

攻防世界 Shuffle

原创

别害怕我在  于 2021-08-06 20:33:25 发布  38  收藏

分类专栏: [CTF逆向reverse新手](#)

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/afanzcf/article/details/119462993>

版权



[CTF逆向reverse新手](#) 专栏收录该内容

20 篇文章 1 订阅

订阅专栏

title: 攻防世界 Shuffle

date: 2021年8月6日 20点06分

tags: 攻防世界

categories: 攻防世界

对于这道题, 能解出来, 有点意外, 因为直接降低难度了。但是虽然解出来flag, 但是里面的门道却是不太清楚。

上题:

Shuffle  1 最佳Writeup由admin提供

难度系数:  1.0

题目来源: [SECCON-CTF-2014](#)

题目描述: 找到字符串在随机化之前.

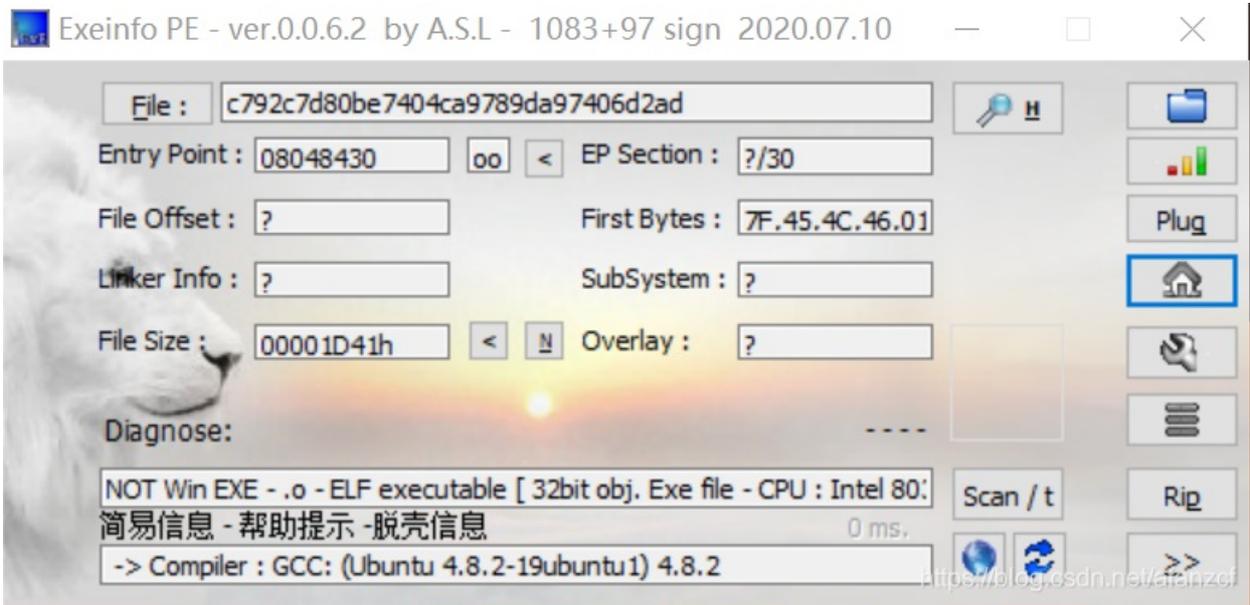
题目场景: 暂无

题目附件: [附件1](#)

<https://blog.csdn.net/afanzcf>

首先看题, 难度一星, 但是放在了高手进阶区, 集合题目描述, 大概猜测这个题的flag会变化?

