

# 南邮ctf题解--逆向第一道Hello,RE!

原创

Air\_cat 于 2019-04-02 17:40:41 发布 1784 收藏

分类专栏: [逆向工程 二进制CTF](#) 文章标签: [南邮ctf writeup](#)

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/Air\\_cat/article/details/88976977](https://blog.csdn.net/Air_cat/article/details/88976977)

版权



[逆向工程](#) 同时被 2 个专栏收录

25 篇文章 0 订阅

订阅专栏



[二进制CTF](#)

21 篇文章 0 订阅

订阅专栏

题目链接: <http://ctf.nuptzj.cn/challenges#>

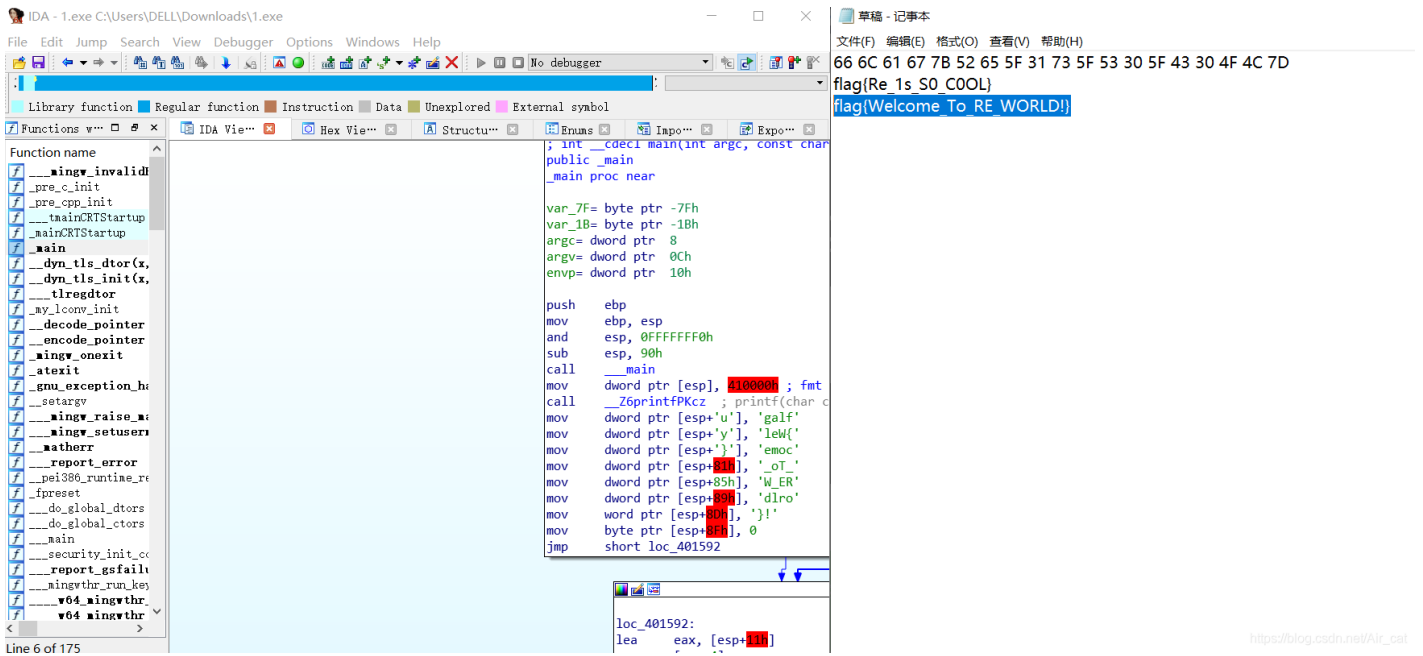
先总结一下, 南邮的题横向对比一下真的是简单(第一次自己做出来的题, 泪目)。下载了源程序之后, 题目都明示了用ida, 那就用ida呗

```
Hex View-1 | Structures | Enums | Imports
; int __cdecl main(int argc, const char **argv, const char **envp)
public _main
_main proc near

var_7F= byte ptr -7Fh
var_1B= byte ptr -1Bh
argc= dword ptr 8
argv= dword ptr 0Ch
envp= dword ptr 10h

push    ebp
mov     ebp, esp
and     esp, 0FFFFFF0h
sub     esp, 90h
call    __main
mov     dword ptr [esp], offset fmt ; fmt
call    __Z6printfPKcz ; printf(char const*,...)
mov     dword ptr [esp+75h], 67616C66h
mov     dword ptr [esp+79h], 6C655778h
mov     dword ptr [esp+7Dh], 656D6F63h
mov     dword ptr [esp+81h], 5F6F545Fh
mov     dword ptr [esp+85h], 575F4552h
mov     dword ptr [esp+89h], 646C726Fh
mov     word ptr [esp+8Dh], 7D21h
mov     byte ptr [esp+8Fh], 0
jmp     short loc_401592
```

可以看到printf的那部分底下是一大堆字符串, 很明显flag就在这里, 不过是用ascii码的形式而已, 接下来有两种方法, 一个是照着这个翻译(比较蠢), 另一个就是利用ida强大的功能, 将光标移到这些字符串上摁r键就好了



乍一看是不是发现是不明白的字符串？因为需要注意的是一般我们默认在汇编里的是小端序（也需要了解不同处理器架构不同），需要转换回正常的格式。

提交，通过。