Traccia:

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
```

00401000 push ebp; Salva il valore del registro base

00401001 mov ebp, esp; Imposta il puntatore di base dello stack

00401003 push ecx; Salva il registro ecx nello stack

00401004 push 0; dwReserved

00401006 push 0; lpdwFlags

00401008 call ds:InternetGetConnectedState; Chiama la funzione InternetGetConnectedState

0040100E mov [ebp+var_4], eax; Muove il valore di eax nella variabile locale var_4

00401011 cmp [ebp+var_4], 0; Confronta il valore nella variabile var_4 con zero

00401015 jz short loc_40102B; Salta a loc_40102B se il confronto precedente è zero (Internet non connesso)

00401017 push offset aSuccessInternet; "Success: Internet Connection\n"

0040101C call sub_40105F; Chiama la subroutine sub_40105F (presumibilmente per stampare il messaggio)

00401021 add esp, 4; Pulisce gli argomenti dalla pila

00401024 mov eax, 1; Imposta eax a 1 (presumibilmente per indicare successo)

00401029 jmp short loc_40103A; Salta a loc_40103°

Costrutti identificati:

1. **push**: Per mettere valori nello stack prima di una chiamata di funzione.
2. **mov**: Per spostare dati da un registro a un altro o dalla memoria a un registro.
3. **call**: Per chiamare una funzione.
4. **cmp**: Per confrontare due valori.
5. **jz**: Salto condizionale, qui utilizzato per controllare se la connessione Internet è presente.
6. **add**: Per aggiungere un valore a un registro o alla memoria.
7. **jmp**: Salto incondizionato, qui utilizzato per saltare a loc_40103A.