H GAME Week 2 Write Up

CRYPTO

浪漫的足球圣地

一看到标题，显然了，曼彻斯特密码了。

```
966A969596A9965996999565A5A59696A5A6A59A9699A599A596A595A599A569A5A99699A56996A596A696A996A6A5A696A9A595969AA5A69696A5A99696A595A59AA56A96A9A5A9969AA59A9559
```

百度一查，曼彻斯特加密有两种规则。

第一种G. E. Thomas

第二种IEEE 802.4（令牌总线）和低速版的IEEE 802.3（以太网）中规定

这里的是第二种（第一种试过了，错的）

- 编码0101（0x5）表示11；
- 编码1001（0x9）表示01；
- 编码0110（0x6）表示10；
- 编码1010（0xA）表示00；

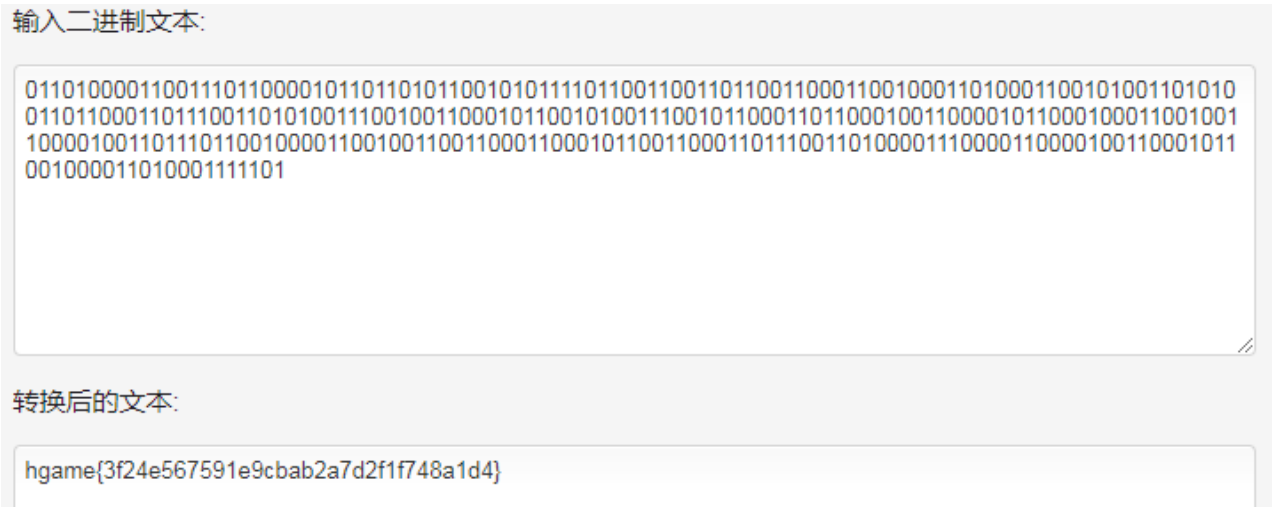
  那么一个简单C语言的脚本就够了，

```
#include <stdio.h>
#include <string.h>

int main()
{
  char
a[]="966A969596A9965996999565A5A59696A5A6A59A9699A599A596A595A599A569A5A99699A56996A596A696A996A6A5A696A9A595969AA5A69696A5A99696A595A59AA56A96A9A5A9969AA59A9559"
;
    for(int i=0;i<strlen(a);i++)
    {
        if(a[i]=='5')printf("11");
        else if(a[i]=='6')printf("10");
        else if(a[i]=='9')printf("01");
        else if(a[i]=='A')printf("00");

    }
      return 0;
}
```

  拿到二进制数，在线二进制转ASCII码，



拿到flag:hgame{3f24e567591e9cbab2a7d2f1f748a1d4}

hill

题目说的很明显了，希尔加密，并且给了明文**BABYSHILL**，密文为**TCSHXZTCXAPBDKJVJDOHJEAE**

希尔加密规则为，密匙*明文=密文（mod 26），这个**（mod 26）**很重要，有用到数学的**模逆元**的知识。

（学长对着我这榆木脑袋讲了好久我才理解2333麻烦了）明文为9个字符，密文有24字符，所以有16种明文对应密文的可能，有16种密匙，（并不是每一个算出来的“密匙”都可以成为密匙的，因为有的密匙的逆矩阵不能进行模逆运算，也就不能成为密匙）学长本意应该是让我们写脚本爆破，无奈编程技术欠佳（好吧，不想编，有点恐惧，害怕debug到半夜）于是选择了手动爆破。最终找出的对应关系为**明文BABYSHILL**对应**密文HXZTCXAPB**，然后用该密匙的逆矩阵解密密文，

