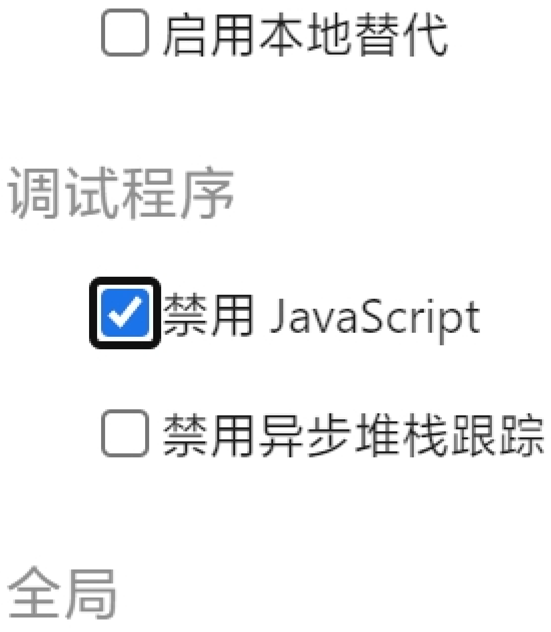
WEB

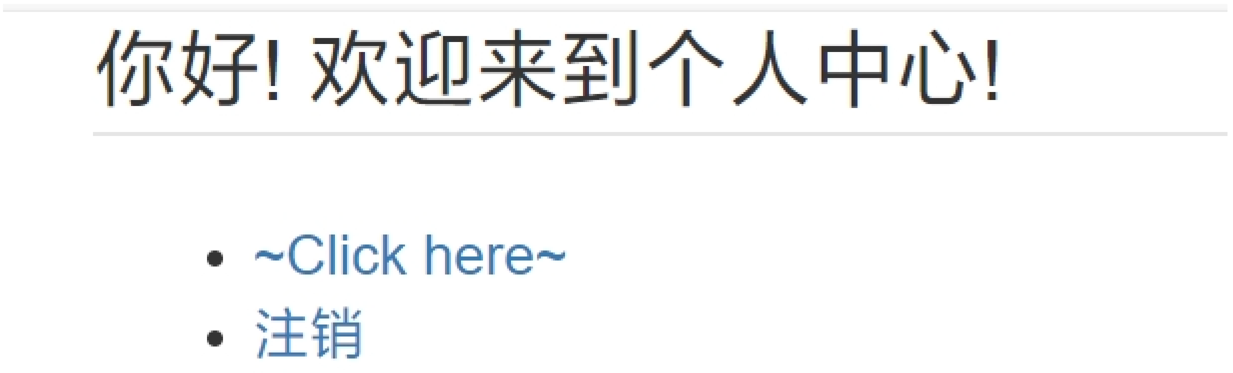
Bypass it

禁用js防止注册时弹窗即可



[✓ 禁用 JavaScript]

随后进行注册登陆即可获得flag



你好! 欢迎来到个人中心!

