buu-re-CrackCTF & easyre

原创

「已注销」 于 2021-03-17 12:59:41 发布 33 收藏

分类专栏： buuoj CTF Reverse

本文链接：https://blog.csdn.net/OERR0R_/article/details/114925234


buuoj 同时被 3 个专栏收录

8 篇文章 0 订阅

订阅专栏


CTF

18 篇文章 0 订阅

订阅专栏


Reverse

14 篇文章 0 订阅

订阅专栏

CrackCTF

总共有两个加密过程，第一个过程是调用了Windows的sha1加密，第二个过程是调用了类MD5加密，大致就是把AAA文件复制到了lPBuffer中，然后调用sub_401005函数，对lpString 和 lpBuffer进行异或操作，输入的是6个字符，因此异或的也应该是6个字节，在这里记录一下异或的具体py代码

```
import hashlib

passwd1="6e32d0943418c2c33385bc35a1470250dd8923a9"
passwd2="27019e688a4e62a649fd99cadaafdb4e"
suffix="@DBApp"
aaa=[0x5, 0x7d, 0x41, 0x15, 0x26, 0x1]
rtf_h=[0x7b, 0x5c, 0x72, 0x74, 0x66, 0x31]

pass1=""
pass2=""
for i in range(100000, 1000000):
    instr=str(i)+suffix
    res=hashlib.sha1(instr.encode('utf-8')).hexdigest()
    if res==passwd1:
        pass1=str(i)
        print("passwd1 = ", pass1)
        break

for i in range(6):
    res=aaa[i]^rtf_h[i]
    pass2+=chr(res)
print("passwd2 = ", pass2)
```

[ACTF新生赛2020]easyre

观察for循环就行，从for循环了解到flag长度应该是11，将flag的ASCII值作为下标取值，与v4数组比较。很简单，只需要利用v4数组在_data_start_中找位置，就是我们flag的值

```
v4=[42, 70, 39, 34, 78, 44, 34, 40, 73, 63, 43, 64]

str=r"}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)(" + chr(0x27) + r'&%
$# !"'

pos=[]

for i in v4:
    pos.append(str.find(chr(i))+1)
s=[chr(x+1) for x in pos]
flag=''.join(s)

print('flag{'+flag+'}')
```