



S10-L3 Emanuele Di Stefano

Analisi del Codice Assembly





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

Questa istruzione sposta il valore immediato 0x20 (che è 32 in decimale) nel registro EAX.

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

Questa istruzione sposta il valore immediato 0x38 (che è 56 in decimale) nel registro EDX.

0x00001155 <+28>: add EAX,EDX

Questa istruzione aggiunge il valore nel registro EDX al valore nel registro EAX e memorizza il risultato in EAX.

0x00001157 <+30>: mov EBP,EAX

Questa istruzione sposta il valore dal registro EAX nel registro EBP.

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

Questa istruzione confronta il valore nel registro EBP con il valore immediato 0xa (che è 10 in decimale).

0x0000115c <+37>: jge 0x1176 <main+61>

Questa istruzione salta all'indirizzo 0x1176 (l'istruzione a main+61) se il risultato del confronto precedente indica che EBP è maggiore o uguale a 10.

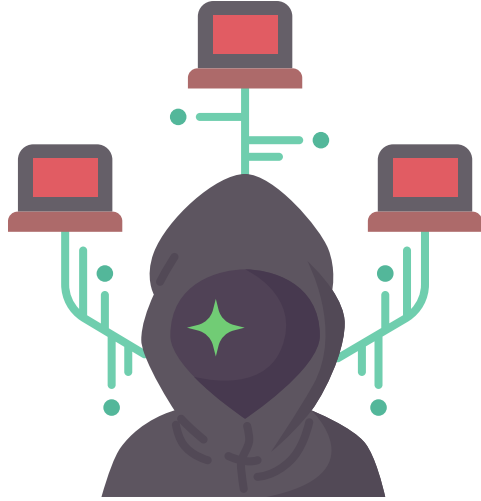
0x0000116a <+49>: mov eax,0x0

Questa istruzione sposta il valore immediato 0x0 (che è 0 in decimale) nel registro EAX.

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

Questa istruzione chiama la funzione printf, che si trova all'indirizzo 0x1030.

Riassunto



Inizializzazione

- EAX viene inizializzato a 32 (0x20).
- EDX viene inizializzato a 56 (0x38).

Addizione

- EAX viene sommato a EDX, risultando in EAX che contiene 88 (32 + 56).

Movimento e Confronto

- Il risultato in EAX (88) viene spostato in EBP.
- EBP viene confrontato con 10.

Salto Condizionale

- Se EBP (che è 88) è maggiore o uguale a 10, il codice salta a main+61.

Impostare EAX a Zero e Chiamare printf

- EAX viene impostato a 0.
- Viene chiamata la funzione printf.