

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](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

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

版权



[逆向工程 专栏收录该内容](#)

9 篇文章 0 订阅

订阅专栏

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, 也确实学到了很多, 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

```
1 int wmain()
2 {
3     FILE *v0; // eax
4     FILE *v1; // eax
5     char v3; // [esp+3h] [ebp-405h]
6     char v4; // [esp+4h] [ebp-404h]
7     char v5; // [esp+5h] [ebp-403h]
8     char v6; // [esp+104h] [ebp-304h]
9     char v7; // [esp+105h] [ebp-303h]
10    char v8; // [esp+204h] [ebp-204h]
11    char v9; // [esp+205h] [ebp-203h]
12    char v10; // [esp+304h] [ebp-104h]
13    char v11; // [esp+305h] [ebp-103h]
14
15    printf("Come one! Crack Me---\n");
16    v10 = 0;
17    memset(&v11, 0, 0xFFu);
18    v8 = 0;
19    memset(&v9, 0, 0xFFu);
20    while ( 1 )
21    {
22        do
23        {
24            do
25            {
26                printf("user(6-16 letters or numbers:");
27                scanf("%s", &v10);
28                v0 = (FILE *)sub_4024BE();
29                fflush(v0);
30            }
31            while ( ! (unsigned __int0)sub_401000(&v10) );
32            printf("password(6-16 letters or numbers:");
33            scanf("%s", &v8);
34            v1 = (FILE *)sub_4024BE();
35            fflush(v1);
36        }
37    }
38}
```

把函数名字改成看得懂的单词

```

1 while ( 1 )
2 {
3     do
4     {
5         do
6         {
7             printf("user(6-16 letters or numbers):");
8             scanf("%s", &name);
9             v0 = (FILE *)sub_4024BE();
10            fflush(v0);
11        }
12        while ( !sub_401000(&name) );
13        printf("password(6-16 letters or numbers):");
14        scanf("%s", &password);
15        v1 = (FILE *)sub_4024BE();
16        fflush(v1);
17    }
18    while ( !sub_401000(&password) );
19    sub_401090(&name);
20    v6 = 0;
21    memset(&v7, 0, 0xFFu);
22    v4 = 0;
23    memset(&v5, 0, 0xFFu);
24    v3 = ((int (__cdecl *)(char *, char *))loc_4011A0)(&v6, &v4);
25    if ( (unsigned __int8)sub_401830(&name, &password) )
26    {
27        if ( v3 )
28            break;
29    }
30    printf(&v4);
31    printf(&v6);
32    return 0;
33 }

