攻防世界 Reverse高手进阶区 2分题 notsequence

原创

思源湖的鱼 于 2021-01-04 15:15:26 发布 120 收藏
分类专栏： ctf 文章标签： ctf 攻防世界 reverse 杨辉三角

本文链接：https://blog.csdn.net/weixin_44604541/article/details/112176443

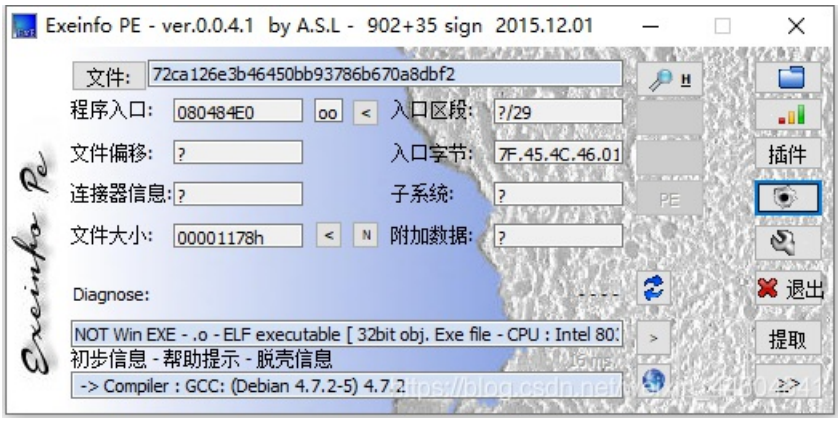
前言

继续ctf的旅程
攻防世界Reverse高手进阶区的2分题
本篇是notsequence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 main()
{
  _DWORD *v0; // eax
  int v2; // [esp+14h] [ebp-Ch]
  _DWORD *v3; // [esp+1Ch] [ebp-4h]

  memset(&unk_8049BE0, 0, 0x4000u);
  puts("input raw_flag please:");
  v3 = &unk_8049BE0;
  do
  {
    v0 = v3;
    ++v3;
    scanf("%d", v0);
  }
  while ( *(v3 - 1) != 0 );
  v2 = sub_80486CD(&unk_8049BE0);
  if ( v2 == -1 )
  {
    printf("check1 not pass");
    system("pause");
  }
  if ( (unsigned __int8)sub_8048783(&unk_8049BE0, v2) ^ 1 )
  {
    printf("check2 not pass!");
    exit(0);
  }
  if ( v2 == 20 )
  {
    puts("Congratulations! fl4g is :\nRCTF{md5(/*what you input without space or \\n~*/)}");
    exit(0);
  }
  return 0;
}
```


计算v2
然后三个check

v2如下

```
signed int __cdecl sub_80486CD(int a1)
{
  int j; // [esp+0h] [ebp-14h]
  int v3; // [esp+4h] [ebp-10h]
  int i; // [esp+8h] [ebp-Ch]
  int v5; // [esp+Ch] [ebp-8h]

  v5 = 0;
  for ( i = 0; i <= 1024 && *(_DWORD *)(4 * i + a1); i = v5 * (v5 + 1) / 2 )
  {
    v3 = 0;
    for ( j = 0; j <= v5; ++j )
      v3 += *(_DWORD *)(4 * (j + i) + a1);
    if ( 1 << v5 != v3 )
      return -1;
    ++v5;
  }
  return v5;
}
```


- i为0,1,3,6,10,15
- v3为input[i]后面v5个数的和
- 2^v5==v3

check函数如下

```
signed int __cdecl sub_8048783(int a1, signed int a2)
{
  int v3; // [esp+10h] [ebp-10h]
  int v4; // [esp+14h] [ebp-Ch]
  signed int i; // [esp+18h] [ebp-8h]
  int v6; // [esp+1Ch] [ebp-4h]

  v6 = 0;
  for ( i = 1; i < a2; ++i )
  {
    v4 = 0;
    v3 = i - 1;
    if ( !*(_DWORD *)(4 * i + a1) )
      return 0;
    while ( a2 - 1 > v3 )
    {
      v4 += *(_DWORD *)(4 * (v3 * (v3 + 1) / 2 + v6) + a1);
      ++v3;
    }
    if ( *(_DWORD *)(4 * (v3 * (v3 + 1) / 2 + i) + a1) != v4 )
      return 0;
    ++v6;
  }
  return 1;
}
```


基本上是指

[0]—[k-1]行的[j]列求和等于[k]行的 [i]

有点懵

试着写了几行符合要求的数字

发现是杨辉三角

那就简单了

Python3简单实现杨辉三角

```
def createL(l):
    L = [1]
    for x in range(1, len(l)):
        L.append(l[x] + l[x-1])
    L.append(1)
    return L

def printL(L, W):
    s = ""
    for x in L:
        s += str(x) + " "
    print(s.center(W))

def str_(s, L):
    for x in L:
        s += str(x)
    return s

import hashlib
L = [1]
s = ""
row = int(input("请输入行数: "))
width =row* 6
for x in range(row):
    printL(L, width)
    s = str_(s, L)
    L = createL(L)
print(s)
m = hashlib.md5(s.encode()).hexdigest()
print("RCTF{%s" %m + "}")
```

得到flag：`RCTF{37894beff1c632010dd6d524aa9604db}`

结语

就是如何发现是个杨辉三角