BUUCTF 新手五道题

原创

别害怕我在 于 2021-07-15 09:47:03 发布 354 收藏

分类专栏： CTF逆向reverse新手 文章标签： 反编译

本文链接：https://blog.csdn.net/afanzcf/article/details/118752327

BUUCTF五道reverse新手题目

1.易怒

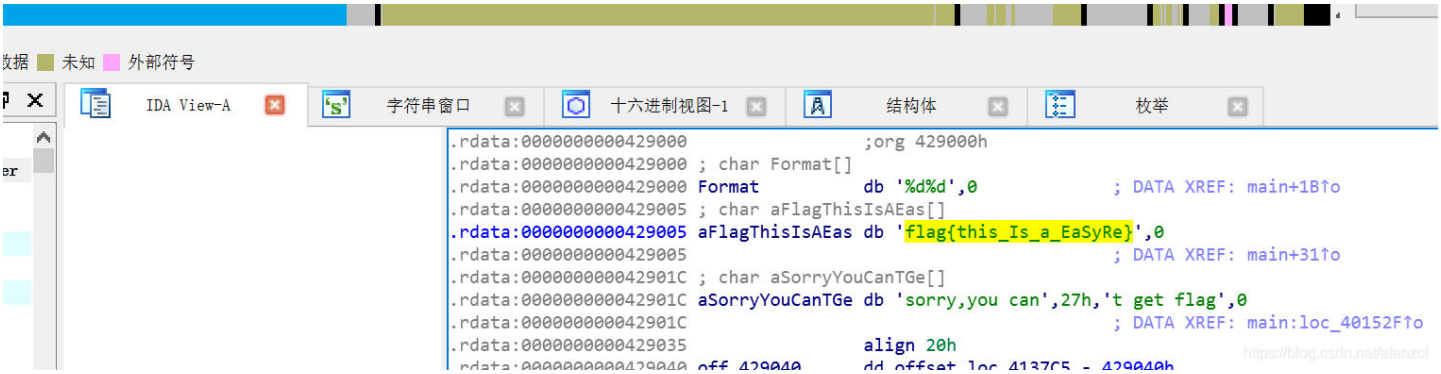
考点：基础题，shift+F12出字符串显示窗口

下载文件之后，是一个压缩包。正常解压。

打开之后，没有任何东西，输入字符之后，直接就退出。

PE查软件之后，64位，直接IDA打开。

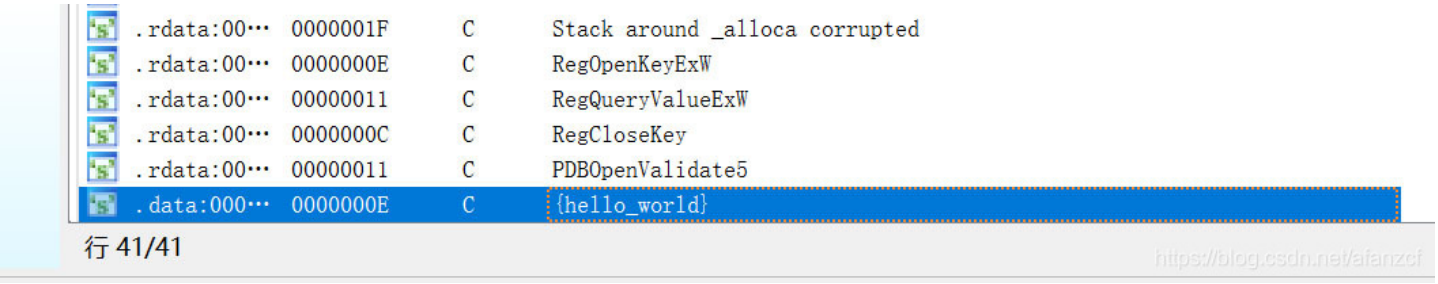
然后shift+F12 显示出字符串窗口，flag直接出现。



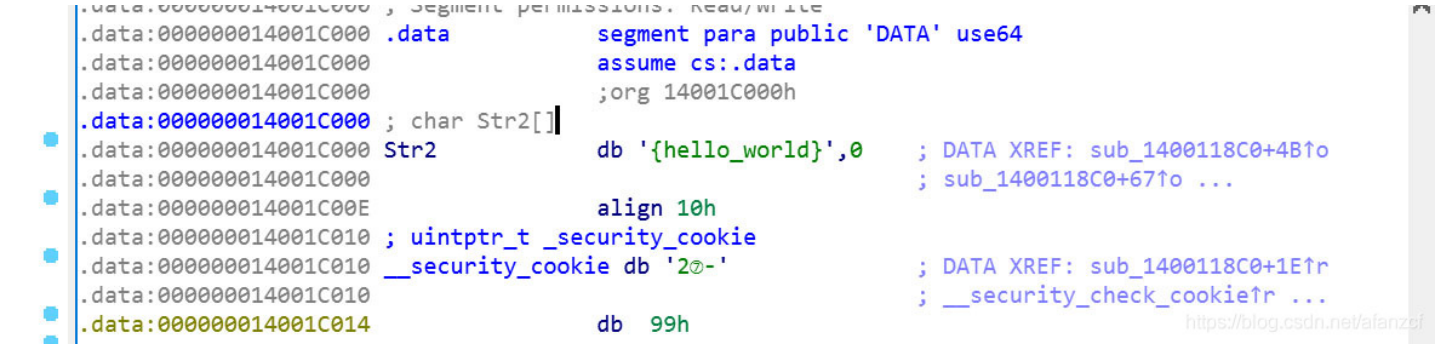
2.reverse1

考点：

首先下载文件之后，又是一个压缩包，解压之后打开。
输入字符之后，直接退出了。
继续用PE查看软件的基本信息，64位，无壳。
用64位的IDA打开。
没有main函数。
直接shift+F12 查看字符串窗口。

