

BUUCTF解析

原创

霖~霖 于 2022-02-21 14:42:00 发布 39 收藏

文章标签: [p2p](#) [linux](#) [开发语言](#)

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/qq_63676207/article/details/123015970

版权

easyre 1

这个在IDI中打开用 **SHIFT+F12** 查看字符串窗口发现flag

段	地址	长度	类型	字符串
.t	.rdata:0...	00000005	C	%d%d
.t	.rdata:0...	00000017	C	flag{this_Is_a_EaSyRe}
.t	.rdata:0...	00000019	C	sorry,you can't get flag
.t	.rdata:0...	0000000F	C	std::exception
.t	.rdata:0...	00000013	C	std::bad_exception
.t	.rdata:0...	0000000B	C	eh_globals
.t	.rdata:0...	00000005	C	init

CSDN @霖~霖

reverse1

下好后是一个压缩包的形式于是解压后查看是一个exe文件倒是不能用



CSDN @霖~霖

接着用IDI打开并SHIFT+F12 查看字符串窗口

段	地址	长度	类型	字符串
.t	.rdata:0...	00000009	C	_ArgList
.t	.rdata:0...	0000000C	C	wrong flag\n
.t	.rdata:0...	00000009	C	_ArgList
.t	.rdata:0...	00000019	C	this is the right flag!\n
.t	.rdata:0...	00000006	C	input
.t	.rdata:0...	00000005	C	%20s
.t	.rdata:0...	00000010	C	input the flag:
.t	.rdata:0...	0000002B	C	' is being used without being initialized.
.t	.rdata:0...	0000001C	C	Stack around the variable '
.t	.rdata:0...	00000011	C	' was corrupted.
.t	.rdata:0...	0000000F	C	The variable '
.t	.rdata:0...	000000DD	C	The value of ESP was not properly saved across a function cal...

