

# Assembly x86

Istruzione

0x00001141 <+8>: mov EAX,0x20 » copia il valore esadecimale della sorgente 0x20 nel registro EAX

0x00001141 <+15>: mov EDX,0x38 » copia il valore esadecimale della sorgente 0x38 nel registro EDX

0x00001155 <+28>: add EAX,EDX » somma il valore della sorgente EDX con quello del destinatario EAX

0x00001157 <+30>: mov EBP,EAX » copia il valore EAX nel registro destinatario EBP

0x0000115a <+33>: cmp EBP, 0xa » mette a confronto destinatario e sorgente per restituire il valore binario (ZF 0 – CF 0)

0x0000115e <+37>: jge 0x1176 <main+61> » salta alla locazione di memoria 0x1176 se la destinazione è > = della sorgente *cmp*

0x0000116a <+49>: mov EAX,0x0 » copia il valore esadecimale della sorgente 0x0 nel registro EAX

0x0000116f <+54>: call 0x1030 <printf@plt> » l'istruzione *call* chiama una funzione facendo passare l'esecuzione del programma alla funzione chiamata, creando un nuovo *stack*

A= 32

C= 56

Somma= A+C= 88

E= A= 88

E (88) -0xA (10) - if E (88) >= 0xa (10) - (ZF 0 – CF 0)

then E >= 0xa – condizione verificata, avviene il salto e, forse, l'istruzione

A = 0

Call 0 x 1030 <print@pit>