

**S10-L5**

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

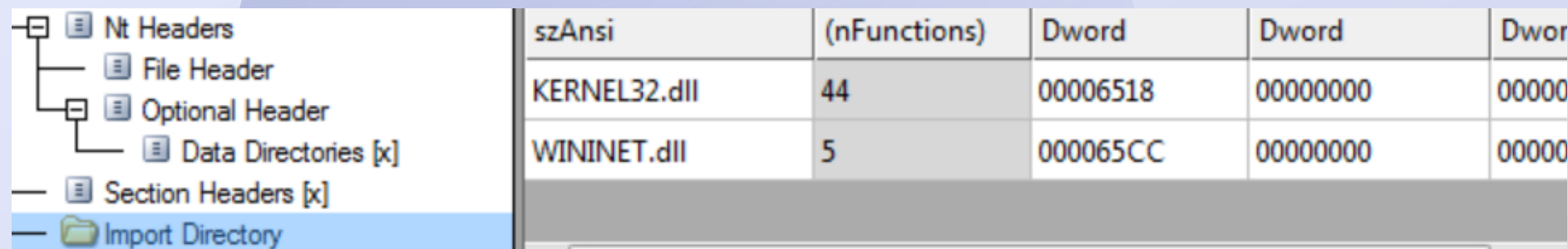
Con riferimento alla figura in slide 3, risponde ai seguenti quesiti:

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

```
.text:00401000  push ebp
.text:00401001  mov ebp, esp
.text:00401003  push ecx
.text:00401004  push 0 ; dwReserved
.text:00401006  push 0 ; lpdwFlags
.text:00401008  call ds:InternetGetConnectedState
.text:0040100E  mov [ebp+var_4], eax
.text:00401011  cmp [ebp+var_4], 0
.text:00401015  jz short loc_40102B
.text:00401017  push offset asuccessInterne ; "Succes Internet Connection\n"
.text:0040101C  call sub_40105F
.text:00401021  add esp, 4
.text:00401024  mov eax, 1
.text:00401029  jmp short loc_40103A
.text:0040102B  push offset aError1_1NoInte ; "Error 1.1: No Internet\n"
                call sub_40117F
                add esp, 4
                xor eax, eax
.text:0040103A:  mov esp, ebp
                pop ebp
                retn
sub_401000 endp
```

## Traccia 1-2:

Per eseguire un'analisi statica basica del malware in questione, per poter quindi verificare quale librerie importa e in che sezioni è diviso, utilizzeremo CFF explorer, una suite di strumenti per l'analisi e modifica di file eseguibili.



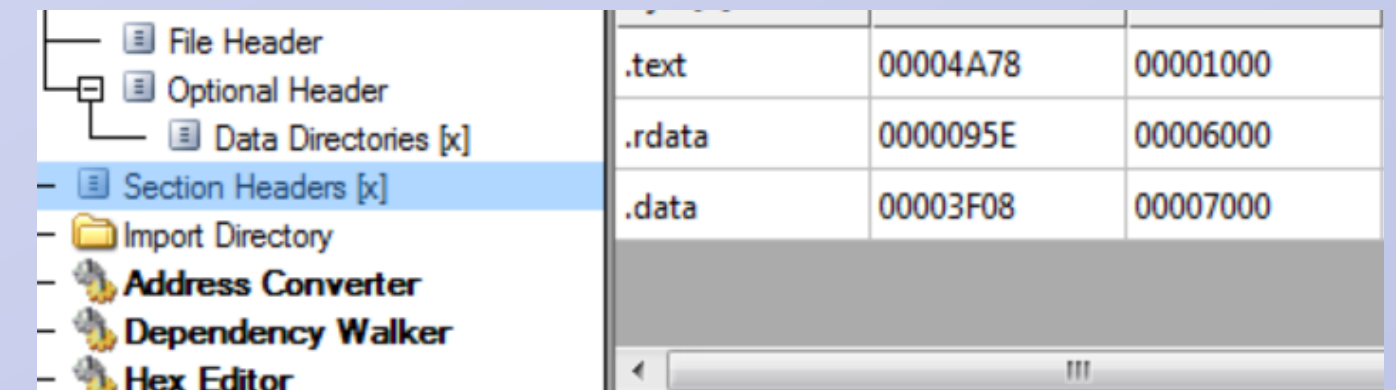
The screenshot shows the 'Import Directory' tab in CFF Explorer. The left pane shows a tree view with 'Nt Headers' expanded, showing 'File Header', 'Optional Header', 'Data Directories [x]', 'Section Headers [x]', and 'Import Directory'. The right pane displays a table of imported DLLs.

szAnsi	(nFunctions)	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000
WININET.dll	5	000065CC	00000000	00000000

In questo caso il malware importa Kernel32.dll e Wininet.dll, grazie a queste librerie potrebbe ad esempio:

- comunicare su server remoti
- manipolare i processi
- accedere ai file di sistema
- monitoraggio delle attività

Le sezioni di un malware si riferiscono alle diverse parti del codice malevolo che svolgono funzioni specifiche. Per analizzare queste sezioni a volte è necessario utilizzare la funzione "UPX Utiliy" messa a disposizione da "CFF Explorer" perché molte volte gli exe sono compressi o protetti da tecniche di "packing" che ne rendono difficile l'analisi diretta.



