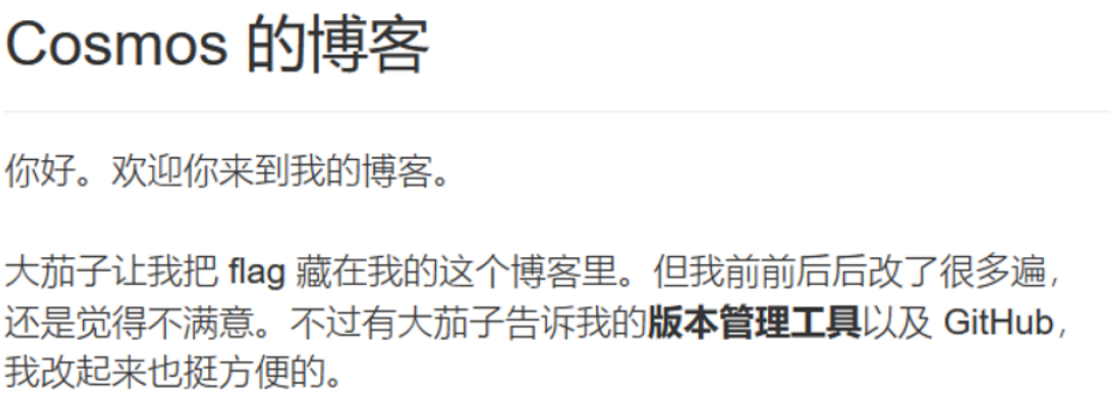
Web

Cosmos的博客



题干中提示版本管理工具和GitHub,不难猜出考的是 .git泄露，flag应该存在历史版本中,所以现在需要找到GitHub项目的地址

```
[*] http://cosmos.hgame.n3ko.co/.git/ is not support Directory Listing
[-] [Skip][First Try] Target is not support Directory Listing
[*] Try to clone with Cache
[*] Initialize Git
[*] Cache files
[*] packed-refs
[*] config
[*] HEAD
[*] COMMIT_EDITMSG
[*] FETCH_HEAD
[*] /refs/heads/master
[*] index
[*] logs/HEAD
[*] refs/heads/master
[*] logs/refs/heads/master
[*] Fetch Commit Objects
[*] objects/05/1894e2ed400a7195008f7022a241e68f5a1335
[*] objects/00/6aca924eaf556120ca6728099ed0233c10a679
[*] objects/1e/b0928962cc436d9e647f32f68018a13656a908
[*] objects/1f/c55fea295446d597542447165f9c57b1c54bfa
[*] objects/d4/c5b9733ca1e27fa7a6cb2e6f24608db7fe4571
[*] objects/ed/3905e0e0c91d4ed7d8aa14412dffeb038745ff
[*] objects/b6/699e80ede013112e3210bd47655438fa57541f
[*] Fetch Commit Objects End
[*] refs/stash
[*] Valid Repository
[+] Valid Repository Success

[+] Clone Success. Dist File : D:\tool\GitHack-master\dist\cosmos.hgame.n3ko.co
```

先用GitHack把.git下载下来

D:\tool\GitHack-master\dist\cosmos.hgame.n3ko.co\.git\ 的索引

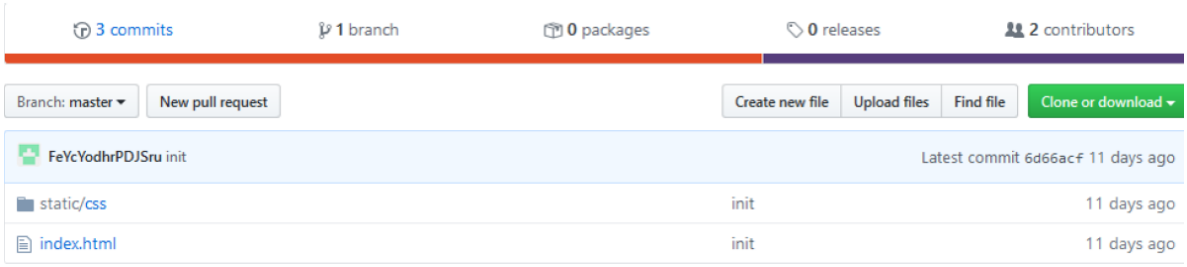
[上级目录]

名称	大小	修改日期
hooks/		2020/1/18 下午8:17:52
info/		2020/1/18 下午8:17:52
logs/		2020/1/18 下午8:17:54
objects/		2020/1/18 下午8:17:53
refs/		2020/1/18 下午8:17:52
COMMIT_EDITMSG	5 B	2020/1/18 下午8:17:52
config	213 B	2020/1/18 下午8:17:52
description	73 B	2020/1/18 下午8:17:52
HEAD	23 B	2020/1/18 下午8:17:52
index	396 B	2020/1/18 下午8:17:54
ORIG_HEAD	41 B	2020/1/18 下午8:17:54

