2019年CTF5月比赛记录（一）：ISCC_2019线上赛部分题目writeup

原创

極品一═☆宏 于 2019-05-25 08:54:59 发布 2400 收藏 11

分类专栏： 2019年CTF比赛—5月赛 CTF_web

本文链接：https://blog.csdn.net/qq_43214809/article/details/90273936


2019年CTF比赛—5月赛 同时被 2 个专栏收录

1 篇文章 0 订阅

订阅专栏


CTF_web

13 篇文章 0 订阅

订阅专栏

ISCC战线拉得蛮长的，从5月1号一直干到25号。我之前也没参加过ISCC，就第一次参加做题情况来看，除了第一天结束后排名在前100，之后就再也没往上走过，尼玛一直掉啊，每天退步一点点，垂直下降，心疼啊。我一个web手竟然要靠做Misc上分？

就参赛的选手来讲，直观感觉就是HN-大佬真多啊，前面有好多诶??；就题目而言，web前四道题好水啊，不是原题就是改编题，Misc纯脑洞，不解释。

强网杯前练练手

比赛时间：2019年5月1日至2019年5月25日

Writeup

一、Web：

1.web1：

这题改编来的：

不安全 | 39.100.83.188:8001

```php
<?php
error_reporting(0);
require 'flag.php';
$value = $_GET['value'];
$password = $_GET['password'];
$username = '';

for ($i = 0; $i < count($value); ++$i) {
        if ($value[$i] > 32 && $value[$i] < 127) unset($value);
        else $username .= chr($value[$i]);
        if ($username == 'w3lc0me_To_ISCC2019' && intval($password) < 2333 && intval($password + 1) > 2333) {
                echo 'Hello '.$username.'!', '<br>', PHP_EOL;
                echo $flag, '<hr>';
        }
}

highlight_file(__FILE__);
```

直接代码审计，需要GET传两个参数，一个value，一个password。

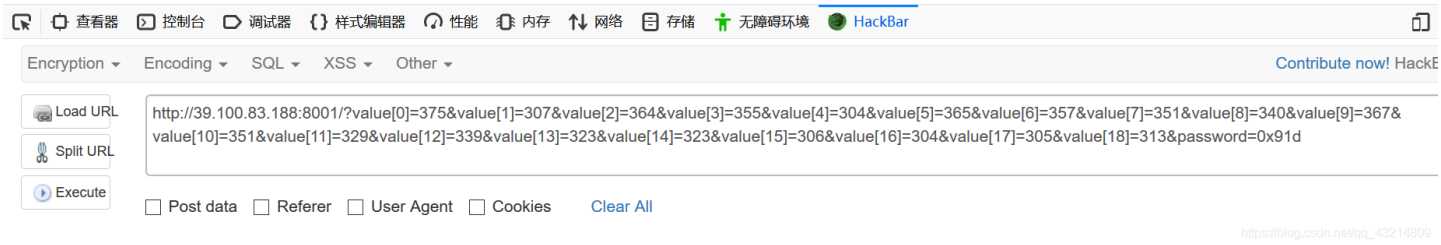
先看value，通过后面的代码不难看出，value是数组赋值，而且value[$i]的ascii值不能在32到127之间，然后经过chr()把value传到username里，从而进行username的比较，需要使username='w3lc0me_To_ISCC2019'，这样看的话矛盾出现了，因为前面的条件限制了我们直接对value赋username所对应的值，既然这样，看看能不能用别的方法。注意到chr()函数在转换时会自动取模256，所以我们只需在相对应的ascii码上加上256就可以。

再看password，主要解决intval()的问题就行了，如果我们传入16进制的数值，intval(password)在这里会直接返回0，但是intval(password+1)在转换时会将16进制数先转换成10进制数再加1，然后输出。

最后构造payload得到flag：

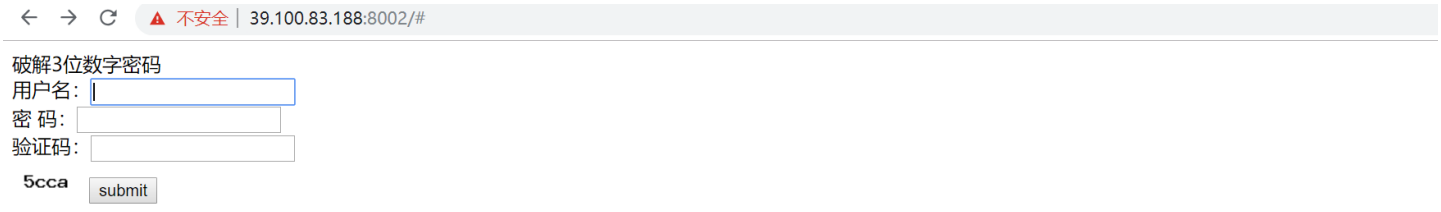
```
Hello w3lc0me_To_ISCC2019!
flag{8311873e241ccad54463eaa5d4efc1e9}
```

```php
<?php
error_reporting(0);
require 'flag.php';
$value = $_GET['value'];
$password = $_GET['password'];
$username = '';

for ($i = 0; $i < count($value); ++$i) {
        if ($value[$i] > 32 && $value[$i] < 127) unset($value);
        else $username .= chr($value[$i]);
        if ($username == 'w3lc0me_To_ISCC2019' && intval($password) < 2333 && intval($password + 1) > 2333) {
                echo 'Hello '.$username.'!', '<br>', PHP_EOL;
                echo $flag, '<hr>';
        }
}
```

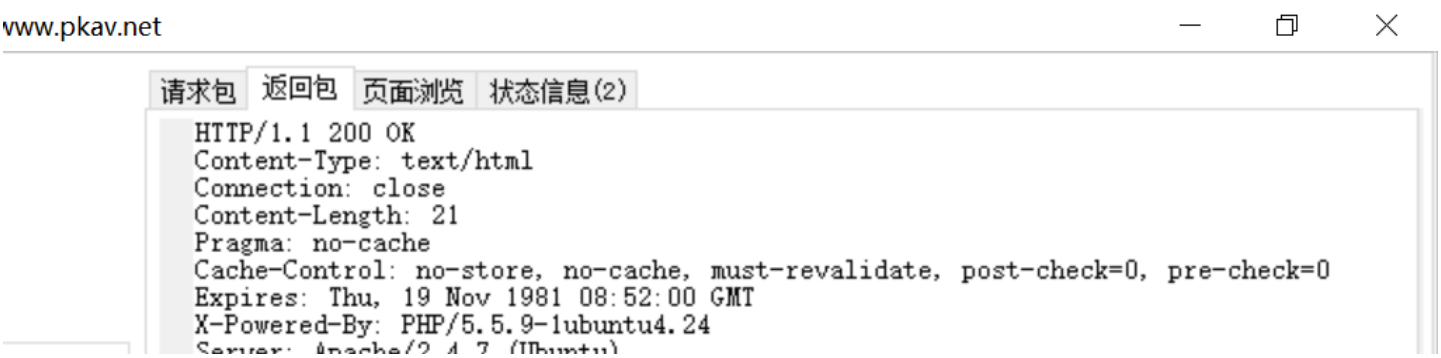


2.web2：

这题考的算是burp的intruder和图形验证码，但是最后是可以试出来的。而且最后的密码是"996"不得不说，还是紧贴时政的?。



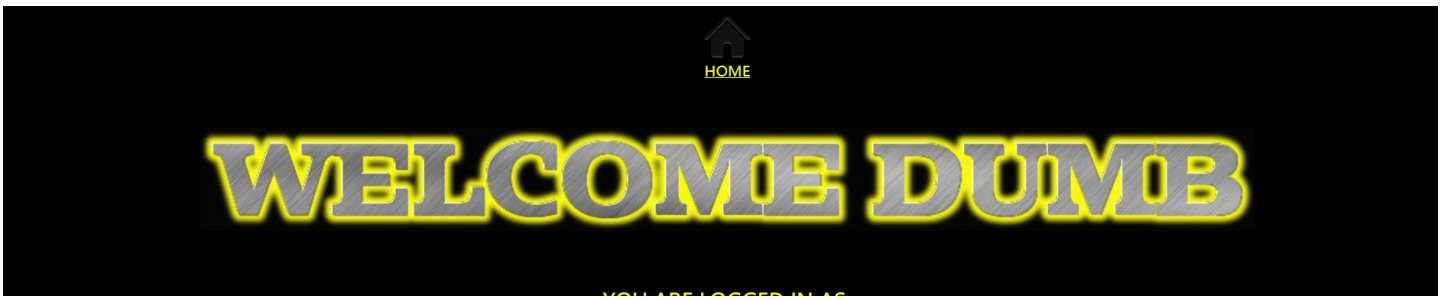
三位数字密码直接000-999，关键就是验证码。这道题是图片的，我曾经在蓝鲸CTF打卡练习场里做过另一个含验证码爆破的，但那个比这个简单，直接在burp里设置宏。对于这道题，这是个图片的，所以直接上工具PKAV，最后得到flag：

