

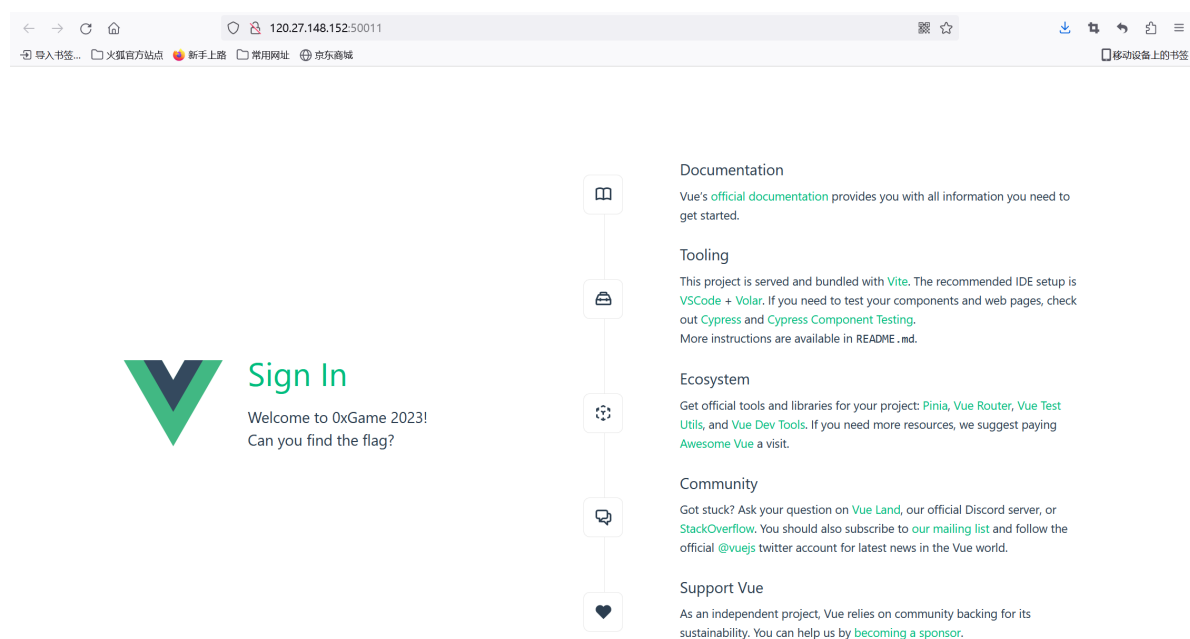
0xGame2023 Week1 Writeup

这是我作为一个零基础noob第一次打CTF，因此在做题过程中遇到了重重困难o(ᑦ_____ᑦ)o，基本上是做题两分钟，搜索两小时。。。没关系，不求跟佬比，在比赛中学到的一切，那都是收获。以下是我做出来的题的解题报告。

Web

signin

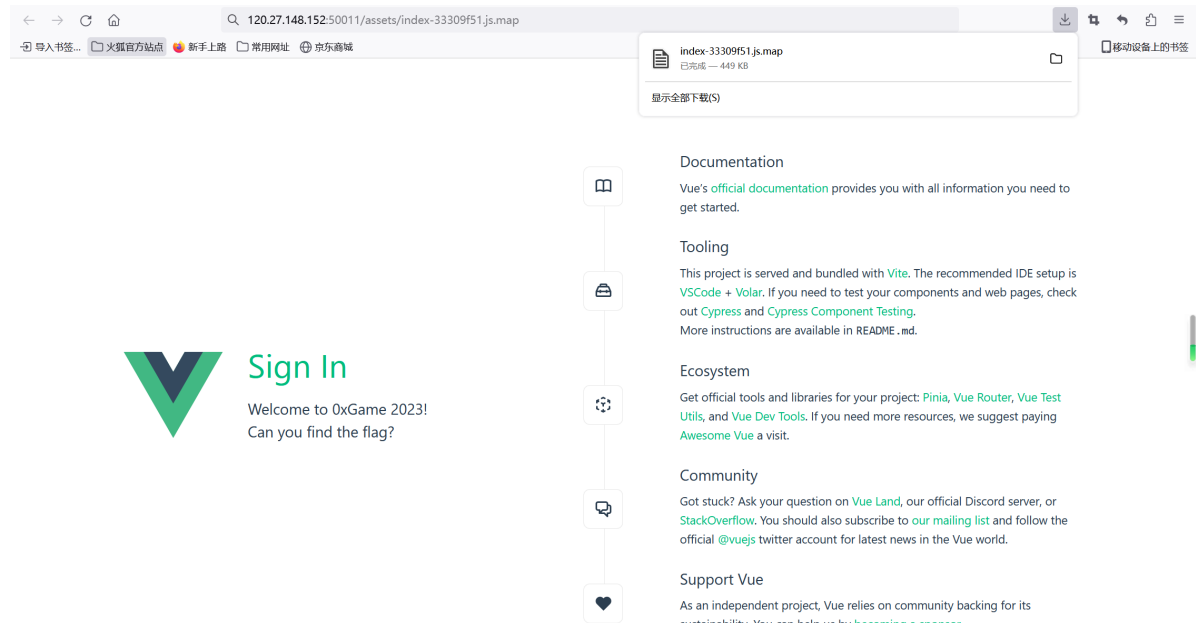
签到题，先上题目



查看一下源代码，发现一个js文件，打开后发现注释



那就根据他给的URL去找咯~由于发现源代码中所有文件都放在了/assets路径下，于是



在下载下来的文件中搜索0xGame，得到flag。

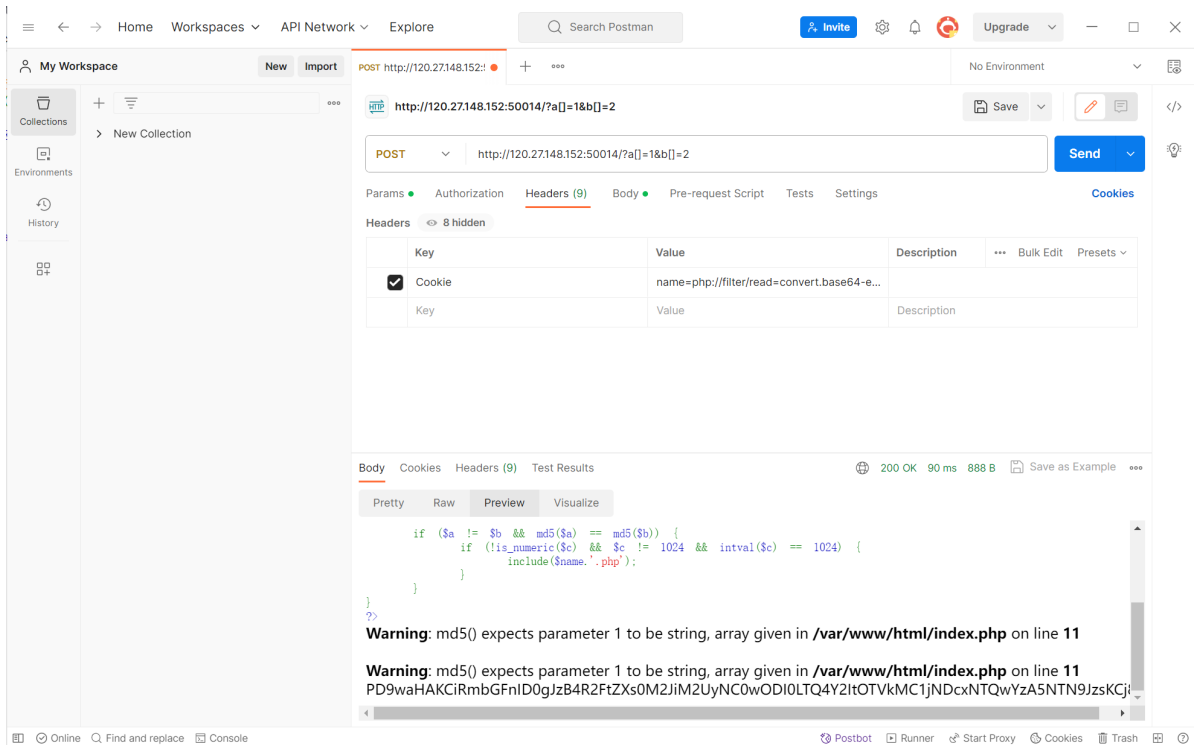
```
./App.vue \n\ncreateApp(App).mount( #app )\n\nlet flag = false;
'0xGame{c788fa80-2184-429f-b410-48cb8e2de0ff}',";"], "names": ["makeMap", "str", "expectsLowerCase", "map", "l
```