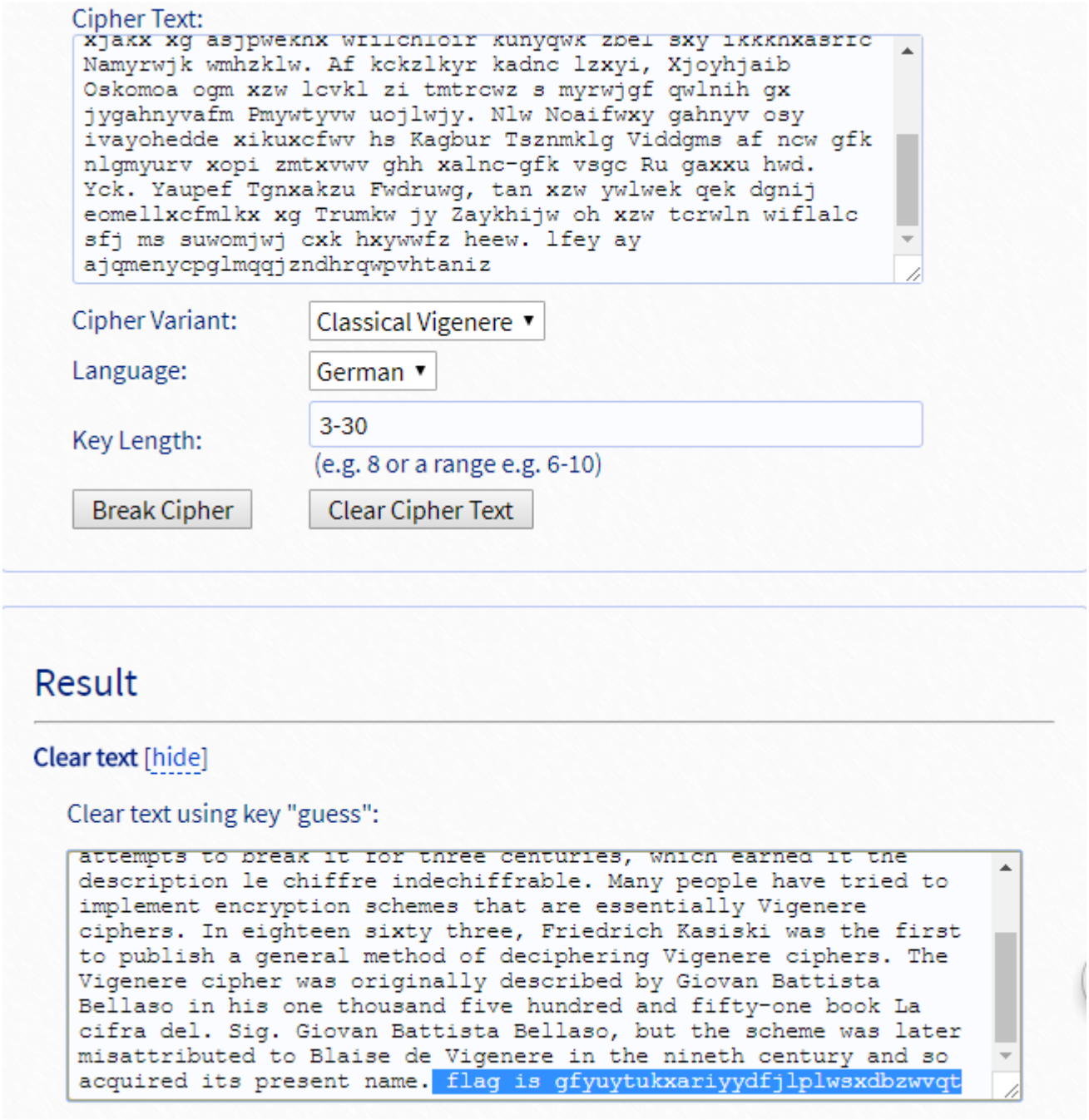
拿到flag:hgame{THEBABYSHILLCIPHERATTACK}

Vigener~

（好恨当初没第一个写这题，唯一一个可能一血的题目呵~~）

```
Zbi Namyrwjk wmhzk cw s eknlgv uz ifuxstlata edhnufwlow xwpz vc mkohk s kklmwk uz
mflklagnkh Gswyuv uavbijk, huwwv uh xzw ryxlwxm sx s qycogxx. Ml ay u jgjs ij
hgrsedhnufwlow wmtynmlmzcsf. Lny gahnyv ak kuwq lu orvwxmxsfj urv asjpwekhx, tmz cx
jwycwlwj upd szniehzm xg txyec az zsj lnliw ukhxmjoyw, ozowl wsxhiv az nlw vkmgjavnmgf
ry gzalzvw atxiuzozjjshfi. Ests twgvfi zsby xjakx xg asjpwekhx wfilchloir kunyqwk zbel
sxy ikkkhxasrfc Namyrwjk wmhzklw. Af kckzlkyr kadnc lzxyi, Xjoyhjaib Oskomoa ogm xzw
lcvkl zi tmtrcwz s myrwjgf qwlnih gx jygahnyvafm Pmywtyvw uojlwjy. Nlw Noaifwxy gahnyv
osy ivayohedde xikuxcfwv hs Kagbur Tsznmklg Viddgms af ncw gfk nlgmyurv xopi zmtxvwv
ghh xalnc-gfk vsgc Ru gaxxu hwd. Yck. Yaupef Tgnxakzu Fwdruwg, tan xzw ywlwek qek
dgnij eomellxcfmlkx xg Trumkw jy Zaykhijw oh xzw tcrwln wiflalc sfj ms suwomjwj cxk
hxywwfz heew. lfey ay ajqmenycpglmqqjzndhrqwpvhtaniz
```

一拿到题目，有点慌，慌得我下意识直接百度有没有在线解密，呀，还真有，



拿到flag：hgame{gfyuytukxariyydfjlplwsxdbzwvqt}（要不是提醒过有的flag是脸滚键盘.... ）

（后来一看，Vigener密码是凯撒密码的拓展，感觉像是多出了个栅栏密码的样子，解密要先确定密匙长度，还要进行频度分析，balabala....要真让我写脚本，怕是凉凉~ ）

MISC

Are You Familiar with DNS Records?

（No，i'm not familiar with DNS Records)

好一个送分题.....可我找了半天，才终于了解到，原来DNS的记录的信息，是可以在线查的，

拿到flag：hgame{seems_like_you_are_familiar_with_dns}

快到火炉旁找个位置坐坐！

（炉石传说？！这很misc。卡组代码原来还有这么高深学问的。）面向百度，拿到一张炉石卡组代码的字节数组分析。

```
byte[43] {
0,                +---------->   placeholder
1,                |---------->   version always 1
2,                |---------->   FormatType
1,                |---------->   Num Heroes + always 1
7,                |---------->   HeroId
2,                +---------->   numSingleCards
175, 4,         + |
145, 188, 2,    + +------------> single card part
14,               +---------^   numDoubleCards
28,           +
176, 2,       |
145, 3,       |
255, 3,       |
142, 5,       |
168, 5,       |
212, 5,       |
164, 6,       |  +------------>  double cards part
238, 6,       |
231, 7,       |
239, 7,       |
130, 176, 2,  |
136, 176, 2,  |
185, 191, 2,  +
0                 +------------>  multi cards (more than 2) count
}
```

将

AAECAf0EBu0FuAju9gLQwQIMigGcAq4DyQOrBMsE5gSYxALaxQKW5AK0/ALSiQOmmAMA

解码为

\x00\x01\x02\x01\xfd\x04\x06\xed\x05\xb8\x08\xee\xf6\x02\xd0\xc1\x02\x0c\x8a\x01\x9c\x02\xae\x03\xc9\x03\xab\x04\xcb\x04\xe6\x04\x98\xc4\x02\xda\xc5\x02\x96\xe4\x02\xb4\xfc\x02\xd2\x89\x03\xa6\x98\x03\x00

