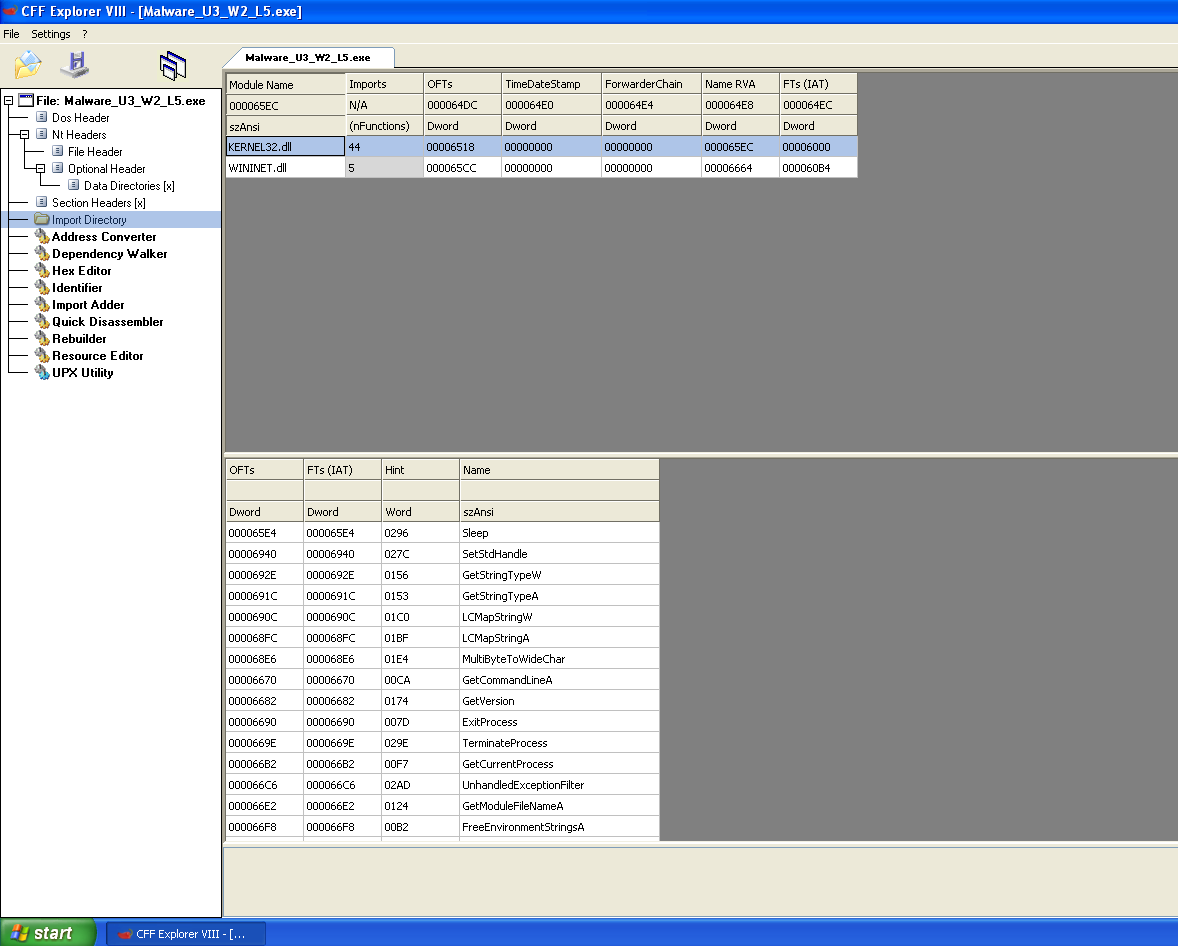
Malware analysis

L’esercizio di oggi richiede di analizzare il malware e la sua struttura. Nella prima parte ho usato un tool **CFF EXPLORER**, usato per controllare le funzioni importate ed esportare da un malware e le sezioni da cui è composto.