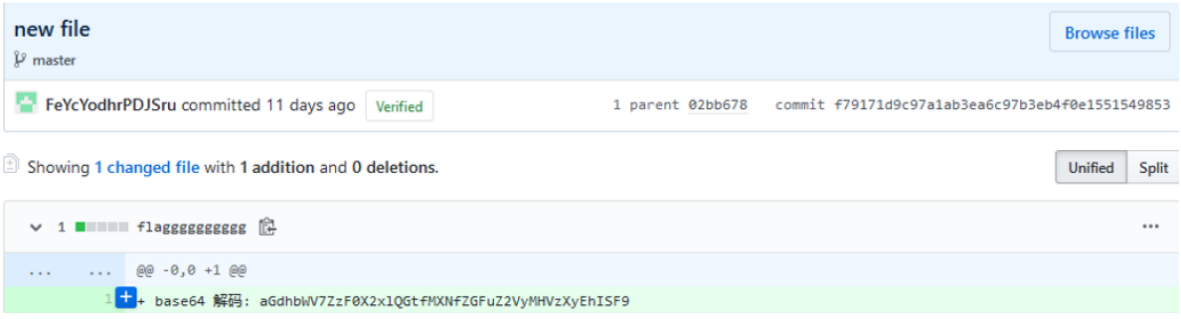
查了一下.git文件的结构结构信息一般存放在config文件中

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "origin"]
    url = https://github.com/FeYcYodhrPDJSru/8LTUKCL83VLhXbc
    fetch = +refs/heads/*:refs/remotes/origin/*
```

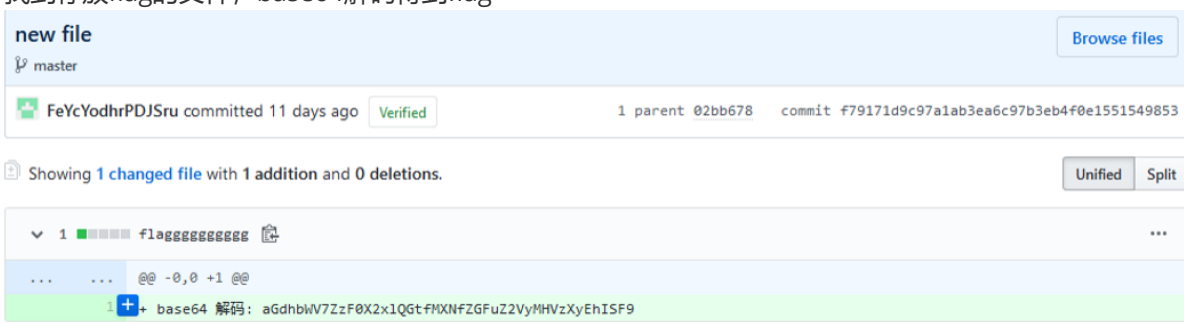
打开发现GitHub的项目地址



再进入commits



找到存放flag的文件，base64解码得到flag



接头霸王

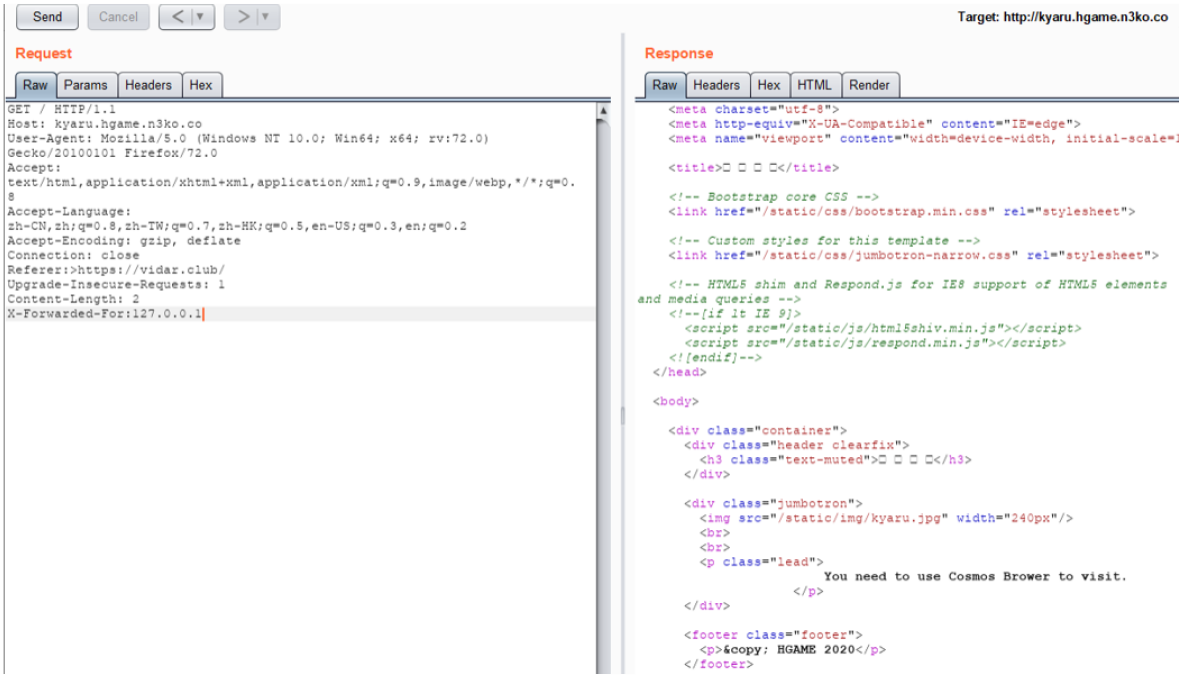
You need to come from https://vidar.club/.

要求从指定网站访问页面，考察Referer来源伪造

```
GET / HTTP/1.1
Host: kyaru.hgame.n3ko.co
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0)
Gecko/20100101 Firefox/72.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Referer:https://vidar.club/
Upgrade-Insecure-Requests: 1
Content-Length: 2
```

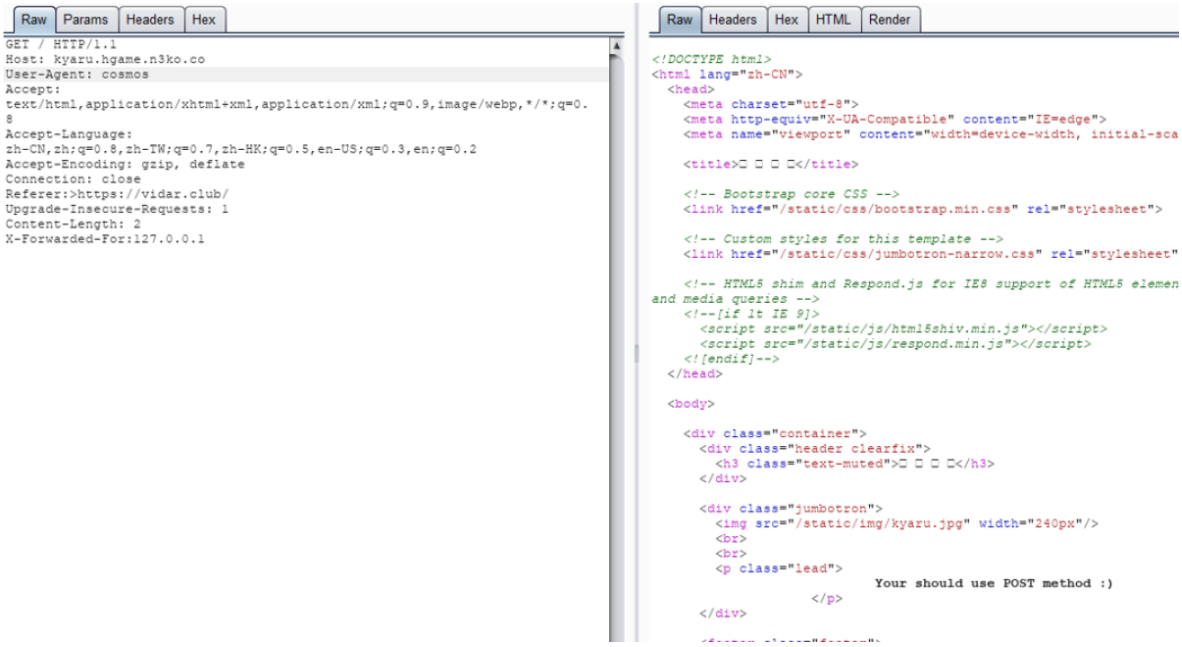
```
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-

    <title>□ □ □ □</title>

    <!-- Bootstrap core CSS -->
    <link href="/static/css/bootstrap.min.css" rel="stylesheet"

    <!-- Custom styles for this template -->
    <link href="/static/css/jumbotron-narrow.css" rel="styleshe

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 ele
and media queries -->
    <!--[if lt IE 9]>
      <script src="/static/js/html5shiv.min.js"></script>
      <script src="/static/js/respond.min.js"></script>
    <![endif]-->
  </head>

  <body>

    <div class="container">
      <div class="header clearfix">
        <h3 class="text-muted">□ □ □ □</h3>
      </div>

      <div class="jumbotron">
        <img src="/static/img/kyaru.jpg" width="240px"/>
        <br>
        <br>
        <p class="lead">
                        You need to visit it locally.
                    </p>
      </div>

      <footer class="footer">
        <p>&copy; HGAME 2020</p>
```

接下来要求从本地访问

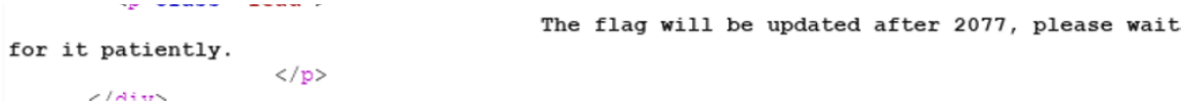


伪造X-Forward-For,接下来要求使用cosmos browers浏览器访问（我查了一下真的有这个浏览器。。。）

考察伪造UA，将UA改为cosmos



要求使用POST方法



最后告诉我们2077年会更新flag，请耐心等待（2077又跳票了XD

Response

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 200 OK
Content-Length: 1231
Content-Type: text/html; charset=UTF-8
Date: Sat, 18 Jan 2020 09:16:20 GMT
Last-Modified: Fri, 01 Jan 2077 00:00:00 GMT
Server: HGAME 2020
Server: Apache/2.4.29 (Ubuntu)
Vary: Accept-Encoding
Connection: close
```

```
If-Unmodified-Since:Tue,06 May 2078 02:42:43 GMT
```

