buuctf-crackMe题解及感悟

原创

夏男人 于 2021-05-19 18:58:38 发布 149 收藏 1

分类专栏： 逆向工程

本文链接：https://blog.csdn.net/weixin_51373492/article/details/117041292

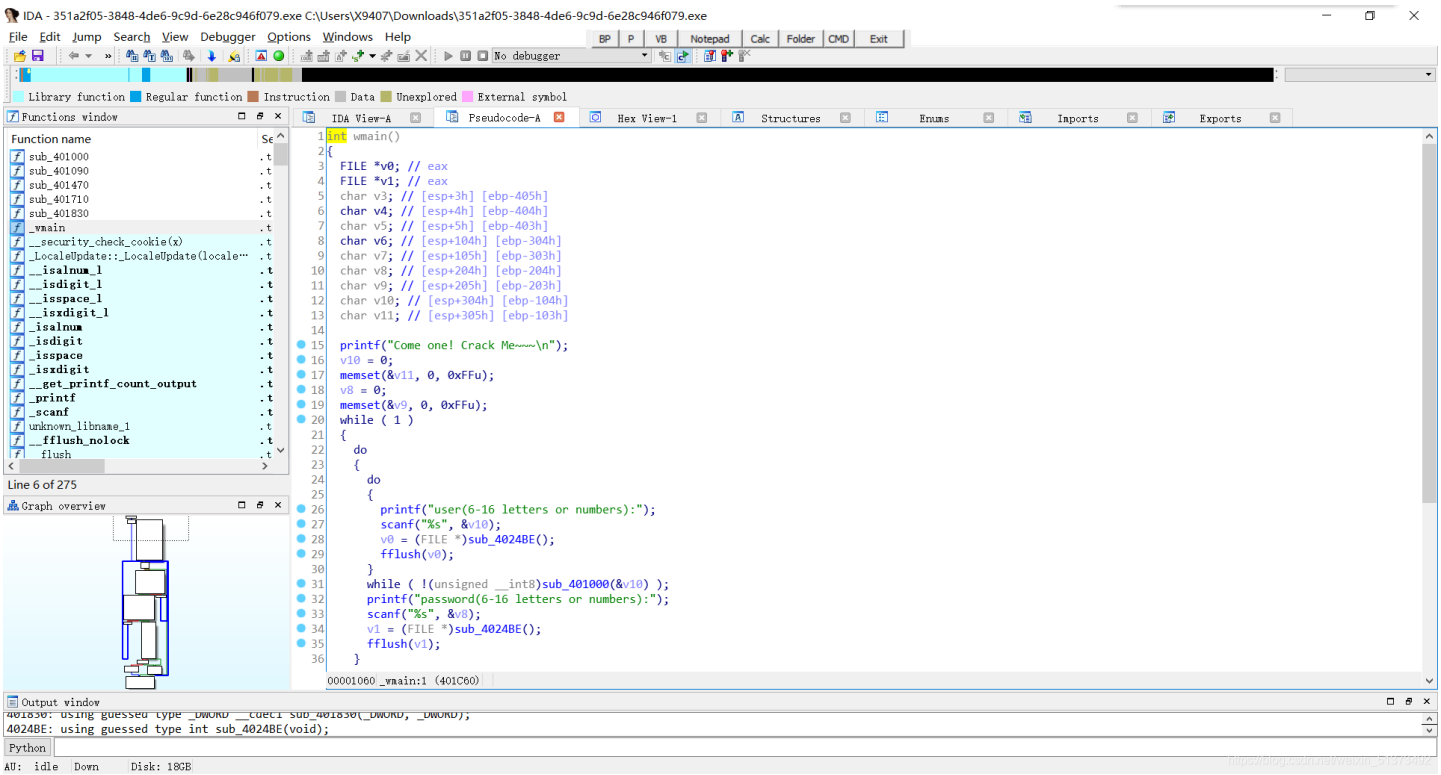
这道题花了我将近两天的时间，期间因为看wp完全看不懂疯狂请教学长，最终自己动手调试出来了。实在是不容易，也确实学到了许多东西，记录于此。

