Progetto S10/L5

Traccia 1) :

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:

1) Quali librerie vengono importate dal file eseguibile?

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

Strumento utilizzato :

- CFF Explorer

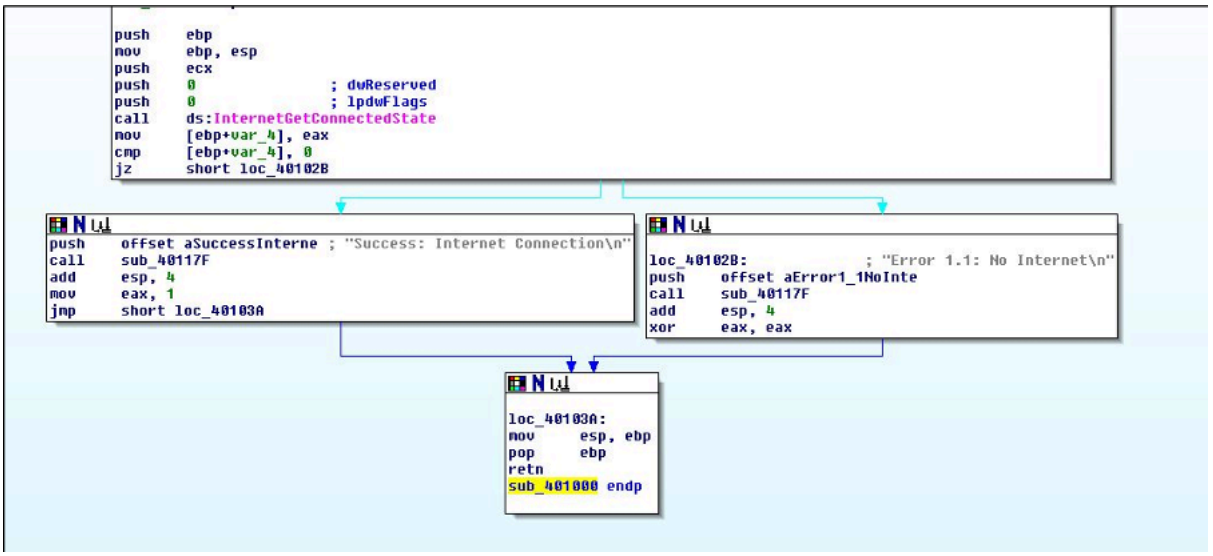
Traccia 2) :

Con riferimento alla figura 1, risponde ai seguenti quesiti:

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

2) Ipotizzare il comportamento della funzionalità implementata

figura 1



1 CFF Explorer

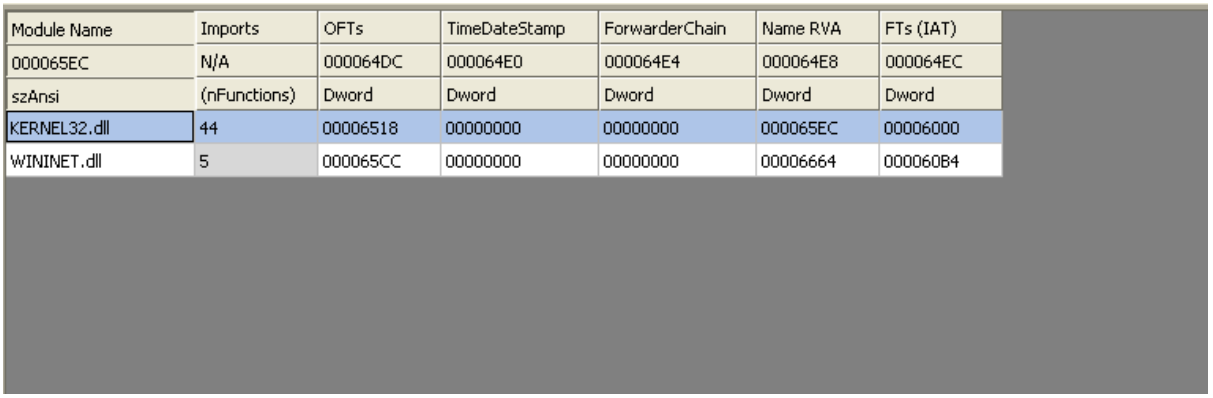
CFF Explorer è uno strumento software utilizzato principalmente per analizzare e modificare file eseguibili di Windows, come file eseguibili (.exe), file di librerie di collegamento dinamico (.dll) e file di sistema (.sys).
Questo strumento è ampiamente utilizzato da sviluppatori software, analisti di sicurezza informatica e ricercatori per esaminare e comprendere il funzionamento interno dei file eseguibili, identificare potenziali vulnerabilità di sicurezza e effettuare reverse engineering su software.

