



# Progetto S10L5

## Tracce:

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:

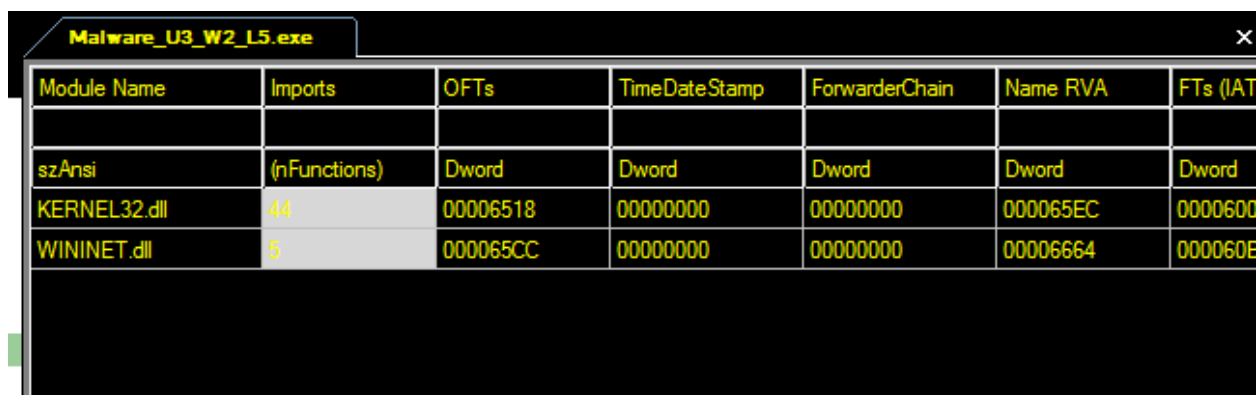
1. Quali librerie vengono importate dal file eseguibile?
2. Quali sono le sezioni di cui si compone il file eseguibile del malware?

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

3. Identificare i costrutti noti (creazione dello stack, eventuali cicli, altri costrutti )
4. Ipotizzare il comportamento della funzionalità implementata
5. BONUS fare tabella con significato delle singole righe di codice assembly

## Traccia 1

Quali librerie vengono importate dal file eseguibile?



Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	00006000

Le librerie importate dal malware in questione sono 2

- **Kernel32.dll**

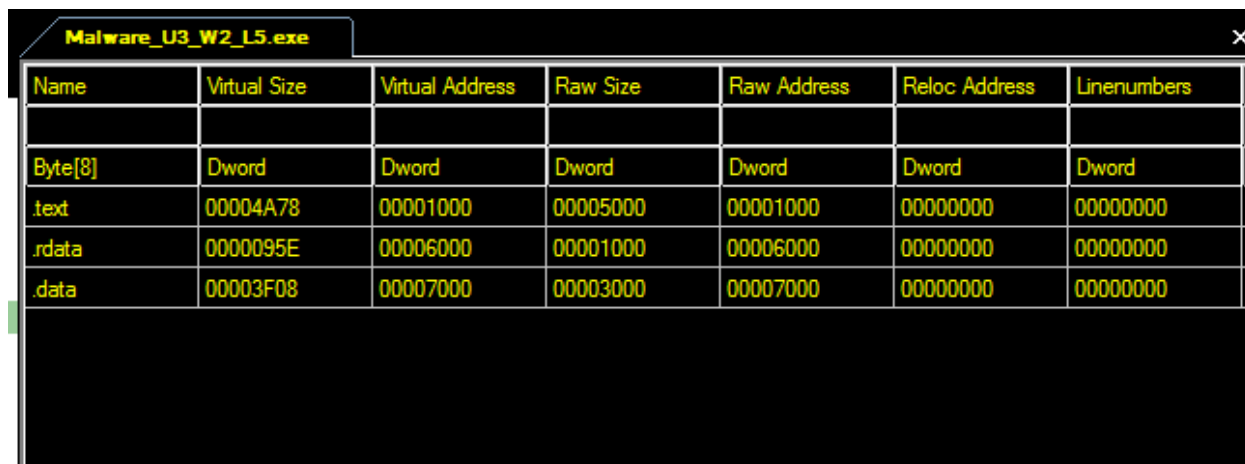
- è una libreria di sistema in ambienti Windows che contiene funzioni essenziali per la gestione della memoria, dei processi, dei file, dei dispositivi di input/output.

- **Wininet.dll**

- Contiene le funzioni necessarie per eseguire operazioni come il download e l'upload di file, la gestione delle connessioni HTTP, HTTPS e FTP, la gestione dei cookie, la cache dei dati Web e altre operazioni di rete.

