CYBER SECURITY & ETHICAL HACKING

GIORNO 5 –PROGETTO

TRACCIA

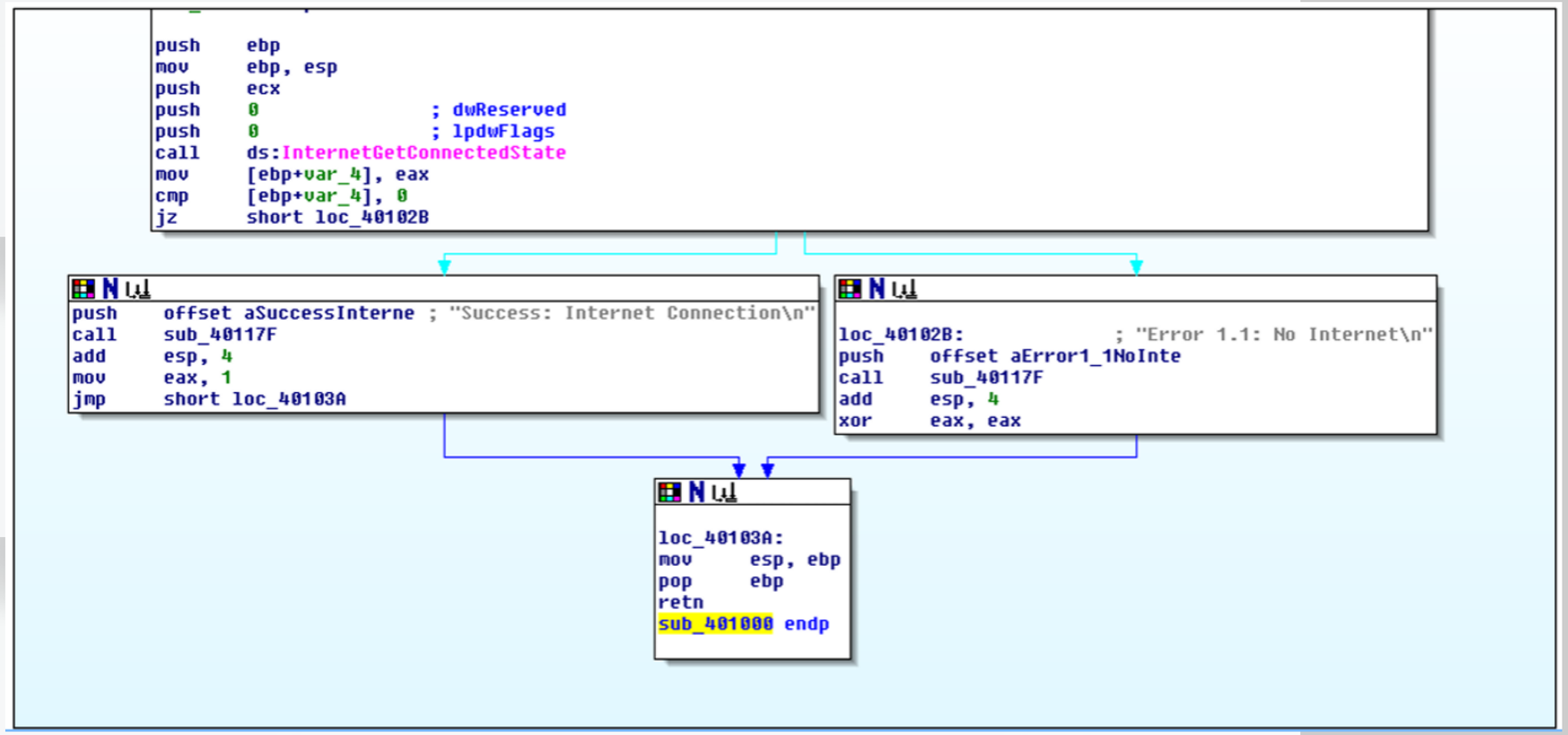
Con riferimento al file **Malware_U3_W2_L5** presente all'interno della cartella «**Esercizio_Pratico_U3_W2_L5** » sul desktop della macchina virtuale dedicata per l'analisi dei malware, rispondere ai seguenti quesiti:

1. Quali **librerie** vengono importate dal file eseguibile?
2. Quali sono le **sezioni** di cui si compone il file eseguibile del malware?