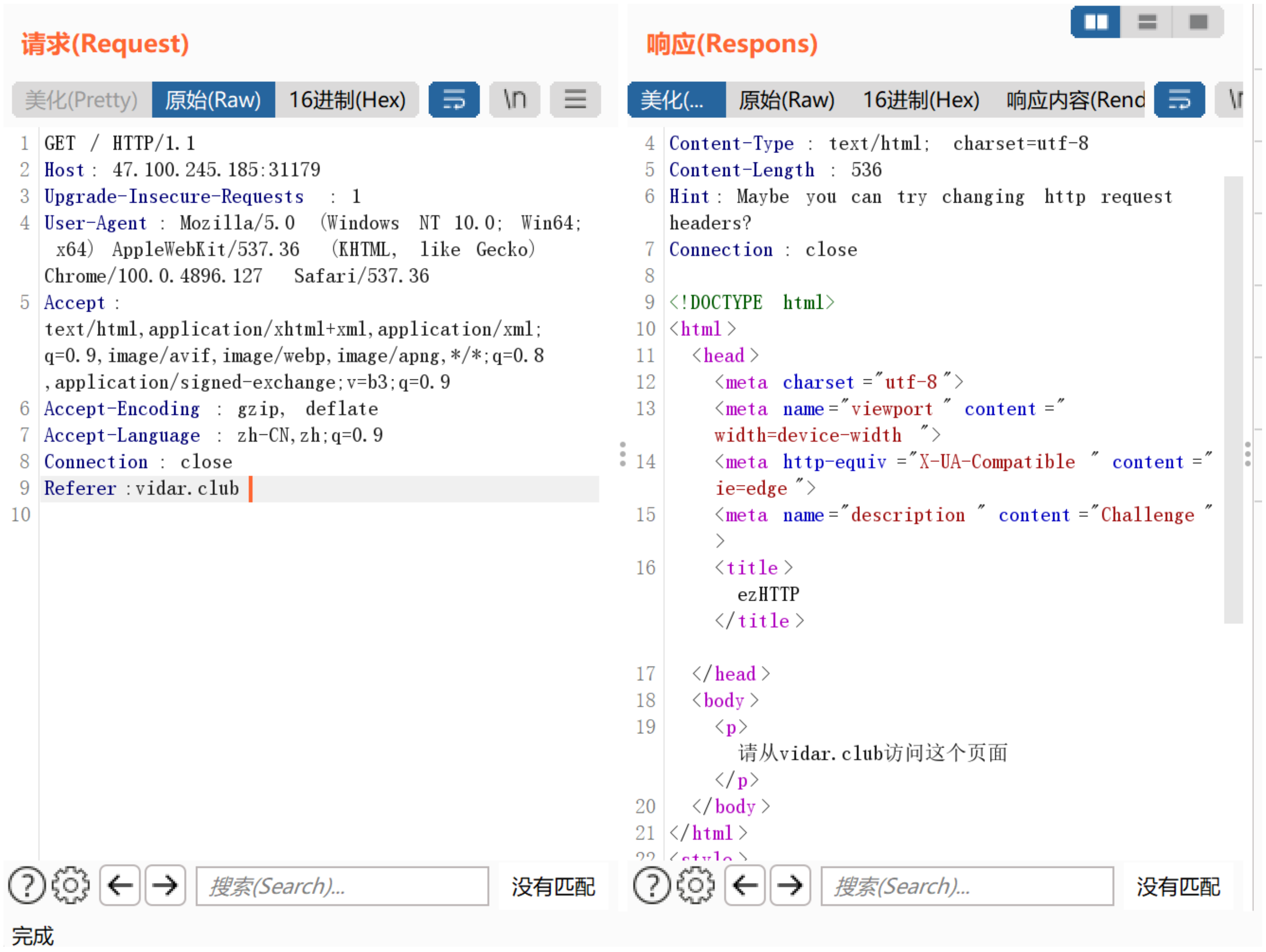
- ~Click here~
- 注销

点击获取flag

```
hgame{e3983cd6c6de1c35e21e592d443965ac6c2eb93c}
```

ezHTTP

从vidar.club访问这个页面
将请求中的Referer值修改为vidar.club

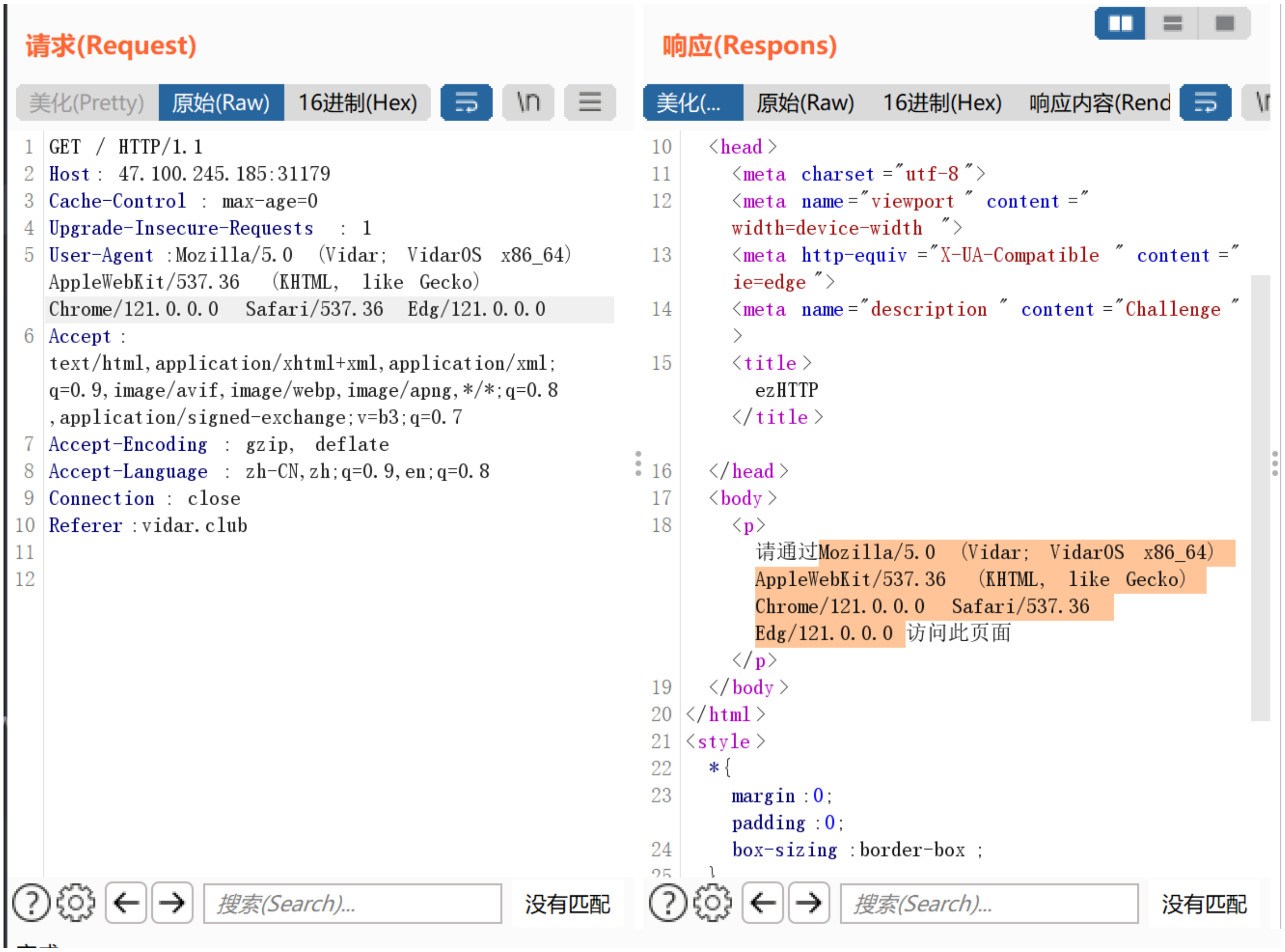


请求(Request)
美化(Pretty) 原始(Raw) 16进制(Hex)
响应(Respons)
美化(... 原始(Raw) 16进制(Hex) 响应内容(Rend
搜索(Search)...
没有匹配
完成

```
GET / HTTP/1.1
Host: 47.100.245.185:31179
Upgrade-Insecure-Requests : 1
User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127 Safari/537.36
Accept : text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding : gzip, deflate
Accept-Language : zh-CN,zh;q=0.9
Connection : close
Referer :vidar.club
```

```
Content-Type : text/html; charset=utf-8
Content-Length : 536
Hint: Maybe you can try changing http request headers?
Connection : close

<!DOCTYPE html>
<html>
  <head>
    <meta charset ="utf-8">
    <meta name="viewport" content ="width=device-width ">
    <meta http-equiv ="X-UA-Compatible " content ="ie=edge ">
    <meta name="description " content ="Challenge "
    >
    <title>
      ezHTTP
    </title>

  </head>
  <body>
    <p>
      请从vidar.club访问这个页面
    </p>
  </body>
</html>
<style>
```

请通过Mozilla/5.0 (Vidar; VidarOS x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0访问此页面
将请求中的User-Agent值修改为上述值


请求(Request)
美化(Pretty) 原始(Raw) 16进制(Hex)
响应(Respons)
美化(... 原始(Raw) 16进制(Hex) 响应内容(Rend
搜索(Search)... 没有匹配
搜索(Search)... 没有匹配

