Rev-basic-5문제 write-up

강승민

```
1  int __cdecl main(int argc, const char **argv, const char **e
2  {
3    char v4[256]; // [rsp+20h] [rbp-118h] BYREF
4
5    memset(v4, 0, sizeof(v4));
6    sub_1400011C0("Input : ", argv, envp);
7    sub_140001220("%256s", v4);
8    if ( (unsigned int)sub_140001000(v4) )
9      puts("Correct");
10   else
11     puts("Wrong");
12   return 0;
13 }
```

문제파일을 IDA를 이용해 열어보면 sub_140001000 함수의 결과에 따라 Correct, Wrong 이 출력되는데 해당함수를 열어보면

```
1  __int64 __fastcall sub_140001000(__int64 a1)
2  {
3    int i; // [rsp+0h] [rbp-18h]
4
5    for ( i = 0; (unsigned __int64)i < 0x18; ++i )
6    {
7      if ( *(unsigned __int8 *)(a1 + i + 1) + *(unsigned __int8 *)(a1 + i) != byte_140003000[i] )
8        return 0i64;
9    }
10   return 1i64;
11 }
```

입력한 배열의 글자와 다음 글자의 합을 어떤 배열과 비교한다. 이를 반대로 얻어내기 위해 생각을 해보면 배열의 끝은 0x0이기 때문에 뒤에서부터 글을 복구하면 Flag를 얻어 낼 수 있다.

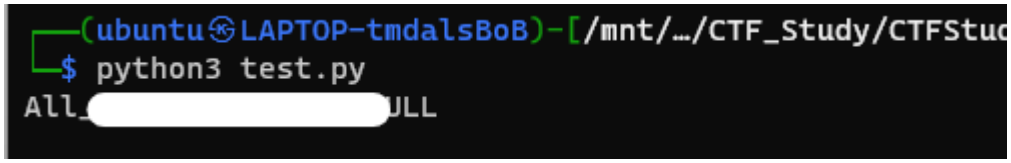
```
; unsigned __int8 byte_140003000[32]
byte_140003000  db 0ADh, 0D8h, 2 dup(0CBh), 9Dh, 97h, 0CBh, 0C4h, 92h
                                          ; DATA XREF: sub_140001000+48↑o
                db 0A1h, 0D2h, 0D7h, 0D2h, 0D6h, 0A8h, 0A5h, 0DCh, 0C7h
                db 0ADh, 0A3h, 0A1h, 98h, 4Ch, 9 dup(0)
```

배열은 이렇게 생겼고 이를 이용해 파이썬 코드를 짜보면

```
temp = [0xAD,0xD8,0xCB,0xCB,0x9D,0x97,0xCB,0xC4,0x92,0xA1,0xD2,0xD7,0xD2,0xD6,0xA8,0xA5,0xDC,0xC7,0xAD,0xA3,0xA1,0x98,0x4C,0x0

asdf = 0
result = []
for i in range(0x18,-1,-1):
    result.append(chr(temp[i]-asdf))
    asdf = temp[i]-asdf
for i in reversed(result):
    print(i,end="")
```

이렇게 짤 수 있다. 실행해보면



이렇게 문자열을 복구하여 flag를 얻어낼 수 있다.