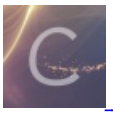
[BUUCTF]Reverse——[ACTF新生赛2020]SoulLike

原创

Angel~Yan 于 2021-04-07 15:53:08 发布 218 收藏

分类专栏： BUUCTF刷题记录 REVERSE

本文链接：https://blog.csdn.net/mcmuyanga/article/details/115485086



BUUCTF刷题记录 同时被 2 个专栏收录

198 篇文章 14 订阅

订阅专栏

REVERSE

75 篇文章 1 订阅

订阅专栏

[ACTF新生赛2020]SoulLike

1. ELF文件，应该没壳儿，在ubuntu里运行一下看看大概的情况

```
giantbranch@ubuntu:~/Desktop/ctf/buuctf/Reverse/ACTF新生赛2020-SoulLike$ ./SoulL
ike
input flag:123
Try another time...giantbranch@ubuntu:~/Desktop/ctf/buuctf/Reverse/ACTF新生赛202
```

2. 直接上ida了

```
  v10 = __readfsqword(0x28u);
  printf("input flag:");
  scanf("%s", &v9[6]);                          // 输入flag
  strcpy(v9, "actf{");
  v5 = 1;
  for ( i = 0; i <= 4; ++i )                    // 判断输入flag的前5位是否为actf{
  {
    if ( v9[i] != v9[i + 6] )
    {
      v5 = 0;
      goto LABEL_6;
    }
  }
  if ( !v5 )
    goto LABEL_16;
LABEL_6:
  for ( j = 0; j <= 11; ++j )
    v8[j] = v9[j + 11];                         // 将v9中11~22下标的值传入v8，就是我们输入的值actf{后面的数据
  if ( (unsigned __int8)sub_83A(v8) && v9[23] == '}' )// flag最后一位是"}"
  {
    printf("That's true! flag is %s", &v9[6]);
    result = 0LL;
  }
  else
  {
LABEL_16:
    printf("Try another time...");
    result = 0LL;
  }
```

```
  return result;
}
```


如果能满足sub_83A（）的条件，就能输出flag

看一下sub_83A（）

sub_83A（），将传进去的v8进行了3007行的异或操作，然后得到v3里的值，如果不对，会将错误的地方的下标打印出来

```
2998  a1[2] ^= 0x4Fu;
2999  a1[3] ^= 0x2Bu;
3000  a1[4] ^= *a1;
3001  a1[5] ^= 0x25u;
3002  a1[6] ^= 0x2Eu;
3003  a1[7] ^= 0x3Cu;
3004  a1[8] ^= 0x6Bu;
3005  a1[9] ^= 0x70u;
3006  a1[10] ^= 0x29u;
3007  a1[11] ^= 0x3Bu;
3008  v3[0] = 126;
3009  v3[1] = 50;
3010  v3[2] = 37;
3011  v3[3] = 88;
3012  v3[4] = 89;
3013  v3[5] = 107;
3014  v3[6] = 53;
3015  v3[7] = 110;
3016  v3[8] = 0;
3017  v3[9] = 19;
3018  v3[10] = 30;
3019  v3[11] = 56;
3020  for ( i = 0; i <= 11; ++i )
3021  {
3022    if ( v3[i] != a1[i] )
3023    {
3024      printf("wrong on #%d\n", (unsigned int)i);
3025      return 0LL;
3026    }
3027  }
3028  return 1LL;
3029}
```


拿v3里的值反向异或一下应该就能得到flag，但是将sub_83A（）里的代码复制出来后还要修改好一会儿，既然会将错误的那一位给打印出来，那就干脆爆破每一位吧，也不多，12位

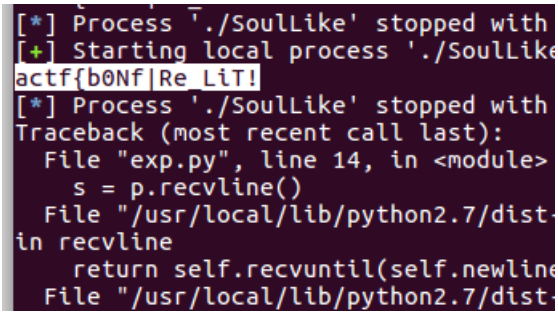
贴一下爆破exp

```
from pwn import *
import re

flag = "actf{"
# context.log_level="debug"
k = 0
for n in range(12):
    for i in range(33,127):
        p = process('./SoulLike')
        _flag = flag + chr(i)
        print _flag
        p.sendline(_flag)
        s = p.recvline()
        r = re.findall("on #(.*)\n", s)[0]
        r = int(r)
        if r==k+1:
            print s
            flag += chr(i)
     k+=1
 p.close()

print(flag)
```

不懂为什么最后结束的时候会报错，但是能爆破到正确的flag



flag{b0Nf|Re_LiT!}