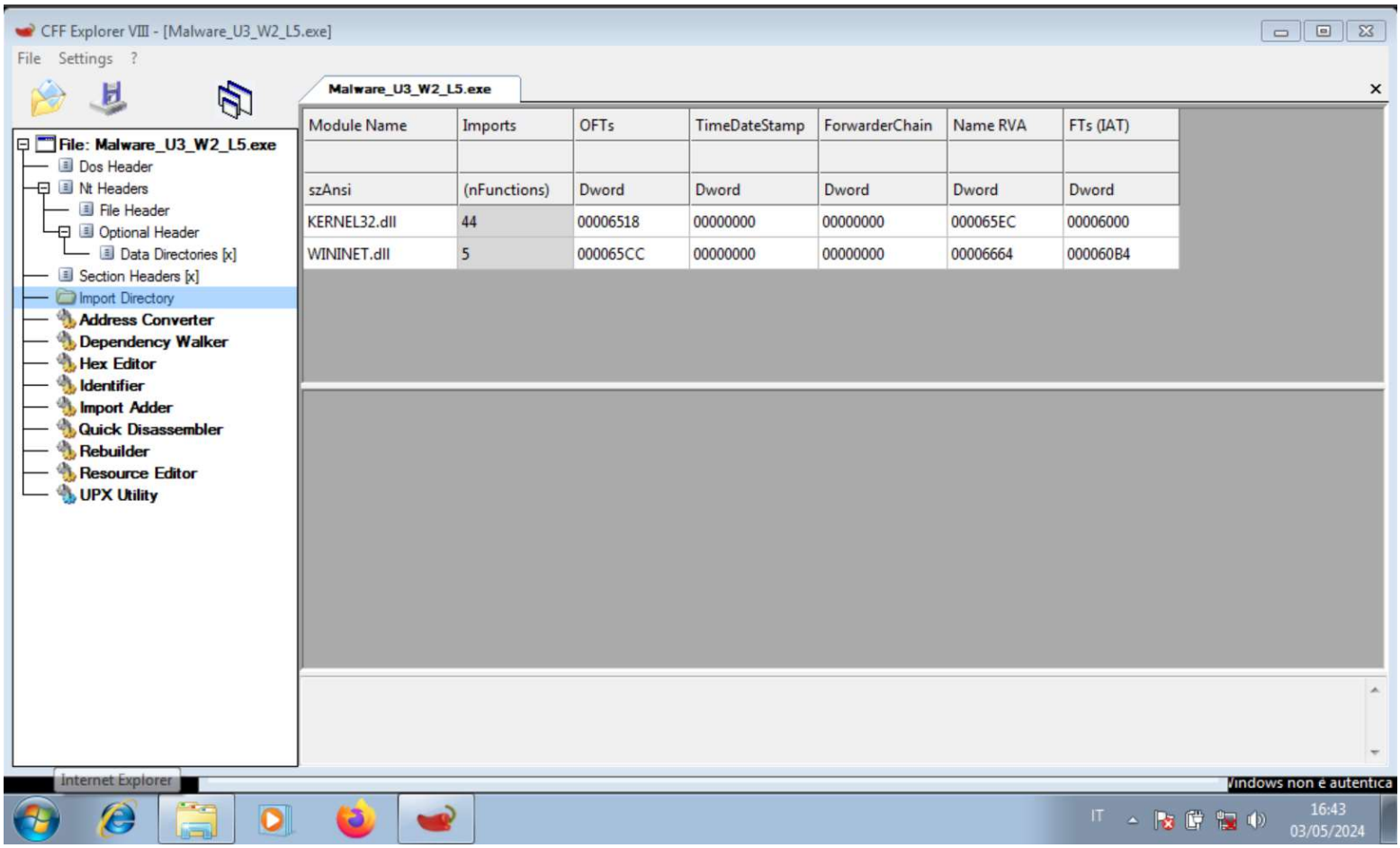
PROGETTO S10/L5

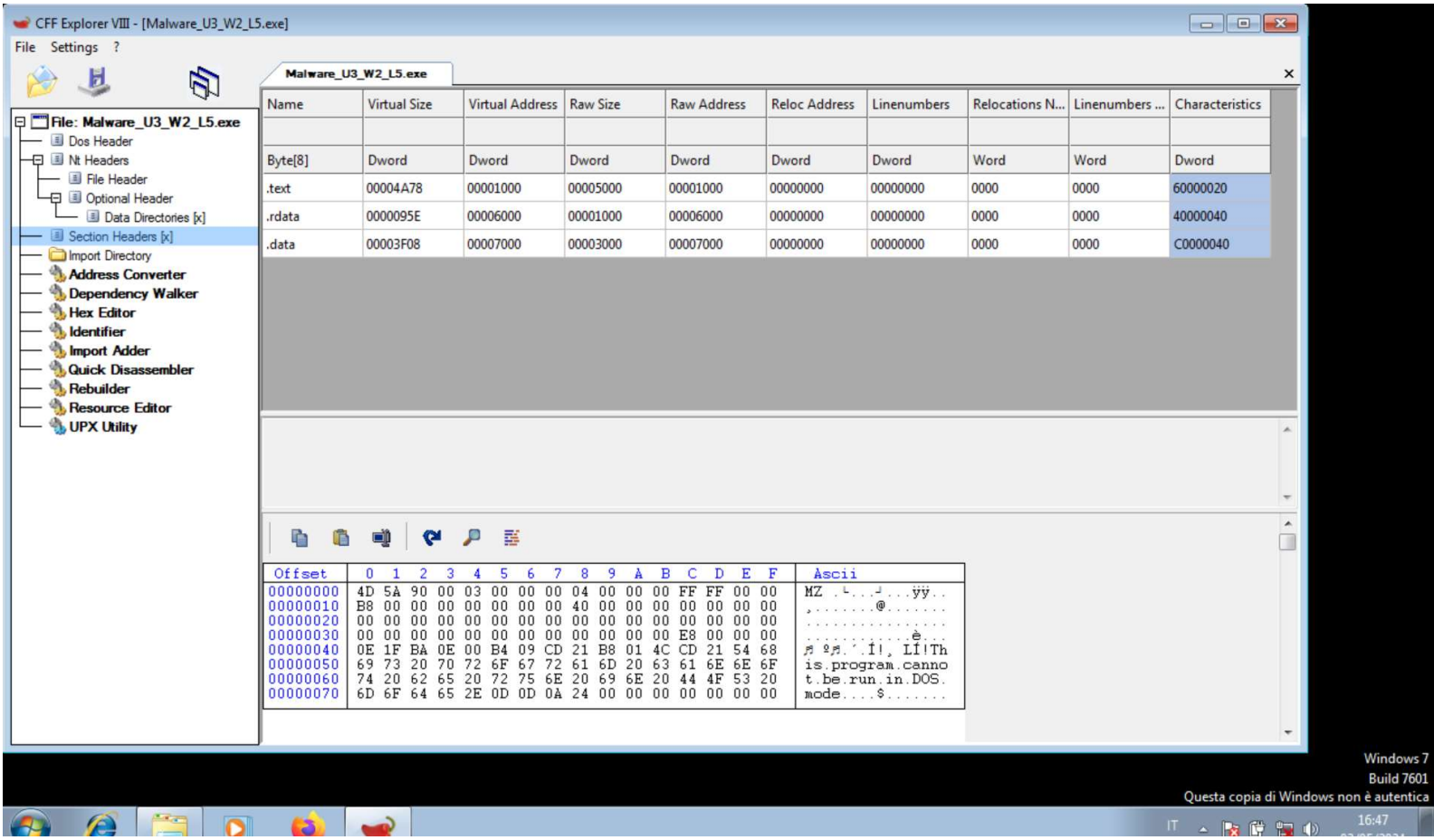
Utilizzando il tool di analisi CFF Explorer è possibile riconoscere come vengano richiamate le seguenti librerie dal file eseguibile Malware_U3_W2_L5.exe (**Punto 1**)



