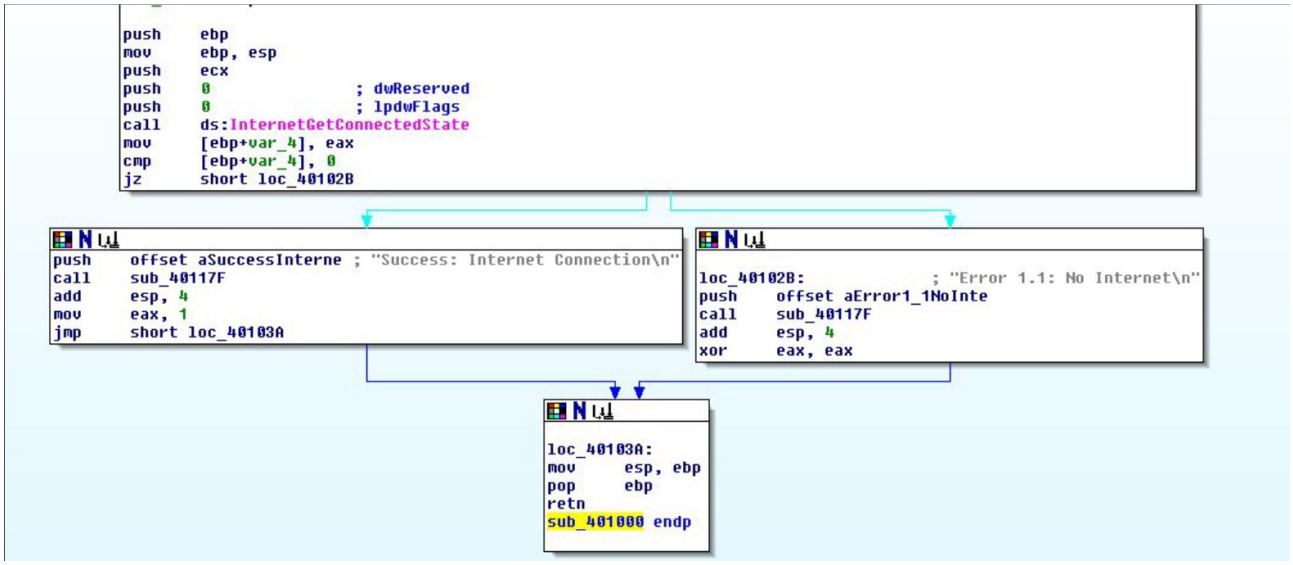
REPORT 13/01/2023 **LORENZO SCORBAIOLI**

TASK:

Con riferimento al file Malware_U3_W2-L5.exe presente sulla macchina 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?

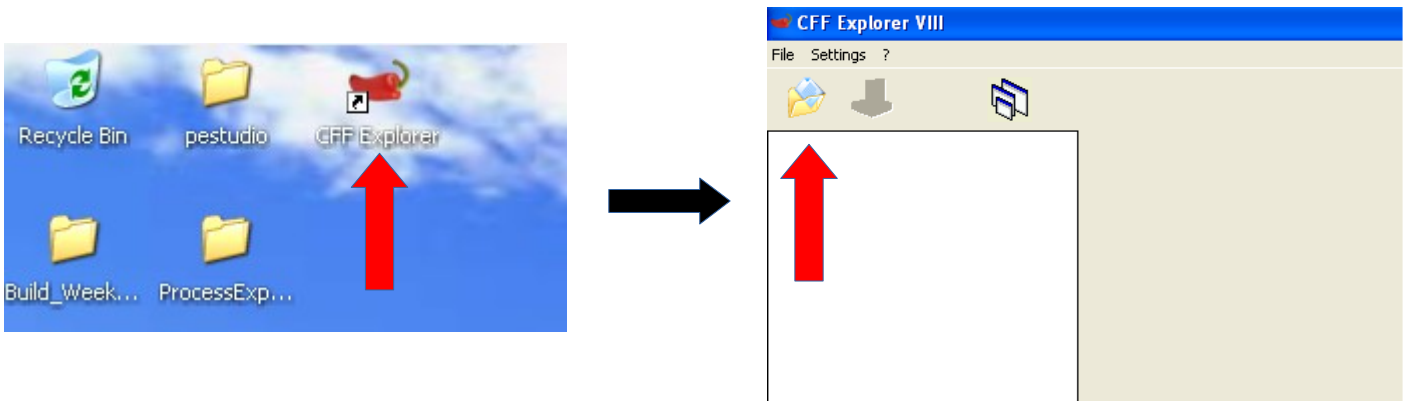
Con riferimento alla figura di seguito, rispondere ai seguenti quesiti:

