# Esercizio  5 Settimana  10

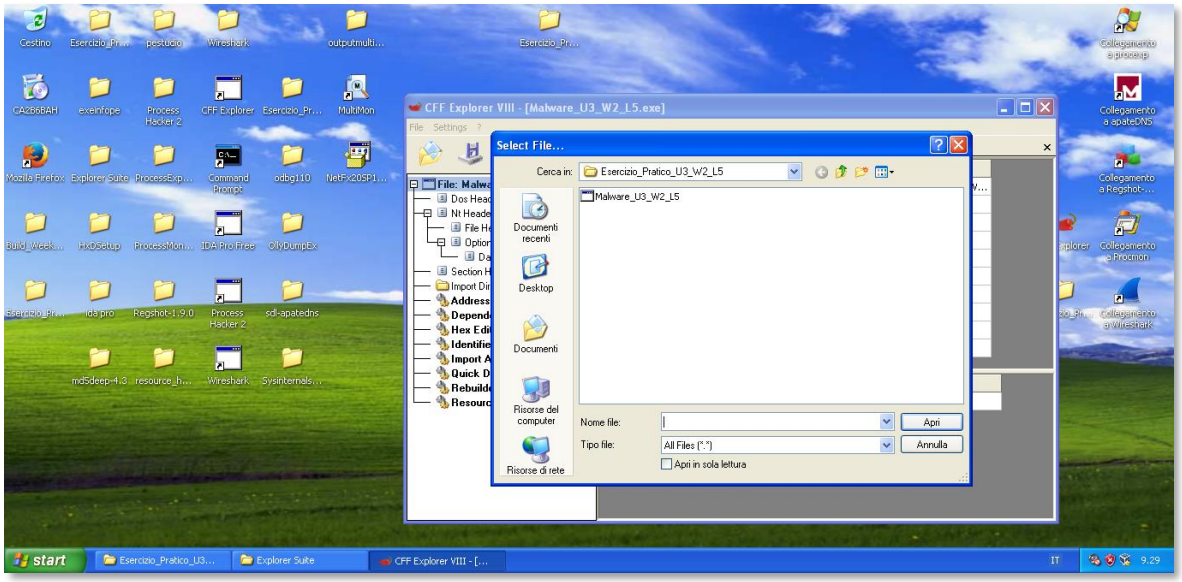
- Analisi statica e dinamica: Un approccio pratico

Con riferimento al file Malware 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:

- Quali librerie vengono importate dal file eseguibile?
- Quali sono le sezioni di cui si compone il file eseguibile del malware?

**Svolgimento:**

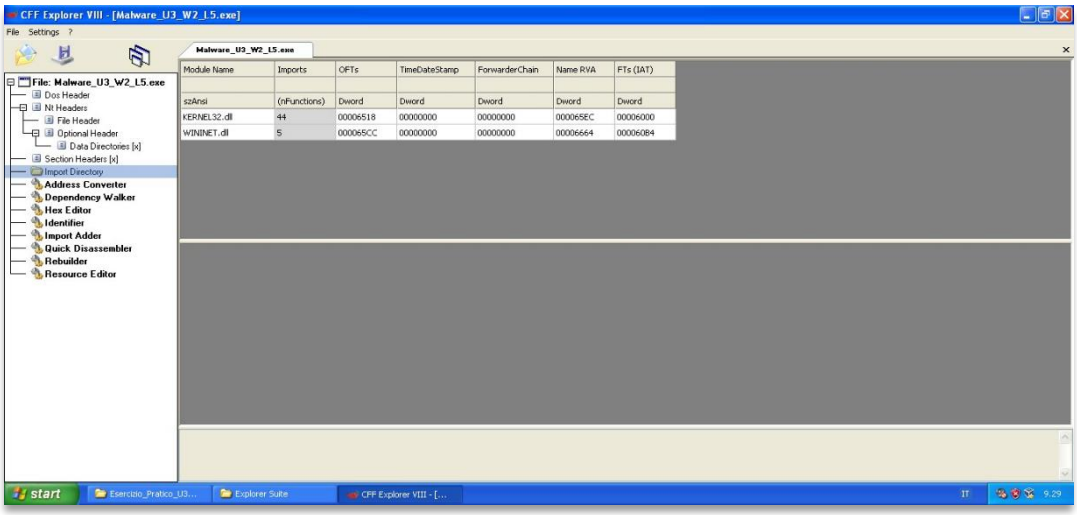
Per lo studio delle librerie e delle sezioni del malware adoperiamo il software **CFF explorer**: CFF Explorer è un programma progettato per esaminare e modificare file eseguibili, in




particolare i file di formato **PE** (Portable Executable) utilizzati nei sistemi operativi Windows. È uno strumento di analisi e editing binario avanzato che consente agli sviluppatori e agli esperti di sicurezza di esaminare la struttura interna dei file eseguibile

Quali librerie sono importate nel file eseguibile?

