

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
```

1. Identificare i costrutti noti (es. while, for, if, switch, ecc.)
2. Ipotizzare la funzionalità –esecuzione ad alto livello
3. BONUS: studiare e spiegare ogni singola riga di codice

1. Identificare i costrutti noti (es. while, for, if, switch, ecc.)

In questo primo caso abbiamo un costrutto *if-else*

```
.text:00401011      cmp     [ebp+var_4], 0
.text:00401015      jz      short loc_40102B
```

In questo secondo caso abbiamo un costrutto *go-to*

```
.text:00401029      jmp     short loc_40103A
```

2. Ipotizzare la funzionalità –esecuzione ad alto livello

Sembrerebbe la parte di un malware che va a verificare la presenza o meno di connessione a internet. Se la connessione è presente esso restituirà il messaggio “Success: Internet connection”, in caso contrario si prosegue nell’esecuzione del codice fino al jmp (go-to) short loc_40102B

3. BONUS: studiare e spiegare ogni singola riga di codice

- **push ebp** e **mov ebp, esp** servono per creare lo stack della funzione
- **push ecx** setta il valore di ecx nello stack
- **push 0** and **push 0** pongono due valori zero sullo stack, come argomento per la funzione call successiva
- **call ds:InternetGetConnectedState** chiama una funzione per verificare la presenza di connessione ad internet e salva il risultato nella variabile locale **[ebp+var_4]**
- **cmp [ebp+var_4], 0** confronta il risultato precedente con zero.

- **jz short loc_40102B** rimanda alla location loc_40102B se il risultato è zero, quindi se non c'è connessione ad internet
- **push offset aSuccessInterne** manda allo stack l'indirizzo della stringa "Success: Internet Connection\n"
- **call sub_40105F** chiama una subroutine con la locazione del messaggio di avvenuta connessione
- **add esp, 4** setta il puntatore stack per eliminare gli argomenti
- **mov eax, 1** setta il valore di eax a 1
- **jmp short loc_40103A** ci rimanda direttamente alla locazione loc_40103A