下载题目的程序后，惯例查壳，发现莫得

先打开程序，程序要我们输入用户名和密码，了解。

丢进ida

进入主函数wmain



把函数名字改成看得懂的单词

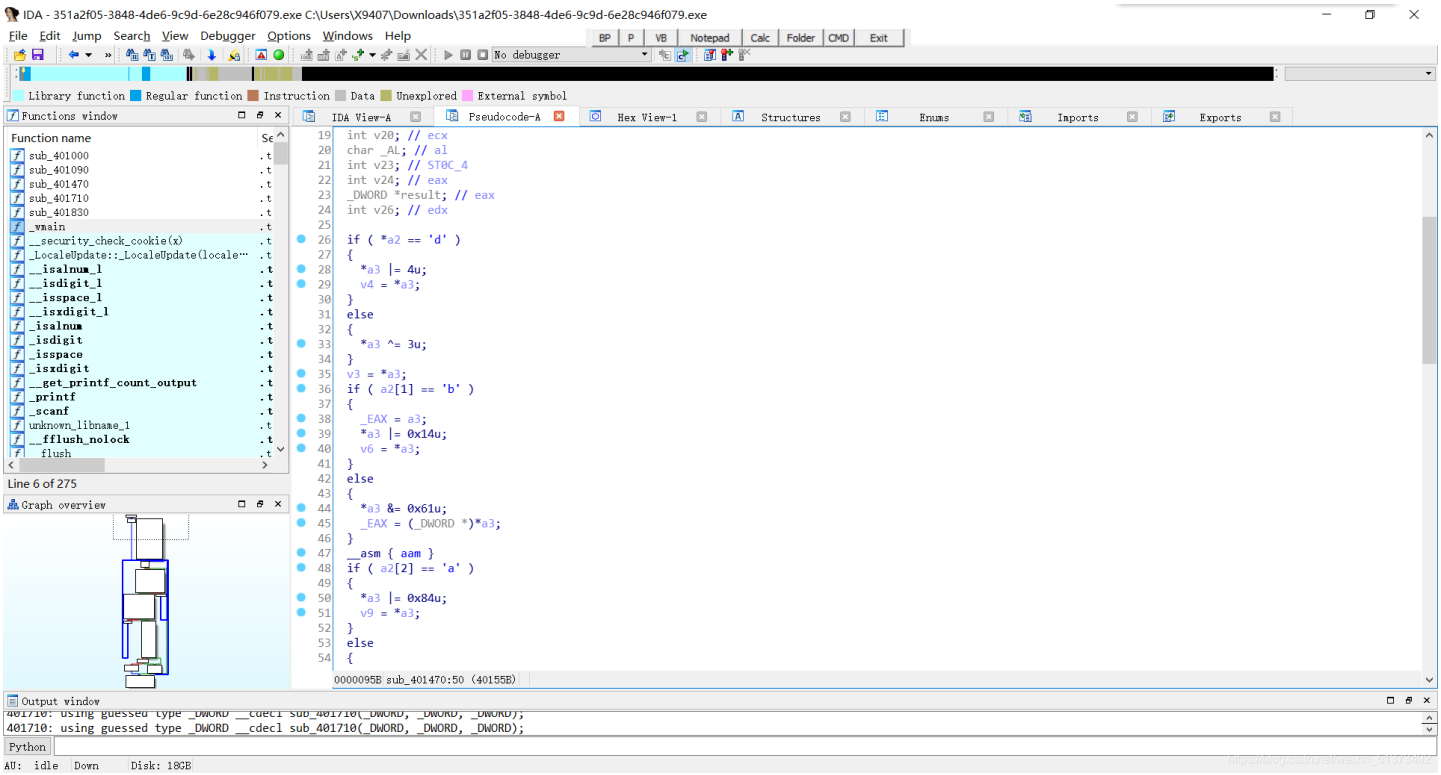
```
while ( 1 )
{
  do
  {
    do
    {
      printf("user(6-16 letters or numbers):");
      scanf("%s", &name);
      v0 = (FILE *)sub_4024BE();
      fflush(v0);
    }
    while ( !sub_401000(&name) );
    printf("password(6-16 letters or numbers):");
    scanf("%s", &password);
    v1 = (FILE *)sub_4024BE();
    fflush(v1);
  }
  while ( !sub_401000(&password) );
  sub_401090(&name);
  v6 = 0;
  memset(&v7, 0, 0xFFu);
  v4 = 0;
  memset(&v5, 0, 0xFFu);
  v3 = ((int (__cdecl *)(char *, char *))loc_4011A0)(&v6, &v4);
  if ( (unsigned __int8)sub_401830(&name, &password) )
  {
    if ( v3 )
      break;
  }
  printf(&v4);
}
printf(&v6);
return 0;
}
```