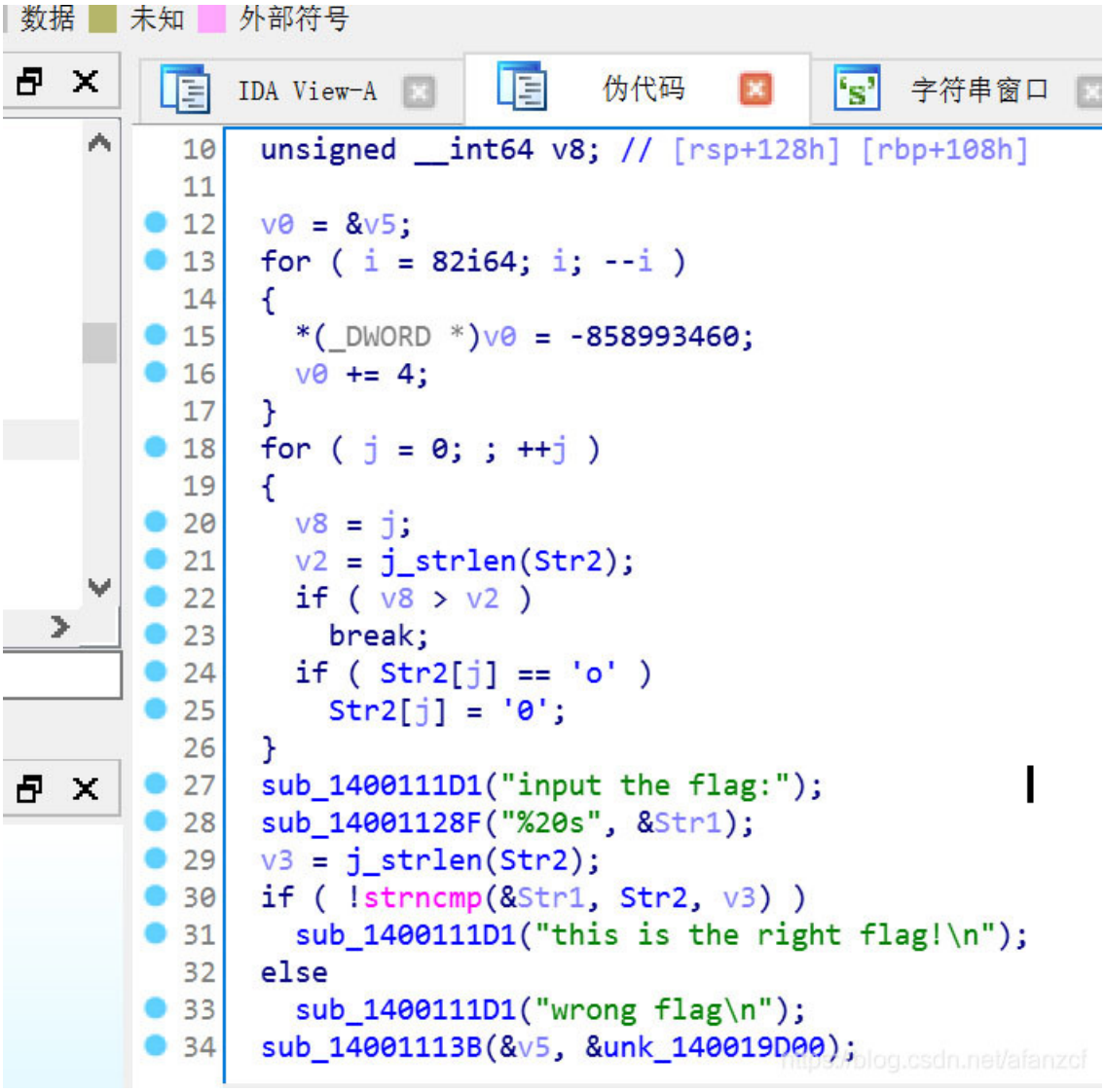


找到一个好像是flag的{hello_world}直接复制加上flag，答案错误。
双击{hello_world}



可以看到在 sub_1400118C0处被引用了。
去找到 sub_1400118C0。
按F5之后，反编译。
疑惑：F5反编译不是只能在32位中的IDA中才可以么？