## Traccia 2

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



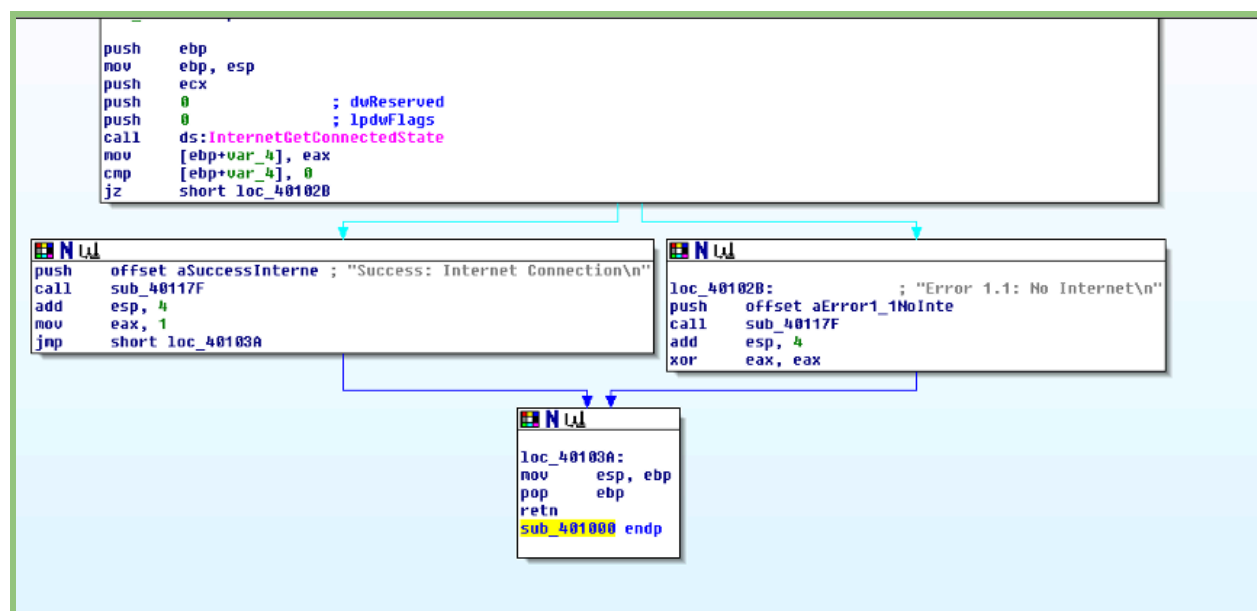
The screenshot shows a debugger window titled 'Malware\_U3\_W2\_L5.exe'. It displays a table of PE sections. The table has columns for Name, Virtual Size, Virtual Address, Raw Size, Raw Address, Reloc Address, and Linenumbers. The sections listed are .text, .rdata, and .data.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword
.text	00004A78	00001000	00005000	00001000	00000000	00000000
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000
.data	00003F08	00007000	00003000	00007000	00000000	00000000

Le sezioni di cui si compone il malware sono 3

- **.text**
  - Questa sezione contiene le istruzioni macchina che vengono eseguite dal processore.
- **.rdata**
  - è una sezione di dati di sola lettura in un file eseguibile.
- **.data**
  - è una sezione di dati in un file eseguibile che contiene variabili globali e statiche che possono essere modificate durante l'esecuzione del programma.

# Figura 3



## Traccia 3

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

```

push ebp
mov ebp,esp
push ecx
push 0 ;dwReserved
push 0 ;lpdwFlags
call ds:internetGetConnectedState
mov [ebp+var_4],eax
cmp [ebp+var_4],0
jz short loc_40102B

push offset aSuccessinterne ;"success:internet connection\n"
call sub_40117F
add esp,4
mov eax,1
jmp short loc_40103A

loc_40102B: ;"error 1.1: no internet\n"
push offset aError1_1Nointe
call sub_40117F
add esp,4
xor eax,eax

loc_40103A:
mov esp,ebp
pop ebp
retn
sub_401000 endp

```

### creazione stack

```

push ebp
mov ebp,esp

```

### chiamata alla funzione

```
call ds:internetGetConnectedState
```

### gestioni delle condizioni

```

cmp [ebp+var_4],0
jz short loc_40102B

```

### gestione delle condizioni

```
jmp short loc_40103A
```

## Traccia 4

1. La funzione `InternetGetConnectedState` è chiamata per verificare lo stato della connessione Internet.
2. Il risultato della chiamata a `InternetGetConnectedState` viene memorizzato nella variabile locale `[ebp+var_4]`.
3. Se il risultato è zero , il programma stampa un messaggio di errore(`error 1.1: no internet\n`).
4. Se il risultato non è zero , il programma stampa un messaggio di successo(`success:internet connection\n`).

## Traccia 6

Assembly	Spiegazione
push ebp	Salva un registro nello stack.
mov ebp,esp	sposta esp all'interno di ebp
push ecx	Salva un altro registro nello stack
push 0 ;dwReserved	Pone il numero zero nello stack
push 0 ;lpdwFlags	Pone il numero zero nello stack
call ds:internetGetConnectedState	Chiama una funzione per controllare la connessione Internet
mov [ebp+var_4],eax	Memorizza il risultato della funzione in una variabile
cmp [ebp+var_4],0	Confronta il valore memorizzato con zero
jz short loc_40102B	Salta a un'etichetta se il valore è zero
push offset aSuccessinterne ;"success:internet connection\n"	Pone un'etichetta di stringa nello stack
call sub_40117F	Chiama una funzione per gestire la stringa
add esp,4	Libera spazio nello stack dopo la chiamata della funzione
mov eax,1	Imposta un valore a 1 nel registro EAX
jmp short loc_40103A	Salta a un'altra parte del codice
loc_40102B: ;"error 1.1: no internet\n"	Etichetta per gestire l'assenza di connessione
push offset aError1_1Nointe	Pone un'etichetta di stringa nello stack
call sub_40117F	Chiama una funzione per gestire la stringa
add esp,4	Libera spazio nello stack
xor eax,eax	Imposta il registro EAX a zero
loc_40103A:	Etichetta per segnare un punto nel codice
mov esp,ebp	Inserisce ebp all'interno di esp
pop ebp	Ripristina il registro di base dello stack
retn	Ritorna dalla funzione
sub_401000 endp	Finisci la definizione della funzione

