PROGETTO SETTIMANA 10

Analizzando il Malware tramite CFF riusciamo a capire che librerie vengono importate e di quali sezioni si compone il file. Prima di parlare delle librerie e delle sezioni bisogna introdurre cos'è e cosa fa CFF Explorer.

CFF EXPLORER

CFF Explorer è un potente strumento software utilizzato principalmente per esplorare e analizzare file eseguibili, come file di programma e librerie di collegamento dinamico (DLL). È noto per la sua capacità di fornire informazioni dettagliate sui file binari, nonché per la sua utilità nel debug e nell'analisi dei processi e delle strutture dei file.

Le funzionalità principali di CFF Explorer includono:

- **Analisi dei file eseguibili: CFF Explorer consente agli sviluppatori di esaminare i dettagli interni dei file eseguibili, come i segmenti di memoria, le intestazioni del file, le tabelle di esportazione e importazione delle funzioni e altro ancora.**

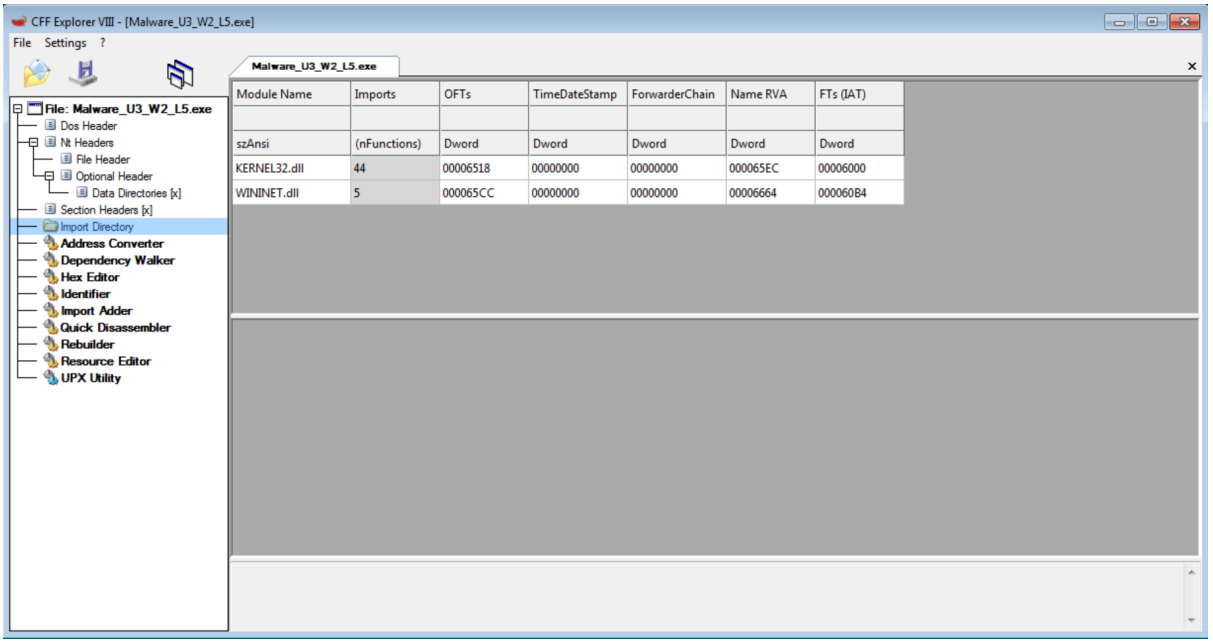
- **Modifica dei file binari: Il tool permette agli utenti di modificare alcune parti dei file eseguibili, come i valori delle variabili, le intestazioni e altre informazioni strutturali.**

- **Debugging: CFF Explorer offre funzionalità di debug per aiutare gli sviluppatori a individuare e risolvere problemi nei loro programmi, consentendo loro di esaminare lo stato delle variabili, i registri e l'esecuzione del codice.**

- **Esportazione di risorse: Gli utenti possono estrarre risorse come icone, immagini e stringhe di testo dai file eseguibili per analisi o utilizzo in altri progetti.**

In sintesi, CFF Explorer è uno strumento versatile per esaminare, analizzare e modificare file eseguibili, che offre funzionalità utili per gli sviluppatori e gli esperti di sicurezza informatica.

1)LIBRERIE



Nell'immagine possiamo notare che il Malware preso in considerazione importa due librerie che sono Kernel32.dll e Wininet.dll, andiamo a visualizzare nello specifico entrambe le librerie.

KERNEL32.dll

Il file "KERNEL.dll" è una libreria di collegamento dinamico (DLL) di sistema di Microsoft Windows. Questa DLL contiene funzioni e risorse fondamentali per il corretto funzionamento del sistema operativo Windows. Il nome "KERNEL" potrebbe essere una storpiatura di "kernel", che è il nucleo del sistema operativo e gestisce le funzioni di base del sistema.

Le DLL come "KERNEL.dll" forniscono un modo per le applicazioni di accedere a funzionalità di sistema comuni in modo condiviso, riducendo la duplicazione del codice e permettendo una gestione più efficiente delle risorse di sistema.