然后一一对应上图，排列为

\x00

\x01

\x02

\x01

\xfd\x04 //英雄ID（法师）

\x06 // 单张卡牌数量

\xed\x05

\xb8\x08

\xee\xf6\x02

\xd0\xc1\x02

\x0c // 两张卡牌数量

\x8a\x01

\x9c\x02

\xae\x03

\xc9\x03

\xab\x04

\xcb\x04

\xe6\x04

\x98\xc4\x02

\xda\xc5\x02

\x96\xe4\x02

\xb4\xfc\x02

\xd2\x89\x03

\xa6\x98\x03

\x00

显然，卡牌数量出现了问题，分别更改为**\x04** 和 **\x0d**，然后根据数组的递增关系，不难看出，

\xee\xf6\x02

\xd0\xc1\x02

处位置发生了颠倒；

调整后，在线编码，拿到flag

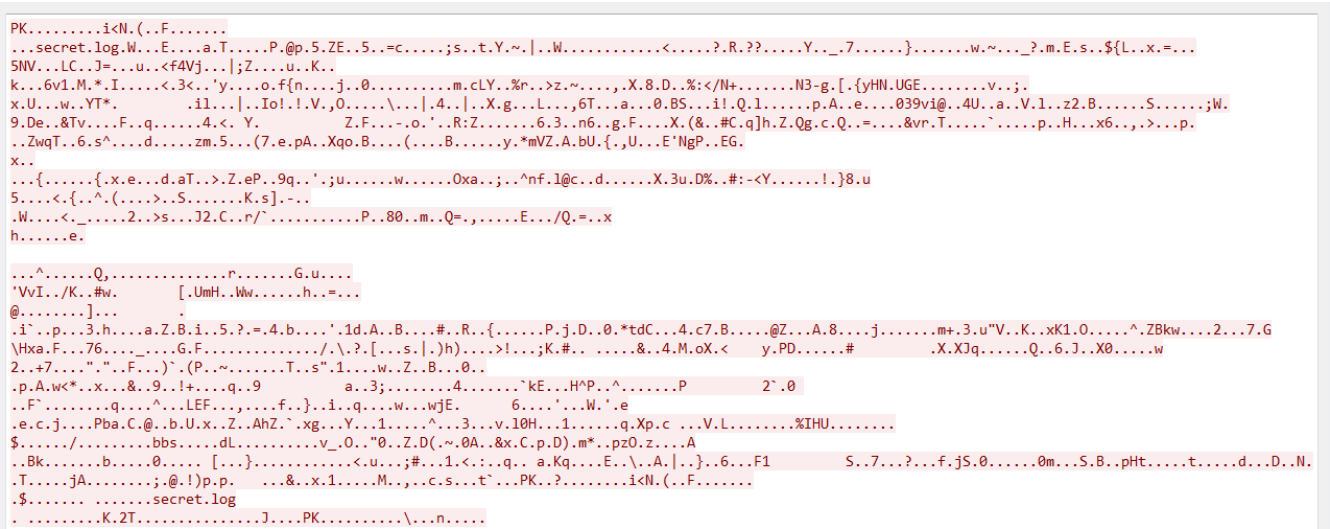
hgame{AAECAf0EBO0FuAjQwQLu9gINigGcAq4DyQOrBMsE5gSYxALaxQKW5AK0/ALSiQOmmAMA}

找得到我嘛？小火汁

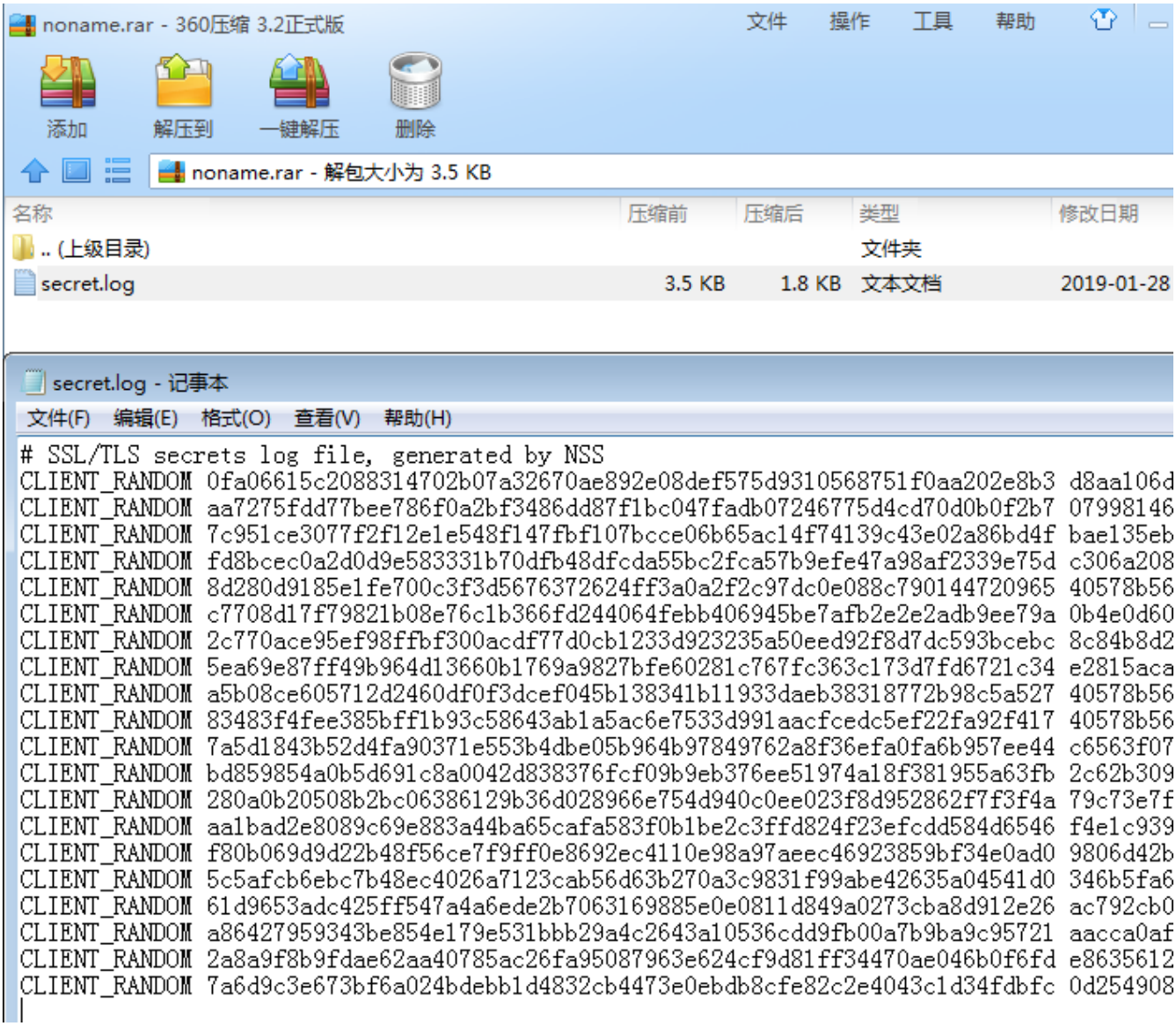
流量包？放wireshark！套路过滤HTTP，跟踪流（明明才做第二题流量包，就套路了，呵呵。。）拉到最下面，只有一句

```
<h1><HaHaHa!flag is very safe now!!!/h1>
```

