

HGAME Week3 WriteUp

RE

oooollvm

1. 进去先看了字符串 发现 flag 又是所输入的内容

```
61 ry.again..Congra      29      it s your turn now );
73 tulations!.Looks     30      __isoc99_scanf("%34s", s);
6C .like.you.are.al     31      v18 = strlen(s);
77 ready.familiar.w    32      v14 = -834745915;
79 ith.it.Flag.is.y     33      while ( 1 )
3B our.input.....;    34      {
                        35      while ( 1 )
```

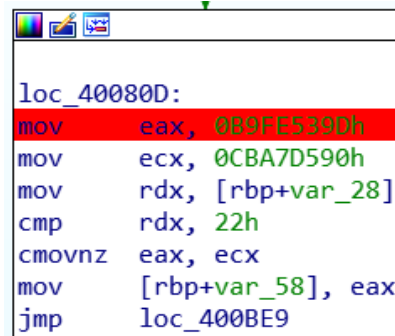
2. 看到 scanf 和 strlen 先猜测 s 的长度为 34，照着学习资料先对

各个分支进行断点，开始动态调试

第一个分块

不难看出是对长度的比较

验证了上面的猜测

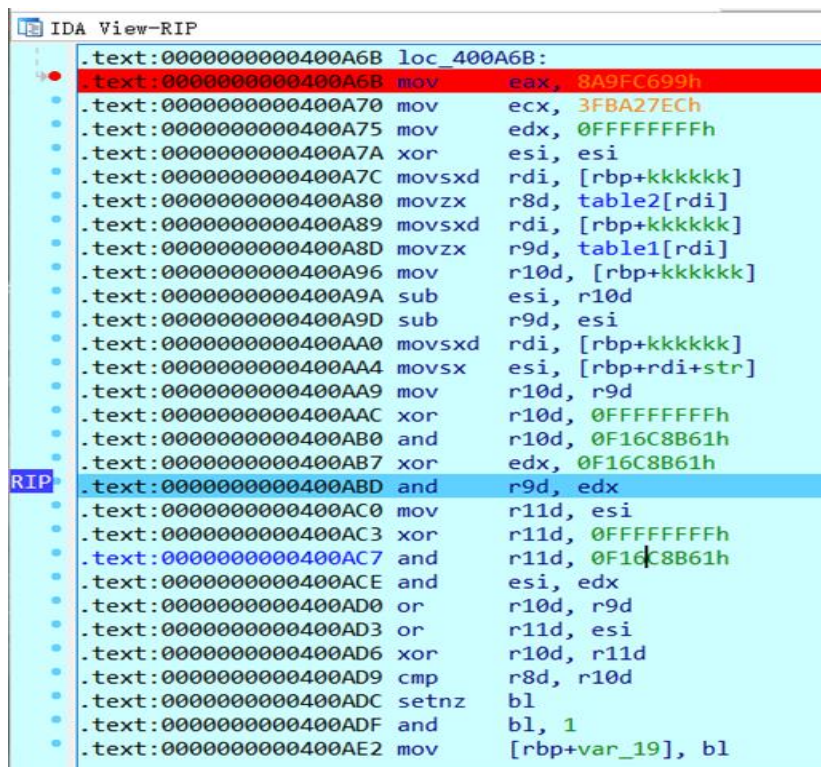


```
loc_40080D:
mov     eax, 0B9FE539Dh
mov     ecx, 0CBA7D590h
mov     rdx, [rbp+var_28]
cmp     rdx, 22h
cmovnz  eax, ecx
mov     [rbp+var_58], eax
jmp     loc_400BE9
```