请求头里加一句If-Unmodified-Since，得到flag

```
  <h3 class="text-muted">□ □ □ □</h3>
</div>

<div class="jumbotron">
  <img src="/static/img/kyaru.jpg" width="240px"/>
  <br>
  <br>
  <p class="lead">
                        hgame{W0w!Your_heads_@re_s0_many!}
                </p>
</div>

<footer class="footer">
  <p>&copy; HGAME 2020</p>
```

Code World

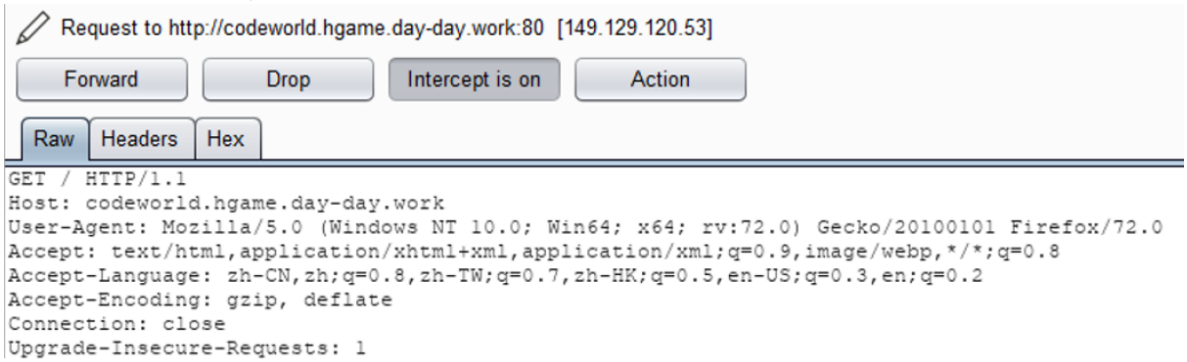
打开后发现403禁止访问，F12看一下源码

403 Forbidden

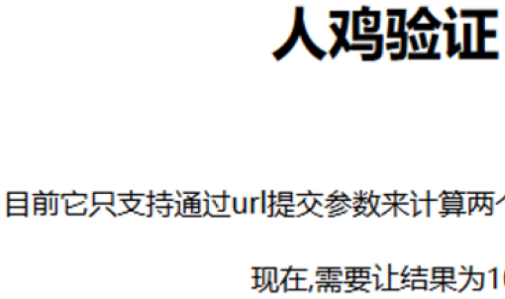
nginx/1.14.0 (Ubuntu)

```
<script>
  console.log("This new site is building....But our stupid developer Cosmos did 302 jump to this page..F**k!")
</script>
```

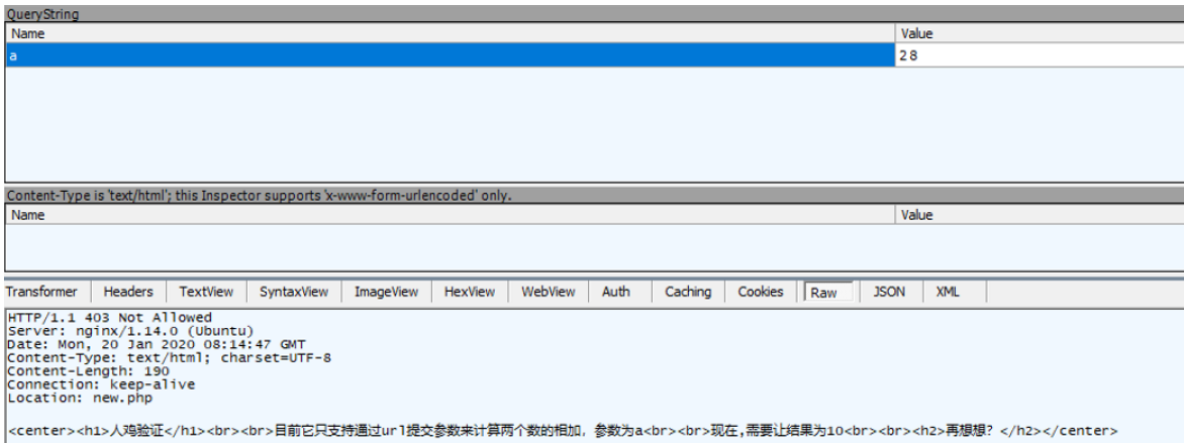
提示302跳转，用burpsuite抓包试试



试一试POST请求



要求通过url提交参数来计算，这一步我卡了好一会，因为不知道url编码，一直在用 + 来提交



突然发现2+8的值是2 8，百度查一下url编码的加号

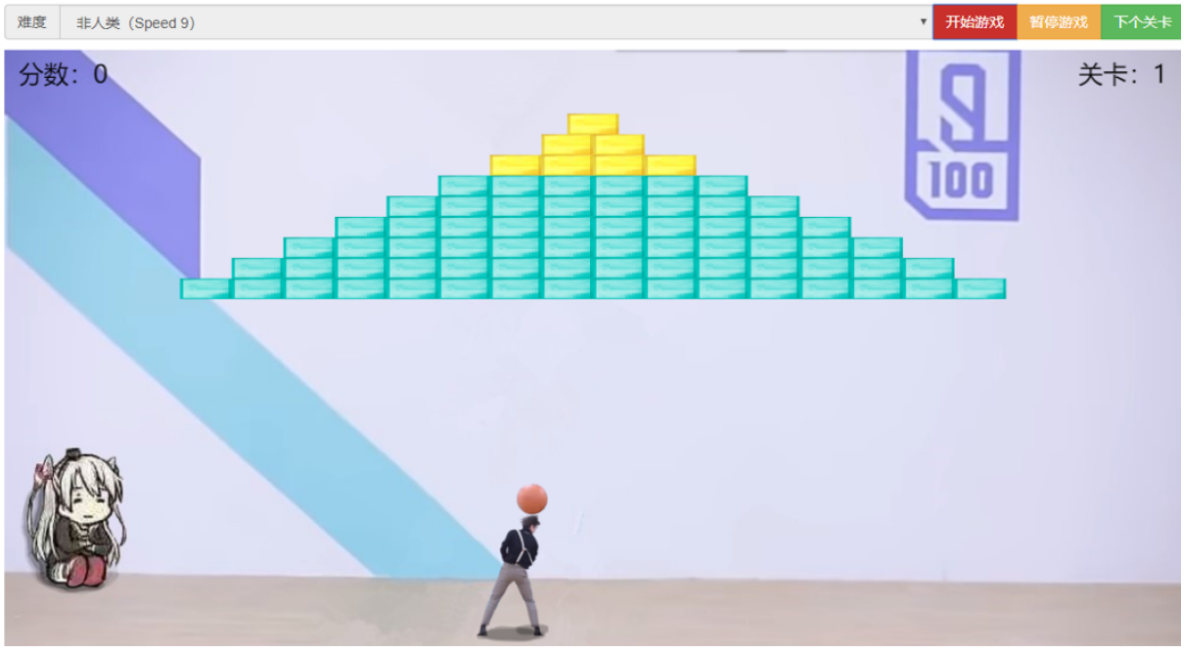
特殊字符	代表含义	替换内容
+	URL中+号表示空格	%2B
空格	URL中的空格可以用+号或者编码	%20
/	分隔目录和子目录	%2F
?	分隔实际的URL和参数	%3F
%	指定特殊字符	25%
#	表示书签	%23
&	URL 中指定的参数间的分隔符	%26
=	URL 中指定参数的值	%3D

