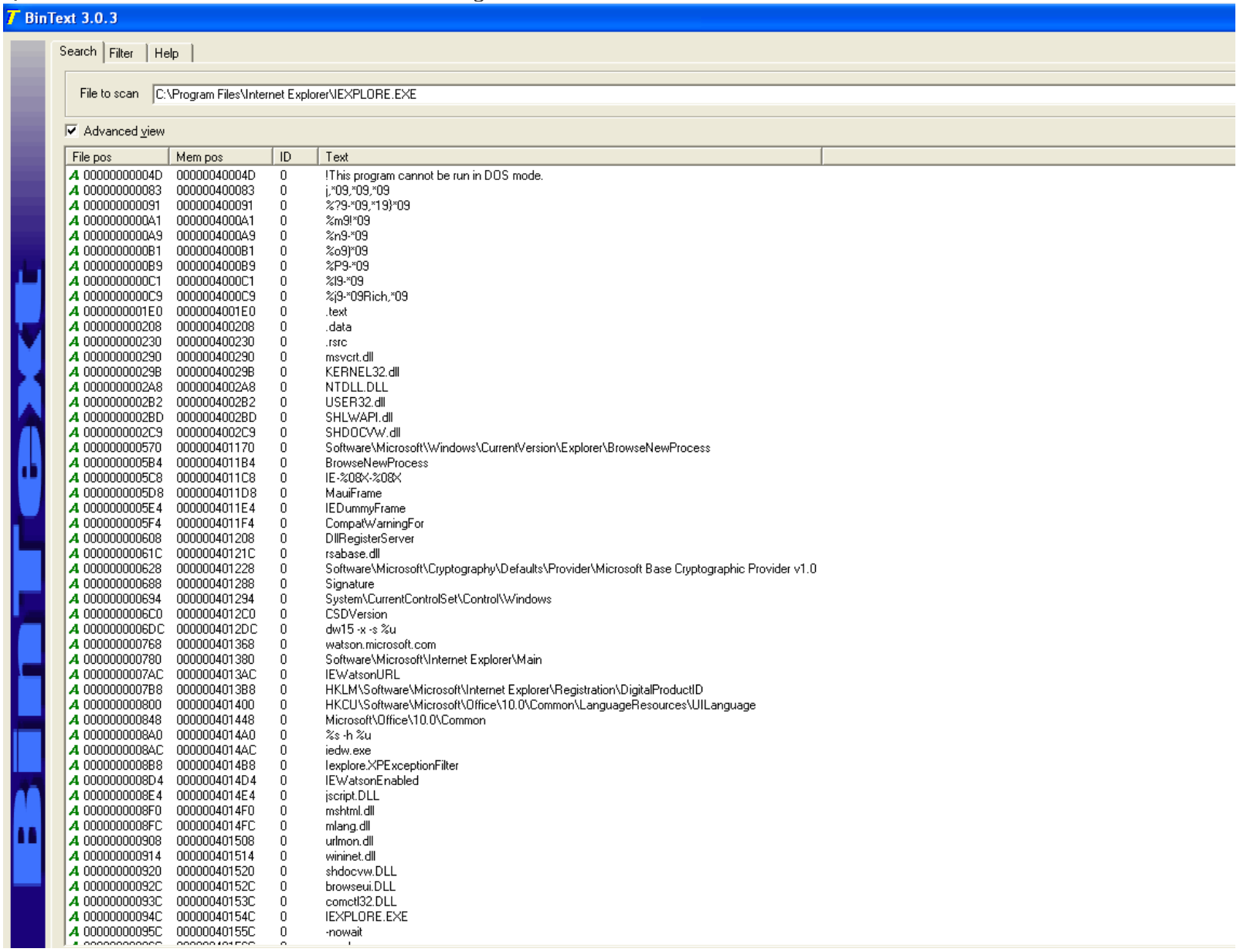
Per l'ultimo punto potrei utilizzare l'immagine precedende dove spiegavo alcuni punti extra del linguaggio assembly:



3. BONUS:

Per convincere il dipendende che il processo e il file IEXPLORE.EXE non sono maligni, si possono effettuare diverse scansioni, tra qui quella statica e dinamica:

