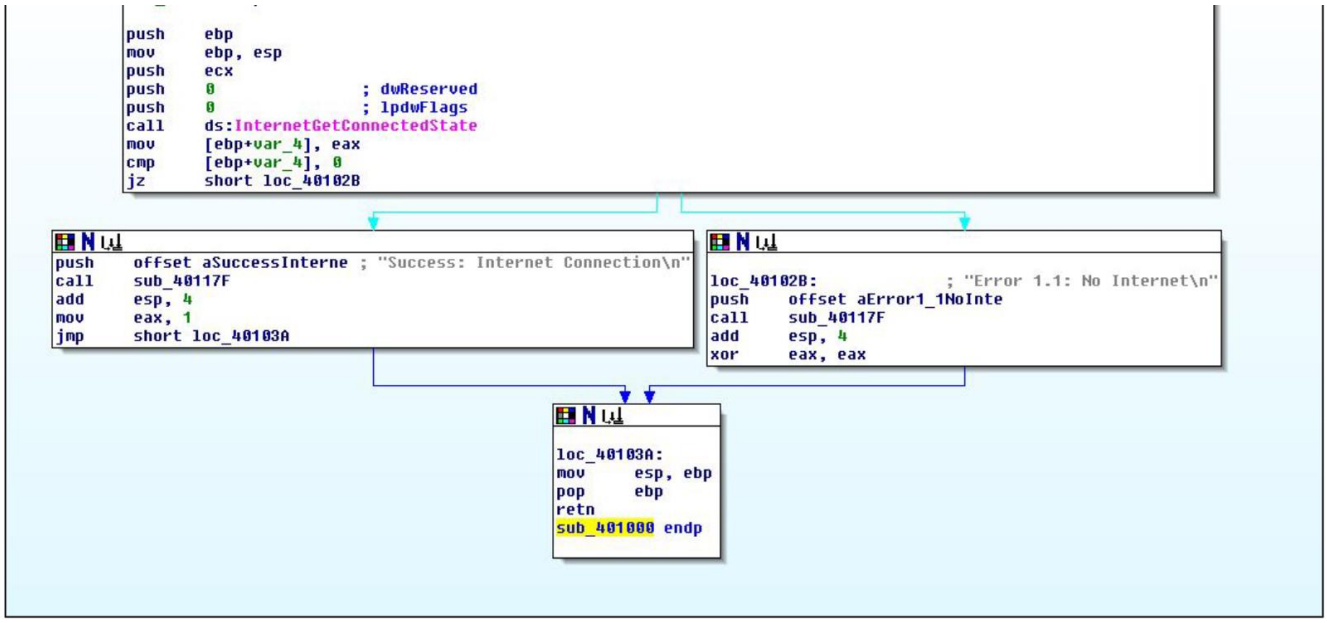

1.) *CREAZIONE DELLO STACK*: una funzione chiamante chiama un'altra funzione (funzione chiamata ovvero ***InternetGetConnectedState***) e viene creato un nuovo stack

Push ebp

Mov ebp,esp

2.) *CHIAMATA DI FUNZIONE E PASSAGGIO DI PARAMETRI :* Vengono passati i parametri(nel nostro caso 3 parametri) alla funzione chiamata sullo stack tramite istruzioni *push* e secondo la regola del LIFO, poi la funzione viene chiamata direttamente con l'istruzione *call:*

Push ecx

Push 0

Push 0

Call ds: InternetGetConnectedState

3.) *ASSEGNAZIONE VALORE:* viene assegnato il valore contenuto nel registro eax all'indirizzo di memoria specificato tramite l'istruzione *mov: (praticamente è il valore di ritorno della funzione che viene messo in eax e poi assegnato all'indirizzo di memoria specificato quindi viene creata una variabile locale contenente il valore di ritorno)*

Mov [ebp+var_4], eax

4.) *IF/ELSE:* viene controllato il valore di ritorno(return) della funzione chiamata tramite un ciclo if, confrontando il valore assegnato all'indirizzo di memoria **ebp+var_4 con 0.**

Se questo valore di ritorno è uguale a 0 allora parte l'else e significa che non c'è connessione, se diverso da 0 allora c'è connessione:

Cmp [ebp+var_4],0

Jz short_loc…..

Infatti ricordiamo che dopo il confronto con CMP c'è un jump da effettuare e nel nostro caso viene fatto se lo ZF è impostato a 1(JZ) ovvero quando il valore di sorgente e destinazione sono uguali, per cui se il valore di ritorno della funzione è 0 allora è uguale al valore con cui si fa il confronto (0) e di conseguenza lo ZF si setterà su 1 e viene effettuato il jump (in C partirebbe l'ELSE)

Possiamo infatti ipotizzare il codice in C come il seguente:

```
state=internetgetconnectedstate[par1,0,0];
if (state!=0)
    printf("internet connection");
else
    printf ("no internet");
    return 0;
```