COSTRUTTI NOTI DEL SEGUENTE PROGRAMMA IN ASSEMBLY

-Creazione dello stack

```
push ebp
mov epb,esp
-Chiamata di funzione. I parametri sono passati sullo stack tramite le istruzioni push
push ecx
push 0 ;dwReserved
push 0 ;IdpwFlags
call ds:InternetGetConnectedState
mov [ebp+var_4],eax
-Ciclo IF
cmp [ebp+var_4],0
jz short loc_40102B
push offset aSucessInterne; "Success: internet Connection\n"
call sub_40105f
add esp,4
mov eax,1
jmp short loc_40103A
-ELSE
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
```

ANALISI DEL PROGRAMMA ASSEMBLY (FUNZIONALITA' ED FUNZIONAMENTO DEL CODICE)

1-push ebp ; Istruzione che porta il processore ad inserire il valore "ebp"nello stack

2-mov epb,esp ;Istruzione che porta il processore a copiare il valore di "esp" nel registro "ebp".Questa e' una operazione comune per la creazione di uno stack.

3-push ecx ;Istruzione per inserire il registro "ecx" nello stack,potrebbe essere usato come parametro nella funzione successiva.

4-push 0 ;dwReserved ;Parametro 2 5-push 0 ;ldpwFlags ;Parametro 3

pop ebp retn

sub_401000 endp

6-call ds:InternetGetConnectedState ;chiama la funzione "InternetGetConnectedState",se la funzione restituisce un valore diverso da 0 il sistema e' connesso altrimenti no.

7-mov [ebp+var_4],eax ;Copia il valore di ritorno della funzione nella variabile [ebp=var_4].

8-cmp [ebp+var_4],0

9-jz short loc_40102B ;IF questaistruzione confronta il valore restituito dalla funzione "InternetGetConnectedState" con 0

10-push offset aSucessInterne; "Success: internet Connection\n" 11-call sub_40105f 12-add esp,4

13-mov eax,1

14-jmp short loc_40103A ;Queste istruzioni vengono eseguite nel caso di successo alla connessione ad internet,"eax" viene impostato ad 1 ed il programma salta ad "loc_40103A"

15-loc_40102B: ;"Error 1,1 : No Internet\n" 16-push offset aError1_1NoInte 17-call sub_40117F

18-add esp,4

19-xor eax,eax

20-loc_40103A: ;Queste istruzioni vengono eseguite nel caso di errore,"eax viene azzerato ed il programma salta ad "loc_40103A.Xor E' un operatore logico che restituisce in uscita VERO(1) solo se gli ingressi sono diversi tra di loro.

21-mov esp,ebp

22-pop ebp

23-retn ;Queste istruzioni servono a ripristinare lo stack e tornare quindi al chiamante

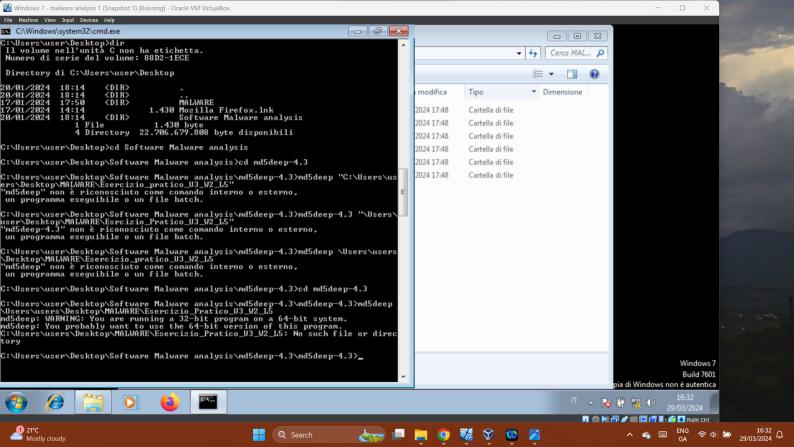
24-sub_401000 endp ;endp(end procedure)

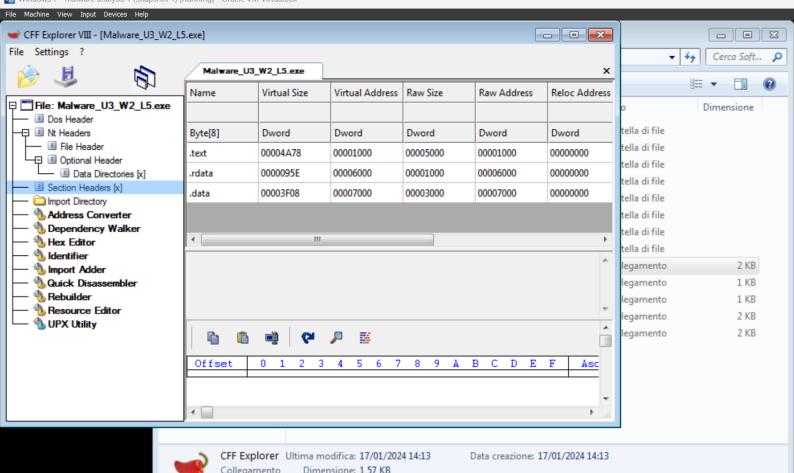
Le righe:16-,17- e 10- sono responsabili di stampare un messaggio di notifica dello stato di connessione a internet le opzioni sono "Success/Error".

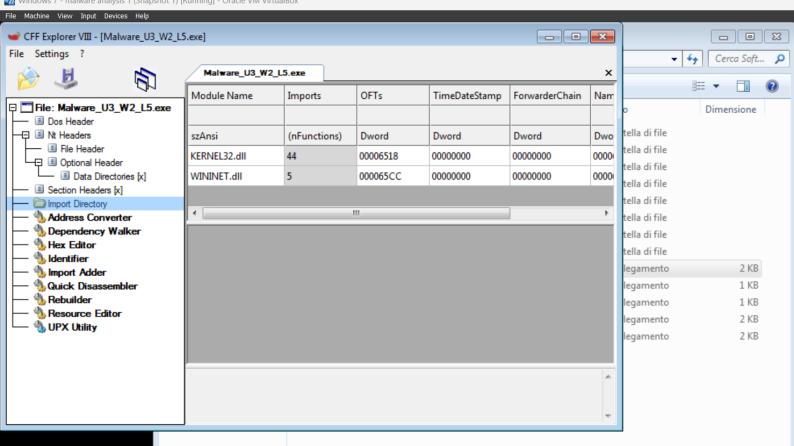
Analizzando il codice, sembra che il programma controlli se il sistema è connesso a Internet e poi stampi un messaggio di successo o di errore a seconda del risultato.

Analisi del malware Esercizio_Pratico_U3_W2_L5

Analizzando il programma (analisi statica base) con il tool md5deep abbiamo ottenuto l'informazione che il programma malevolo eseguibile e' dedicato ad un sistema 32 bit quindi non e' eseguibile sulla nostra macchina windows 7 64 bit, successivamente con il tool CFF si vanno ad analizzare le librerie importate che in questo caso sono:KERNEL32.dll e ed WININET.dll, conoscendo dunque le librerie importate sempre con il tool CFF ne studieremo l'ascii e le sezioni si puo' così definire la natura del malware.







 \Box

Windows 7

Build 7601