3. Malware analysis: Assembly fondamenti

Identificare lo scopo di ogni istruzione dei seguenti codici in Assembly.

0×00001141 <+8>: mov EAX,0×20
Una volta trovato il valore decimale, cioè 32, possiamo capire che lo sposta nel registro EAX.

0×00001148 <+15>: mov EDX,0×38
Una volta trovato il valore decimale, cioè 56, possiamo capire che lo sposta nel registro EDX

0×00001155 <+28>: add EAX,EDX
Somma i due registri cioè 32 + 56 ed aggiorna il registro AEX col valore 88

0×00001157 <+30>: mov EBP, EAX
Muove il contenuto di EAX che è uguale a 88 nel registro EBP

0×0000115a <+33>: mp EBP,0xa
Compara il valore 0xa=10 col valore nel registro EBP=88

0×0000115e <+37>: jge 0x1176 <main+61>
Salto all'allocazione specificata se la destinazione è maggiore o uguale alla sorgente dell'allocazione cmp, quindi essendo 88>10 il salto viene effettuato

0×0000116a <+49>: mov eax,0x0
Sposta il valore 0 nel registro AEX sovrascrivendolo

0×0000116f <+54>: call 0x1030 <printf@plt>
Chiamata della funzione printf