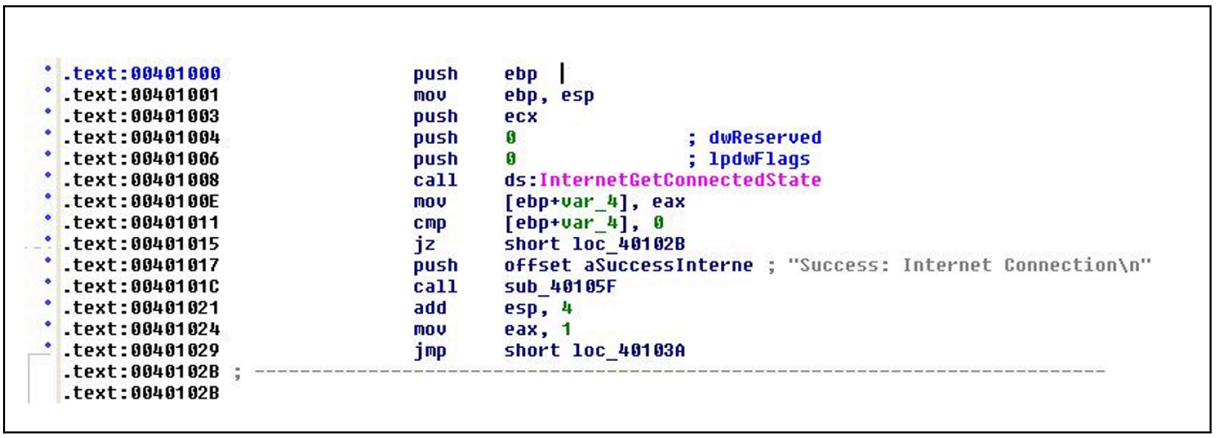
CONSEGNA S10/L4

La figura seguente mostra un estratto del codice di un malware.
Identificare i **costrutti** noti visti durante la lezione teorica.



```
.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
```

Opzionale:
Provate ad ipotizzare che funzionalità è implementata nel codice assembly.

Hint:
La funzione internetgetconnectedstate prende in input 3 parametri e permette di controllare se una macchina ha accesso ad internet.

SVOLGIMENTO

Setup del Frame Pointer: *push ebp* seguito da *mov ebp*, *esp* sono istruzioni standard per la configurazione del frame pointer in una nuova funzione, che serve a salvare il contesto attuale dello stack prima di eseguire il resto del codice.

Push di argomenti per una chiamata di funzione: Le istruzioni *push 0* (due volte) stanno probabilmente mettendo due argomenti sullo stack.
Questo è tipico per una funzione che accetta argomenti, che in assembly vengono passati tramite lo stack.

Chiamata di funzione: *call ds:InternetGetConnectedState* è una chiamata a una funzione importata che, come suggerisce l'hint, verifica se la macchina ha accesso a internet. In Windows, InternetGetConnectedState è una funzione dell'API di Windows che determina lo stato della connessione a internet.

Gestione del valore di ritorno: Dopo la chiamata alla funzione, il valore di ritorno viene confrontato con zero *(cmp [ebp+var_4], 0*). Se la funzione restituisce zero (che significa nessuna connessione internet), il flusso del programma salta ad una locazione etichettata (*jz short loc_40192B*), che è tipicamente usata per la gestione dell'errore o per eseguire codice condizionale basato sul risultato di una funzione.

Gestione dello stack post-chiamata: *add esp, 4* pulisce lo stack dopo la chiamata alla funzione, rimuovendo gli argomenti che erano stati passati.

Messaggio di successo: L'istruzione *push offset aSuccessInterne* sembra riferirsi al push di un messaggio di successo sulla pila, forse per essere usato in seguito per un output o un log, suggerendo che il codice successivo (non mostrato nell'immagine) gestirà il caso di connessione internet riuscita.

OPZIONALE: FUNZIONALITA' IMPLEMENTATA NEL CODICE ASSEMBLY

Basandomi su questi elementi e sull'hit fornito, posso ipotizzare che questo frammento di codice è progettato per verificare se il sistema su cui il malware è eseguito ha una connessione internet attiva. Se la connessione è presente, potrebbe proseguire con il suo comportamento malintenzionato, come connettersi a un server di comando e controllo o scaricare ulteriori payload malevoli. Nel caso in cui la connessione non sia attiva, potrebbe saltare alcune funzionalità o tentare di stabilire una connessione in altri modi.