1 题目来源

实验吧：http://www.shiyanbar.com/ctf/28

2 逆向分析

首先我们先运行下程序，根据程序提示输入一段字符，程序输出为“You Don't Guess it~”。

```
Can you Guess the Code: ddddddddddd
You Don't Guess it~
```

用 OD 打开程序，鼠标右键选择查找->所有参考文本字串，找到“Can you Guess the Code:”，单击得到结果如下：

```
. 68 70704000   push    00407070                 ASCII "Can you Guess the Code: "
. E8 AA010000   call    004011BA
. 8D4424 04     lea     eax, dword ptr [es
. 50            push    eax
. E8 56010000   call    00401170
. 8D4C24 08     lea     ecx, dword ptr [es
. 51            push    ecx
. E8 3C000000   call    00401060
. 83C4 0C       add     esp, 0C
. 85C0          test    eax, eax
. 74 16         je      short 00401041
. 68 48704000   push    00407048                 ASCII "Good! The Key is your inp
. E8 85010000   call    004011BA
. 83C4 04       add     esp, 4
. 33C0          xor     eax, eax
. 81C4 D007000( add     esp, 7D0
. C3            retn
> 68 30704000   push    00407030                 ASCII "You Don't Guess it~",LF
```

使用快捷键 F4，运行到光标所在处，我们从这里开始分析。程序首先将该字符串压栈，然后调用 0x004011ba，这个函数基本就是打印字符串，所以我们直接 F8 单步运行，接下来将一个地址压栈并调用了 0x00401170，这也很明显，是读取我们的输入到一个地址。接下来遇到的 0x401060 则是我们主要研究对象，使用 F7 单步进入。

```
00401060  $  83EC 10      sub     esp, 10
00401063  .  A1 8C704000  mov     eax, dword ptr [40708C]
00401068  .  8B0D 9070400 mov     ecx, dword ptr [407090]
0040106E  .  53           push    ebx
0040106F  .  894424 04    mov     dword ptr [esp+4], eax
00401073  .  66:A1 987040 mov     ax, word ptr [407098]
00401079  .  894C24 08    mov     dword ptr [esp+8], ecx
0040107D  .  8A0D 9A70400 mov     cl, byte ptr [40709A]
00401083  .  56           push    esi
00401084  .  8B7424 1C    mov     esi, dword ptr [esp+1C]
00401088  .  57           push    edi
00401089  .  66:894424 18 mov     word ptr [esp+18], ax
0040108E  .  884C24 1A    mov     byte ptr [esp+1A], cl
00401092  .  8BFE         mov     edi, esi
00401094  .  83C9 FF      or      ecx, FFFFFFFF
00401097  .  33C0         xor     eax, eax
00401099  .  8B15 9470400 mov     edx, dword ptr [407094]
0040109F  .  F2:AE        repne   scas byte ptr es:[edi]
004010A1  .  F7D1         not     ecx
004010A3  .  49           dec     ecx
004010A4  .  895424 14    mov     dword ptr [esp+14], edx
004010A8  .  8BD1         mov     edx, ecx
004010AA  .  8D7C24 0C    lea     edi, dword ptr [esp+C]
004010AE  .  83C9 FF      or      ecx, FFFFFFFF
004010B1  .  F2:AE        repne   scas byte ptr es:[edi]
```

我们看到程序首先保存一个地址到 eax，我们在数据段找到该地址，得到结果如下：

```
0040708C  68 57 19 48  50 6E 58 78  54 6A 19 58  5E 06 00 00
```

接下来程序所做的是将该地址的内容保存到栈中，并通过“push esi”,“move edi,esi”两句将用户输入保存到 edi 中，接下来分别计算两个字符串的长度并比较，由此我们可以得知，输入字符串的长度为 14。

```
004010CC  >  8A0432          mov     al, byte ptr [edx+esi]
004010CF  .  8BFE            mov     edi, esi
004010D1  .  34 20           xor     al, 20
004010D3  .  83C9 FF         or      ecx, FFFFFFFF
004010D6  .  880432          mov     byte ptr [edx+esi], al
004010D9  .  33C0            xor     eax, eax
004010DB  .  42              inc     edx
004010DC  .  F2:AE           repne   scas byte ptr es:[edi]
004010DE  .  F7D1            not     ecx
004010E0  .  49              dec     ecx
004010E1  .  3BD1            cmp     edx, ecx
004010E3  .^ 72 E7           jb      short 004010CC
```

这一段程序的作用是将我们的输入依次与 0x20 异或，并将结果存在原地址中，接下来就是对内存中的字符串操作，依次减 5。

```
004010F7  >  8A4414 0C       mov     al, byte ptr [esp+edx+C]
004010FB  .  8D7C24 0C       lea     edi, dword ptr [esp+C]
004010FF  .  04 FB           add     al, 0FB
00401101  .  83C9 FF         or      ecx, FFFFFFFF
00401104  .  884414 0C       mov     byte ptr [esp+edx+C], al
00401108  .  33C0            xor     eax, eax
0040110A  .  42              inc     edx
0040110B  .  F2:AE           repne   scas byte ptr es:[edi]
0040110D  .  F7D1            not     ecx
0040110F  .  49              dec     ecx
00401110  .  3BD1            cmp     edx, ecx
00401112  .^ 72 E3           jb      short 004010F7
```

经过这两个操作后，进行对比，相等则注册成功，不等则失败。

```
0040112C  >  8A5C14 0C       mov     bl, byte ptr [esp+edx+C]
00401130  .  8D4414 0C       lea     eax, dword ptr [esp+edx+C]
00401134  .  8A0C06          mov     cl, byte ptr [esi+eax]
00401137  .  3ACB            cmp     cl, bl
00401139  .v 75 1F           jnz     short 0040115A
0040113B  .  8D7C24 0C       lea     edi, dword ptr [esp+C]
0040113F  .  83C9 FF         or      ecx, FFFFFFFF
00401142  .  33C0            xor     eax, eax
00401144  .  42              inc     edx
00401145  .  F2:AE           repne   scas byte ptr es:[edi]
00401147  .  F7D1            not     ecx
00401149  .  49              dec     ecx
0040114A  .  3BD1            cmp     edx, ecx
0040114C  .^ 72 DE           jb      short 0040112C
```

此时，我们在栈中看见内存中的字符串内容为：

```
0019F754  43145263
0019F758  7353694B
0019F75C  5314654F
0019F760  00000159
```

就可以编写脚本计算正确答案了。

3 脚本

Python 脚本如下：

```
s = [0x63, 0x52, 0x14, 0x43, 0x4b, 0x69, 0x53, 0x73, 0x4f, 0x65,
0x14, 0x53, 0x59, 0x1]
flag = ""
for i in s:
    flag += chr(i ^ 0x20)
print(flag)
```

得到结果为：Cr4ckIsSoE4sy!