```

https://blog.csdn.net/weixin_51373492

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

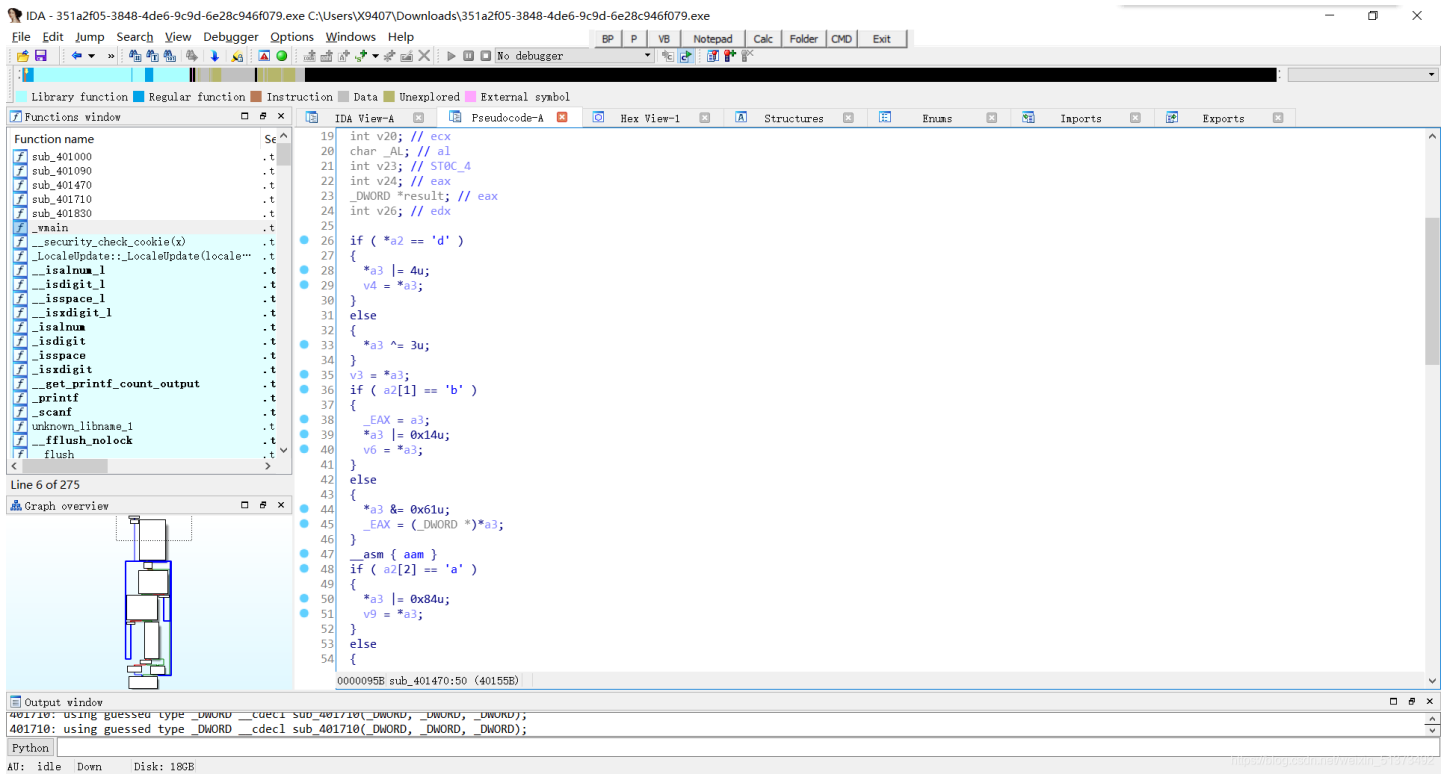
