[BUUCTF]REVERSE------[ACTF新生赛2020]rome

原创

HAIANAWEI 于 2021-02-04 13:36:05 发布 96 收藏

分类专栏： 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/BangSen1/article/details/113638415

版权

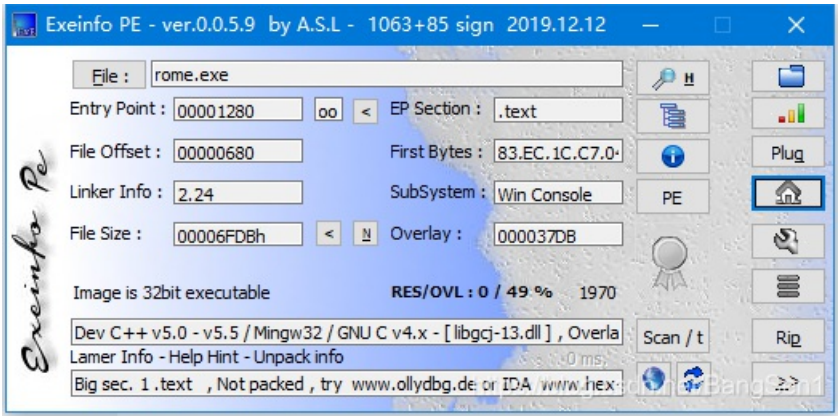
REVERSE 专栏收录该内容

30 篇文章 0 订阅

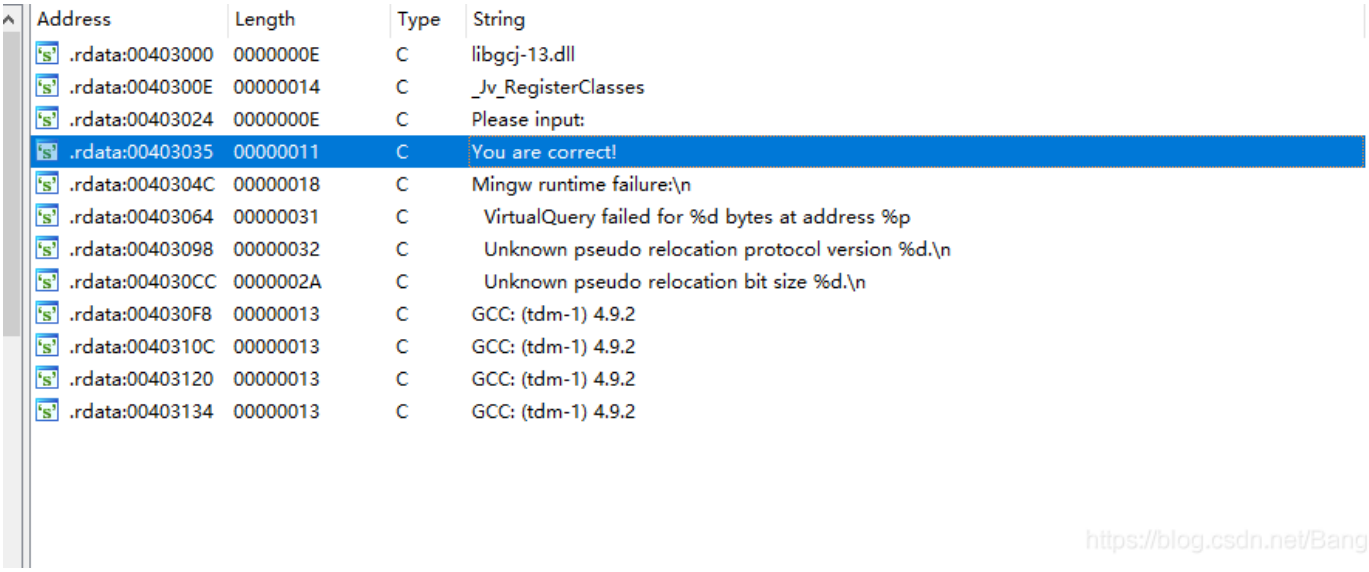
订阅专栏

[ACTF新生赛2020]rome

查壳，32bit，无壳



用32位IDA打开，从 you are correct 进入。



Address	Length	Type	String
.rdata:00403000	0000000E	C	libgcj-13.dll
.rdata:0040300E	00000014	C	_Jv_RegisterClasses
.rdata:00403024	0000000E	C	Please input:
.rdata:00403035	00000011	C	You are correct!
.rdata:0040304C	00000018	C	Mingw runtime failure:\n
.rdata:00403064	00000031	C	VirtualQuery failed for %d bytes at address %p
.rdata:00403098	00000032	C	Unknown pseudo relocation protocol version %d.\n
.rdata:004030CC	0000002A	C	Unknown pseudo relocation bit size %d.\n
.rdata:004030F8	00000013	C	GCC: (tdm-1) 4.9.2
.rdata:0040310C	00000013	C	GCC: (tdm-1) 4.9.2
.rdata:00403120	00000013	C	GCC: (tdm-1) 4.9.2
.rdata:00403134	00000013	C	GCC: (tdm-1) 4.9.2

https://blog.csdn.net/BangSen1

就是让我们输入一个字符串，然后判断大小写，进行相应的运算，最后得到了程序开头的数组

v12 ['Q','s','w','3','s','j', '','l','z','4','','U','j','w','@','l']

第45和47是进行大小写判断

```
int v9; // [esp+31h] [ebp-27h]
int v10; // [esp+35h] [ebp-23h]
unsigned __int8 v11; // [esp+39h] [ebp-1Fh]
char v12[29]; // [esp+3Bh] [ebp-1Dh] BYREF

strcpy(v12, "Qsw3sj_lz4_Ujw@l");
printf("Please input:");
scanf("%s", &v2);
result = v2;
if ( v2 == 'A' )
{
  result = v3;
  if ( v3 == 'C' )
  {
    result = v4;
    if ( v4 == 'T' )
    {
      result = v5;
      if ( v5 == 'F' )
      {
        result = v6;
        if ( v6 == '{' )
