

serial-150 攻防世界

原创

北风~ 于 2020-04-08 22:00:04 发布 880 收藏

分类专栏: [逆向与保护](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_45055269/article/details/105397707

版权



[逆向与保护](#) 专栏收录该内容

65 篇文章 4 订阅

订阅专栏

ida64 直接反编译, 你会发现没有函数, 而且代码段里是数据, 这一点决定让我们使用动态调试, 看程序跑起来之后, 代码段里的数据会变成什么代码。

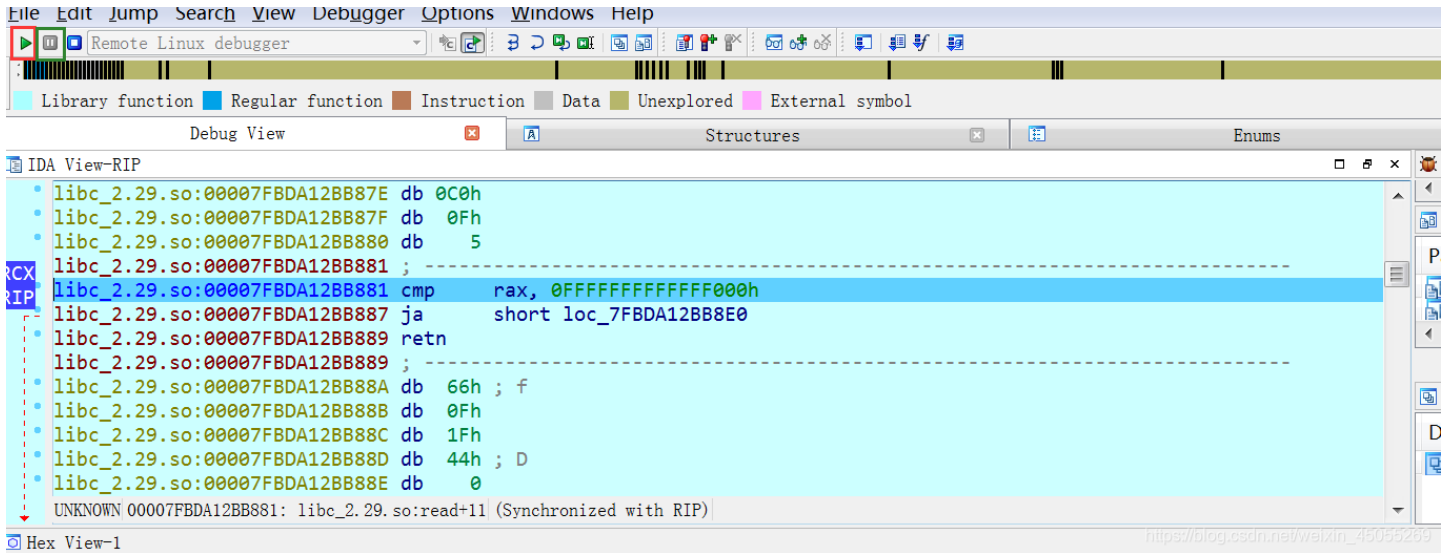
IDA远程动态调试配置: [无坑版](#)

这里分析如何找到输入字符后对字符检测的部分

1.刚打开IDA时的样子

```
IDA View-RIP
.text:000000000400890 ; Attributes: noreturn
.text:000000000400890
.text:000000000400890 public _start
.text:000000000400890 _start proc near
RIP R12 .text:000000000400890 xor     ebp, ebp
.text:000000000400892 mov     r9, rdx
.text:000000000400895 pop     rsi
.text:000000000400896 mov     rdx, rsp
.text:000000000400899 and     rsp, 0FFFFFFFFFFFFFF0h
.text:00000000040089D push   rax
.text:00000000040089E push   rsp
.text:00000000040089F mov     r8, offset __libc_csu_fini
.text:0000000004008A6 mov     rcx, offset __libc_csu_init
UNKNOWN 000000000400890: _start (Synchronized with RIP)
https://blog.csdn.net/weixin_45055269
```

