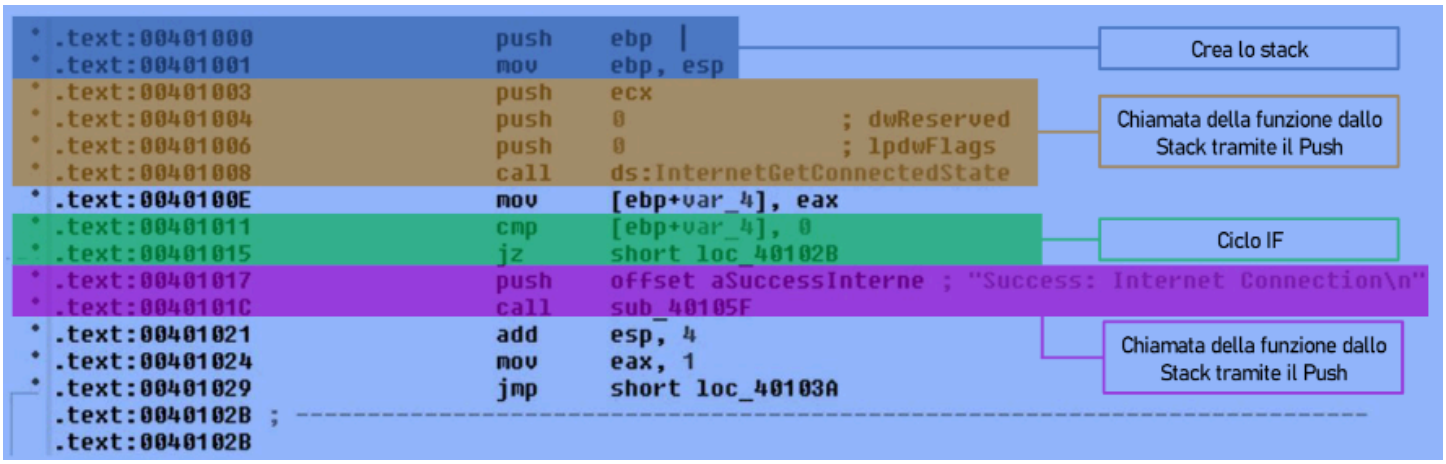
Malware Analysis / Costrutti C-assembly x86

Guardando il codice fornito, possiamo identificare alcuni costrutti noti e operazioni comuni:

- **Push e Pop nello stack:** le istruzioni push e pop vengono utilizzate per manipolare lo stack, inserendo e rimuovendo valori.
- **Movimento di valori tra registri e memoria:** le istruzioni mov vengono utilizzate per spostare dati tra registri e posizioni di memoria.
- **Chiamata di funzione:** l'istruzione call viene utilizzata per chiamare una funzione.
- **Controllo del flusso condizionale:** l'istruzione cmp viene utilizzata per confrontare due valori, mentre l'istruzione jz (jump if zero) viene utilizzata per saltare a una posizione specificata se il confronto precedente ha dato esito zero.
- **Salti incondizionati:** l'istruzione jmp viene utilizzata per saltare incondizionatamente a una posizione specificata nel codice. Non ci sono cicli while o for espliciti nel codice fornito. Potrebbero esserci all'interno delle funzioni chiamate, ma non sono visibili nel frammento fornito.

```
.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
```

Analizzando il codice troviamo 4 costrutti con le rispettive funzionalità



Dal codice del malware, si può dedurre che la sua funzionalità principale consiste nell'eseguire una chiamata di funzione a InternetGetConnectedState, che controlla lo stato della connessione Internet. Il codice verifica il valore di ritorno di questa funzione tramite un'istruzione condizionale. Se il valore di ritorno della funzione è diverso da zero, indica che la connessione è

attiva. Dopo aver determinato che la connessione Internet è attiva, il malware inserisce nello stack di memoria una stringa costante contenente il messaggio "Success: Internet connection".e successivamente richiamato tramite una chiamata della funzione.