攻防世界 Reverse高手进阶区 2分题 reverse-for-the-holy-grail-350

原创

思源湖的鱼 于 2020-12-28 16:45:28 发布 136 收藏

分类专栏： ctf 文章标签： ctf 攻防世界 reverse

本文链接：https://blog.csdn.net/weixin_44604541/article/details/111868596

版权

ctf 专栏收录该内容

200 篇文章 23 订阅

订阅专栏

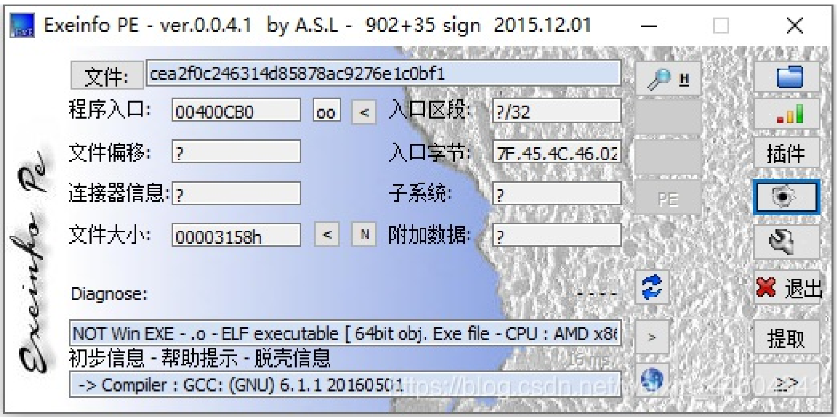
前言

继续ctf的旅程
攻防世界Reverse高手进阶区的2分题