baby_php

题目



经过对程序的阅读分析，a和b是md5弱类型绕过，都Get传参数组即可。根据要求，c非数字或数字字符串，不等于1024，取整型的1024，于是进行Post传参c为1024.1a。d很简单，值应为flag，但由于注意到include，所以使用php伪协议读取，得到base64编码后的flag。

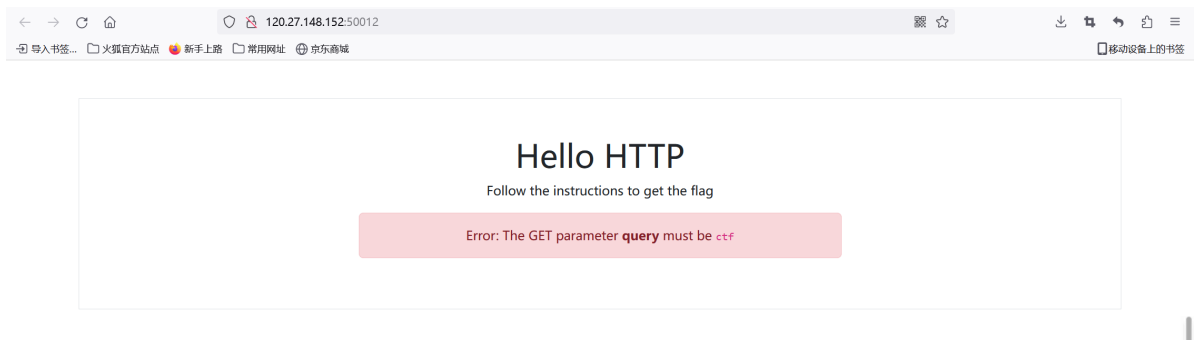


解码后得到flag。

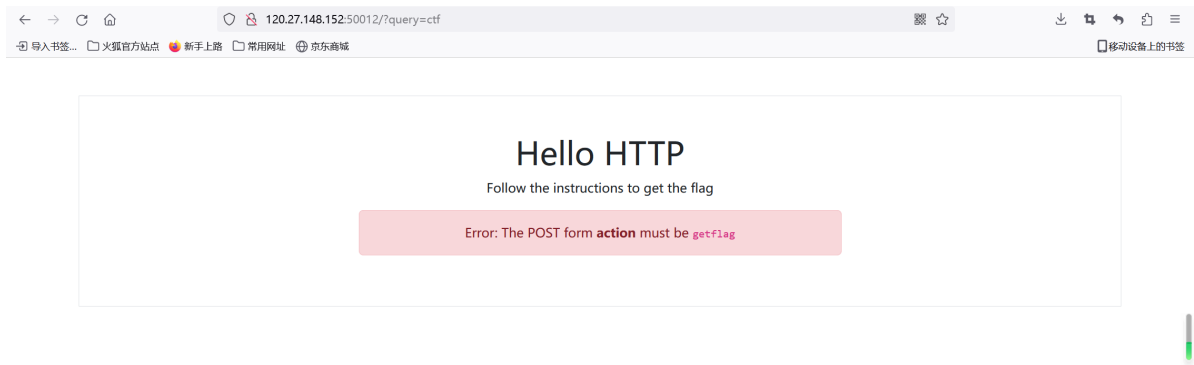


hello_http

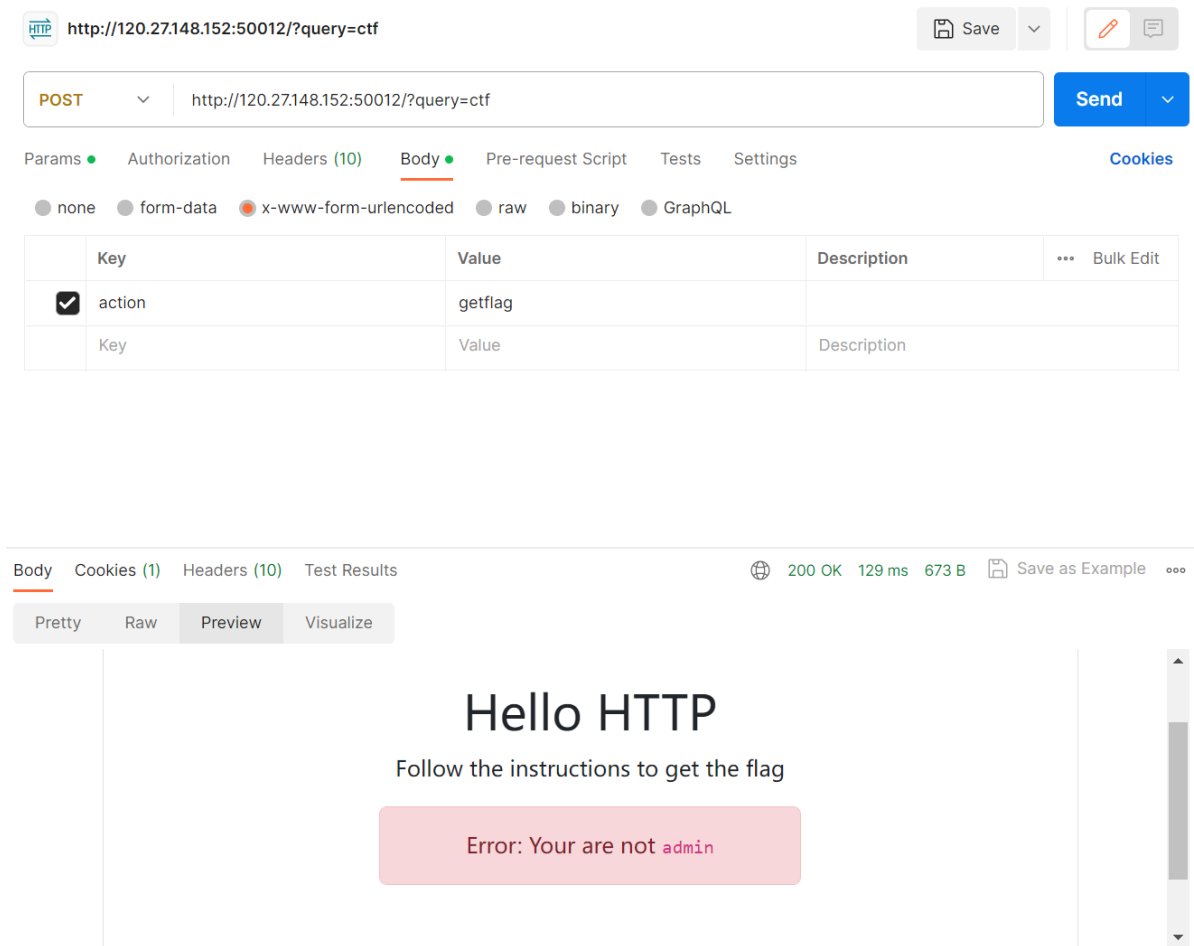
一道套娃题。。。



显然Get传参?query=ctf



显然Post传参action=getflag



对我来说这里开始就有点艰难了o(╥﹏╥)o，经过一番对http报文header的恶补之后才得以继续。这里是Cookie头

HTTP <http://120.27.148.152:50012/?query=ctf> Save

POST <http://120.27.148.152:50012/?query=ctf> Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Headers 9 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Cookie	role=admin				
	Key	Value	Description			

