Easy_KeygenMe 문제 write-up

강승민

```
Input Name:

asd
Input Serial: asd
Wrong
```

프로그램을 실행 시키면 이렇게 입력을 두개 받고 결과를 출력 해주는데 readme파일을 읽어보니

ReversingKr KeygenMe

Find the Name when the Serial is 5B134977135E7D13

일정 문자열을 넣으면 그에 대한 시리얼 값을 입력해서 알맞으면 다른 결과를 출력해주는 것 같다. 이 문제에서는 5B134977135E7D13에 대한 입력을 찾으면 되는 것 같다.

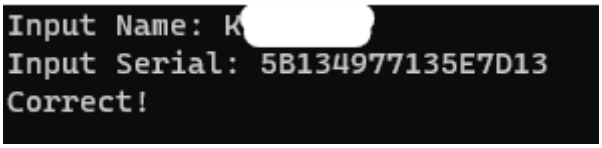
```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    signed int v3; // ebp
4    int i; // esi
5    char v6; // [esp+Ch] [ebp-130h]
6    char v7[2]; // [esp+Dh] [ebp-12Fh] BYREF
7    char v8[100]; // [esp+10h] [ebp-12Ch] BYREF
8    char Buffer[197]; // [esp+74h] [ebp-C8h] BYREF
9    __int16 v10; // [esp+139h] [ebp-3h]
10   char v11; // [esp+13Bh] [ebp-1h]
11
12   memset(v8, 0, sizeof(v8));
13   memset(Buffer, 0, sizeof(Buffer));
14   v10 = 0;
15   v11 = 0;
16   v6 = 16;
17   qmemcpy(v7, " 0", sizeof(v7));
18   puts(aInputName);
19   scanf("%s", v8);
20   v3 = 0;
21   for ( i = 0; v3 < (int)strlen(v8); ++i )
22   {
23     if ( i >= 3 )
24       i = 0;
25     sprintf(Buffer, "%s%02X", Buffer, v8[v3++] ^ v7[i - 1]);
26   }
27   memset(v8, 0, sizeof(v8));
28   puts(aInputSerial);
29   scanf("%s", v8);
30   if ( !strcmp(v8, Buffer) )
31     puts(aCorrect);
32   else
33     puts(aWrong);
34   return 0;
35 }
```

IDA를 이용해 열어보니 위 그림과 같고 이를 해석해보면 입력한 Name의 문자열을 3자리 배열인 v7배열과 순서대로 XOR 연산을 하여 Buffer에 넣고 이를 입력한 Serial값과 비교하는 것이라 볼 수 있다.

```
mov     [esp+140h+var_130], 10h
mov     [esp+140h+var_12F], 20h ; ' '
mov     [esp+140h+var_12E], 30h ; '0'
```

V7 배열은 0x10, 0x20, 0x30이고 XOR연산은 다시 한번 하면 원래의 값을 얻을 수 있기에 5B134977135E7D13 을 순서대로 XOR연산 해주면

이렇게 일정 문자열을 얻을 수 있고



프로그램에서 입력해도 Correct! 가 출력되며 맞는 답인 것을 알 수 있다.