

Nell'esercizio di oggi siamo andati ad identificare i costrutti che formano l'estratto di codice malware sotto riportato.

```
.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
```

Abbiamo individuato 5 macrocategorie:

- 1) text 00401000/1 Creazione Stack
- 2) text 00401003/4/6 push dei valori dentro la funzione dello stack
- 3) text 00401008 chiamata della funzione
- 4) text 0040100E/11/15/17 dove avviene la comparazione e l'eventuale salto
- 5) text 0040101C/21/24/29

I [text 00401000/1/3](#) creano lo stack della funzione. Dove allo [stack ebp](#) di partenza, viene impilato lo stack esp (LIFO). Dopodiché viene inserito l'elemento dello [stack ecx](#) allo [stack ebp](#).

Dopodiché vengono inseriti (tramite il push)gli elementi con valore 0.

Presupponendo che nel codice, la variabile [lpdwFlags](#) corrisponda alla mancata connessione nel caso si avveri la condizione e [dwReserved](#) corrisponda all'effettiva connessione purché si avveri sempre la condizione:

[condizione](#) — — — —> [[lpdwFlags](#) = 0] [[dwReserved](#) = 1]

abbiamo dunque due valori pushati pari a 0.

Avviene a questo punto la call di verifica Connessione. [InternetGetConnectedState]

Al [text 0040100E](#) viene aggiunto allo [stack ebp](#) che adesso prevede una variabile ([var_4 = 0](#)) (poiché in precedenza sono stati pushati due valori booleani pari a 0).

Subito dopo avviene la comparazione tra il valore sorgente di 0 e la variabile inserita in precedenza sempre uguale a 0.

Ci troviamo dunque di fronte ad una situazione dove lo [ZF è settato a 1 e il CF settato a 0 poiché la sorgente è uguale alla destinazione](#).

A questo punto si salta alla [locazione 40102B](#).

Questo sta a significare che non si è avverata la condizione booleana del dwReserved (si sarebbe avverata nel caso il valore pushato fosse stato 1 e nella comparazione non avremmo avuto un situazione dove la destinazione aveva lo stesso valore della sorgente) e quindi avverandosi la condizione lpdwFlags viene istruito il salto jzloc.