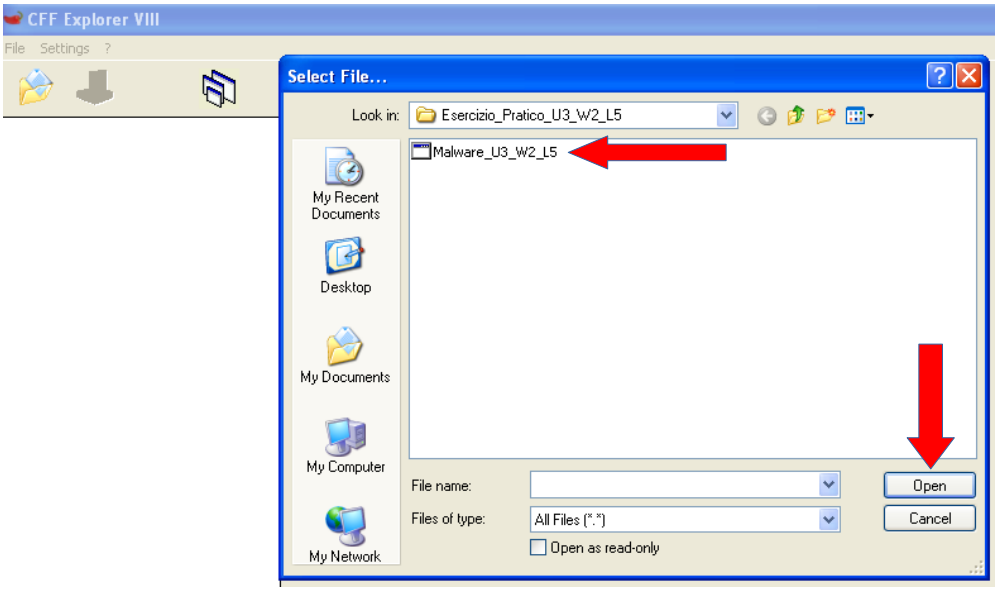


3 - Identificare i costrutti noti
4 - Ipotizzare il comportamento della funzionalità implementata
5 - BONUS fare una tabella con il significato delle singole righe di codice