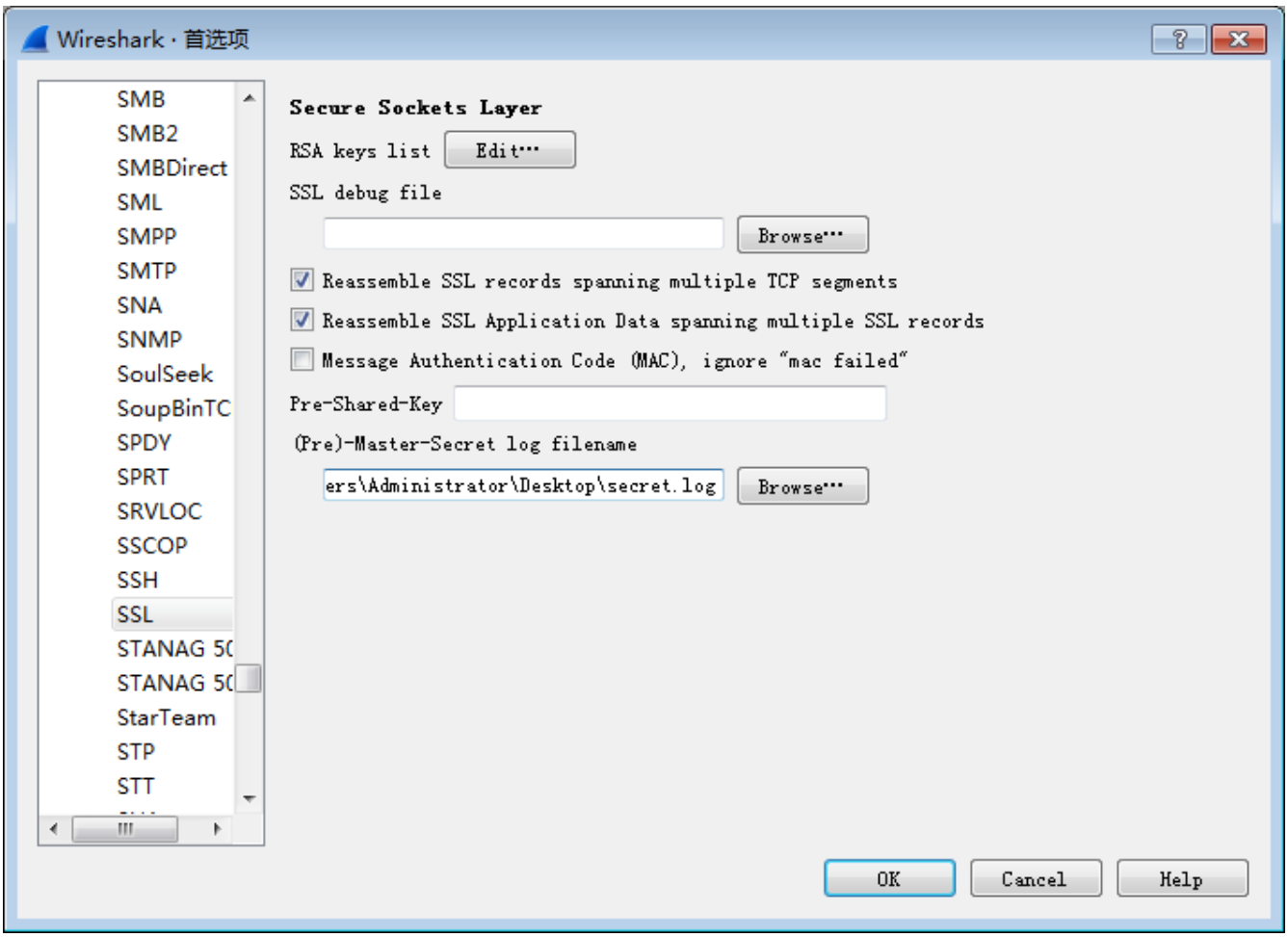
呵呵。那就试试跟踪TCP流，真多呢，看了好多，没看出啥来，直到跟踪到TCP流10，



PK！这玩儿意有点熟悉啊，转成原始数据，复制到winhex里，保存为rar，打开，发现一个文件，打开，是一堆SSL会话密匙，**HTTPS**的**S**就体现出来了呢。



