【CSICTF】RicknMorty WriteUp

原创

古月浪子 于 2020-07-23 16:00:56 发布 109 收藏

文章标签： CTF

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/tqydyqt/article/details/107463894

版权

RicknMorty

452

binary reversing

Rick has been captured by the council of Rick's and in this dimension Morty has to save him, the chamber holding Rick needs a key. Can you help him find the key?

nc chall.csivit.com 30827

RickNMorty

第一道逆向题

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  __int64 v3; // rsi
  unsigned int v4; // eax
  __int64 v5; // rax
  __int64 v6; // rax
  __int64 v8; // [rsp+0h] [rbp-40h]
  time_t v9; // [rsp+8h] [rbp-38h]
  time_t v10; // [rsp+10h] [rbp-30h]
  time_t timer; // [rsp+18h] [rbp-28h]
  __int64 v12; // [rsp+20h] [rbp-20h]
  __int64 v13; // [rsp+28h] [rbp-18h]
  double v14; // [rsp+30h] [rbp-10h]
  int i; // [rsp+38h] [rbp-8h]
  int v16; // [rsp+3Ch] [rbp-4h]

  setbuf(stdin, 0LL);
  setbuf(stdout, 0LL);
  v3 = 0LL;
  setbuf(stderr, 0LL);
  v4 = time(&timer);
  srand(v4);
  time(&v10);
  v16 = 1;
  for ( i = 0; i <= rand() % 3 + 4; ++i )
  {
    v13 = rand() % 10 + 6;
    v12 = rand() % 10 + 6;
    printf("%d %d\n", v13, v12);
    __isoc99_scanf("%lld", &v8);
    v3 = v12;
    v5 = function1(v13, v12);
    v6 = function2(v5 + 3);
    if ( v6 != v8 )
      v16 = 0;
  }
  time(&v9);
  v14 = (v9 - v10);
  printf("fun() took %f seconds to execute \n", v3, v14);
  if ( v16 != 1 || v14 > 30.0 )
  {
```

```
    printf("Nahh.");
  }
  else
  {
    puts("Hey, you got me!");
    system("cat flag.txt");
  }
  return 0;
}
```

```
__int64 __fastcall function1(signed __int64 a1, signed __int64 a2)
{
  signed int i; // [rsp+18h] [rbp-8h]
  signed int v4; // [rsp+1Ch] [rbp-4h]

  v4 = 0;
  for ( i = 1; a1 >= i || a2 >= i; ++i )
  {
    if ( !(a1 % i) && !(a2 % i) )
      v4 = i;
  }
  return v4;
}
```

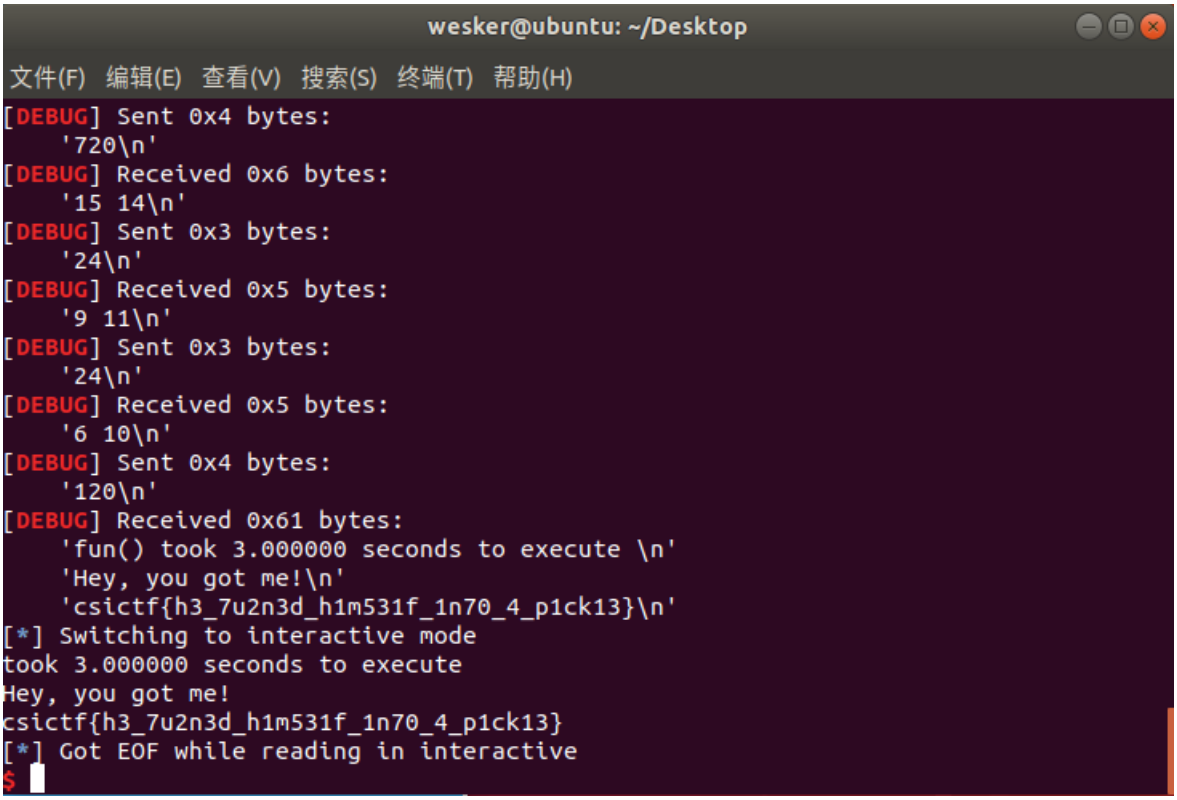
```
__int64 __fastcall function2(__int64 a1)
{
  __int64 result; // rax

