Cuore Andrea

11/01/2023

Malware Analysis : Studio codice Assembly

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:
  - EAX = 32
- 0x00001148<+15> MOV EDX,0x38:
  - EDX = 56
- 0x00001155<+28> ADD EAX,EDX
  - EAX + EDX => 32 + 56 => EAX = 88
- 0x00001157<+30> MOV EBP, EAX:
  - EBP = 88
- 0x0000115a<+33> CMP EBP,0xa:
  - Compara EBP con 10
- 0x0000115e<+37> JGE 0x1176 <main +61>
  - Nella condizione di confronto se la comparazione riporta EBP > 10, vai al label 0x1176
- 0x0000116a<+49> MOV EAX,0x0:
  - EAX = 0
- 0x0000116f<+54> CALL 0x1030 <printf@plt>:
  - Invoca il label prinf@plt

Traduzione in linguaggio di alto livello (C).

N.B: Non conoscendo il codice intero darò dei valori fittizi alle label richiamate.

```
#include <stdio.h>

void print_plt(int);

int main()

{

   int c = 2;

   int a = 3;

   int EAX = 32;

   int EDX = 56;

   int EBP;

   EAX = EAX + EDX;

   EBP = EAX;

   LOOP: do {

     if( EBP < 10) { //IN QUESTO IF BISOGNA METTERE EBP > 10 ESSENDO JGE, METTO MINORE PER USCIRE DAL LOOP

        goto LOOP;

     }

   EAX = 0;

   print_plt(a);


   return 0;

}


void print_plt(int a){

   a = a*a;

   printf("\n%d",a);

}
```

IL SEGUENTE CODICE NON E' ASSOLUTAMENTE OTTIMIZZATO, SERVE SOLO PER DARE UN IDEA DEL CODICE IN ASSEMBLY.