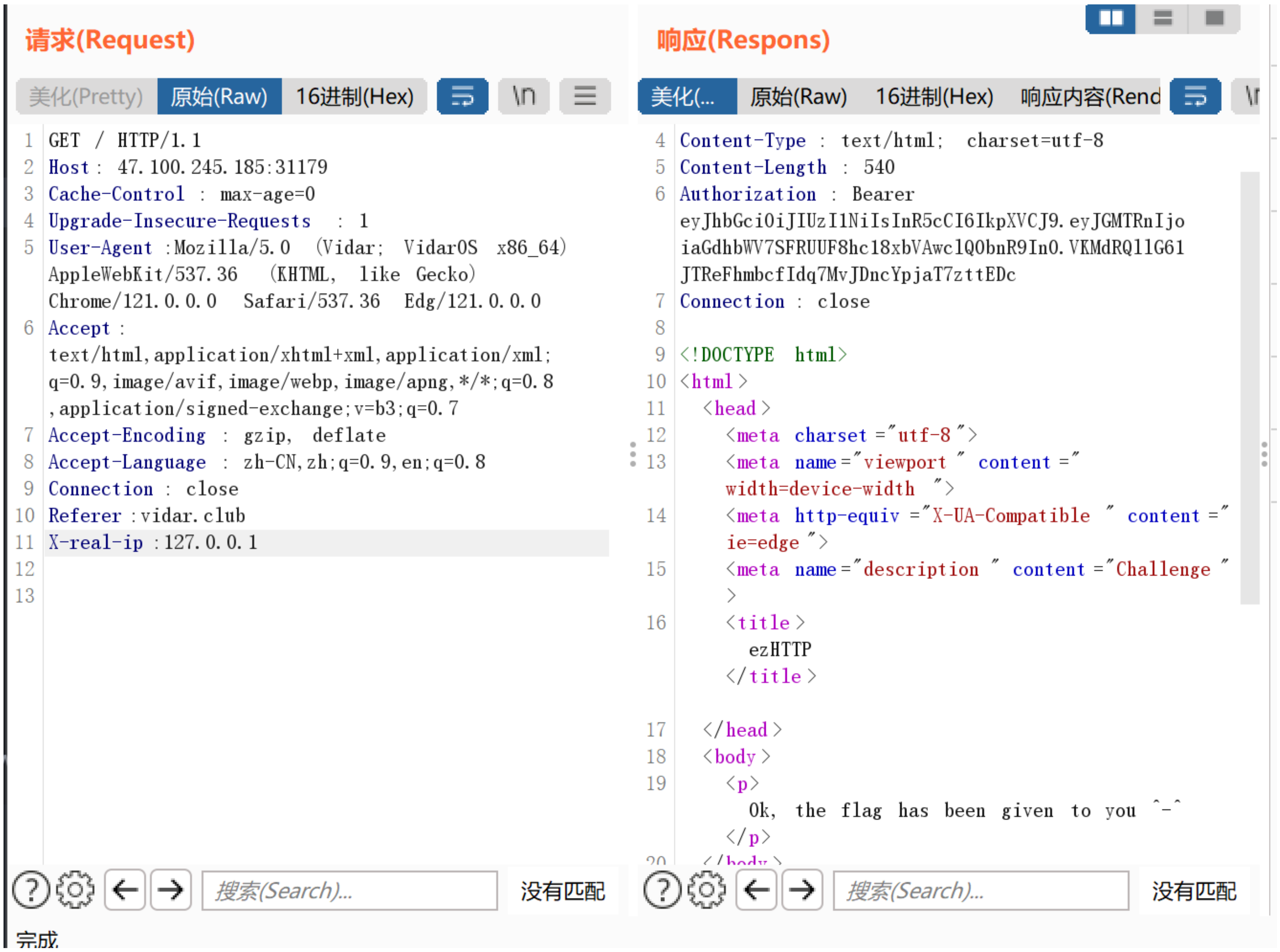
```
GET / HTTP/1.1
Host: 47.100.245.185:31179
Cache-Control : max-age=0
Upgrade-Insecure-Requests : 1
User-Agent :Mozilla/5.0 (Vidar; VidarOS x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding : gzip, deflate
Accept-Language : zh-CN,zh;q=0.9,en;q=0.8
Connection : close
Referer :vidar.club
```

```
  <head>
    <meta charset ="utf-8">
    <meta name="viewport" content ="
    width=device-width ">
    <meta http-equiv ="X-UA-Compatible" content ="
    ie=edge ">
    <meta name="description" content ="Challenge "
    >
    <title>
      ezHTTP
    </title>

  </head>
  <body>
    <p>
      请通过Mozilla/5.0 (Vidar; VidarOS x86_64)
      AppleWebKit/537.36 (KHTML, like Gecko)
      Chrome/121.0.0.0 Safari/537.36
      Edg/121.0.0.0 访问此页面
    </p>
  </body>
</html>
<style>
  *{
    margin :0;
    padding :0;
    box-sizing :border-box ;
```

请从本地访问这个页面

在响应中有Hint:Not xff

尝试过许多姿势，将请求种X-real-ip值修改为127.0.0.1



```
GET / HTTP/1.1
Host: 47.100.245.185:31179
Cache-Control : max-age=0
Upgrade-Insecure-Requests : 1
User-Agent :Mozilla/5.0 (Vidar; VidarOS x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0
Accept : text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding : gzip, deflate
Accept-Language : zh-CN,zh;q=0.9,en;q=0.8
Connection : close
Referer :vidar.club
X-real-ip :127.0.0.1
```

```
Content-Type : text/html; charset=utf-8
Content-Length : 540
Authorization : Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJGMTRnIjoiaGdhbWV7SFRUUF8hc18xbVAwclQ0bnR9In0.VKMdRQllG61JTReFhmbcfIdq7MvJDncYpjaT7zttEDc
Connection : close

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width ">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <meta name="description" content="Challenge">
    <title>
      ezHTTP
    </title>

  </head>
  <body>
    <p>
      Ok, the flag has been given to you ^-^
    </p>
```