  if ( a1 )
    result = a1 * function2(a1 - 1);
  else
    result = 1LL;
  return result;
}
```

大致流程就是，随机给你出几道题，题目是这样的：给你2个不是太大的数，求最小公因数，然后计算最小公因数+3的阶乘
所有题目回答正确并且用时小于30秒就能获得flag

> 讲真的，这种程度的题目我室友应该能2秒一道口算出结果

我算不了这么快，还是老老实实上python脚本吧
这里借用一下pwn题的模板

```python
from pwn import *
from LibcSearcher import *
from struct import pack

context.os='linux'
context.arch='amd64'
context.log_level='debug'

sd=lambda x:io.send(x)
sl=lambda x:io.sendline(x)
ru=lambda x:io.recvuntil(x)
rl=lambda :io.recvline()
ra=lambda :io.recv()
rn=lambda x:io.recv(x)
sla=lambda x,y:io.sendlineafter(x,y)

def fun1(a,b):
    r=0
    i=1
    while True:
        if a<i or b<i:
            break
        if not(a%i) and not(b%i):
            r=i
        i=i+1
    return r

def fun2(a):
    if a!=0:
        return a*fun2(a-1)
    else:
        return 1

io=remote('chall.csivit.com',30827)

while True:
    t=ru(' ')[:-1]
    if t=='fun()':
        break
    a=int(t)
    b=int(rl()[:-1])
    sl(str(fun2(fun1(a,b)+3)))

io.interactive()
```


wesker@ubuntu: ~/Desktop
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[DEBUG] Sent 0x4 bytes:
'720\n'
[DEBUG] Received 0x6 bytes:
'15 14\n'
[DEBUG] Sent 0x3 bytes:
'24\n'
[DEBUG] Received 0x5 bytes:
'9 11\n'
[DEBUG] Sent 0x3 bytes:
'24\n'
[DEBUG] Received 0x5 bytes:
'6 10\n'
[DEBUG] Sent 0x4 bytes:
'120\n'
[DEBUG] Received 0x61 bytes:
'fun() took 3.000000 seconds to execute \n'
'Hey, you got me!\n'
'csictf{h3_7u2n3d_h1m531f_1n70_4_p1ck13}\n'
[*] Switching to interactive mode
took 3.000000 seconds to execute
Hey, you got me!
csictf{h3_7u2n3d_h1m531f_1n70_4_p1ck13}
[*] Got EOF while reading in interactive
$