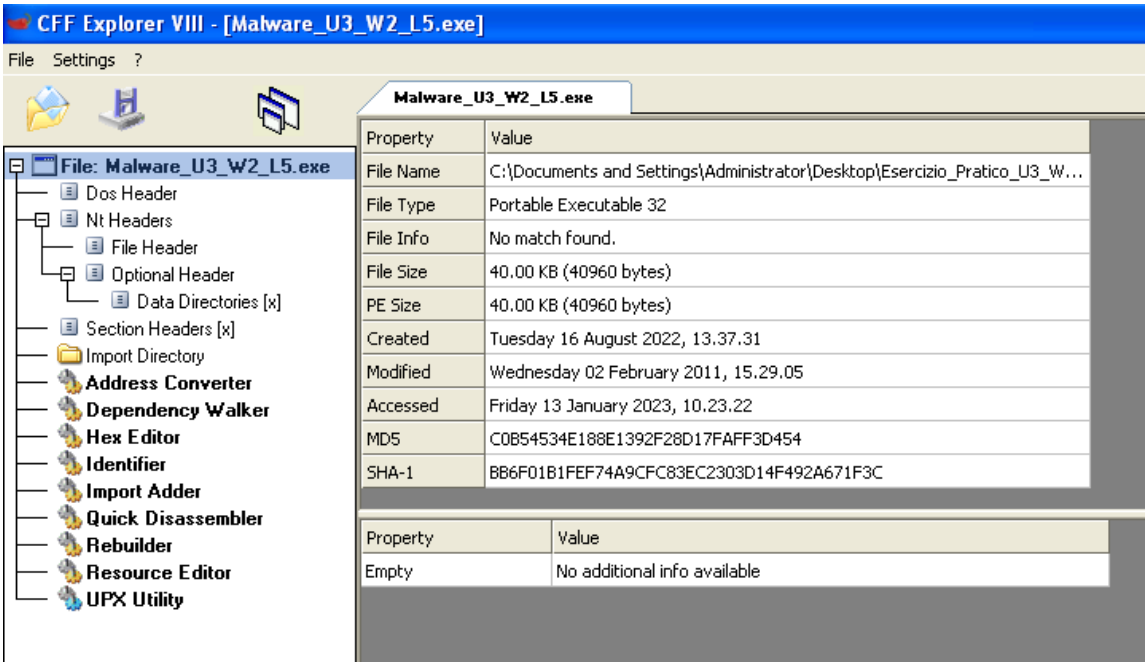
1)

Per andare a rispondere alle prime 2 domande poste dalla traccia, sono andato ad eseguire l'analisi statica basica tramite il tool CFF Explorer, con il quale ho aperto il file eseguibile Malware_U3_W2-L5.exe, il nostro eseguibile da analizzare:

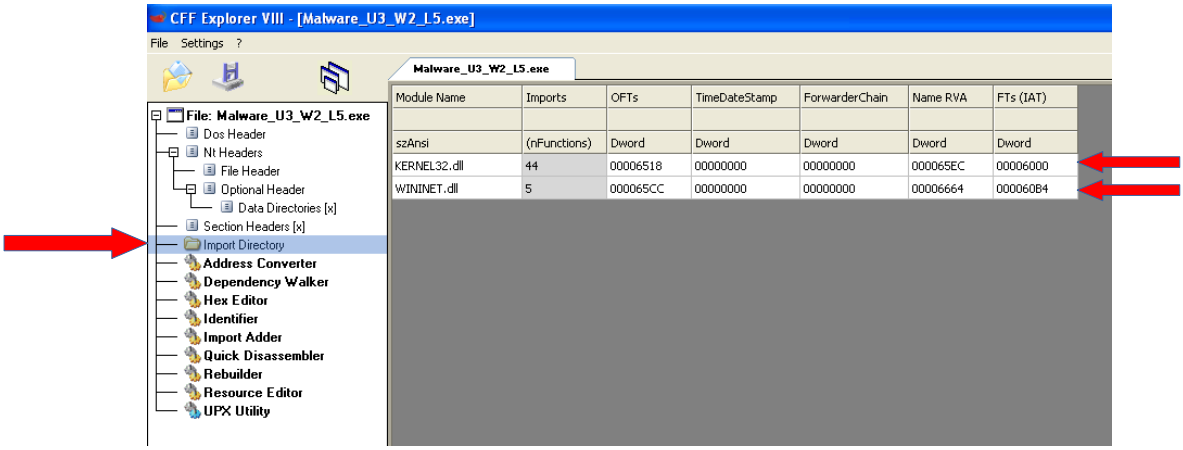




Una volta aperto il file che vogliamo analizzare la schermata che mi ha restituito CFF Explorer è la seguente:



A questo punto mi sono spostato alla voce “Import Directory” per andare a vedere quali librerie vengono importate dall’eseguibile:



Dall’immagine sopra possiamo vedere quindi che le librerie importate dal programma sono 2:

- KERNEL32.dll:

Libreria piuttosto comune che contiene le funzioni principali per interagire con il sistema operativo. Nell’immagine qui sotto possiamo nelle cornici rosse possiamo vedere i nomi delle funzioni importate da questa libreria.

