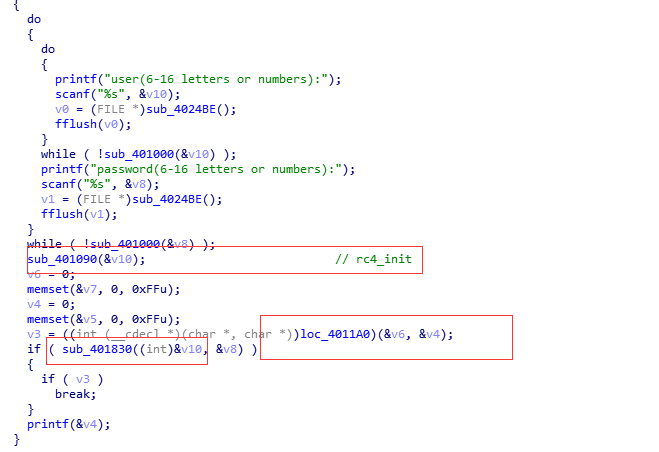
# 题目分析

题目让找到输入用户名为welcomebeijing对应的密码。

我们进行分析。



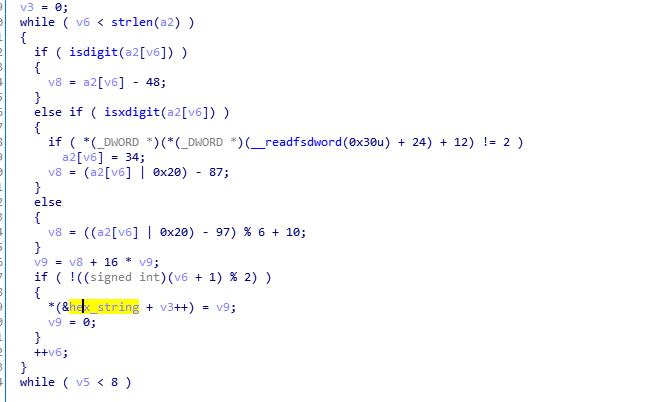
Loc\_4011A0不能反编译，我们nop掉一些字节修复后进行反编译可以看出是对成功失败字符串进行赋值。



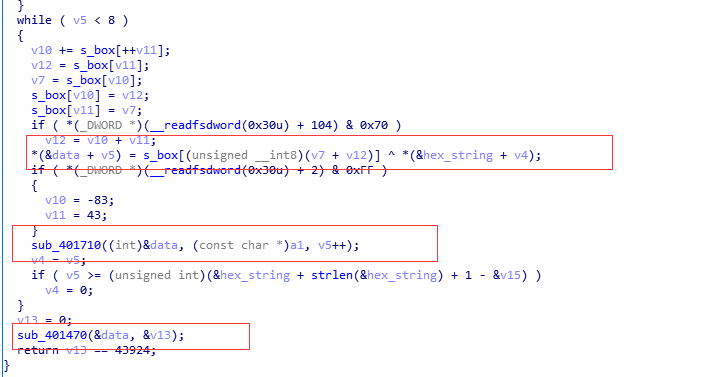
sub\_401090(&v10)里面的操作像是在进行rc4加密的s盒初始化。

然后我们进入sub\_401830函数进行分析。

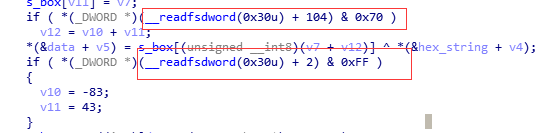
通过调试发现这一堆操作是输入的密码（要求是十六进制字符）从hex\_string转换位内存中的bytes。

