Claudio De Cicco
13/01/2023

Report

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:

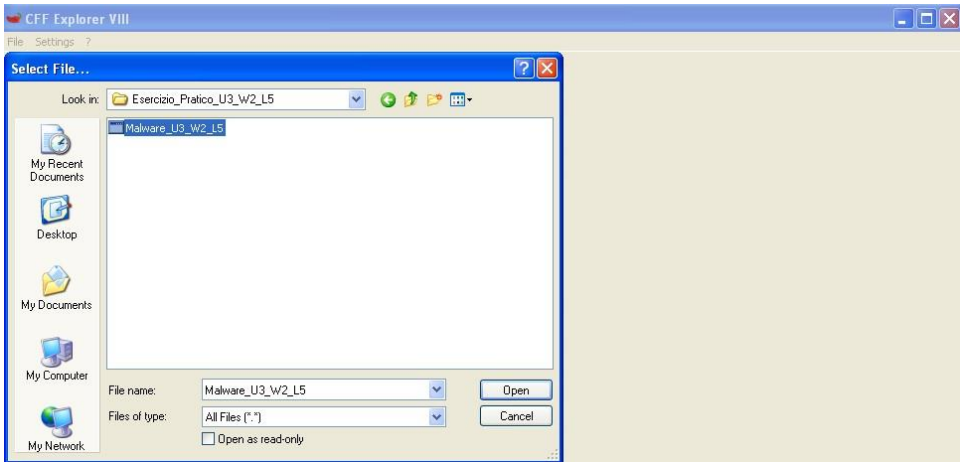
- Quali librerie vengono importate dal file eseguibile?
- Quali sono le sezioni di cui si compone il file eseguibile del malware?

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

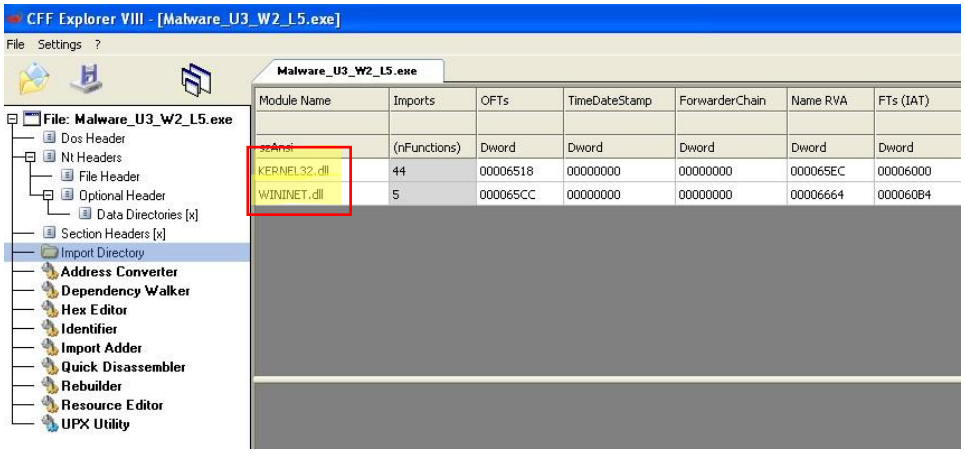
- Identificare i costrutti noti (creazione dello stack, eventuali cicli, costrutti)
- Ipotizzare il comportamento della funzionalità implementata

Punto 1

Andiamo ad eseguire un analisi statica basica del Malware per identificare le librerie importate e le sue sezioni. Abdrò ad utilizzare il tool CFF Explorer. Vado a selezionare il file da esaminare.



Una volta caricato andiamo nella sezione "Import Directory" possiamo vedere quindi le due librerie:
-**Kernel32.dll** , contiene funzioni principali per interagire con il sistema operativo (es. gestione della memoria)
-**Wininet.dll** , contiene funzioni per l'implementazione di protocolli di rete (HTTP,FTP,NTP)