首先我们分析第一个if，点进sub_401830并开始分析

```

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

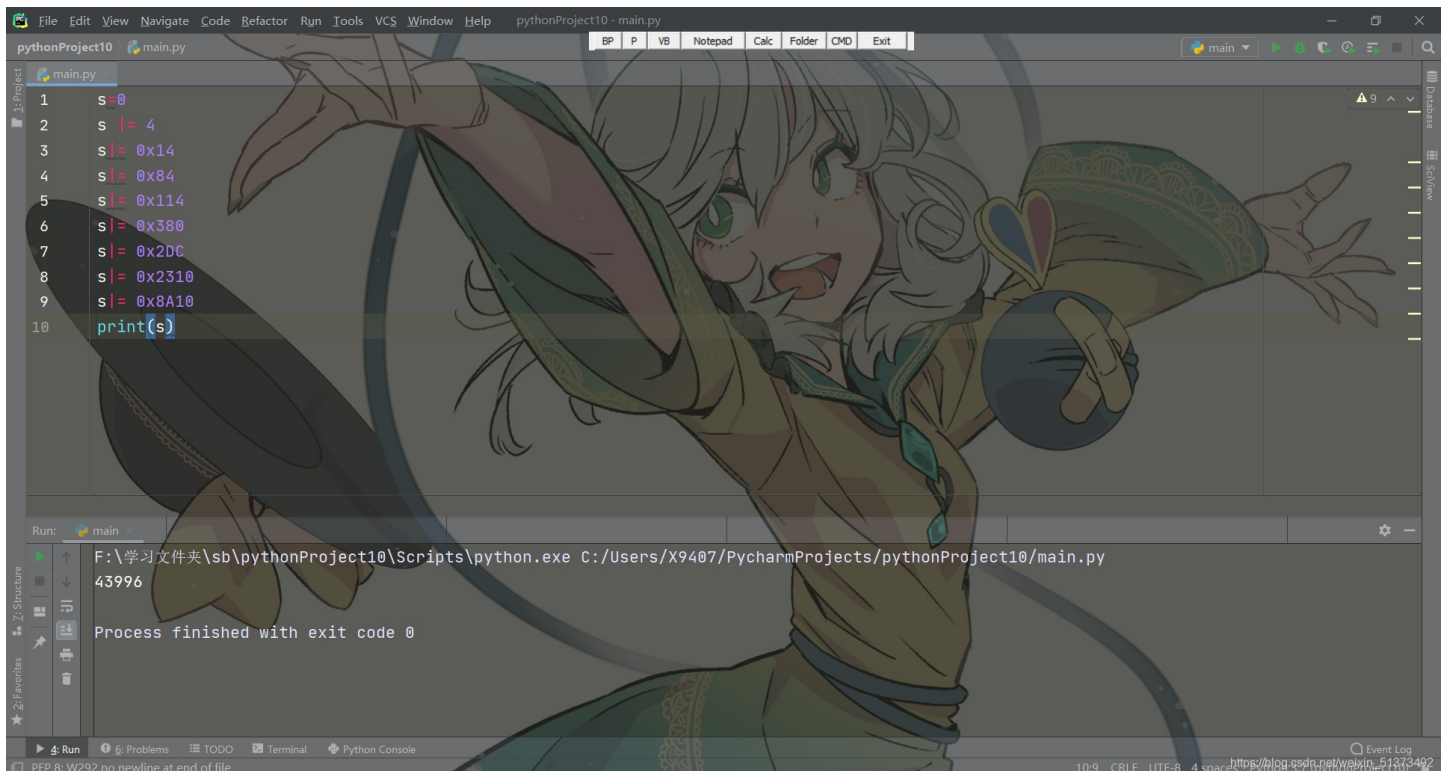
```