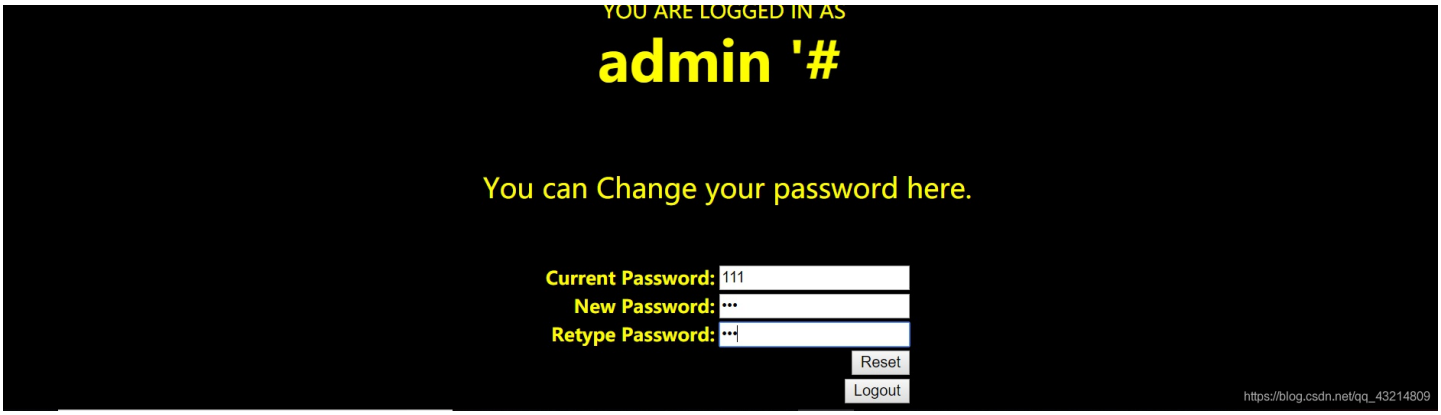


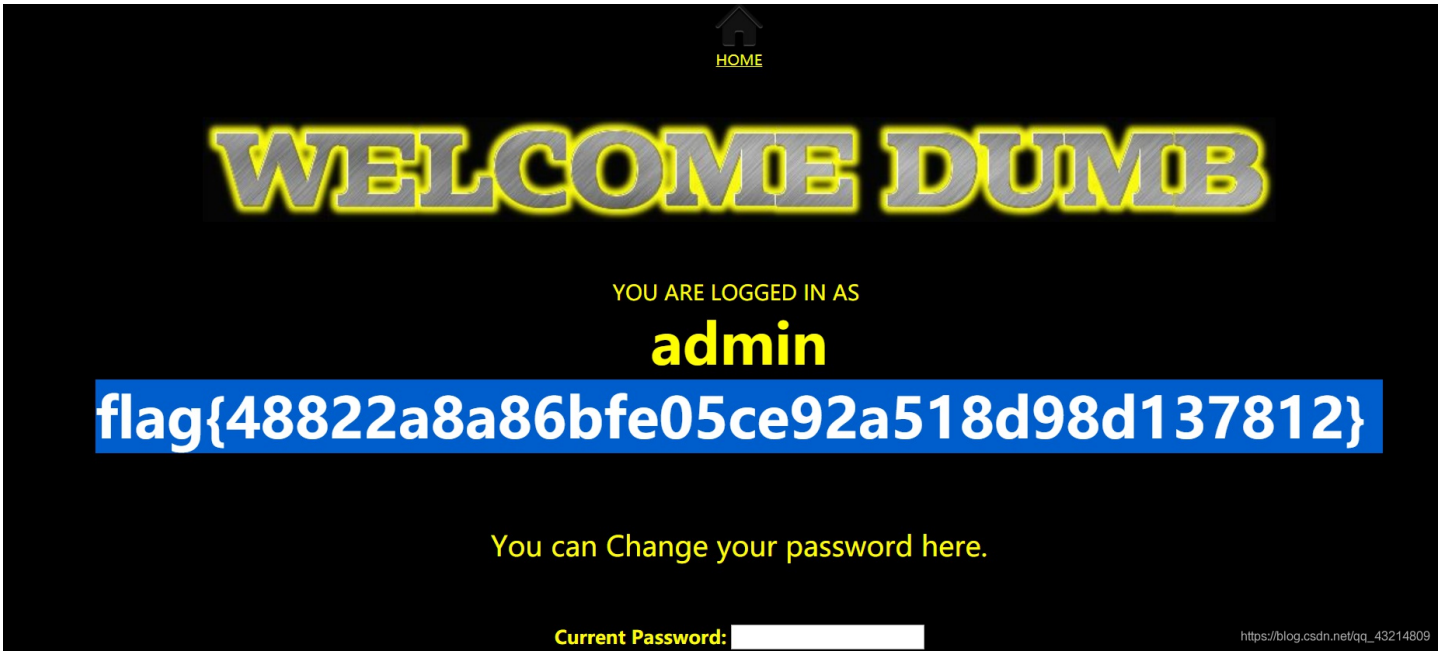
flag is flag{996_ICU}

3.web3：

这题考的二次注入，是sql-libs上的题目，应该是没怎么改动，比较讨厌的是大佬们把能注册的用户名都注册了一遍?，后来实在没办法注册了个"admin '#"，然后就是常规的修改密码，最后再用admin登陆拿到flag：







