【UIUCTF】Redd‘s Art WriteUp

原创

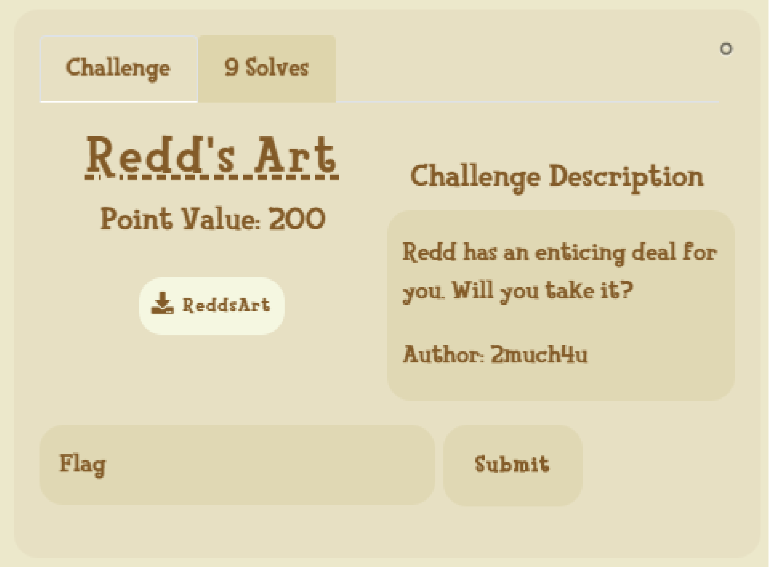
古月浪子 于 2020-07-20 10:57:26 发布 127 收藏

文章标签： CTF

本文链接：https://blog.csdn.net/tqydyqt/article/details/107458916

版权

题外话：这个CTF的前端UI非常不错！



Redd's Art

Point Value: 200

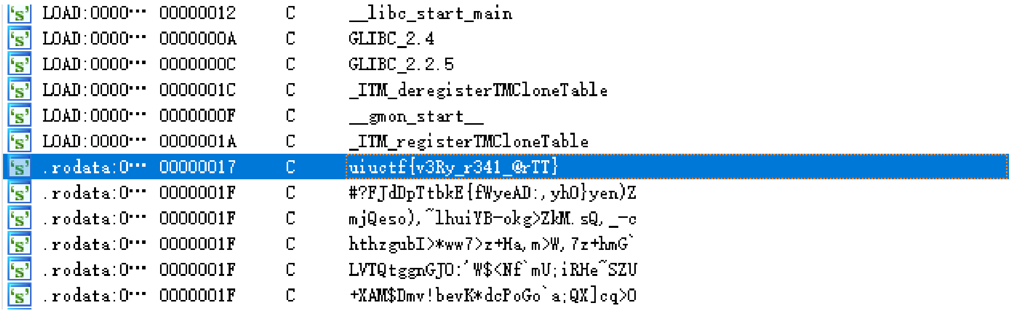
Challenge Description

Redd has an enticing deal for you. Will you take it?

Author: 2much4u

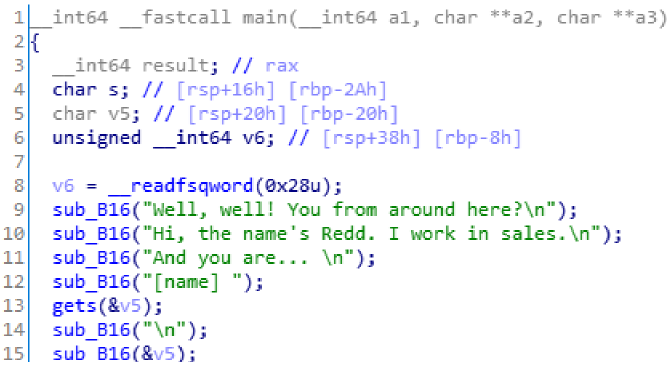
一道逆向题（截至写WP的时间，这比赛总共就2道逆向，一道200分的 一道500分的 =_=）