如果在IDA64位中也可以的话，那怎么判断使用F5反编译。（后面知道了在64位和32位中都可以反编译）

何时使用，判断依据是什么？



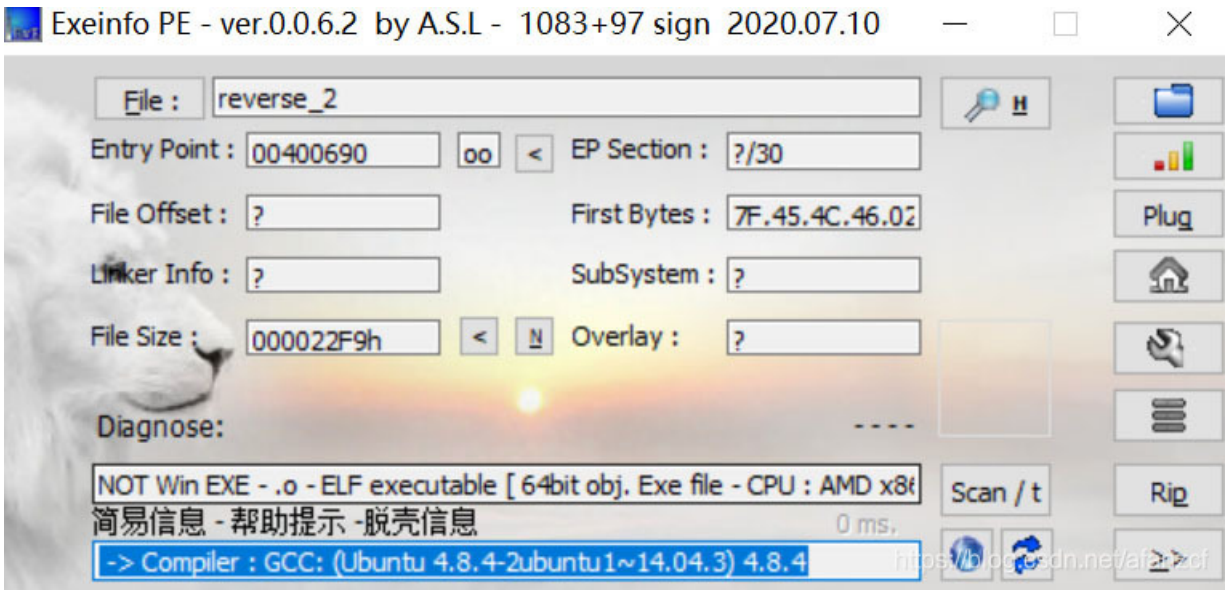
分析伪代码。

在if语句中，我们发现，如果str【2】== o ，那么就会把它变成 0。
所有我们把{hello_world}中的o 变成 0
得到flag{hell0_w0rld}