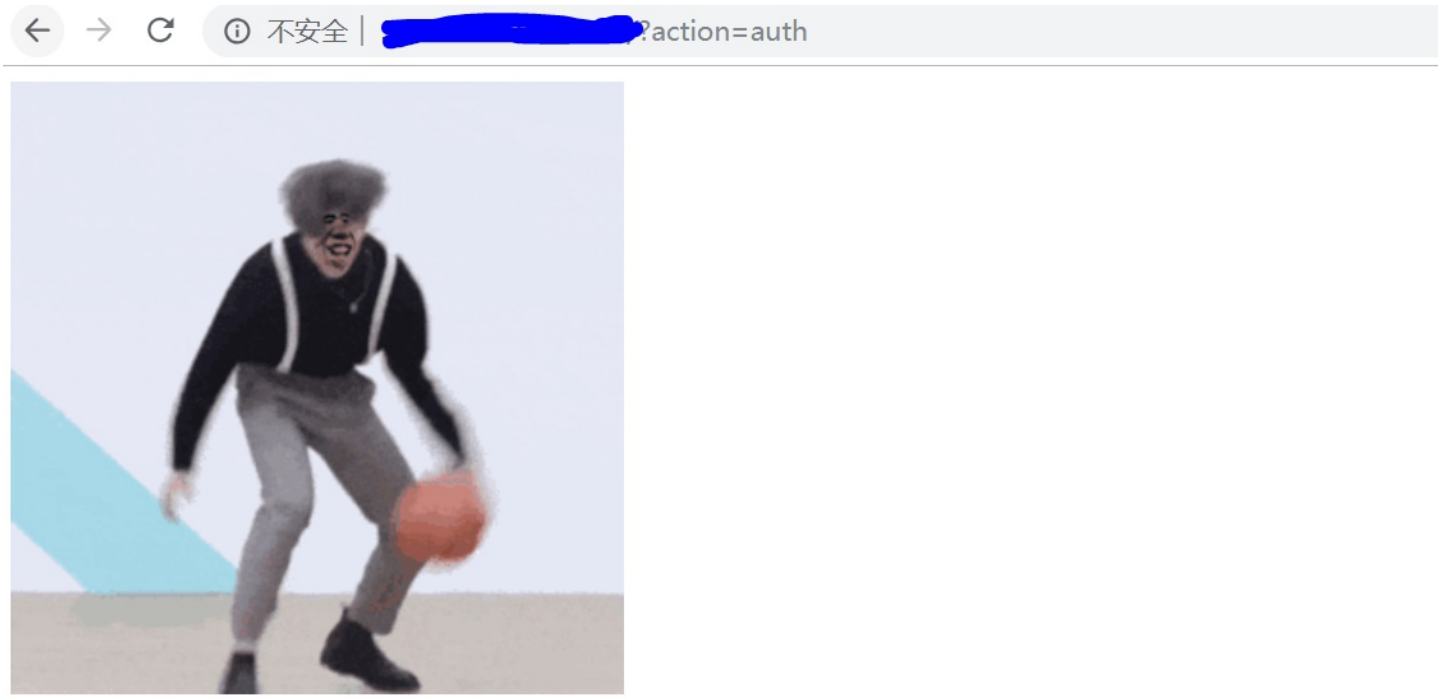
4.web4：

这道题其实也是个改编题，来自国外某CTF的一个php审计题目。打开后一段代码，开始审计：

不安全 | 39.100.83.188:8066

```php
<?php
error_reporting(0);
include("flag.php");
$hashed_key = 'ddbafb4eb89e218701472d3f6c087fdf7119dfdd560f9d1fcbe7482b0feea05a';
$parsed = parse_url($_SERVER['REQUEST_URI']);
if(isset($parsed["query"])){
        $query = $parsed["query"];
        $parsed_query = parse_str($query);
        if($parsed_query!=NULL){
                $action = $parsed_query['action'];
        }

        if($action==="auth"){
                $key = $_GET["key"];
                $hashed_input = hash('sha256', $key);
                if($hashed_input!==$hashed_key){
                        die("<img src='cxk.jpg'>");
                }

                echo $flag;
        }
}else{
        show_source(__FILE__);
}?>
```

这尼玛有点秀啊，cxk都出来了，怀着一刻好奇心看一下：



呵呵呵，我尼玛动态图：

好了回归正题，开始代码审计，需要GET传参，先是action，直接赋值"auth"，比较有意思的是下面的内容，他是想让我们sha256的值相等，一开觉得有点意思，后来又看了看前面的代码，发现一个东西"parse_str()“这是一个变量覆盖的函数，可以查php手册找到，我在别的地方查的手册：

定义和用法

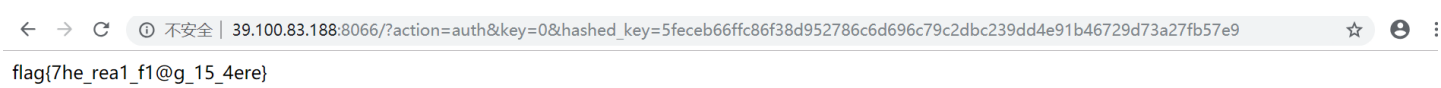
parse_str() 函数把查询字符串解析到变量中。

注释：

如果未设置 array 参数，由该函数设置的变量将覆盖已存在的同名变量。

注释： php.ini 文件中的 magic_quotes_gpc 设置影响该函数的输出。如果已启用，那么在 parse_str() 解析之前，变量会被 addslashes() 转换。

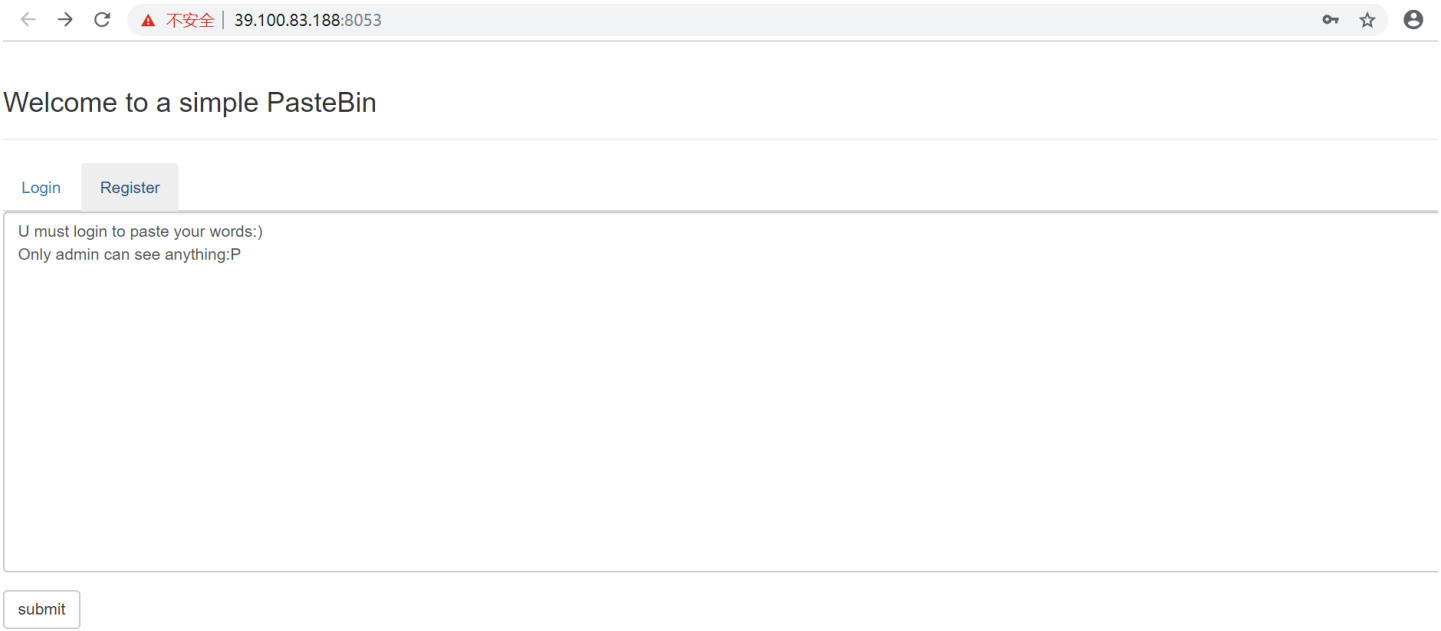
所以一切就变得简单了，我们只要GET传一个key值，然后再传key的sha256值，进行覆盖，我这里取得0，直接就可以得到flag。