本篇是reverse-for-the-holy-grail-350的writeup

发现攻防世界的题目分数是动态的
就仅以做题时的分数为准了

解题过程

PE查壳



Exeinfo PE - ver.0.0.4.1 by A.S.L - 902+35 sign 2015.12.01

文件: cea2f0c246314d85878ac9276e1c0bf1

程序入口: 00400CB0

入口区段: ?/32

文件偏移: ?

入口字节: 7F.45.4C.46.02

连接器信息: ?

子系统: ?

文件大小: 00003158h

附加数据: ?

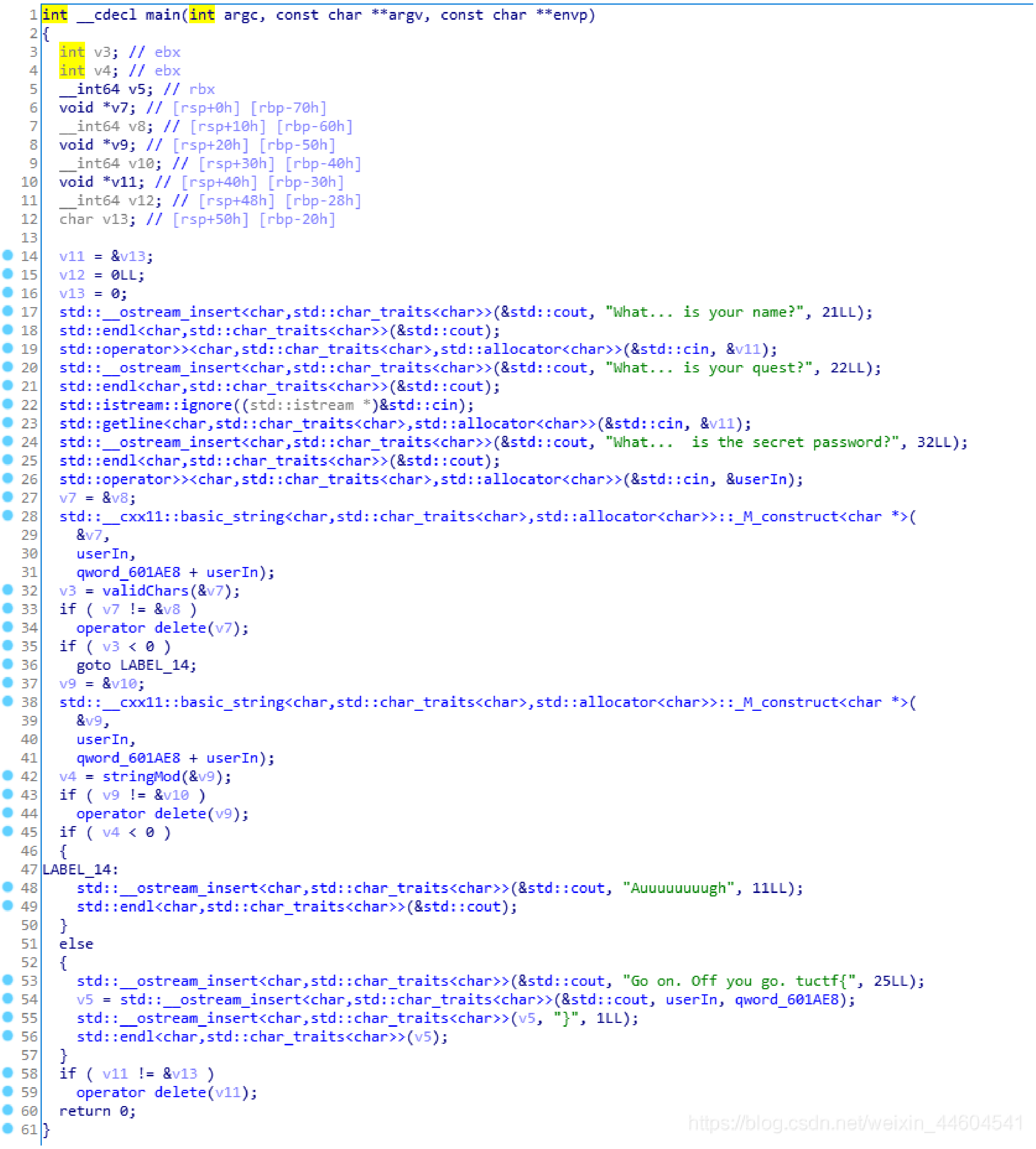
Diagnose:

NOT Win EXE - .o - ELF executable [64bit obj. Exe file - CPU : AMD x86

初步信息 - 帮助提示 - 脱壳信息

-> Compiler : GCC: (GNU) 6.1.1 20160501

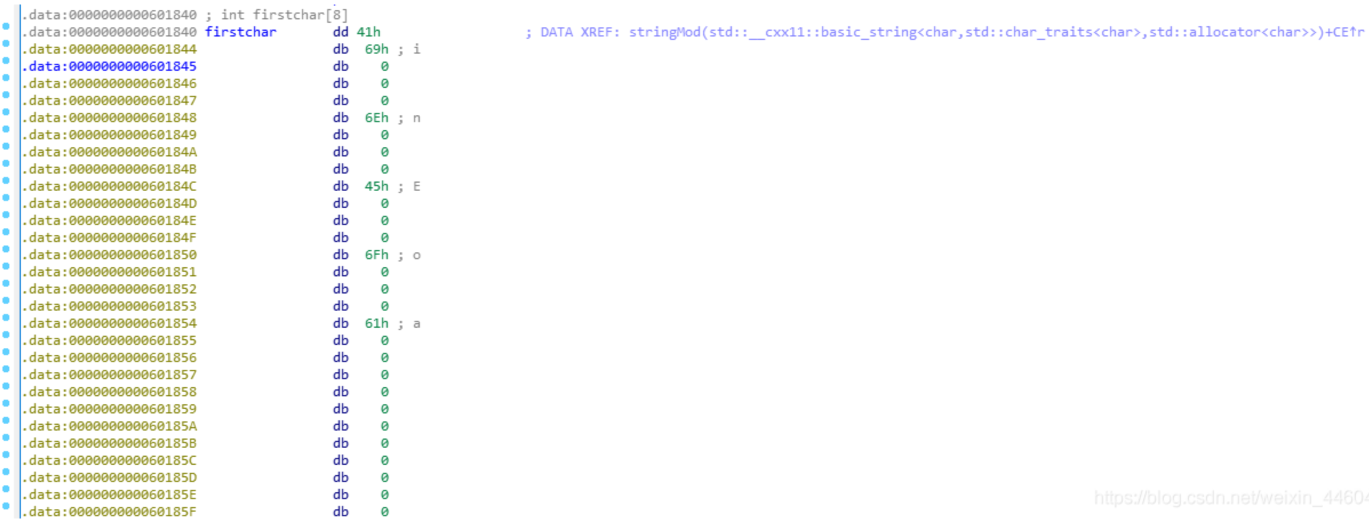
扔进IDA



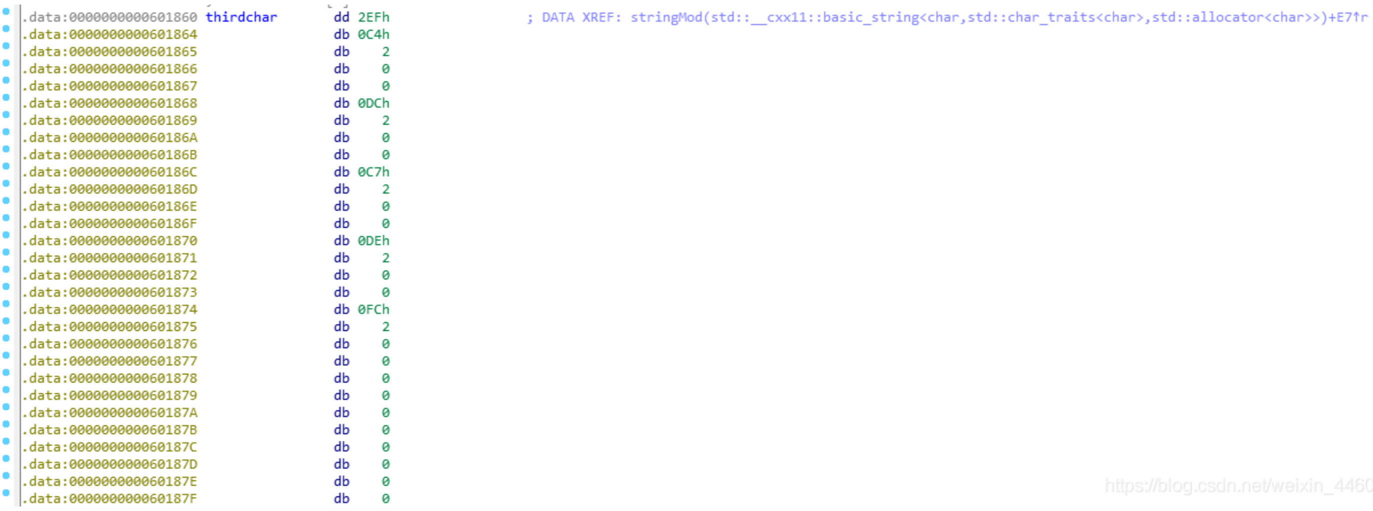
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // ebx
  int v4; // ebx
  __int64 v5; // rbx
  void *v7; // [rsp+0h] [rbp-70h]
  __int64 v8; // [rsp+10h] [rbp-60h]
  void *v9; // [rsp+20h] [rbp-50h]
  __int64 v10; // [rsp+30h] [rbp-40h]
  void *v11; // [rsp+40h] [rbp-30h]
  __int64 v12; // [rsp+48h] [rbp-28h]
  char v13; // [rsp+50h] [rbp-20h]