3. 跳过了一些无关紧要的分块，找到一个比较可疑的代码块

经过这个分块后程序 就会提示 wrong 尝试将 r10d 的值修改与

r8d 相同 果然调试可以继续进行



```
IDA View-RIP
.text:0000000000400A6B loc_400A6B:
.text:0000000000400A6B mov     eax, 8A9FC699h
.text:0000000000400A70 mov     ecx, 3FBA27ECh
.text:0000000000400A75 mov     edx, 0FFFFFFFh
.text:0000000000400A7A xor     esi, esi
.text:0000000000400A7C movsxd  rdi, [rbp+kkkkkk]
.text:0000000000400A80 movzx   r8d, table2[rdi]
.text:0000000000400A89 movsxd  rdi, [rbp+kkkkkk]
.text:0000000000400A8D movzx   r9d, table1[rdi]
.text:0000000000400A96 mov     r10d, [rbp+kkkkkk]
.text:0000000000400A9A sub     esi, r10d
.text:0000000000400A9D sub     r9d, esi
.text:0000000000400AA0 movsxd  rdi, [rbp+kkkkkk]
.text:0000000000400AA4 movsx   esi, [rbp+rdi+str]
.text:0000000000400AA9 mov     r10d, r9d
.text:0000000000400AAC xor     r10d, 0FFFFFFFh
.text:0000000000400AB0 and     r10d, 0F16C8B61h
.text:0000000000400AB7 xor     edx, 0F16C8B61h
RIP .text:0000000000400ABD and     r9d, edx
.text:0000000000400AC0 mov     r11d, esi
.text:0000000000400AC3 xor     r11d, 0FFFFFFFh
.text:0000000000400AC7 and     r11d, 0F16C8B61h
.text:0000000000400ACE and     esi, edx
.text:0000000000400AD0 or      r10d, r9d
.text:0000000000400AD3 or      r11d, esi
.text:0000000000400AD6 xor     r10d, r11d
.text:0000000000400AD9 cmp     r8d, r10d
.text:0000000000400ADC setnz  bl, 1
.text:0000000000400ADF and     bl, 1
.text:0000000000400AE2 mov     [rbp+var_19], bl
```

4. 直接根据程序写脚本 得到 flag

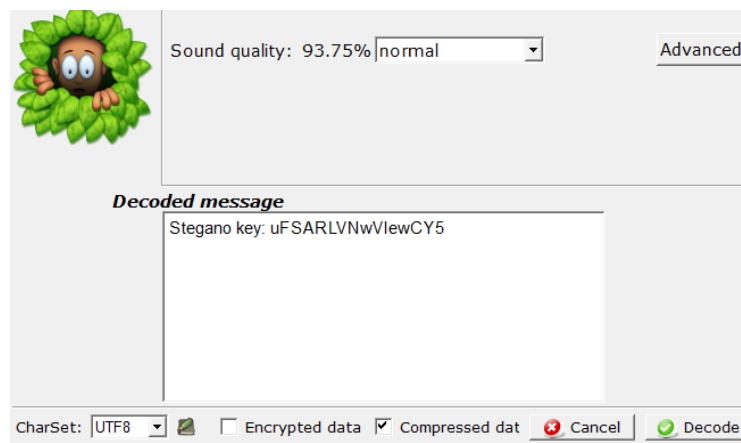
```
int t1[34] = {0x59,0x67,0x41,0x0B,0x67,0xC8,0x95,0x70,0,0x53,0x54,0x2D,0x27,0xB  
int t2[34] = {0X31,0X0F,0X22,0X63,0X0E,0XB6,0XD4,0X1B,0X64,0X0A,0X33,0X66,0X4F,  
for(int i=0;i<34;i++){  
    unsigned long long r8 = t2[i];  
    unsigned long long r9 = t1[i] + i;  
    unsigned long long r10 = (r9 ^ 0xFFFFFFFF) & 0xF16C8B61;  
    unsigned long long edx = 0xFFFFFFFF ^ 0xF16C8B61;  
    r9 = r9 & edx;  
    for(int esi = 0;esi<256;esi++){  
        {  
            unsigned long long r11 = esi;  
            r11 = (r11 ^ 0xFFFFFFFF) & 0xF16C8B61;  
            unsigned long long eesi = esi & edx;  
            unsigned long long rr10 = r10 | r9;  
            r11 = r11 | eesi;  
            rr10 = rr10 ^ r11;  
            if(r8 == rr10)  
            {  
                printf("%c",esi);  
                break;  
            }  
        }  
        if(es i == 255)  
        {  
            printf("N ");  
        }  
    }  
}
```

选择C:\Windows\system32\cmd.exe
hgame{011Vm~|s-complex~but-51MPLE
请按任意键继续. . .

MISC

三重隐写

1. 打开解压包得到三份音频和一个安装包，一个 LSB.wav 给的提示很足，放到 silenteye 里直接得到第一层密码



2. 用了 Audacity 没有什么发现，根据剩下的两个 mp3 文件和得到的 key，用 MP3Stego 并加上得到的 key

根据 上裹与手抄卷.mp3 得到一个 txt 文件



3. 根据密码解开 flag 压缩包，发现需要解密密码

猜测线索在 Unlasting.mp3 里，用自带的播放器打开，发现专辑封面是条形码，查了下是 pdf417 的条形码 手机下了个软件

扫出 AES key: 1ZmmeaLL^Typbcg3

4. 把 key 输入 即可解出含有 flag 的 txt 文件