一道加密题目

首先看输入

```
void __fastcall __noreturn main(int a1, char **a2, char **a3)
{
  int i; // [rsp+8h] [rbp-48h]
  char s[40]; // [rsp+20h] [rbp-30h] BYREF
  unsigned __int64 v5; // [rsp+48h] [rbp-8h]

  v5 = __readfsqword(0x28u);
  __isoc99_scanf("%39s", s);
  if ( (unsigned int)strlen(s) != 32 )
  {
    puts("Wrong!");
    exit(0);
  }
  mprotect(&dword_400000, 0xF000uLL, 7);
  for ( i = 0; i <= 223; ++i )
    *((_BYTE *)sub_402219 + i) ^= 0x99u;
  sub_40207B(&unk_603170);
  sub_402219();
}
```

输入长度为32

然后是代码自修改

0x402219处的代码每个都异或0x99

得到

```
void __fastcall __noreturn sub_402219(__int64 a1)
{
  int i; // [rsp+1Ch] [rbp-D4h]
  char v2[200]; // [rsp+20h] [rbp-D0h] BYREF
  unsigned __int64 v3; // [rsp+E8h] [rbp-8h]

  v3 = __readfsqword(0x28u);
  sub_400A71((__int64)v2, (__int64)&unk_603170);
  sub_40196E((__int64)v2, a1);
  sub_40196E((__int64)v2, a1 + 16);
  for ( i = 0; i <= 31; ++i )
    ;
}
```

暂且不看，跳到前一个函数sub_40272b

```
IDA View-A   Pseudocode-A   Hex View-1   Structures
 1 unsigned __int64 __fastcall sub_40207B(__int64 a1)
 2 {
 3   char v2[16]; // [rsp+10h] [rbp-50h] BYREF
 4   __int64 v3; // [rsp+20h] [rbp-40h] BYREF
 5   __int64 v4; // [rsp+30h] [rbp-30h] BYREF
 6   __int64 v5; // [rsp+40h] [rbp-20h] BYREF
 7   unsigned __int64 v6; // [rsp+58h] [rbp-8h]
 8
 9   v6 = __readfsqword(0x28u);
10   sub_401CF9(&unk_603120, 64LL, v2);
11   sub_401CF9(&unk_603100, 20LL, &v3);
12   sub_401CF9(&unk_6030C0, 53LL, &v4);
13   sub_401CF9(&dword_4025C0, 256LL, &v5);
14   sub_401CF9(v2, 64LL, a1);
15   return __readfsqword(0x28u) ^ v6;
16 }
```

大概猜测对0x603100~之后的若干字节进行了处理
然后看sub_402219的函数
动调后发现
传参a1就是我们的输入

```
Legend: code, data, rodata, value
0x0000000000402262 in ?? ()
gdb-peda$ x/32bc 0x603170
0x603170:       0xcb    0x8d    0x49    0x35    0x21    0xb4    0x7a    0x4c
0x603178:       0xc1    0xae    0x7e    0x62    0x22    0x92    0x66    0xce
0x603180:       0x0     0x0     0x0     0x0     0x0     0x0     0x0     0x0
0x603188:       0x0     0x0     0x0     0x0     0x0     0x0     0x0     0x0
```

调试后发现 0x603170处是该值

看到a1和a1+16传入同一个函数，可以猜一下是分组加密方式
看了答案后发现是AES加密

```
from Crypto.Cipher import AES
from binascii import b2a_hex,a2b_hex
mode = AES.MODE_ECB

key = b'\xcb\x8d\x49\x35\x21\xb4\x7a\x4c\xc1\xae\x7e\x62\x22\x92\x66\xce'
text=
b'\xBC\x0A\xAD\xC0\x14\x7C\x5E\xCC\xE0\xB1\x40\xBC\x9C\x51\xD5\x2B\x46\xB2\xB9\x43\x4D\xE5
\x32\x4B\xAD\x7F\xB4\xB3\x9C\xDB\x4B\x5B'
encrypt = AES.new(key,mode)
cipher_text = encrypt.decrypt(text)
print(cipher_text)
```

```
thon.exe e:/python/virtual_environment/decrypt.py
b'flag{924a9ab2163d390410d0a1f670}'
```