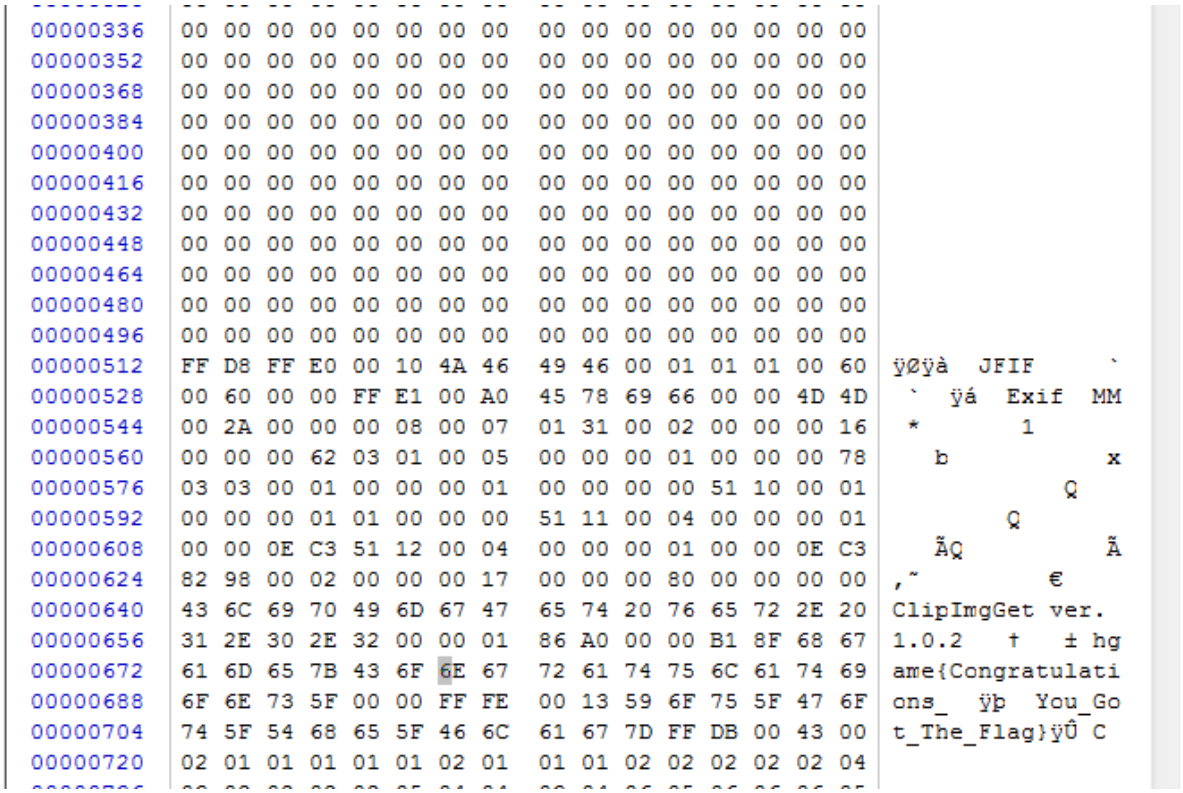
然后将其导入wireshark，



然后导出HTTP对象中新出现的1.tar

里面有个未知格式的文件，用记事本凑合着打开吧。。

哇，瞎了，扛不住，用winhex试一试？



明显很多，但还是要自己组，要不是这句话挺连贯的，还以为中间还有啥呢。。。

所以最终flag：hgame{Congratulations_You_Got_The_Flag}

（所以正确打开方式是啥，，，？）

初识二维码

打开文件，**data:image/png;base64**,在线base转图片！

拿到一张二维码

一眼打过去，显然少了点啥，是三个定位标识符！

结合提示，破损的二维码，是少了三个定位标识符么？那就PS加上去呗，最终得到一张图，

”破损的二维码也能扫的出来“，这才意识到，组合后的才是**破损的二维码**，

最后扫描

拿到flag：hgame{Qu1ck_ReSp0nse_cODe}

RE

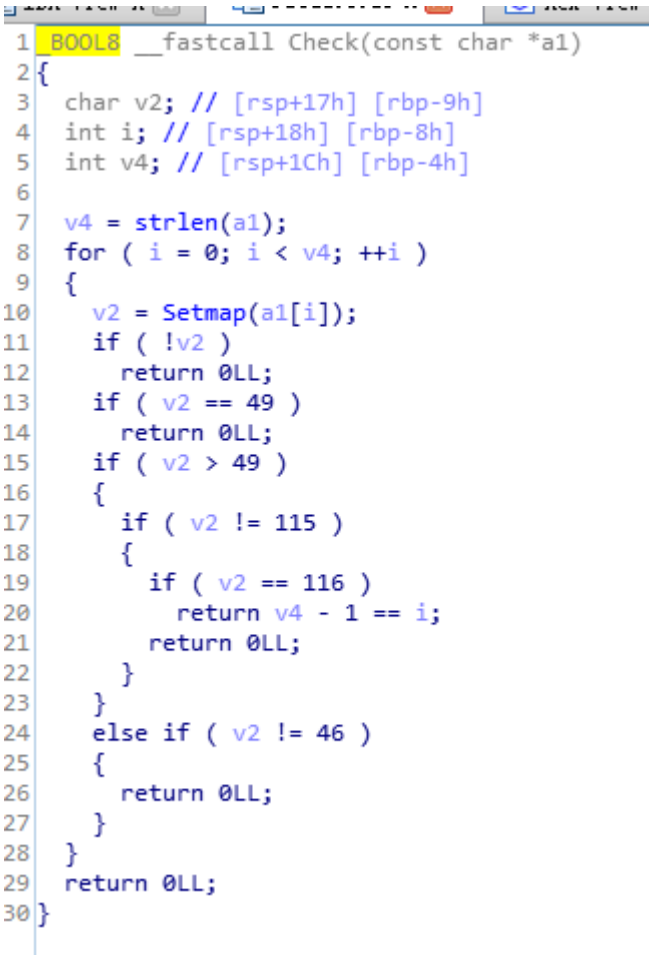
maze

拿到题目，放进IDA。

IDA View-A | Pseudocode-A | Hex View-1 | Structures | Enums | Imports | Exports

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    char v4; // [rsp+0h] [rbp-D0h]
4    unsigned __int64 v5; // [rsp+C8h] [rbp-8h]
5
6    v5 = __readfsqword(0x28u);                    //
7                                                  //
8    puts(
9      "Before finishing this problem\n"
10     "I recommend you to read\n"
11     "<One Hundred Years of Solitude> <The City in History: Its Origins, Its Transformations, and Its Pros> <the Death and"
