

S10L5

EPICODE PROJECT



DATE :
2 Agosto 2024

PRESENTED BY :
sarah Ortiz

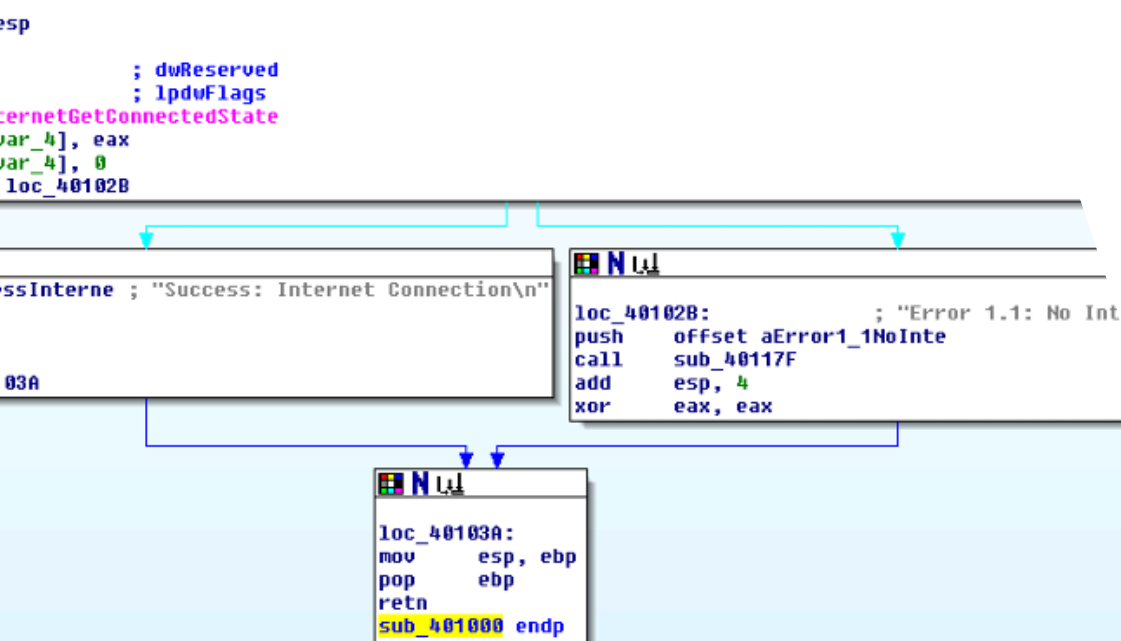
Traccia

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:

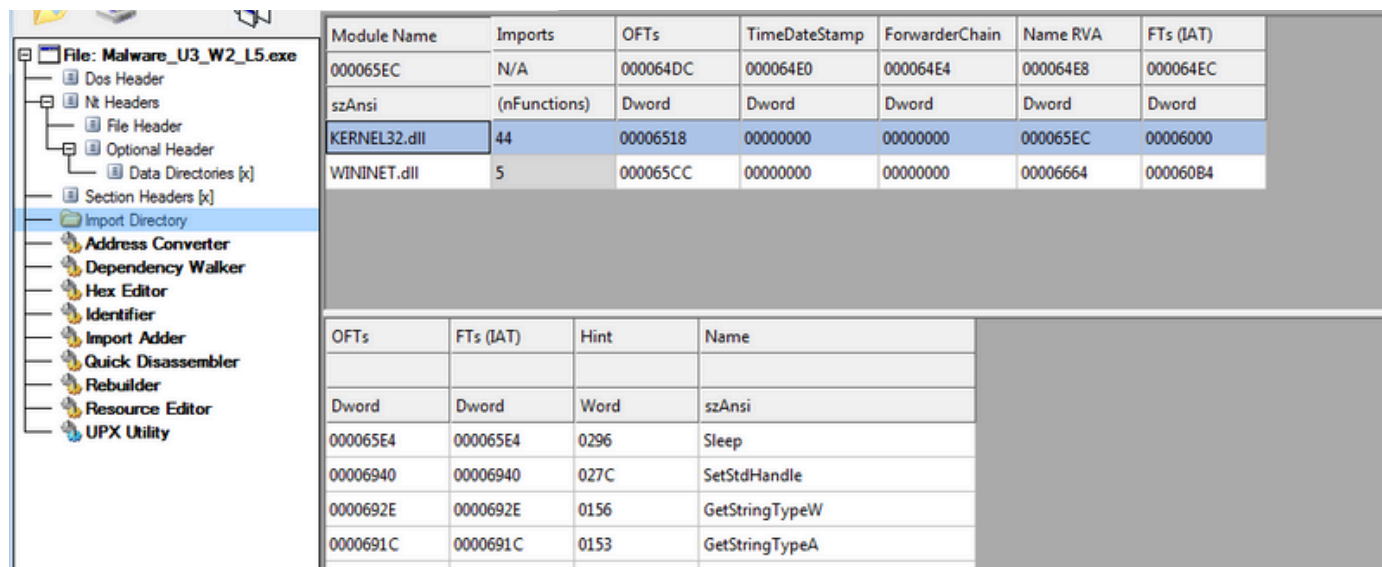
1. Quali librerie vengono importate dal file eseguibile? Fare anche una descrizione
2. Quali sono le sezioni di cui si compone il file eseguibile del malware? Fare anche una descrizione.