老规矩，用IDA打开，先Shift+F12看一看这是不是一道送分题

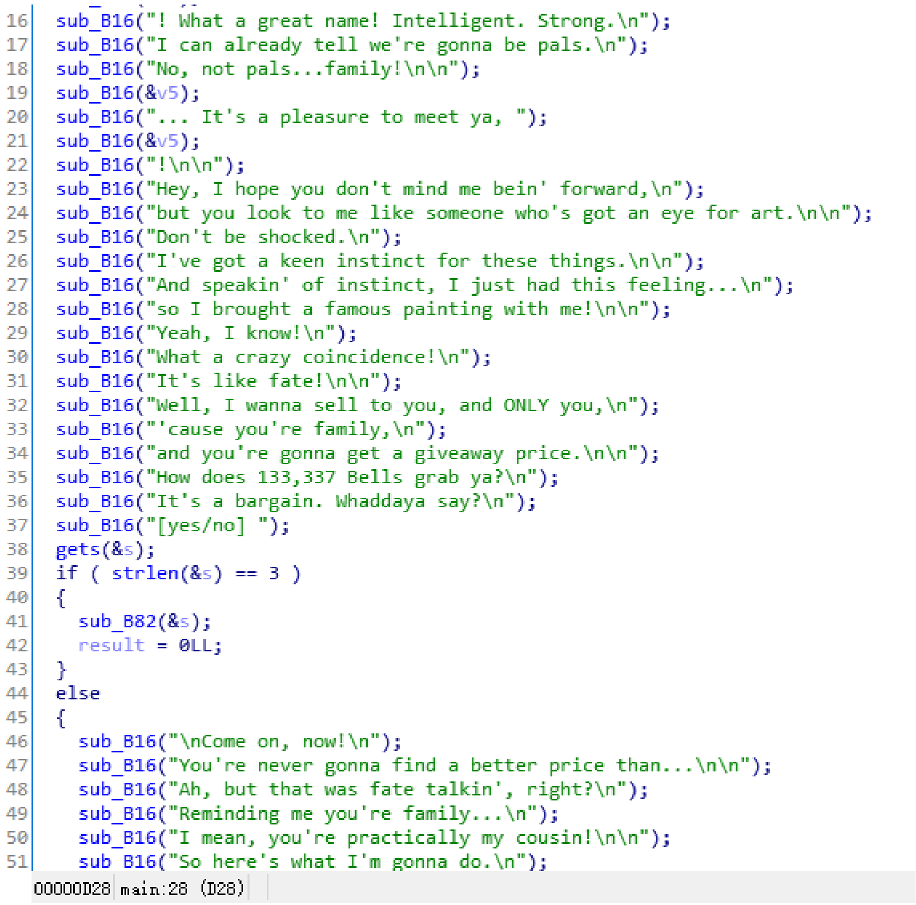


```
LOAD:0000···   00000012   C   __libc_start_main
LOAD:0000···   0000000A   C   GLIBC_2.4
LOAD:0000···   0000000C   C   GLIBC_2.2.5
LOAD:0000···   0000001C   C   _ITM_deregisterTMCloneTable
LOAD:0000···   0000000F   C   __gmon_start__
LOAD:0000···   0000001A   C   _ITM_registerTMCloneTable
.rodata:0···   00000017   C   uiuctf{v3Ry_r341_@rTT}
.rodata:0···   0000001F   C   #?FJdDpTtbkE{fWyeAD:,yh0}yen)Z
.rodata:0···   0000001F   C   mjQeso),~lhuiYB-okg>ZkM.sQ,_-c
.rodata:0···   0000001F   C   hthzgubI>*ww7>z+Ha,m>W,7z+hmG`
.rodata:0···   0000001F   C   LVTQtggnGJO:'W$<Nf`mU;iRHe~SZU
.rodata:0···   0000001F   C   +XAM$Dmv!bevK*dcPoGo`a;QX]cq>O
```