由此处的while和最里层的那个break可知，要从该循环跳出，这俩if都得满足

首先我们分析第一个if，点进sub_401830并开始分析

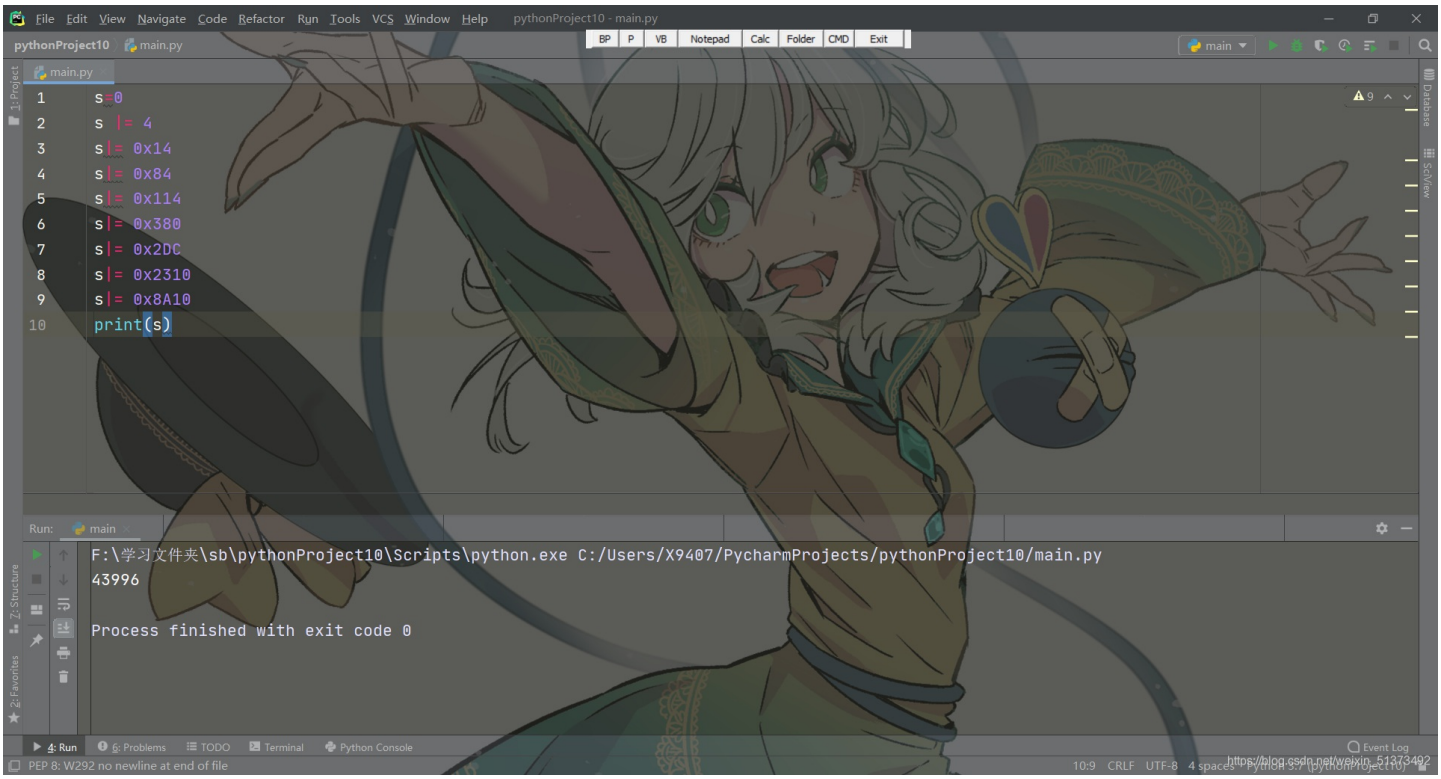
```
76  sub_401470(a1, &v17, &v14);
77  return v14 == 43924;
78}
```

因为这个if要成立，所以这个return后面的表达式必须成立，及v14==43924成立，我们此处逆向分析（字面意思）



此处出现了很多很多if判断，按照经验我们可以把if后面的数字按r转换成字符，此处我们猜测if全部满足后结果刚好等于43924（不猜搞不出，全部分析。。想想都恐怖）