Body Cookies (1) Headers (10) Test Results 200 OK 133 ms 668 B Save as Example

Pretty Raw Preview Visualize

Hello HTTP

Follow the instructions to get the flag

Error: Only allow local IP

XFF头，伪造本地请求

HTTP <http://120.27.148.152:50012/?query=ctf> Save

POST <http://120.27.148.152:50012/?query=ctf> Send

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

Headers 9 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Cookie	role=admin				
<input checked="" type="checkbox"/>	X-Forwarded-For	127.0.0.1				
	Key	Value	Description			

Body Cookies (1) Headers (10) Test Results 200 OK 86 ms 702 B Save as Example

Pretty Raw Preview Visualize

Hello HTTP

Follow the instructions to get the flag

Error: You are not using HarmonyOS Browser

Postbot Runner Start Proxy Cookies Trash

UA头，标识使用的浏览器（遥遥领先）

HTTP <http://120.27.148.152:50012/?query=ctf> Save

POST <http://120.27.148.152:50012/?query=ctf> Send

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

Headers 9 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Cookie	role=admin				
<input checked="" type="checkbox"/>	X-Forwarded-For	127.0.0.1				
<input checked="" type="checkbox"/>	User-Agent	HarmonyOS Browser				
	Key	Value	Description			

Body Cookies (1) Headers (10) Test Results 200 OK 84 ms 701 B Save as Example

Pretty Raw Preview Visualize

Hello HTTP

Follow the instructions to get the flag

Error: Only allow access from ys.mihoyo.com 😊

Postbot Runner Start Proxy Cookies Trash

Referer头，表示请求来源（玩原神玩的↑）

HTTP <http://120.27.148.152:50012/?query=ctf> Save

POST <http://120.27.148.152:50012/?query=ctf> Send

Params Authorization Headers (13) Body Pre-request Script Tests Settings Cookies

Headers 9 hidden


	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Cookie	role=admin				
<input checked="" type="checkbox"/>	X-Forwarded-For	127.0.0.1				
<input checked="" type="checkbox"/>	User-Agent	HarmonyOS Browser				
<input checked="" type="checkbox"/>	Referer	ys.mihoyo.com				
	Key	Value	Description			

Body Cookies (1) Headers (10) Test Results 200 OK 104 ms 728 B Save as Example

Pretty Raw Preview Visualize

Hello HTTP

Follow the instructions to get the flag

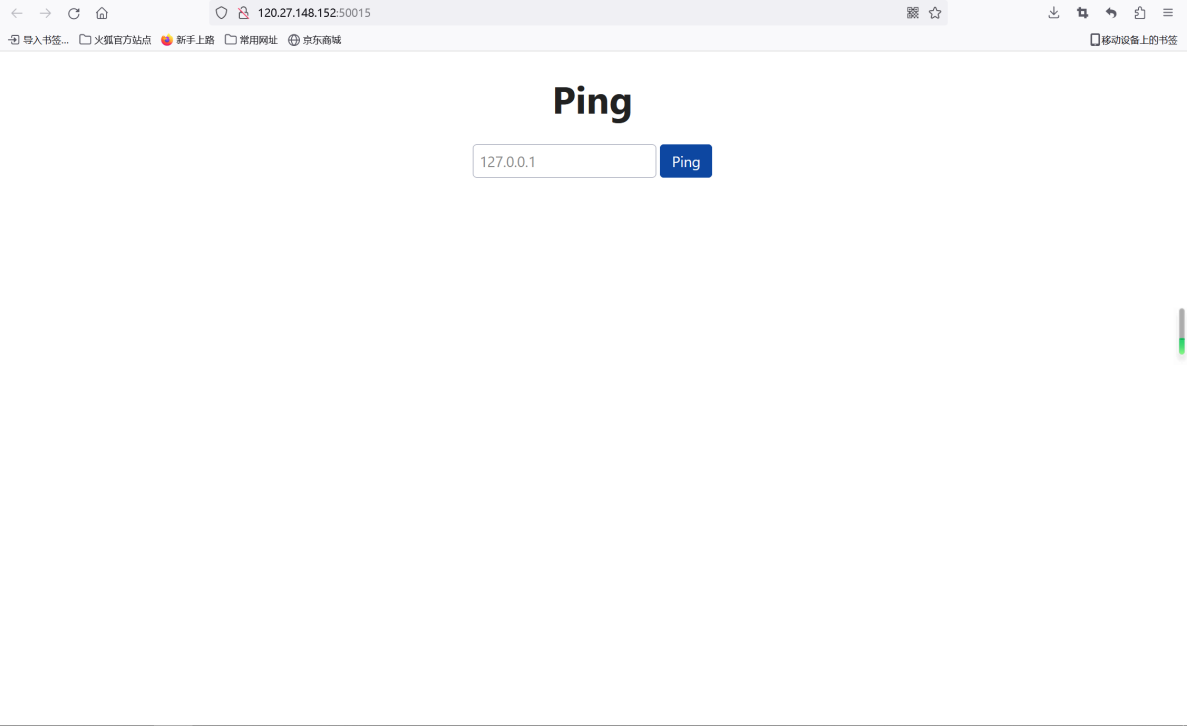
 Congratulations! Flag is
`0xGame{2c1a10fb-921e-4250-820f-5ce36940b8b5}`

