**ESERCITAZIONE WEEK 22 DAY 5**

A computer screen shot of a computer code

Description automatically generated

Le prime due istruzioni servono a **creare lo stack della funzione** che il codice del malware richiamerà nelle righe successive. Quindi esegue una **push** per il puntatore alla base dello stack ed esegue una **mov** per definire il puntatore alla fine dello stack.

Seguono tre azioni di **push** per definire spazio in memoria per 3 variabili che la funzione adopererà.

Dopodiché con una **call** viene chiamata la funzione **InternetGetConnectedState**, che serve a valutare se una connesisone va a buon fine o meno.

A seguito evidentemente di un calcolo, viene spostato il valore del registro accumulatore in una variabile [ebp+var\_4] che viene confrontata con il valore 0: quindi segue un **costrutto IF** del linguaggio C che qui è rappresentato in Assembly con una compare **cmp** che confronta la variabile con 0, ovvero esegue la sottrazione tra le due variabili. Se la differenza è 0, si valorizza il **flag ZF=1**, **altrimenti ZF=0;** Per completare l’IF, viene eseguito il **jump di** **tipo jz:** se ZF=1 e quindi se la variabile è nulla (e quindi la connessione non è avvenuta) si salta all’istruzione presente in loc\_40102B; se invece ZF=0, quindi la variabilie non è nulla, la connessione è evidentemente avvenuta, non viene eseguito il jump e si prosegue con le righe successive.

Con una **push** si spinge la stringa “Success: Internet Conncetion\n” che verrà stampata in seguito alla **call sub\_40102F** che possiamo affermare con buona probabilità essere la funzione **printf**.

Segue una **add che chiude lo stack della funzione** **appena chiamata**, una **mov** che sposta il valore 1 sul registro accumulatore e un **jump jmp** incondizionale all’istruzione loc\_40103A.