“+”改成%2B重新发送POST请求，得到flag

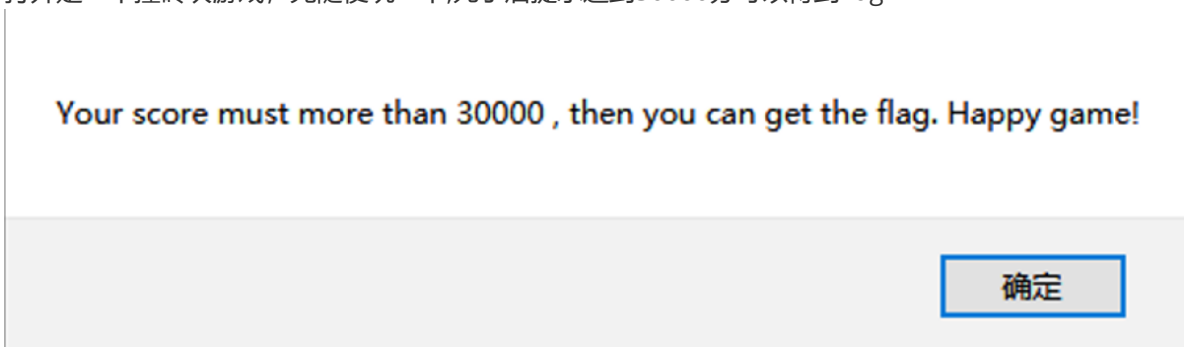
尼泰玫

CXK 打篮球

CXK，出来打球！



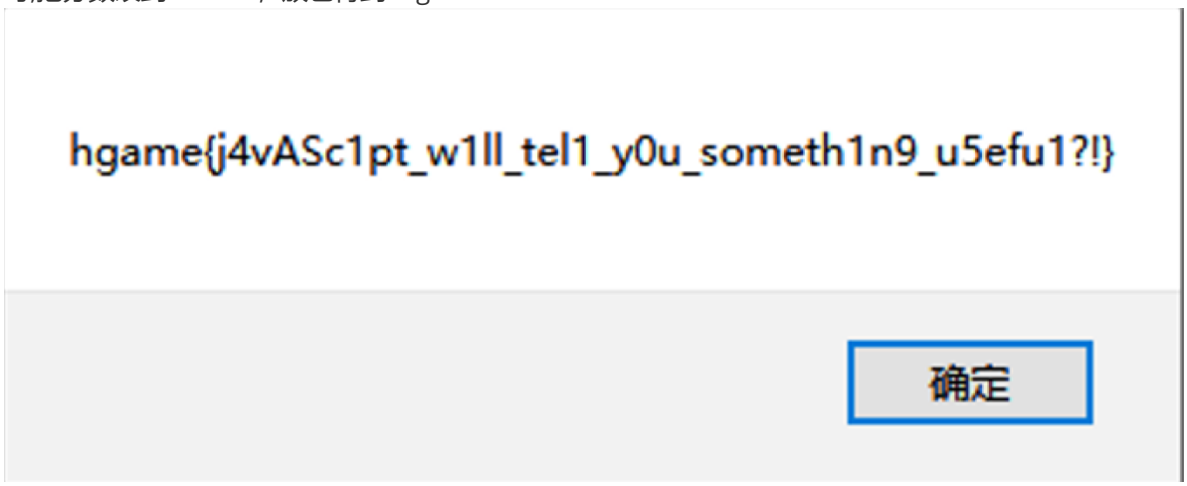
打开是一个撞砖块游戏，先随便玩一下,死了后提示达到30000分可以得到flag



再玩一次,这次在游戏结束的时候用burpsuite抓包

```
Raw | Params | Headers | Hex
POST /submit HTTP/1.1
Host: cxk.hgame.wz22.cc
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 42
Origin: http://cxk.hgame.wz22.cc
Connection: close
Referer: http://cxk.hgame.wz22.cc/
Cookie: __cfduid=dfb4a0738fb2b4294d3b5ff50b916a5001579188285

score=200|63c433d45028ca0f037d32dbf79f9c3c
```

发现score,分数后面一串是防止改包的签名, 但其实只对时间做了签名，并且前后宽限5秒钟,直接改包即可,把分数改到30000+，放包得到flag



Crypto

InfantRSA

描述

真*签到题

p = 681782737450022065655472455411;

q = 675274897132088253519831953441;

e = 13;

c = pow(m,e,p*q) = 27569846508236107014517368841149631154217290260855985901984l

看标题考的应该是RSA加密，百度搜到的一个解密算法

```
def egcd(a, b):
#扩展欧几里德算法
if a == 0:
return (b, 0, 1)
```

```
else:
g, y, x = egcd(b % a, a)
return (g, x - (b // a) * y, y)
def modinv(a, m):
#d=modinv(e,(p-1)*(q-1))
g, x, y = egcd(a, m)
if g != 1:
raise Exception('modular inverse does not exist')
else:
return x % m
p = 681782737450022065655472455411
q = 675274897132088253519831953441
e = 13
c = pow(m, e, p*q)
c = 27569846508236107014517368841149631154217290260855985901984 1
n = p * q
d = modinv(e,(p-1)*(q-1))
m = pow(c,d,n)
print(m)
print("m=\n%s"%m)
#算出M`
print((39062110472669388914389428064087335236334831991333245).to_bytes(1000,
byteorder='big'))
#得到flag
```

Affine

题目给出了加密的算法，以及flag加密后的密文，由于前五个字符hgame已知，根据加密算法可以算出一百以内a,b可能的值（c语言实现，python还不会用QAQ）

```
#include<stdio.h>
int main(void)
{
    int a,b;
    int x,y,z,k,m;
    for(a=1; a<100; a++)
    {
        for(b=1; b<100; b++)
        {
            x=7*a+b;
            y=11*a+b;
            k=6*a+b;
            z=18*a+b;
            m=12*a+b;
            if(x%62==43 && y%62==33 && k%62==30 && z%62==0 && m%62==46)//)
            {
                printf("a=%d,b=%d\n",a,b);
            }
        }
    }
    return 0;
}
```

把13 14带进去，再把明文字典全部加密一遍

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)        #62
cipher = ''             #存放加密后的数据
flag='abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'
A=13
B=14
for b in flag:
    i = TABLE.find(b)   #在TABLE中检索是否包含字符串，返回索引值
    if i == -1:         #不包含子串
        cipher += b     #下划线直接加上去
    else:               #包含子串,返回索引值
        ii = (A*i + B) % MOD
        cipher += TABLE[ii]

print(cipher)
mw=  'IbTazt8AvBfi5wq4QjX1JKF2RkOUZmr7PVduLWgeh6OClsy9xG3YDpnEcoMNHS'
```

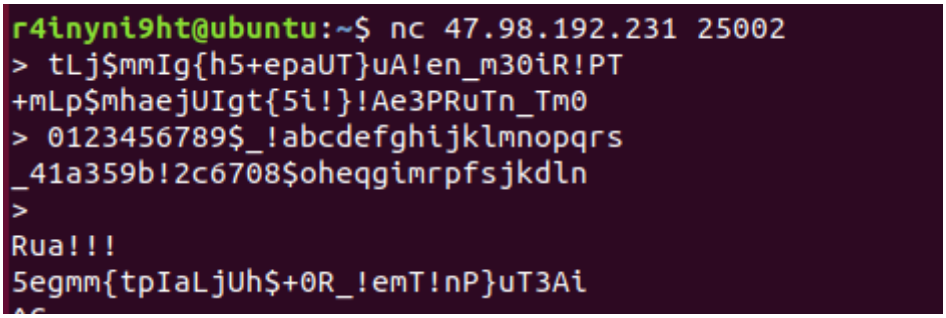
就得到了明文与密文的对应关系，然后把题目给的加密后的flag对应字典翻译得到flag