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，全分析。。想想都恐怖）

可以写个python脚本验证一下我们的猜测



噢，答案不是43924，这是为什么呢（棒读）

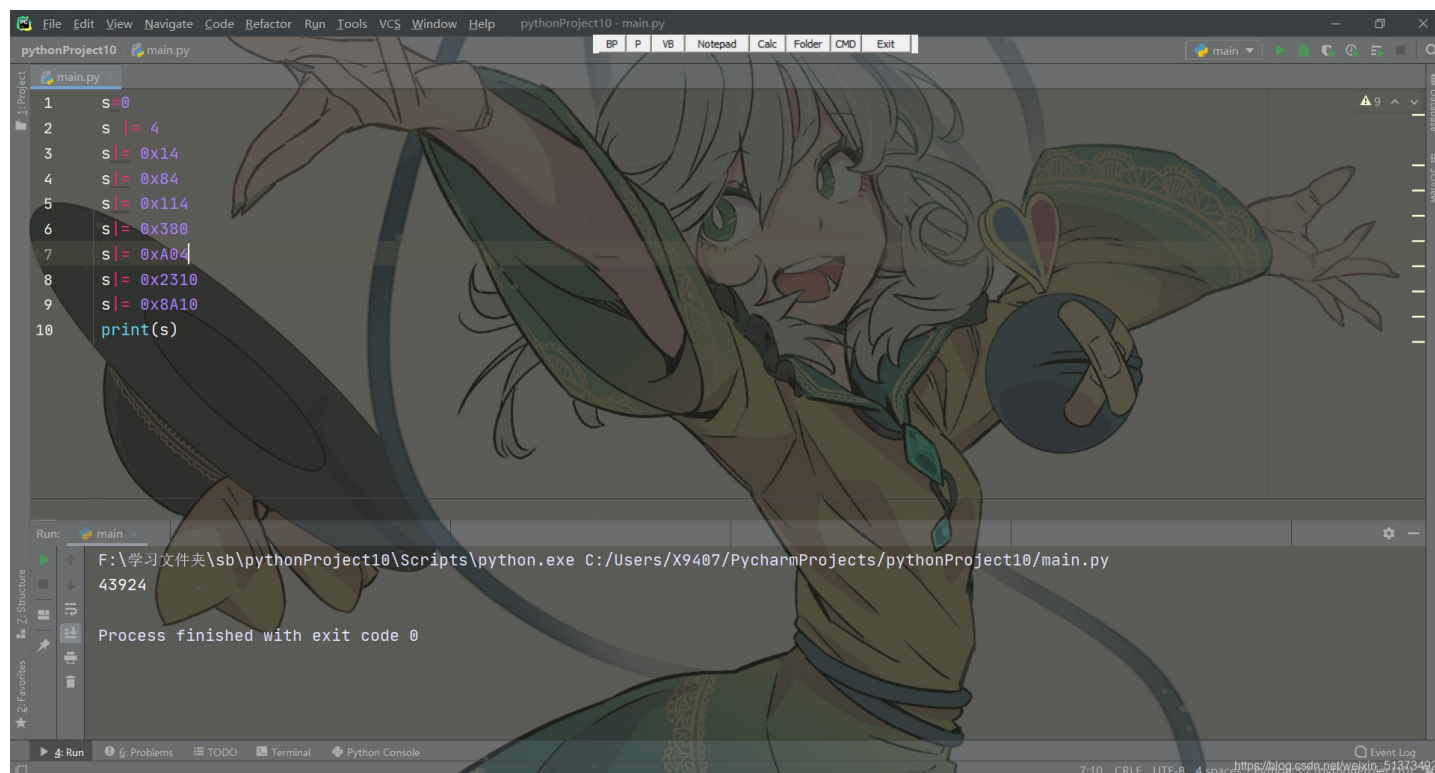
原来是

```
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
```

