

Progetto S10L5

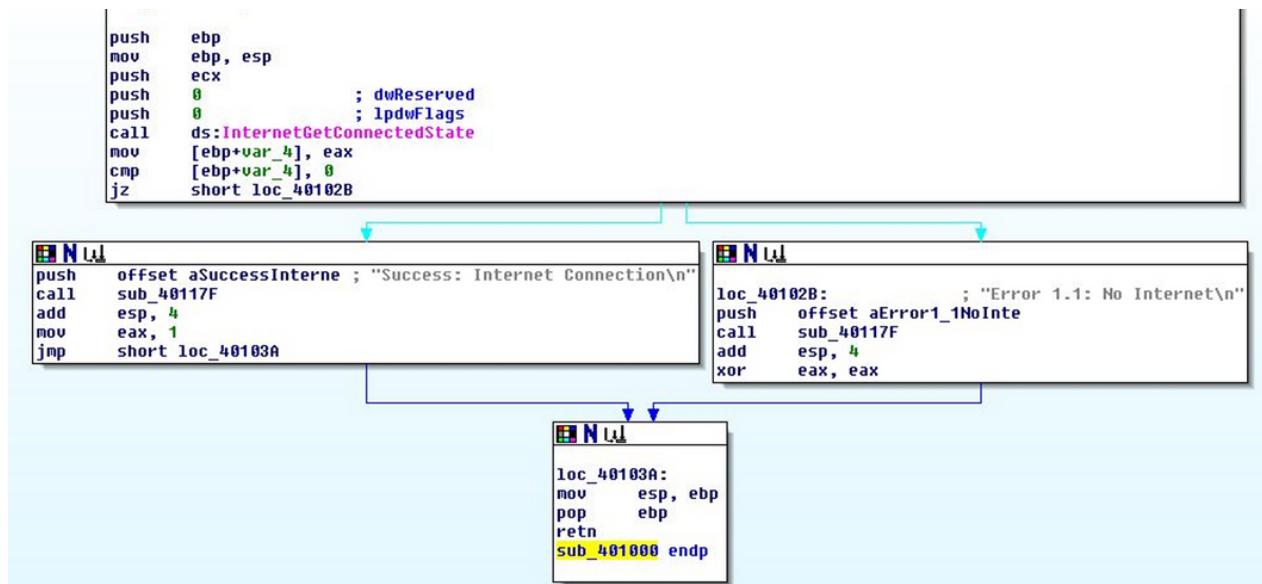
Traccia:

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:

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

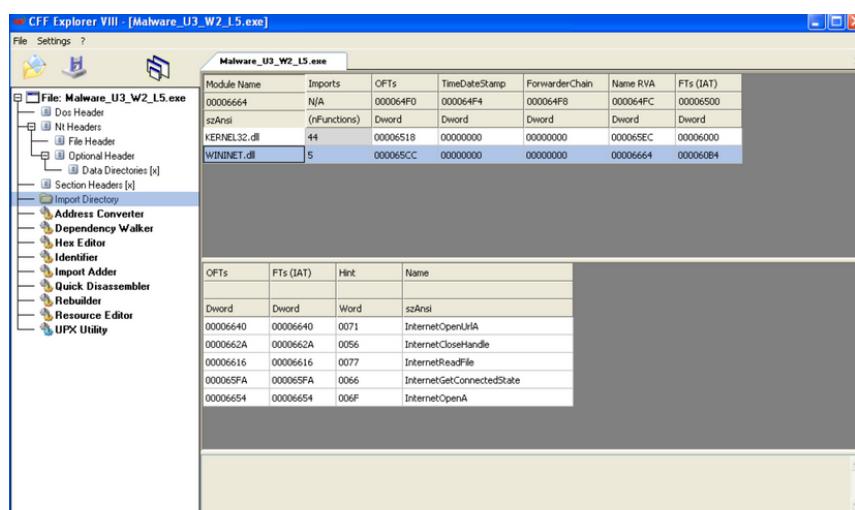
Con riferimento alla figura in slide 3, risponde ai seguenti quesiti:

- Identificare i costrutti noti (creazione dello stack, eventuali cicli, costrutti)
- Ipotizzare il comportamento della funzionalità implementata



L'esercizio ci chiede per prima cosa di effettuare un'analisi senza eseguire il malware (analisi statica basica) da qui andare ad individuare le librerie e le sezioni per poi capire il comportamento del malware; come seconda cosa ci chiede di individuare eventuali costrutti e capire il funzionamento.