```
# A8I5z{xr1A_J7ha_vG_TpH41O}
# hgame{M4th_u5Ed_iN_cRYpt0}
```

ps.不会用python所以只能用这种笨方法，后悔没早点学好pythonQAQ

Reorder

先nc连上看看，随便输一些字符，输入都会被加密，仔细看加密后的密文，字符内容没有变，只是顺序改变，看样子是只是重排了明文，那我们只需要找到重排的位移关系就能复原加密后的flag，输入跟flag等长的每个字符都不相同的明文，看加密后的结果，这样就得到了每个位置上字符的位移关系，然后一个个对过去，把加密后的flag归位，得到flag.....（又是没啥技术含量的解法XD



Misc

欢迎参加HGame（签到题）

欢迎大家参加 HGAME 2020!

来来来，签个到吧~

Li0tIC4uLi0tIC4tLi4gLS4tLiAtLS0tLSAtLSAuIC4uLS0uLSAtIC0tLSAuLi0tLi0gLi4tLS0gLS0tLS0gLi4tLS0gLS0tLS0gLi4tLS4tIC4uLi4gLS0uIC4tIC0tIC4uLi0t

注：若解题得到的是无hgame{}字样的flag花括号内内容，请手动添加hgame{}后提交。

【Notice】解出来的字母均为大写

题目地址 https://www.baidu.com

base64解码，得到flag

壁纸

题目下载下来是一张jpg，用Winhex打开看看

Pixiv@纯白可憐.jpg	1,318,549	1,321,183	JPG 文件	2020/1/9 ...	Deflate

Pixiv@纯白可憐.jpg

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
01320720	09	92	28	A6	22	40	28	01	68	01	A6	90	C6	83	C8	14	’(¦"@(h ¦ ÆƒÈ
01320736	C0	74	88	08	F7	A0	64	01	19	A5	58	F2	06	E3	8C	D2	Àtˆ ÷ d ¥Xò ãŒÒ
01320752	B6	A2	7A	1A	10	DB	AA	15	76	3B	9C	0C	0F	41	5A	24	¶¢z Ûª v;œ AZ$
01320768	4B	64	F4	C4	14	00	53	00	A0	02	98	05	00	34	F1	48	KdôÄ S ˜ 4ñH
01320784	40	C3	34	00	C0	A3	AD	30	1C	40	C5	02	23	75	A0	63	@Ã4 À£-0 @Å #u c
01320800	3A	74	A4	00	A7	34	00	F5	3C	53	01	4F	5A	00	5F	EC	:t¤ §4 õ<S OZ _ì
01320816	F1	7F	1B	21	60	A5	46	54	FB	D5	46	9F	B4	25	CF	94	ñ !`¥FTûÕFŸ´%Ï”
01320832	A3	64	3F	D1	C6	EC	1E	70	38	E9	8E	2B	9C	DD	6C	3D	£d?ÑÆì p8éŽ+œÝl=
01320848	E2	8C	8E	50	7E	54	58	77	32	A7	FF	00	58	40	F5	A9	âŒŽP~TXw2§ÿ X@õ©
01320864	14	98	C1	9F	5A	09	0A	06	90	E8	F2	0F	B5	03	88	F5	˜ÁŸZ èò µ ˆõ
01320880	50	5F	14	8A	2D	DB	D8	79	A9	B8	BE	07	A5	52	8D	C4	P_ Š-ÛØy©¸¾ ¥R Ä
01320896	48	61	11	0D	A3	B5	3B	58	64	6E	31	CD	0C	02	37	39	Ha £µ;Xdn1Í 79
01320912	C5	20	2C	AF	4A	A1	0D	EF	4C	60	FF	00	28	26	80	3F	Å ,¯J¡ ïL`ÿ (&€?
01320928	FF	D9	50	4B	03	04	14	00	09	00	08	00	37	8F	29	50	ÿÙPK 7)P
01320944	51	22	B3	CE	50	00	00	00	6C	00	00	00	08	00	00	00	Q"³ÎP l
01320960	66	6C	61	67	2E	74	78	74	C3	7C	21	3D	CC	ED	C8	A7	flag.txtÃ|!=ÌíÈ§
01320976	6A	8D	B5	21	E9	B8	54	CC	DF	CC	C7	F9	62	70	76	41	j µ!é¸TÌßÌÇùbpvA
01320992	57	44	37	8D	87	51	D2	6C	AD	E7	07	30	83	2E	64	90	WD7 ‡QÒl-ç 0ƒ.d
01321008	58	A1	10	45	3A	9C	E1	A5	50	FA	D4	5B	81	6C	42	77	X¡ E:œá¥PúÔ[lBw
01321024	14	BB	0A	B3	E1	AE	9F	7D	8C	2C	90	AB	7D	49	73	87	» ³á®Ÿ}Œ, «}Is‡
01321040	A2	A1	23	51	E9	25	B3	D4	50	4B	01	02	14	00	14	00	¢¡#Qé%³ÔPK
01321056	09	00	08	00	37	8F	29	50	51	22	B3	CE	50	00	00	00	7)PQ"³ÎP
01321072	6C	00	00	00	08	00	24	00	00	00	00	00	00	00	20	00	l $
01321088	00	00	00	00	00	00	66	6C	61	67	2E	74	78	74	0A	00	flag.txt
01321104	20	00	00	00	00	00	01	00	18	00	45	A8	40	3D	D3	C6	E¨@=ÓÆ
01321120	D5	01	45	A8	40	3D	D3	C6	D5	01	E7	B8	9C	1C	D3	C6	Õ E¨@=ÓÆÕ ç¸œ ÓÆ
01321136	D5	01	50	4B	05	06	00	00	00	00	01	00	01	00	5A	00	Õ PK Z
01321152	00	00	76	00	00	00	17	00	50	61	73	73	77	6F	72	64	v Password
01321168	20	69	73	20	70	69	63	74	75	72	65	20	49	44	2E		is picture ID.

发现一个flag.txt，看样子是图片隐写，用foremost分离一下，又得到一个加密的压缩包

00000000.jpg 2020/1/17 1:49 JPG 文件 1,290 KB

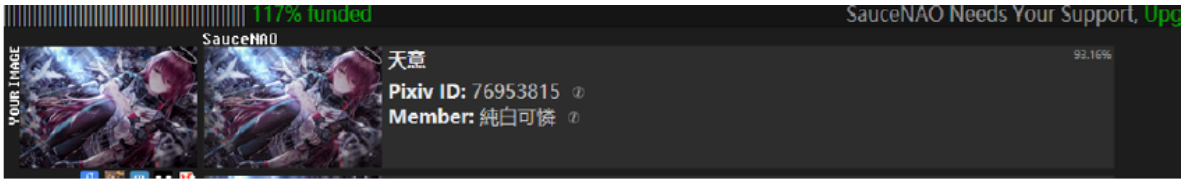
00002579.zip 2020/1/17 1:49 ZIP 压缩文件 21 KB

名称	压缩后大小	原始大小	类型	修改日期	压缩方法	加密
flag.txt*	80	108	TXT 文件	2020/1/9 ...	Deflate	ZipCry

Password is picture ID.

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	48	ÿØÿà JFIF H
00000016	00	48	00	00	FF	E1	00	A0	45	78	69	66	00	00	4D	4D	H ÿá Exif MM
00000032	00	2A	00	00	00	08	00	05	01	1A	00	05	00	00	00	01	*
00000048	00	00	00	4A	01	1B	00	05	00	00	00	01	00	00	00	52	J R
00000064	01	28	00	03	00	00	00	01	00	02	00	00	01	32	00	02	(2
00000080	00	00	00	14	00	00	00	5A	87	69	00	04	00	00	00	01	Z‡i
00000096	00	00	00	6E	00	00	00	00	00	00	00	48	00	00	00	01	n H
00000112	00	00	00	48	00	00	00	01	32	30	32	30	3A	30	31	3A	H 2020:01:
00000128	30	39	20	31	38	3A	31	35	3A	31	38	00	00	03	A0	01	09 18:15:18
00000144	00	03	00	00	00	01	00	01	00	00	A0	02	00	03	00	00	
00000160	00	01	0B	B8	00	00	A0	03	00	03	00	00	00	01	07	E6	¸ æ
00000176	00	00	00	00	00	00	FF	E1	0B	40	68	74	74	70	3A	2F	ÿá @http:/
00000192	2F	6E	73	2E	61	64	6F	62	65	2E	63	6F	6D	2F	78	61	/ns.adobe.com/xa
00000208	70	2F	31	2E	30	2F	00	3C	3F	78	70	61	63	6B	65	74	p/1.0/ <?xpacket
00000224	20	62	65	67	69	6E	3D	22	EF	BB	BF	22	20	69	64	3D	begin="ï»¿" id=
00000240	22	57	35	4D	30	4D	70	43	65	68	69	48	7A	72	65	53	"W5M0MpCehiHzreS
00000256	7A	4E	54	63	7A	6B	63	39	64	22	3F	3E	20	3C	78	3A	zNTczkc9d"?> <x:
00000272	78	6D	70	6D	65	74	61	20	78	6D	6C	6E	73	3A	78	3D	xmpmeta xmlns:x=

再看一下Winhex,发现ID，但是并不是解压密码，那看来ID只能是p站的id了，打开搜p站原图的网站



成功解压，得到flag.txt

flag.txt