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

3. Identificare i costrutti noti (creazione dello stack, eventuali cicli, ...)
4. Ipotesizzare il comportamento della funzionalità implementata (altri costrutti)
5. Fare una tabella per spiegare il significato delle singole righe di codice



Punto 1



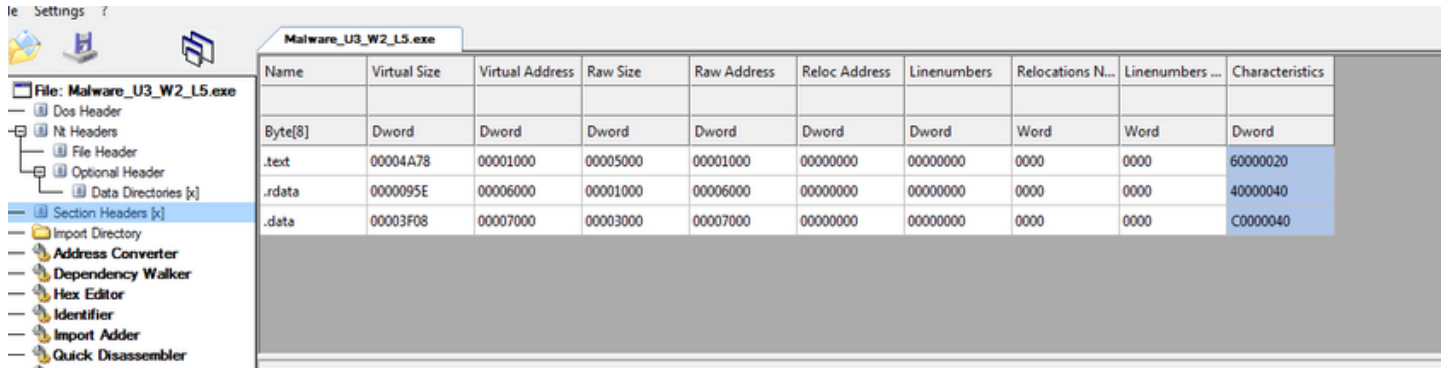
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000065EC	N/A	000064DC	000064E0	000064E4	000064E8	000064EC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000065E4	000065E4	0296	Sleep
00006940	00006940	027C	SetStdHandle
0000692E	0000692E	0156	GetStringTypeW
0000691C	0000691C	0153	GetStringTypeA

Le librerie che troviamo all'interno del Maware_U3_W2_L5, tutto grazie all'applicazione CFF Explorer, sono le seguenti:

- **karnel32.dll**: è un file DLL di Windows, acronimo di Dynamic Link Library. è una libreria che gestisce operazioni di basso livello relative al sistema operativo, come la gestione della memoria, la gestione dei file e le operazioni di I/O.
- **wininet.dll**: anch'esso è un file DLL di Windows. è una libreria che fornisce funzioni per l'accesso a Internet, compreso HTTP e FTP. Se la cancelli avrai problemi a connetterti ad internet.