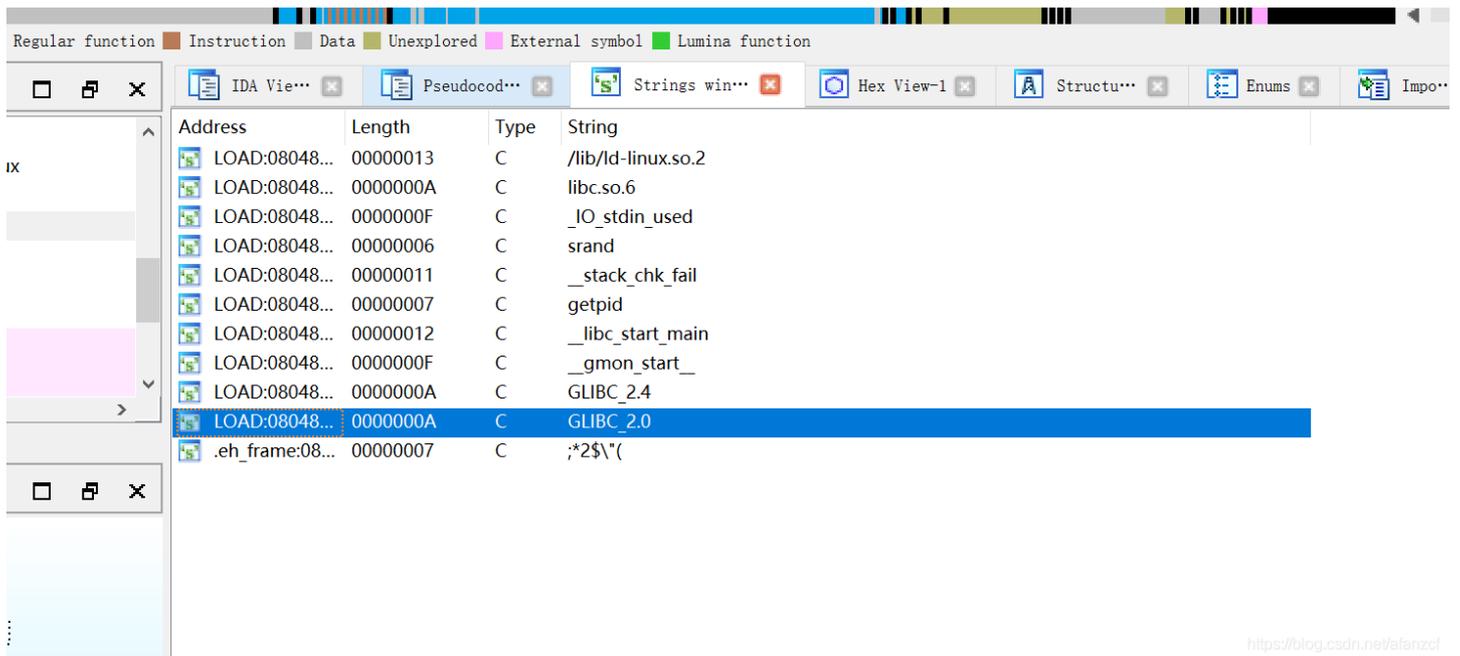
1、PE分析



又是一道ELF文件的，linux。32位，直接上手IDA32位

2、IDApr 32位

(1) shift + F12 查看字符串窗口



没有什么有用的信息。

(2) 找到main函数进入

```

push    ebp
mov     ebp, esp
push    esi
push    ebx
and     esp, 0FFFFFF0h
sub     esp, 50h
mov     eax, [ebp+argv]
mov     [esp+0Ch], eax
mov     eax, large gs:14h
mov     [esp+4Ch], eax
xor     eax, eax
mov     eax, 53h ; 'S'
mov     [esp+24h], al
mov     eax, 45h ; 'E'
mov     [esp+25h], al
mov     eax, 43h ; 'C'
mov     [esp+26h], al
mov     eax, 43h ; 'C'
mov     [esp+27h], al
mov     eax, 4Fh ; 'O'
mov     [esp+28h], al
mov     eax, 4Eh ; 'N'
mov     [esp+29h], al
mov     eax, 7Bh ; '{'
mov     [esp+2Ah], al
mov     eax, 57h ; 'W'
mov     [esp+2Ch], al
mov     eax, 6Ch ; 'l'
mov     [esp+2Dh], al
mov     dword ptr [esp], 0 ; timer
call    _time
mov     ebx, eax
call    _getpid
add     eax, ebx
mov     [esp], eax ; seed
call    _srand
mov     dword ptr [esp+14h], 0
jmp     loc_8048769

```

```

loc_8048769:
cmp     dword ptr [esp+14h], 63h ; 'c'
jle     loc_80486DD

```

```

lea     eax, [esp+58h+s]
mov     [esp], eax ; s
call    puts

```

```

loc_80486DD:
call    rand

```

这几个字符很是可疑，先放一边。

(3) 直接F5反汇编

```
4  __pid_t v4; // eax
5  int i; // [esp+14h] [ebp-44h]
6  unsigned int v7; // [esp+18h] [ebp-40h]
7  unsigned int v8; // [esp+1Ch] [ebp-3Ch]
8  char v9; // [esp+20h] [ebp-38h]
9  char s[40]; // [esp+24h] [ebp-34h] BYREF
10 unsigned int v11; // [esp+4Ch] [ebp-Ch]
11
12 v11 = __readgsdword(20u);
13 strcpy(s, "SECCON{Welcome to the SECCON 2014 CTF!}");
14 v3 = time(0);
15 v4 = getpid();
16 srand(v3 + v4);
17 for ( i = 0; i <= 99; ++i )
18 {
19     v7 = rand() % (unsigned int)';
20     v8 = rand() % 0x28u;
21     v9 = s[v7];
22     s[v7] = s[v8];
23     s[v8] = v9;
24 }
25 puts(s);
26 return 0;
27 }
```

000006D0 main:17 (80486D0)

<https://blog.csdn.net/a1anzcf>

把这个字符串复制给s，SECCON{Welcome to the SECCON 2014 CTF!}

v3 = time (0) ; //推测这个v3是不是多少时间，然后会变换flag?

v4 = getpid () ; //百度一波，getpid是一种函数，功能是取得进程识别码，许多程序利用取到的此值来建立临时文件，以避免临时文件相同带来的问题。

srand (v3 + v4) ; //srand函数是随机数发生器的初始化函数。原型：void srand(unsigned int seed);srand和rand()配合使用产生伪随机数序列。

推测一开始，先把字符串复制给s，然后通过v3，v4，还有srand函数，最后改变这个s的值，最后输出s，就是最终的答案。这是我的猜想，以及对这个题目的理解。

但是我随即灵机一动，SECCON{Welcome to the SECCON 2014 CTF!}，这个是不是就是flag，因为题目提示是在变化之前，找到字符串。果然不出所料。

(4) 总结

此题虽然被解出，但是我对这其中奥秘却是不解，百度一波wp，发现全都是怎么做出来这个题，这个题的深入都没讲，甚至大部分博客，全是复制粘贴别人的，到处都是一模一样的文章，或许这个题目是直接告诉了flag，但是如果其他题，就按这个思想，来一个随机生成flag，然后会变换的题目，又该怎么去跟踪，当然这还不是我现在所想的问题，肯定会有这样的问题，只不过我还没遇到。

继续变强，写出能让自己，让新手，恍然大悟的wp，那不是网上复制粘贴。