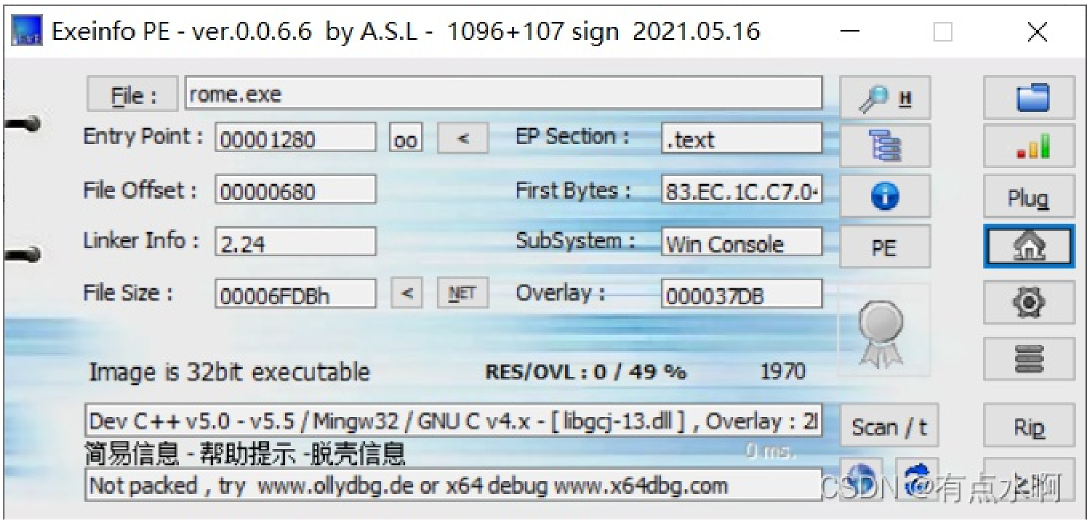
buu-[ACTF新生赛2020]rome

原创

有点水啊 于 2022-02-08 23:19:36 发布 1217 收藏
分类专栏： buuctf-reserve 文章标签： c语言 安全

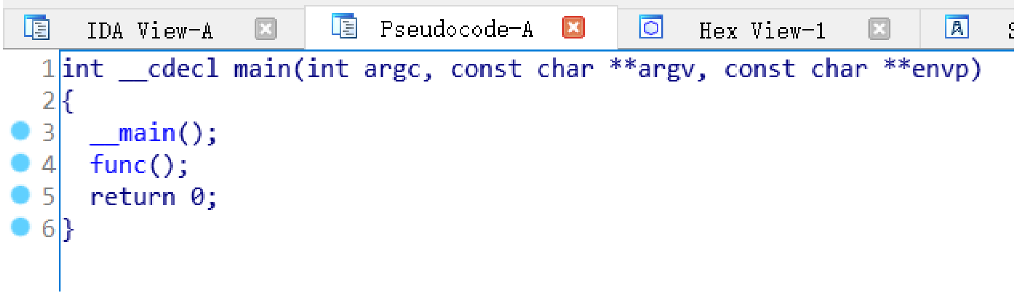
本文链接：https://blog.csdn.net/qaq517384/article/details/122830382
版权
buuctf-reserve 专栏收录该内容
59 篇文章 0 订阅
订阅专栏

无壳32位文件


Exeinfo PE - ver.0.0.6.6 by A.S.L - 1096+107 sign 2021.05.16

File : rome.exe
Entry Point : 00001280
EP Section : .text
File Offset : 00000680
First Bytes : 83.EC.1C.C7.0
Linker Info : 2.24
SubSystem : Win Console
File Size : 00006FDBh
Overlay : 000037DB
Image is 32bit executable RES/OVL : 0 / 49 % 1970
Dev C++ v5.0 - v5.5 / Mingw32 / GNU C v4.x - [libgcj-13.dll] , Overlay : 2
简易信息 - 帮助提示 -脱壳信息
Not packed , try www.ollydbg.de or x64 debug www.x64dbg.com

32位ida打开查看main函数



```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  __main();
  func();
  return 0;
}
```

_main();太乱了不想看
先看func()