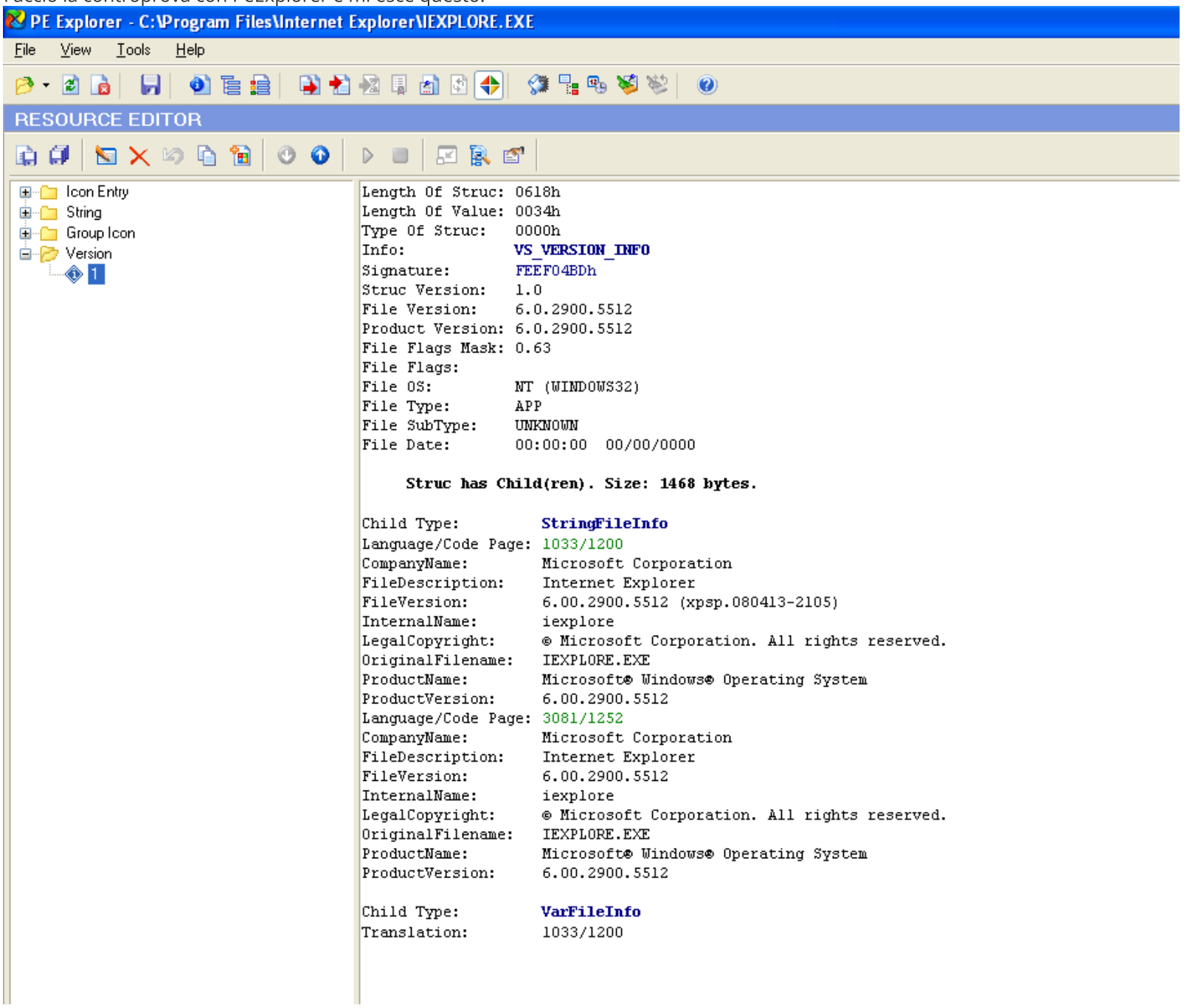
Quindi inizio a fare una scansione statica delle stringhe del file IEXPLORE.EXE con bintext:



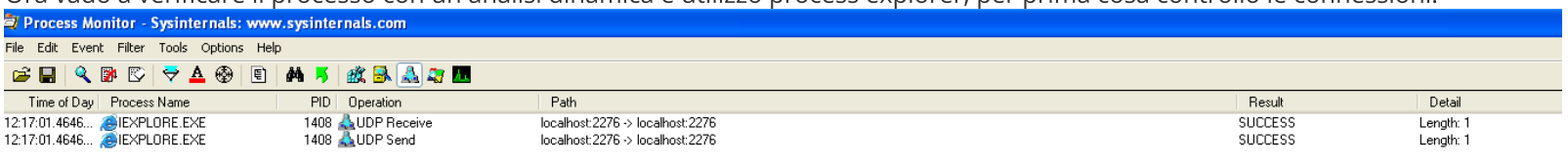
Mi sembra tutto ok, vado a verificare il produttore tramite le stringhe ed esce che la compagnia della creazione del file è proprio microsoft quindi sembra sicuro:

