# buu-[ACTF新生赛2020]SoulLike

原创

有点水啊 于 2022-04-03 18:18:27 发布 140 收藏

分类专栏： buuctf-reserve 文章标签： 安全



本文链接：https://blog.csdn.net/qaq517384/article/details/123881365

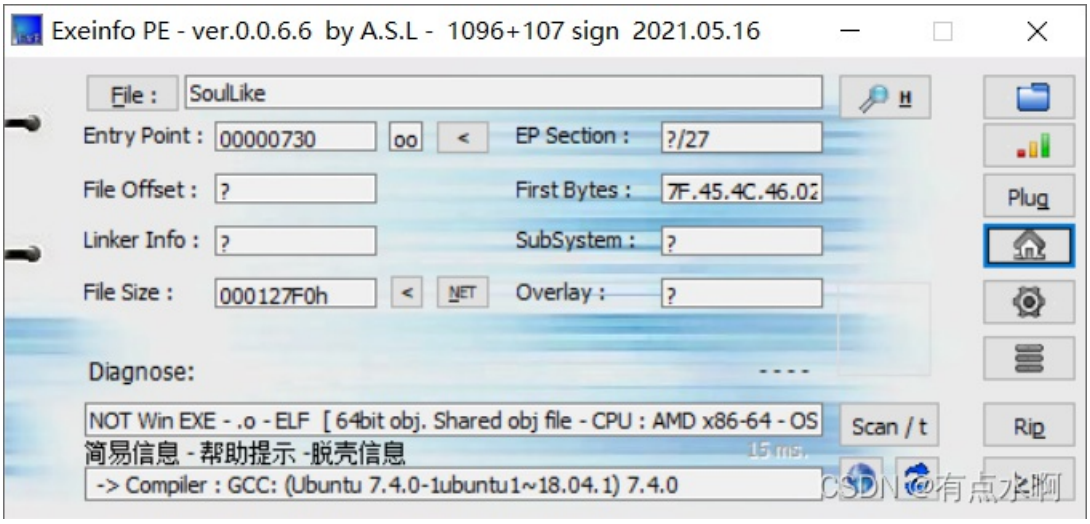


buuctf-reserve 专栏收录该内容

59 篇文章 0 订阅

订阅专栏

64位elf文件

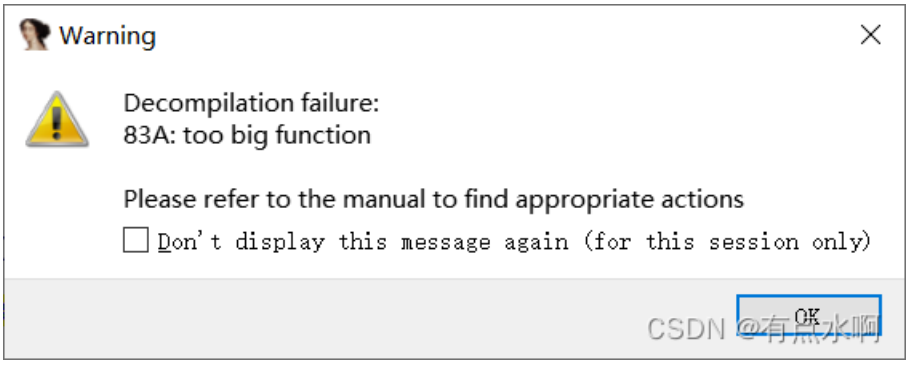



64位ida查看字符串跟进main函数

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  char v3; // al
  __int64 result; // rax
  char v5; // [rsp+7h] [rbp-B9h]
  signed int i; // [rsp+8h] [rbp-B8h]
  signed int j; // [rsp+Ch] [rbp-B4h]
  int v8[14]; // [rsp+10h] [rbp-B0h]
  int v9; // [rsp+4Ah] [rbp-76h]
  __int16 v10; // [rsp+4Eh] [rbp-72h]
  char v11[17]; // [rsp+50h] [rbp-70h]
  char v12; // [rsp+61h] [rbp-5Fh]
  unsigned __int64 v13; // [rsp+B8h] [rbp-8h]

  v13 = __readfsqword(0x28u);
  printf("input flag:", a2, a3);
  scanf("%s", v11);
  v9 = 'ftca';
  v10 = '{';
  v5 = 1;
  for ( i = 0; i <= 4; ++i )
  {
    if ( *((_BYTE *)&v9 + i) != v11[i] )
    {
      v5 = 0;
      goto LABEL_6;
    }
  }
  if ( !v5 )
    goto LABEL_19;
LABEL_6:
  for ( j = 0; j <= 11; ++j )
    v8[j] = v11[j + 5];
  v3 = (unsigned __int8)sub_83A(v8) && v12 == '}' ? 1 : 0;
  if ( v3 )
  {
    printf("That's true! flag is %s", v11);
    result = 0LL;
  }
  else
  {
LABEL_19:
    printf("Try another time...");
    result = 0LL;
  }
  return result;
}
```

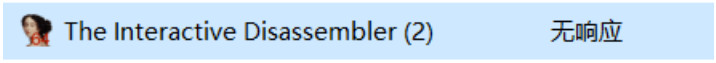
上下是判定头字符串的`actf{}`
主要看sub_83A
ida提示too big function




百度得到只需要修改配置文件`IDA\cfg\hexrays.cfg`

```
找到：
MAX_FUNCSIZE            = 64        // Functions over 64K are not decompiled
修改为：
MAX_FUNCSIZE            = 1024        // Functions over 64K are not decompiled
```

重启ida，让函数再跑一会儿




卡完是一个三千多行的函数

```
3021  v4 = 126;
3022  v5 = 50;
3023  v6 = 37;
3024  v7 = 88;
3025  v8 = 89;
3026  v9 = 107;
3027  v10 = 53;
3028  v11 = 110;
3029  v12 = 0;
3030  v13 = 19;
3031  v14 = 30;
3032  v15 = 56;
3033  for ( i = 0; i <= 11; ++i )
3034  {
3035    if ( *(&v4 + i) != a1[i] )
3036    {
3037      printf("wrong on #%d\n", (unsigned int)i);
3038      return 0LL;
3039    }
3040  }
3041  return 1LL;
3042}
```



总结：
输入的值经历三千行的变换后等于数组v4的值
逆向是不可能逆向的，直接开始爆破

```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int v4[]={126,50,37,88,89,107,53,110,0,19,30,56};
int sub_83A(char *v1,int i)
{
 *v1 ^= 43u;
 v1[1] ^= 108u;
 v1[2] ^= 126u;
 v1[3] ^= 86u;
 //略 直接复制源代码就行
 v1[8] ^= 0x6Bu;
 v1[9] ^= 0x70u;
 v1[10] ^= 0x29u;
 v1[11] ^= 0x3Bu;
 if(v1[i] == v4[i])
  return 1;
 else
     return 0;
}

int main()
{
 int i,j;
 char flag[13] = "";
 char tmp[13] = "";
 for(i = 0; i < 12 ; i++)
 {
     for(j = 33; j <= 126 ; j++)
     {
         strcpy(flag,tmp);
         flag[i] = j;
         if(sub_83A(flag,i))
         {
             tmp[i] = j;
             break;
         }
     }
 }
 printf("actf{%s}\n",tmp);
 return 0;
}
```

```
actf{b0Nf|Re_LiT!}

--------------------------------
Process exited after 0.444 seconds
```

`flag{b0Nf|Re_LiT!}`