Computer Security HW0x2

R08946007 資科所 蔡昀達

Nickname：r08946007

1. 使用 ida pro & hex ray decompiler 先 decompile IDAmudamudamuda.exe 可以得到原始碼大概理解這隻程式的功能

```
int sub_13312F0()
{
  int v1; // [esp+0h] [ebp-2Ch]
  char v2; // [esp+4h] [ebp-28h]

  sub_1331000();
  puts("Oh, it's another ez Reversing Challenge");
  puts("Leverage all you learned in class to solve this one");
  puts("2 step to get the flag: \n\n\n");
  puts("First, give me the seed: ");
  scanf("%d", &v1);
  sub_1331070(v1);
  puts("\n\n\nOK then whats the flag ?");
  scanf("%s", &v2);
  if ( sub_1331270(&v2) )
  {
    puts("OH yeah, you got the flag\n\n\n");
  }
  else
  {
    puts("That's not right, go back to review the slides on htt
    puts("Hint: file format, calling convention, shellcode, sta
  }
  system("pause");
  return 0;
}
```

2. 找到 seed 跟 flag 兩個涵式

Flag:

```
int __cdecl sub_1331270(const char *a1)
{
  void *v2; // eax

  if ( strlen(a1) != 32 )
    return 0;
  v2 = VirtualAlloc(0, 0xC8u, 0x1000u, 0x40u);
  qmemcpy(v2, &unk_1334058, 0xC8u);
  return ((int (__cdecl *)(const char *, void *))v2)(a1, &unk_1334018);
}
```

可以發現 flag 的涵式將 &4058 位置載入記憶體並做為涵式執行

&4058 記憶體位置內容為一段 shellcode

輸入的 flag 和 &4018 位置的內容做為參數傳進涵式

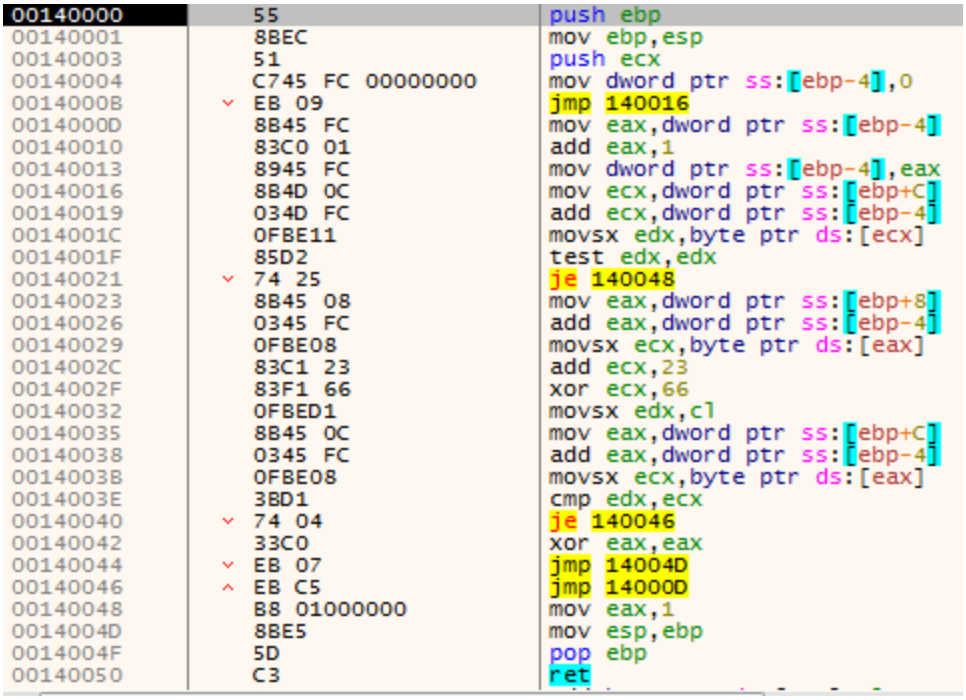
Seed:

```
4  int result; // eax
5  int v3; // [esp+4h] [ebp-28h]
6  char *v4; // [esp+8h] [ebp-24h]
7  int m; // [esp+14h] [ebp-18h]
8  int i; // [esp+18h] [ebp-14h]
9  signed int k; // [esp+1Ch] [ebp-10h]
10 int v8; // [esp+20h] [ebp-Ch]
11 int j; // [esp+24h] [ebp-8h]
12 int l; // [esp+24h] [ebp-8h]
13
14 v1 = (char *)dword_1334488(0);
15 dword_133448C = (int)v1;
16 result = *v1;
17 if ( result == 77 )
18 {
19   result = *(char *)(dword_133448C + 1);
20   if ( result == 90 )
21   {
22     v4 = (char *)(*(_DWORD *)(dword_133448C + 60) + dword_133448C);
23     result = *v4;
24     if ( result == 80 )
25     {
26       result = v4[1];
27       if ( result == 69 )
28       {
29         for ( i = (int)(v4 + 248); ; i += 40 )
30         {
31           v3 = strcmp((const char *)i, ".data");
32           if ( v3 )
33             v3 = -(v3 < 0) | 1;
34           if ( !v3 )
35             break;
```

可以發現 seed 涵式只有在最下面的地方使用到輸入的 seed 變數
觀察可以發現他在更改某一段記憶體
正好是 flag 涵式中 shellcode 的位置
因此可以理解為 seed 改寫 &4058 的記憶體內容讓他成為一段 shellcode

3. 根據 x86 calling convention 合理猜測 shellcode 開頭為 push 結尾為 ret 來反推 seed，可以得到正確的 seed 為 16

4. 取的 seed 之後就得到了 shellcode 然後反組譯之後
   可以發現這段組語將 &4018 的記憶體內容 add 23 然後 xor 66 就會是正確的 flag



5. 最終得到 flag : FLAG{y3s!!y3s!!y3s!!0h_my_g0d!!}