File pos	Mem pos	ID	Text
U 0000000164E	00000041800E	0	VS_VERSION_INFO
U 0000000164A	0000041806A	0	StringFileInfo
U 0000000164E	00000041808E	0	040904B0
U 0000000164A6	000000418A6	0	CompanyName
U 0000000164C0	000000418C0	0	Microsoft Corporation
U 0000000164F2	000000418F2	0	FileDescription
U 000000016514	000000418114	0	Internet Explorer
U 0000000153E	00000041813E	0	FileVersion
U 000000016558	000000418158	0	6.00.2900.5512 (xpsp.080413-2105)
U 0000000165A2	000000418A2	0	InternalName
U 0000000165BC	000000418BC	0	iexplore
U 0000000165D6	000000418D6	0	LegalCopyright
U 0000000165F6	000000418F6	0	Microsoft Corporation. All rights reserved.
U 000000016656	000000418256	0	OriginalFilename
U 000000016678	000000418278	0	IEXPLORE.EXE
U 0000000169A	00000041829A	0	ProductName
U 0000000166DA	000000418DA	0	Operating System
U 000000016706	000000418306	0	ProductVersion
U 000000016724	000000418324	0	6.00.2900.5512
U 00000001674A	00000041834A	0	0c0904E4
U 000000016762	000000418362	0	CompanyName
U 00000001677C	00000041837C	0	Microsoft Corporation
U 0000000167AE	000000418AE	0	FileDescription
U 0000000167D0	000000418D0	0	Internet Explorer
U 0000000167FA	000000418FA	0	FileVersion
U 000000016814	000000418414	0	6.00.2900.5512
U 00000001683A	00000041843A	0	InternalName
U 000000016854	000000418454	0	iexplore
U 00000001686E	00000041846E	0	LegalCopyright
U 00000001688E	00000041848E	0	Microsoft Corporation. All rights reserved.
U 0000000168EE	000000418EE	0	OriginalFilename
U 000000016910	000000418510	0	IEXPLORE.EXE
U 000000016932	000000418532	0	ProductName
U 000000016972	000000418572	0	Operating System
U 00000001699E	00000041859E	0	ProductVersion
U 0000000169BC	000000418BC	0	6.00.2900.5512
U 0000000169E2	000000418E2	0	VarFileInfo
U 000000016A02	000000418602	0	Translation
U 000000016B18	000000418718	0	Internet Explorer
R 000000016A38	000000418638	700	This is being run in compatibility mode and not all features are enabled.
R 000000016ACC	000000418CC	701	Internet Explorer Compatibility mode
R 000000016B16	000000418716	702	Internet Explorer
A 00000000004D	00000040004D	0	!This program cannot be run in DOS mode.
A 000000000083	000000400083	0	j,*09,*09,*09
A 000000000091	000000400091	0	%?9-*09,*19}*09
A 0000000000A1	0000004000A1	0	%m9!*09
A 0000000000A9	0000004000A9	0	%n9-*09

Faccio la controprova con PeExplorer e mi esce questo:



Ora vado a verificare il processo con un analisi dinamica e utilizzo process explorer, per prima cosa controllo le connessioni:



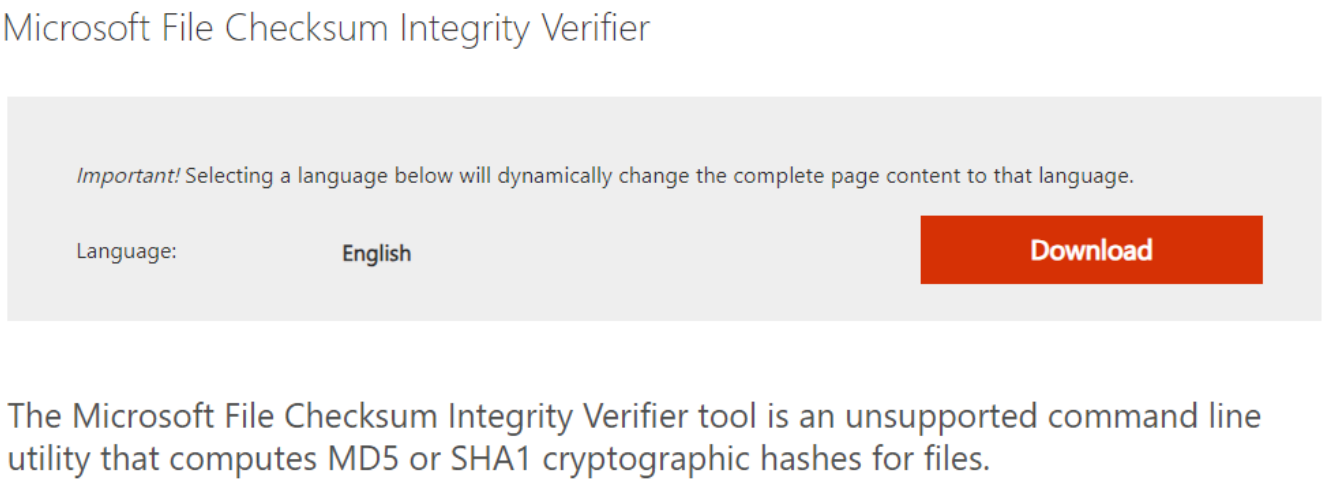
Sembrano connessioni legittime, ora vado a verificare cosa fa il processo:

12:16:56.1474...	IEXPLORE.EXE	1408	CreateFile	C:\DOCUMENTS AND SETTINGS
12:16:56.1474...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings
12:16:56.1474...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings
12:16:56.1478...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings
12:16:56.1478...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\ADMINISTRATOR
12:16:56.1478...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator
12:16:56.1482...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator
12:16:56.1482...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\Administrator
12:16:56.1486...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\Administrator\Cookies
12:16:56.1486...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Cookies
12:16:56.1487...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Cookies
12:16:56.1492...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\Administrator\Cookies
12:16:56.1493...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\Administrator\FAVORITES
12:16:56.1493...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Favorites
12:16:56.1496...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Favorites
12:16:56.1496...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\Administrator\Favorites
12:16:56.1499...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\Administrator\LOCAL SETTINGS
12:16:56.1499...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings
12:16:56.1499...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings
12:16:56.1503...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\Administrator\Local Settings
12:16:56.1504...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\Administrator\Local Settings\History
12:16:56.1504...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings\History
12:16:56.1508...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings\History
12:16:56.1509...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\Administrator\Local Settings\History
12:16:56.1514...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\Administrator\Local Settings\History\History.IE5
12:16:56.1514...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings\History\History.IE5
12:16:56.1516...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings\History\History.IE5
12:16:56.1523...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\Administrator\Local Settings\History\History.IE5
12:16:56.1524...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\Administrator\Local Settings\TEMPORARY INTERNET FILES
12:16:56.1524...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
12:16:56.1524...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
12:16:56.1525...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
12:16:56.1526...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5
12:16:56.1526...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5
12:16:56.1527...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5
12:16:56.1529...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5
12:16:56.1530...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\ALL USERS
12:16:56.1530...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\All Users
12:16:56.1531...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\All Users
12:16:56.1531...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\All Users
12:16:56.1531...	IEXPLORE.EXE	1408	CreateFile	C:\Documents and Settings\All Users\APPLICATION DATA
12:16:56.1532...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\All Users\Application Data
12:16:56.1532...	IEXPLORE.EXE	1408	QueryDirectory	C:\Documents and Settings\All Users\Application Data
12:16:56.1532...	IEXPLORE.EXE	1408	CloseFile	C:\Documents and Settings\All Users\Application Data
12:16:56.1533...	IEXPLORE.EXE	1408	CreateFile	C:\PROGRAM FILES
12:16:56.1533...	IEXPLORE.EXE	1408	QueryDirectory	C:\Program Files

Il processo sembra legittimo poichè come si vede dallo screen va a caricare tutti gli eventuali cookie , cache e preferiti presenti sul nostro computer,
quindi deduco che si tratta assolutamente del browser della microsoft internet explorer, e comunque non vedo azioni maligne da parte del processo.