12     " Life of Great American Cities>");
13   sleep(5u);
14   puts("Have you finished reading? Let's submit the flag:");
15   __isoc99_scanf("%s", &v4);
16   if ( (unsigned int)Check(&v4) )
17     puts("Congratulations, you are a qualified Zhou Dynasty's fan.");
18   else
19     puts("Wrong flag! You are a fake fan of Zhou Dyasty");
20   return 0;
21 }
```

没什么特别的，点开check函数，



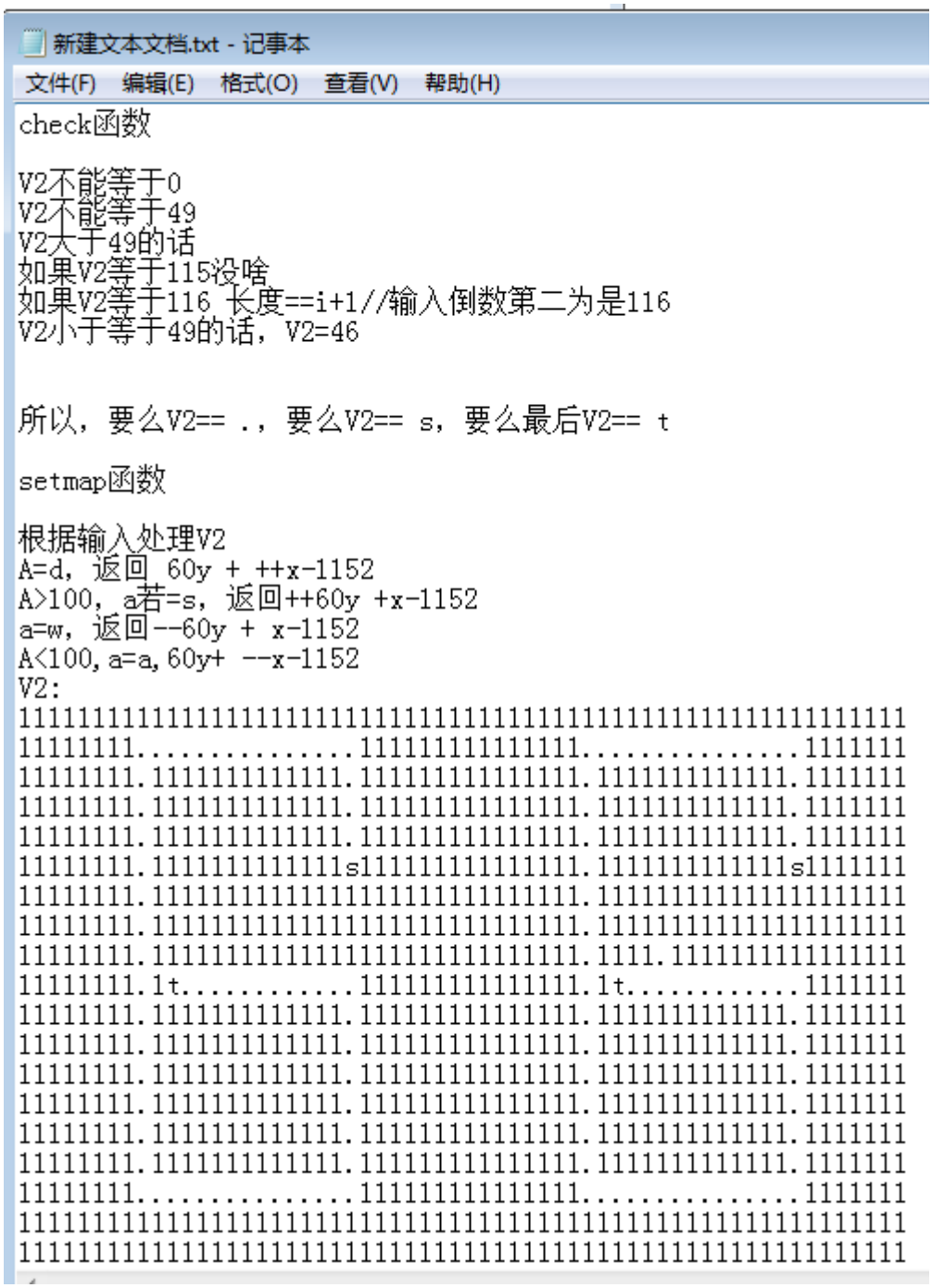
```
1  _BOOL8 __fastcall Check(const char *a1)
2  {
3    char v2; // [rsp+17h] [rbp-9h]
4    int i; // [rsp+18h] [rbp-8h]
5    int v4; // [rsp+1Ch] [rbp-4h]
6
7    v4 = strlen(a1);
8    for ( i = 0; i < v4; ++i )
9    {
10     v2 = Setmap(a1[i]);
11     if ( !v2 )
12       return 0LL;
13     if ( v2 == 49 )
14       return 0LL;
15     if ( v2 > 49 )
16     {
17       if ( v2 != 115 )
18       {
19         if ( v2 == 116 )
20           return v4 - 1 == i;
21         return 0LL;
22       }
23     }
24     else if ( v2 != 46 )
25     {
26       return 0LL;
27     }
28   }
29   return 0LL;
30 }
```

是一个布尔函数。接着点开setmap函数