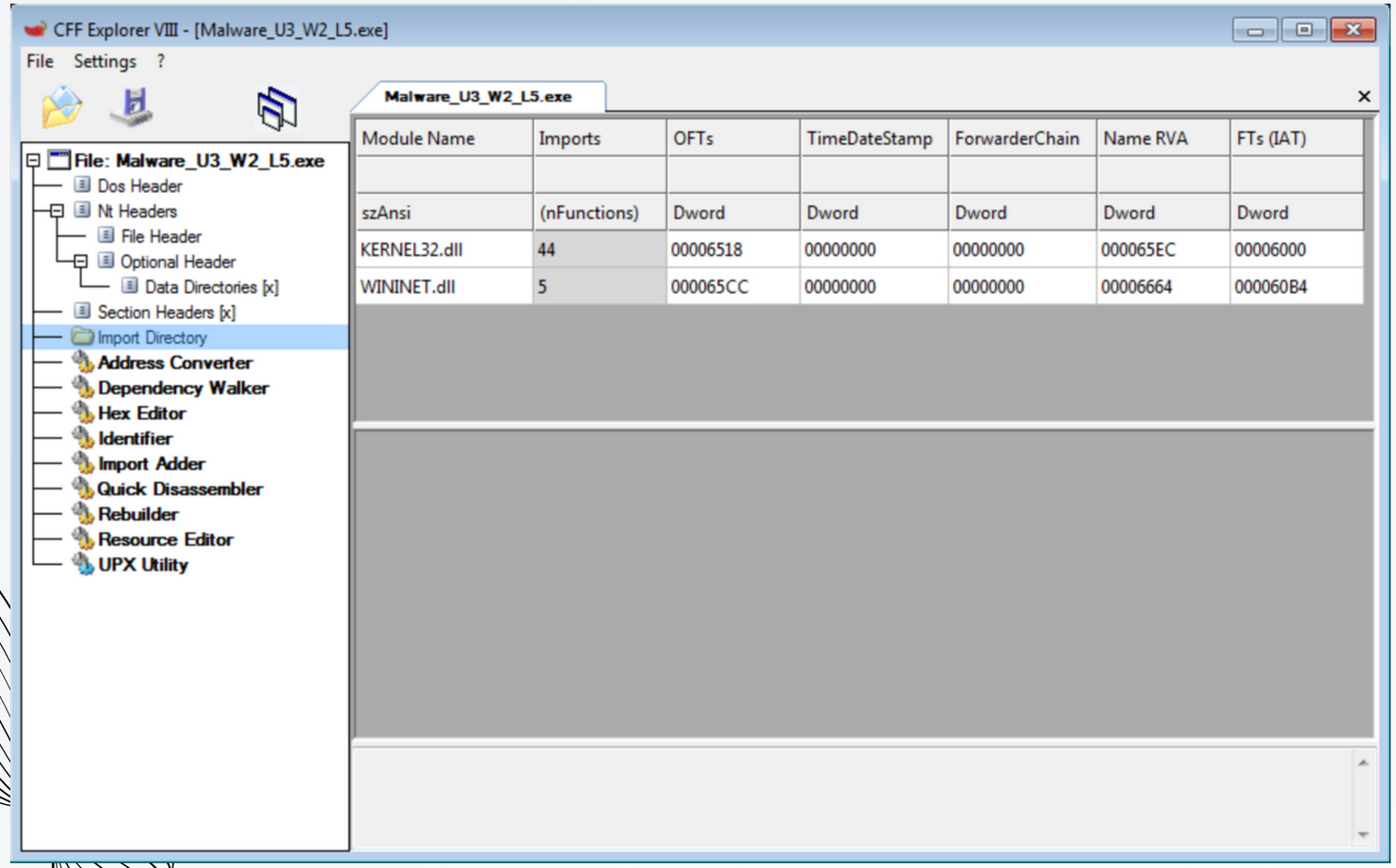
Con riferimento alla figura in slide 3, risponde ai seguenti quesiti:

3. Identificare i **costrutti** noti (creazione dello stack, eventuali cicli, altri costrutti)
4. **Ipotizzare il comportamento della funzionalità implementata**
5. BONUS fare tabella con significato delle singole righe di codice assembly

FIGURA 1


push ebp
mov ebp, esp
push ecx
push 0 ; dwReserved
push 0 ; lpdwFlags
call ds:InternetGetConnectedState
mov [ebp+var_4], eax
cmp [ebp+var_4], 0
jz short loc_40102B
push offset aSuccessInterne ; "Success: Internet Connection\n"
call sub_40117F
add esp, 4
mov eax, 1
jmp short loc_40103A
loc_40102B: ; "Error 1.1: No Internet\n"
push offset aError1_1NoInte
call sub_40117F
add esp, 4
xor eax, eax
loc_40103A:
mov esp, ebp
pop ebp
retn
sub_401000 endp