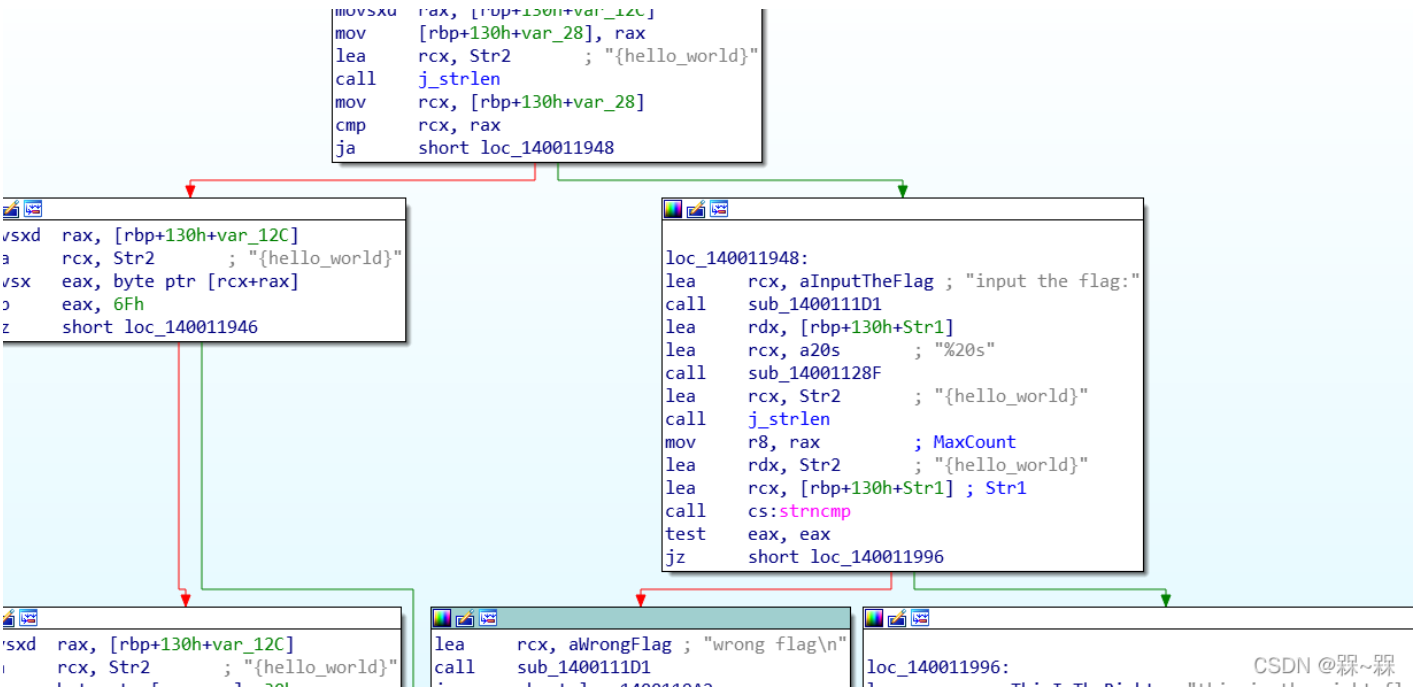
CSDN @霖~霖

发现flag进行跟进

```
.rdata:000000140019C06 db 0
.rdata:000000140019C07 db 0
.rdata:000000140019C08 dq offset asc_140019BC0 ; "H"
.rdata:000000140019C10 aWrongFlag db 'wrong flag',0Ah,0 ; DATA XREF: sub_1400118C0+C8f0
.rdata:000000140019C1C db 0
.rdata:000000140019C1D db 0
.rdata:000000140019C1E db 0
.rdata:000000140019C1F db 0
.. -
```

CSDN @霖~霖

再次跟进



发现flag 提示是hello_world 并且与Str2函数有关F5查看

```
7 char v5; // [rsp+0h] [rbp-20h]
8 int j; // [rsp+24h] [rbp+4h]
9 char Str1; // [rsp+48h] [rbp+28h]
10 unsigned __int64 v8; // [rsp+128h] [rbp+108h]
11
12 v0 = &v5;
13 for ( i = 82i64; i; --i )
14 {
15     *(_DWORD *)v0 = -858993460;
16     v0 += 4;
17 }
18 for ( j = 0; ; ++j )
19 {
20     v8 = j;
21     v2 = j_strlen(Str2);
22     if ( v8 > v2 )
23         break;
24     if ( Str2[j] == 111 )
25         Str2[j] = 48;
26 }
27 sub_1400111D1("input the flag:");
28 sub_14001128F("%20s", &Str1);
29 v3 = j_strlen(Str2);
30 if ( !strcmp(&Str1, Str2, v3) )
31     sub_1400111D1("this is the right flag!\n");
32 else
33     sub_1400111D1("wrong flag\n");
34 sub_14001113B(&v5, &unk_140019D00);
```

CSDN @霖~霖

发现有一个for循环并包含一个if比较 字符与数字选中数字+R进行转义

```
if ( Str2[j] == 'o' )
    Str2[j] = '0';
sub_1400111D1("input the flag:");
```

及意思是将o变成0

得到flag{hello_w0rld}

reverse2

一个没有提示的压缩包解压后是一个

e8722e94-93d7-45d5-aa06-a7aa26c...	2022/2/19 11:31	RAR 文件	3 KB
reverse_2	2016/9/4 16:33	文件	9 KB

CSDN @霖~霖

用神IDI进行反汇编

.plt	LOAD:000...	00000011	C	__stack_chk_fail	
t_main	.plt	LOAD:000...	00000007	C	printf
	.plt	LOAD:000...	00000007	C	strlen
-	.plt	LOAD:000...	00000008	C	waitpid
	.plt	LOAD:000...	00000007	C	strcmp
f	.plt	LOAD:000...	00000012	C	__libc_start_main
	.plt	LOAD:000...	0000000F	C	__gmon_start__
clones	.text	LOAD:000...	0000000A	C	GLIBC_2.7
ones	.text	LOAD:000...	0000000A	C	GLIBC_2.4
ors_aux	.text	LOAD:000...	0000000C	C	GLIBC_2.2.5
	.text	.rodata:...	00000010	C	input the flag:
	.text	.rodata:...	00000005	C	%20s
	.text	.rodata:...	0000000C	C	wrong flag!
t	.text	.rodata:...	00000018	C	this is the right flag!
i	.text	.eh_frame	00000006	C	;*3\$\"
	.fini	.data:00...	00000011	C	hacking_for_fun}
	.exte:				
	.exte:				
il	.exter				
	.exte:				

CSDN @霖~霖

发现有flag提示还有个比较特殊的{hacking_for_fun}不鸟他打开flag

```
.init 2{
.plt 3
.plt 4 int result; // eax
.plt 5 int stat_loc; // [rsp+4h] [rbp-3Ch]
.plt 6 int i; // [rsp+8h] [rbp-38h]
.plt 7 __pid_t pid; // [rsp+Ch] [rbp-34h]
.plt 8 char s2; // [rsp+10h] [rbp-30h]
.plt 9 unsigned __int64 v8; // [rsp+28h] [rbp-18h]
.plt 10
.plt 11 v8 = __readfsqword(0x28u);
.plt 12 pid = fork();
.plt 13 if ( pid )
.plt 14 {
.plt 15     argv = (const char **)&stat_loc;
.plt 16     waitpid(pid, &stat_loc, 0);
.plt 17 }
.plt 18 else
.plt 19 {
.plt 20     for ( i = 0; i <= strlen(&flag); ++i )
.plt 21     {
.plt 22         if ( *(&flag + i) == 105 || *(&flag + i) == 114 )
.plt 23             *(&flag + i) = 49;
.plt 24     }
.plt 25 }
.plt 26 printf("input the flag:", argv);
.plt 27 __isoc99_scanf("%20s", &s2);
.plt 28 if ( !strcmp(&flag, &s2) )
.plt 29     result = puts("this is the right flag!");
.plt 30 else
.plt 31     result = puts("wrong flag!");
.plt 32 return result;
.plt 33 }
```