Postbot Runner Start Proxy Cookies Trash

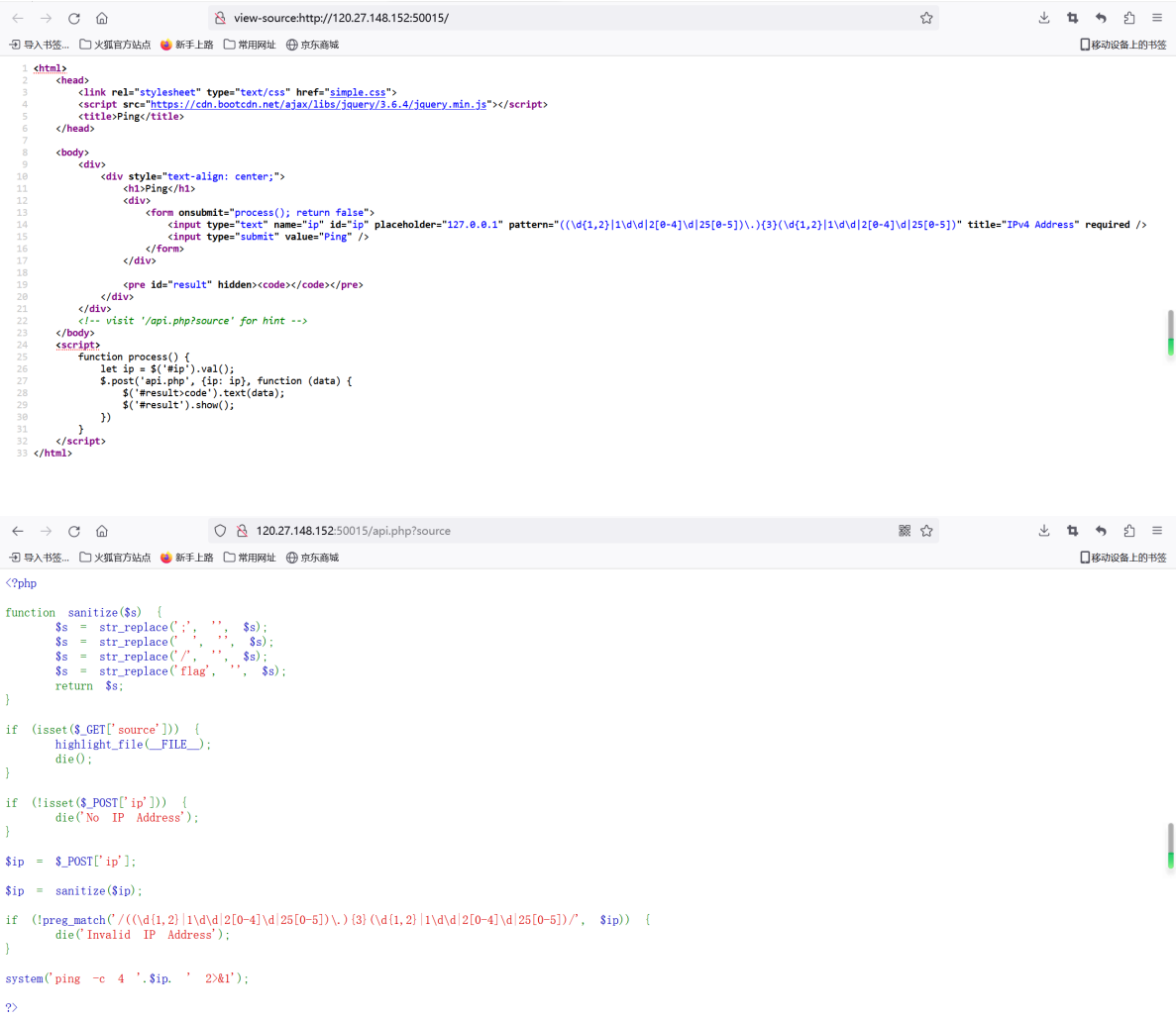
Finally,I got the flag.=)

ping

题目



检查一下源代码，发现注释，根据其指示路径找到提示。



搜索了亿下，是ping命令注入，其中过滤了字符和字符串、\、/、flag。经过noob一个晚上的奋战o(π_π)o，总算搞清楚了应该使用怎样的命令：ls /，但是得注意过滤。。。

HTTP New Collection / http://120.27.148.152:50015/api.php?source

Save

POST http://120.27.148.152:50015/api.php Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	ip	127.0.0.1&echo\${IFS}"bHMGlw=" base64...			
	Key	Value	Description		

Body Cookies Headers (9) Test Results 200 OK 3.17 s 800 B Save as Example

Pretty Raw Preview Visualize HTML

```
1 PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
2 64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.028 ms
3 bin
4 boot
5 dev
6 etc
7 flag
8 home
9 lib
10 lib64
11 media
12 mnt
```

Postbot Runner Start Proxy Cookies Trash

最终我使用了base64绕过，ls发现了根目录下的flag，这时只需用同样方式注入cat /flag命令即可

HTTP New Collection / http://120.27.148.152:50015/api.php?source

Save

POST http://120.27.148.152:50015/api.php Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	ip	127.0.0.1&echo\${IFS}"Y2F0IC9mbGFn" bas...			
	Key	Value	Description		

Body Cookies Headers (9) Test Results 200 OK 3.12 s 755 B Save as Example

Pretty Raw Preview Visualize HTML

```
1 PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
2 64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.047 ms
3 0xGame{19c71976-d7d8-4ab8-9ea5-6ea3800f59f6}
4 64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.029 ms
5 64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.034 ms
6 64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.039 ms
7
8 --- 127.0.0.1 ping statistics ---
9 4 packets transmitted, 4 received, 0% packet loss, time 3076ms
10 rtt min/avg/max/mdev = 0.029/0.037/0.047/0.006 ms
```

Postbot Runner Start Proxy Cookies Trash

Misc

SignIn

本人认为的week1中最简单的一道题

打开文件给了一个字符串，一眼看到最后的=，得知base64



结尾六个=，盲猜base32

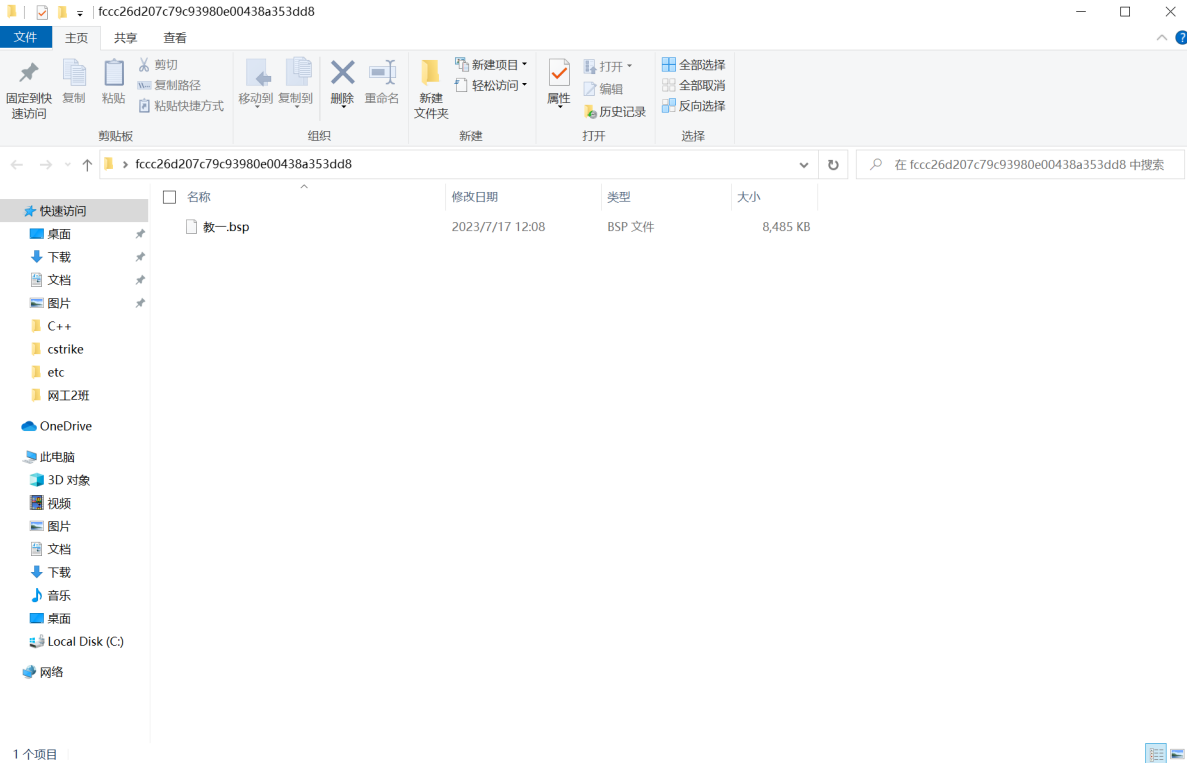


完成！

重生之我在教学楼打cs

学长竟然用学校的电脑打游戏

下载附件，得到一个.bsp文件



费尽千辛万苦，总算下载到了（



CS, 启动!