Per sicurezza effettuo un altra prova con un software della microsoft che si chiama Microsoft File Checksum Integrity Verifier

Così andrò a calcolare l'hash di del file e verificarlo su virustotal:

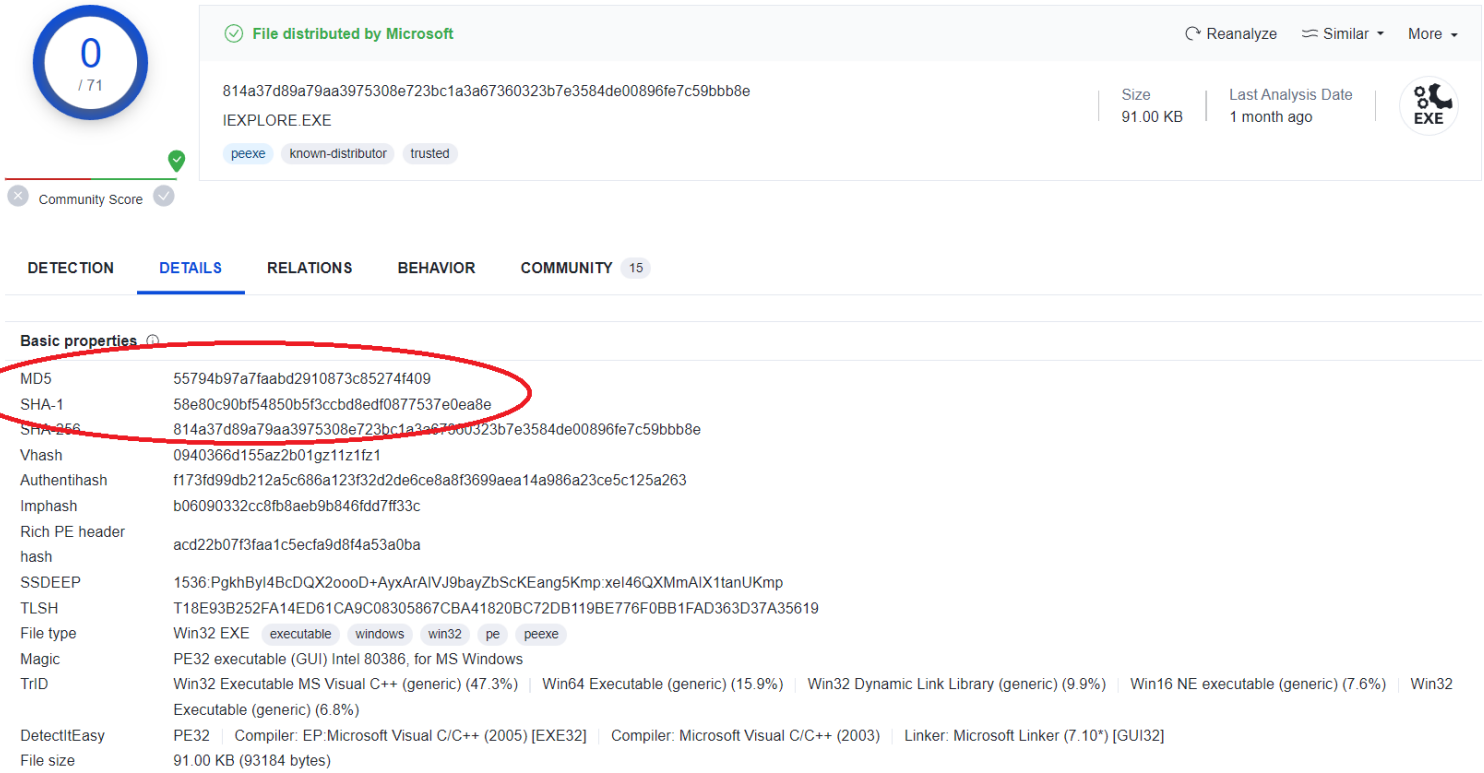


Calcolo l'hash del file con il programma :

```
C:\Program Files\Internet Explorer>fciv IEXPLORE.exe -sha1
//
// File Checksum Integrity Verifier version 2.05.
//
58e80c90bf54850b5f3ccbd8edf0877537e0ea8e iexplore.exe

C:\Program Files\Internet Explorer>
```

Verifico l'hash del file su virustotal, in questo modo controllo anche la scansione dei virus :



L'hash(SHA1) risulta uguale al programma della microsoft che ha comparato il file, quindi il file è sicuro.

Compito di Mendolia Valerio