1) Quali librerie vengono importate dal file eseguibile?

Aprendo il fine Malware_U3_W2_L5 con il programma CFF Explorer e scegliendo <<Import directory>>, possiamo notare che il file contiene 2 libreri :

- KERNEL32.DLL
- WININET.DDL

KERNEL32.DLL fornisce funzionalità di sistema di base, mentre WININET.DLL fornisce funzionalità di comunicazione su Internet, entrambe essenziali per il funzionamento di molte applicazioni Windows.



Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000065EC	N/A	000064DC	000064E0	000064E4	000064E8	000064EC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

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

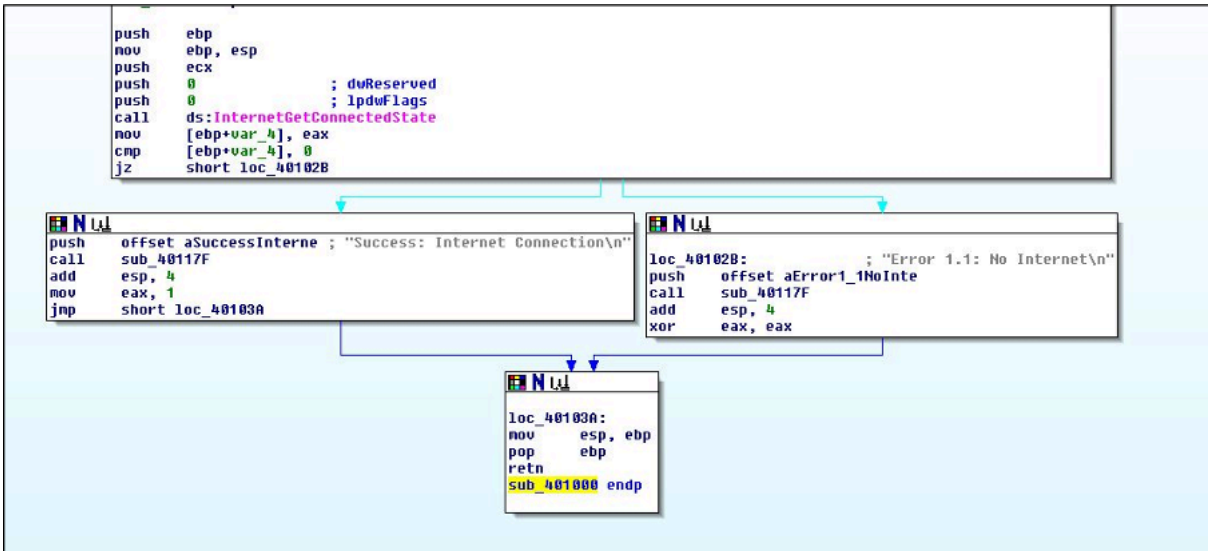
Per trovare le sezioni di cui si compone il file dobbiamo spostarci nella sezione <<section headers>> e vediamo che ci sono 3 sezioni presenti che sono :

- .test
- .rdata
- .data

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	Linenumber...	Characteristics
00000230	00000238	0000023C	00000240	00000244	00000248	0000024C	00000250	00000252	00000254
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00004A78	00001000	00005000	00001000	00000000	00000000	0000	0000	60000020
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000	0000	0000	40000040
.data	00003F08	00007000	00003000	00007000	00000000	00000000	0000	0000	C0000040

2 Assembly

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



Troviamo tre blocchi di costrutti :

Primo blocco

- push ebp
- mov ebp ,esp

Queste due istruzioni iniziano la procedura di configurazione del frame di stack per la funzione, salvando il valore corrente di `ebp` nello stack e quindi impostando `ebp` in modo che punti al nuovo frame di stack.

Secondo blocco

- cmp [ebp+var_4] , 0
- jz short loc_40102B

Queste istruzioni stanno controllando se il valore della variabile locale `var_4` è uguale a zero. Se lo è, il programma salta a un'istruzione specificata (`loc_40102B`), altrimenti continua l'esecuzione dalla linea successiva.

Terzo blocco

- mov esp, ebp
- pop ebp

Queste istruzioni vengono utilizzate per ripristinare lo stack al suo stato precedente all'esecuzione della funzione e per ripristinare il registro base `ebp` al suo valore originale, così da poter tornare al chiamante.