        {
          result = v11;
          if ( v11 == '}' )
          {
            v1[0] = v7;
            v1[1] = v8;
            v1[2] = v9;
            v1[3] = v10;
            *(_DWORD *)&v12[17] = 0;
            while ( *(int *)&v12[17] <= 15 )
            {
              if ( *((char *)v1 + *(_DWORD *)&v12[17]) > 64 && *((char *)v1 + *(_DWORD *)&v12[17]) <= 90 )
                *((_BYTE *)v1 + *(_DWORD *)&v12[17]) = (*((char *)v1 + *(_DWORD *)&v12[17]) - 51) % 26 + 65;
              if ( *((char *)v1 + *(_DWORD *)&v12[17]) > 96 && *((char *)v1 + *(_DWORD *)&v12[17]) <= 122 )
                *((_BYTE *)v1 + *(_DWORD *)&v12[17]) = (*((char *)v1 + *(_DWORD *)&v12[17]) - 79) % 26 + 97;
              ++*(_DWORD *)&v12[17];
            }
            *(_DWORD *)&v12[17] = 0;
            while ( *(int *)&v12[17] <= 15 )
            {
              result = (unsigned __int8)v12[*(_DWORD *)&v12[17]];
              if ( *((_BYTE *)v1 + *(_DWORD *)&v12[17]) != (_BYTE)result )
                return result;
              ++*(_DWORD *)&v12[17];
            }
            result = printf("You are correct!");
          }
        }
```

00000862 _func:46 (401462)

https://blog.csdn.net/BangSen1

利用穷举

```
v12= [ 'Q','s','w','3','s','j',  '_','l','z','4','_','U','j','w','@','l' ]
flag=""

for i in range(16):
    for j in range(128):
        x=j
        if chr(x).isupper():
            x=(x-51)%26+65
        if chr(x).islower():
            x=(x-79)%26+97
        if chr(x)==v12[i]:
            flag+=chr(j)

print ('flag{'+flag+'}')
```

https://blog.csdn.net/BangSen1

```
flag{Cae3ar_th4_Gre@t}
>>>
```