

BUUCTF Reverse/[ACTF新生赛2020]SoulLike

原创

这就是强者的世界么  于 2021-08-16 21:07:51 发布  81  收藏

分类专栏: [# BUUCTF Reverse](#)

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/ookami6497/article/details/119739530>

版权



[BUUCTF Reverse 专栏收录该内容](#)

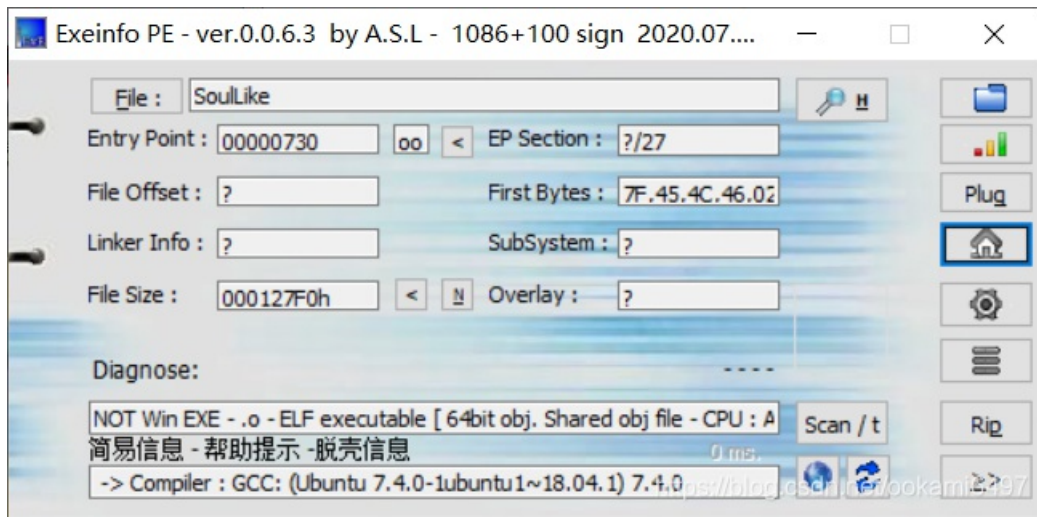
58 篇文章 2 订阅

订阅专栏

BUUCTF Reverse/[ACTF新生赛2020]SoulLike



看下文件信息, 没有加壳



IDA64位打开，还是字符串比较题目

```

__int64 __fastcall main(int a1, char **a2, char **a3)
{
    __int64 result; // rax
    char v5; // [rsp+7h] [rbp-B9h]
    int i; // [rsp+8h] [rbp-B8h]
    int j; // [rsp+Ch] [rbp-B4h]
    int v8[14]; // [rsp+10h] [rbp-B0h] BYREF
    char v9[110]; // [rsp+4Ah] [rbp-76h] BYREF
    unsigned __int64 v10; // [rsp+B8h] [rbp-8h]

    v10 = __readfsqword(0x28u);
    printf("input flag:");
    scanf("%s", &v9[6]);
    strcpy(v9, "actf{");
    v5 = 1;
    for ( i = 0; i <= 4; ++i )
    {
        if ( v9[i] != v9[i + 6] )
        {
            v5 = 0;
            goto LABEL_6;
        }
    }
    if ( !v5 )
        goto LABEL_16;
LABEL_6:
    for ( j = 0; j <= 11; ++j )
        v8[j] = v9[j + 11];
    if ( (unsigned __int8)sub_83A(v8) && v9[23] == 125 )
    {
        printf("That's true! flag is %s", &v9[6]);
        result = 0LL;
    }
    else
    {
LABEL_16:
        printf("Try another time...");
        result = 0LL;
    }
    return result;
}