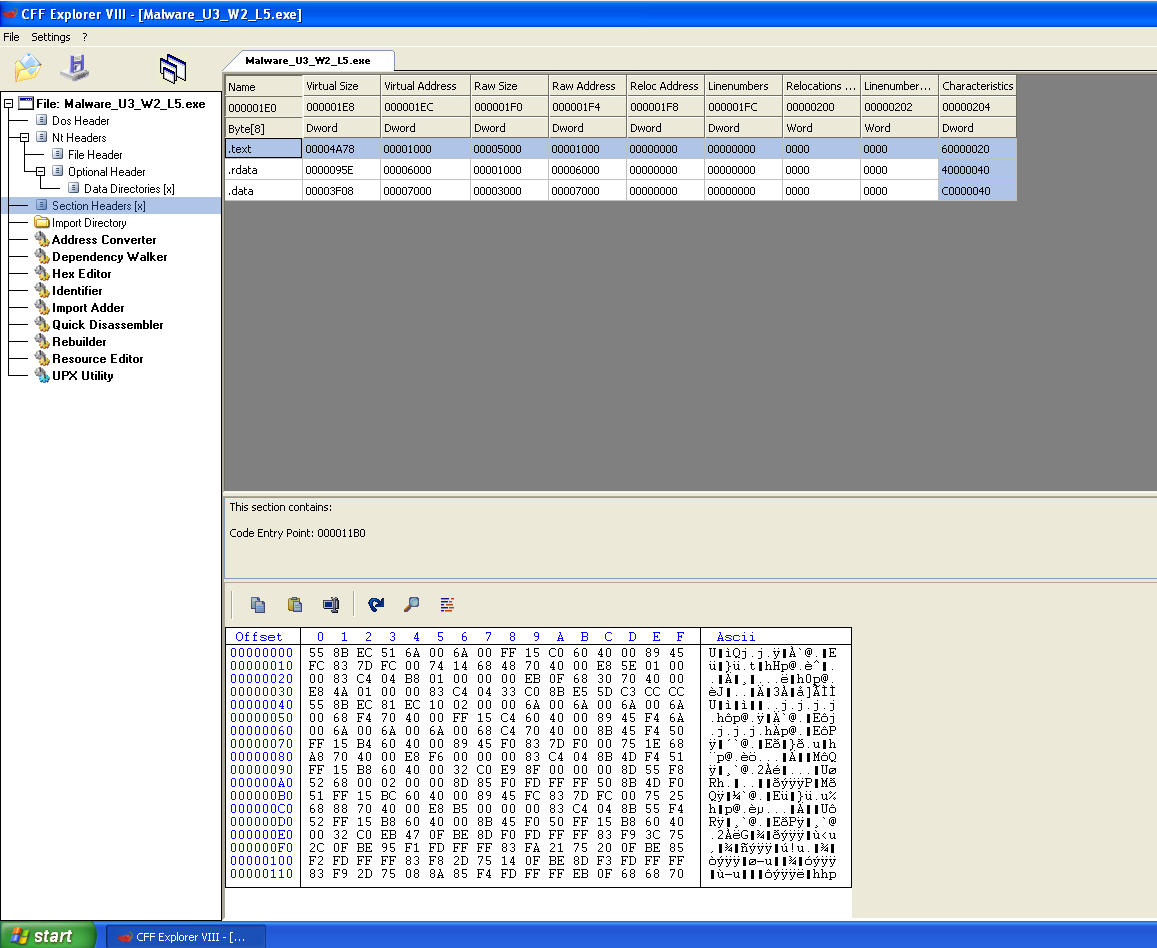
Librerie importate



Le librerie trovate sono:

* **Kernel32.ddl** > libreria che contiene le funzioni principali per interagire col sistema: gestione della memoria, manipolazione dei file.
* **Wininet.ddl** > libreria che contiene le funzioni per implementare alcuni protocolli di rete.

