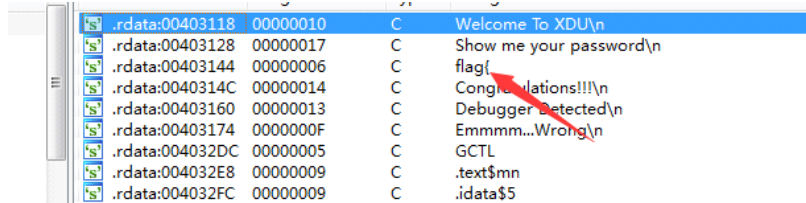


## Re\_2

这题只要知道是s盒还是挺容易的。

先打开ida，拖进去，f5,f5,f5!!!

在搜索字符的时候发现了flag的关键字。



然后跟进去，f5,f5,f5(0\_0)

```

char SubStr[]
sub_401020
    db 'flag{',0 ; DATA XREF: sub_401020+70fo
    align 4
    db 'Congratulations!!!',0Ah,0 ; DATA XREF: sub_401020+169fo
    db 'Debugger Detected',0Ah,0 ; DATA XREF: Callback+Cfo
    align 4
    db 'Emmm...Wrong',0Ah,0 ; DATA XREF: sub_401340+3fo
    align 1Ah

```

通过交叉引用去到函数内部。发现是主函数。

```

1 int sub_401020()
2 {
3     HANDLE TimerQueue; // [sp+4h] [bp-B8h]@1
4     int v2; // [sp+8h] [bp-B4h]@4
5     HANDLE phNewTimer; // [sp+18h] [bp-A4h]@1
6     char v4; // [sp+1Ch] [bp-A0h]@4
7     char Dst; // [sp+50h] [bp-6Ch]@4
8     char Str; // [sp+84h] [bp-38h]@1
9     char Src; // [sp+89h] [bp-33h]@4
10    char v8; // [sp+A9h] [bp-13h]@2
11
12    TimerQueue = CreateTimerQueue();
13    CreateTimerQueueTimer(&phNewTimer, TimerQueue, Callback, 0, 0, 0x3E8u, 0x20u);
14    sub_401450("Welcome To XDU\n");
15    sub_401450("Show me your password\n");
16    sub_4013C0("%s", &Str, 50);
17    if ( (char *)sub_401000(&Str, 'fFlag{') != &Str || v8 != 125 )
18    {
19        sub_401340();
20        sub_401360(&Dst, &Src, 0x20u);
21        sub_4011F0(&Dst);
22        sub_401270(&unk_4040C0, &Dst, &v4);
23        v2 = strcmp(&v4, a23gjf13au98hk3);
24        if ( v2 )
25            v2 = -(v2 < 0) | 1;
26        if ( v2 )
27            sub_401340();
28        sub_401450("Congratulations!!!\n");
29        DeleteTimerQueueEx(TimerQueue, (HANDLE)0xFFFFFFFF);
30        return 0;
31    }
32 }

```

程序大概就是这样，先打印一堆字符，然后获取我们的输入，通过str和v8的地址我们可以知道flag的长度为38。

```
print int("0x38",16) - int("0x13",16) + 1
```

然后发现又调用了sub\_4011F0函数

```

1 char __cdecl sub_4011F0(char *a1)
2 {
3     char result; // a1@2
4     char v2; // $T17_1@2
5     char *v3; // [sp+Ch] [bp-Ch]@1
6     signed int i; // [sp+10h] [bp-8h]@5
7
8     v3 = a1;
9     do
10    {
11        result = *v3;
12        v2 = *v3++;
13    }
14    while ( v2 );
15    if ( v3 - (a1 + 1) != 32 )
16        sub_401340();
17    for ( i = 0; i < 32; ++i )
18    {
19        a1[i] ^= dword_404040[4 * i];
20        result = i + 1;
21    }
22    return result;
23 }

```

一个亦或操作，接着又调用了sub\_401270

```

1 int __cdecl sub_401270(int a1, const char *a2, int a3)
2 {
3     signed int v3; // kr00_4@1
4     int result; // eax@4
5     signed int v5; // [sp+10h] [bp-8h]@1
6
7     v5 = 0;
8     v3 = strlen(a2);
9     while ( v5 < v3 )
10    {
11        *(_BYTE *)(a3 + *(_DWORD *)(a1 + 4 * v5)) = a2[v5];
12        ++v5;
13    }
14    result = v5 + a3;
15    *(_BYTE *)(v5 + a3) = 0;
16    return result;
17 }

```

发现是一个s盒加密，关于s盒加密的话

#### 1、S盒加密

首先在进行s盒加密之前，我们要找一个盒子，盒子的长度要和被加密的明文长度一样。然后有了盒子有被加密的明文之后我们就可以进行加密了。对了，还需要一个长度和他们一样的接受变量。定义盒子为A，要加密的明文为b，接受值为C。

