Malware analysis - Assembly x86

Sommario

Traccia S10-L3

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).

```
0x00001141 <+8>:     mov    EAX,0x20

0x00001148 <+15>:    mov    EDX,0x38

0x00001155 <+28>:    add    EAX,EDX

0x00001157 <+30>:    mov    EBP, EAX

0x0000115a <+33>:    cmp    EBP,0xa

0x0000115e <+37>:    jge    0x1176 <main+61>

0x0000116a <+49>:    mov    eax,0x0

0x0000116f <+54>:    call   0x1030 <printf@plt>
```

Analisi del codice

```
0x00001141 <+8>:     mov    EAX,0x20  // Copia il valore esadecimale 0x20 (32 in decimale) nel registro EAX

0x00001148 <+15>:    mov    EDX,0x38  // Copia il valore esadecimale 0x38 (56 in decimale) nel registro EDX

0x00001155 <+28>:    add    EAX,EDX   // Somma il valore contenuto in EDX ad EAX e salva il risultato in EAX

0x00001157 <+30>:    mov    EBP, EAX   // Copia il contenuto del registro EAX nel registro EBP

0x0000115a <+33>:    cmp    EBP,0xa    // Confronta il valore in EBP con 0xa (10 in decimale)

0x0000115e <+37>:    jge    0x1176 <main+61>  // Se il valore in EBP è maggiore o uguale a 0xa, salta all'istruzione a 0x1176

0x0000116a <+49>:    mov    eax,0x0    // Copia il valore 0 nel registro EAX

0x0000116f <+54>:    call   0x1030 <printf@plt>  // Chiama la funzione printf
```