In breve, "KERNEL.dll" è un file critico del sistema operativo Windows che fornisce funzionalità essenziali per il funzionamento del sistema e delle applicazioni su di esso.

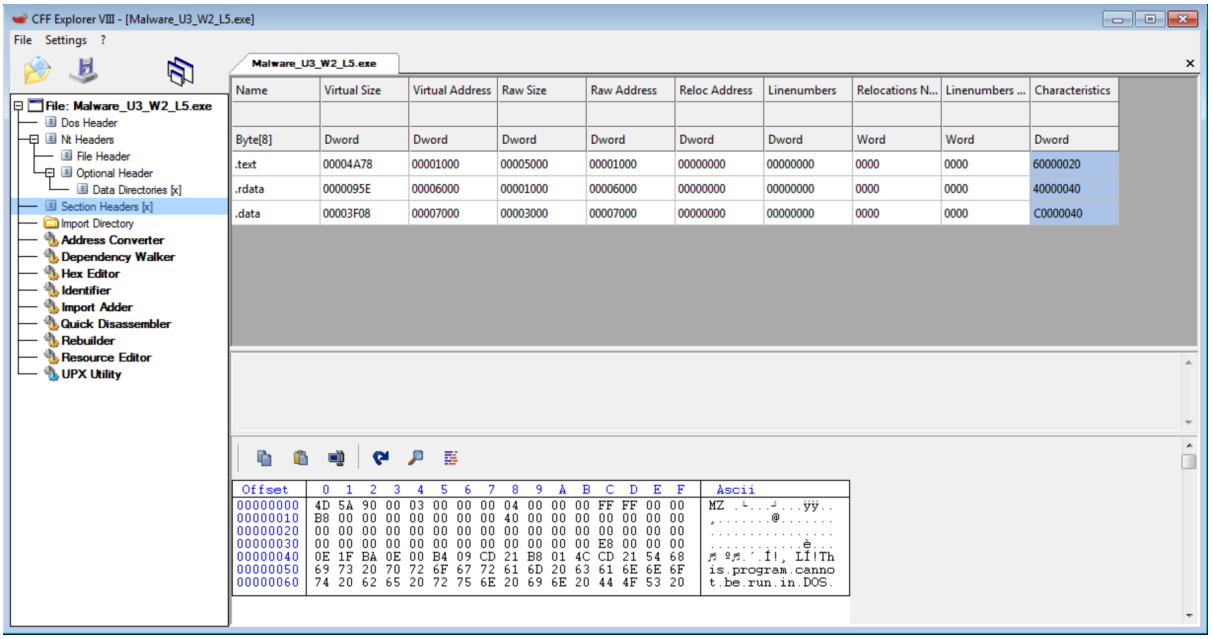
WININET.dll

"WININET.dll" è un'altra libreria di collegamento dinamico (DLL) di sistema di Microsoft Windows. Questa DLL fornisce funzionalità di networking per le applicazioni Windows, consentendo loro di comunicare attraverso Internet utilizzando protocolli come HTTP, HTTPS, FTP e altri.

Le funzioni contenute in "WININET.dll" consentono alle applicazioni di eseguire operazioni di rete, come il download e l'upload di file da server remoti, la navigazione web, l'invio di richieste HTTP e molto altro ancora. Questa libreria è ampiamente utilizzata da molte applicazioni Windows che richiedono accesso a risorse di rete.

In breve, "WININET.dll" è una libreria fondamentale per il networking su Windows, che consente alle applicazioni di comunicare attraverso Internet e di accedere alle risorse di rete.

2)SEZIONI



Come osserviamo nello screenshot le sezioni presenti nel Malware sono tre, ovvero .text .rdata e .data, andiamo a vedere nello specifico cosa fanno.

.text

In breve, la sezione .text è la parte di un file eseguibile che contiene il codice eseguibile del programma, ed è fondamentale per il funzionamento e l'esecuzione delle istruzioni da parte del processore.

.rdata

La sezione .rdata, all'interno di un file eseguibile o di una libreria condivisa (DLL), contiene principalmente dati di sola lettura. Questi dati sono generalmente costanti o informazioni che non vengono modificate durante l'esecuzione del programma.

