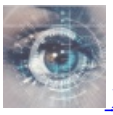
XCTF re-100

原创

YenKoc 于 2020-03-22 23:28:49 发布 94 收藏

分类专栏： XCTF

本文链接：https://blog.csdn.net/YenKoc/article/details/105038799



一.无壳并拉入ida64静态调试（注释说的很明白了）

```
close(pParentRead[0]);
while ( 1 )
{
  memset(bufParentRead, 0, 0xC8uLL);
  numRead = read(pParentWrite[0], bufParentRead, 0xC8uLL);
  if ( numRead == -1 )
    break;
  if ( numRead )
  {
    if ( childCheckDebugResult() )
    {
      responseFalse();
    }
    else if ( bufParentRead[0] == '{' )
    {
      if ( strlen(bufParentRead) == 42 )
      {
        if ( !strncmp(&bufParentRead[1], "53fc275d81", 0xAuLL) )// 从数组的第2个开始比较，10个字节
        {
          if ( bufParentRead[strlen(bufParentRead) - 1] == '}' )
          {
            if ( !strncmp(&bufParentRead[31], "4938ae4efd", 0xAuLL) )// 相当于从数组32个开始比较，后面限定了字节数
            {                                                       // 将flag开头和结尾都限定了
              if ( !confuseKey(bufParentRead, 42) )
              {
                responseFalse();
              }
              else if ( !strncmp(bufParentRead, "{daf29f59034938ae4efd53fc275d81053ed5be8c}", 0x2AuLL) )
              {
                responseTrue();
              }
              else
              {
                responseFalse();
              }
            }
            else
            {
              responseFalse();
            }
          }
          else
```


二.confuseKey是个关键函数，进入看看

```
IDA View-A | Pseudocode-A | Stack of main | Hex View-1
1 bool __cdecl confuseKey(char *szKey, int iKeyLength)
2 {
3   char szPart1[15]; // [rsp+10h] [rbp-50h]
4   char szPart2[15]; // [rsp+20h] [rbp-40h]
5   char szPart3[15]; // [rsp+30h] [rbp-30h]
6   char szPart4[15]; // [rsp+40h] [rbp-20h]
7   unsigned __int64 v7; // [rsp+58h] [rbp-8h]
8
9   v7 = __readfsqword(0x28u);
10  *(_QWORD *)szPart1 = 0LL;
11  *(_DWORD *)&szPart1[8] = 0;
```

```
12  *(_WORD *)&szPart1[12] = 0;
13  szPart1[14] = 0;
14  *(_QWORD *)szPart2 = 0LL;
15  *(_DWORD *)&szPart2[8] = 0;
16  *(_WORD *)&szPart2[12] = 0;
17  szPart2[14] = 0;
18  *(_QWORD *)szPart3 = 0LL;
19  *(_DWORD *)&szPart3[8] = 0;
20  *(_WORD *)&szPart3[12] = 0;
21  szPart3[14] = 0;
22  *(_QWORD *)szPart4 = 0LL;
23  *(_DWORD *)&szPart4[8] = 0;
24  *(_WORD *)&szPart4[12] = 0;
25  szPart4[14] = 0;
26  if ( iKeyLength != 42 )
27    return 0;
28  if ( !szKey )
29    return 0;
30  if ( strlen(szKey) != 42 )
31    return 0;
32  if ( *szKey != '{' )
33    return 0;
34  strncpy(szPart1, szKey + 1, 0xAuLL);
35  strncpy(szPart2, szKey + 11, 0xAuLL);
36  strncpy(szPart3, szKey + 21, 0xAuLL);
37  strncpy(szPart4, szKey + 31, 0xAuLL);
38  memset(szKey, 0, iKeyLength);
39  *szKey = 123;
40  strcat(szKey, szPart3);                       // 将顺序变换了
41  strcat(szKey, szPart4);
42  strcat(szKey, szPart1);
43  strcat(szKey, szPart2);
44  szKey[41] = 125;
45  return 1;
46}
```


发现就是将我们所输入的字符串分割，并把顺序调换了，调回来就是我们的flag。

三.flag:

```
{53fc275d81053ed5be8cdaf29f59034938ae4efd}
```

提交的时候，把花括号去掉，太坑了，题目没提示的。。。。