还真找着了，可惜提交了不正确，明显不是一道送分题



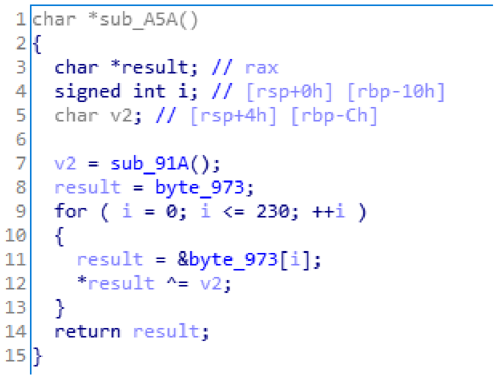
```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  __int64 result; // rax
  char s; // [rsp+16h] [rbp-2Ah]
  char v5; // [rsp+20h] [rbp-20h]
  unsigned __int64 v6; // [rsp+38h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  sub_B16("Well, well! You from around here?\n");
  sub_B16("Hi, the name's Redd. I work in sales.\n");
  sub_B16("And you are... \n");
  sub_B16("[name] ");
  gets(&v5);
  sub_B16("\n");
  sub_B16(&v5);
```



```
  sub_B16("! What a great name! Intelligent. Strong.\n");
  sub_B16("I can already tell we're gonna be pals.\n");
  sub_B16("No, not pals...family!\n\n");
  sub_B16(&v5);
  sub_B16("... It's a pleasure to meet ya, ");
  sub_B16(&v5);
  sub_B16("!\n\n");
  sub_B16("Hey, I hope you don't mind me bein' forward,\n");
  sub_B16("but you look to me like someone who's got an eye for art.\n\n");
  sub_B16("Don't be shocked.\n");
  sub_B16("I've got a keen instinct for these things.\n\n");
  sub_B16("And speakin' of instinct, I just had this feeling...\n");
  sub_B16("so I brought a famous painting with me!\n\n");
  sub_B16("Yeah, I know!\n");
  sub_B16("What a crazy coincidence!\n");
  sub_B16("It's like fate!\n\n");
  sub_B16("Well, I wanna sell to you, and ONLY you,\n");
  sub_B16("'cause you're family,\n");
  sub_B16("and you're gonna get a giveaway price.\n\n");
  sub_B16("How does 133,337 Bells grab ya?\n");
  sub_B16("It's a bargain. Whaddaya say?\n");
  sub_B16("[yes/no] ");
  gets(&s);
  if ( strlen(&s) == 3 )
  {
    sub_B82(&s);
    result = 0LL;
  }
  else
  {
    sub_B16("\nCome on, now!\n");
    sub_B16("You're never gonna find a better price than...\n\n");
    sub_B16("Ah, but that was fate talkin', right?\n");
    sub_B16("Reminding me you're family...\n");
    sub_B16("I mean, you're practically my cousin!\n\n");
    sub_B16("So here's what I'm gonna do.\n");
00000D28 main:28 (D28)
```

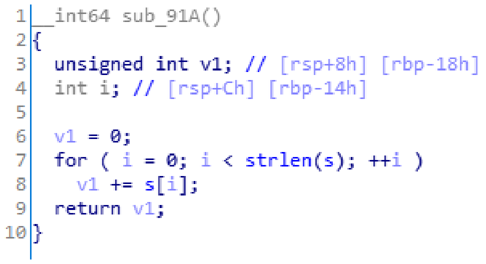
大致看了一下，main函数就是哔哔了一大堆，没有任何逻辑

于是从函数窗口中的函数里找一找



```
char *sub_A5A()
{
  char *result; // rax
  signed int i; // [rsp+0h] [rbp-10h]
  char v2; // [rsp+4h] [rbp-Ch]

  v2 = sub_91A();
  result = byte_973;
  for ( i = 0; i <= 230; ++i )
  {
    result = &byte_973[i];
    *result ^= v2;
  }
  return result;
}
```