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**)



L'eseguibile è quindi composto da queste sezioni:

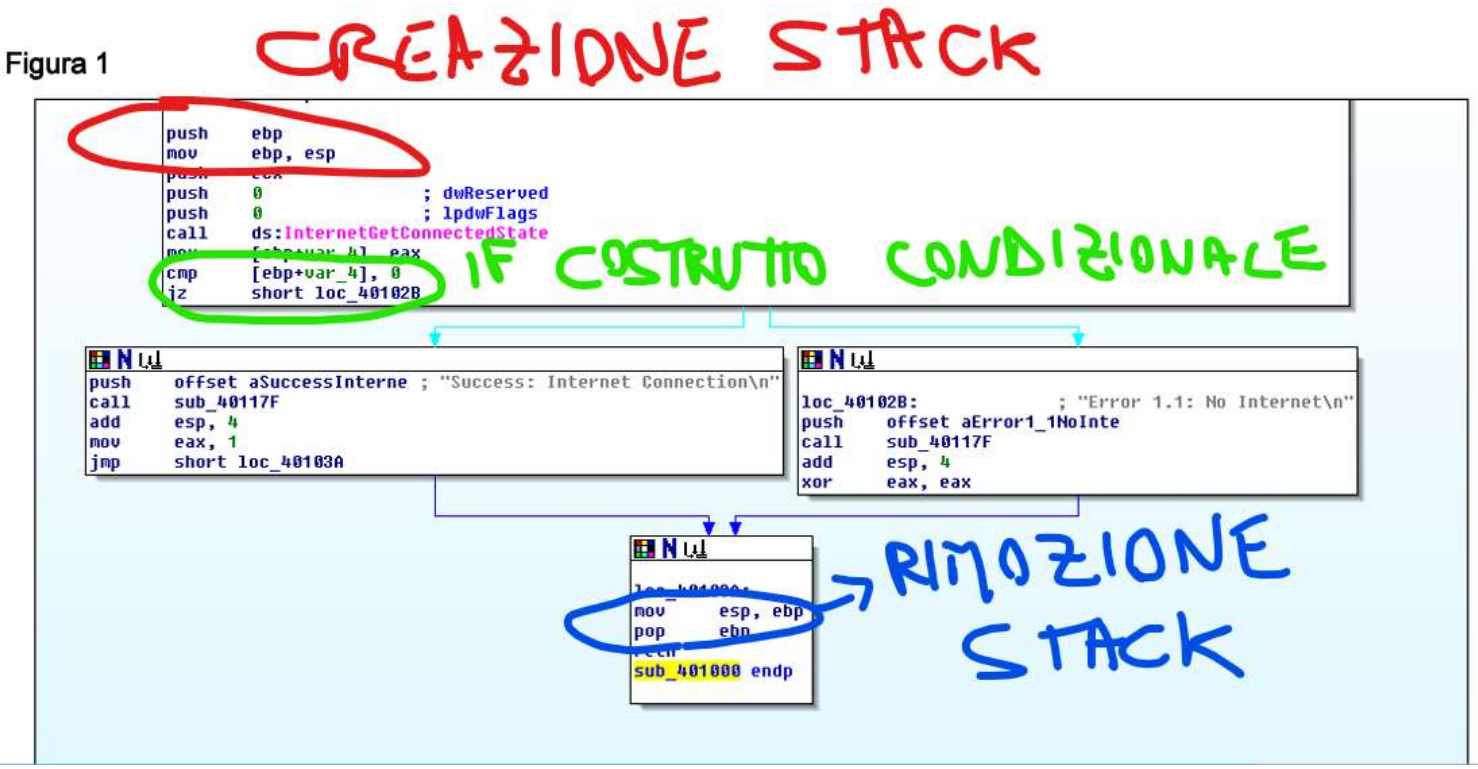
.text : che contiene tutte le istruzioni che la CPU eseguirà una volta che il programma viene avviato;

.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 exc 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