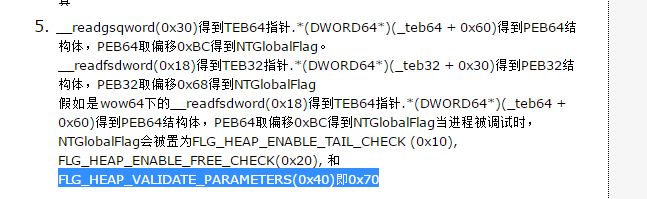


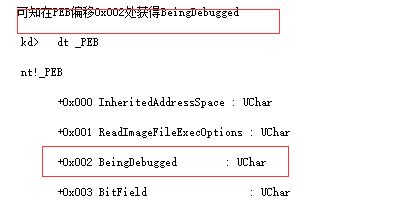
然后下面是魔改的rc4后半部分，进行了反调试操作，关键的地方是在异或那里，八个s\_box值和hex\_string转化后的八个byte值进行异或。



需要注意的是两处的反调试。







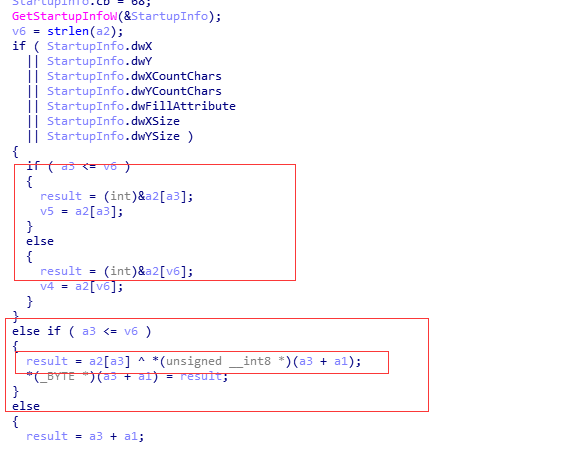
因这两处反调试，所以我们进行调试的话得到的八个s\_box值是和正常运行不一样的。

我们把00401AE3248 和 00401B5E248处的jz改为jnz，然后进行调试即可得到和正常运行一样的八个s\_box值，可以得到是0x2A, 0xD7, 0x92, 0xE9, 0x53, 0xE2, 0xC4, 0xCD。

然后看sub\_401710，sub\_401710函数是个骚操作，好像出题人原意是想用GetStartupInfoW来进行反调试，程序对打开它自身的方式进行检查。

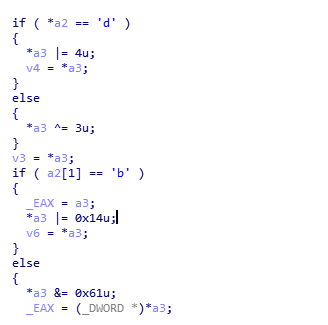
经过测试，双击运行和使用几个调试器运行执行的都是第二部分，将前面异或得到的data值和"welcomebeijing"的前八个字节进行异或。

但是在cmd中打开时候就执行第一部分，相当于没操作。



最后的校验是sub\_401470，跟进去后发现是对八个字符进行选择运算，运算后的结果为43924 。我们可以得到最后的字符串为"dbappsec" 。





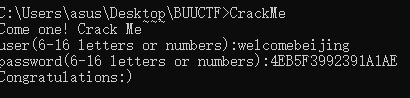
然后我们根据sub\_401710函数执行的不同部分，可以写两个脚本得到两个结果。

# 脚本

Cmd模式打开

arr = [0x2A, 0xD7, 0x92, 0xE9, 0x53, 0xE2, 0xC4, 0xCD]  
str = "dbappsec"  
password = ""  
for i in range(len(arr)):  
 arr[i] = ord(str[i]) ^ arr[i]  
 password += "%X"%arr[i]  
print password

得到4EB5F3992391A1AE（小写也可）



其他模式打开

arr = [0x2A, 0xD7, 0x92, 0xE9, 0x53, 0xE2, 0xC4, 0xCD]  
str = "dbappsec"  
str1 = "welcomebeijing"  
password = ""  
for i in range(len(arr)):  
 arr[i] = ord(str[i]) ^ ord(str1[i])^ arr[i]  
 password += "%X"%arr[i]  
print password

得到39D09FFA4CFCC4CC（小写也可）