可以写个python脚本验证一下我们的猜测



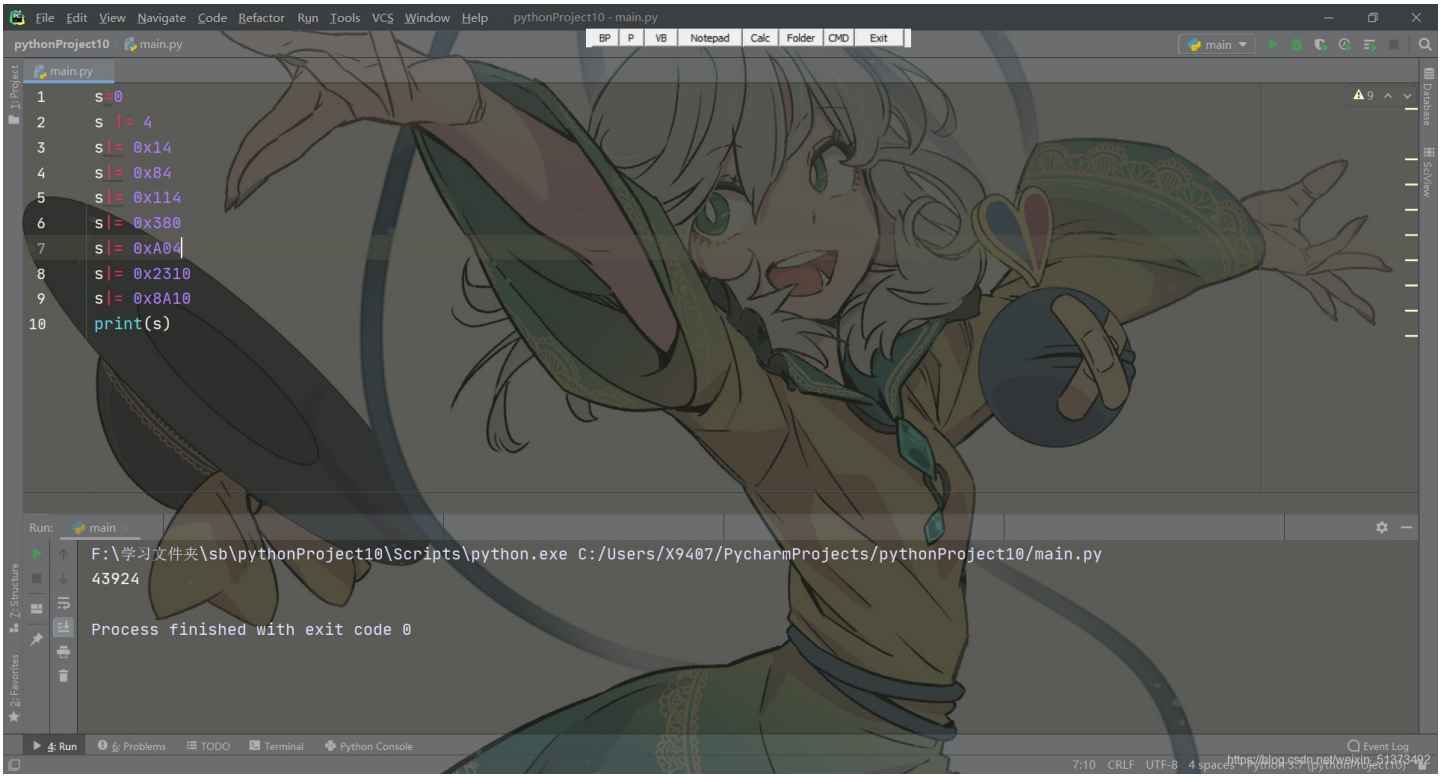
咦，答案不是43924，这是为什么呢（棒读）

原来是

```
80  if ( *(_DWORD *)(*(_DWORD *)(__readfsdword(0x30u) + 24) + 12) != 2 )
81  {
82    if ( a2[5] == 'f' )
83    {
84      *a3 |= 0x2DCu;
85      v17 = *a3;
86    }
87    else
88    {
89      *a3 |= 0x21u;
90    }
91    v16 = *a3;
92  }
93  if ( a2[5] == 's' )
94  {
95    *a3 |= 0xA04u;
96    v18 = (char)a3;
97    v20 = *a3;
98  }
99  else
```


这个地方有两个选择，我选了上面那个。

改成下面那个后



完美。由此得出v17为"dbappsec"

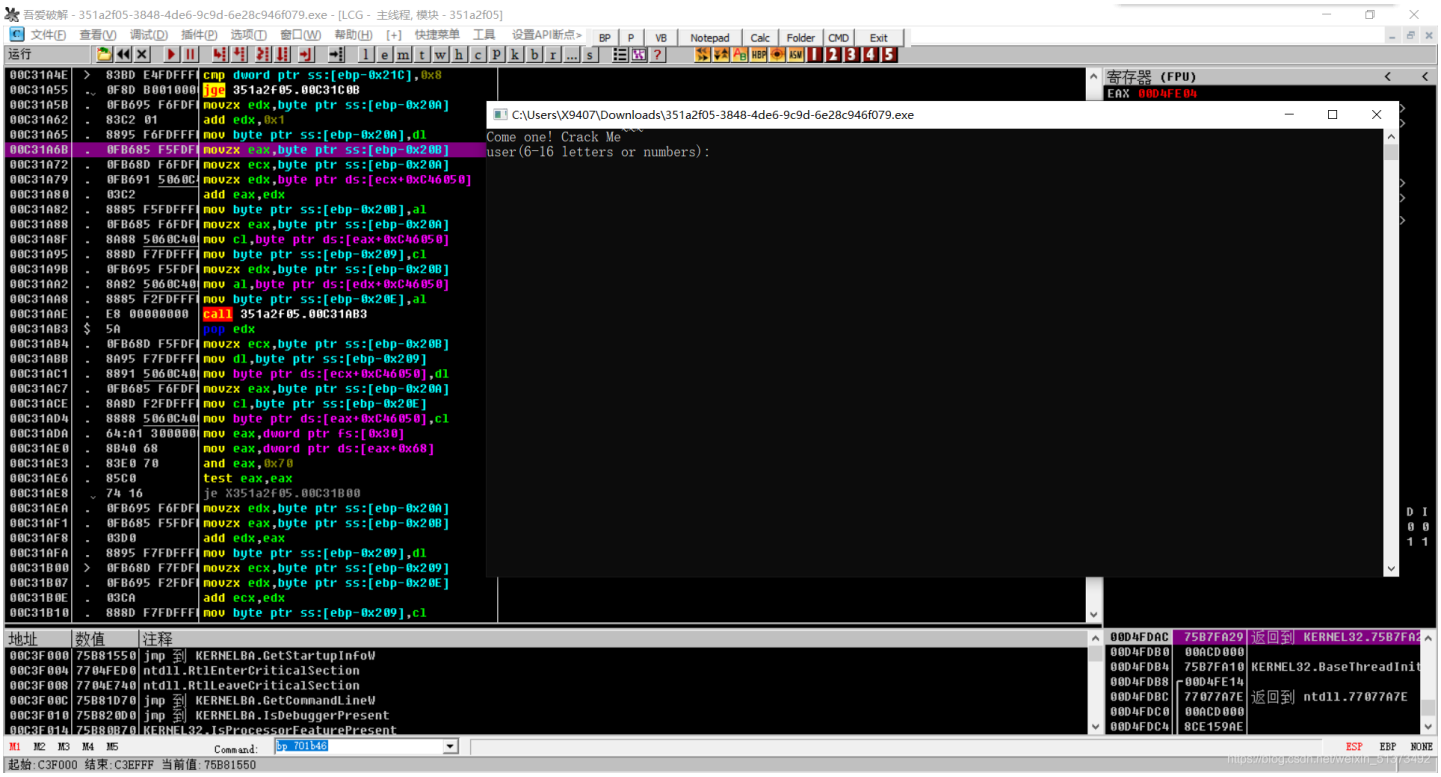
ok,继续反推

```
while ( (signed int)v6 < 8 )
{
  v11 += byte_416050[++v12];
  v13 = byte_416050[v12];
  v8 = byte_416050[v11];
  byte_416050[v11] = v13;
  byte_416050[v12] = v8;
  if ( *(_DWORD *)(__readfsdword(0x30u) + 104) & 0x70 )
    v13 = v11 + v12;
  *(&v17 + v6) = byte_416050[(unsigned __int8)(v8 + v13)] ^ *(&v15 + v5);
  if ( *(_DWORD *)(__readfsdword(0x30u) + 2) & 0xFF )
  {
    v11 = -83;
    v12 = 43;
  }
  sub_401710(&v17, name, v6++);
  v5 = v6;
  if ( v6 >= &v15 + strlen(&v15) + 1 - &v16 )
    v5 = 0;
}
```