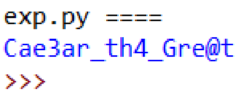
```
int func()
{
  int result; // eax
  int v1; // [esp+14h] [ebp-44h]
  int v2; // [esp+18h] [ebp-40h]
  int v3; // [esp+1Ch] [ebp-3Ch]
  int v4; // [esp+20h] [ebp-38h]
  unsigned __int8 v5; // [esp+24h] [ebp-34h]
unsigned __int8 v5; // [esp+24h] [ebp-34h]
unsigned __int8 v6; // [esp+25h] [ebp-33h]
unsigned __int8 v7; // [esp+26h] [ebp-32h]
unsigned __int8 v8; // [esp+27h] [ebp-31h]
unsigned __int8 v9; // [esp+28h] [ebp-30h]
int v10; // [esp+29h] [ebp-2Fh]
int v11; // [esp+2Dh] [ebp-2Bh]
int v12; // [esp+31h] [ebp-27h]
int v13; // [esp+35h] [ebp-23h]
unsigned __int8 v14; // [esp+39h] [ebp-1Fh]
char v15; // [esp+3Bh] [ebp-1Dh]
char v16; // [esp+3Ch] [ebp-1Ch]
char v17; // [esp+3Dh] [ebp-1Bh]
char v18; // [esp+3Eh] [ebp-1Ah]
char v19; // [esp+3Fh] [ebp-19h]
char v20; // [esp+40h] [ebp-18h]
char v21; // [esp+41h] [ebp-17h]
char v22; // [esp+42h] [ebp-16h]
char v23; // [esp+43h] [ebp-15h]
char v24; // [esp+44h] [ebp-14h]
char v25; // [esp+45h] [ebp-13h]
char v26; // [esp+46h] [ebp-12h]
char v27; // [esp+47h] [ebp-11h]
char v28; // [esp+48h] [ebp-10h]
char v29; // [esp+49h] [ebp-Fh]
char v30; // [esp+4Ah] [ebp-Eh]
char v31; // [esp+4Bh] [ebp-Dh]
int i; // [esp+4Ch] [ebp-Ch]

v15 = 81;
v16 = 115;
v17 = 119;
v18 = 51;
v19 = 115;
v20 = 106;
v21 = 95;
v22 = 108;
v23 = 122;
v24 = 52;
v25 = 95;
v26 = 85;
v27 = 106;
v28 = 119;
v29 = 64;
v30 = 108;
v31 = 0;
printf("Please input:");
scanf("%s", &v5);
result = v5;
if ( v5 == 'A' )
{
  result = v6;
  if ( v6 == 'C' )
  {
    result = v7;
    if ( v7 == 'T' )
    {
      result = v8;
      if ( v8 == 'F' )
      {
          result = v9;
          if ( v9 == '{' )
          {
            result = v14;
            if ( v14 == '}' )
            {
              v1 = v10;
              v2 = v11;
              v3 = v12;
              v4 = v13;
              for ( i = 0; i <= 15; ++i )
              {
                if ( *((_BYTE *)&v1 + i) > 64 && *((_BYTE *)&v1 + i) <= 90 )
                  *((_BYTE *)&v1 + i) = (*((char *)&v1 + i) - 51) % 26 + 65;
                if ( *((_BYTE *)&v1 + i) > 96 && *((_BYTE *)&v1 + i) <= 122 )
                  *((_BYTE *)&v1 + i) = (*((char *)&v1 + i) - 79) % 26 + 97;
              }
              for ( i = 0; i <= 15; ++i )
              {
                result = (unsigned __int8)*(&v15 + i);
                if ( *((_BYTE *)&v1 + i) != (_BYTE)result )
                  return result;
              }
              result = printf("You are correct!");
            }
          }
        }
      }
    }
  }
  return result;
}
```

ACTF{}不用管，直接看for循环
第一个for循环
65-90,97-122,对大小写字母分别加密
第二个for就是单纯的比较

变化后的值等于v15的值就行
能正着跑绝不逆向）

```
v15 = [81,115,119,51,115,106,95,108,122,52,95,85,106,119,64,108]
flag = ''
for i in range(0,16):
     for j in range(0,127):
          f=j
          if j > 64 and j <= 90:
               j = (j-51)%26 + 65
          if j > 96 and j <= 122:
               j = (j-79)%26 + 97
          if(j == v15[i]):
               flag += chr(f)
print(flag)
```



```
exp.py ====
Cae3ar_th4_Gre@t
>>>
```

`flag{Cae3ar_th4_Gre@t}`