https://blog.csdn.net/weixin_51373492

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

```

55 while ( (signed int)v6 < 8 )
56 {
57     v11 += byte_416050[++v12];
58     v13 = byte_416050[v12];
59     v8 = byte_416050[v11];
60     byte_416050[v11] = v13;
61     byte_416050[v12] = v8;
62     if ( *( _DWORD *) ( __readfsdword( 0x30u ) + 104 ) & 0x70 )
63         v13 = v11 + v12;
64     *( &v17 + v6 ) = byte_416050[ ( unsigned __int8 ) ( v8 + v13 ) ] ^ *( &v15 + v5 );
65     if ( ( _DWORD *) ( __readfsdword( 0x30u ) + 2 ) & 0xFF )
66     {
67         v11 = -83;
68         v12 = 43;
69     }
70     sub_401710( &v17, name, v6++ );
71     v5 = v6;
72     if ( v6 >= &v15 + strlen( &v15 ) + 1 - &v16 )
73         v5 = 0;
74 }

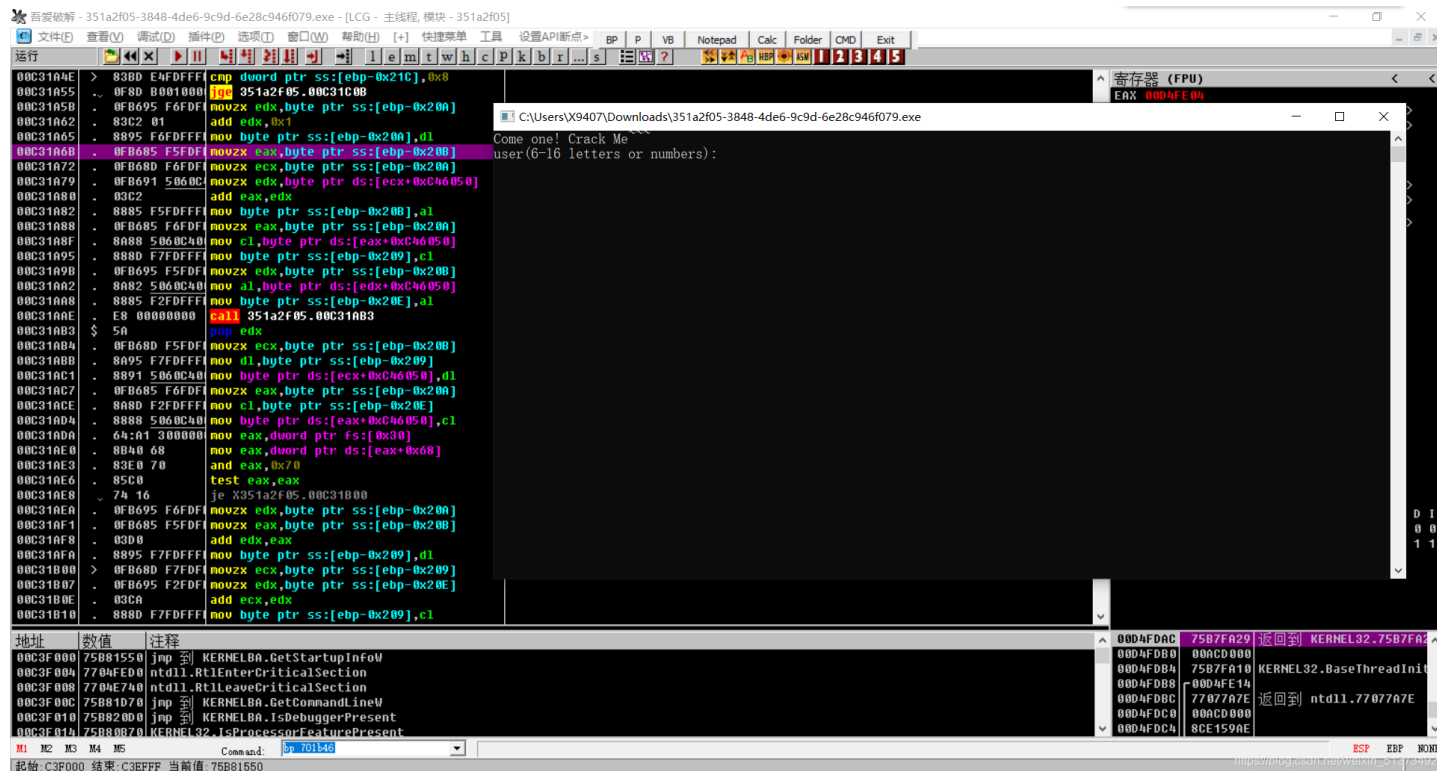
```

