METASPLOIT – MODULI MS08-067 E MS17_010

Traccia:

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).

1. 0x00001141 <+8>: mov EAX,0x20
2. 0x00001148 <+15>: mov EDX,0x38
3. 0x00001155 <+28>: add EAX,EDX
4. 0x00001157 <+30>: mov EBP, EAX
5. 0x0000115a <+33>: cmp EBP,0xa
6. 0x0000115e <+37>: jge 0x1176 <main+61>
7. 0x0000116a <+49>: mov eax,0x0
8. 0x0000116f <+54>: call 0x1030 <printf@plt>

Esecuzione:

1. **0x00001141 <+8>: mov EAX,0x20**

Con il comando **mov** sposto il valore 0x20(32) nel registro **EAX**

2. **0x00001148 <+15>: mov EDX,0x38**

Con il comando **mov** sposto il valore 0x38(56) nel registro **EAX**

3. **0x00001155 <+28>: add EAX,EDX**

Viene sommato il valore contenuto nel registro **EDX** con il valore contenuto nel registro **EAX** e aggiornato quest'ultimo

4. **0x00001157 <+30>: mov EBP, EAX**

sposto il valore contenuto nel registro **EAX** nel registro **EBP**

5. **0x0000115a <+33>: cmp EBP,0xa**

Verifico la condizione: Il valore contenuto nel registro **EBP** (attualmente 0x58, cioè 88) è maggiore del valore 0xa (ovvero 10)? In questo caso sì

6. **0x0000115e <+37>: jge 0x1176 <main+61>**

Essendo la condizione del punto **5.** si può procedere con la seguente istruzione

7. **0x0000116a <+49>: mov eax,0x0**

Sposto il valore 0x0 (essere 0) al registro **EAX**

8. **0x0000116f <+54>: call 0x1030 <printf@plt>**