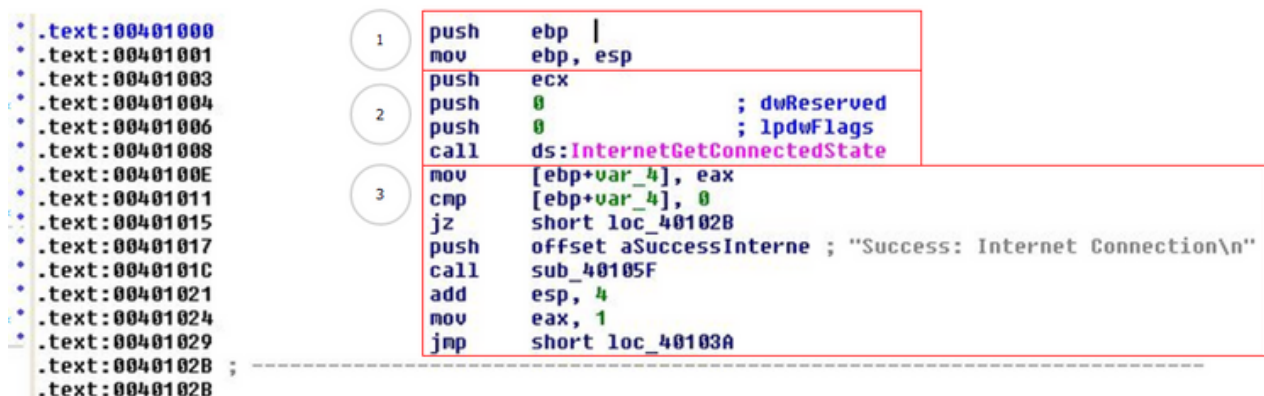


Esercizio S10 L4

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



The image shows a snippet of assembly code with memory addresses on the left. Three specific instructions are highlighted with red boxes and numbered circles:

- 1** points to the first instruction: `push ebp` followed by `mov ebp, esp`.
- 2** points to the second instruction: `push ecx` followed by `push 0` (twice) with comments `; dwReserved` and `; lpdwFlags`.
- 3** points to the third instruction: `call ds:InternetGetConnectedState`.

The code continues with a comparison of a memory location to zero, a jump if zero instruction, and a call to a function that prints a success message.

1) Crea lo stack.

2) Tramite push passa i parametri allo stack e chiama la funzione "InternetGetConnectedState".

3) Fa un compare e poi un jump, se la variabile è uguale a 0 (ZF=1) salta all'indirizzo di memoria indicato. Si tratta di un if(x != 0).

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

Il malware chiama una funziona che ricevendo tre variabili in input da in output 0 se la connessione ad internet non è presente. Nel caso in cui risponda un valore diverso da 0 stampa a schermo un messaggio di conferma.

Bonus:

1. Definisce il puntatore alla base dello stack EBP tramite push.
2. Definisce ESP che punta alla cima dello stack.
- 3, 4, 5. Tramite push mette le variabili necessarie alla funzione che sta per chiamare nella riga 6.
6. Chiama una funzione con Call.
7. Copia EAX in EBP.
8. Compara EBP con 0, se uguali imposta ZF=1.
9. Jz salta all'indirizzo indicato se ZF=1 quindi se EBP=0.
10. Push per fornire i parametri necessari alla call nella riga successiva.
11. call di una funzione che sembra essere una printf.
12. Sposta lo stack.
13. Setta eax a 1.
14. Salta ad un indirizzo con le istruzioni successive a questa porzione di codice.