

Analisi Malware e Linguaggio Assembly

Analisi malware

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?

Linguaggio Assembly

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

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

Analisi Malware

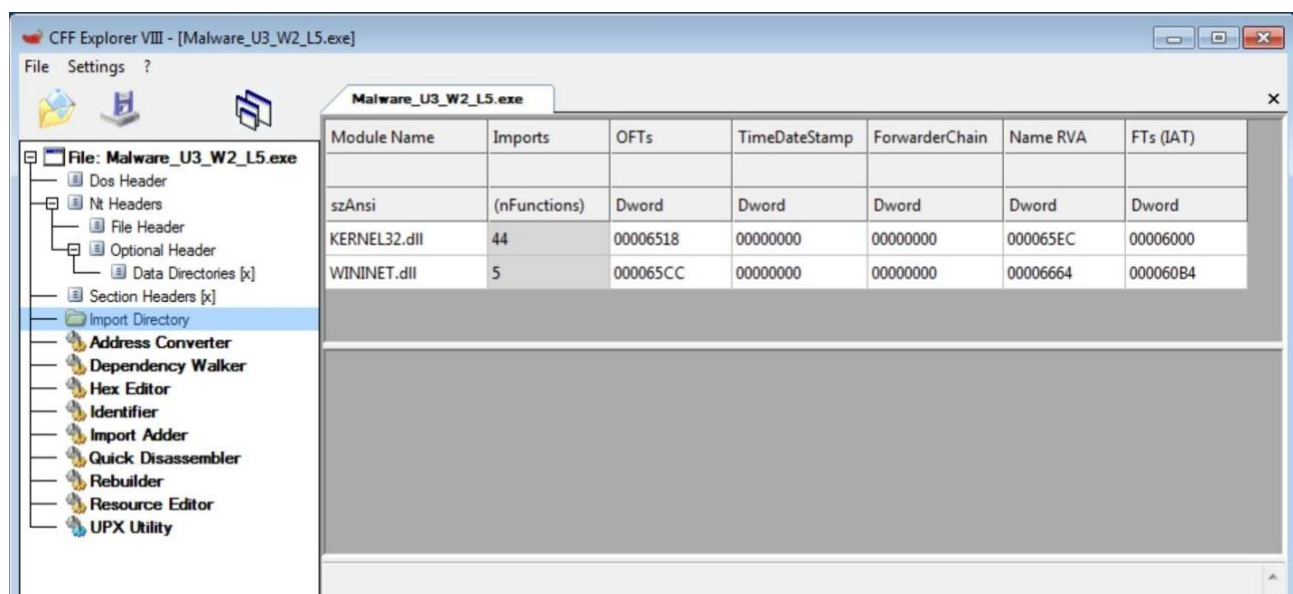
Con riferimento al file Malware_U3_W2_L5 presente all'interno della cartella «Esercizio_Pratico_U3_W2_L5» sul desktop della macchina virtuale dedicata per l'analisi dei malware, rispondere ai seguenti quesiti:

- Quali librerie vengono importate dal file eseguibile?

Il primo obiettivo delle due tasks richiede l'analisi statica di un malware identificato nella cartella "Malware_U3_W2_L5" con il nome "Esercizio_Pratico_U3_W2_L5".

Per poter eseguire il compito ci avvaleremo di un potente tool presente sulla nostra macchina virtuale "CFF Explorer" che analizza le caratteristiche del malware senza la necessità di avviarlo.

Come prima azione individueremo le librerie importate dal file eseguibile accedendo alla cartella "import directory".

