2024

EPICODE

CS0124

REPORT

Week 10 Lesson 4

PREPARED BY : Bruno Falconi

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
```

EPICODE

Esercizio
Linguaggio Assembly

Provate ad ipotizzare che funzionalità è implementata nel codice assembly.

Hint:
La funzione internetgetconnectedstate permette di controllare se una macchina ha accesso ad Internet.

4

EPICODE

Esercizio
Linguaggio Assembly

Consegna:

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

1) Per quanto riguarda il codice assembly dell'esercizio i costrutti noti identificabili sono un costrutto "IF" che potrebbe essere un if-else in quanto compara la variabile [ebp+var_4] con 0 e quindi con un risultato salterà tramite JZ allo stack di memoria 40192B mentre altrimenti svolgerà un'altra operazione. Il secondo costrutto identificabile è un costrutto "CALL" in quanto va a chiamare i "PUSH" precedenti.
2) Per quanto riguarda la possibile funzionalità possiamo ipotizzare che il malware cerchi di connettersi ad internet dalla macchina dell'attaccato e che in caso di connessione effettuata vada a printare sulla macchina dell'attaccante il messaggio "Success: Internet Connection".
3) Per le righe di codice troviamo un push che va a creare lo stack dal puntatore ebp, sposta il esp in ebp il quale acquisisce valore esp. Subito dopo va a dare i tre valori definiti con "push" che sono ecx, 0 ; dwReserved e 0; lpdwFlags che va a chiamare con "Call" per la funzione ds;InternetGetConnectedState.

Muove dalla sorgente “eac” verso la destinazione [ebp+var_4], dopodiché compara “eax” quindi [ebp+var_4] a “0” e se i due valori sono uguali salterà alla posizione specificata, altrimenti continuerà con il codice. In caso i due valori non si fossero dimostrati uguali andrebbe a fare il Push dell’avvenuta connessione ed andrebbe successivamente a chiamare una subroutine o comunque una funzione che non è visibile dall’immagine. Infine con le ultime tre righe libera dello spazio nella memoria e salta con l’ultimo jump al resto del codice.
Da notare che all’inizio non va ad allocare uno spazio di menmoria tramite il “sub” ad “esp” allocandogli uno spazio di memoria.