ASSEMBLY : INGENERIA REVERSA

S10 - L4

```
.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 è l'estratto di un codice assembly. Questo codice prima salva il valore del *base pointer* con push ebp dentro lo stack. Poi copia il valore dello *stack pointer* all'interno base pointer attraverso la funzione mov, in modo da creare un nuovo stack frame

```
.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
```

Questa è la sezione più importante del codice esaminato. Per prima cosa mette 2 valori pari a 0 nello stack con l'istruzione push. Questi valori sono spiegati come *dwReserved* e *lpwFlags*. Sembra che il codice si stia preparando a chiamare una funzione con 2 argomenti.

La funzione chiamata dopo è InternetGetConnectedState che, come dice il nome, è in grado di verificare se il computer che esegue il codice ha accesso a Internet.

```
.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
```

La prima riga con la istruzione **mov** sposta il valore del registro **eax** (che probabilmente contiene il valore restituito dalla funzione *InternetGetConnectedState*) nella memoria in **[ebp+var_4]**. Probabilmente si tratta di memorizzare il risultato del controllo della connessione Internet per un uso successivo.

La seconda linea confronta il risultato ottenuto con l'istruzione precedente contro il valore **0**, se il risultato sarà uguale (con un valore di 0) quindi la linea successiva avrà un salto condizionale con **jz** verso la direzione della memoria **40102B**, che probabilmente contiene un messaggio di "*connessione internet fallita*"

```
.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
```

Le prime 2 righe sono semplicemente responsabili della creazione di una funzione printf con la stringa "Success: Internet Connection\n"
La seconda riga esegue la chiamata al comando call sub_40105F, che fa riferimento all'indirizzo dove probabilmente si trova la funzione printf.

```
.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
```

Infine queste ultime 3 righe puliscono lo stack e lo riportano allo stato precedente

In conclusione, questo estratto di codice malware svolge la funzione di verificare se il computer della vittima dispone di una connessione Internet, questo è un passaggio cruciale per qualsiasi tipo di attacco remoto che consenta all'aggressore di comunicare con la vittima, sia per ricevere dati che per invialo.