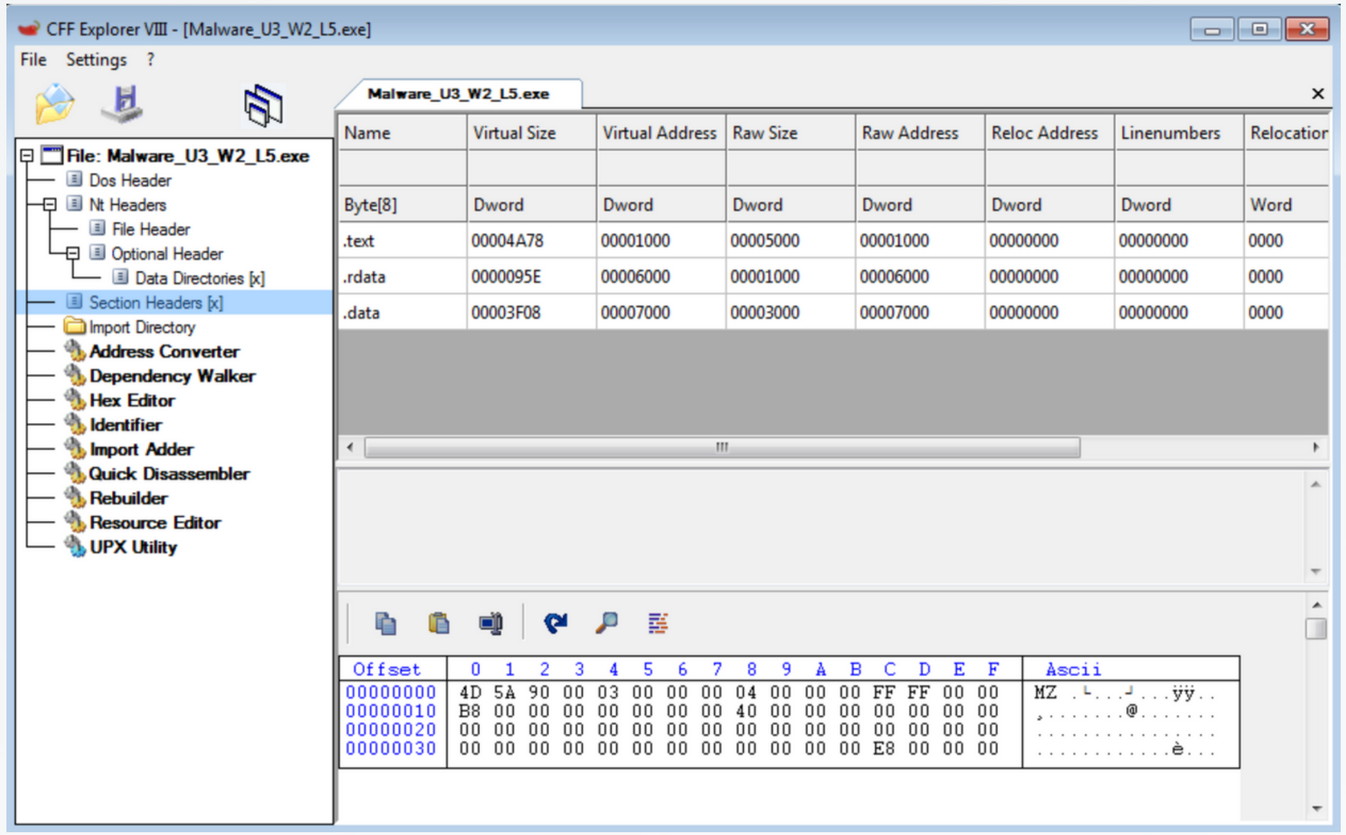
ESERCIZIO 1



Kernel32.dll: contiene le funzioni principali per interagire con il sistema operativo, ad esempio: manipolazione dei file, la gestione della memoria.

Wininet.dll:contiene le funzioni per l'implementazione di alcuni protocolli di rete come HTTP, FTP, NTP.

ESERCIZIO 2



.text: contiene le istruzioni (le righe di codice) che la CPU eseguirà una volta che il software sarà avviato. Generalmente questa è l'unica sezione di un file eseguibile che viene eseguita dalla CPU, in quanto tutte le altre sezioni contengono dati o informazioni a supporto.

.rdata: include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile, informazione che come abbiamo visto possiamo ricavare con CFF Explorer.