可以看到响应中有一串jwt token，解码可得flag

```
hgame{HTTP_!s_1mP0rT4nt}
```

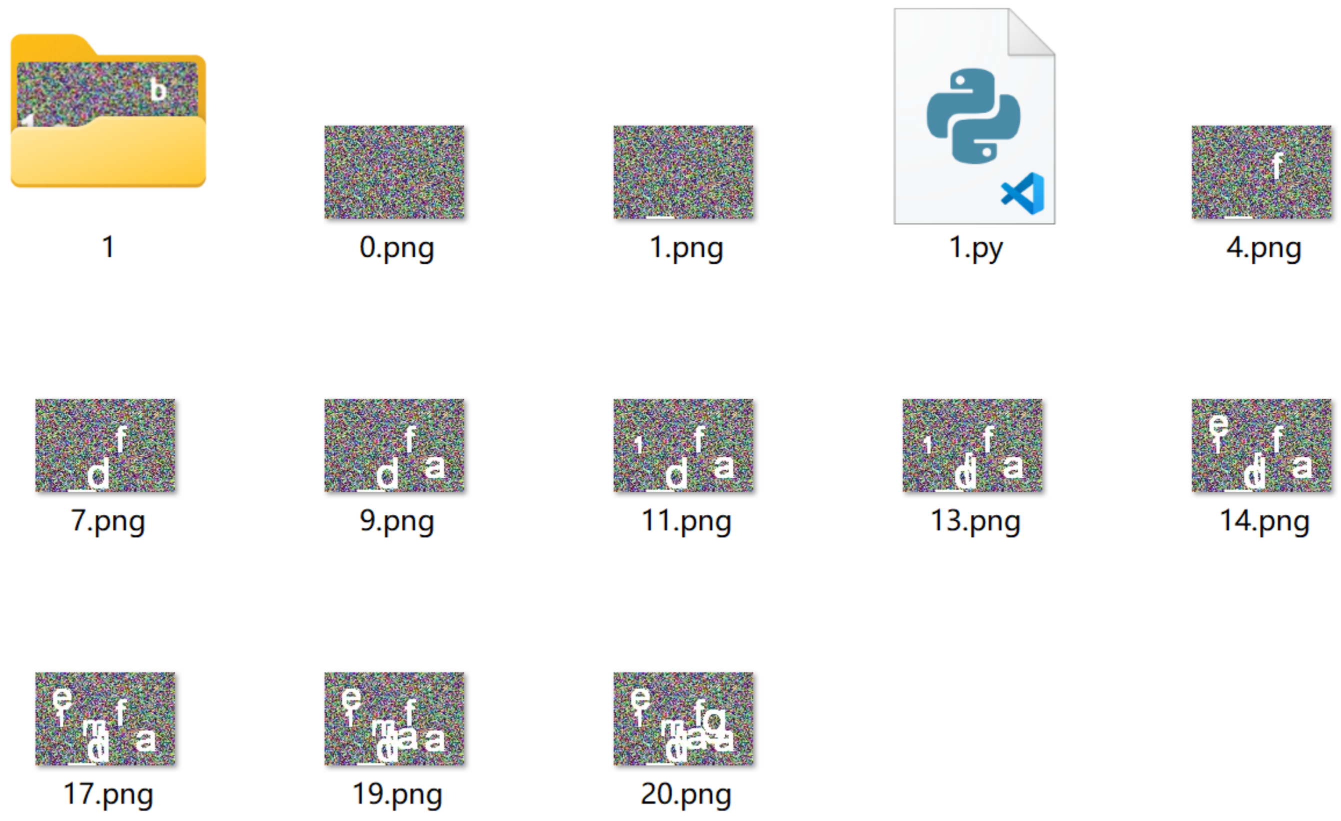
CRYPTO

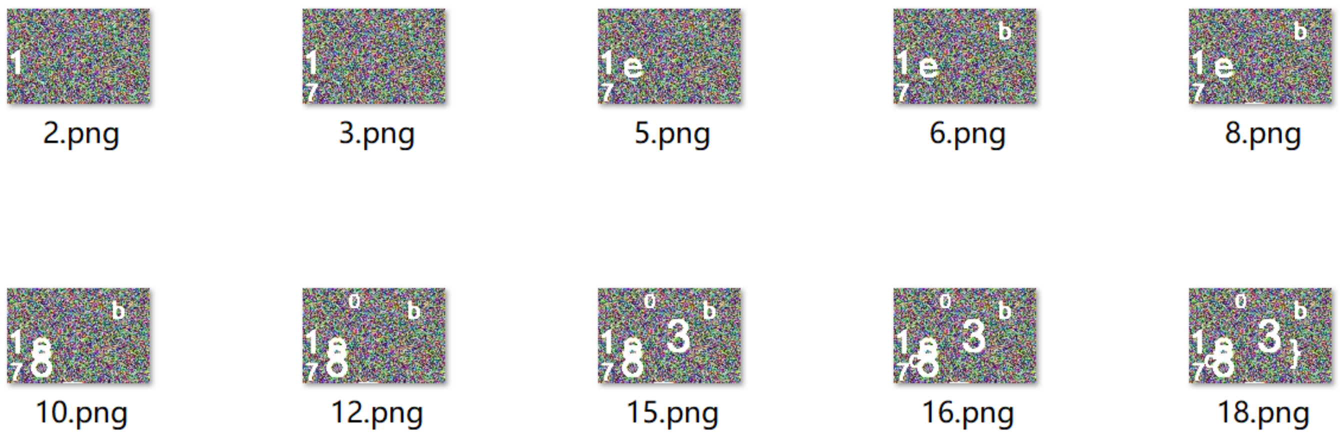
奇怪的图片

题目将flag的每个字符按序draw在一张随机生成的图片中，并每张与随机生成的key image进行xor操作，save。但由于time.sleep()，无法确认每张image的生成顺序。
按顺序来说，每张image都会在前一张添加一个字符，而未改变的像素点，两张图片进行xor后仍然相同，那么就随意取一张image与另外所有的image进行比较，将相同的像素点draw在一张白底的image上，剩下的白色部分即draw在上面的字符。

```python
from PIL import Image, ImageDraw
import os

def list_files_in_directory(directory):
    name=[]
    files = os.listdir(directory)
    for file in files:
        file_path = os.path.join(directory, file)
        if os.path.isfile(file_path):
            name.append(file)
    return name

directory_path = "D:/crane/Desktop/attachment (3)/png_out"



def compare_images(image1, image2):
    i=0
    if image1.size != image2.size:
        raise ValueError("Images must have the same dimensions.")

    width, height = image1.size
    result_image = Image.new("RGB", (width, height), "white")
    result_draw = ImageDraw.Draw(result_image)

    for x in range(width):
        for y in range(height):
            pixel1 = image1.getpixel((x, y))
            pixel2 = image2.getpixel((x, y))

            if pixel1 == pixel2:
                i=i+1
                result_draw.point((x, y), fill=pixel1)

    return result_image,i

name=list_files_in_directory(directory_path)
rank=[]
result=[]
for i in range(len(name)):
    image1_path = "D:/crane/Desktop/attachment (3)/png_out/4e8a536e.png"
    image2_path = "D:/crane/Desktop/attachment (3)/png_out/"+name[i]
    image1 = Image.open(image1_path, 'r')
    image2 = Image.open(image2_path, 'r')
    result_image,r = compare_images(image1, image2)
    rank.append(r)
    result.append(result_image)
    result_image.save("{}.png".format(i))

for i in range(len(rank)):
    index=i
    max=rank[i]
    for j in range(i+1,len(rank)):
        if rank[index]<rank[j]:
            max=rank[j]
            index=j
    if max!=rank[i]:
            rank[index]=rank[i]
            rank[i]=max
            temp=result[i]
            result[i]=result[index]
            result[index]=temp

for i in range(len(result)):
    result[i].save("{}.png".format(i))
    print(rank[i])
    print(name[i])
    print("{}.png".format(i))
    print("--------------------------------------------------")