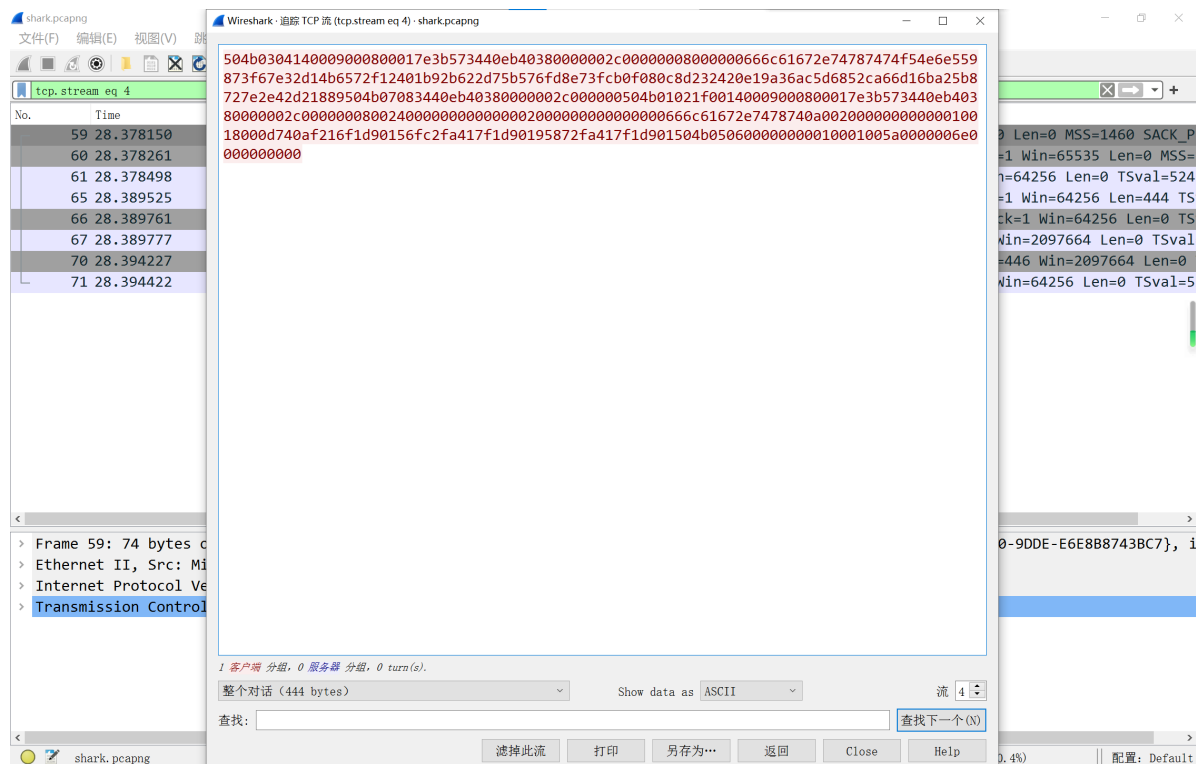
不得不说，当看到地图还真是我们的教一的时候，还是很震撼的。由于hint说一眼鼎真，flag也许就在脚下，可知flag应该就在司母戊鼎附近，于是我在司母戊鼎周围转了一圈又一圈。。。最后总算在基座的底部找到了那个极其细小而暗淡的字符串.....



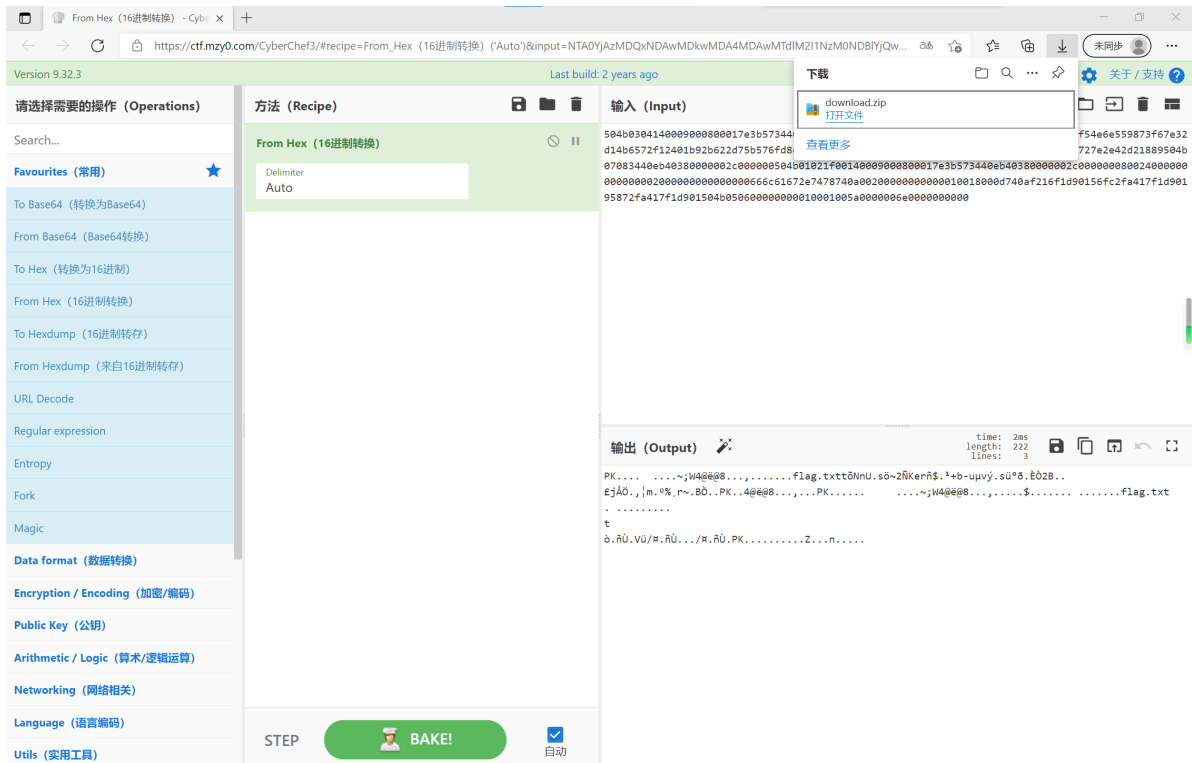
shark shark

一个流量分析题，但由于我是个noob，没有做这种题的经验，所以看了几天毫无头绪。。。最终问了出题人才知道，这题中间传输了一个hex形式的压缩包，下载下来即可获得flag。下面是操作过程。