Lo vediamo dopo aver caricato il file eseguibile su CFF explorer nella sezione **import directory** (Questa sezione è parte integrante della struttura interna del file e contiene informazioni sulle funzioni che il programma utilizza da altre librerie dinamiche durante l'esecuzione.)




Le librerie importate all'interno del file sono due:

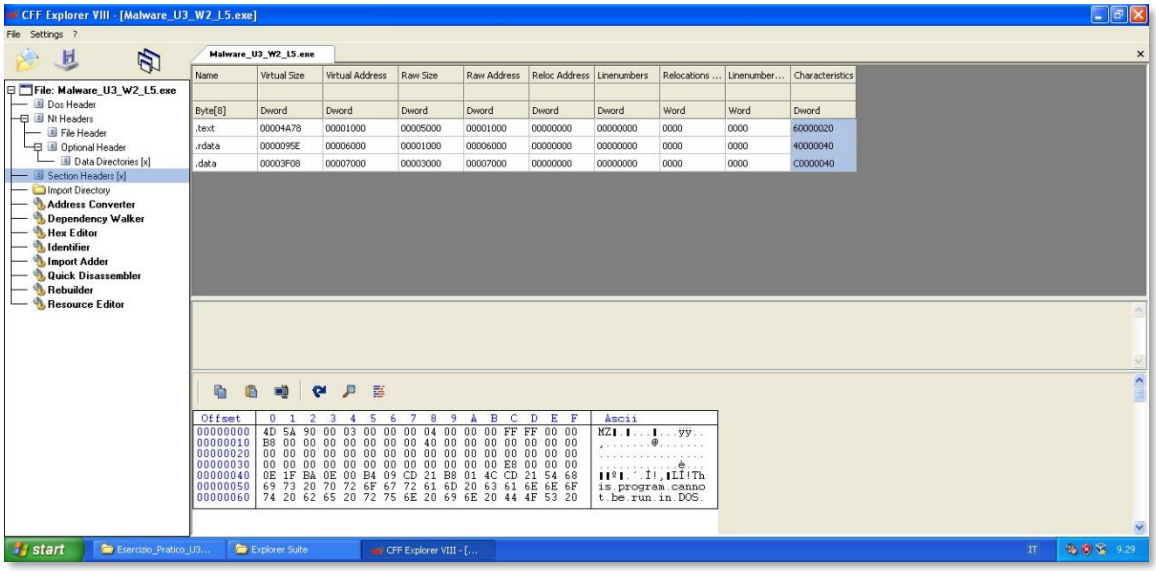
**KERNEL32.dll**

Libreria che contiene le funzioni principali per interagire col sistema operativo, come per esempio la manipolazione di file e la gestione della memoria.

**WININET.dll**

Libreria che contiene le funzioni per implementazione protocolli di rete come HTTP, FTP ed NTP.

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



Lo vediamo attraverso l'opzione **Section headers** che in CFF Explorer mostra informazioni dettagliate sulle varie sezioni di un file eseguibile PE (Portable Executable).

Le sezioni sono:

## .text (sezione del codice):

Raccoglie le istruzioni eseguibili, costituendo il cuore del programma, determinando il comportamento che la CPU seguirà durante l'esecuzione.

## .rdata (sezione dati di sola lettura):

Contiene informazioni critiche sulle librerie e le funzioni importate ed esportate, fornendo le dipendenze esterne essenziali per il corretto funzionamento dell'eseguibile.