```

由于是随机选择的image，需要将生成的image进行排序分类，以便分析。





一组是在选择的image后添加的字符，一组为image前添加的字符，分析添加的顺序，与flag格式hgame{}可得flag

```
hgame{1adf_17eb_803c}
```

ezMath

分析代码，采用AES将flag加密，key由以下语句生成

```
D = 114514
assert x**2 - D * y**2 == 1
key=pad(long_to_bytes(y))[:16]
```

可以发现这是佩尔方程，用连分数法解出y即可

```python
import numpy as np
from collections import deque
from Crypto.Util.number import *
from Crypto.Cipher import AES

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)
print(c)

def pad(x):
    return x+b'\x00'*(16-len(x)%16)

def decrypt(KEY):
    cipher= AES.new(KEY,AES.MODE_ECB)
    decrypted =cipher.decrypt(enc)
    return decrypted

key=pad(long_to_bytes(c))[:16]
enc=b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9ad\x9e\x08b\xbf\x8b\xd3\r\xe2\x81\x17g\x9c\xd7\x10\x1
print(decrypt(key))
```

解出flag

```
hgame{G0od!_Yo3_k1ow_C0ntinued_Fra3ti0ns!!!!!!!}
```

ezRSA

解出p，q即可

```
leak1=pow(p,q,n)
leak2=pow(q,p,n)
```

可以发现leak1*leak2与leak1，leak2的公因数即是p，q

```
from Crypto.Util.number import *

leak1=149127170073611271968182576751290331559018441805725310426095412837589227670757540743929865
leak2=116122992714670915381309916967490436489020001172880644167179915467021794892927977272080596
c=105294818675325200342580567738640740170270195780418662454006478402302516616529997097159196208

p = GCD(leak1*leak2, leak2)
q = GCD(leak1*leak2, leak1)
phi=(p-1)*(q-1)
e=0x10001
d = inverse(e, phi)
n=p*q
m = pow(c, d, n)
plain=long_to_bytes(m)
print(plain)
```

解出flag

```
hgame{F3rmat_l1tt1e_the0rem_is_th3_bas1s}
```

ezPRNG

LFSR问题，可以发现生成的output每32位为一周期，将随机生成的uuid的去除‘-’，每8位进行加密。mask只有第1、4、8、11、15、20、25、28、32这几位为1，其余位均为0，反馈函数即每1位上的异或。最后再依次异或解出flag。

```
mask=0b10001001000010000100010010001001
input=['1111110110111011110000101011010001000111111001111110100101000011101111111000100001111
result=''
for i in range(4):
      key1=input[i][0:32]
      key2=key1
      flag=[]
      for i in range(32):
            output='?'+key1[:31]
            flag.append(str(int(key2[-1-i])^int(output[-1])^int(output[-4])^int(output[-8])^int
            key1=str(flag[i])+key1[:31]
      result+=hex(int(''.join(flag[::-1]),2)).replace('0x','')
result=list(result)
result.insert(8,'-')
result.insert(13,'-')
result.insert(18,'-')
result.insert(23,'-')
result=''.join(result)
print("hgame{"+result+"}")
```