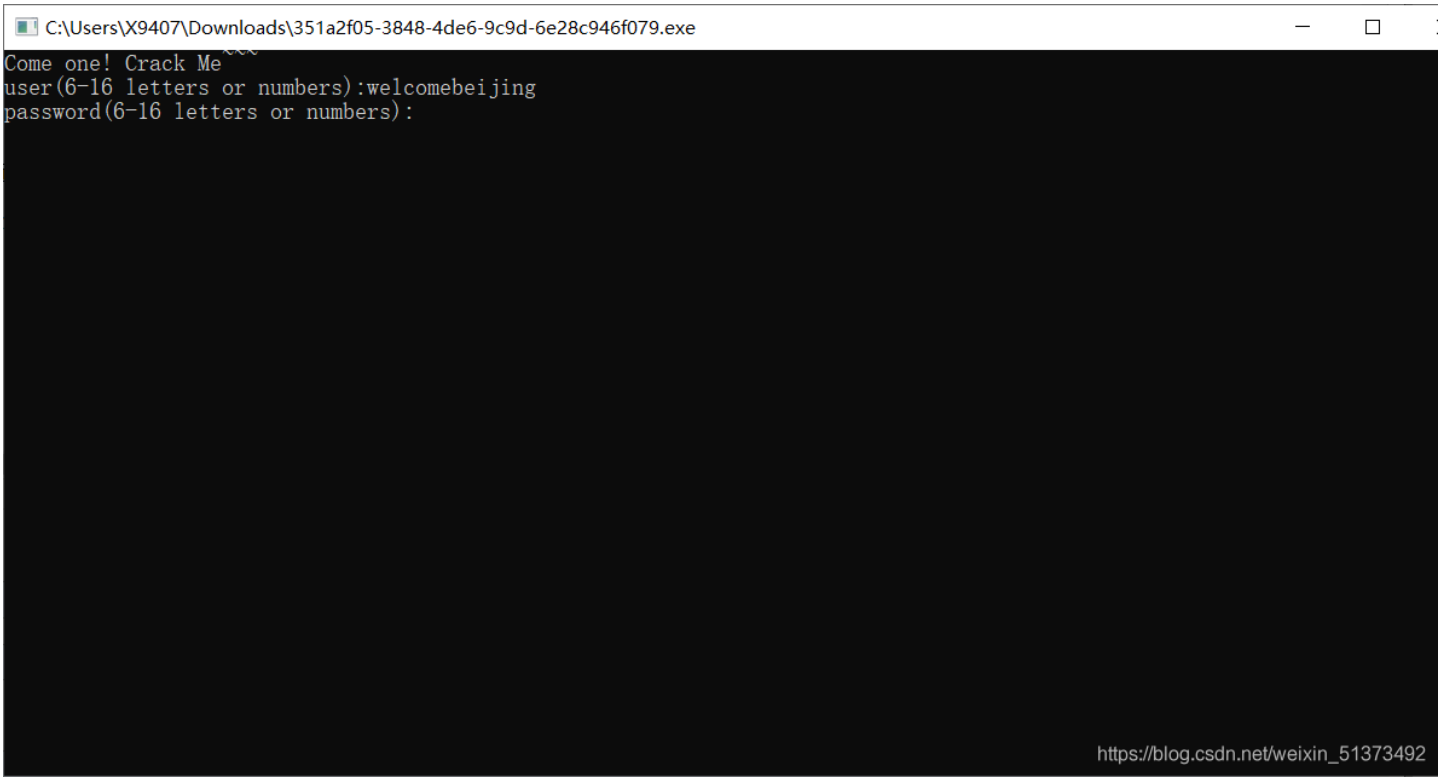
这俩我第一次做的时候分析好久，不知道这是啥玩意儿，上网查了查后发现是反调试的函数，直接不管。