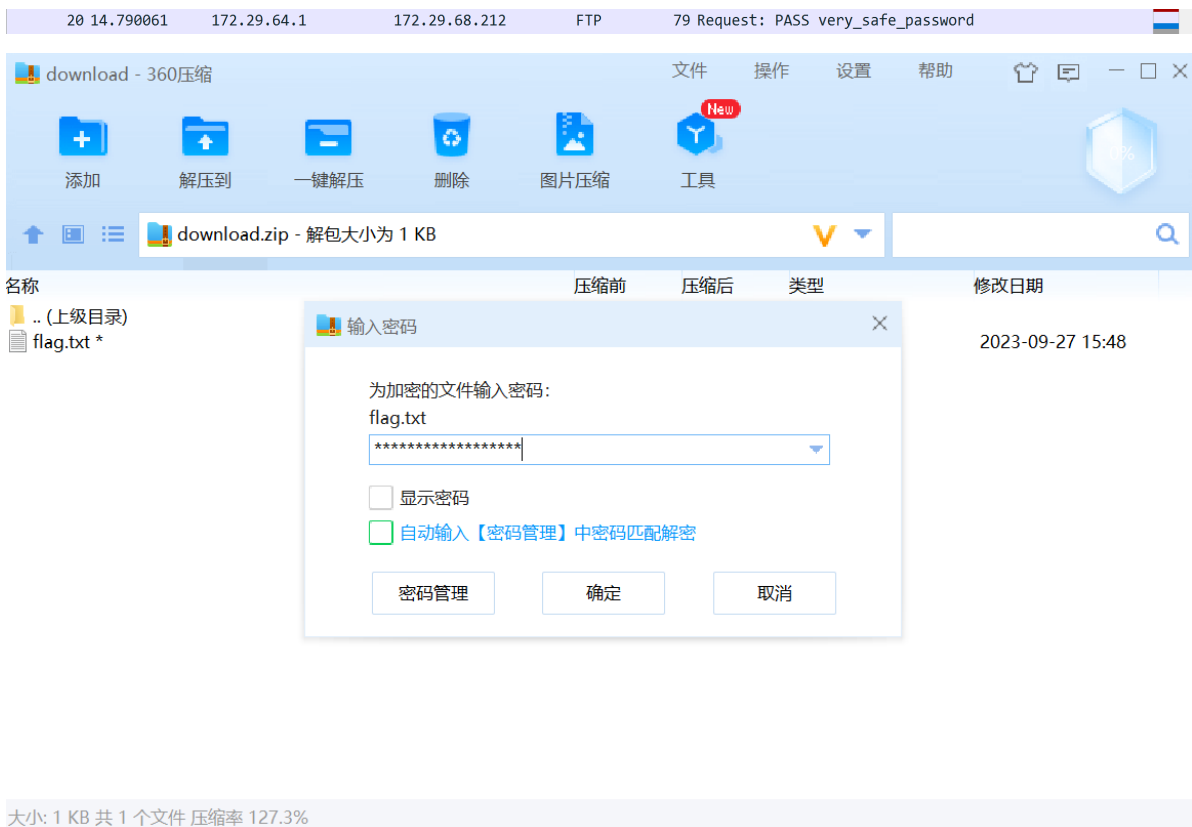
找到传输压缩包的那条并追踪tcp流，得到一串十六进制数。



用解码网站<https://ctf.mzy0.com/CyberChef3/>对其进行解码并导出为压缩包。



压缩包密码即为登录密码



得到flag

0xGame{7a504dab-ada6-4c41-adb3-0d1530098cd4}

hide and seek

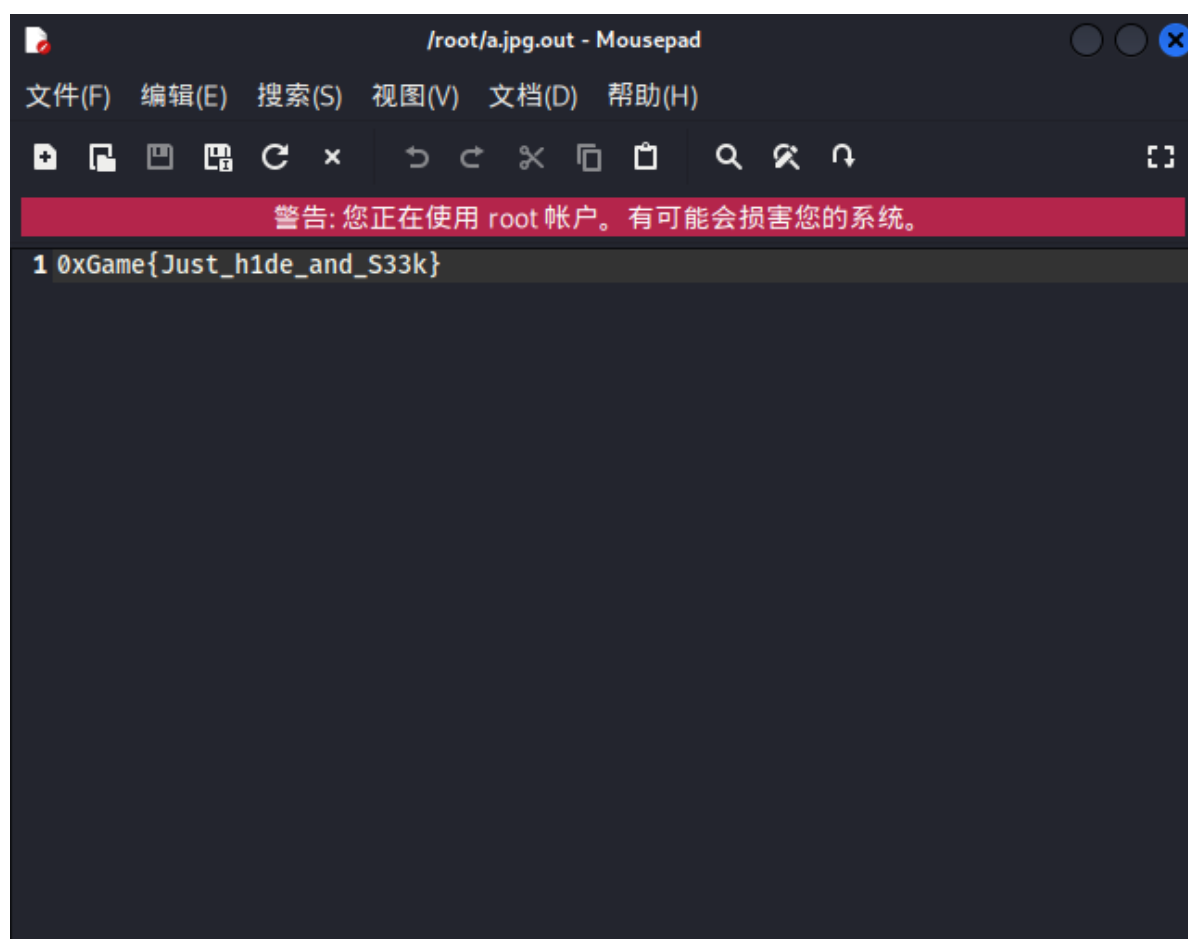
从本题提示中提取关键信息：steghide、seek

可知本题是使用了steghide将flag隐藏在了附件的图片中。当我在尝试使用steghide将其提取出来时发现需要密码，而题目中未给提示，那就只能使用爆破了，但用怎样的字典这个问题又困扰了本noob很久o(╥╰╚╥)o，在询问佬之后得知，有个叫做rockyou.txt的字典，于是我用stegseek对其进行了爆破，果然得到了flag。

```
(root@kali)~[~]
# stegseek a.jpg rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

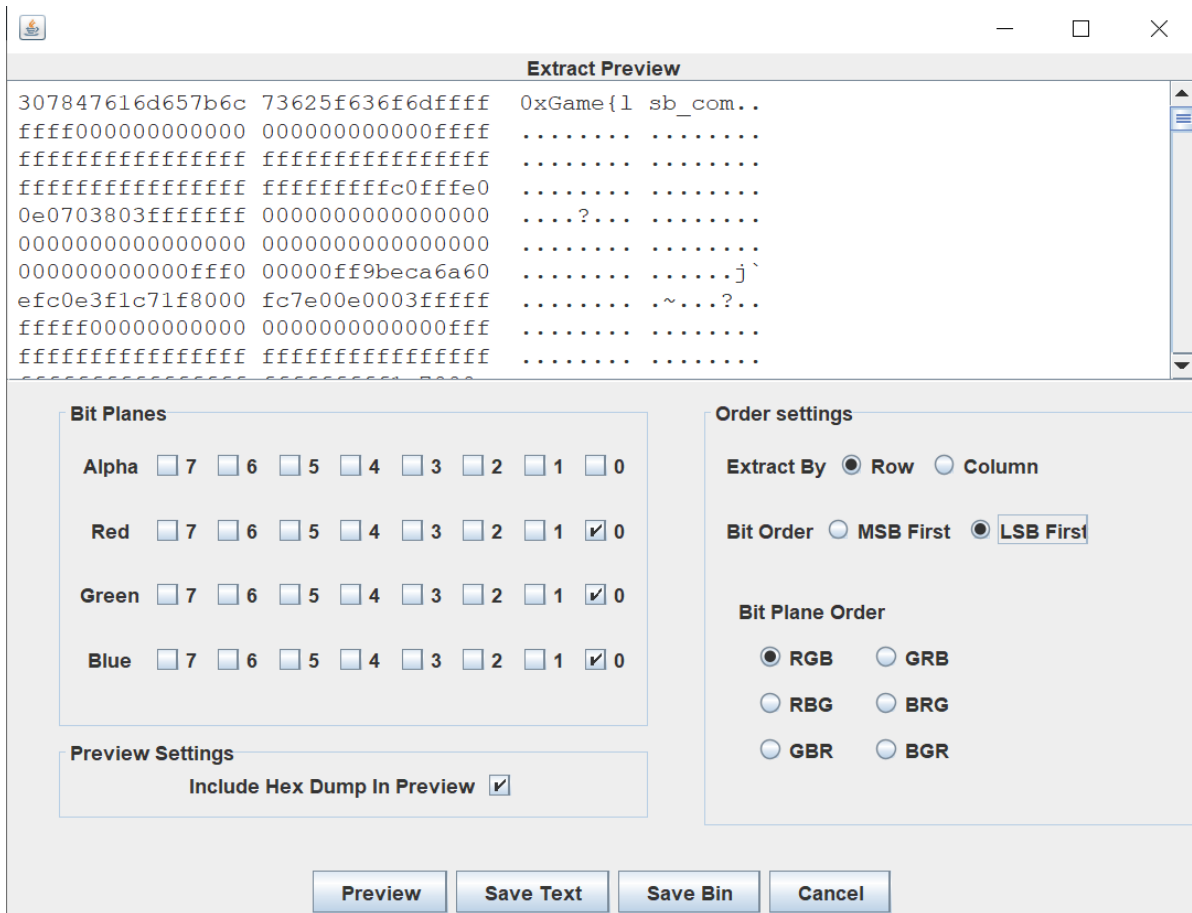
[i] Found passphrase: "07lsbrmw"
[i] Original filename: "flag.txt".
[i] Extracting to "a.jpg.out".

(root@kali)~[~]
#
```

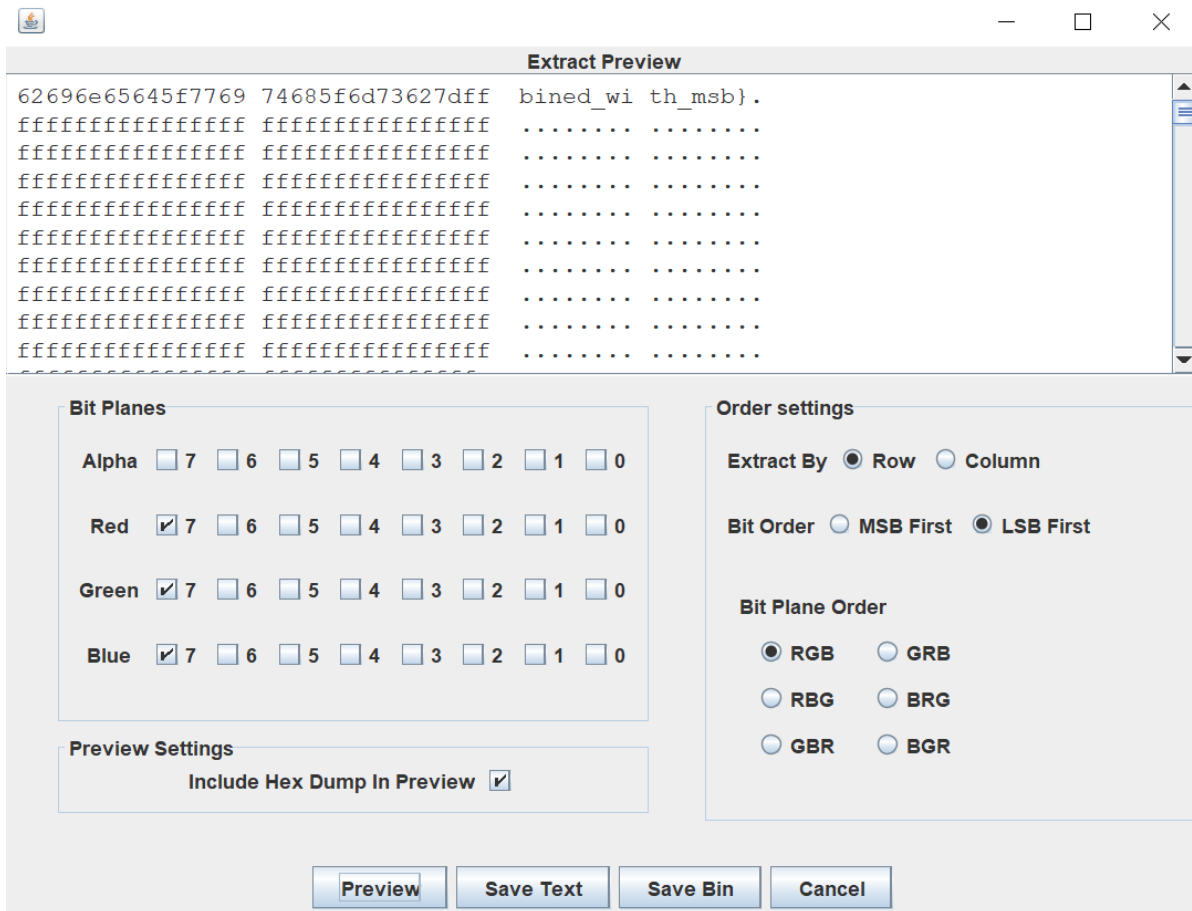


least and most

给了一张图片，多半还是图片隐写。先放到stegsolve中，根据题目least and most，盲猜先提取出RGB的0位平面，果然发现



那么可想而知，再提取出“most”的7位平面



拼接两段得flag

Crypto

密码，觅码，先有*再密

题目

```
from secret import flag #从中导入秘密的flag，这是我们要破解的信息
from Crypto.Util.number import bytes_to_long #从函数库导入一些编码函数
from base64 import b64encode