```
    &v2,
    "11111111111111111111111111111111111111111111111111111111111111111111...............111111111111111...............111"
    "111111111111.1111111111111.111111111111111.1111111111111.111111111111111.1111111111111.111111111111111.1111111111111"
    ".111111111111111.1111111111111.111111111111111.1111111111111.111111111111111.1111111111111s111111111111111.111111111"
    "1111s111111111111111.11111111111111111111111111111.11111111111111111111111111111.11111111111111111111111111111.11111"
    "111111111111111111111111.11111111111111111111111111111.1111.111111111111111111111111.1t............111111111111111.1"
    "t............111111111111111.1111111111111.111111111111111.1111111111111.111111111111111.1111111111111.1111111111111"
    "11.1111111111111.111111111111111.1111111111111.111111111111111.1111111111111.111111111111111.1111111111111.111111111"
    "111111.1111111111111.111111111111111.1111111111111.111111111111111.1111111111111.111111111111111.1111111111111.11111"
    "1111111111.1111111111111.111111111111111...............111111111111111...............1111111111111111111111111111111"
    "111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111",
    0x474uLL);
  if ( a1 == 100 )
  {
    if ( y > 17 )
      result = 0LL;
    else
      result = (unsigned __int8)s[60 * y - 1152 + ++x];
  }
  else if ( a1 > 100 )
  {
    if ( a1 == 115 )
    {
      if ( x > 58 )
        result = 0LL;
      else
        result = (unsigned __int8)s[60 * ++y - 1152 + x];
    }
    else
    {
      if ( a1 != 119 )
        return 0LL;
      if ( y <= 0 )
        result = 0LL;
      else
        result = (unsigned __int8)s[60 * --y - 1152 + x];
    }
  }
  else
  {
    if ( a1 != 97 )
      return 0LL;
    if ( x <= 0 )
      result = 0LL;
    else
      result = (unsigned __int8)s[60 * y - 1152 + --x];
  }
```

也不难读懂。并发现一个**60**这样的倍数，试着将setmap函数中v2数组的值排列成一个60列的矩阵，

得到下图



一个迷宫？不看题了，来玩吧，根据V2最后是t，那么起点就是s了，所以最终玩出

flag：hgame{wwwwaaaaaaaaaaaaaasssssssssssssssddddddddddddddwwwwwwwwaaaaaaaaaaaa}

Pro的Python教室(二)

拿到题目.pyc文件？这怎么玩？原来百度就能找到在线反编译Python的工具。

逆向后拿到代码：

```
#!/usr/bin/env python
# encoding: utf-8
print "Welcome to Processor's Python Classroom Part 2!\n"
print "Now let's start the origin of Python!\n"
print 'Plz Input Your Flag:\n'
```

```
enc = raw_input()
len = len(enc)
enc1 = []
enc2 = ''
aaa = 'ioOavquaDb}x2ha4[~ifqZaujQ#'
for i in range(len):
    if i % 2 == 0:
        enc1.append(chr(ord(enc[i]) + 1))
        continue
    enc1.append(chr(ord(enc[i]) + 2))

s1 = []
for x in range(3):
    for i in range(len):
        if (i + x) % 3 == 0:
            s1.append(enc1[i])
            continue

enc2 = enc2.join(s1)
if enc2 in aaa:
    print "You 're Right!"
else:
    print "You're Wrong!"
    exit(0)
print "Welcome to Processor's Python Classroom Part 2!\n"
print "Now let's start the origin of Python!\n"
print 'Plz Input Your Flag:\n'
enc = raw_input()
len = len(enc)
enc1 = []
enc2 = ''
aaa = 'ioOavquaDb}x2ha4[~ifqZaujQ#'
for i in range(len):
    if i % 2 == 0:
        enc1.append(chr(ord(enc[i]) + 1))
        continue
    enc1.append(chr(ord(enc[i]) + 2))

s1 = []
for x in range(3):
    for i in range(len):
        if (i + x) % 3 == 0:
            s1.append(enc1[i])
            continue

enc2 = enc2.join(s1)
if enc2 in aaa:
    print "You 're Right!"
else:
    print "You're Wrong!"
    exit(0)

```

思路很简单，就是对一个字符串，第奇数个字符，右移一位，第偶数个字符右移两位，然后是一个“栅栏”加密，之所以打引号，因为它不是真的栅栏，（ABCDEF栅栏分三栏后是AD BE CF,这个程序是分成AD CF BE).

所以逆向的思路也很简单，先把**'ioOavquaDb}x2ha4[~ifqZaujQ#'**反“栅栏”，再把所有字符按坐标左移一位或两位。

C语言代码如下（Python还是不太会。。）

```
#include <stdio.h>
#include <string.h>

int main()
{
    char s[]="ioOavquaDb}x2ha4[~ifqZaujQ#",r[27];
    for(int i=0;i<strlen(s);i++)
    {
        if(0<=i&&i<=8)r[3*i]=s[i];
        else if(9<=i&&i<=17)r[3*(i-8)-1]=s[i];
        else if(18<=i&&i<=26)r[3*(i-18)+1]=s[i];
    }
    for(int i=0;i<strlen(r);i++)
    {
        if(i%2==0)r[i]-=1;
        else r[i]-=2;
        printf("%c",r[i]);
    }
      return 0;
}
```

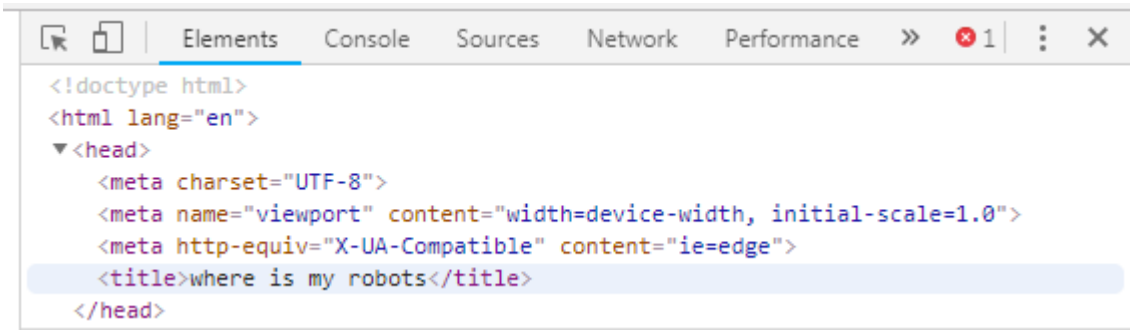
最终拿到flag：hgame{Now_Y0u_got_th3_PYC!}

WEB

easy_php

打开网站：come on ! second wait you

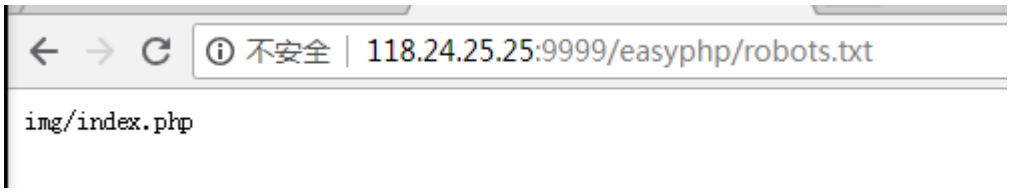
嗯？然后呢，就莫得给我随便点点的东西嘛？那么只能f12瞧瞧了。



where is my robots ?

那就想到机器人协议了，就讲地址栏改为：