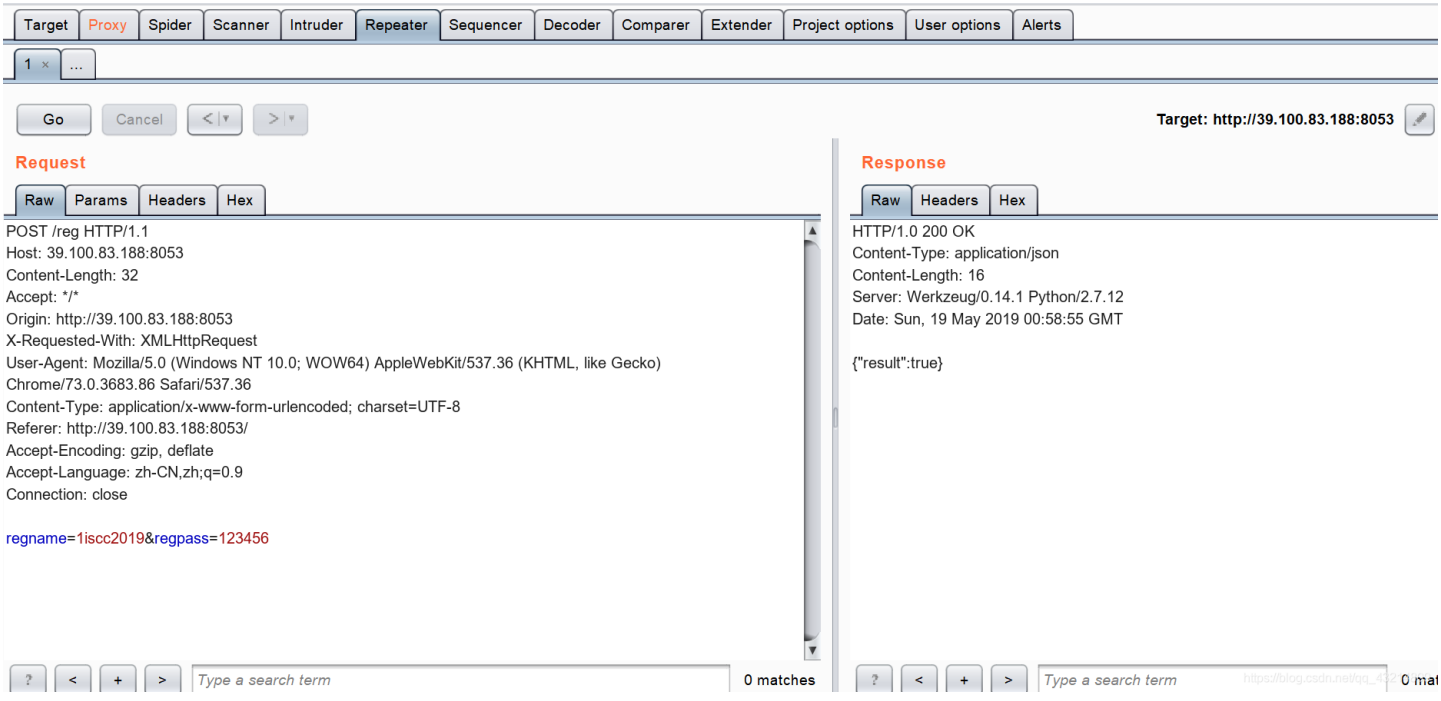


flag{7he_rea1_f1@g_15_4ere}

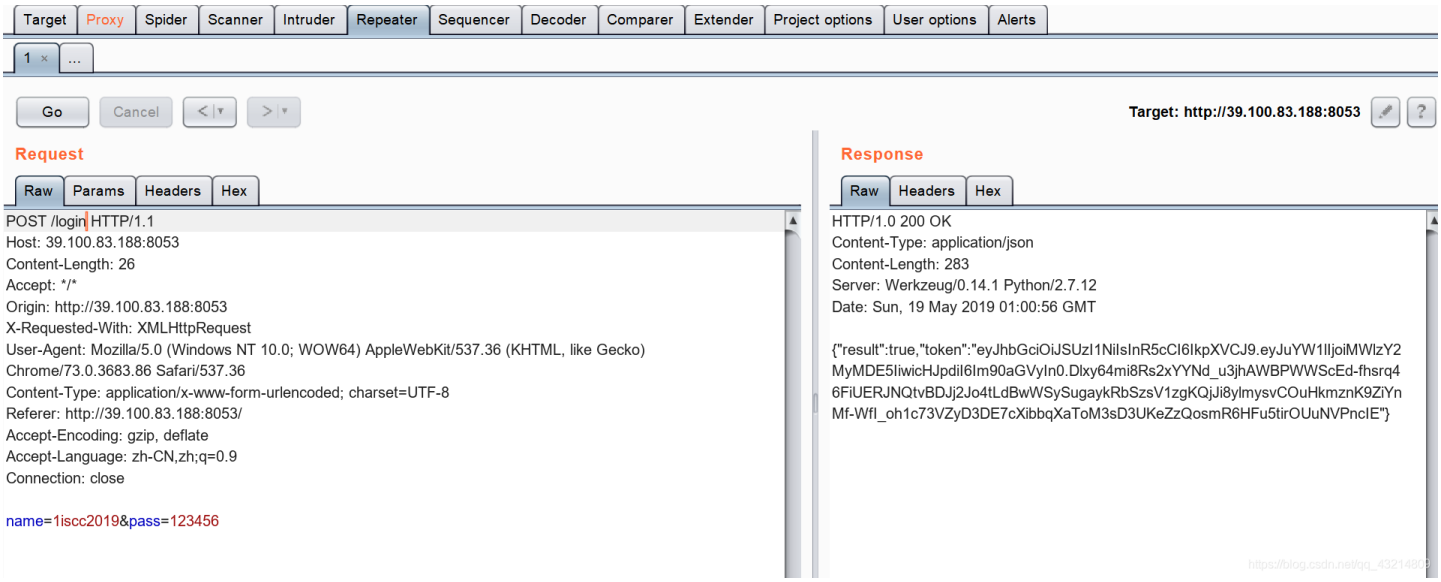
5.web5、6：（未完成）

web6这道题的话先注册一个账号，注册这就卡了一会，一开始注册的几个用户名都会显示"false"：





然后继续，找到一段代码，和相关的源码：




```
function getlist(){
    token = window.localStorage.getItem("token");
    if (token==null||token==undefined){
        alert("u must login first");
        window.location.href = "/";
        return;
    }
    auth = "iscc19 " + token;
    $.ajax({
        url: '/list',
        type: 'GET',
        headers:{"Authorization":auth},
    })
    .success(function(data) {
        result = data.result;
        if(result){
            content = "the user " + data.username +" has these links:\n";
            for (var i in data.links){
                content = content + "/text/" + data.links[i] + "\n";
            }
            alert(content);
        }else{
            alert("list fail");
        }
    });

}
```


不安全 | 39.100.83.188:8053/static/js/common.js

```
function paste(){
    var content = escape($("#content").val());
    token = window.localStorage.getItem("token");
    if (token==null||token==undefined){
        alert("u must login first");
        window.location.href = "/";
        return;
    }
    auth = "iscc19 " + token;
    $.ajax({
        url: '/paste',
        type: 'POST',
        headers:{"Authorization":auth},
        data: {"content": content},
    })
    .success(function(data) {
        result = data.result;
        if(result){
            alert("u can open it with:" + "/text/" + data.link);
        }else{
            alert("paste fail");
        }
    });


}

function logout(){
    localStorage.clear();
    window.location.href = "/";
}

function getpubkey(){
    /*
    get the pubkey for test
```

```
    /pubkey/{md5(username+password)}
    */
}
```

然后就没然后了，等着看师傅们的wp了

web5应该是注入，但是没找到登陆口，一开始的话修改一下User-Agent，里面加上一个Union.373，提示输入用户名，密码。这一部分应该是要有注入的了，然后同理，坐等…

二、MISC：

MISC的题目看脑洞了，只要脑洞大，就能拿flag，但是flag的提交格式真是千变万化??。

1.隐藏的信息：

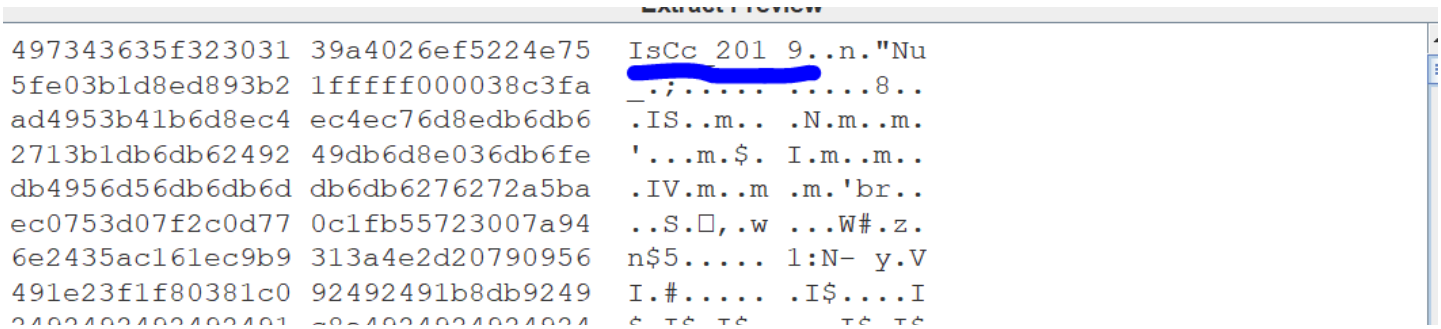
打开文件后是一串进制数，先观察一下，就我当时想的就是应该是8进制，因为没有出现7以上的数，那么直接找对应的ASCII码：