找到一个明显是动态patch代码的函数，没有发现交叉引用

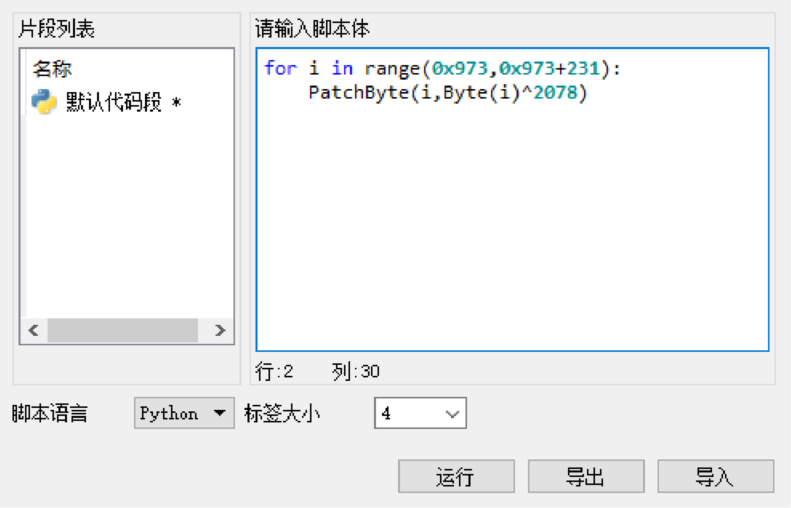


```
__int64 sub_91A()
{
  unsigned int v1; // [rsp+8h] [rbp-18h]
  int i; // [rsp+Ch] [rbp-14h]

  v1 = 0;
  for ( i = 0; i < strlen(s); ++i )
    v1 += s[i];
  return v1;
}
```

这个函数的作用是返回一个整数，把s的每个char转成int加起来，而这个s就是前面看到的假flag，写脚本跑一下很容易得到该函数返回的是2078

也就是说函数将0x973开始的231个字节异或2078，我们用IDAPython脚本跑一下

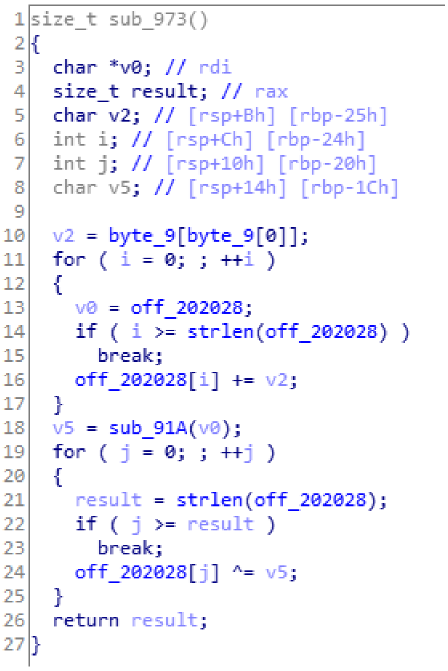
执行脚本



```
for i in range(0x973,0x973+231):
    PatchByte(i,Byte(i)^2078)
```