https://blog.csdn.net/weixin_51373492

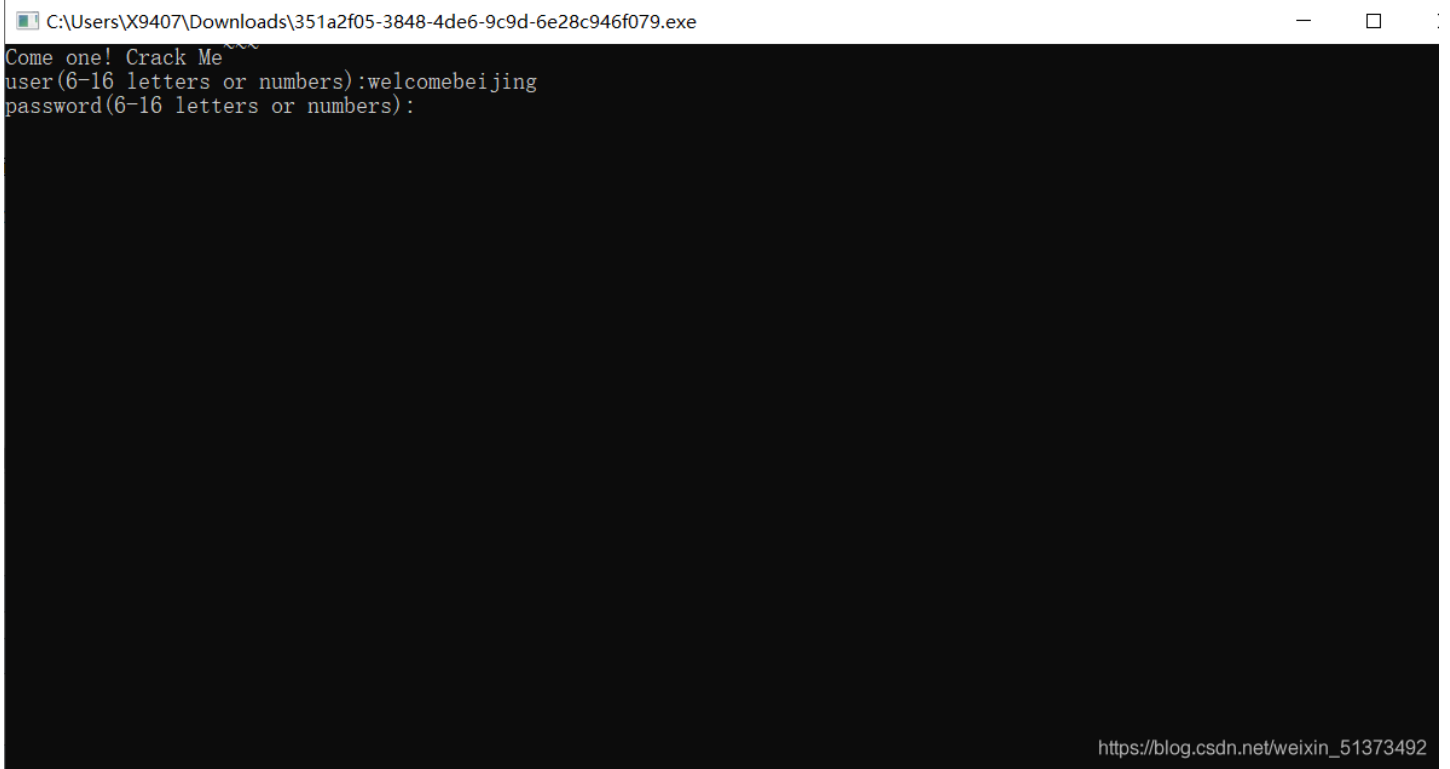
这俩我第一次做的时候分析好久，不知道这是啥玩意儿，上网查了查后发现是反调试的函数，直接不管。