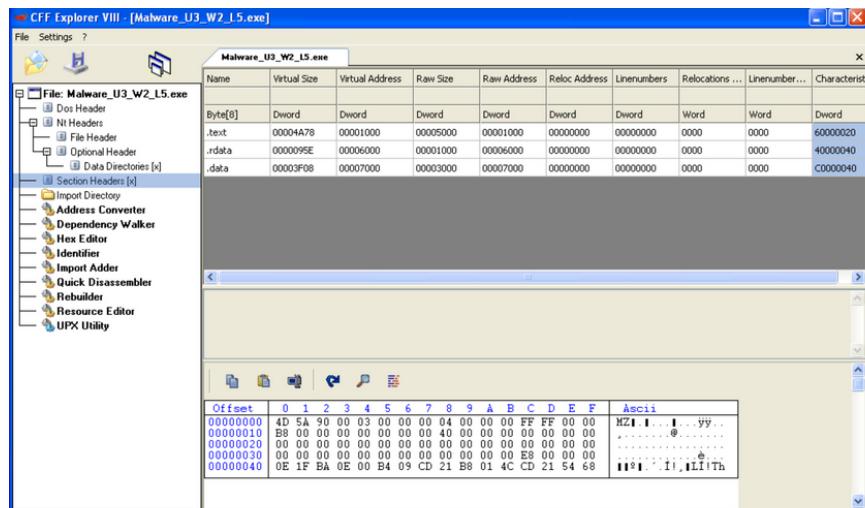
1) Librerie importate



Kernel32.dll: contiene le funzioni principali per interagire con il sistema operativo.

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

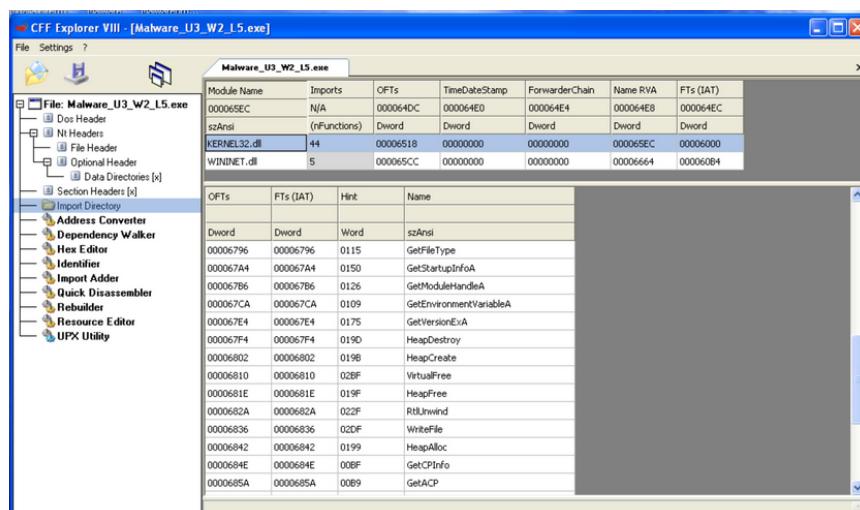
2) Sezioni



.text: contiene le istruzioni (le righe di codice) che la CPU eseguirà una volta che il software sarà avviato.

.rdata: include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile.