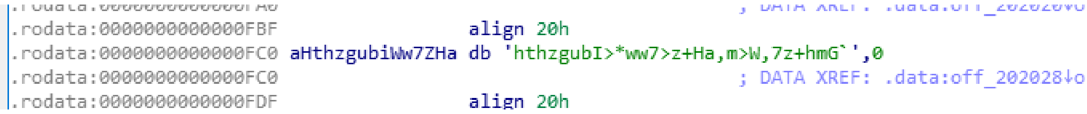
脚本语言 Python　标签大小 4

然后按P将数据转成函数（转不了的点击编辑-修补程序-应用到输入文件，然后用IDA再打开修补后的程序，应该自动就帮你转好了）



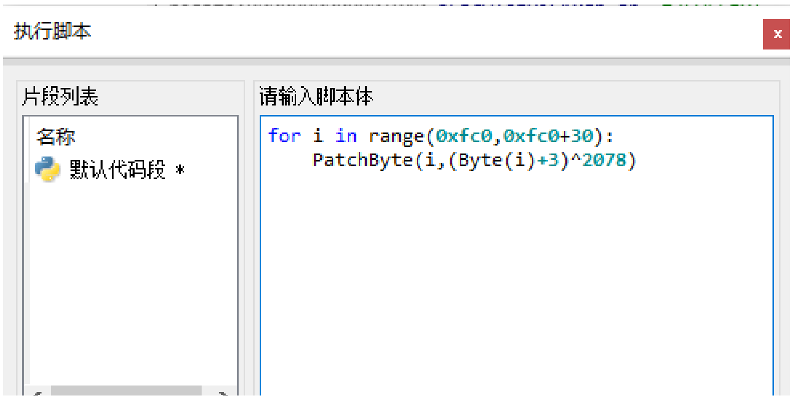
```
size_t sub_973()
{
  char *v0; // rdi
  size_t result; // rax
  char v2; // [rsp+Bh] [rbp-25h]
  int i; // [rsp+Ch] [rbp-24h]
  int j; // [rsp+10h] [rbp-20h]
  char v5; // [rsp+14h] [rbp-1Ch]

  v2 = byte_9[byte_9[0]];
  for ( i = 0; ; ++i )
  {
    v0 = off_202028;
    if ( i >= strlen(off_202028) )
      break;
    off_202028[i] += v2;
  }
  v5 = sub_91A(v0);
  for ( j = 0; ; ++j )
  {
    result = strlen(off_202028);
    if ( j >= result )
      break;
    off_202028[j] ^= v5;
  }
  return result;
}
```

v2很轻松能看出等于3

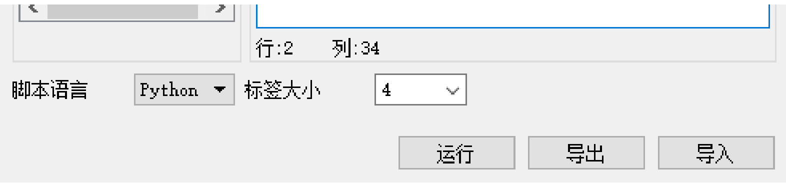


```
.rodata:0000000000000FBF                 align 20h
.rodata:0000000000000FC0 aHthzgubiWw7ZHa db 'hthzgubI>*ww7>z+Ha,m>W,7z+hmG`',0
.rodata:0000000000000FC0                                         ; DATA XREF: .data:off_202028↓o
.rodata:0000000000000FDF                 align 20h
```

off_202028是个指向一段乱码的地址，这个函数应该是给乱码解密，我们再用脚本跑一下



```
for i in range(0xfc0,0xfc0+30):
    PatchByte(i,(Byte(i)+3)^2078)
```

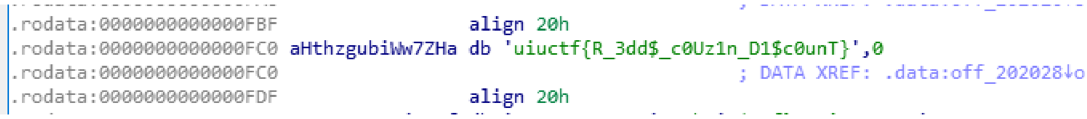


行:2　列:34

脚本语言 Python

标签大小 4

又一个flag出现啦~



```
.rodata:0000000000000FBF                  align 20h
.rodata:0000000000000FC0 aHthzgubiWw7ZHa db 'uiuctf{R_3dd$_c0Uz1n_D1$c0unT}',0
.rodata:0000000000000FC0                                          ; DATA XREF: .data:off_202028↓o
.rodata:0000000000000FDF                  align 20h
```

正当我在猜测是不是套娃题的时候，这个flag提交上去发现对了 (•̀ ω •́)y