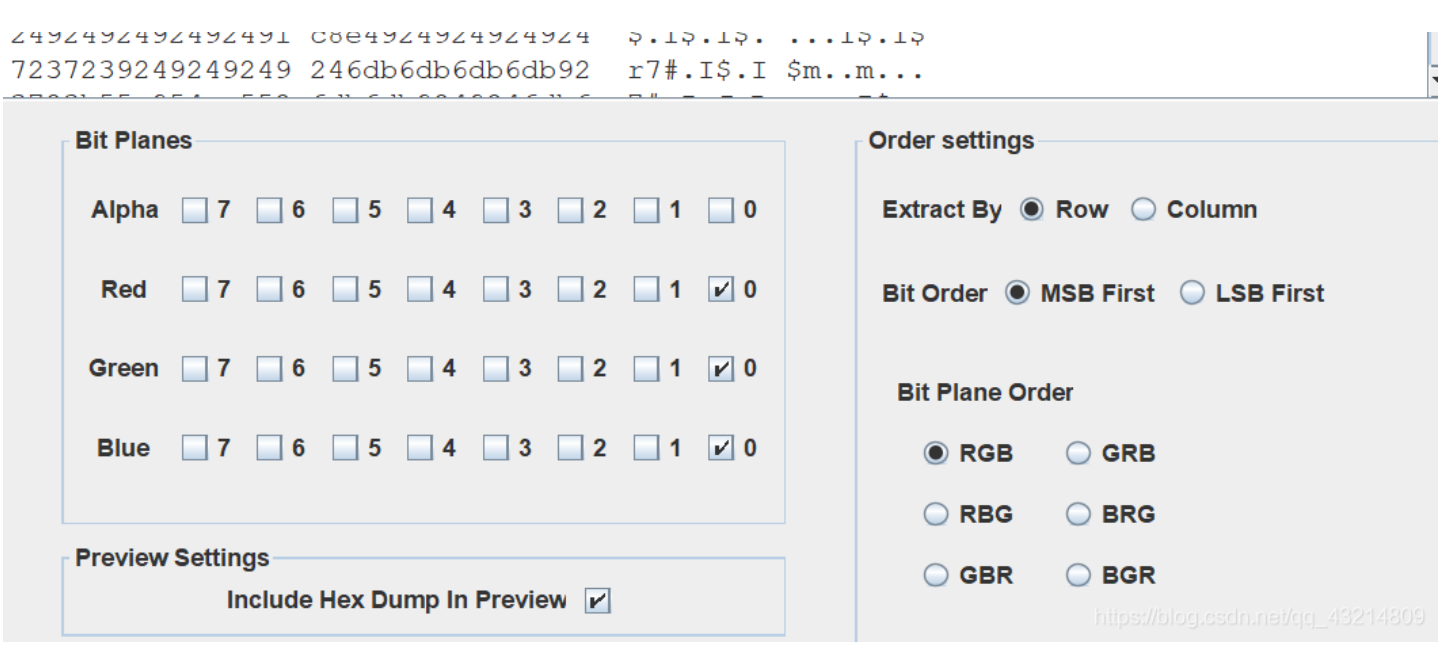
```
0126 062 0126 0163 0142 0103 0102 0153 0142 062 065 0154 0111 0121 0157 0113 0111 0105 0132 0163 0131 0127 0143 066 0111 0105 0154 0124
0121 060 0116 067 0124 0152 0102 0146 0115 0107 065 0154 0130 062 0116 0150 0142 0154 071 0172 0144 0104 0102 0167 0130 063 0153 0167
0144 0130 060 0113

126 62 126 163 142 103 102 153 142 62 65 154 111 121 157 113 111 105 132 163 131 127 143 66 111 105 154 124 121 60 116 67 124 152 102 146
115 107 65 154 130 62 116 150 142 154 71 172 144 104 102 167 130 63 153 167 144 130 60 113

V2VsbCBkb25lIQoKIEZsYWc6IElTQ0N7TjBfMG5lX2Nhbl9zdDBwX3kwdX0K
```


很明显是一段base64编码，直接解码：

2.倒立屋：

这题我尼玛真是神了，先给一张图，直接放stegsolve里分析，然后一个一个试，一开始看到了一个：



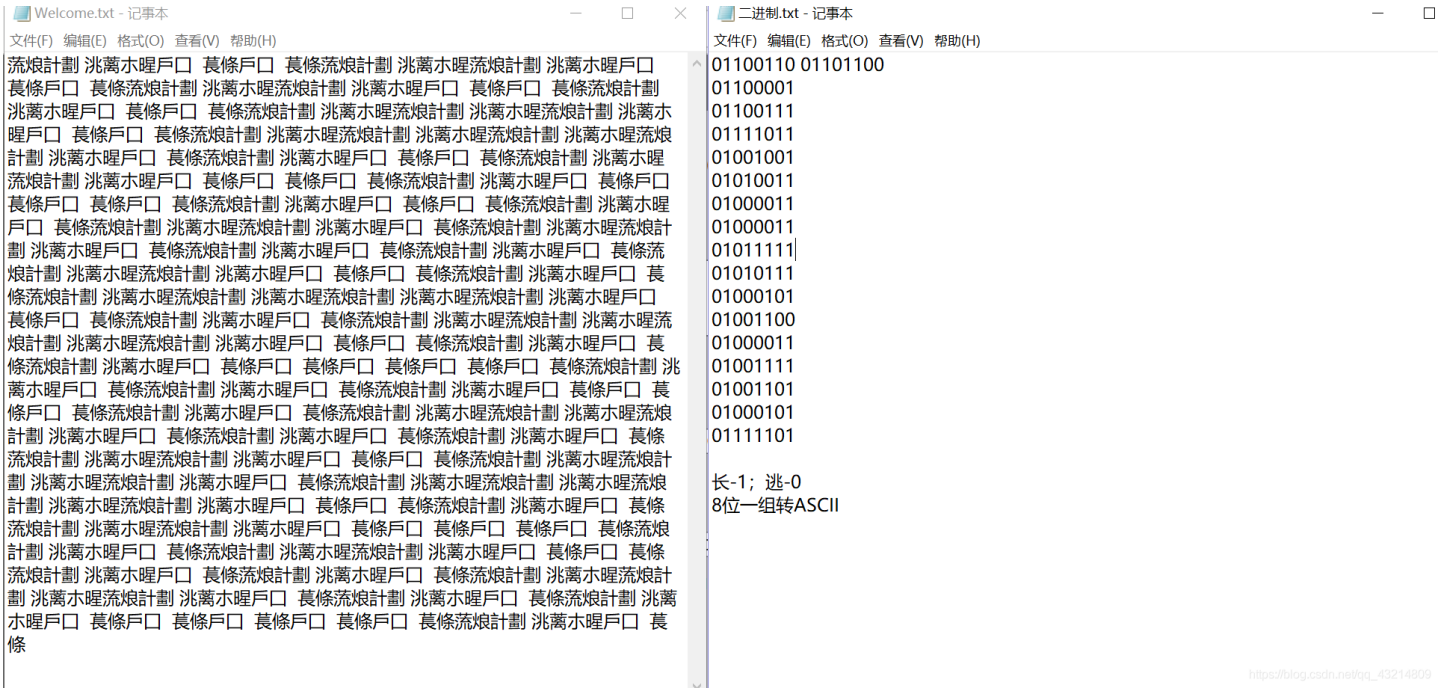


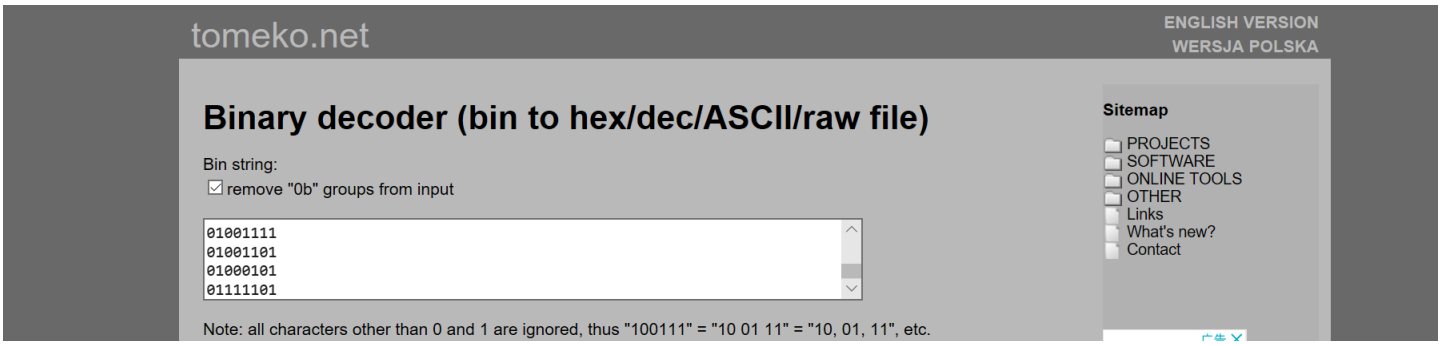
我一开始以为没个什么卵用，继续找，找了好长一段时间，啥也没找到，什么方法都试过了，最后又回到这个地方，那就试试这个呗，正着输没用，那就反着来，最后尼玛竟然真是flag，"9102_cCsl"我勒个擦，真是服了。??