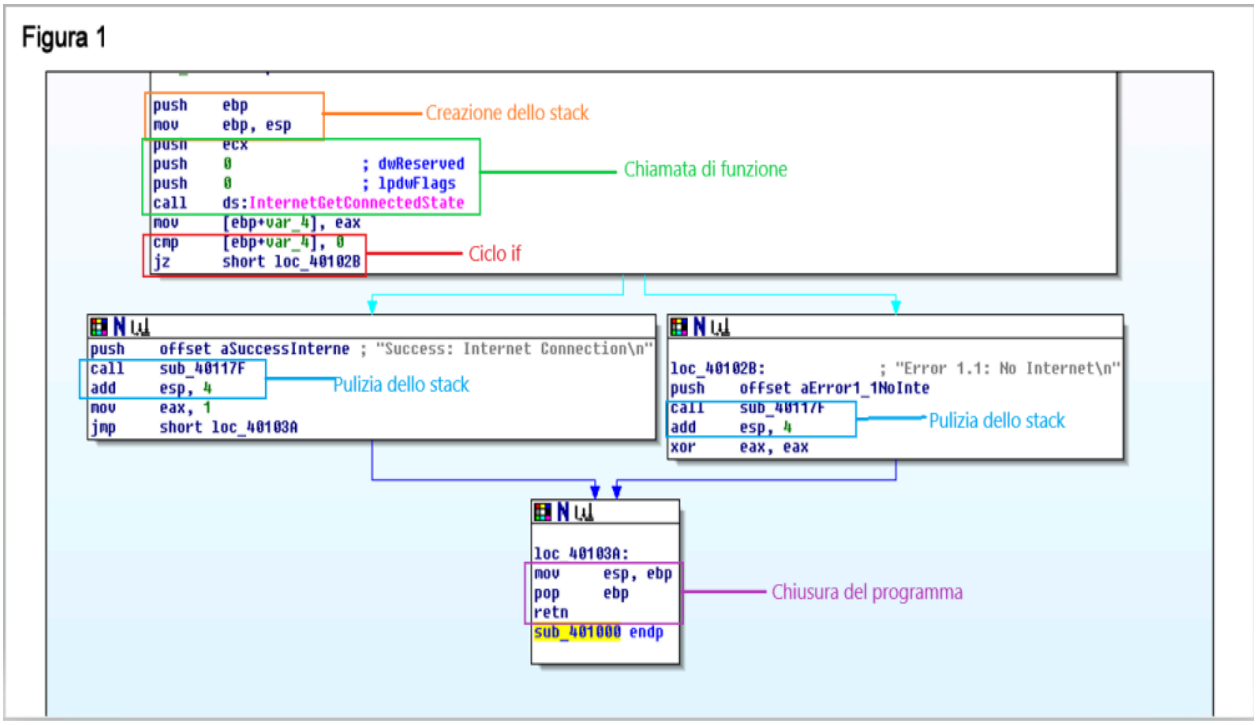
.data

La sezione .data, all'interno di un file eseguibile o di una libreria condivisa (DLL), contiene variabili globali e dati inizializzati che possono essere modificati durante l'esecuzione del programma.

3)COSTRUTTI

Per identificare i costrutti noti nel codice assembly assegnato sono andato ad illustrate graficamente modificando l'immagine assegnatami.

Figura 1



4)COMPORTAMENTO FUNZIONALITÀ

La funzionalità implementata nel codice assembly è "ds:InternetGetConnectedState".

La funzione InternetGetConnectedState è un'API di Windows utilizzata per determinare se il computer è attualmente connesso a Internet e, in caso affermativo, il tipo di connessione attiva. Questa funzione restituisce un valore booleano che indica se c'è una connessione attiva e, se sì, fornisce informazioni aggiuntive sul tipo di connessione, come una connessione modem, LAN o via cavo. È utile per le applicazioni che devono adattare il loro comportamento in base alla disponibilità di una connessione Internet.

5)BONUS

Codice assembly	Descrizione
push ebp	Salva il valore del registro di base (EBP) nello stack.
mov ebp, esp	Inizializza il registro di base (EBP) con il valore dello stack corrente.
push ecx	Salva il valore del registro ECX nello stack.
push 0	Mette nello stack il valore 0, che rappresenta il parametro dwReserved per la funzione `InternetGetConnectedState`.
push 0	Mette nello stack il valore 0, che rappresenta il parametro lpdwFlags per la funzione `InternetGetConnectedState`.
call ds:InternetGetConnectedState	Chiama la funzione `InternetGetConnectedState` per verificare lo stato della connessione a Internet.
mov [ebp+var_4], eax	Salva il risultato della chiamata alla funzione in una variabile locale.
cmp [ebp+var_4], 0	Confronta il valore salvato con 0 per verificare se la chiamata alla funzione ha avuto successo.
jz short loc_40102B	Salta a loc_40102B se il risultato è 0, cioè se non c'è connessione Internet.
push offset asuccessInterne	Mette nello stack l'indirizzo della stringa "Succes Internet Connection\n".
call sub_40105F	Chiama una subroutine per gestire il caso di successo della connessione Internet.
add esp, 4	Libera lo stack da eventuali argomenti passati alla subroutine.
mov eax, 1	Imposta il registro EAX a 1, indicando un successo generale.
jmp short loc_40103A	Salta a loc_40103A per uscire dalla funzione.

push offset aError1_1NoInte	Mette nello stack l'indirizzo della stringa "Error 1.1: No Internet\n".
call sub_40117F	Chiama una subroutine per gestire il caso in cui non c'è connessione a Internet.
add esp, 4	Libera lo stack da eventuali argomenti passati alla subroutine.
mov esp, ebp	Ripristina lo stack pointer allo stato precedente all'inizio della funzione.
pop ebp	Ripristina il valore del registro di base (EBP) dallo stack.
retn	Restituisce il controllo al chiamante.