OFTs	FTs (IAT)	Hint	Name	OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi	Dword	Dword	Word	szAnsi
000065E4	000065E4	0296	Sleep	000067A4	000067A4	0150	GetStartupInfoA
00006940	00006940	027C	SetStdHandle	000067B6	000067B6	0126	GetModuleHandleA
0000692E	0000692E	0156	GetStringTypeW	000067CA	000067CA	0109	GetEnvironmentVariableA
0000691C	0000691C	0153	GetStringTypeA	000067E4	000067E4	0175	GetVersionExA
0000690C	0000690C	01C0	LCMapStringW	000067F4	000067F4	019D	HeapDestroy
000068FC	000068FC	01BF	LCMapStringA	00006802	00006802	019B	HeapCreate
000068E6	000068E6	01E4	MultiByteToWideChar	00006810	00006810	02BF	VirtualFree
00006670	00006670	00CA	GetCommandLineA	0000681E	0000681E	019F	HeapFree
00006682	00006682	0174	GetVersion	0000682A	0000682A	022F	RtlUnwind
00006690	00006690	007D	ExitProcess	00006836	00006836	02DF	WriteFile
0000669E	0000669E	029E	TerminateProcess	00006842	00006842	0199	HeapAlloc
000066B2	000066B2	00F7	GetCurrentProcess	0000684E	0000684E	00BF	GetCPInfo
000066C6	000066C6	02AD	UnhandledExceptionFilter	0000685A	0000685A	00B9	GetACP
000066E2	000066E2	0124	GetModuleFileNameA	00006864	00006864	0131	GetOEMCP
000066F8	000066F8	00B2	FreeEnvironmentStringsA	00006870	00006870	02BB	VirtualAlloc
00006712	00006712	00B3	FreeEnvironmentStringsW	00006880	00006880	01A2	HeapReAlloc
0000672C	0000672C	02D2	WideCharToMultiByte	0000688E	0000688E	013E	GetProcAddress
00006742	00006742	0106	GetEnvironmentStrings	000068A0	000068A0	01C2	LoadLibraryA
0000675A	0000675A	0108	GetEnvironmentStringsW	000068B0	000068B0	011A	GetLastError
00006774	00006774	026D	SetHandleCount	000068C0	000068C0	00AA	FlushFileBuffers
00006786	00006786	0152	GetStdHandle	000068D4	000068D4	026A	SetFilePointer
00006796	00006796	0115	GetFileType	00006950	00006950	001B	CloseHandle

Possiamo notare nel riquadro verde le funzioni LoadLibraryA e GetProcAddress. Possiamo dedurre che la modalità di importazione è a runtime, cioè l’eseguibile richiama la libreria soltanto quando ha la necessità di usare una specifica funzione durante l’esecuzione del file.

