攻防世界 Reverse高手进阶区 2分题 re1-100

原创

思源湖的鱼 于 2020-12-11 13:12:07 发布 116 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/111033272

CTF

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re1-100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

扔进IDA

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  __pid_t v3; // eax
  size_t v4; // rax
```

```
size_t v4; // rax
ssize_t v5; // rbx
bool v6; // al
char **argva; // [rsp+0h] [rbp-1D0h]
bool bCheckPtrace; // [rsp+13h] [rbp-1BDh]
ssize_t numRead; // [rsp+18h] [rbp-1B8h]
ssize_t numReada; // [rsp+18h] [rbp-1B8h]
char bufWrite[200]; // [rsp+20h] [rbp-1B0h]
char bufParentRead[200]; // [rsp+F0h] [rbp-E0h]
unsigned __int64 v13; // [rsp+1B8h] [rbp-18h]

argva = (char **)argv;
v13 = __readfsqword(0x28u);
bCheckPtrace = detectDebugging();
if ( pipe(pParentWrite) == -1 )
  exit(1);
if ( pipe(pParentRead) == -1 )
  exit(1);
v3 = fork();
if ( v3 != -1 )
{
  if ( v3 )
  {
    close(pParentWrite[0]);
    close(pParentRead[1]);
    while ( 1 )
    {
      printf("Input key : ", argva);
      memset(bufWrite, 0, 0xC8uLL);
      gets(bufWrite, 0LL);
      v4 = strlen(bufWrite);
      v5 = write(pParentWrite[1], bufWrite, v4);
      if ( v5 != strlen(bufWrite) )
        printf("parent - partial/failed write", bufWrite);
      do
      {
        memset(bufParentRead, 0, 0xC8uLL);
        numReada = read(pParentRead[0], bufParentRead, 0xC8uLL);
        v6 = bCheckPtrace || checkDebuggerProcessRunning();
        if ( v6 )
        {
          puts("Wrong !!!\n");
        }
        else if ( !checkStringIsNumber(bufParentRead) )
        {
          puts("Wrong !!!\n");
        }
        else
        {
          if ( atoi(bufParentRead) )
          {
            puts("True");
            if ( close(pParentWrite[1]) == -1 )
              exit(1);
            exit(0);
          }
          puts("Wrong !!!\n");
        }
      }
      while ( numReada == -1 );
```

```
    }
  }
  close(pParentWrite[1]);
  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] == 123 )
      {
        if ( strlen(bufParentRead) == 42 )
        {
          if ( !strncmp(&bufParentRead[1], "53fc275d81", 0xAuLL) )
          {
            if ( bufParentRead[strlen(bufParentRead) - 1] == 125 )
            {
              if ( !strncmp(&bufParentRead[31], "4938ae4efd", 0xAuLL) )
              {
                if ( !confuseKey(bufParentRead, 42) )
                {
                  responseFalse();
                }
                else if ( !strncmp(bufParentRead, "{daf29f59034938ae4efd53fc275d81053ed5be8c}", 0x2AuLL) )
                {
                  responseTrue();
                }
                else
                {
                  responseFalse();
                }
              }
              else
              {
                responseFalse();
              }
            }
            else
            {
              responseFalse();
            }
          }
          else
          {
            responseFalse();
          }
        }
        else
        {
          responseFalse();
        }
      }
      else
```

```
        else
        {
          responseFalse();
        }
      }
    }
    exit(1);
  }
  exit(1);
}
```

先看到return true这里一堆判断

- 第一个字符为`{`
- 总长42
- 前10位为`53fc275d81`
- 最后一个字符为`}`
- 最后的字符为`4938ae4efd`
- `confuseKey`函数判断
- 然后与`{daf29f59034938ae4efd53fc275d81053ed5be8c}`比较

跟踪`confuseKey`函数

```
bool __cdecl confuseKey(char *szKey, int iKeyLength)
{
  char szPart1[15]; // [rsp+10h] [rbp-50h]
  char szPart2[15]; // [rsp+20h] [rbp-40h]
  char szPart3[15]; // [rsp+30h] [rbp-30h]
  char szPart4[15]; // [rsp+40h] [rbp-20h]
  unsigned __int64 v7; // [rsp+58h] [rbp-8h]

  v7 = __readfsqword(0x28u);
  *(_QWORD *)szPart1 = 0LL;
  *(_DWORD *)&szPart1[8] = 0;
  *(_WORD *)&szPart1[12] = 0;
  szPart1[14] = 0;
  *(_QWORD *)szPart2 = 0LL;
  *(_DWORD *)&szPart2[8] = 0;
  *(_WORD *)&szPart2[12] = 0;
  szPart2[14] = 0;
  *(_QWORD *)szPart3 = 0LL;
  *(_DWORD *)&szPart3[8] = 0;
  *(_WORD *)&szPart3[12] = 0;
  szPart3[14] = 0;
  *(_QWORD *)szPart4 = 0LL;
  *(_DWORD *)&szPart4[8] = 0;
  *(_WORD *)&szPart4[12] = 0;
  szPart4[14] = 0;
  if ( iKeyLength != 42 )
    return 0;
  if ( !szKey )
    return 0;
  if ( strlen(szKey) != 42 )
    return 0;
  if ( *szKey != 123 )
    return 0;
  strncpy(szPart1, szKey + 1, 0xAuLL);
  strncpy(szPart2, szKey + 11, 0xAuLL);
  strncpy(szPart3, szKey + 21, 0xAuLL);
  strncpy(szPart4, szKey + 31, 0xAuLL);
  memset(szKey, 0, iKeyLength);
  *szKey = 123;
  strcat(szKey, szPart3);
  strcat(szKey, szPart4);
  strcat(szKey, szPart1);
  strcat(szKey, szPart2);
  szKey[41] = 125;
  return 1;
}
```


发现就是前半和后半互换了

那就换回来得到`53fc275d81053ed5be8cdaf29f59034938ae4efd`

提交时一开始加了`{}`说错误
后来发现不用括号

结语

简单题