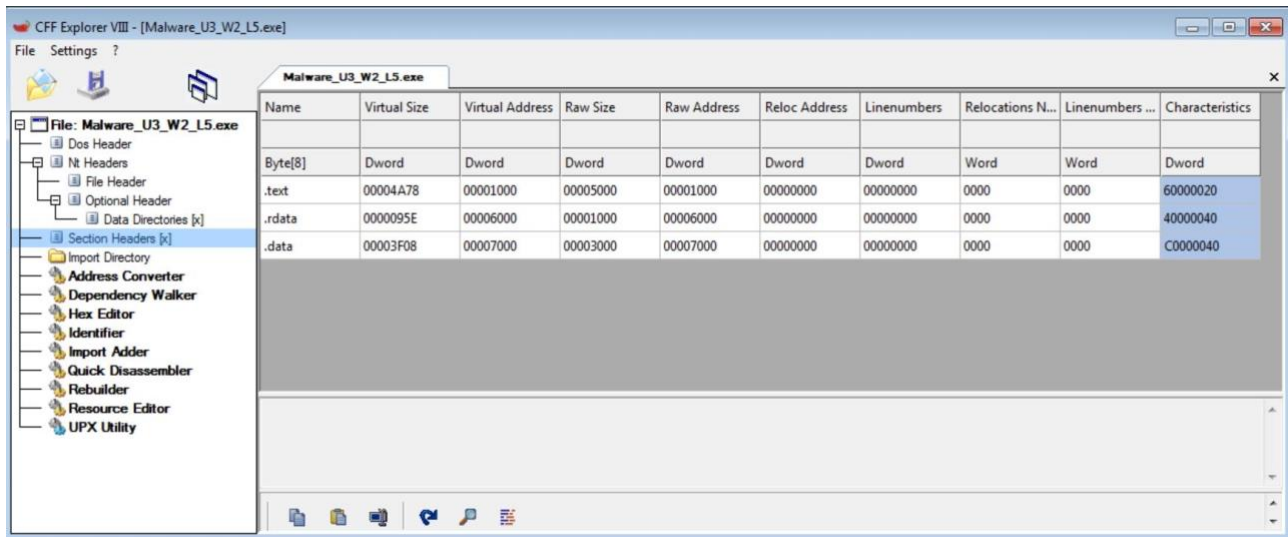


Come possiamo notare dalla figura riportata le librerie importate sono due:

- KERNEL32.dll
- WININET.dll

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

Procedendo per la sezione “Section Headers” possiamo già rispondere al seguente quesito.



Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
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

Come possiamo vedere dalla figura le sezioni di cui si compone il file eseguibile del malware sono tre:

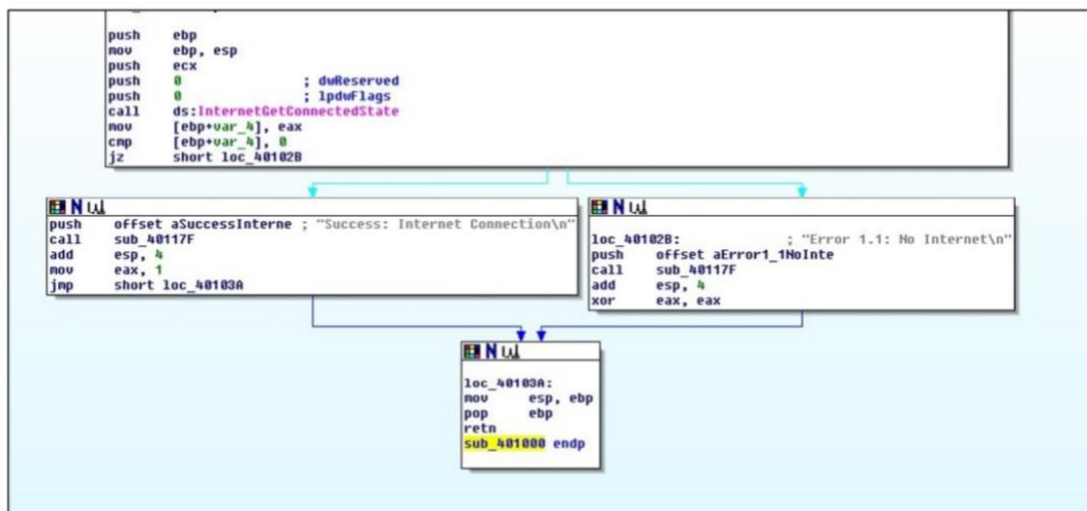
- .text
- .rdata
- .data

Linguaggio Assembly:

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

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

Figura 1



3

Dando un primo sguardo al codice in Assembly possiamo notare come la prima parte e l'ultima aprono e chiudono lo stack.

In particolare "push ebp" e "mov ebp,esp" ne permettono l'apertura

Mentre "mov esp, ebp" e "pop ebp" ne consente la chiusura.

Osservando con più attenzione possiamo dedurre che ci sia un ciclo if alla fine del primo rettangolo, più nello specifico le ultime due righe:

"Cmp [ebp+var_4], 0"

"jz short loc_40102B"

Il secondo compito richiede di ipotizzare il comportamento della funzionalità implementata, considerata l'esistenza del costrutto if e del fatto che il programma vuole valutare se c'è la connessione ad internet "call ds:InternetGetConnectedState" possiamo avanzare l'ipotesi che il codice si comporterà in maniera differente a seconda del fatto che sia connesso o meno ad internet.