  v11 = &v13;
  v12 = 0LL;
  v13 = 0;
  std::__ostream_insert<char,std::char_traits<char>>(&std::cout, "What... is your name?", 21LL);
  std::endl<char,std::char_traits<char>>(&std::cout);
  std::operator>><char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v11);
  std::__ostream_insert<char,std::char_traits<char>>(&std::cout, "What... is your quest?", 22LL);
  std::endl<char,std::char_traits<char>>(&std::cout);
  std::istream::ignore((std::istream *)&std::cin);
  std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v11);
  std::__ostream_insert<char,std::char_traits<char>>(&std::cout, "What...  is the secret password?", 32LL);
  std::endl<char,std::char_traits<char>>(&std::cout);
  std::operator>><char,std::char_traits<char>,std::allocator<char>>(&std::cin, &userIn);
  v7 = &v8;
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::_M_construct<char *>(
    &v7,
    userIn,
    qword_601AE8 + userIn);
  v3 = validChars(&v7);
  if ( v7 != &v8 )
    operator delete(v7);
  if ( v3 < 0 )
    goto LABEL_14;
  v9 = &v10;
  std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::_M_construct<char *>(
    &v9,
    userIn,
    qword_601AE8 + userIn);
  v4 = stringMod(&v9);
  if ( v9 != &v10 )
    operator delete(v9);
  if ( v4 < 0 )
  {
LABEL_14:
    std::__ostream_insert<char,std::char_traits<char>>(&std::cout, "Auuuuuuuugh", 11LL);
    std::endl<char,std::char_traits<char>>(&std::cout);
  }
  else
  {
    std::__ostream_insert<char,std::char_traits<char>>(&std::cout, "Go on. Off you go. tuctf{", 25LL);
    v5 = std::__ostream_insert<char,std::char_traits<char>>(&std::cout, userIn, qword_601AE8);
    std::__ostream_insert<char,std::char_traits<char>>(v5, "}", 1LL);
    std::endl<char,std::char_traits<char>>(v5);
  }
  if ( v11 != &v13 )
    operator delete(v11);
  return 0;
}
```

关键函数 stringMod

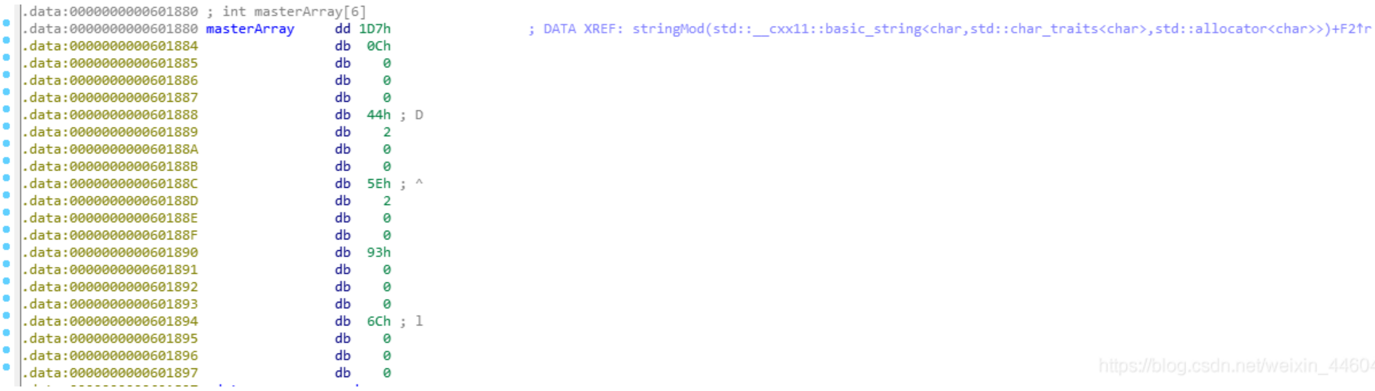
```
__int64 __fastcall stringMod(__int64 *a1)
{
  __int64 v1; // r9
  __int64 v2; // r10
  __int64 v3; // rcx
  signed int v4; // er8
  int *v5; // rdi
  int *v6; // rsi
  signed int v7; // ecx
  signed int v8; // er9
  int v9; // er10
  unsigned int v10; // eax
  int v11; // esi
  int v12; // esi
  int v14[24]; // [rsp+0h] [rbp-60h]
  int _48[24]; // [rsp+48h] [rbp-18h]

  memset(v14, 0, 0x48uLL);
  v1 = a1[1];
  if ( v1 )
  {
    v2 = *a1;
    v3 = 0LL;
    v4 = 0;
    do
    {
      v12 = *(char *)(v2 + v3);
      v14[v3] = v12;
      if ( 3 * ((unsigned int)v3 / 3) == (_DWORD)v3 && v12 != firstchar[(unsigned int)v3 / 3] ) //3的倍数对应firstchar
        v4 = -1;
      ++v3;
    }
    while ( v3 != v1 );
  }
  else
  {
    v4 = 0;
  }
  v5 = v14;
  v6 = v14;
  v7 = 666;
  do
  {
    *v6 = v7 ^ *(unsigned __int8 *)v6; //异或
    v7 += v7 % 5;
    ++v6;
  }
  while ( _48 != v6 ); //18次
  v8 = 1;
  v9 = 0;
  v10 = 1;
  v11 = 0;
  do
  {
    if ( v11 == 2 )
    {
      if ( *v5 != thirdchar[v9] )
        v4 = -1;
      if ( v10 % *v5 != masterArray[v9] )
        v4 = -1;
      ++v9;
      v10 = 1;
      v11 = 0;
    }
    else
    {
      v10 *= *v5;
      if ( ++v11 == 3 )
        v11 = 0;
    }
    ++v8;
    ++v5;
    ++v5;
  }
  while ( v8 != 19 );
  return (unsigned int)(v7 * v4);
}
```