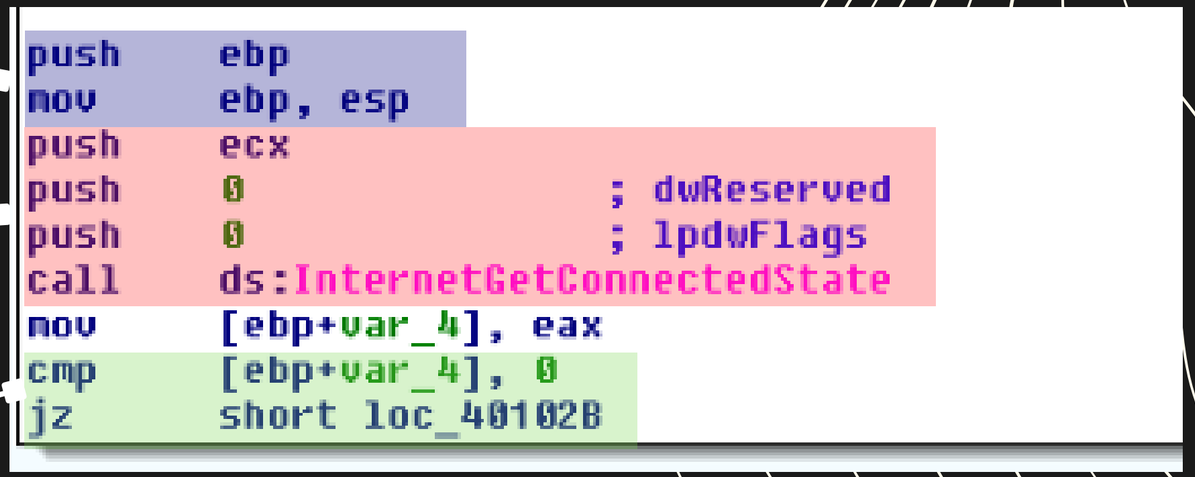
.data: contiene tipicamente i dati / le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma. Una variabile si dice globale quando non è definita all'interno di un contesto di una funzione, ma bensì è globalmente dichiarata ed è di conseguenza accessibile da qualsiasi funzione all'interno dell'eseguibile.

ESERCIZIO 3

Creazione dello stack

Chiamata di funzione. I parametri sono passati sullo stack tramite le istruzioni push

Ciclo IF



Pulizia dello stack

```
push    offset aSuccessInterne ; "Success: Internet Connection\n"
call    sub_40117F
add     esp, 4
mov     eax, 1
jmp     short loc_40103A
```

Pulizia dello stack

```
loc_40102B:                     ; "Error 1.1: No Internet\n"
push    offset aError1_1NoInte
call    sub_40117F
add     esp, 4
xor     eax, eax
```

Chiusura del programma

```
loc_40103A:
mov     esp, ebp
pop     ebp
retn
sub_401000 endp
```

ESERCIZIO 4

InternetGetConnectedState della libreria Wininet di Windows. Questa funzione è utilizzata per determinare se il sistema è connesso a Internet o meno. Il suo comportamento può variare a seconda delle condizioni di rete e delle impostazioni del sistema.
In generale, la funzione InternetGetConnectedState restituirà TRUE se il sistema è connesso a Internet e FALSE in caso contrario. Potrebbe anche fornire ulteriori informazioni sulla connessione, come la presenza di una connessione dial-up.

ESERCIZIO 5

Istruzioni	Descrizione
"push ebp"	Salva il valore corrente di ebp nello stack.
"mov ebp, esp"	Imposta ebp al valore corrente di esp, creando un nuovo frame nella pila.
"push ecx"	Salva il valore corrente di ecx nello stack.
"push 0"	Mette il valore 0 nello stack.
"push 0"	Mette un altro valore 0 nello stack.
"call ds:InternetGetConnectedState"	Chiama la funzione InternetGetConnectedState dalla sezione dati.
"mov [ebp+var_4], eax"	Salva il risultato della chiamata a InternetGetConnectedState nella variabile locale.
"cmp [ebp+var_4], 0"	Compara il valore salvato con 0.
"jz short loc_40102B"	Salta a loc_40102B se la connessione a Internet è assente.
"push offset aSuccessInterne"	Mette l'offset della stringa "success: Internet Connection\n" nello stack.
"call sub_40117F"	Chiama la subroutine con l'indirizzo della stringa come argomento.
"add esp, 4"	Libera lo spazio nello stack utilizzato per l'argomento della chiamata alla subroutine.
"mov eax, 1"	Imposta eax a 1 (successo).
"jmp short loc_40103A"	Salta a loc_40103A.
"push offset aError1_1NoInte"	Mette l'offset della stringa "Error 1.1: No internet\n" nello stack.
"call sub_40117F"	Chiama la subroutine con l'indirizzo della stringa come argomento.
"add esp, 4"	Libera lo spazio nello stack utilizzato per l'argomento della chiamata alla subroutine.
"xor eax, eax"	Imposta eax a 0 (errore).
"loc_40102A"	Etichetta di destinazione per il salto condizionale positivo.
"mov esp, ebp"	Ripristina il puntatore allo stack.
"pop ebp"	Ripristina la base del frame precedente.
"retn"	Restituisce il controllo al chiamante.