

# Esercizi Assembly

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
```

- push ebp -> mette ebp sullo stack
- mov ebp, esp -> inserisce il valore di esp in ebp
- push ecx -> mette ecx sullo stack
- Le seguenti due push indicano dei parametri false messi sullo stack che serviranno per la chiamata di funzione successiva
- call ds:internetgetconnectedstate indica la chiamata di questa funzione che permette di controllare se una macchina ha accesso a internet, prende in input 3 parametri
- mov [ebp...], eax -> inserisce il valore di eax nella variabile
- cmp [ebp...], 0 -> compara il valore 0 con ebp eseguendo una sottrazione
- jz short loc... -> jz indica che se lo ZF è 1 allora fa il jump, short indica un indirizzo di memoria vicino sul quale fare il jump
- push offset aSuccess... -> inserisce la stringa sullo stack
- call sub... -> chiamata di funzione che probabilmente in questo caso è una printf
- add, esp, 4 -> aggiunge 4 ad esp
- mov eax, 1 -> imposta il valore di eax a 1
- jmp short loc... -> esegue il jump ad un determinato indirizzo di memoria vicino

Considerazioni finali - -> Il codice di riferimento potrebbe rappresentare un malware come un downloader o una backdoor che tentano di ottenere connessione ad internet da una determinata macchina