#hint:也许下列函数库会对你有些帮助，但是要怎么用呢.....
from base64 import b64decode
from gmpy2 import iroot
from Crypto.Util.number import long_to_bytes

flag = flag.encode()
lent = len(flag)
flag = [flag[i*(lent//4):(i+1)*(lent//4)] for i in range(4)] #将flag切割成四份

c1 = bytes_to_long(flag[0])
c2 = ''.join([str(bin(i))[2:] for i in flag[1]])
c3 = b64encode(flag[2])
c4 = flag[3].hex()
print(f'c1?= {pow(c1,5)}\nc2 = {c2}\nc3 = {c3}\nc4 = {c4}')

...

c1?=
26070762378724562657013944088592866603683274155821065086836488347720208878013530
62171214554351749058553609022833985773083200356284531601339221590756213276590896
14389495405390297340763821485116417196863060231384402201613542856008184449935667
2695981757804756591891049233334352061975924028218309004551

c2 =
1001000010000110111010001010011110100011111001001011101010000110111001001011111
101000011110011010000001101011111100110100110001010111111001011010011010000010
11100100101110110010101111001111011100

c3 = b'lueggeeahO+8jOmCo+S5iOW8gOWni+aIkQ=='
c4 = e4bbace79a8443727970746fe68c91e68898e590a72121217d
...

#全是乱码，那咋办嘛？
```