因为我们已经得到v17，同时用户名也知道是welcomebeijing，而最开始说点开程序的时候要输入用户名。于是我们可以通过用ollydbg动态调试的方法得出byte_416050的值

拖进ollydbg，算好动态偏移后，执行程序



输入用户名 welcomebeijing



此时我们停下，找到ida的

```
    v13 = v11 + v12;
4   *(&v17 + v6) = byte_416050[(unsigned __int8)(v8 + v13)] ^ *(&v15 + v5);
```

的地址

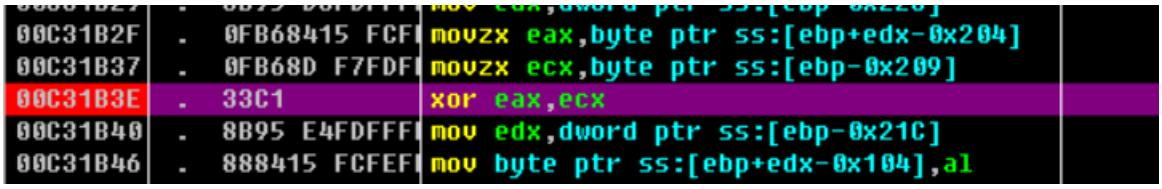
```
.text:00401B00
.text:00401B00 loc_401B00:                             ; CODE XREF: sub_401830+2B8↑j
.text:00401B00                 movzx   ecx, [ebp+var_209]
.text:00401B07                 movzx   edx, [ebp+var_20E]
.text:00401B0E                 add     ecx, edx
.text:00401B10                 mov     [ebp+var_209], cl
.text:00401B16                 movzx   eax, [ebp+var_209]
.text:00401B1D                 mov     cl, byte_416050[eax]
.text:00401B23                 mov     [ebp+var_209], cl
.text:00401B29                 mov     edx, [ebp+var_228]
.text:00401B2F                 movzx   eax, [ebp+edx+var_204]
.text:00401B37                 movzx   ecx, [ebp+var_209]
.text:00401B3E                 xor     eax, ecx
.text:00401B40                 mov     edx, [ebp+var_21C]
.text:00401B46                 mov     [ebp+edx+var_104], al
.text:00401B4D                 mov     eax, large fs:30h
.text:00401B53                 inc     eax
.text:00401B54                 inc     eax
.text:00401B55                 mov     eax, [eax]
```


上面那个xor eax，ecx即为关键步骤，象征着

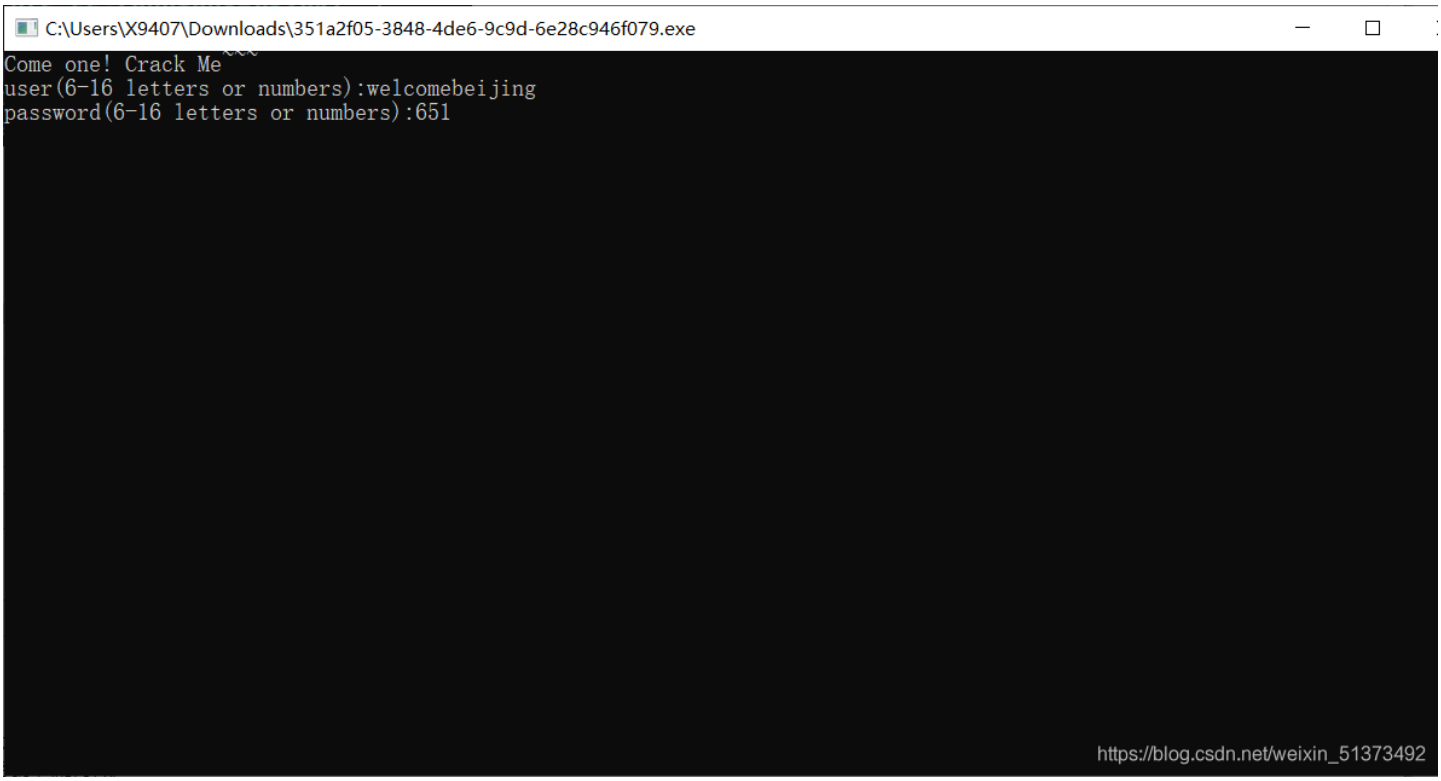
```
 v12;
 = byte_416050[(unsigned __int8)(v8 + v13)] ^ *(&v15 + v5);
```

