MALWARE ANALYSIS

ASSEMBLY X86

S10/L3

NELL'ESERCIZIO DI OGGI ANDIAMO AD ANALIZZARE IL CODICE ASSEMBLY PER LA CPU X86

Traccia:

Nella lezione teorica del mattino, abbiamo visto i fondamenti del linguaggio Assembly. Dato il codice in Assembly per la CPU x86 allegato qui di seguito, identificare lo scopo di ogni istruzione, inserendo una descrizione per ogni riga di codice. Ricordate che i numeri nel formato 0xYY sono numeri esadecimali. Per convertirli in numeri decimali utilizzate pure un convertitore online, oppure la calcolatrice del vostro computer (per programmatori).

```
0×00001141 <+8>:     mov    EAX,0×20
0×00001148 <+15>:    mov    EDX,0×38
0×00001155 <+28>:    add    EAX,EDX
0×00001157 <+30>:    mov    EBP, EAX
0×0000115a <+33>:    cmp    EBP,0xa
0×0000115e <+37>:    jge    0×1176 <main+61>
0×0000116a <+49>:    mov    eax,0×0
0×0000116f <+54>:    call   0×1030 <printf@plt>
```

Cerchiamo di identificare lo scopo di ogni istruzione, facciamo una descrizione per ogni riga del nostro codice.

ISTRUZIONE	DESCRIZIONE
0x00001141 <+8>: movEAX,0x20	Con l'istruzione mov andiamo a spostare il valore 0x20(esadecimale) o 32(decimale) nel registro di memoria EAX.
0x00001148 <+15>: mov EDX,0x38	Con l'istruzione mov andiamo a spostare il valore 0x38(esadecimale) o 56(decimale) nel registro di memoria EDX.
0x00001155 <+28>: add EAX,EDX	L'istruzione add somma il valore del registro EDX e di EAX, 56+32=88(decimale) oppure 0x38+0x20=58 (esadecimale) e aggiorna il registro EAX con la somma 88
0x00001157 <+30>: mov EBP, EAX	Con l'istruzione mov andiamo a spostare il contenuto del registro EAX con valore 88 (decimale) o 58 (esadecimale) nel registro di memoria EBP.
0x0000115a <+33>: cmp EBP,0xa	Con il comando cmp andiamo a comparare se il numero è 0, ha avuto un riporto oppure nessuno dei due. Nel nostro caso abbiamo: EBP 88 >10 (decimale) oppure 0xa (esadecimale) che ci setta la Zero Flag a 0 e la Carry Flag a 0.
0x0000115e <+37>: jge 0x1176 <main+61>	Con il comando jge andiamo a fare un salto nell'indirizzo di memoria 0x1176 solo nel caso in cui la destinazione sia di valore maggiore o uguale alla sorgente nell'istruzione cmp. Visto che 88 > 10 , viene effettuato il salto.
0x0000116a <+49>: mov eax,0x0	Con l'istruzione mov andiamo a spostare 0x0(esadecimale) o 0(decimale) nel registro di memoria EAX.
0x0000116f <+54>: call 0x1030 <printf@plt>	Con l'istruzione call andiamo ad effettuare una chiamata corrispondete alla funzione di print, ovvero di una stampa a schermo di un codice e di una sotto funzione.