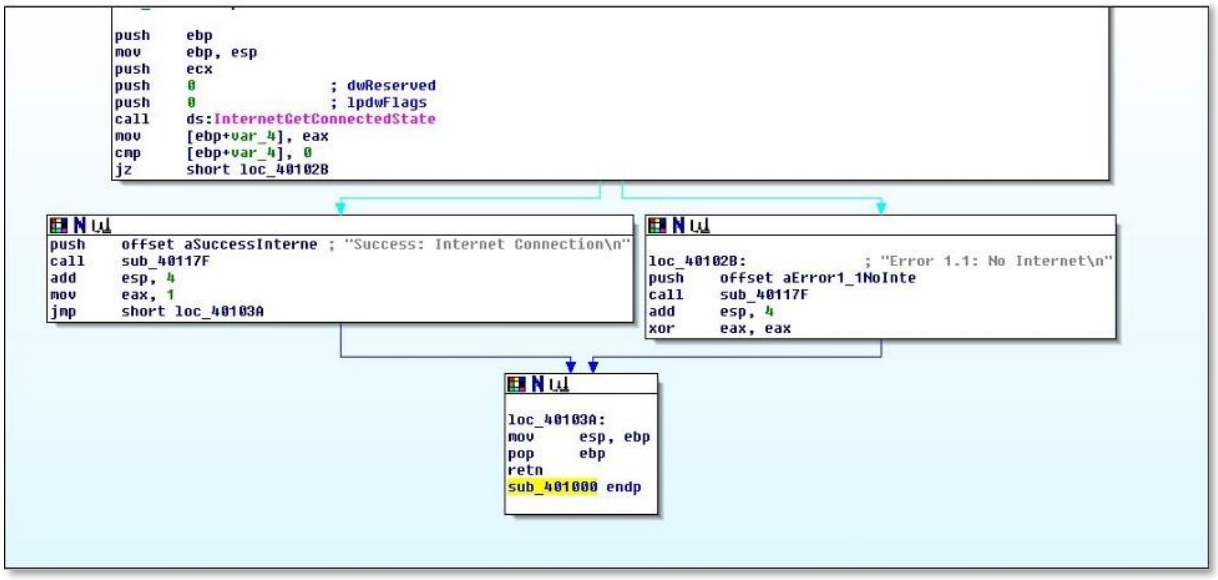
## .data (sezione dei dati):

Ospita dati e variabili globali fondamentali per lo stato del programma, consentendo la memorizzazione di informazioni dinamiche durante l'esecuzione.

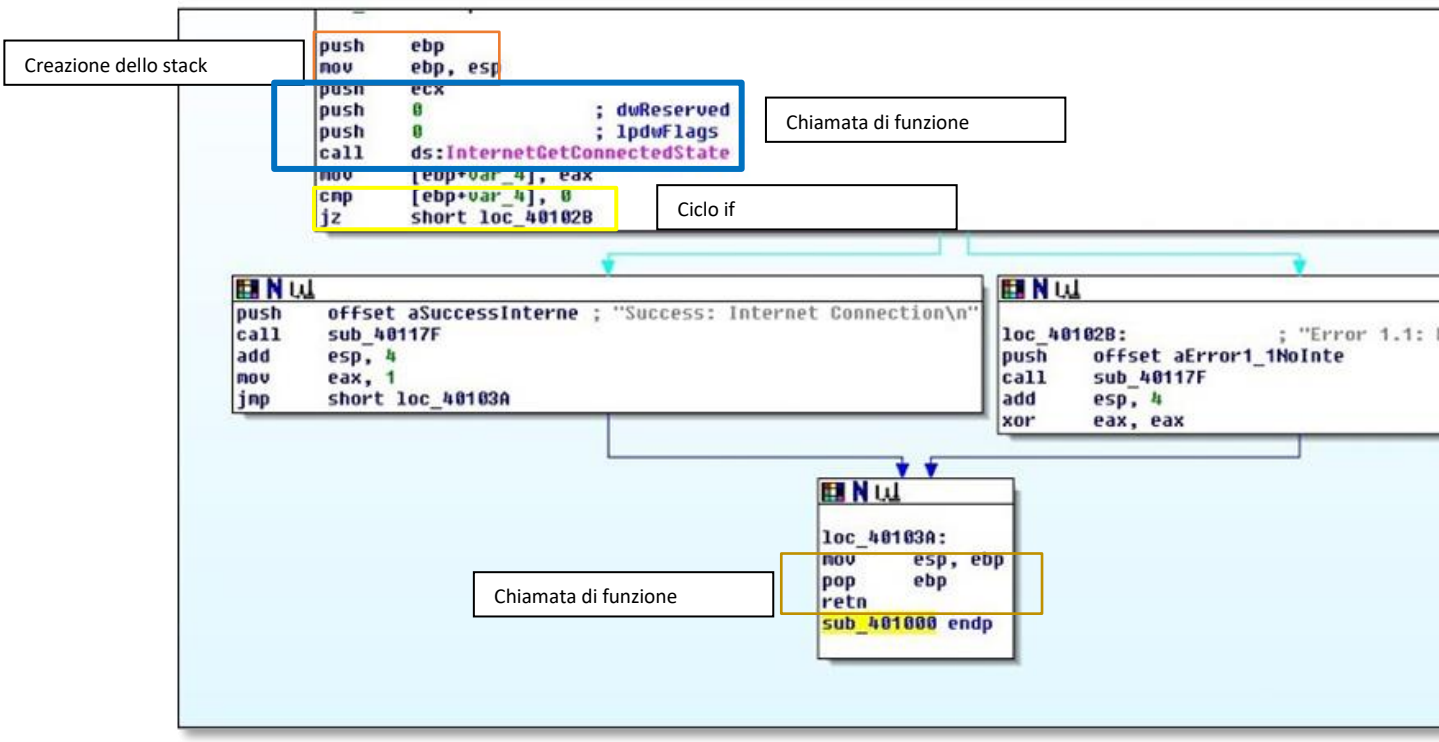
Con riferimento alla seguente figura, risponde ai seguenti quesiti:

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

Ipotizzare il comportamento della funzionalità implementata



## Costrutti noti:




## Ipotizzare il comportamento della funzionalità implementata

. Questo frammento di codice in Assembly ci indica che attraverso la funzione **InternetGetConnectedState,** si determina se su una macchina è presente una connessione internet. Attraverso il **costrutto IF**, avviene un controllo sulla funzione, che a seconda del parametro restituito (uguale a 0/diverso da zero) ci indica a schermo la presenza o meno di una connessione internet sulla macchina target. Come possiamo vedere in figura, se c'è la presenza di una connessione internet, ci restituisce il messaggio '**Success: Internet Connection'**, viceversa ci restituisce '**Error 1.1: No Internet'**

# Parte bonus

## Fare una tabella con spiegato ogni riga del codice Assembly

**; Inizializzazione del frame della funzione**

**push ebp** ; Salva il valore corrente di ebp nello stack

**mov ebp, esp** ; Imposta ebp al valore corrente di esp (crea un nuovo frame)

**push ecx** ; Salva il valore corrente di ecx nello stack

**push 0:** Mette il valore 0 nello stack..

**push 0**: Ancora una volta, mette il valore 0 nello stack.

**ds:InternetGetConnectedState** ; Chiamata a una funzione esterna

**mov [ebp+var_4], eax** ; Memorizza il risultato della chiamata in [ebp+var_4]

**cmp [ebp+var_4], 0** ; Compara [ebp+var_4] con 0

**jz short loc_40102B** ; Salta a loc_40102B se il risultato è zero

; Successo nella connessione a Internet

**push offset aSuccessInterne** ; "Success: Internet Connection\n"

**call sub_40117F** ; Chiamata a una funzione per stampare il messaggio

**add esp, 4** ; Aggiusta lo stack rimuovendo il puntatore alla stringa

**mov eax, 1** ; Imposta eax a 1 (successo)

**jmp short loc_40103A** ; Salta a loc_40103A

; Errore nella connessione a Internet

**loc_40102b:** ; Etichetta per l'errore

**push offset aError1_NoInte** ; "Error 1.1: No Internet\n"

**call sub_40117F** ; Chiamata a una funzione per stampare il messaggio

**add esp, 4** ; Aggiusta lo stack rimuovendo il puntatore alla stringa

**xor eax, eax** ; Resetta eax a 0 (errore)

; Epilogo della funzione

**loc_40103A:**

**mov esp, ebp** ; Ripristina esp al valore di ebp

**pop ebp** ; Ripristina ebp dalla pila

**retn** ; Ritorna dalla funzione

**sub_40100 endp** ; Fine della definizione della funzione