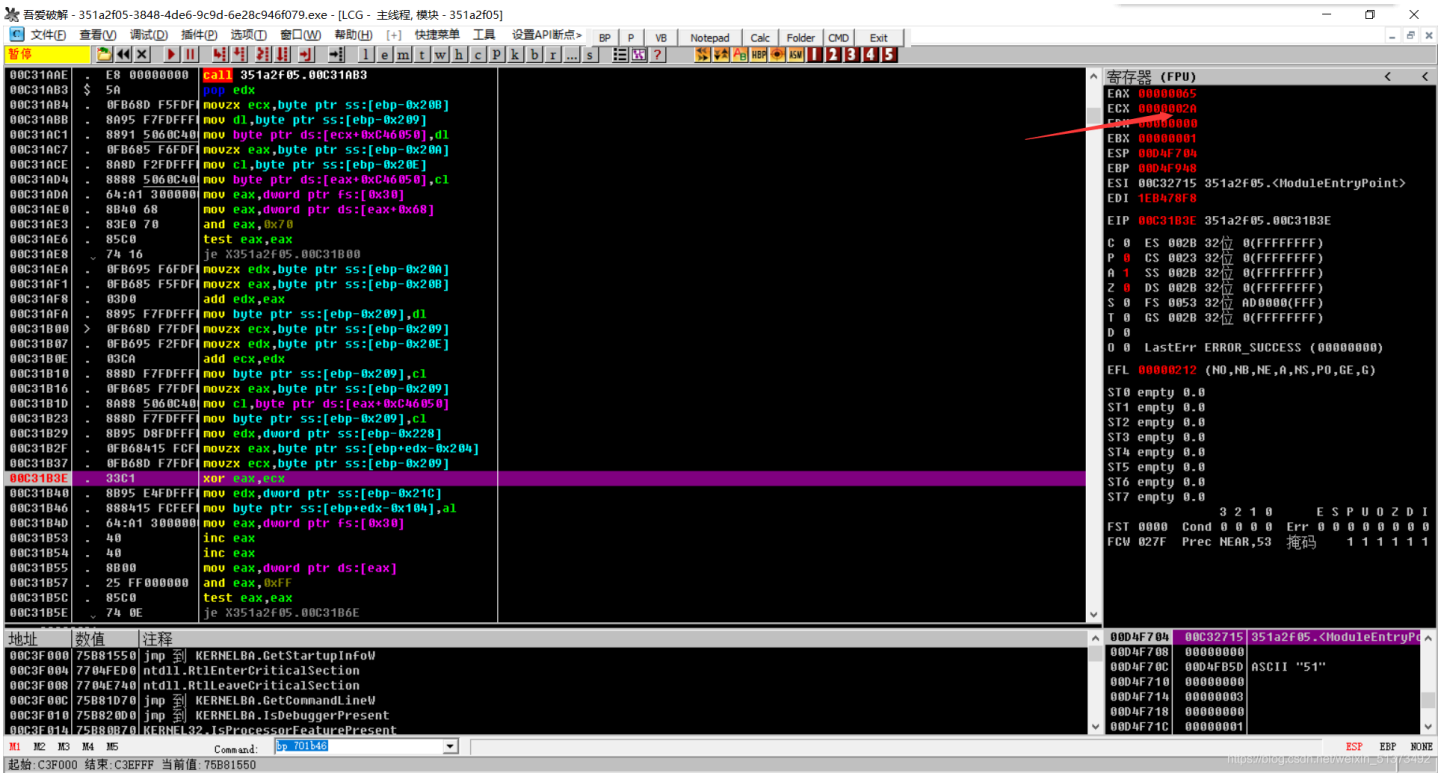
此处的异或，而通过上方的汇编代码我们可以知道上一行的 movzx ecx, [ebp+var_209]是将byte_416050的值存入ecx，此处的ecx的值即为我们想要知道的值！