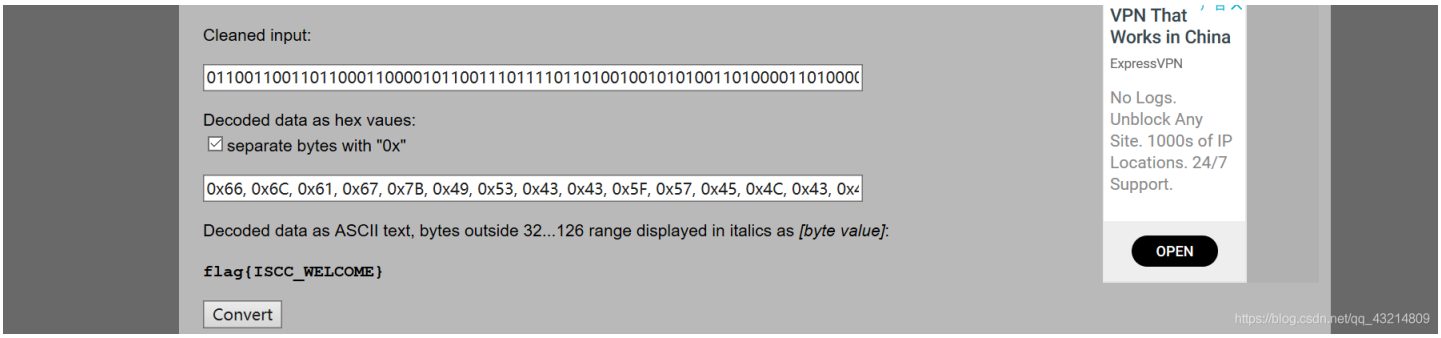
3.Welcome：

这题也还好吧，打开后又是一个文本文件，一堆繁体字，还是蛮有规律的繁体字，当时想的是'01'编码或是'—.'之类的，然后就是一个一个试。

先是01，根据他的分组，有两个情况，第一个是以逃开始的文字段编为1；长开头的编为0。另外一个是逃开始的位0，长开头的为1。无论怎样都是8位编为1组，最后在第二组里二进制转码得到flag：





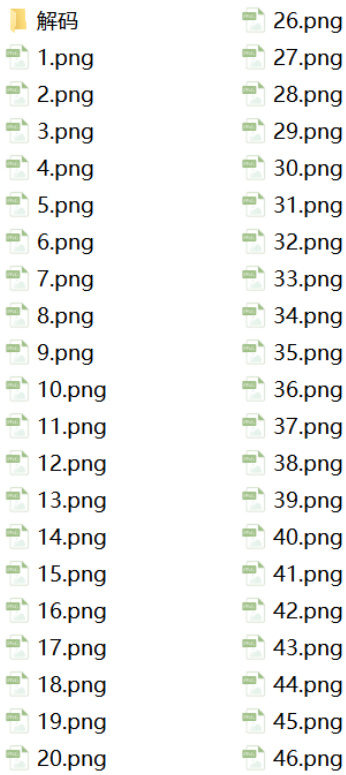


4.最危险的地方就是最安全的地方：

这个也是先打开发现图片文件损坏，没办法看，notepad打开看了一下，里面还有其他的隐藏图片，打开foremost分离出来：

```
00000036.png         833 B          18611          (410 x 410)2:      00000038.png         873 B          19492          (410 x
410)3:     00000039.png         811 B          20413          (410 x 410)4:      00000041.png         834 B          21272
(410 x 410)5:      00000043.png         794 B          22154          (410 x 410)6:      00000044.png         858 B          22996
 (410 x 410)7:      00000046.png         814 B          23902          (410 x 410)8:      00000048.png         872 B          24764
 (410 x 410)9:      00000050.png         855 B          25684          (410 x 410)10:     00000051.png         856 B          26587
 (410 x 410)11:     00000053.png         826 B          27491          (410 x 410)12:     00000055.png         833 B          28364
 (410 x 410)13:     00000057.png         849 B          29245          (410 x 410)14:     00000058.png         809 B          30142
 (410 x 410)15:     00000060.png         817 B          30999          (410 x 410)16:     00000062.png         855 B          31864
 (410 x 410)17:     00000063.png         817 B          32767          (410 x 410)18:     00000065.png         851 B          33632
 (410 x 410)19:     00000067.png         861 B          34531          (410 x 410)20:     00000069.png         857 B          35440
 (410 x 410)21:     00000070.png         853 B          36345          (410 x 410)22:     00000072.png         817 B          37246
 (410 x 410)23:     00000074.png         824 B          38110          (410 x 410)24:     00000076.png         827 B          38982
 (410 x 410)25:     00000077.png         862 B          39857          (410 x 410)26:     00000079.png         843 B          40767
 (410 x 410)27:     00000081.png         846 B          41658          (410 x 410)28:     00000083.png         845 B          42552
 (410 x 410)29:     00000084.png         842 B          43445          (410 x 410)30:     00000086.png         826 B          44335
 (410 x 410)31:     00000088.png         857 B          45209          (410 x 410)32:     00000090.png         826 B          46114
 (410 x 410)33:     00000091.png         864 B          46988          (410 x 410)34:     00000093.png         816 B          47899
 (410 x 410)35:     00000095.png         844 B          48763          (410 x 410)36:     00000096.png         857 B          49655
 (410 x 410)37:     00000098.png         850 B          50560          (410 x 410)38:     00000100.png         857 B          51458
 (410 x 410)39:     00000102.png         864 B          52363          (410 x 410)40:     00000104.png         814 B          53275
 (410 x 410)41:     00000105.png         840 B          54137          (410 x 410)42:     00000107.png         836 B          55025
 (410 x 410)43:     00000109.png         828 B          55909          (410 x 410)44:     00000110.png         836 B          56785
 (410 x 410)45:     00000112.png         865 B          57668          (410 x 410)46:     00000150.png         838 B          77030
 (410 x 410)47:     00000152.png         862 B          77915          (410 x 410)48:     00000153.png         848 B          78824
 (410 x 410)49:     00000155.png         883 B          79719          (410 x 410)Finish: Fri May 17 12:22:56 201950 FILES EXTRACTED
zip:= 1png:= 49----------------------------------------------------------------Foremost finished at Fri May 17 12:22:56 2019
```


看了一下全部都是二维码，批量解码：



- 21.png
- 22.png
- 23.png
- 24.png
- 25.png
- 47.png
- 48.png
- 49.png
- 50.png

30	35. png	remake:最危险的地方就是最安全的地方+35
31	36. png	remake:最危险的地方就是最安全的地方+36
32	37. png	remake:最危险的地方就是最安全的地方+37
33	38. png	remake:最危险的地方就是最安全的地方+38
34	39. png	remake:最危险的地方就是最安全的地方+39
35	4. png	remake:最危险的地方就是最安全的地方+4
36	40. png	remake:最危险的地方就是最安全的地方+40
37	41. png	remake:最危险的地方就是最安全的地方+41
38	42. png	remake:最危险的地方就是最安全的地方+42
39	43. png	remake:最危险的地方就是最安全的地方+43
40	44. png	remake:最危险的地方就是最安全的地方+44
41	45. png	remake:最危险的地方就是最安全的地方+45
42	46. png	remake:最危险的地方就是最安全的地方+46
43	47. png	remake:最危险的地方就是最安全的地方+47
44	48. png	remake:最危险的地方就是最安全的地方+48
45	49. png	remake:最危险的地方就是最安全的地方+49
46	5. png	remake:最危险的地方就是最安全的地方+5
47	50. png	remake:最危险的地方就是最安全的地方+10086
48	6. png	remake:最危险的地方就是最安全的地方+6
49	7. png	remake:最危险的地方就是最安全的地方+7
50	8. png	remake:最危险的地方就是最安全的地方+8
51	9. png	remake:最危险的地方就是最安全的地方+9