```
\u68\u67\u61\u6d\u65\u7b\u44\u6f\u5f\u79\u30\u75\u5f\u4b\u6e\u4f\u57\u5f\u75\u4e\u69\u43\u30\u64\u33\u3f\u7d
```

猜测Unicode，但是在线无法解码，可能经过了处理，看了一下十六进制数都是小于7F的，对照ASCII表看一下

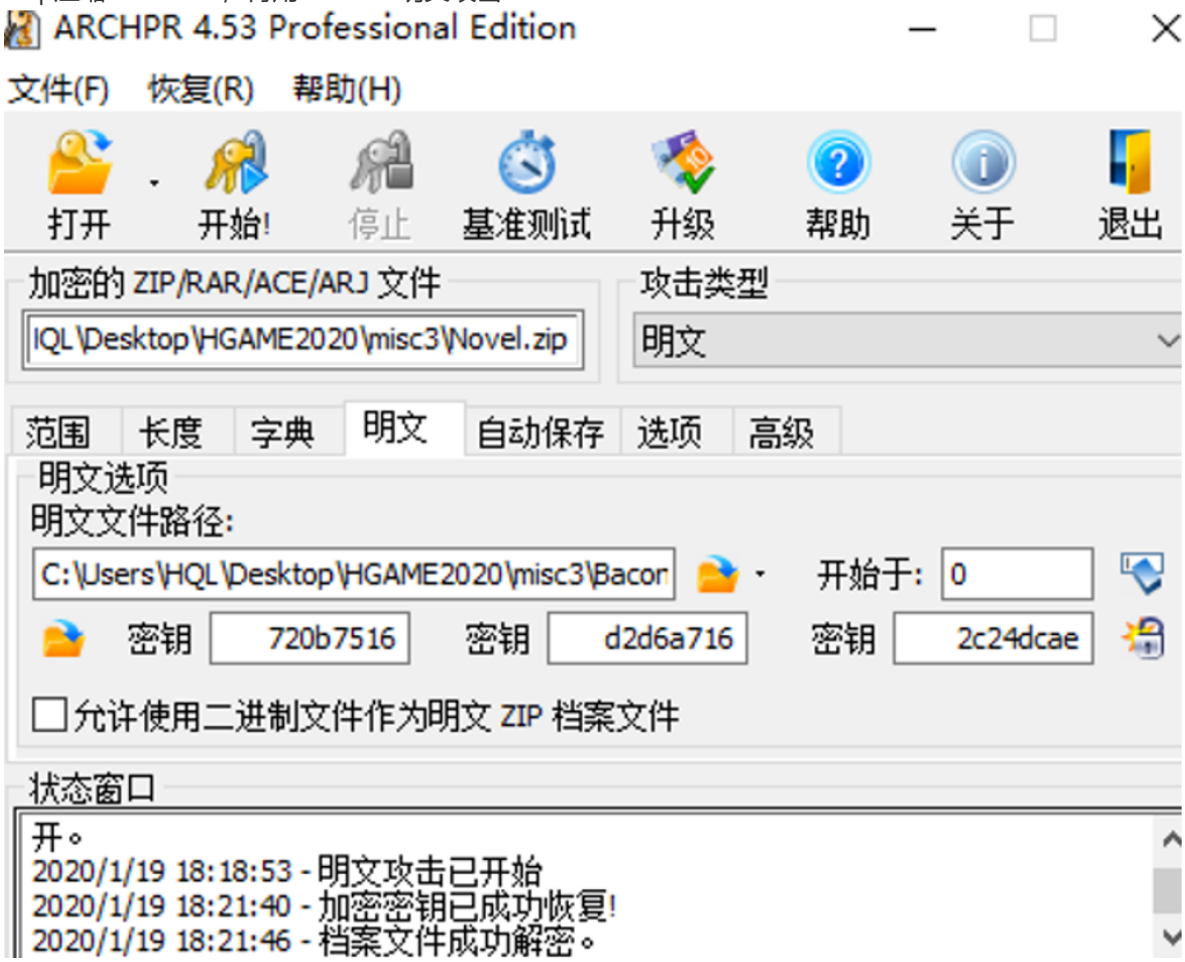
解出来 flag：hgame{Do_y0u_KnOW_uNiC0d3?}

克苏鲁神话

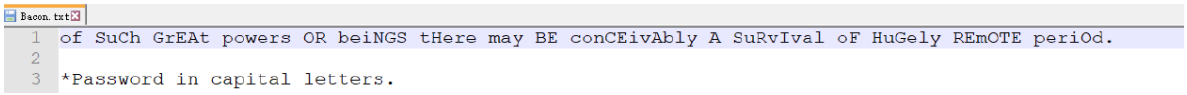
文件下载下来是一个bacon.txt和一个加密的压缩包，压缩包里也有一个Bacon.txt，看样子可以用明文攻击

名称	压缩后大小	原始大小	类型	修改日期	压缩方法	加密方
The Call of Cthulhu.doc*	25,389	28,672	Microsoft W...	2020/1/1...	Deflate	ZipCryp
Bacon.txt*	126	124	TXT 文件	2020/1/1...	Deflate	ZipCryp

7zip压缩Bacon.txt，再用ARCHPR明文攻击



打开word，又是加密过的.....再看看Bacon.txt

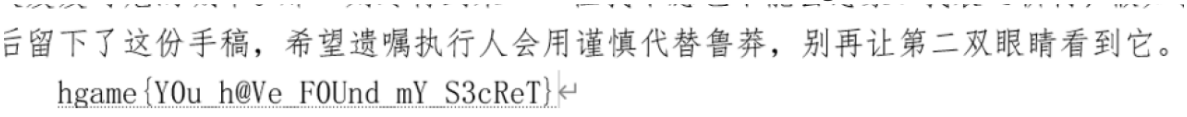


培根密码，小写字母用a代替，大写用b代替，用在线解密工具解密

Bugku|培根密码加解密