The screenshot shows the 'Section Headers' tab in CFF Explorer. The left pane shows a tree view with 'File Header', 'Optional Header', 'Data Directories [x]', 'Section Headers [x]', 'Import Directory', 'Address Converter', 'Dependency Walker', and 'Hex Editor'. The right pane displays a table of section headers.

Section Name	Virtual Address	Size
.text	00004A78	00001000
.rdata	0000095E	00006000
.data	00003F08	00007000



.data:

- Questa sezione di un file eseguibile contiene principalmente dati inizializzati, come variabili globali o altre strutture dati che devono essere allocate e inizializzate prima che il programma venga eseguito.

.text:

- La sezione .text contiene il codice eseguibile, ovvero le istruzioni macchina che costituiscono il programma stesso. Questa sezione contiene il codice sorgente compilato che viene eseguito quando avvii il programma.

.rdata:

- Questa sezione contiene principalmente dati di sola lettura, come costanti o stringhe di testo che il programma utilizza durante l'esecuzione ma non modifica.

## Traccia 3:

Creazione della stack,  
Il comando "push" in  
assembly viene  
utilizzato per inserire  
un valore sullo stack  
della CPU.

Ciclo "if"

Costrutto  
"go to"

```
.text:00401000  push ebp
.text:00401001  mov ebp, esp
.text:00401003  push ecx
.text:00401004  push 0 ; dwReserved
.text:00401006  push 0 ; lpdwFlags
.text:00401008  call ds:InternetGetConnectedState
.text:0040100E  mov [ebp+var_4], eax
.text:00401011  cmp [ebp+var_4], 0
.text:00401015  jz short loc_40102B
.text:00401017  push offset asuccessInterne ; "Succes Internet Connection\n"
.text:0040101C  call sub_40105F
.text:00401021  add esp, 4
.text:00401024  mov eax, 1
.text:00401029  jmp short loc_40103A
.text:0040102B  push offset aError1_1NoInte ; "Error 1.1: No Internet\n"
                    call sub_40117F
                    add esp, 4
                    xor eax, eax
.text:0040103A:  mov esp, ebp
                    pop ebp
                    retn
sub_401000 endp
```

## Traccia 4:

Il codice assembly sembra essere una funzione che verifica lo stato della connessione Internet e visualizza un messaggio relativo. La funzione "CheckInternetConnection" richiama "InternetGetConnectedState" per controllare lo stato della connessione. Poi, in base al risultato ottenuto, mostra un messaggio adeguato. Questo potrebbe far parte di un programma più ampio che gestisce la connettività di rete o verifica la presenza di Internet.

**Traccia 5:** .text:00401000 push ebp ;;mette il valore corrente di EBP nello stack.

.text:00401001 mov ebp,esp ;;copia su EBP il valore di ESP

.text:00401003 push ecx ;;mette il valore corrente di ECX nello stack.

.text:00401004 push 0 ;dwReserved ;;Queste due istruzioni mettono il valore 0 nello stack.

.text:00401006 push 0 ;lpdwFlags

.text:00401008 call ds:InternetGetConnectedState ;;chiama la funzione "InternetGetConnectedState". I valori 0 precedentemente inseriti nello stack potrebbero essere i suoi argomenti.

.text:0040100E mov [ebp+var\_4],eax ;;copia il valore di EAX nella posizione di memoria [ebp+var\_4].

.text:00401011 cmp [ebp+var\_4],0 ;;confronta il valore memorizzato in [ebp+var\_4] con 0.

.text:00401015 jz short loc\_40102B ;;salta all'indirizzo "loc\_40102B" se il confronto precedente ha dato esito positivo (ossia se [ebp+var\_4] è uguale a 0).



.text:00401017 push offset asuccessInterne; "Succes Internet Connection\n" ;;mette l'indirizzo della stringa "Succes Internet Connection\n" nello stack.

.text:0040101C call sub\_40105F ;;chiama la funzione "sub\_40105F".

.text:00401021 add esp,4 ;;aggiunge il valore di 4 al registro ESP.

.text:00401024 mov eax,1 ;;imposta il valore 1 nel registro EAX.

.text:00401029 jmp short loc\_40103A ;;effettua un salto non condizionale all'indirizzo "loc\_40103A".

.text:0040102B push offset aError1\_1NoInte; "Error 1.1: No Internet\n" ;;mette l'indirizzo della stringa "Error 1.1: No Internet\n" nello stack.

call sub\_40117F ;;chiama la funzione "sub\_40117F".

add esp,4 ;;aggiunge il valore di 4 al registro ESP.

xor eax,eax ;;esegue un'operazione XOR tra EAX e se stesso, azzerando quindi il registro EAX.

.text:0040103A: mov esp,ebp ;;copia il valore di EBP in ESP

pop ebp ;;ripristina il valore di EBP dallo stack.

ret ;;restituisce il controllo al chiamante.

sub 401000 endp ;;indica la fine di una procedura o di una funzione.