总结：这个题，最关键的一步就是F5反编译查看伪代码。之前我一直以为在IDA64位中不能反编译。
增加了一个知识点。能大概看懂了，被哪个地址引用的。

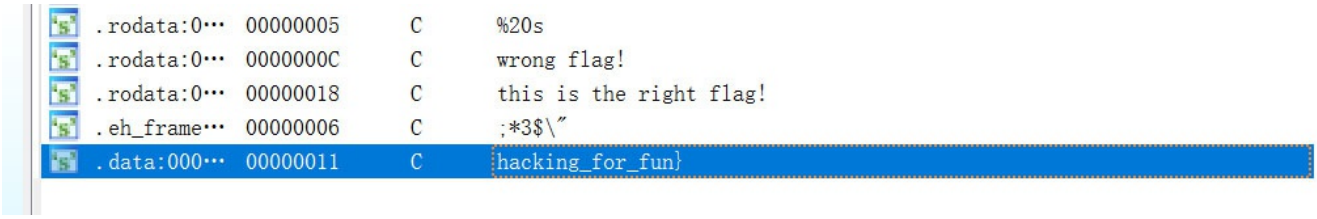
3.reverse2

考点：跟reverse1 一样。主要F5反编译。

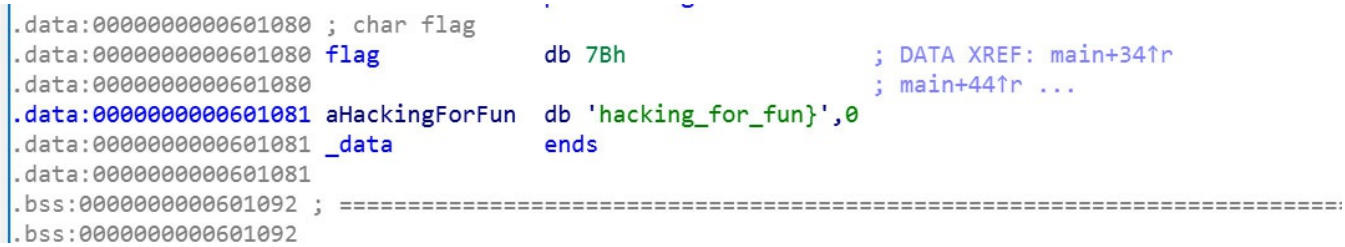


不是exe文件，用PE查看。

IDA64位打开。
继续shift+F12查看字符串窗口。



找到好像flag的字符串，双击。



好像是在主main函数中引用了。
找到主main函数，直接F5反编译。

```
unsigned __int64 v8; // [rsp+28h] [rbp-18h]

v8 = __readfsqword(0x28u);
pid = fork();
if ( pid )
{
  argv = (const char **)&stat_loc;
  waitpid(pid, &stat_loc, 0);
}
else
{
  for ( i = 0; i <= strlen(&flag); ++i )
  {
    if ( *(&flag + i) == 'i' || *(&flag + i) == 'r' )
      *(&flag + i) = '1';
  }
}
printf("input the flag:", argv);
__isoc99_scanf("%20s", &s2);
if ( !strcmp(&flag, &s2) )
  result = puts("this is the right flag!");
else
  result = puts("wrong flag!");
return result;
```