```
FLAGHIDDENINDOA
flaghiddenindoa
```

发现密码还是不对，把DOA改成DOC成功打开，显示隐藏文字，得到flag



签到题ProPlus

依旧是一个加密的压缩包和一个Password.txt

Password.txt

```
Rdjxfwxjfimkn□z,ts□wntzi□xtjrwm□xsfjt□jm□ywt□rtntwhf□f□y□□□h□jnsxf□qjFjf□jnb□□rg□fiy

JRFVJYFZVRUAGMAI

* Three fenses first, Five Caesar next. English sentense first,  zip password next.
```

提示先三位一组栅栏密码解密，然后位移五位的凯撒密码解密，得到password：

EAVMUBAQHQMVEPDT

压缩包里一个OK.txt

```
data:text;ook,
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook!
Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook. Ook? Ook!
Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook? Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook! Ook! Ook! Ook!
Ook! Ook! Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook!
Ook! Ook! Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook! Ook? Ook! Ook! Ook. Ook? Ook! Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook! Ook! Ook! Ook! Ook? Ook. Ook? Ook! Ook. Ook? Ook! Ook! Ook! Ook!
```

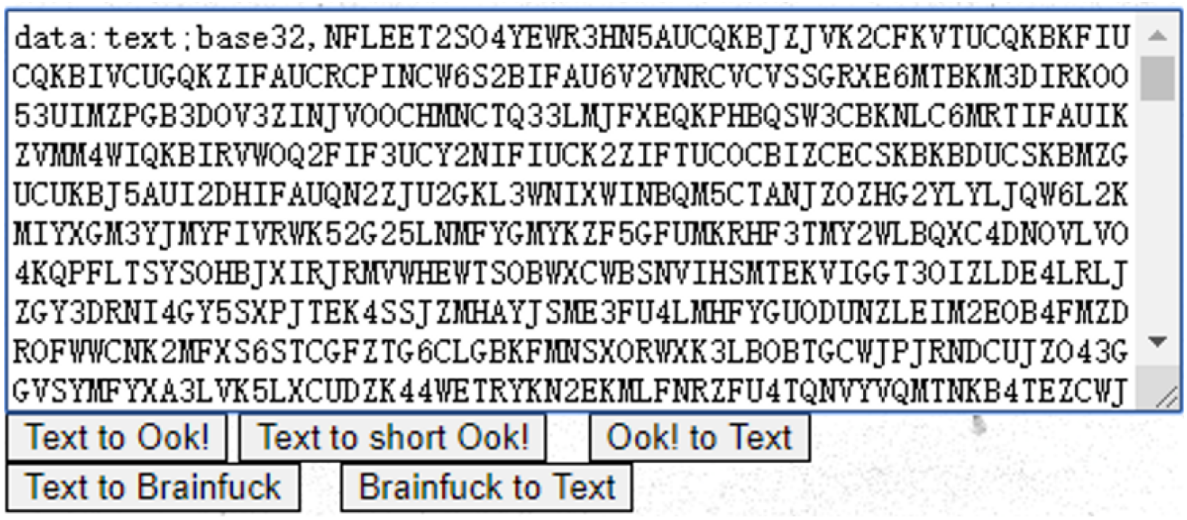
这啥呀，百度一下Ook，发现是Ook语言（命名鬼才.....

brainfuck的更多资料

而另一种类似的**Ook!** 编程语言也偶尔遇到，如下：

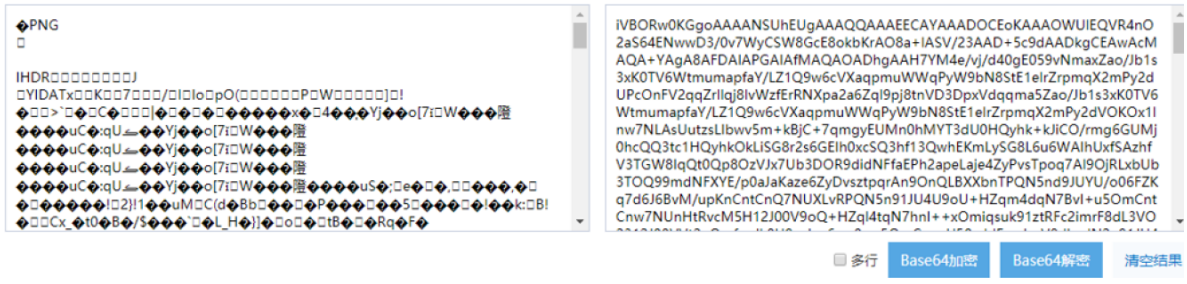
> Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
> Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
> Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook
> Ook! Ook. Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.

在线解码后提示base32解码



u/Xc54U4whAepC+k+wLTXrf8uJ9wJhvAgdSHdB5j2uvXf5YQ7wRAepC6k+wDTXrf+u5xwJxjCg9SFdB9g2uvWf5cT7kRqCLc8p'
ZKunt01ebo85qc3YWd8JSaKXEskKM2R5/X5Owu7ISn1EyJY4EctTn6vCZnd2EnPKVmShwL5KjN0ec1ObsLO+EpNVPiWCBHbY4
r8nZXdgJT6mZEscCOWpz9HlNzu7CTnhKzZQ4FshRm6PPa3J2F3bCU2qmxLFAjtocfV6Ts7uwE55SMyWOBXLU5ujzmpzdhZ3w
opcSyQozZHn9fk3P4SAI4HQwCADxgCAHzAEADgA4YAAB8wBAD4gCEAwAcMAQA+YAgA8AFDAIAP/wAFo0hUZrh1mAAAA
BJRU5ErkJggg==

解密后在结尾发现“==”，应该是base64，在用base64解密一下

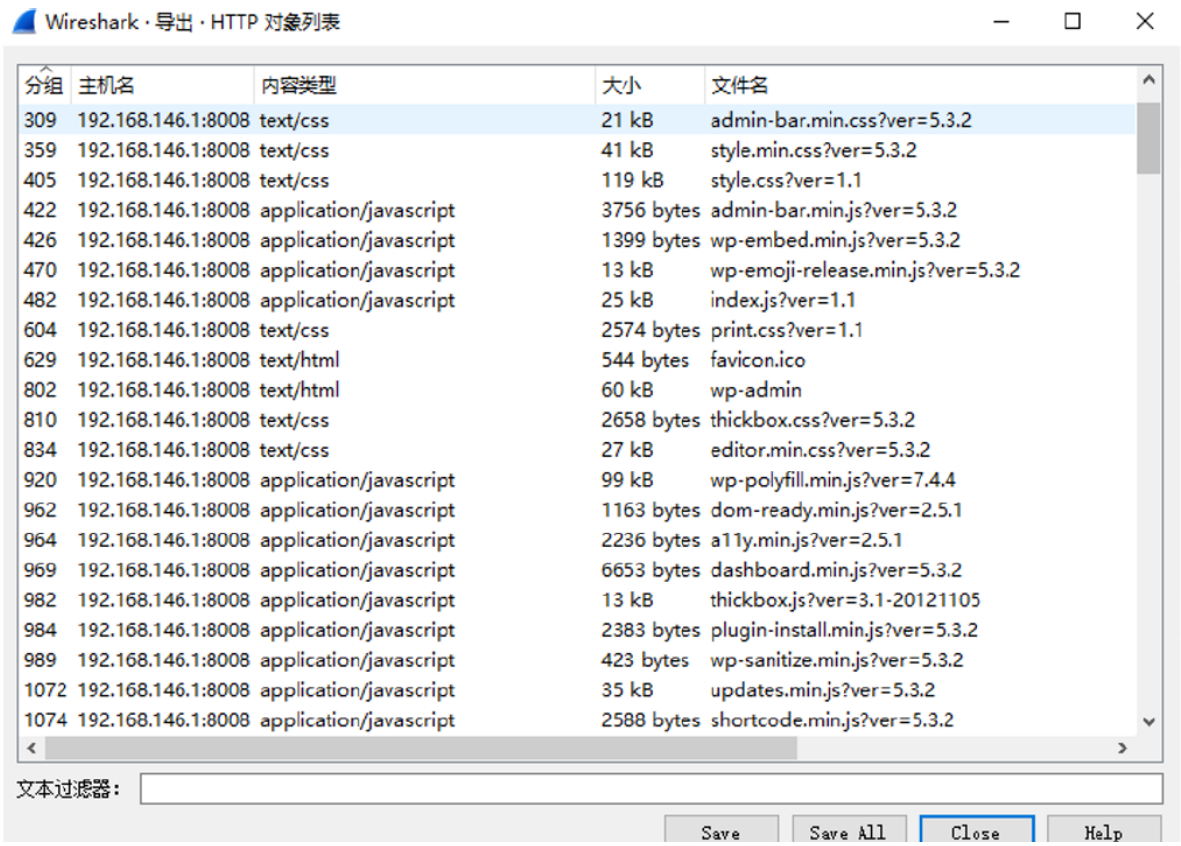


看样子是让我们转成png，在百度无数在线工具均失败后，我选择打开Google，发现一个超好用的工具，不仅能将base64转成不同格式，还能修复base64编码（所以说一定要会用Google XD

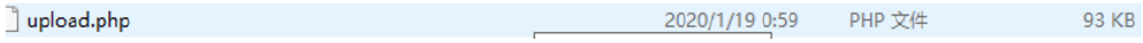
修复后解码得到一个二维码，扫描后得到flag

每日推荐

流量分析题，先拖进Wireshark分析一下，导出对象HTTP



在upload.php中找到了一个song.zip，用foremost分离

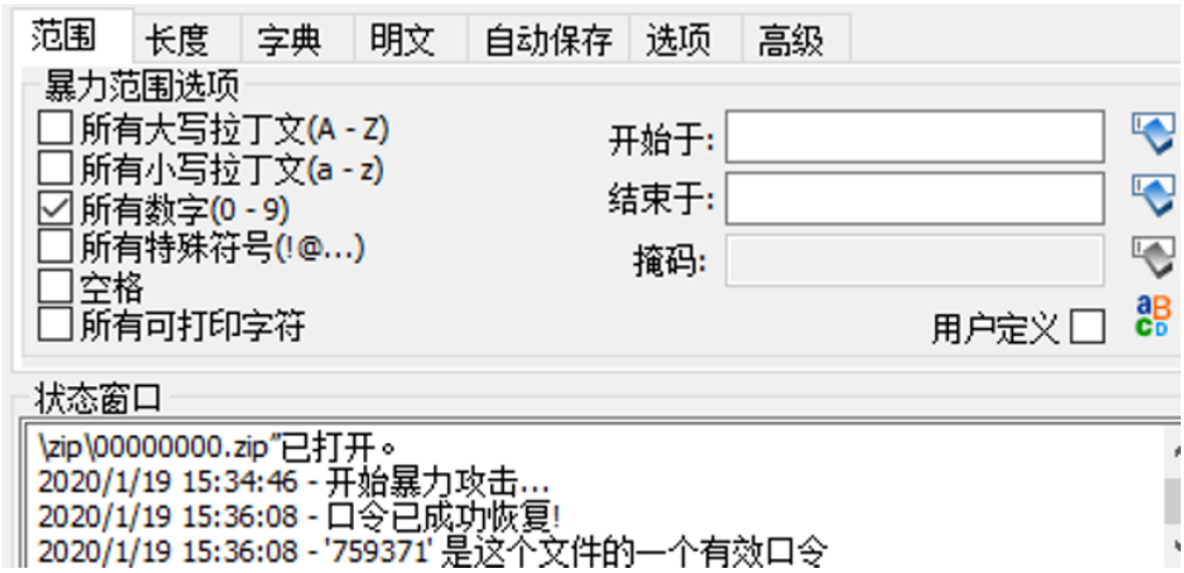



