CONSEGNA U3 S10 L4

- **Creazione dello stack**

```
.text:00401000                  push    ebp
.text:00401001                  mov     ebp, esp
```

Con queste istruzioni il programma sta ripristinando lo stato dello stack, probabilmente dopo aver già eseguito operazioni con delle funzioni.

- push ebp: questa istruzione carica il **puntatore alla base dello stack (Extended Base Pointer)** sullo **stack**.
- mov ebp, esp: copia il valore dello **stack pointer** all'interno del **base pointer**.

- **Parametri di funzione**

```
.text:00401003                  push    ecx
.text:00401004                  push    0               ; dwReserved
.text:00401006                  push    0               ; lpdwFlags
```

- push ecx: copia il valore del registro **ecx** sullo **stack.**
- push 0: copia il valore **0** sullo **stack**.
- push 0: copia il valore **0** sullo **stack**.

Il **push** del valore **0** sullo **stack** probabilmente è utile per riservare dello spazio per l'esecuzione di qualche specifica istruzione. Possibilmente sono i **3 parametri** della funzione che verrà chiamata successivamente.

- **Chiamata di funzione**

```
.text:00401008                  call    ds:InternetGetConnectedState
```

- call ds:InternetGetConnectedState: qui viene chiamata la funzione **InternetGetConnectedState. ds** è un **segment register** in assembly e fa riferimento ad una precisa zona di memoria.

- **Costrutto IF**

```
.text:0040100E                  mov     [ebp+var_4], eax
.text:00401011                  cmp     [ebp+var_4], 0
.text:00401015                  jz      short loc_40102B
```

- mov [ebp+var_4], eax: copia il valore contenuto nel registro **eax** all'interno della variabile **var_4**.

- cmp [ebp+var_4], 0: qui avviene il confronto tra **sorgente (0)** e **destinazione ([ebp+var_4])** per capire quale fra le due sia maggiore o se si equivalgono. In pratica esegue una istruzione **sub [ebp+var_4], 0** senza però modificare il valore della variabile **var_4**. Se si equivalgono lo zero flag **ZF** sarà **true** ovvero **1**, mentre il carry flag **CF** sarà **0**. Se **var_4** sarà maggiore del valore **0**, **ZF** sarà **0** come anche **CF**. Se **var_4** sarà minore del valore **0, ZF** sarà **0** mentre **CF** sarà **1**. Il valore degli **EFLAGS (ZF e CF)** è fondamentale per l'esecuzione dei salti condizionali in assembly.
- jz short loc_40102B: questa istruzione è un salto condizionale. Esegue il salto alla locazione di memoria **40102B** se **ZF** vale **1** ovvero se è settato a **true**.

- **Stampa messaggio a schermo**

- push offset aSuccessInterne: fa il **push** dell'offset della variabile stringa **aSuccessInterne** sullo **stack**, ovvero fa il **push** del suo indirizzo. Serve quindi per stampare a schermo un messaggio testuale per l'utente, in questo caso per l'avvenuta connessione.

- **Chiamata di funzione**

- call sub_40105F: chiama la funzione all'indirizzo di memoria **40105F**.

- **Pulizia dello stack**

- add esp, 4: somma il valore **4** allo **stack pointer** per ripristinare il contenuto dello **stack** dopo il **push** della stringa **(aSuccessInterne)** avvenuto in precedenza. Il valore usato è **4** perché la variabile stringa occupa **4 byte**, ovvero **32 bit**.

- **Ultime istruzioni**

```
.text:00401024                  mov     eax, 1
.text:00401029                  jmp     short loc_40103A
```

  - mov eax, 1: copia il valore **1** all'interno del registro **eax**.
  - jmp short loc 40103A: esegue un salto all'indirizzo di memoria **40103A**.