接下来分析代码。

这里其实已经得到答案了，一开始没注意if语句中是没有{}的，但是后面那个*(&flag+i) = ' 1 '，就是{}中的语句。

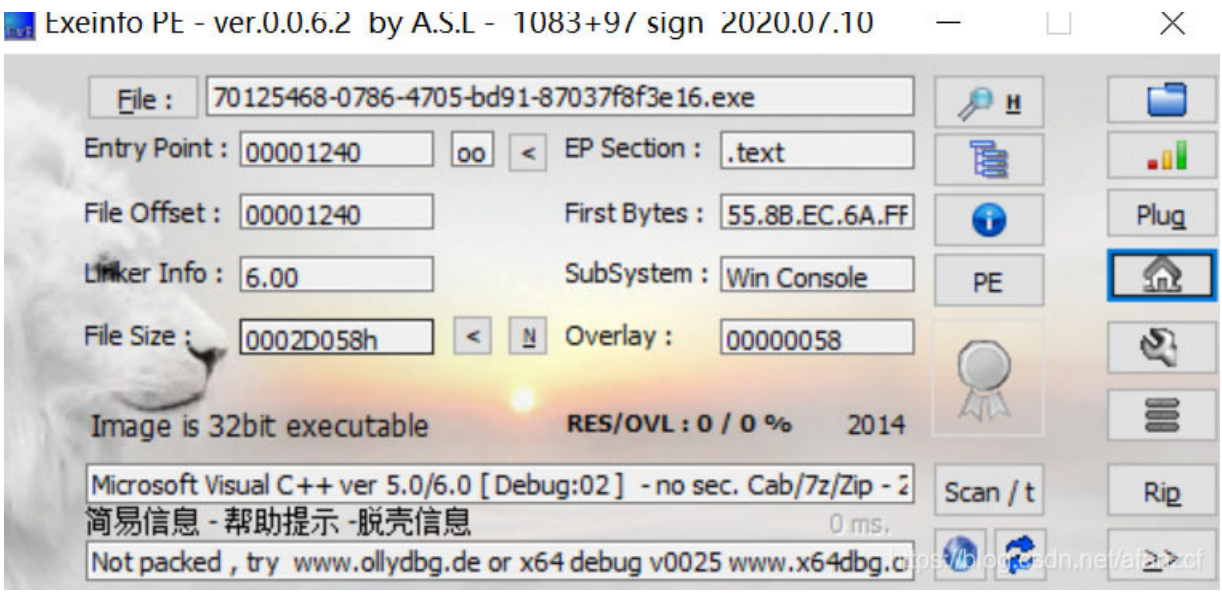
所以这里的意思是把flag中的 i 和 r 换成 1。

得到flag{hack1ng_fo1_fun}。

4.软件的内涵

考点：MD5解密？其实是考脑洞。。。注意结合他的题意和提示。

PE。



32位。

IDA打开。

找到_main_0函数，F5反编译。

```
  "这里本来应该是答案的,但是粗心的程序员忘记把变量写进来了,你要不逆向试试看:(Y/N)\n");
v1 = 1;
scanf("%c", &v1);
if ( v1 == 89 )
{
  printf(aOdIda);
  result = sub_40100A();
}
else
{
  if ( v1 == 78 )
    printf(asc_425034);
  else
    printf("输入错误,没有提示.");
  result = sub_40100A();
}
return result;
```

```
000106F _main_0:23 (40106F)
```


这里就是软件打开之后，按Y提示，和N提示的if else判断语句。

```
 2 {
 3   int result; // eax
 4   char v1; // [esp+4Ch] [ebp-Ch]
 5   const char *v2; // [esp+50h] [ebp-8h]
 6   int v3; // [esp+54h] [ebp-4h]
 7
 8   v3 = 5;
 9   v2 = "DBAPP{49d3c93df25caad81232130f3d2ebfad}";
10   while ( v3 >= 0 )
11   {
12     printf(aD, v3);
13     sub_40100A();
14     --v3;
15   }
```