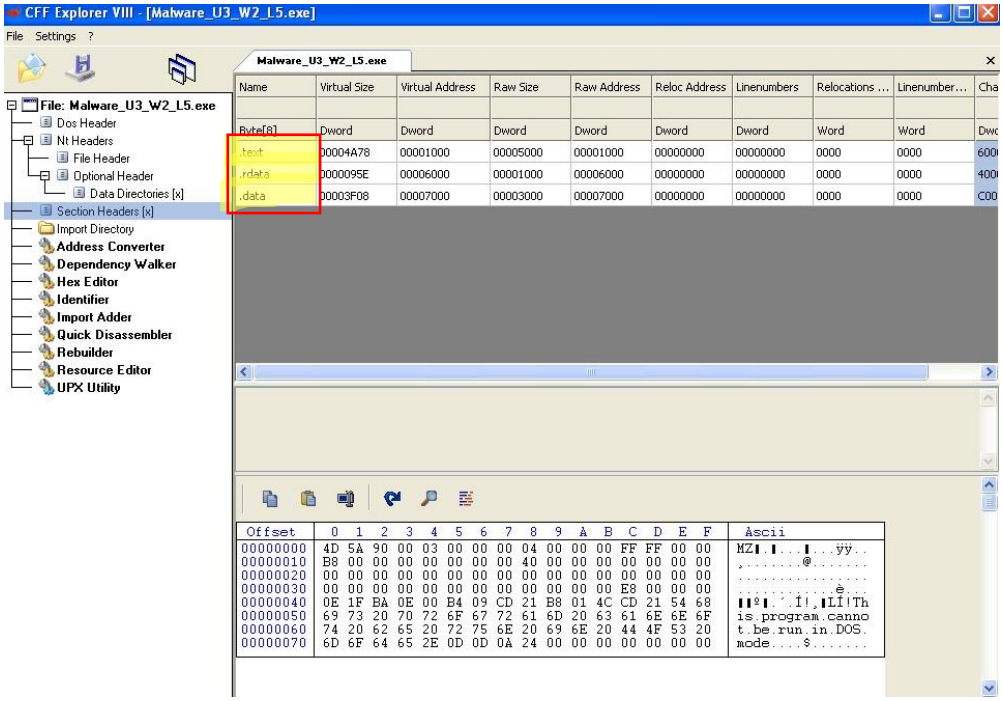


A questo punto ci spostiamo nella sezione “section Headers” per vedere le sezioni di cui è composto il file eseguibile. Possiamo vedere che presenta 3 sezioni:

-**.text ,** contiene le righe di codice che verranno eseguite dalla CPU quando il file sarà avviato.

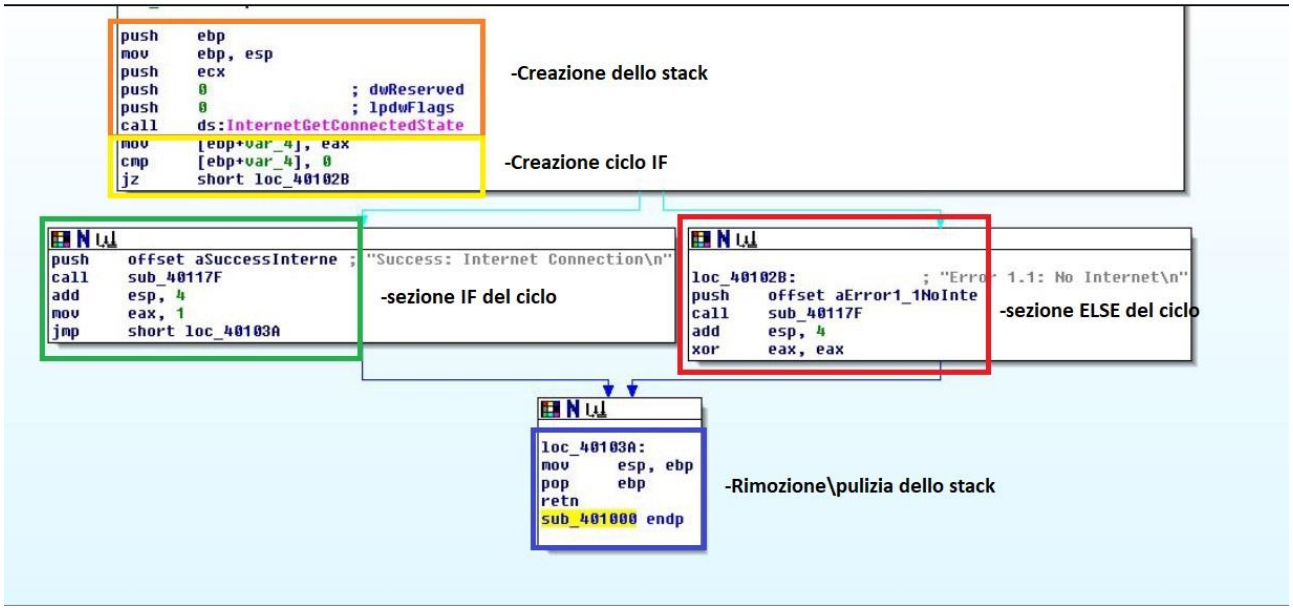
-**.rdata** , contiene le informazioni sulle librerie e le funzioni importate ed esportate dall’eseguibile.

-**.data** , contiene i dati, le variabili globalo del programma eseguibile che devono essere sempre disponibili.



Punto 2

Per prima cosa andiamo ad identificare i costrutti noti.



Ora proviamo ad ipotizzare il comportamento.

-**Creazione dello stack,** lo stack viene creato nelle prime due righe, per quanto riguarda i 3 push successivi vengono creati dei parametri che vengono pushati in cima allo stack e che vengono utilizzati tramite la call.

-**Ciclo IF** , il ciclo viene creato a partite dall’istruzione **cmp** unita all’istruzione **jz** che controllano l’uguaglianza tra due variabili. In questo caso **jz** salta alla locazione di memoria **40102B** se ZF(zero flag) è uguale a 1. Nella parte inn cui la

condizione risulterà vera il programma scrive a schermo **“Success internet connection”** andrà a modificare il valore contenuto in **esp** sommandolo a 4 e copiando il valore 1 nel registro **eax,** dopodiche effettuerà un altro jump alla locazione di memoria **40103A** .
In caso contrario viene visualizzato a schermo **“error 1.1 : no internet”**, anche in questo caso vengono fatte altre operazioni tra cui la somma del valore contenuto in **esp** con 4 e con loperatore logico **xor** inizializza a 0 il registro **eax.**

-Rimozione stack

In conclusione possiamo ipotizzare che questa funzione provi a connettersi ad internet per qualche motivo.