第50个png有问题，winhex看一眼得到flag;

```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000EC0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000ED0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000EE0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000EF0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F00 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F10 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F20 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F30 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F40 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F50 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F60 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F70 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F80 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000F90 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000FA0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000FB0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000FC0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000FD0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000FE0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000FF0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00001000 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00001010 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00001020 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00001030 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00001040 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00001050 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00001060 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00001070 00 00 00 00 00 00 00 00  00 00 00 00 66 00 6C 00                f l
00001080 61 00 67 00 7B 00 31 00  35 00 63 00 43 00 39 00  a g { 1 5 c C 9
00001090 30 00 31 00 32 00 7D 00  00 00 FF E1 08 DD 68 74  0 1 2 }   ÿá Ýht
000010A0 74 70 3A 2F 2F 6E 73 2E  61 64 6F 62 65 2E 63 6F  tp://ns.adobe.co
000010B0 6D 2F 78 61 70 2F 31 2E  30 2F 00 3C 3F 78 70 61  m/xap/1.0/ <?xpa
000010C0 63 6B 65 74 20 62 65 67  69 6E 3D 27 EF BB BF 27  cket begin='ï»¿'
000010D0 20 69 64 3D 27 57 35 4D  30 4D 70 43 65 68 69 48   id='W5M0MpCehiH
000010E0 7A 72 65 53 7A 4E 54 63  7A 6B 63 39 64 27 3F 3E  zreSzNTczkc9d'?>
000010F0 0D 0A 3C 78 3A 78 6D 70  6D 65 74 61 20 78 6D 6C    <x:xmpmeta xml
00001100 6E 73 3A 78 3D 22 61 64  6F 62 65 3A 6E 73 3A 6D  ns:x="adobe:ns:m
```

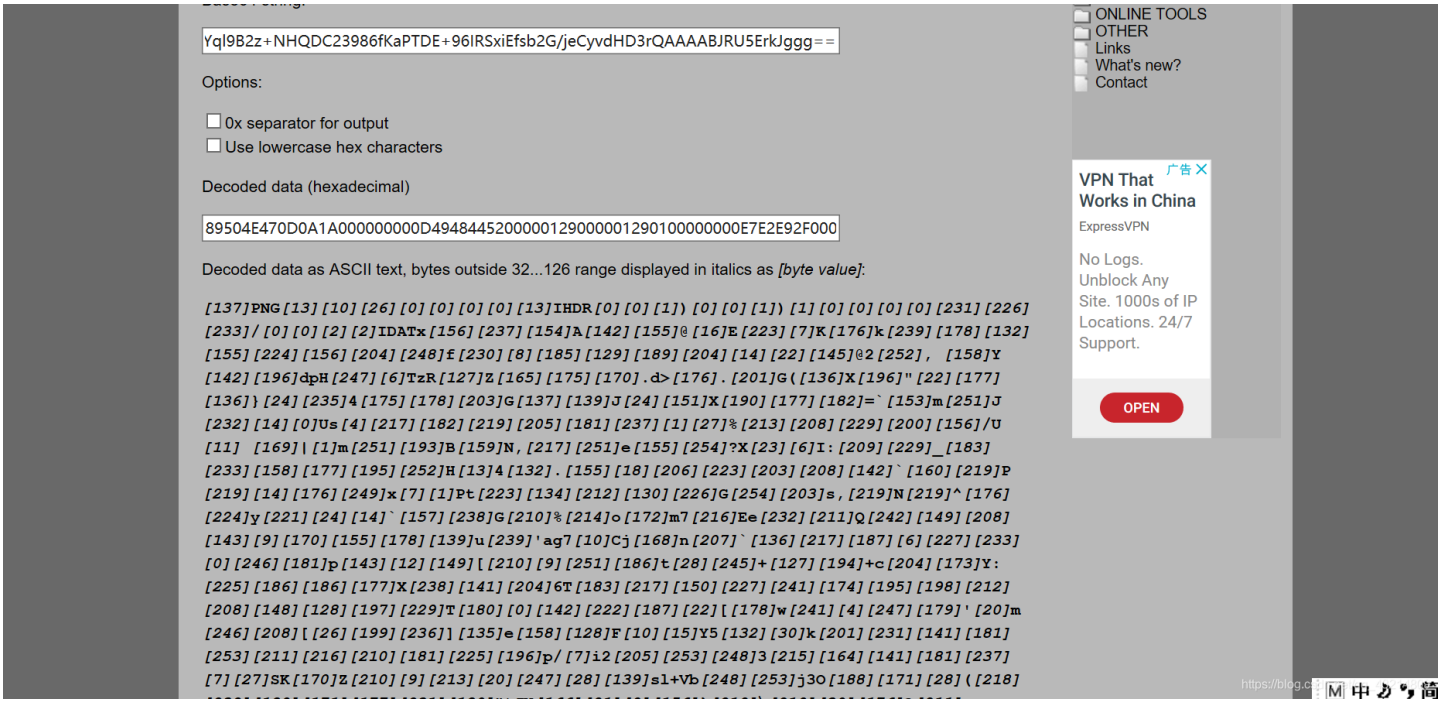

5.无法运行的exe:

打开后一个exe文件，无法运行，惯例notepad看一眼，是一堆编码，姑且当作base64:

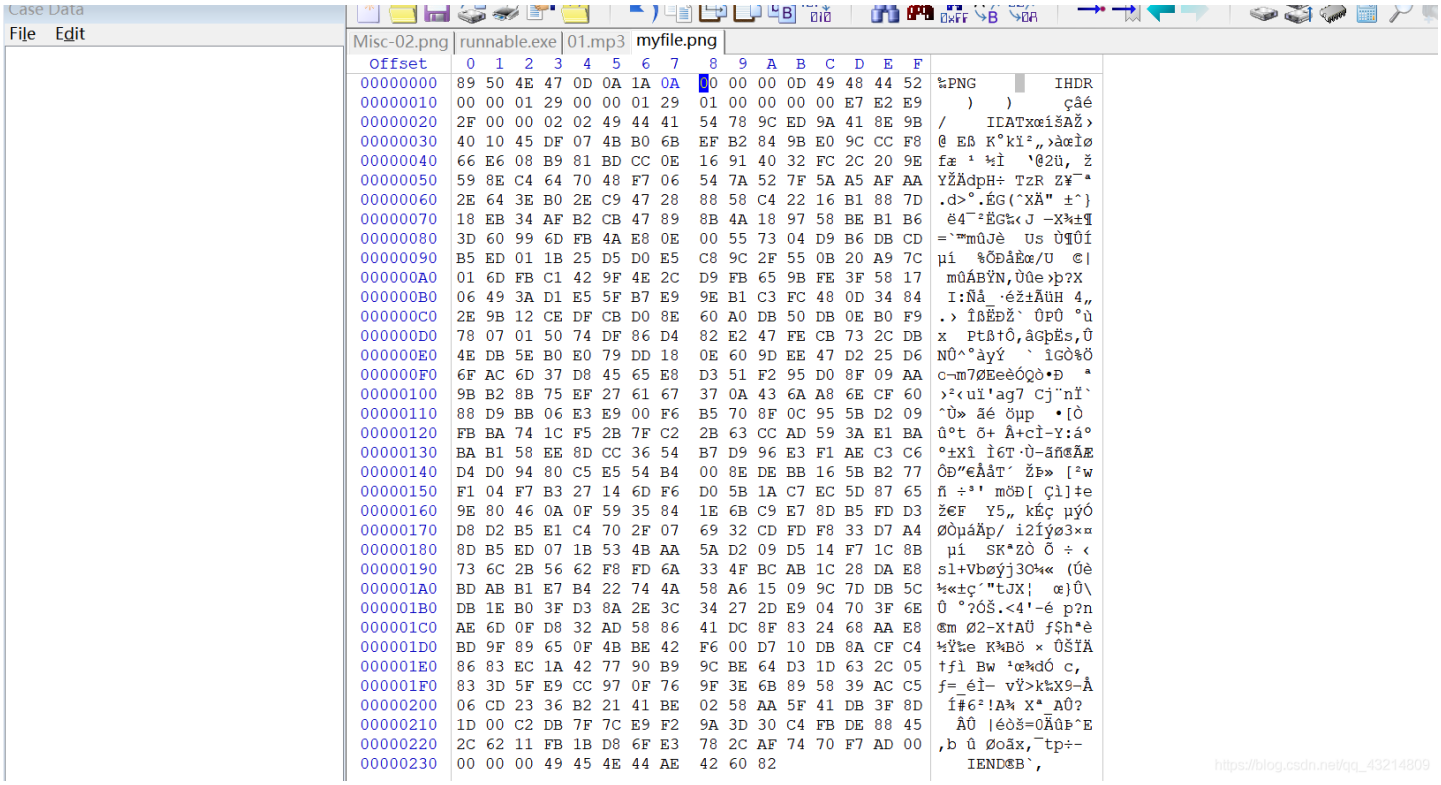
runnable. exe

iVBORw0KGgAAAAANSUhEUgAAASkAAAEpAQAAAADn4ukvAAACAklEQVR4nO2aQY6bQBBF3wdLsGvvsoSb4JzM+GbmCLmBvcwOFpFAMvwsIJ5ZjsRkcEj3BlR6Un9apa+qLmQ+sC7JRyiIWMQiFrGIfRjrNK