加密算法：

```

for i in range(len(A)):
    C[A[i]] = B[i]
print C

```

#### 2、S盒解密

S盒加密其实挺容易的，解密其实也挺容易，就多了一步操作，在还原字符串之前，要先逆S盒。然后进行还原字符串。

解密算法：

```

for i in range(32):
    C[A[i]] = i
for i in range(32):
    A[C[i]] = B[i]
print"".join(A)

```

本菜鸡做的笔记，然后知道这些我们就可以编写Python脚本了。

提取数据发现有点乱

```

unsigned char ida_chars[] =
{
    4,  0,  0,  0,  15,  0,  0,  0,  11,  0,
    0,  0,  30,  0,  0,  0,  14,  0,  0,  0,
    20,  0,  0,  0,  31,  0,  0,  0,  9,  0,
    0,  0,  23,  0,  0,  0,  2,  0,  0,  0,
    25,  0,  0,  0,  28,  0,  0,  0,  18,  0,
    0,  0,  16,  0,  0,  0,  0,  0,  0,  0,
    8,  0,  0,  0,  17,  0,  0,  0,  1,  0,
    0,  0,  21,  0,  0,  0,  3,  0,  0,  0,
    10,  0,  0,  0,  29,  0,  0,  0,  12,  0,
    0,  0,  22,  0,  0,  0,  24,  0,  0,  0,
    13,  0,  0,  0,  27,  0,  0,  0,  5,  0,

```

简单的提取下

```

a = [
    4, 0, 0, 0, 15, 0, 0, 0, 11, 0,
    0, 0, 30, 0, 0, 0, 14, 0, 0, 0,
    20, 0, 0, 0, 31, 0, 0, 0, 9, 0,
    0, 0, 23, 0, 0, 0, 2, 0, 0, 0,
    25, 0, 0, 0, 28, 0, 0, 0, 18, 0,
    0, 0, 16, 0, 0, 0, 0, 0, 0, 0,
    8, 0, 0, 0, 17, 0, 0, 0, 1, 0,
    0, 0, 21, 0, 0, 0, 3, 0, 0, 0,
    10, 0, 0, 0, 29, 0, 0, 0, 12, 0,
    0, 0, 22, 0, 0, 0, 24, 0, 0, 0,
    13, 0, 0, 0, 27, 0, 0, 0, 5, 0,
    0, 0, 7, 0, 0, 0, 6, 0, 0, 0,
    19, 0, 0, 0, 26, 0, 0, 0, 0]

print a[0::4]

```

```

re_1 x
C:\Users\shlllc0de\venv\Scripts\python.exe C:/Users/shlllc0de/Desktop/python/re_1.py
[4, 15, 11, 30, 14, 20, 31, 9, 23, 2, 25, 28, 18, 16, 0, 8, 17, 1, 21, 3, 10, 29, 12, 22, 24, 13, 27, 5, 7, 6, 19, 26]

Process finished with exit code 0

```

首先先逆s盒

```

#coding:utf-8
c = "23gjfl3au98hk3a1090zp8qjs41h39jp"
a = [4, 15, 11, 30, 14, 20, 31, 9, 23, 2, 25, 28, 18, 16, 0, 8, 17, 1, 21, 3, 10, 29, 12, 22, 24, 13, 27, 5, 7, 6, 19, 26]
b = [0] * len(a)
for i in range(len(a)):
    b[a[i]] = i
print b

```

```

re_1 x
C:\Users\shlllc0de\venv\Scripts\python.exe C:/Users/shlllc0de/Desktop/python/re_1.py
[14, 17, 9, 19, 0, 27, 29, 28, 15, 7, 20, 2, 22, 25, 4, 1, 13, 16, 12, 30, 5, 18, 23, 8, 24, 10, 31, 26, 11, 21, 3, 6]

```

然后就能进行解密了

```

#coding:utf-8
c = "23gjfl3au98hk3a1090zp8qjs41h39jp"
xor = [83, 69, 92, 30, 80, 19, 47, 120, 4, 83, 88, 74, 67, 1, 65, 42, 8, 64, 103, 47, 12, 74, 18, 46, 65, 108, 5, 84, 64, 18, 91, 79]
a = [4, 15, 11, 30, 14, 20, 31, 9, 23, 2, 25, 28, 18, 16, 0, 8, 17, 1, 21, 3, 10, 29, 12, 22, 24, 13, 27, 5, 7, 6, 19, 26]
b = [0] * len(a)
for i in range(len(a)):
    b[a[i]] = i
for i in range(len(a)):
    a[b[i]] = c[i]
flag = [chr(ord(a[i]) ^ xor[i]) for i in range(len(a))]
print "flag{" + "".join(flag) + "}"

```