```
http://118.24.25.25:9999/easyphp/robots.txt
```



emmm，那再将地址栏改为

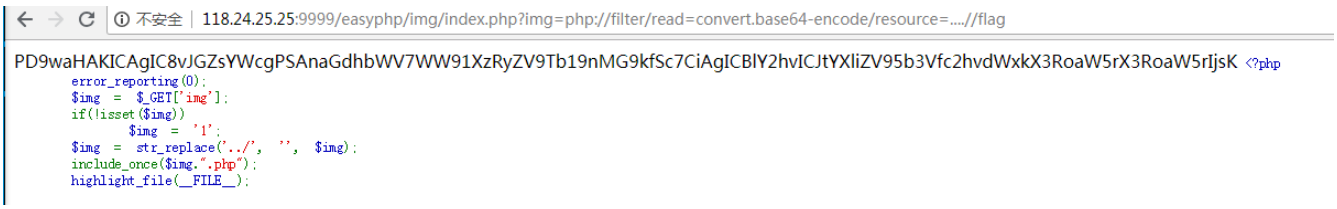
```
http://118.24.25.25:9999/easyphp/img/index.php
```

看一下这“easy”的代码，GET **img**的值，过滤**../**，然后包含并执行，于是想到php伪协议。

所以最终payload为：

```
http://118.24.25.25:9999/easyphp/img/index.php?img=php://filter/read=convert.base64-encode/resource=....//flag
```

考虑到**../**会被过滤，于是键入**....//**，这样就不怕liao。



将上面那一行base64在线解码，

明文:

```
<?php
    //$flag = 'hgame{You_4re_So_g0od}';
    echo "maybe_you_should_think_think";
```

拿到flag：hgame{You_4re_So_g0od}

php trick

打开网页，啊，还行，就是头有点晕23333.

step1，step2：str1和str2不能相等但str1和str2的md5加密后相等，根据php的弱类型比较，只要md5加密后是0e开头的值，都试为相等，所以网上找了俩，分别是：**str1=QNKCDZO；str2=s878926199a。**

step3，step4：str3和str4不能相等但str3和str4的md5加密后相等，这次不类型不弱了，是“！==”，根据md5加密的特性，md5不能加密数组，否则返回false，但程序继续运行，于是只要把str3和str4定义为不相等的数组就好，分别是：**str3[]=QNKCDZO；str4[]=s878926199a。**

step5，step6，step7，step8：首先地址栏“？”后不能出现H_game，然后str5的值不能是数字，其次str5的值要大于9999999999，然后字符串型的str5的值要小于等于0。strp5的绕过，同week1的very easy web一样，只要将H_game或这种一个字符url encode一次就好，至于大于9999999999，我用了一次科学计数法就能绕过，字符串的值小于等于0，把H_game设为数组就能绕过，此时很神奇的也绕过了数字类型判断，所以最终：**H_gam%65[]=1e0**，

step9，step10：直接**url=http：//**www.baidu.com，就好了（真的吗？）

接下来的代码，是要在url的这个网址读取html文本，可百度里有啥好读的

这个时候，我才发现代码最上面有一个admin.php，于是我，把这个题又开了一个网页，直接输入/admin.php，出现的是：**only localhost can see it**，那么这里大概就要绕到localhost去看admin.php了，于是更改url为：**url=**http://@127.0.0.1:80@www.baidu.com/admin.php，这样就绕过了step9，step10，但读的还是127.0.0.1:80/admin.php，回车，

```php
//flag.php
if($_SERVER['REMOTE_ADDR'] != '127.0.0.1') {
    die('only localhost can see it');
}
$filename = $_GET['filename']??'';

if (file_exists($filename)) {
    echo "sorry,you can't see it";
}
else{
    echo file_get_contents($filename);
}
highlight_file(__FILE__);
?>
```

额。还有啊，这题真长啊，代码的意思是，文件存在，就不给你读，不存在，就输出文件的内容。啊？这啥？这是什么鬼套路。百度了半天找不到绕过file_exists（）的方法，最后得知file_get_contents（）属于文件包含的一种（不太确定，应该是吧）然后就可以利用php伪协议（上一题用过了，怎么还用啊），于是试着，**?filename=php://filter/read=convert.base64-encode/resource=flag.php**，得到**PD9waHAgJGZsYWcgPSBoZ2FtZXtUaEVyNF9BcjRfczBtNF9QaHBfVHIxY2tzfSA/Pgo=**

在线base64解码，得到flag：hgame{ThEr4_Ar4_s0m4_Php_Tr1cks}

（啊，这么长，扛不住啊。。。）

PHP Is The Best Language

打开网页，先看到//include 'secret'。然后，emmm，hash_hmac加密，从php.net了解到，hash_hmc有四个参数，分别是alog，data，key，raw_output，这里只有三个，相比第四个不必要？然后data参数不能为数组，否则返回null。于是利用这点，设door为数组类型：**door[]=123**；这样secret的值就为null了，就不用管原来secret是啥值了，接着gate被post的"key"的值用sha256加密，那先看key把。

首先，通过var_dum知道，key用md5加密后为string类型，加1，会被强制转为int类型，（这样就好比多了）让md5('key'])+1== md5(md5('key'))，跑个代码就得，代码如下

```
<?php
for($i=0;$i<=50;$i++)
{
    if ((md5($i)+1) == (md5(md5($i)))+1)
    echo $i;
}
?>
```

运行结果（没想到有那么多）：0,12,14,39,42,49,50

emmm，随便挑一个吧，就决定是49了，这样就是**key=49**

然后回到gate的，

```
<?php
    echo $gate = hash_hmac('sha256', 49, null);
?>
```

运行得到：**c721973cfc184fdc207682888a2357f6a295f46c16221a30f3379f9dc8106334**

所以最后的payload是：?
door[]=123&key=49&gate=c721973cfc184fdc207682888a2357f6a295f46c16221a30f3379f9dc8106334

（因为电脑坏了，让学长帮忙post的23333，最后学长出题人给到flag：hgame{Php_MayBe_Not_Safe}