答案应该在这一串字符串中。

一开始想的是这一串字符解密？

疑问：怎么判断是MD5解密。

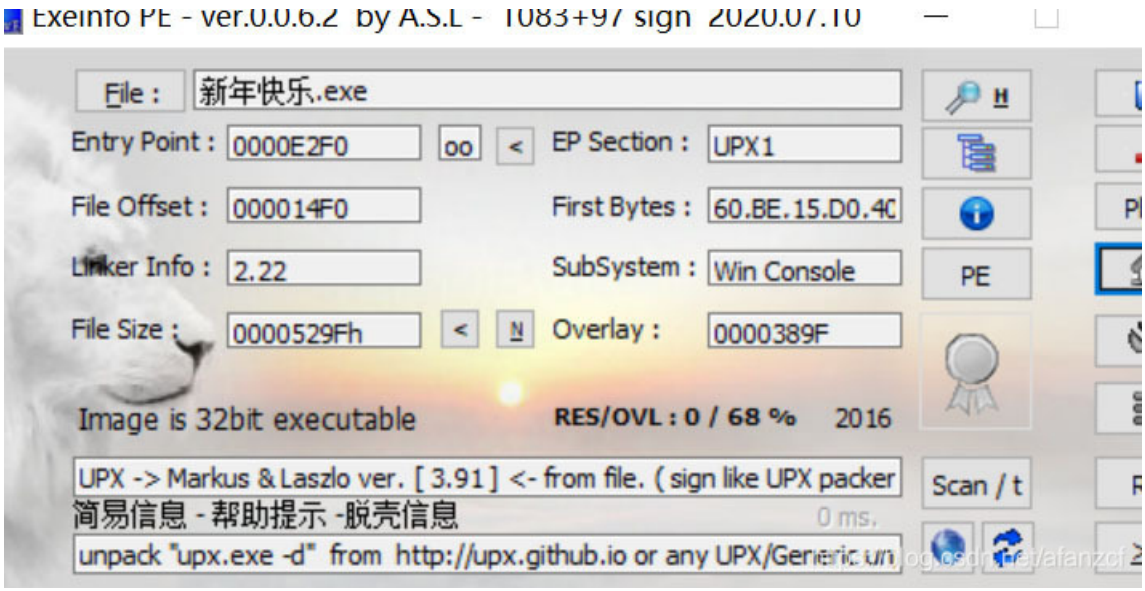
没想到这是MD5加密。

难道flag是 flag{全国人民震惊了}？
不对。。。。。
又去其他函数搞了半天，越搞越复杂。
后面发现这是脑洞题。。。

最后flag其实就是那串字符串。
flag{49d3c93df25caad81232130f3d2ebfad}

5.新年快乐

考点： upx脱壳， upx -d xxxxx



新年快乐

1

过年了要不做个逆向题庆祝一下新年？说不定会有惊喜哦！注意：flag并非是flag{XXX}形式，就是一个字符串，考验眼力的时候到了！ 注意：得到的 flag 请包上 flag{} 提交

根据上一题的经验，结合题意。。。。

首先是32位的，并且有UPX壳。
首先要脱壳。

```
:\桌面\网络安全工作室\upx-3.96-win64>upx -d 新年快乐.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2020
PX 3.96w        Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     27807 <-     21151   76.06%    win32/pe     新年快乐.exe

npacked 1 file.
```


之前脱过一次upx的壳。这次也是顺利一次操作成功。。。

地址	长度	类型	字符串
.rdata:00…	0000000E	C	libgcj-13.dll
.rdata:00…	00000014	C	_Jv_RegisterClasses
.rdata:00…	0000001C	C	please input the true flag:
.rdata:00…	00000013	C	this is true flag!
.rdata:00…	00000007	C	wrong!
.rdata:00…	0000000E	C	HappyNewYear!
.rdata:00…	00000018	C	Mingw runtime failure:\n
.rdata:00…	00000031	C	VirtualQuery failed for %d bytes at address %p
.rdata:00…	00000032	C	Unknown pseudo relocation protocol version %d.\n
.rdata:00…	0000002A	C	Unknown pseudo relocation bit size %d.\n
.eh_frame…	00000007	C	`t\n徘泼
.eh_frame…	00000005	C	K\n闷

shift+F12 查看字符串窗口，发现一串字符，是新年快乐的意思。
结合题意，flag就是一个字符串，再结合题目是新年快乐。
的壳。这次也是顺利一次操作成功。。。

[外链图片转存中…(img-5JTvwoDL-1626313229157)]

shift+F12 查看字符串窗口，发现一串字符，是新年快乐的意思。
结合题意，flag就是一个字符串，再结合题目是新年快乐。
flag{HappyNewYear!