Punto 2



The screenshot shows a software interface for analyzing a Portable Executable (PE) file. The file name is 'Malware_U3_W2_L5.exe'. The 'Section Headers [x]' tab is selected in the left sidebar. The main window displays a table of section headers.

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

I file di tipo PE iniziano con un header che contiene informazioni rispetto al codice da eseguire e all'indirizzo in cui è contenuta la porzione eseguibile, al tipo di applicazione, alle librerie di funzioni da caricare, e così via. Un file eseguibile è composto da diverse sezioni, ognuna con una funzione specifica. Qui presenti vi sono:

- **.text:** contiene il codice eseguibile, è la prima sezione esistente e costituisce il punto di ingresso dell'applicazione, ovvero la prima istruzione che viene eseguita quando il programma viene caricato in memoria.
- **.rdata:** contiene informazioni sull'import e l'export e può contenere anche altri dati read only;
- **.data:** contiene dati globali accessibili da ogni punto del programma

Punto 3

```
push ebp
mov ebp, esp
sub esp, 4
push ecx
push 0
push 0
call ds:InternetGetConnectedState - chiamata di funzione
mov [ebp+var_4], eax
cmp [ebp+var_4], 0
jz short loc_401028 } condizionale e salto
push offset aSuccessInterne
call sub_40117F - Chiamata di funzione
add esp, 4
mov eax, 1
jmp short loc_40103A - Salto
loc_401028:
push offset aError1_1_NoInte
call sub_40117F - Chiamata di funzione
add esp, 4
xor eax, eax
loc_40103A:
mov esp, ebp
pop ebp
retn
sub_40100 - fine procedura
```

Punto 4

La funzione ``sub_401000`` verifica lo stato della connessione Internet usufruendo la funzione ``InternetGetConnectedState``. Se la connessione è attiva, allora stamperà il messaggio "Success: Internet Connection" e restituisce 1. Se la connessione non è attiva, allora stamperà il messaggio "Error 1.1: No Internet" e restituisce 0.

In entrambi i casi, la funzione ripristina lo stack e termina il ciclo in modo corretto

Punto 5

Creazione dello stack

push ebp → crea lo stack inserendo dati nell' EBP
mov ebp, esp → copia i dati di EBP nello stack ESP

push ecx → inserisce un dato nello stack EAX
push 0 → inserisce un dato 0 alla funzione dwkserverd
push 0 → inserisce un dato 0 alla funzione lpdwFlags

Chiama la funzione

call ds:InternetGetConnectedState → Chiama la funzione per verificare lo stato della connessione internet
(InternetGetConnectedState)

mov [ebp+var_4], eax Copia il valore di EAX + var_4 nello stack di EBP

cmp [ebp+var_4], 0 → confronta il risultato di EBP+var_4 con 0
jz short loc_401028 → Salta allo stack loc_401028 se il risultato di prima è 0

push offset aSuccessInterne → inserisce un messaggio "Success: Internet Connection" sul loc_401028

Chiama la funzione

call sub_40117F → chiama la funzione sub_40117F per stampare un messaggio.

add esp, 4 → aggiunge 4 allo stack di ESP
mov eax, 1 → copia 1 nello stack di EAX

Salto

jmp short loc_40103A → Salta a loc_40103A, che sarebbe la fine della funzione

loc_401028: Etichetta in cui non vi è nessuna connessione: "Error 1.1: No Internet"
push offset aError1_1_NoInte → scrive un messaggio di mancanza connessione
sullo stack

Chiama la funzione

call sub_40117F → Chiama la funzione per poter stampare un messaggio

add esp, 4 → aggiunge 4 allo stack ESP
xor eax, eax → azzerà il registro di EAX, tornando così 0
loc_40103A: → Etichetta che indica la fine del ciclo

mov esp, ebp copia il valore di EBP in ESP
pop ebp → estrae il dato di EBP
retn → pone fine alla procedura / ritorno alla procedura
sub_401000 → FINE, **exit**