COSTRUTTI C – ASSEMBLYx86

Traccia:

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
```

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

Hint:
La funzione **internetgetconnectedstate** prende in input 3 parametri e permette di controllare se una macchina ha accesso ad internet.

Consegna:

1. Identificare i costrutti noti (es. while, for, if, switch, ecc.)
2. Ipotizzare la funzionalità – esecuzione ad alto livello
3. Bonus: studiare e spiegare ogni singola riga di codice

Dato il codice del malware iniziamo ad analizzarlo per cercare di comprendere il suo comportamento:

```
.text:00401000                push    ebp
.text:00401001                mov     ebp, esp
```

Queste istruzioni Assembly servono per creare lo stack di una funzione. Lo stack è una porzione di memoria dedicata per il salvataggio delle variabili locali di una data funzione. Esso viene definito dai puntatori allo stack EBP (Istruction Base Pointer) che punta alla sua base ed ESP (Istruction Stack Pointer) che punta alla cima.

```
.text:00401003                push    ecx
.text:00401004                push    0               ; dwReserved
.text:00401006                push    0               ; lpdwFlags
.text:00401008                call    ds:InternetGetConnectedState
```

In queste istruzioni indicano il modo in cui la funzione chiamante invia i parametri necessari alla funzione chiamata per poter svolgere il suo compito sullo stack, i parametri vengono inseriti con “push” sullo stack

prima della chiamata alla funzione InternetGetConnectedState che permetterà di determinare se la macchina ha accesso ad internet.

Inizializzazione della variabile [ebp+var_4] al registro eax, quindi il risultato della chiamata della funzione InternetGetConnectedState viene spostato nella locuzione di memoria [ebp+var_4].

Il costrutto noto in questo codice è uno Statement SWITCH, costrutto che viene utilizzato per prendere decisioni in base al valore di una determinata variabile, e utilizza una sintassi simile ad un ciclo IF (**IF-Style**). La caratteristica principale degli IF-Style è la presenza di una serie di salti condizionali; il blocco dei salti condizionali è composto da un'istruzione "cmp" seguita da un'istruzione di salto, in questo caso "jz". L'istruzione "cmp" unita all'istruzione jz controllano l'uguaglianza tra il valore contenuto in [ebp+var_4] e 0, modificando il flag ZF in base al risultato; ZF sarà 0 quando gli operandi saranno diversi tra loro quindi il risultato della comparazione sarà diversa da 0. Jz salta alla locazione di memoria specificata se gli operandi sono diversi tra di loro, quindi se ZF = 0, riprendendo l'esecuzione del codice da tale locazione.

Se la comparazione sarà quindi diversa da 0 e ZF = 0, il programma ci darà "Success: Internet Connection\n" tramite chiamata di funzione alla funzione sub_40105F.

L'istruzione "add" andrà ad aggiustare la posizione dello stack pointer esp aggiungendo il valore 4.
L'istruzione "mov" sposterà il valore 1 nel registro eax.
L'istruzione "jmp" eseguirà un salto non condizionale alla locazione di memoria specificata.

Dopo aver analizzato il codice ed aver trovato i costrutti noti in esso, possiamo ipotizzare che questo malware ricada nella categoria delle backdoor che utilizzano costrutti di tipo "switch" per consentire localmente alla macchina una serie di azioni/comandi sul valore di una variabile. In questo caso particolare sembra che il malware chiami la funzione InternetGetConnectedState e ne controlli il valore di ritorno con un ciclo IF-Style, avvertendo dell'eventuale connessione attiva quando il risultato della funzione è diverso da 0.