REPORT 12/01/2023

FACCILONGO DOMENICO

COSTRUTTI C – ASSEMBLY x86

TASK

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
```

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

ANALISI E VALUTAZIONE:

1. Andiamo esaminare i costrutti noti del codice:
   - Le sezioni evidenziata in figura è un costrutto di **if – else** in C.

```
.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
```

In cui il cmp va a dare una condizione all'**if** del jz, di fatti se la condizione del cmp è verificata allora verrà eseguito l'**if** del jz che farà un salto alla locazione *loc_40102B* (rettangolo rosso in figura).
Altrimenti verrà eseguito l'**else** in cui ci sarà tutta l'istruzione identificata dal rettangolo verde in figura.

2. 3. Ipotizziamo ora la funzionalità del codice andando ad analizzare ogni passaggio: (include punto 3. BONUS)
   - Possiamo supporre che il codice funge da un controllo ad una connessione ad internet da parte di un malware. Di fatti esaminandolo passo dopo passo potremmo dire che le prime due istruzioni permettono di creare uno stack di lunghezza non indicata. 🔴
   - Le successive tre istruzioni **push** sono parametri associati alla funzione chiamata (**call**), quest'ultima permette di recuperare lo stato di connessione del sistema locale attraverso i risultati dei 3 parametri precedenti booleani. Di fatti sulla base dei valori true (0) o false (1) dei parametri **dwReserved** e **lpdwFlags** è possibile identificare se c'è o meno connessione ad internet. 🟢
   - Nei passaggi successivi infatti, si va a confrontare il valore ottenuto all'interno dello stack, con il valore 0. Se è verificata la loro uguaglianza allora **jz** (jump zero) permette di effettuare un salto condizionale alla locazione di memoria n. *loc_40102B.* 🔵
   - In alternativa alla condizione, se non è verificata, allora si passa all'istruzione successiva al jz, in cui **push** và a creare un testo posizionato in cima allo stack con scritto "Success: Internet Connection " e che fatta apparire a schermo solo con l'ausilio di un pritnf che ipotiziamo venga richiamato dall'istruzione successiva **call**. Infine le ultime 3 istruzioni [**add, mov, jmp**] vanno rispettivamente ad: aggiungere/sommare il valore 4 al registro (esp), settare il registro (eax) a valore 1 e saltare alla locazione di memoria n. *loc_40103A.* 🟡