BUUCTF Reverse/[ACTF新生赛2020]rome

原创

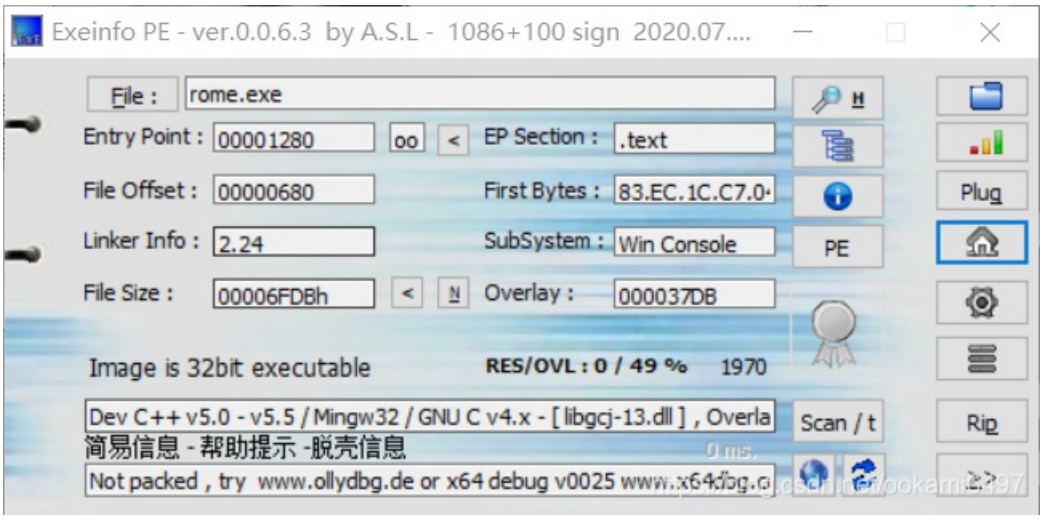
这就是强者的世界么 于 2021-07-20 20:23:04 发布 209 收藏

分类专栏： # BUUCTF Reverse

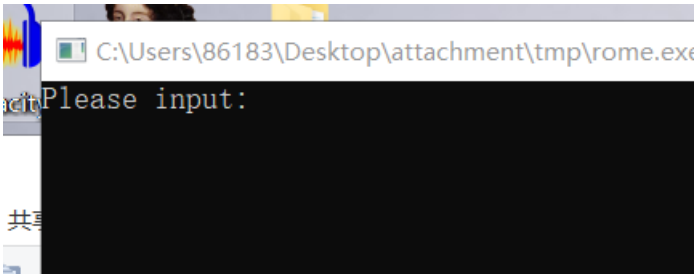
本文链接：https://blog.csdn.net/ookami6497/article/details/118940105