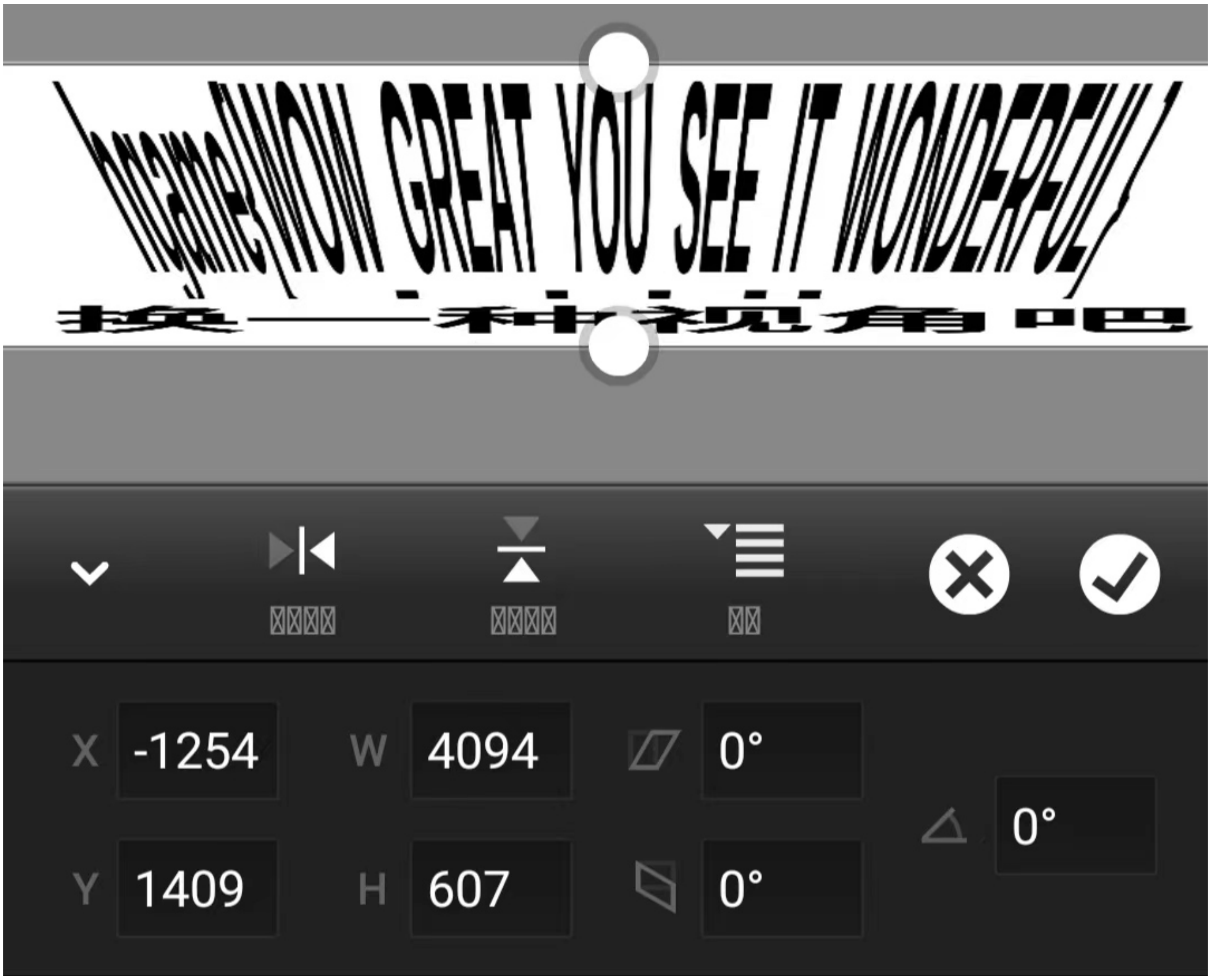
注意结果要根据uuid增加“-”。

```
hgame{fbbbee82-3f43-4f91-9337-907880e4191a}
```