就像拼拼图一样，编个程序将四个变量拼接起来，然后decode()

```
from Crypto.Util.number import *
import base64
from gmpy2 import iroot
a=b""
c1=26070762378724562657013944088592866603683274155821065086836488347720208878013
53062171214554351749058553609022833985773083200356284531601339221590756213276590
89614389495405390297340763821485116417196863060231384402201613542856008184449935
6672695981757804756591891049233334352061975924028218309004551
c1=iroot(c1,5)
print(c1)#根据结果手动给c1赋值
c1=304250111637877466249567661291288030071081703500059143540711
c1=long_to_bytes(c1)
a+=c1
```

```

c2=0b100100001000011011101000101001111010001111100100101110101000011011100100101
11111010000111100110100000011010111111001101001100010101111110010110100110100
000101110010010111101100101011110011110111100
c2=long_to_bytes(c2)
a+=c2
c3 = b'lueggeeahO+8jOmCo+S5iOW8gOWni+aIkQ=='
c3=base64.b64decode(c3)
a+=c3
c4 = 0xe4bbace79a8443727970746fe68c91e68898e590a72121217d
c4=long_to_bytes(c4)
a+=c4
print(a)
print(a.decode())

```

运行输出

```
0xGame{ 恭喜你,已经理解了信息是如何编码的, 那么开始我们的Crypto挑战吧!!!}
```

Take my bag!

题目

```

from Crypto.Util.number import *
from secret import flag

def encrypt(m):
    m = str(bin(m))[2:][::-1]
    enc = 0
    for i in range(len(m)):
        enc += init[i] * int(m[i]) % n
    return enc

w = getPrime(64)
n = getPrime(512)
init = [w*pow(3, i) % n for i in range(512)]

c = encrypt(bytes_to_long(flag))

print(f'w={w}')
print(f'n={n}')
print(f'c={c}')

'''
w=16221818045491479713
n=970207428934876313110217437789988390454858410564104515026976358943129382691334
8632496775173099776917930517270317586740686008539085898910110442820776001061
c=479596928957231459078746799086520554843019092155672287989172110771926282278948
3863742356553249935437004378475661668768893462652103739250038700528111
'''

```

阅读题目, 发现其本质是一个MH背包问题, 于是写出相应的程序求解

```

from Crypto.Util.number import *
w=16221818045491479713
n=970207428934876313110217437789988390454858410564104515026976358943129382691334
8632496775173099776917930517270317586740686008539085898910110442820776001061

```

```

init = [w*pow(3, i) % n for i in range(512)]
c=479596928957231459078746799086520554843019092155672287989172110771926282278948
3863742356553249935437004378475661668768893462652103739250038700528111
a=''
for i in range(511,-1,-1):
    if init[i]<=c:
        c-=init[i]
        a+='1'
    else:
        a+='0'
print(a) #查看一下结果并手动给b赋值
b=0b1100000111100001000111011000010110110101100101011110110101011101100101011011
00011000110011000001101101011001010101111100110010010111110100001101110010011110
01011100000111010000110000010111110100011101000000011011010110010100100001001000
1100100100001001100010010101111101
b=long_to_bytes(b)
print(b)

```

最终得到flag

```
b'\0xGame{Welc0me_2_Crypt0_G@me!#$&%}'
```

BabyRSA

题目

```

from Crypto.Util.number import *
from random import getrandbits
from secret import flag

def getN():
    N = 1
    for i in range(16):
        tmp = getPrime(32)
        N *= tmp
    return N

mask = getrandbits(256)
e = 65537
n = getN()
m = bytes_to_long(flag)
c = pow(m*mask,e,n)
print(f'n = {n}')
print(f'e = {e}')
print(f'c = {c}')
print(f'mask = {mask}')

'''
n =
93099494899964317992000886585964221136368777219322402558083737546844067074234332
564205970300159140111778084916162471993849233358306940868232157447540597
e = 65537
c =
54352122428332145724828674757308827564883974087400720449151348825082737474080849
774814293027988784740602148317713402758353653028988960687525211635107801

```



```
mask =
54257528450885974256117108479579183871895740052660152544049844968621224899247
...
```

经典的RSA

```
import gmpy2
from Crypto.Util.number import *
c =
54352122428332145724828674757308827564883974087400720449151348825082737474080849
774814293027988784740602148317713402758353653028988960687525211635107801
n =
93099494899964317992000886585964221136368777219322402558083737546844067074234332
56420597030015914011177808491616247199384923335830694086823215744754059
E = 0x10001
mask =
54257528450885974256117108479579183871895740052660152544049844968621224899247
data=
[2329990801, 2436711469, 2732757047, 2770441151, 2821163021, 2864469667, 2995527113, 31
11632101, 3162958289, 3267547559, 3281340371, 3479527847, 3561068417, 3978177241, 41347
68233, 4160088337]
phi = 1
for p in data:
    phi = phi * (p-1)
d = gmpy2.invert(E, phi)
m = pow(c, d, n) // mask
print(long_to_bytes(m))
```

运行输出

```
b'\0xGame{Magic_M@th_Make_Crypt0}'
```

Vigenere

本noob认为week1中第二简单的一道题

题目

[Week 1] Vigenere

Hint

密文: 0dGmqk{79ap4i0522g0a67m6i196he52357q60f} 古老而神秘的加密方式?

```
flag{.*}
```

经典的维吉尼亚加密，通过密码表手推出密钥: game, 然后解密得到flag。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

0dGmqk{79ap4i0522g0a67m6i196he52357q60f}

密钥:

解密

0xGame{79ad4e0522a0a67a6e196be52357e60b}

Reverse

不得不说这些题真的中三

数字筑基

题目

欢迎来到数字城市的边缘，这里是进行数字筑基的试炼之地。
在这里，你需要输入神秘符号的16进制形式，才能完成数字筑基。
神秘符号如下：
00110000011110000100001101000001010001100100010101000010010000010100001001000101
请输入神秘符号的16进制形式：|

只要用记事本打开就行子

请输入神秘符号的16进制形式： %s 恭喜，你已经完成了数字筑基！
0xGame{5f4812eb-6dee-46ab-9910-92af643cd911}
抱歉，数字筑基失败。请重试。

根据hint的正规做法：

将题目中所给的二进制数字转化为ASCII文字，输入程序得到flag

☐ 0x / 0b前缀

ASCII文字

0xCAFEBAFE|

十六进制 (字节)

30 78 43 41 46 45 42 41 42 45

二进制 (字节)

00110000011110000100001101000001010001100100010101000010010000010100001
001000101

```
欢迎来到数字城市的边缘，这里是进行数字筑基的试炼之地。
在这里，你需要输入神秘符号的16进制形式，才能完成数字筑基。
神秘符号如下：
00110000011110000100001101000001010001100100010101000010010000010100001001000101
请输入神秘符号的16进制形式：0xCAFEBAFE
恭喜，你已经完成了数字筑基！
0xGame{5f4812eb-6dee-46ab-9910-92af643cd911}
请按任意键继续. . . |
```

代码金丹

这个题我只会用记事本打开了

```
请输入代码金丹: %s 恭喜，你解锁了代码金丹，接触到了赛博空间的真理！
很遗憾，这不是正确的代码金丹。道路还很长，修仙者。
pause 0xGame{620bbfcb-e56f-4e6d-8069-9587e066130a} ?
```

网络元婴

用逆向解析工具IDA打开即可看到flag