```

逻辑很简单，是以 `actf{}` 包裹的flag，且括号里面的字符为12位

这里就是输出flag的条件了

```

if ( (unsigned __int8)sub_83A(v8) && v9[23] == 125 )
{
    printf("That's true! flag is %s", &v9[6]);
    result = 0LL;
}

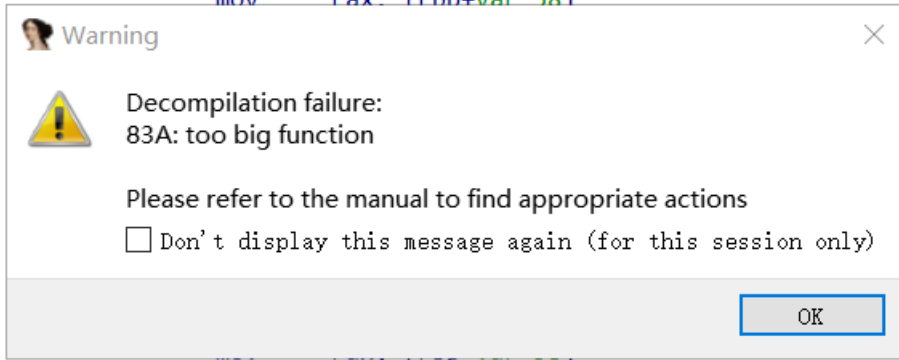
```

`sub_83A(v8)` 对括号里面的12位字符进行变换，但在跟进的时候出了个小问题，说啥太大了

```

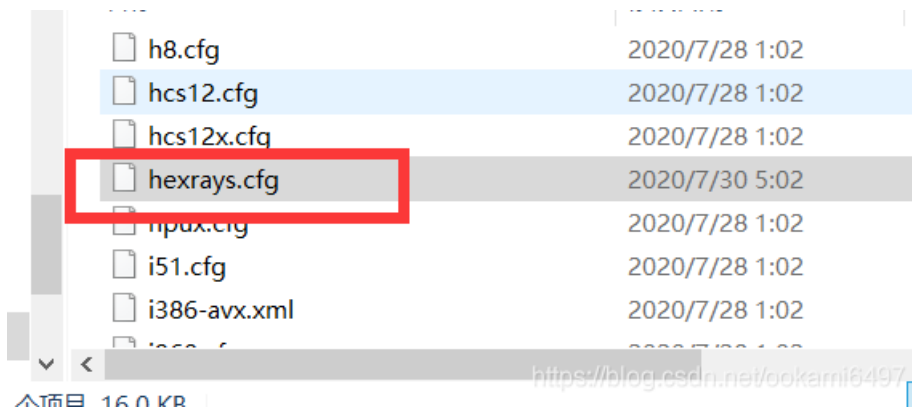
79F4          mov     edx, [rax]
79F6          mov     rax, [rbp+var_58]
79FA
79FE
7A01
7A03
7A07
7A0B
7A0D
7A11
7A15
7A18
7A1A
7A1E          add     rax, 10h
7A22          mov     ecx, [rax]
7A24          mov     rax, [rbp+var_58]

```



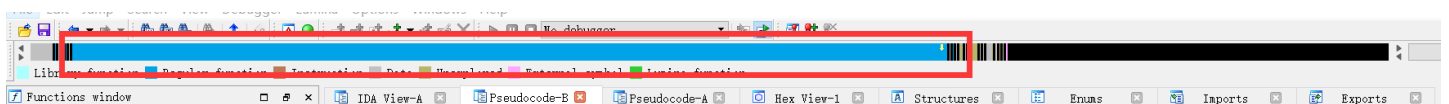
<https://blog.csdn.net/ookami6497>

百度了一下，可以修改配置文件IDA 7.0\cfg\hexrays.cfg来解决



<https://blog.csdn.net/ookami6497>

可编译的最大函数大小为64k，然后改成1024（这个随意，，不过不知道具体要改大多少，这函数确实挺大的，，一整个蓝条都是）



```

// Default constant radix
DEFAULT_RADIX      = 0      // 0 means "decimal for signed, hex for unsigned"
                        // Use 10 for decimal and 16 for hexadecimal

MAX_FUNC_SIZE      = 64     // Functions over 64K are not decompiled

MAX_FUNC_ARGS      = 64     // Max number of function arguments

// Parse format string of called variadic functions in order to detect ellipsis arguments:
#define HPFM_NO      0      // never parse
#define HPFM_STRICT  1      // only if a function is printf/scanf-like

DEFAULT_RADIX      = 0      // 0 means "decimal for signed, hex for unsigned"
                        // Use 10 for decimal and 16 for hexadecimal

MAX_FUNC_SIZE      = 1024   // Functions over 64K are not decompiled

MAX_FUNC_ARGS      = 64     // Max number of function arguments

// Parse format string of called variadic functions in order to detect ellipsis arguments?