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

```

3 |     v15 = 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);

```

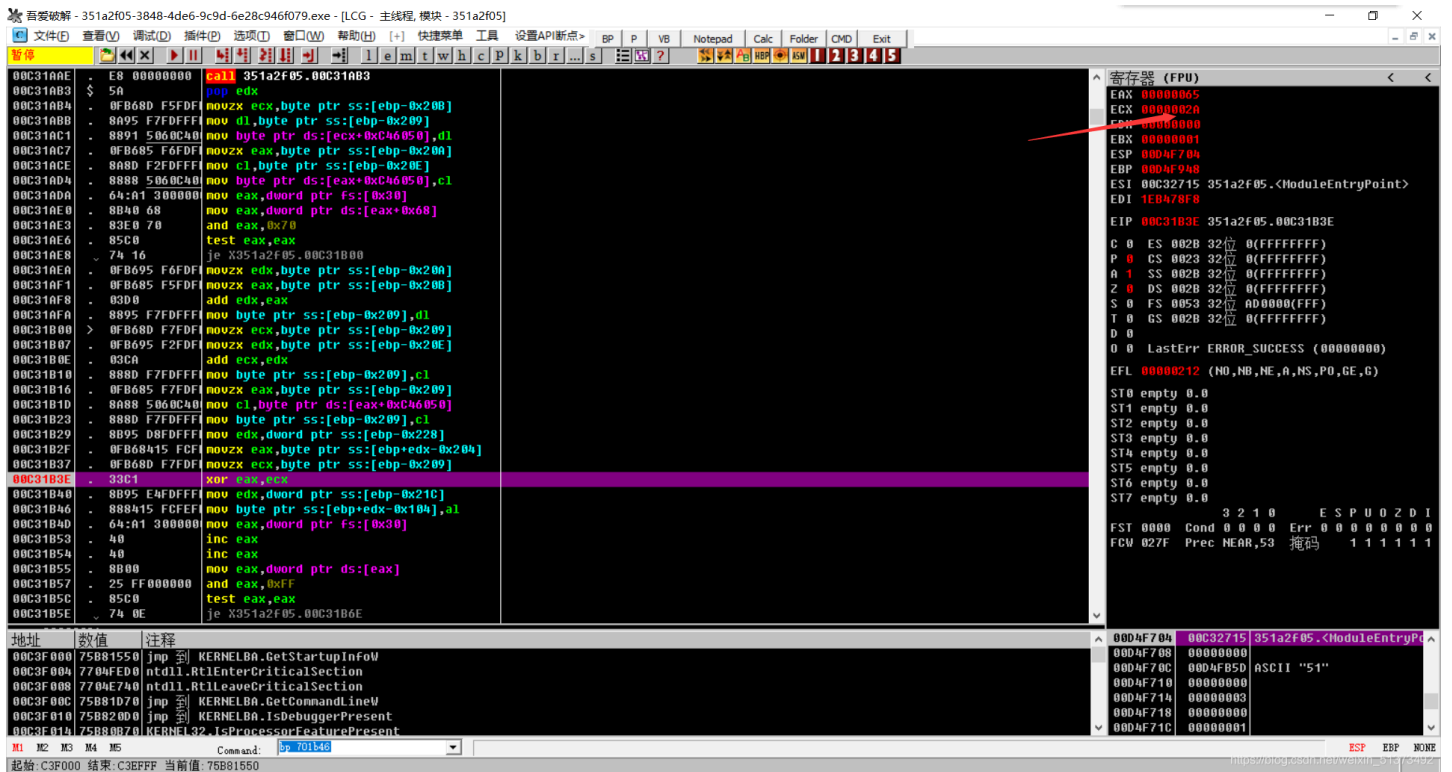
此处的异或，而通过上方的汇编代码我们可以知道上一行的 `movzx ecx, [ebp+var_209]`是将 `byte_416050`的值存入 `ecx`，此处的 `ecx`的值即为我们想要的值！