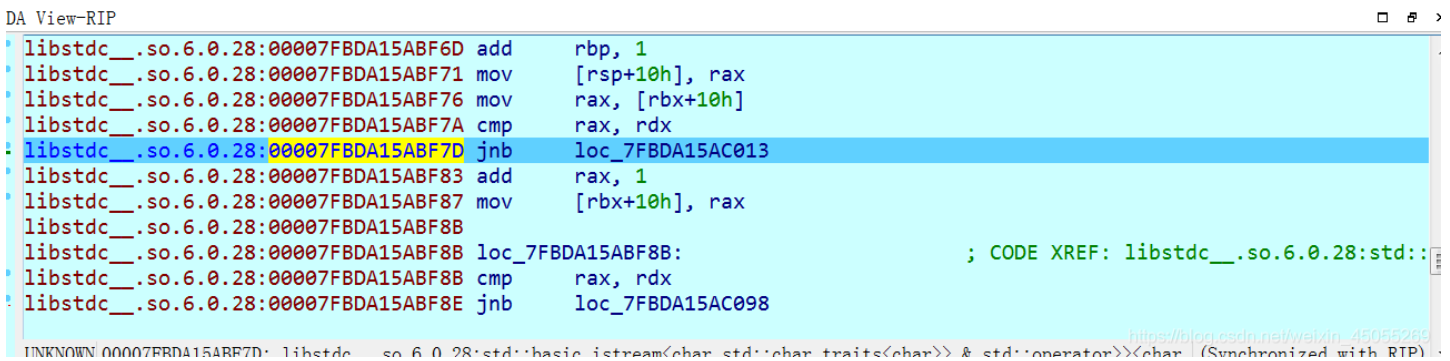
2.先点击红框, 看程序停下来, 再点击绿框内容



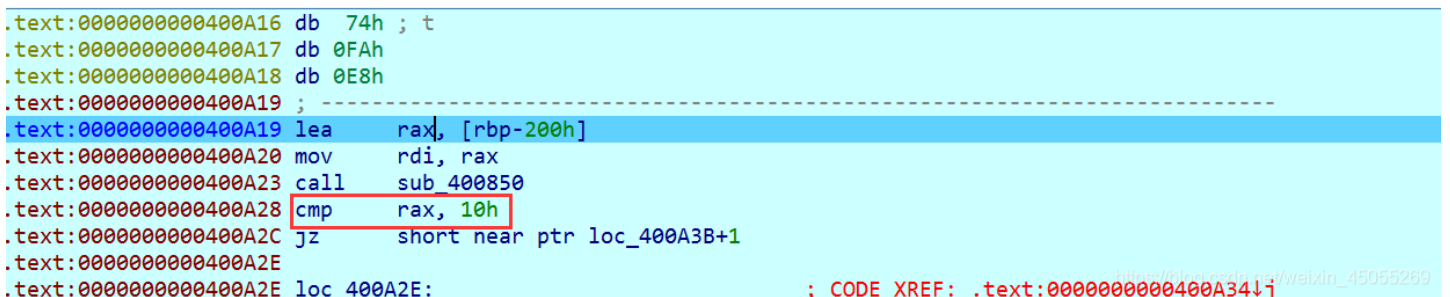
程序停下来，发现是让你输入字符先随便输入一串a



3.然后会发现进入7F开头的地址里，一直接F8，直到出现4开头的地址



4.发现关键比较，字符长度16



5.设置断点，不断满足条件写出flag

首位字符规定好，再规定末位字符加首位字符的和，这样两两一组逐渐往里缩，不断修改尝试

```

RIP .text:000000000400A16 db 74h ; t
    .text:000000000400A17 db 0FAh
    .text:000000000400A18 db 0E8h
    .text:000000000400A19 ; -----
    .text:000000000400A19 lea rax, [rbp-200h]
    .text:000000000400A20 mov rdi, rax
    .text:000000000400A23 call sub_400850
    .text:000000000400A28 cmp rax, 10h
    .text:000000000400A2C jz short near ptr loc_400A3B+1
    .text:000000000400A2E
    .text:000000000400A2E loc_400A2E: ; CODE XREF: .text:000000000400A34↓j
    .text:000000000400A2E mov ax, 5EBh
    .text:000000000400A32 xor eax, eax

```

https://blog.csdn.net/weixin_45055269

EZ9dmq4c8g9G7bAV