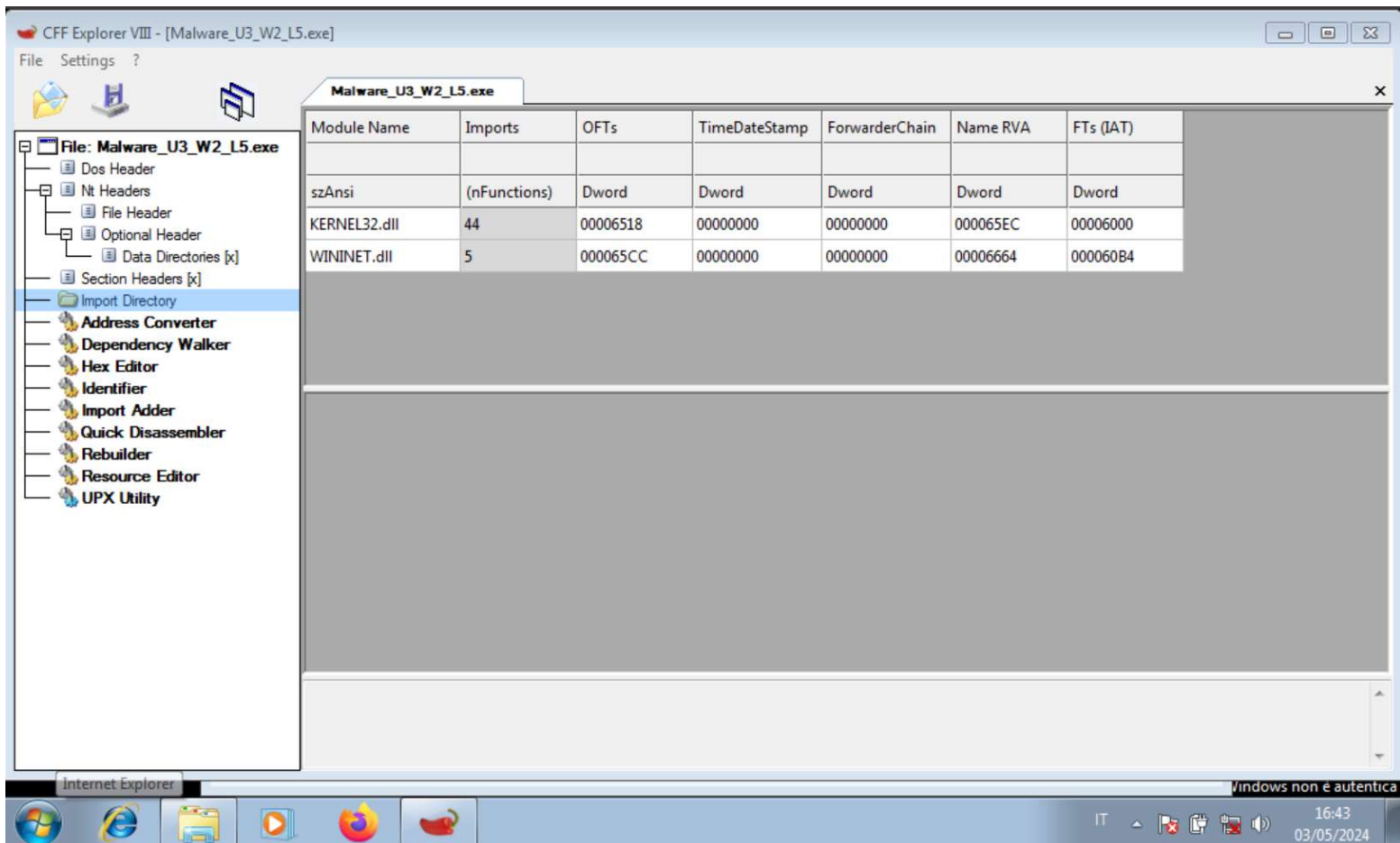


# PROGETTO S10/L5

Utilizzando il tool di analisi CFF Explorer è possibile riconoscere come vengano richiamate le seguenti librerie dal file eseguibile Malware\_U3\_W2\_L5.exe (**Punto 1**)



**KERNEL32.dll** :contiene diverse funzioni che permettono di interagire con il sistema operativo tra cui la gestione e modifica dei file e la gestione della memoria;

**WININET.dll** : contene diverse funzioni per l'implementazione di protocolli di rete come l'HTTP, l'FTP e l'NTP.

e le seguenti Sezioni: (**Punto 2**)

CFF Explorer VIII - [Malware\_U3\_W2\_L5.exe]

File Settings ?

Malware\_U3\_W2\_L5.exe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	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

Section Headers [x]

Import Directory

Address Converter

Dependency Walker

Hex Editor

Identifier

Import Adder

Quick Disassembler

Rebuilder

Resource Editor

UPX Utility

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ .....
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	.....@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	E8	00	00	00	.....e.....
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	is program.canno
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	t.be run.in.DOS.
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	mode...\$.....
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	

Windows 7  
Build 7601  
Questa copia di Windows non è autentica  
16:47

L'eseguibile è quindi composto da queste sezioni:

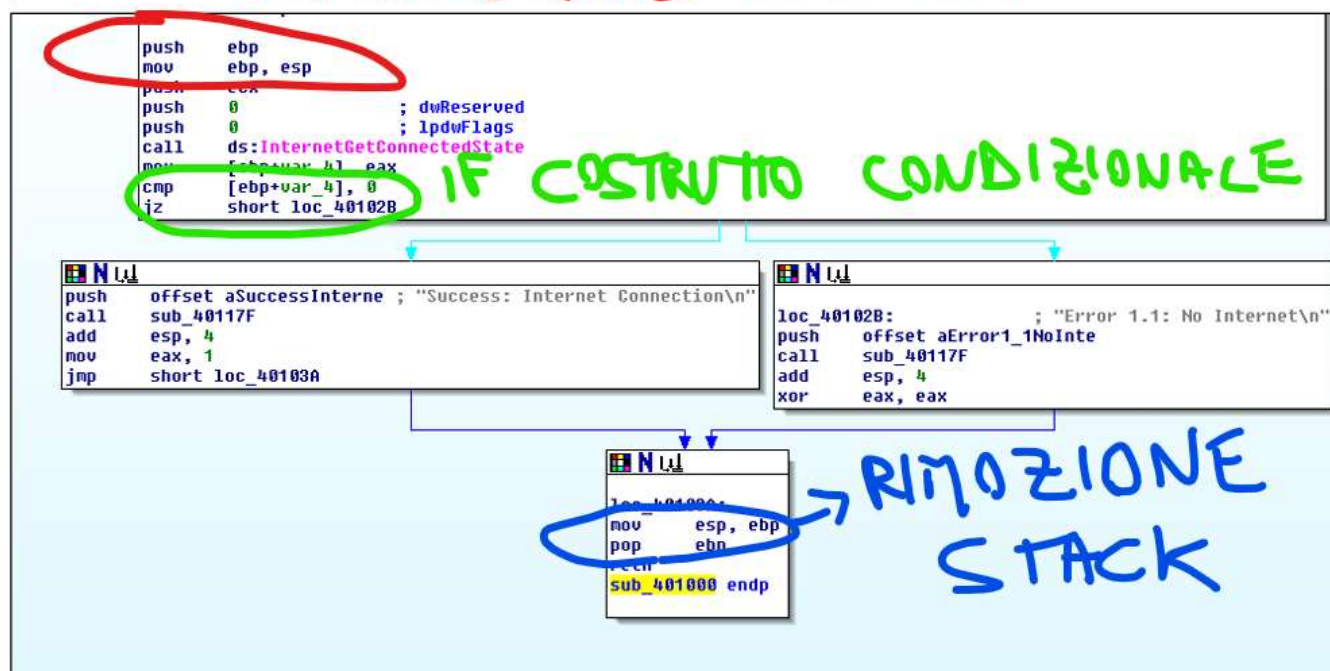
**.text** : che contiene tutte le istruzioni che la CPU eseguirà una volta che il programma viene avviato;

**.rdata** : che contiene informazioni riguardanti le librerie e le funzioni richiamate dal programma;

**.data** : che contiene i dati e le variabili globali utilizzate dal programma.

### Punto 3 - identificazione dei costrutti noti:

Figura 1



### Punto 4: Ipotesi comportamento funzionalità "InternetGetConnectedState":

Con questa funzionalità viene fatto un check dello stato di rete della macchina su cui è in esecuzione.

Con il costrutto condizionale IF viene controllato se il parametro che restituisce la funzione è uguale o diverso da 0.

Se = 0, viene eseguito un jump alla locazione LOC\_40102B e la funzione stampa a schermo : "Error 1.1: No internet" e termina l'esecuzione dopo aver ripulito lo stack;  
se invece il valore è ≠ 0, la funzione restituisce a schermo: "Success: Internet Connection", prima di concludere l'esecuzione e ripulire lo stack.

### Bonus:

#### commento riga per riga.

**push ebp :**

Salva il valore attuale del registro alla base (ebp) dello stack;

**mov ebp, esp :**

Inizializza il registro di base (ebp) con il valore corrente dello stack;

**push ecx :**

inserisce il valore contenuto nel registro ecx nello stack;

**push 0 ; dwReserved :**

esegue push nello stack per i flag della funzione;

**push 0 ; lpdwFlags :**  
**esegue push nello stack per i flag della funzione;**

**call ds:InternetGetConnectedState :**  
**Chiama la funzione InternetGetConnectedState attraverso segmento ds;**

**mov [ebp+var\_4], eax :**  
**Salva il valore di ritorno della funzione nella variabile locale [ebp+var\_4];**

**cmp [ebp+var\_4], 0 :**  
**Confronta il valore precedente salvato con 0;**

**jz short loc\_40102B :**  
**se sono uguali (ZF = 1) e salta alla locazione loc\_40102B;**

**push offset aSuccessInterne :**  
**Push dell'offset con messaggio di successo;**

**call sub\_40117F :**  
**Chiama la funzione sub\_40117F;**

**add esp, 4 :**  
**Aggiunge 4 al valore presente in esp;**

**mov eax, 1 :**  
**Valorizza il registro eax a 1 (probabilmente per indicare un successo);**

**jmp short loc\_40103A :**  
**Salta a loc\_40103A;**

**loc\_40102B :**  
**Punto di destinazione dopo il salto in caso manchi una connessione Internet;**

**push offset aError1\_1NoInte :**  
**Push dell'offset con messaggio di errore;**

**call sub\_40117F :**  
**Chiama la funzione sub\_40117F;**

**add esp, 4 :**  
**Aggiunge 4 al valore presente in esp;**

**xor eax, eax :**  
**Setta il registro eax a 0 con OR esclusiva;**

**loc\_40103A:**  
**Punto di destinazione dopo il salto nel caso abbia stabilito una connessione internet con successo;**

**mov esp, ebp:**  
**Esegue spostamento nella destinazione esp per ripulirlo;**

**pop ebp :**  
**Esegue pop per ripulire stack attuale;**

**retn :**  
**Restituisce il controllo alla funzione chiamante;**

**sub\_401000 endp;**  
**Fine della procedura sub\_401000**