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。

00C31B2F	. 0FB68415 FCF	<code>movzx eax, byte ptr ss:[ebp+edx-0x204]</code>	
00C31B37	. 0FB68D F7DF	<code>movzx ecx, byte ptr ss:[ebp-0x209]</code>	
00C31B3E	. 33C1	<code>xor eax, ecx</code>	
00C31B40	. 8B95 E4FDFF	<code>mov edx, dword ptr ss:[ebp-0x21C]</code>	
00C31B46	. 888415 FCFF	<code>mov byte ptr ss:[ebp+edx-0x104], al</code>	

然后回到程序，随便输入个密码（反正等下会断在那儿

```
C:\Users\X9407\Downloads\351a2f05-3848-4de6-9c9d-6e28c946f079.exe
Come one! Crack Me
user(6-16 letters or numbers):welcomebeijing
password(6-16 letters or numbers):651
https://blog.csdn.net/weixin_51373492
```



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

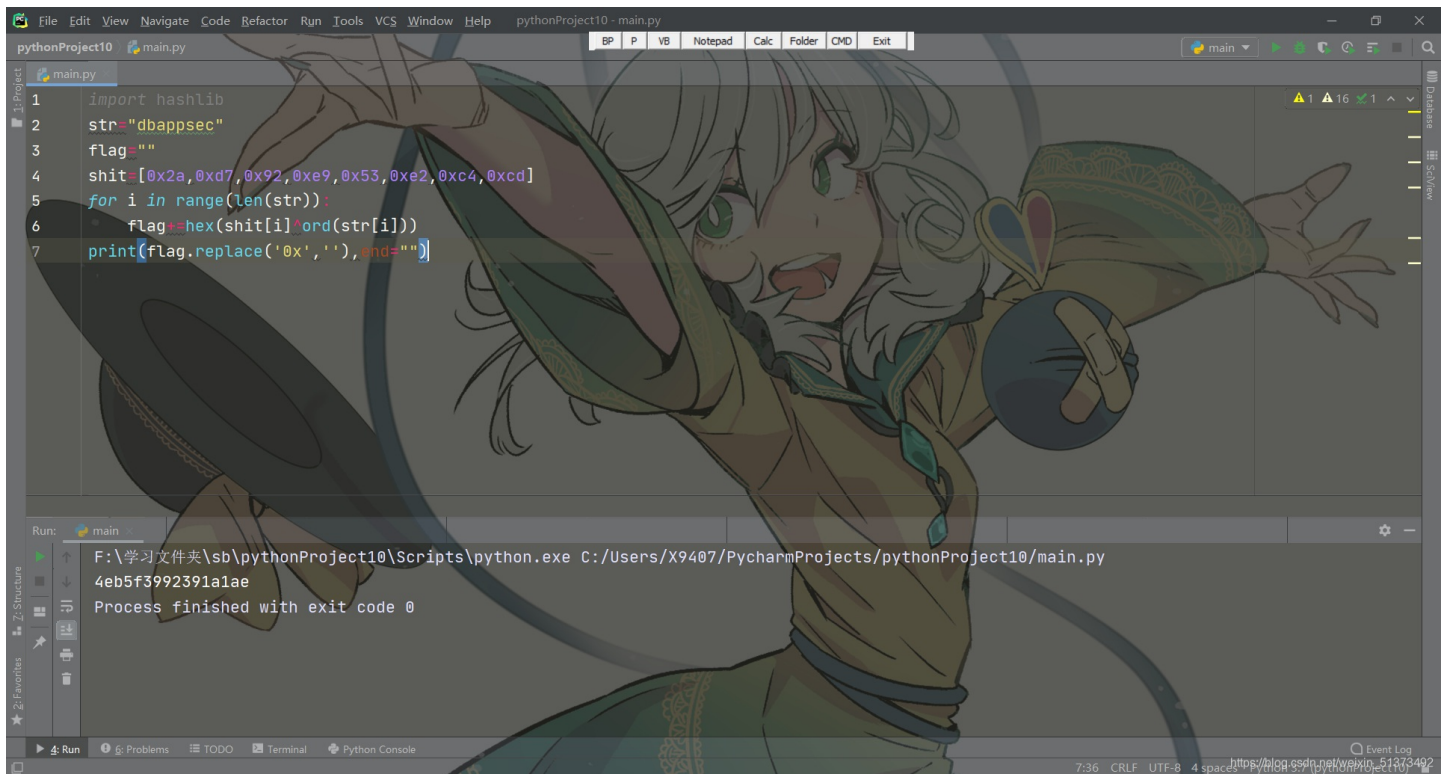
```

9  v4 = 0,
0  while ( v7 < strlen(password) )
1  {
2  if ( isdigit(password[v7]) )
3  {
4  v9 = password[v7] - 48;      字符改10进制
5  }
6  else if ( isxdigit(password[v7]) )
7  {
8  if ( *((_DWORD *) )(*(_DWORD *) )(__readfsdword(0x30u) + 24) + 12) != 2 )
9  password[v7] = 34;
10 v9 = (password[v7] | 0x20) - 87;  16进制字符改16进制
11 }
12 else
13 {
14 v9 = ((password[v7] | 0x20) - 97) % 6 + 10;  不是abcdef的字符按照
15 }                                     6个的循环转为16进制字
16 v10 = v9 + 16 * v10;
17 if ( !((signed int)(v7 + 1) % 2) )
18 {
19 *(&v15 + v4++) = v10;
20 a1 = v4;
21 v10 = 0;
22 }
23 ++v7;
24 }

```


(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



```
1 import hashlib
2 str="dbappsec"
3 flag=""
4 shit=[0x2a,0xd7,0x92,0xe9,0x53,0xe2,0xc4,0xcd]
5 for i in range(len(str)):
6     flag+=hex(shit[i]^ord(str[i]))
7 print(flag.replace('0x',''),end="")
```

Run: main
F:\学习文件夹\sb\pythonProject10\Scripts\python.exe C:/Users/X9407/PycharmProjects/pythonProject10/main.py
4eb5f3992391a1ae
Process finished with exit code 0

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

首页 > MD5在线加密

MD5在线加密

要加密的字符串:

字符串	4eb5f3992391a1ae
16位 小写	b84f2a9056699958
16位 大写	B84F2A9056699958
32位 小写	d2be2981b84f2a905669995873d6a36c
32位 大写	D2BE2981B84F2A905669995873D6A36C

https://blog.csdn.net/weixin_51373492

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