

## PRATICA S10L4

Identificare i costrutti visti a lezione del seguente codice:

```
.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
```

L'unico costrutto riconoscibile all'interno del codice è quello specificato dall'istruzione condizionale `jz`. Viene eseguito un salto se lo Zero Flag è settato (quindi è 1) in riferimento all'istruzione `cmp` in riga sopra. In **rosso** la parte eseguita se il contenuto del registro `eax=0`, in quanto viene copiato all'interno della `var_4` e successivamente confrontato con il valore 0. In **blu** la parte di codice eseguita altrimenti:

```
.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
```

Questo costrutto è quindi paragonabile ad un `if/else` in linguaggio C. Nella parte di codice disponibile non è visibile nessuna istruzione all'indirizzo 0040102B, quindi possiamo ipotizzare che il `jz` in caso la condizione sia soddisfatta serva ad eseguire istruzioni successive. Stesso discorso per il `jmp` incondizionato all'ultima riga, che come detto viene eseguito solo in caso la condizione non sia soddisfatta, essendo l'indirizzo 0040103A successivo alle istruzioni fornite.

### Ipotesi sul funzionamento del programma

Notiamo che all'indirizzo 0040100E il programma copia il contenuto di EAX senza averne inizializzato il valore. Data la chiamata di funzione precedente `ds:InternetGetConnectedState`, possiamo ipotizzare che sia stata quest'ultima a valorizzare il registro EAX. Quindi quello che il programma sta presumibilmente facendo è controllare se EAX è zero, quindi se c'è connessione. Nel caso sia zero infatti il blocco di codice successivo viene saltato, altrimenti aggiunge allo stack la stringa "Success: Internet Connection". Successivamente esegue una subroutine non specificata utilizzando tale stringa.

Un malware potrebbe eseguire questo blocco per verificare se il target infettato ha connessione verso l'esterno ed usare questa connessione per connettersi ad esempio con un eventuale attaccante.