CSDN @霖~霖

也是和上一题一致是一个字符替换转换后

```

ext 16 }
ext 17 else
ext 18 {
ex 19     for ( i = 0; i <= strlen(&flag); ++i )
ext 20     {
ni 21         if ( *(&flag + i) == 'i' || *(&flag + i) == 'r' )
te 22             *(&flag + i) = '1';
te 23     }
er 24 }

```

CSDN @霖~霖

但没有对比文本于是想到这个特殊的{hacking_for_fun}进行替换后即得到

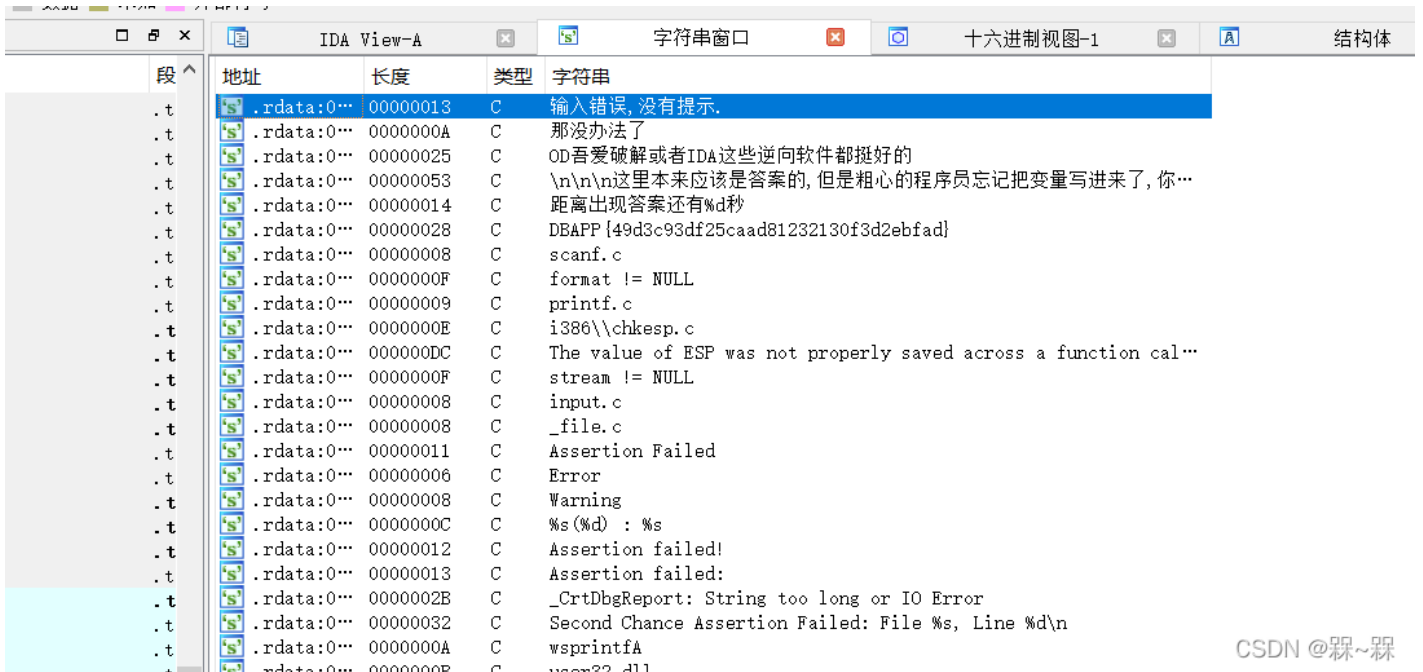
flag{{hack1ng_fo1_fun}}

内涵的软件

这个是一个exe文件打开后是这样的



有点被耍的感觉有IDA打开



CSDN @霖~霖

直接得到flag别想多千万别想多这就是flag



[创作打卡挑战赛](#)

[赢取流量/现金/CSDN周边激励大奖](#)