```

修改后就能正常反编译了，，，函数是在是太太太长了，完整的就不贴了，贴个开头和结尾意思意思。

```

1  int64 __fastcall sub_83A(_DWORD *a1)
2  {
3  int i; // [rsp+1Ch] [rbp-44h]
4  int v3[14]; // [rsp+20h] [rbp-40h]
5  unsigned __int64 v4; // [rsp+58h] [rbp-8h]
6
7  v4 = __readfsqword(0x28u);
8  *a1 ^= 0x2Bu;
9  a1[1] ^= 0x6Cu;
10 a1[2] ^= 0x7Eu;
11 a1[3] ^= 0x56u;
12 a1[4] ^= 0x39u;
13 a1[5] ^= 3u;
14 a1[6] ^= 0x2Du;
15 a1[7] ^= 0x28u;
16 a1[8] ^= 8u;
17 ++a1[9];
18 a1[10] ^= 0x2Fu;
19 a1[11] ^= 0xAu;
20 ++*a1;
21 a1[1] ^= 0xDu;
22 a1[2] ^= 0x73u;
23 a1[3] ^= a1[2];
24 a1[4] ^= 0x37u;
25 ++a1[5];
26 a1[6] ^= 0x69u;
27 a1[7] ^= 0x59u;
28 a1[8] ^= 0xCu;
29 a1[9] ^= 0x70u;
30 ++a1[10];
31 a1[11] ^= 0x1Fu;
32 ++*a1;

```

<https://blog.csdn.net/ookami6497>

结尾这里，变换后的字符串要等于v3，不然输出wrong

```
006 a1[10] ^= 0x29u;
007 a1[11] ^= 0x3Bu;
008 v3[0] = 126;
009 v3[1] = 50;
010 v3[2] = 37;
011 v3[3] = 88;
012 v3[4] = 89;
013 v3[5] = 107;
014 v3[6] = 53;
015 v3[7] = 110;
016 v3[8] = 0;
017 v3[9] = 19;
018 v3[10] = 30;
019 v3[11] = 56;
020 for ( i = 0; i <= 11; ++i )
021 {
022     if ( v3[i] != a1[i] )
023     {
024         printf("wrong on #%d\n", (unsigned int)i);
025         return 0LL;
026     }
027 }
028 return 1LL;
029 }
```

00010FA2 sub 83A:3007 (10FA2)

<https://blog.csdn.net/ookami6497>

显而易见只能爆破了，太长了，只贴主要的

```

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int v[]={126,50,37,88,89,107,53,110,0,19,30,56};
int f(char *a1,int i)
{
    *a1 ^= 0x2Bu;
    a1[1] ^= 0x6Cu;
    a1[2] ^= 0x7Eu;
    a1[3] ^= 0x56u;
    ...
    ...
    ...
    a1[9] ^= 0x70u;
    a1[10] ^= 0x29u;
    a1[11] ^= 0x3Bu;
    if(a1[i] == v[i])
        return 1;
    else
        return 0;
}
int main()
{
    int i,j,k;
    char flag[13] = "";
    char temp[13] = "";
    for(i = 0; i < 12 ; i++)
    {
        for(k = 33; k <= 126 ; k++)
        {
            strcpy(flag,temp);
            flag[i] = k;
            if(f(flag,i))
            {
                temp[i] = k;
                break;
            }
        }
    }
    printf("flag{%s}\n",temp);
    return 0;
}

```

运行结果

```

C:\Users\86183\Desktop\oj\1EXAMPLE\bin\Release\1EXAMPLE.exe
flag{b0Nf|Re_LiT!}
Process returned 0 (0x0)   execution time : 0.622 s
Press any key to continue.

```

<https://blog.csdn.net/ookami6497>

flag{b0Nf|Re_LIT!}