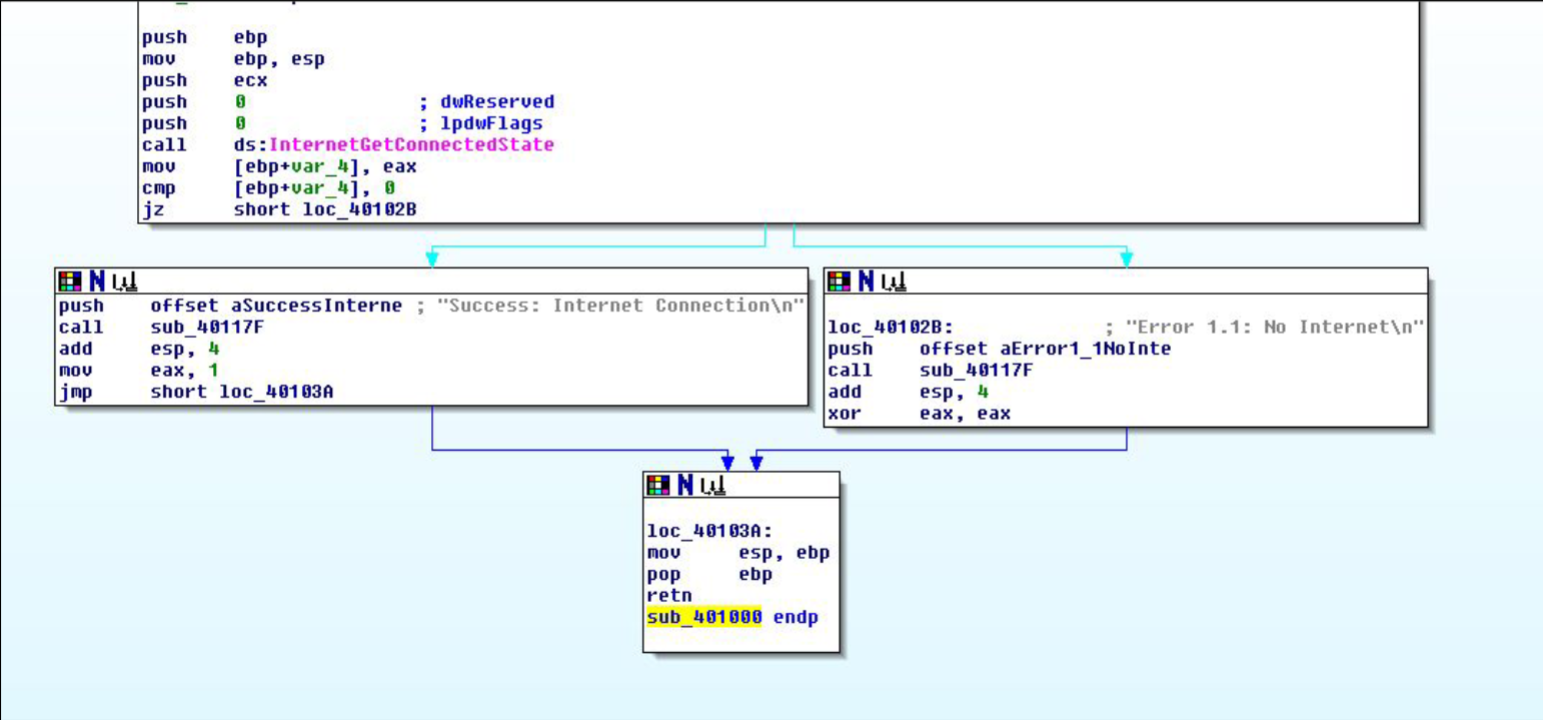
Sezioni del malware



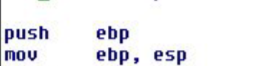
Le sezioni trovate sono:

* **.text** > contiene le istruzioni eseguite dalla CPU una volta avviato il malware.
* **.rdata** > contiene le informazioni sulle librerie e funzioni importate ed esportate.
* **.data** > contiene le variabili globali del malware.

Costrutti del codice

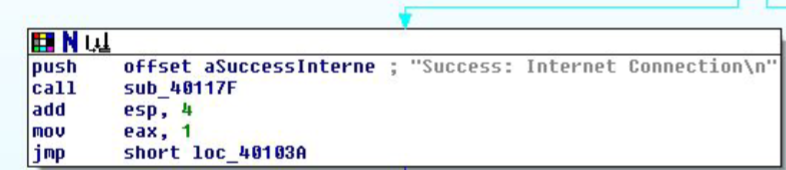


Dato questo codice i costrutti che ho riconosciuto sono:

 Qui c’è la creazione dello stack.

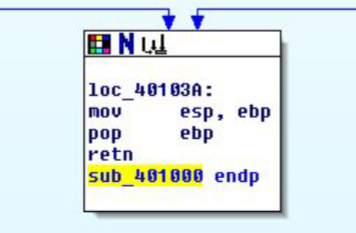
Vengono passati i parametri sullo stack e la chiamata della funzione

 Questo è l’inizio del ciclo if else.

Qui vediamo la prima condizione dell’IF, se la condizione è 1 allora la connessione è attiva.



Mentre se restituisce 0 allora non c’è nessuna connessione.

Questa è l’ultima parte del programma in cui vediamo la pulizia dello stack.

Ipotesi comportamento del malware

Da questo codice si può notare che il malware cerca di capire se la connessione è attiva o meno, quindi si può ipotizzare che il malware richiede una connessione per poter scaricare qualcosa, o creare una backdoor.