

Esercizio 4 Settimana 10

Assembly parte 2

Dato l'estratto di codice di un Malware, la traccia ci chiede di identificare i costrutti noti.

```
*.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 ; "Success: Internet Connection\n"
*.text:0040101C      call   sub_40105F
*.text:00401021      add     esp, 4
*.text:00401024      mov     eax, 1
*.text:00401029      jnp     short loc_40103A
*.text:0040102B      ; -----
*.text:0040102B
```

Il primo costrutto che possiamo notare è dato dalle prime due righe di codice che va a creare uno stack per le variabili locali. Possiamo anche notare la presenza dei due puntatori EBP (Extended Base Pointer) ed ESP (Extended Stack Pointer) che puntano rispettivamente alla base ed alla cima dello stack, una parte della memoria che viene utilizzata per gestire variabili locali e parametri delle funzioni durante l'esecuzione di un programma.

```
*.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 ; "Success: Internet Connection\n"
*.text:0040101C      call   sub_40105F
*.text:00401021      add     esp, 4
*.text:00401024      mov     eax, 1
*.text:00401029      jnp     short loc_40103A
*.text:0040102B      ; -----
*.text:0040102B
```

Il secondo costrutto evidenziato è una chiamata della funzione 'InternetGetConnectedState' tramite le istruzioni 'push' che passano i parametri allo stack.

```

* .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 ; "Success: 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      ; -----
* .text:0040102B

```

Il terzo costrutto è un ciclo if, con una istruzione cmp viene effettuata una sottrazione tra il parametro contenuto nel registro ebp+4_var e 0 la quale modifica la ZF (zero flag). L'istruzione call poi andrà a chiamare la funzione printf.