BUUCTF Reverse/[ACTF新生赛2020]rome

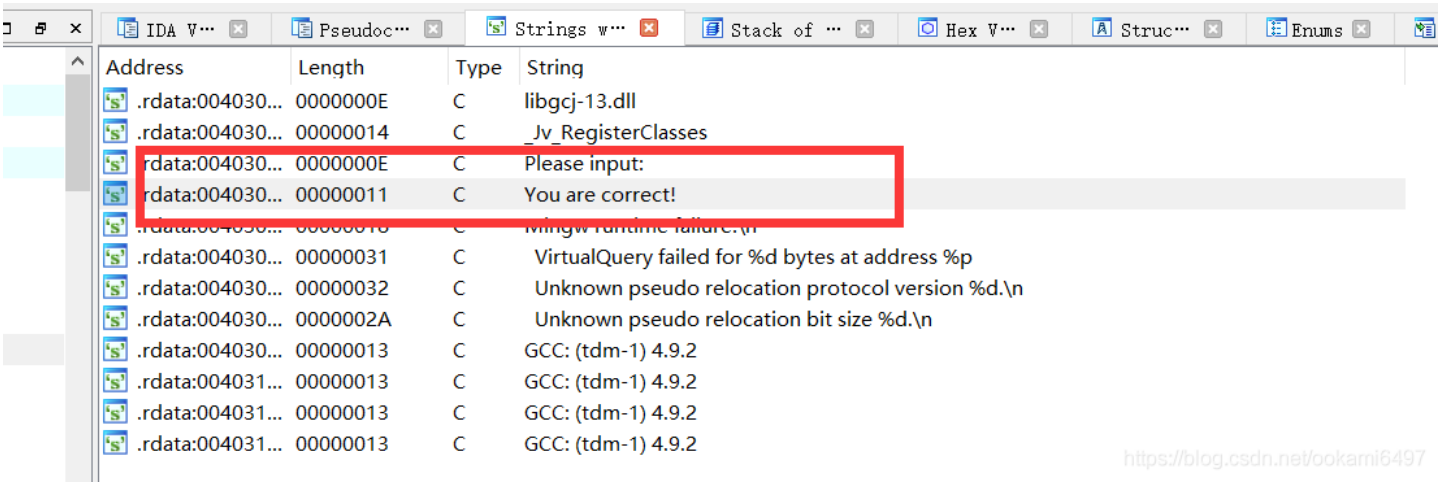
> 先看文件信息，没有加壳，是一个32位程序



打开运行，推测又是字符串比较的题目



拖入IDA32位进行分析，依旧是先找字符串



跟踪跳转，来到func()函数

```
int func()
{
  int result; // eax
  int v1[4]; // [esp+14h] [ebp-44h]
  unsigned __int8 v2; // [esp+24h] [ebp-34h] BYREF
  unsigned __int8 v3; // [esp+25h] [ebp-33h]
  unsigned __int8 v4; // [esp+26h] [ebp-32h]
  unsigned __int8 v5; // [esp+27h] [ebp-31h]
  unsigned __int8 v6; // [esp+28h] [ebp-30h]
  int v7; // [esp+29h] [ebp-2Fh]
  int v8; // [esp+2Dh] [ebp-2Bh]
```

```
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 == 65 )
{
  result = v3;
  if ( v3 == 67 )
  {
    result = v4;
    if ( v4 == 84 )
    {
      result = v5;
      if ( v5 == 70 )
      {
        result = v6;
        if ( v6 == 123 )
        {
          result = v11;
          if ( v11 == 125 )
          {
            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!");
          }
        }
      }
    }
  }
}
return result;
}
```

> 先看这个，输入的字符串经过变换后与v12进行比较，相等则输出 You are correct!

```
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!");
```

> v12的值已经知道

```
strcpy(v12, "Qsw3sj_lz4_Ujw@l");
```

> 这个就是对输入的字符串作变换的算法，将字符中的字母分为，大写和小写分别进行变换

```
*(_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];
            }
```

> 写出脚本

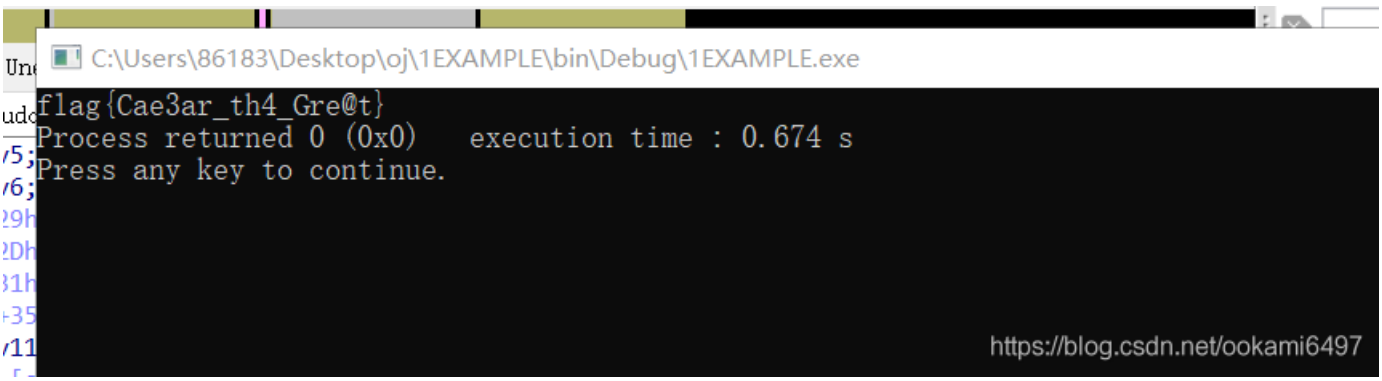
```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main()
{
    char v12[]  = "Qsw3sj_lz4_Ujw@l";  //长度为16
    int flag[17] = {0};
    int i;
    for(i = 0 ; i <= 15; i++)
    {
        if(v12[i] <= 90 && v12[i] > 64)  //大写
        {
            flag[i] = v12[i] - 65 + 51;
            while(flag[i] < 65)
            {
                flag[i] += 26;
            }
        }
        else if(v12[i] <= 122 && v12[i] > 96) //小写
        {
            flag[i] = v12[i] - 97 + 79;
            while(flag[i] < 97)
            {
                flag[i] += 26;
            }

        }
        else
        {
            flag[i] = v12[i];
        }
    }
    printf("flag{");
   for(i = 0 ; i < 16 ; i ++)
   {
       printf("%c",flag[i],flag[i]);
   }
  printf("}");
   return 0;

}
```

运行得到结果



flag{Cae3ar_th4_Gre@t}