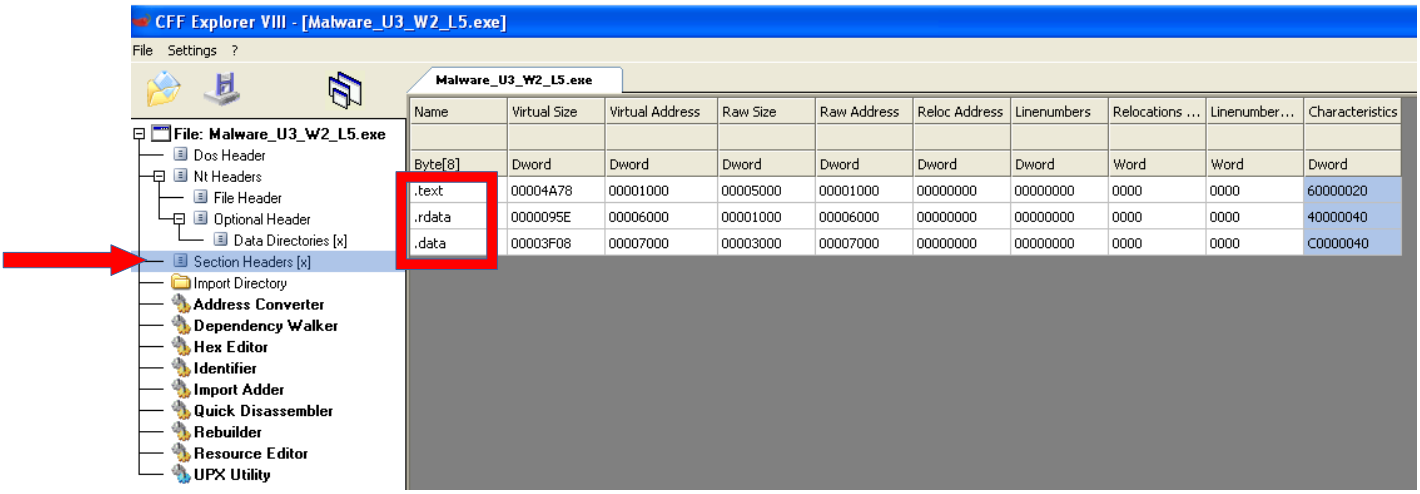
- WININET.dll:

Libreria che contiene le funzioni per l’implementazione di alcuni protocolli di
rete come HTTP, FTP, NTP. Nell’immagine qui sotto nel riquadro blu possiamo vedere i nomi delle funzioni importate da uesta libreria.

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00006640	00006640	0071	InternetOpenUrlA
0000662A	0000662A	0056	InternetCloseHandle
00006616	00006616	0077	InternetReadFile
000065FA	000065FA	0066	InternetGetConnectedState
00006654	00006654	006F	InternetOpenA

2)

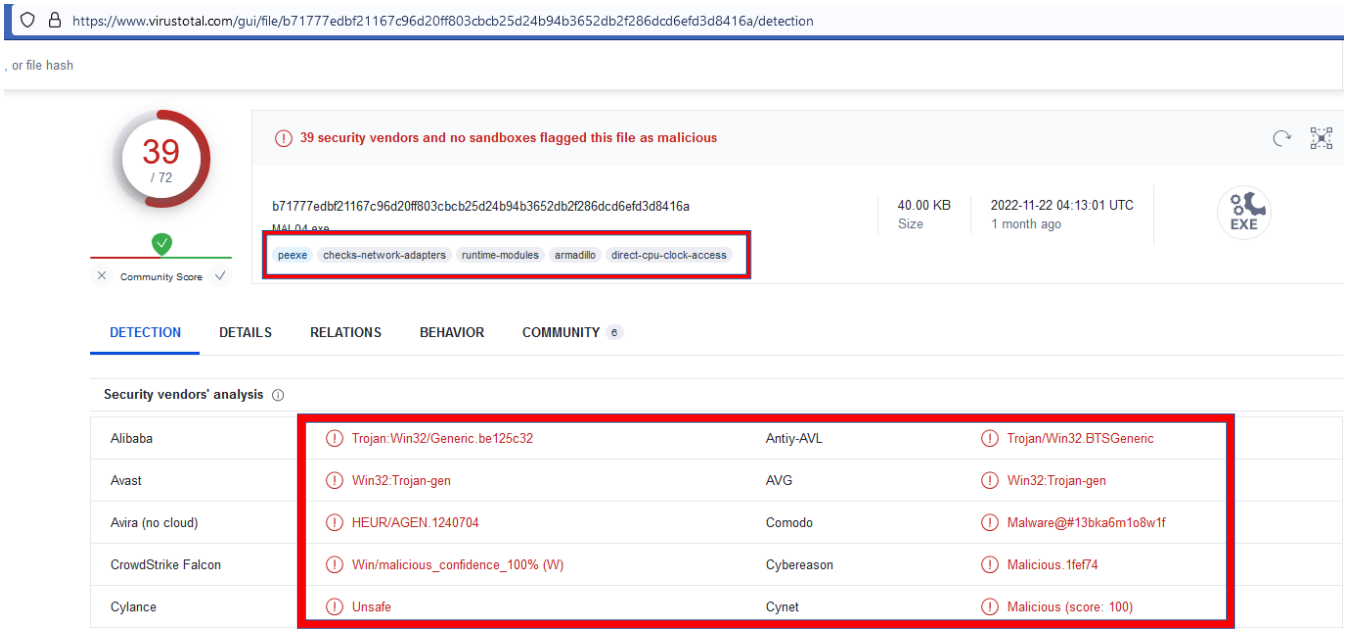
Spostandoci invece sulla voce “Section Headers” nel menù sulla sinistra possiamo andare a vedere quali sono le sezioni che compongono questo eseguibile, come possiamo vedere nell’immagine qui sotto nella cornice rossa:



Possiamo vedere come le sezioni che compongono questo software sono:

- .text: sezione che contiene le righe di codice che verranno eseguite dalla CPU una volta avviato il software.
- .rdata: sezione che include generalmente le informazioni riguardanti le librerie e le funzioni importate dall’eseguibile
- .data: sezione che contiene solitamente le variabili globali del programma eseguibile e che devono essere disponibili da qualsiasi parte del programma

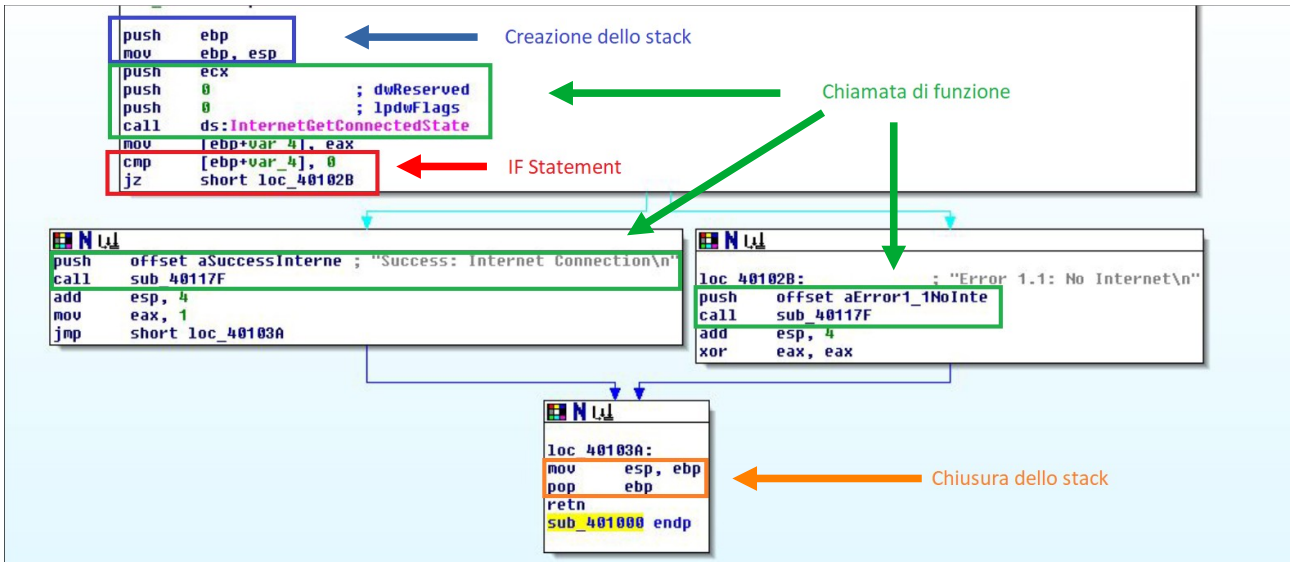
Andando a inserire l’hash md5 dell’eseguibile che abbiamo ottenuto con CFF Explorer su https://www.virustotal.com possiamo ottenere ulteriori informazioni su questo malware, come si può vedere dall’immagine qui sotto:



Possiamo vedere come molti security vendors lo identificano come un malware di tipo trojan

3)

Per quanto riguarda la seconda parte della traccia ho identificato i costrutti visibili nell'immagine qui sotto:



4)

Andando ad ipotizzare il funzionamento di questa porzione di codice fornita nell'immagine, penso che il suo scopo sia quello di controllare se la macchina infettata abbia accesso o meno ad una connessione internet andando forse a stampare a schermo l'eventuale risultato. In caso positivo viene effettuato un salto verso l'indirizzo di memoria "40103A", dove viene effettuata la chiusura dello stack ed il ritorno alla funzione chiamante. Se invece non è presente una connessione internet, dopo avere aumentato lo stack e reinizializzato il registro "eax", vengono eseguite altre attività che non sono presenti nell'immagine, arrivando comunque alla chiusura dello stack e al ritorno della funzione chiamante.

Un trasposizione delle righe di comando mostarte nell'immagine in **pseudocodice** potrebbe essere:

```
State = internetgetconnectedstate (p1, 0, 0)

if (state !=0) printf ("Active connection");
        return 0;
else printf("No connection");
        int a;
        .
        .
        .
        return 0;
```

5)

RIGA DI CODICE	SIGNIFICATO
Push ebp	Viene “pushato” il registro Extended Base Pointer in cima allo stack
Mov ebp, esp	Viene copiato il contenuto del registro Extended stack pointer nel registro Extended Base Pointer
Push ecx	Viene “pushato” il registro ecx in cima allo stack
Push 0 ; dwReserved	Viene “pushato” il parametro 0 in cima allo stack
Push 0 ; lpdwFlags	Viene “pushato” il parametro 0 in cima allo stack
Call ds:InternetGetConnectedState	Viene chiamata la funzione InternetGetConnectedState contenuta nel Data Segment
Mov [ebp+var_4], eax	Viene copiato il contenuto del registro EAX nel registro ebp+var_4
Cmp [ebp+var_4], 0	Viene sottratto il valore 0 alla variabile contenuta in EBP+var_4 andando a modificare Zf e CF a seconda del risultato
Jz short loc_40102B	Viene effettuato uno short jump nel caso ZF della riga precedente abbia assunto valore 1
Push offset aSuccessInterne ; “Success: Internet Connection\n”	Viene “pushato” l’offset aSuccessInterne sulla cima dello stack
Call sub_40117F	Viene chiamata la funzione all’indirizzo di memoria 40117F
Add esp, 4	Viene aggiunto il valore 4 a quello contenuto nel registro ESP
Mov eax, 1	Viene inserito il valore 1 all’interno del registro EAX
Jmp short loc_40103A	Viene effettuato uno shirt jump verso la memoria con indirizzo 40103A
Loc_40102B	Indica la locazione di memoria 40102B
Push offset aError1_NoInte ; “Error 1.1: No Internet\n”	Viene “pushato” l’offset aError1_NoInte in cima allo stack
Call sub_40117F	Viene chiamata la funzione all’indirizzo di memoria 40117F
Add esp, 4	Viene sommato il valore 4 a quello contenuto nel registro ESP
Xor eax, eax	Viene usata l’istruzione XOR per inizializzare a 0 il registro EAX
Loc 40103A	Indica la locazione di memoria 40103A

Mov esp, ebp	Viene copiato il contenuto del registro EBP nel registro ESP
Pop ebp	Viene levato il contenuto del registro EBP dalla cima dello stack
Retn	Organizza il ritorno al programma chiamante al termine di una procedura
Sub_401000 endp	Indica la fine della procedura all’indirizzo di memoria 401000