由此，我们得到了思路，在xor处下断点，使每次运行时程序停在xor的前一命令，方便我们得知ecx的值。



然后回到程序，随便输入个密码（反正等下会断在那儿





看到了吗，ecx的值。

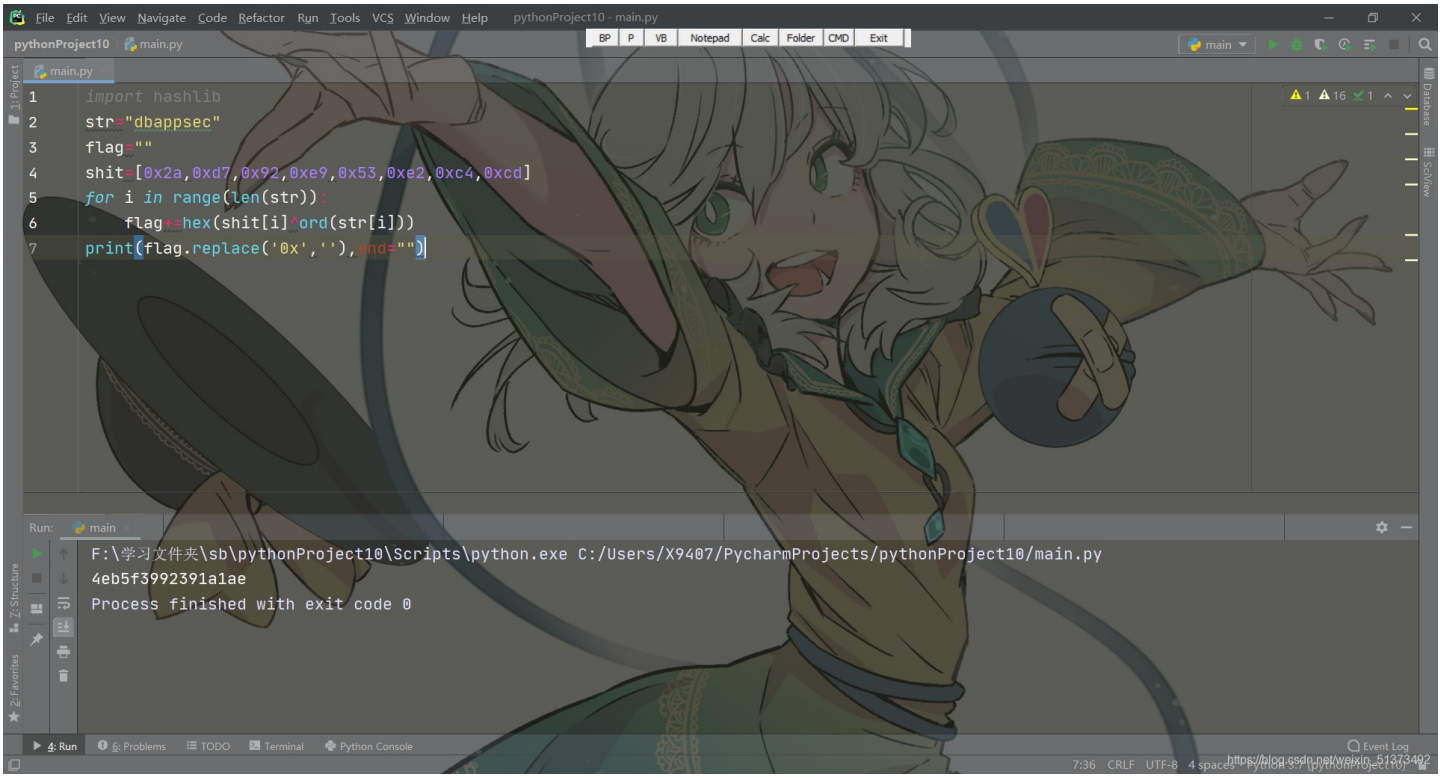
因为while（v6<8），所以执行了8次，此处我们继续f9执行，并记录下每次ecx的值。得到ecx，即byte_416050的存储的值0x2a,0xd7,0x92,0xe9,0x53,0xe2,0xc4,0xcd

然后继续回推

```
v4 = 0;
while ( v7 < strlen(password) )
{
  if ( isdigit(password[v7]) )
  {
    v9 = password[v7] - 48;          字符改10进制
  }
  else if ( isxdigit(password[v7]) )
  {
    if ( *(_DWORD *)(*(_DWORD *)(__readfsdword(0x30u) + 24) + 12) != 2 )
      password[v7] = 34;
    v9 = (password[v7] | 0x20) - 87;  16进制字符改16进制
  }
  else
  {
    v9 = ((password[v7] | 0x20) - 97) % 6 + 10;   不是abcdef的字符按照
  }                                               6个的循环转为16进制字
  v10 = v9 + 16 * v10;                            符
  if ( !((signed int)(v7 + 1) % 2) )
  {
    *(&v15 + v4++) = v10;
    a1 = v4;
    v10 = 0;
  }
  ++v7;
}
```

（我的表达方式可能有点奇怪，大家稍微谅解一下

下面那个if无所谓管不管，直接可以写代码了



把这串字符丢到网上的md5在线解密，得出flag{d2be2981b84f2a905669995873d6a36c}

虽然做的时候感觉啥都不知道，但是做完后发现自己基本理解了这道题，还是很开心的

感悟：

。

。。

恋恋真可爱

（无意识的赞美