```
------WebKitFormBoundaryGjwmn57vGB5LC1Kb
Content-Disposition: form-data; name="name"

song.zip
------WebKitFormBoundaryGjwmn57vGB5LC1Kb
Content-Disposition: form-data; name="action"

upload-attachment
------WebKitFormBoundaryGjwmn57vGB5LC1Kb
Content-Disposition: form-data; name="_wpnonce"

8b76e8452a
------WebKitFormBoundaryGjwmn57vGB5LC1Kb
Content-Disposition: form-data; name="async-upload"; filename="song.zip"
Content-Type: application/x-zip-compressed
```

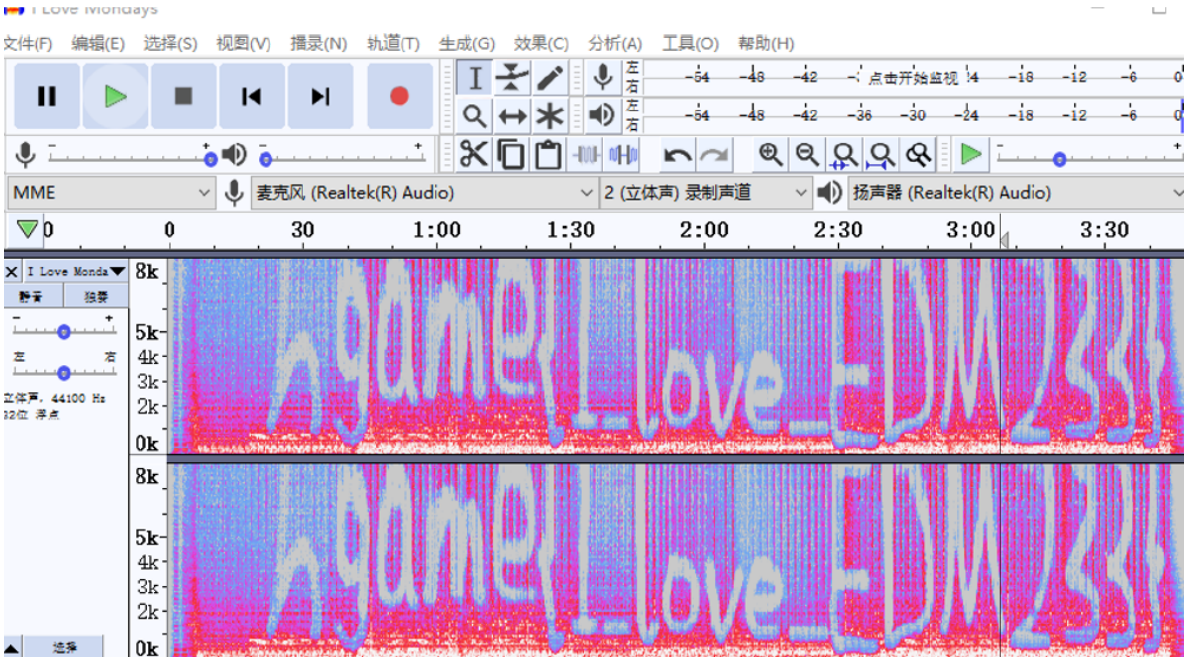
得到一个压缩包，打开提示密码为6位数字，直接爆破

名称	压缩后大小	原始大小	类型	修改日期
I Love Mondays.mp3*	8,289,673	9,388,953	cloudmusic.mp3	2020/1/13 16:

密码为6位数字



打开是一个音频文件，应该是音频隐写，拖进Audacity分析



看一下频谱图，得到flag