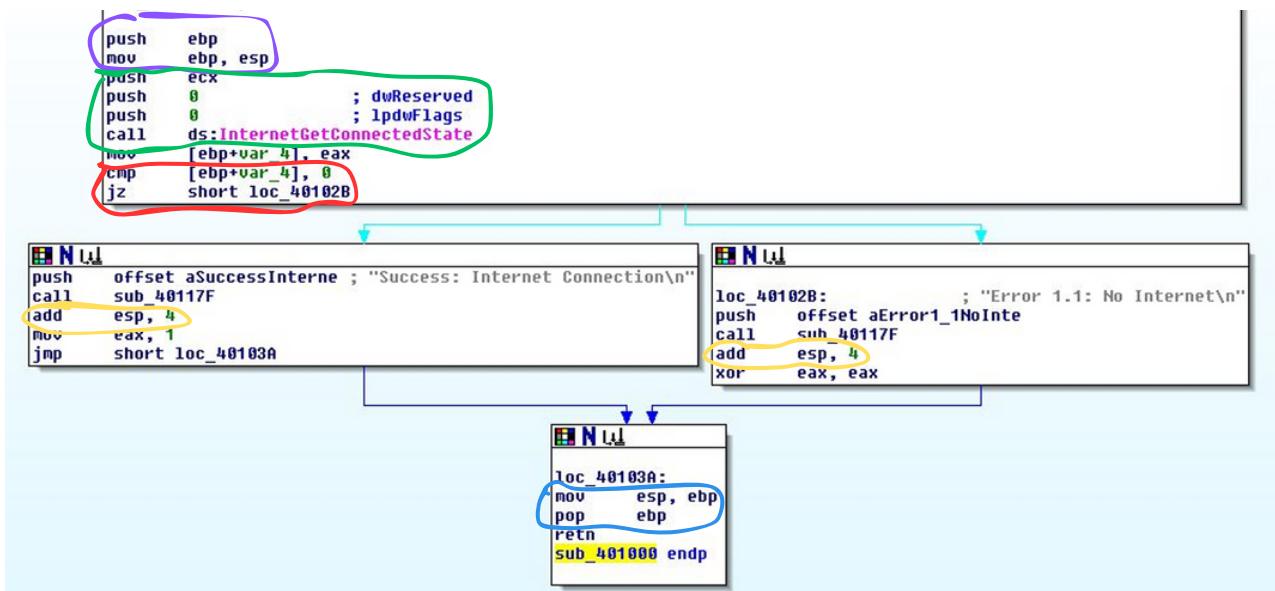
.data: contiene tipicamente i dati / le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma. Una variabile si dice globale quando non è definita all'interno di una funzione, ma è accessibile da qualsiasi funzione all'interno dell'eseguibile



Il malware in questione potrebbe essere un trojan horse lo si può dedurre dal fatto che tra le funzioni della libreria Kernel32.dll vi sono scritte delle azioni riguardanti a un file eseguito in tempo reale dedotto dalla funzione Heap

Heap: viene utilizzato per l'allocazione di memoria dinamicamente durante l'esecuzione di un programma

Trojan horse: è un malware che si nasconde all'interno di un file apparentemente innocuo, come potrebbe essere un eseguibile oppure un documento di office 365, oppure un PDF.



Creazione dello stack



stack è un'area di memoria dove sono memorizzate lo stato di esecuzione del programma, gli alias delle variabili locali di ogni metodo, il loro indirizzo di memoria e il loro valore.

Chiamata della funzione



si verifica quando un programma o un'istruzione richiama l'esecuzione di una funzione.

Ciclo if



consente di eseguire blocchi di codice solo se una determinata condizione è vera.

Rimozione dello stack



Addizione



Dal codice assembly si può capire che il malware sta verificando lo stato della connessione internet e agisce in base al risultato.

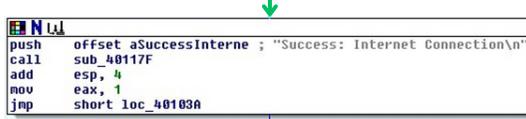


Viene chiamata la funzione InternetGetConnectedState per verificare lo stato della connessione internet.

Il risultato viene memorizzato in [ebp+var_4].

Viene confrontato con zero, e se è uguale a zero, salta a loc_40102B (nel caso di connessione internet assente).

Gestione Successo Connessione Internet:



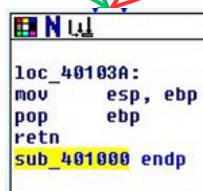
Gestione Errore Connessione Internet:



Se la connessione internet è assente, viene stampato il messaggio "Error 1.1: No Internet". Viene eseguito un XOR di eax con se stesso (azzerato).

Se la connessione internet è riuscita, viene stampato il messaggio "Success: Internet Connection". Viene eseguito un salto incondizionato a loc_40103A dopo aver impostato eax a 1.

Pulizia Stack e Ritorno:



Il blocco di codice termina con la pulizia dello stack (mov esp, ebp e pop ebp), seguito da un'istruzione di ritorno (ret), indicando la fine della funzione.