MISC

SignIn

可以发现图片宽度上进行了压缩，将图片拉长即可

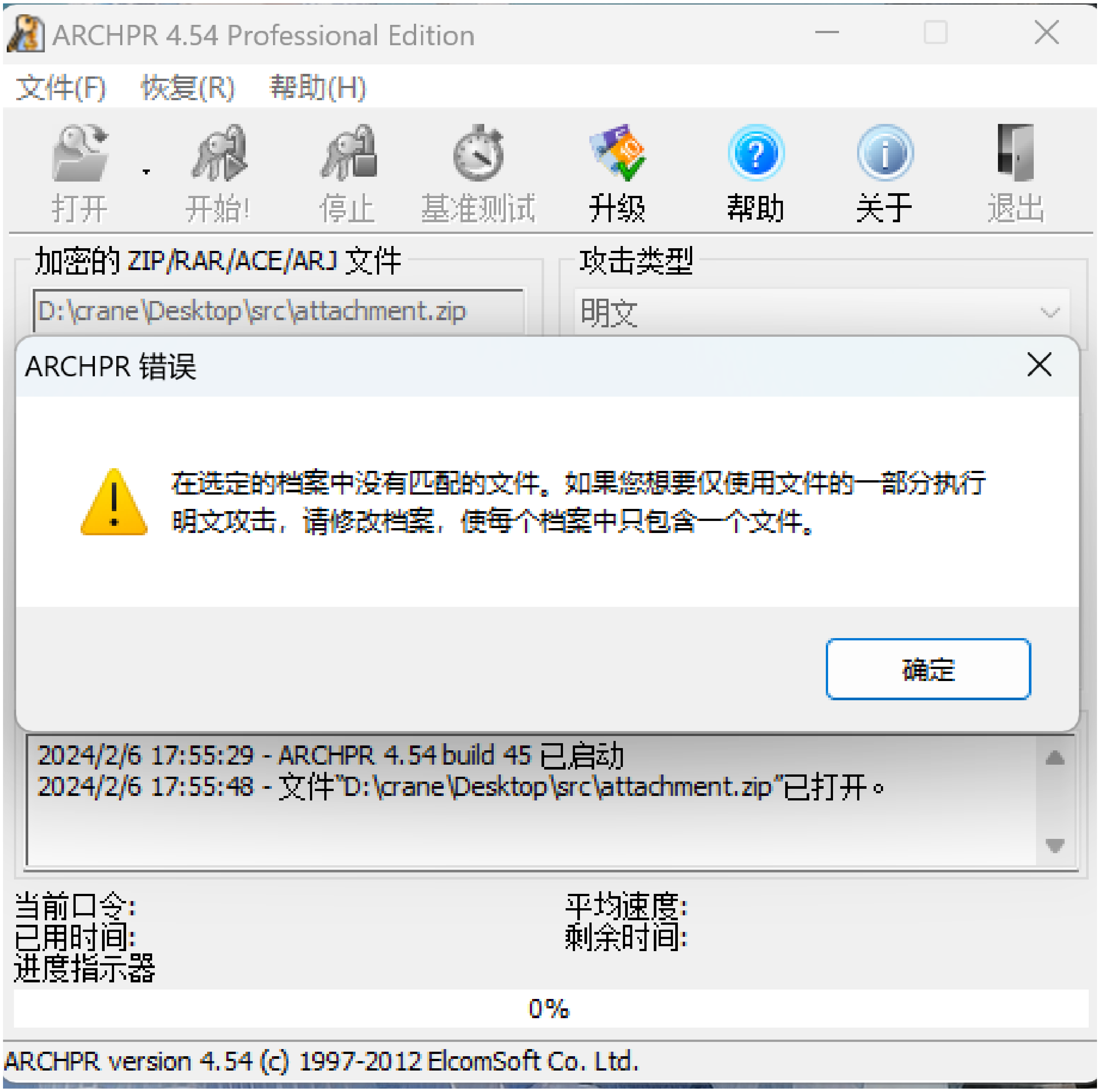


可以看到flag为

```
hgame{WOW_GREAT_YOU_SEE_IT_WONDERFUL}
```

simple_attack

是个zip加密，给了一张jpg与一个加密zip，查看zip内文件名，发现有与给出的jpg相同名称的文件。那么即是压缩包已知明文攻击，采用ARCHPR工具进行攻击。
初始攻击时，用winrar压缩图片，会报如图错误



加密的 ZIP/RAR/ACE/ARJ 文件

D:\crane\Desktop\src\attachment.zip

攻击类型

明文

ARCHPR 错误

在选定的档案中没有匹配的文件。如果您想要仅使用文件的一部分执行明文攻击，请修改档案，使每个档案中只包含一个文件。

```
2024/2/6 17:55:29 - ARCHPR 4.54 build 45 已启动
2024/2/6 17:55:48 - 文件"D:\crane\Desktop\src\attachment.zip"已打开。
```

当前口令:

平均速度:

已用时间:

剩余时间:

进度指示器

0%

ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co. Ltd.

试过了许多压缩方式都不行（，最后询问出题人，用的是bandzip压缩。

攻击完成后压缩包内有photo.txt文件，为图片base64编码，且为url链接，复制到游览器即可打开。

```
hgame{s1mple_attack_for_zip}
```

得到flag

```
hgame{s1mple_attack_for_zip}
```