MALWARE ANALYSIS: ASSEMBLY FONDAMENTI EPICODE

Giovanni Pisapia

ANALISI ISTRUZIONI CODICE ASSEMBLY

Esercizio :

- L'obbiettivo dell' esercizio di oggi era di comprendere lo scopo di ogni istruzione del codice Assembly fornito.

Analisi istruzioni codice:

- 0x00001141 <+8>: Mov EAX, 0x20: Questa istruzione sposta il valore esadecimale 0x20 (32 in decimale) nel registro EAX. Risultato: EAX = 32.
- 0x00001148 <+15>: Mov EDX, 0x38: Questa istruzione sposta il valore esadecimale 0x38 (56 in decimale) nel registro EDX. Risultato: EDX = 56.
- 0x00001155 <+28>: Add EAX, EDX: Questa istruzione somma il valore contenuto nel registro EDX (56) al valore contenuto nel registro EAX (32) e salva il risultato (88) nel registro EAX. Risultato: EAX = 88.
- 0x00001157 <+30>: Mov EBP, EAX: Questa istruzione sposta il valore contenuto nel registro EAX (88) nel registro EBP. Risultato: EBP = 88.
- 0x0000115a <+33>: Mov EBP, 0xa: Questa istruzione sposta il valore esadecimale 0xa (10 in decimale) nel registro EBP. Risultato: EBP = 10.
- 0x0000115e <+37>: Cmp EBP, 0xa: Questa istruzione confronta il valore contenuto nel registro EBP (10) con il valore esadecimale 0xa (10 in decimale). Nel caso dell'istruzione "cmp EBP, 0xa", viene confrontato il valore contenuto nel registro EBP con il valore esadecimale 0xa. Il confronto avviene sottraendo il secondo valore dal primo.

  Se il risultato della sottrazione è uguale a zero, significa che i due valori sono uguali.
- 0x0000116a <+49>: Jge 0x1176 : si basa sul valore della flag JGE. Se la flag è impostata (cioè il confronto è risultato uguale), viene eseguito il salto all'indirizzo 0x1176 come in questo caso.
- 0x0000116f <+54>: Mov EAX, 0x0: Questa istruzione sposta il valore esadecimale 0x0 (0 in decimale) nel registro EAX. Risultato: EAX = 0.
- 0x00001174 <+59>: Call 0x1030 printf@plt Questa istruzione chiama la funzione printf situata all'indirizzo 0x1030 (printf@plt).