```
.data:0000000000601840 ; int firstchar[8]
.data:0000000000601840 firstchar       dd 41h                  ; DATA XREF: stringMod(std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>)+CE↑r
.data:0000000000601844                 db  69h ; i
.data:0000000000601845                 db    0
.data:0000000000601846                 db    0
.data:0000000000601847                 db    0
.data:0000000000601848                 db  6Eh ; n
.data:0000000000601849                 db    0
.data:000000000060184A                 db    0
.data:000000000060184B                 db    0
.data:000000000060184C                 db  45h ; E
.data:000000000060184D                 db    0
.data:000000000060184E                 db    0
.data:000000000060184F                 db    0
.data:0000000000601850                 db  6Fh ; o
.data:0000000000601851                 db    0
.data:0000000000601852                 db    0
.data:0000000000601853                 db    0
.data:0000000000601854                 db  61h ; a
.data:0000000000601855                 db    0
.data:0000000000601856                 db    0
.data:0000000000601857                 db    0
.data:0000000000601858                 db    0
.data:0000000000601859                 db    0
.data:000000000060185A                 db    0
.data:000000000060185B                 db    0
.data:000000000060185C                 db    0
.data:000000000060185D                 db    0
.data:000000000060185E                 db    0
.data:000000000060185F                 db    0
```



```
.data:0000000000601860 thirdchar       dd 2EFh                 ; DATA XREF: stringMod(std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>)+E7↑r
.data:0000000000601864                 db 0C4h
.data:0000000000601865                 db    2
.data:0000000000601866                 db    0
.data:0000000000601867                 db    0
.data:0000000000601868                 db 0DCh
.data:0000000000601869                 db    2
.data:000000000060186A                 db    0
.data:000000000060186B                 db    0
.data:000000000060186C                 db 0C7h
.data:000000000060186D                 db    2
.data:000000000060186E                 db    0
.data:000000000060186F                 db    0
.data:0000000000601870                 db 0DEh
.data:0000000000601871                 db    2
.data:0000000000601872                 db    0
.data:0000000000601873                 db    0
.data:0000000000601874                 db 0FCh
.data:0000000000601875                 db    2
.data:0000000000601876                 db    0
.data:0000000000601877                 db    0
.data:0000000000601878                 db    0
.data:0000000000601879                 db    0
.data:000000000060187A                 db    0
.data:000000000060187B                 db    0
.data:000000000060187C                 db    0
.data:000000000060187D                 db    0
.data:000000000060187E                 db    0
.data:000000000060187F                 db    0
```



```
.data:0000000000601880 ; int masterArray[6]
.data:0000000000601880 masterArray     dd 1D7h                 ; DATA XREF: stringMod(std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>)+F2↑r
.data:0000000000601884                 db  0Ch
.data:0000000000601885                 db    0
.data:0000000000601886                 db    0
.data:0000000000601887                 db    0
.data:0000000000601888                 db  44h ; D
.data:0000000000601889                 db    2
.data:000000000060188A                 db    0
.data:000000000060188B                 db    0
.data:000000000060188C                 db  5Eh ; ^
.data:000000000060188D                 db    2
.data:000000000060188E                 db    0
.data:000000000060188F                 db    0
.data:0000000000601890                 db  93h
.data:0000000000601891                 db    0
.data:0000000000601892                 db    0
.data:0000000000601893                 db    0
.data:0000000000601894                 db  6Ch ; l
.data:0000000000601895                 db    0
.data:0000000000601896                 db    0
.data:0000000000601897                 db    0
```

```
firstchar =   [0x41,  0x69,  0x6e,  0x45,  0x6f,  0x61]
thirdchar =   [0x2ef, 0x2c4, 0x2dc, 0x2c7, 0x2de, 0x2fc]
masterarray = [0x1d7, 0xc,   0x244, 0x25e, 0x93,  0x6c]

xor_number=0x29a
xor_array=[]
for i in range(18):
 xor_array.append(xor_number)
 xor_number += xor_number % 5

flag=""
for i in range(6):
 one=firstchar[i]
 three=thirdchar[i] ^ xor_array[(i*3) + 2]
 for j in range(256):
  if masterarray[i]==(j^xor_array[i*3+1])*(one^xor_array[i*3])%thirdchar[i]:
   flag+=chr(one)+chr(j)+chr(three)
   break
print("tuctf{" + flag + "}")
```

得到flag：`tuctf{AfricanOrEuropean?}`

结语

简单题