然后转一下：



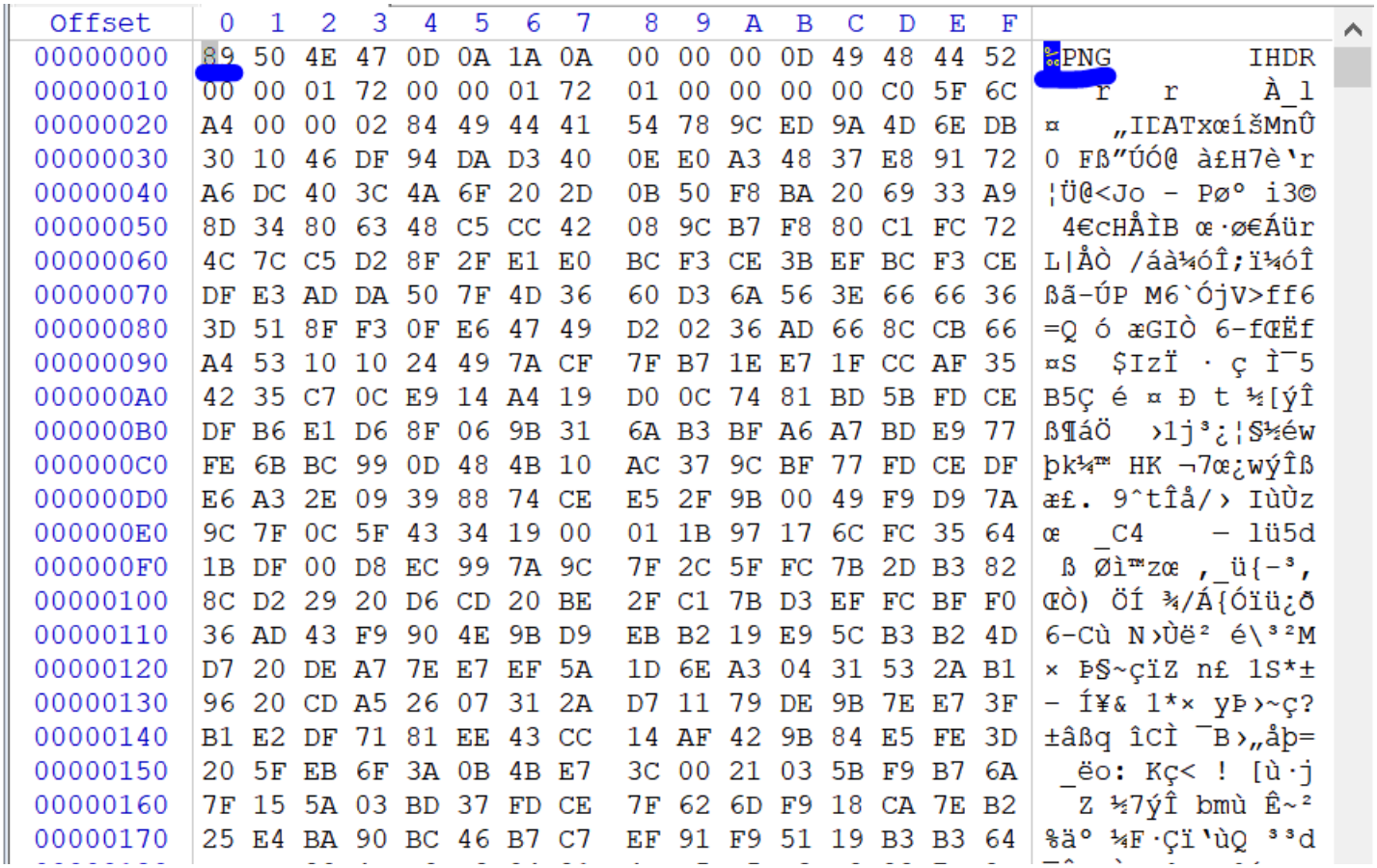
发现出来个png，先把转换出来的转换到本地，后缀名改为png，发现仍然打不开，还是winhex走一波，这里比较靠脑洞的就是要修改一个值"00"改为"0A"，再保存到本地，出现了二维码：



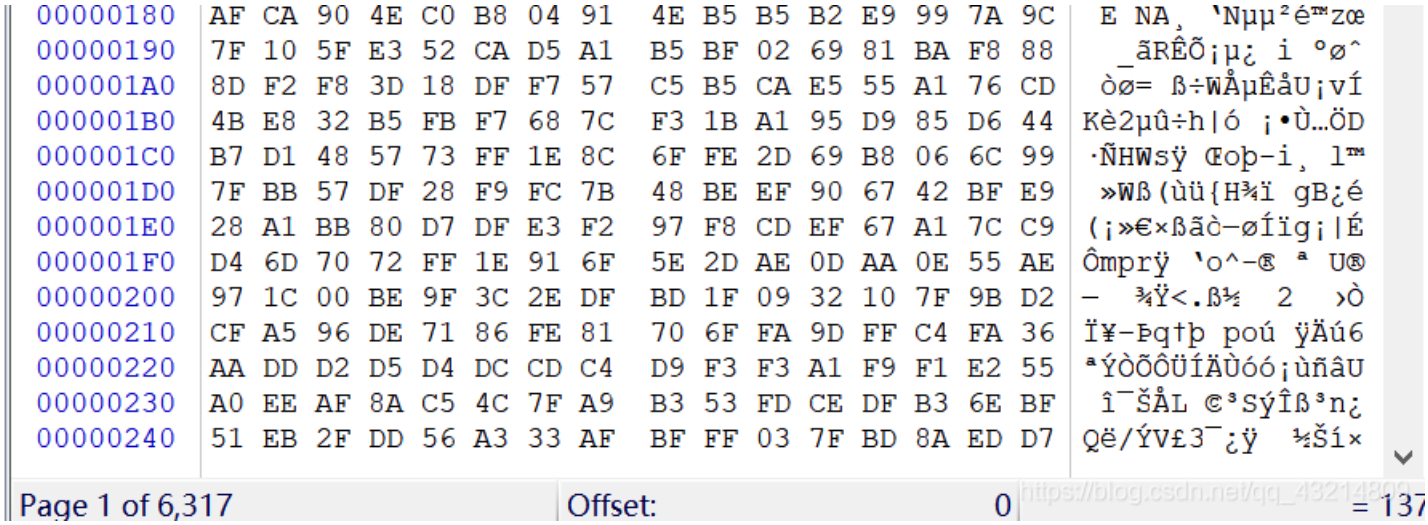
解码即可得到flag：

6.High起来！：

这个也是一张图片文件打不开，老套路winhex看一眼，发现文件头不是png的，改一下保存出现二维码，扫一下：



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNG IHDR
00000010	00	00	01	72	00	00	01	72	01	00	00	00	00	C0	5F	6C	r r À_l
00000020	A4	00	00	02	84	49	44	41	54	78	9C	ED	9A	4D	6E	DB	¤ „IDATxœíšMnÛ
00000030	30	10	46	DF	94	DA	D3	40	0E	E0	A3	48	37	E8	91	72	0 FßÚÓ@ à£H7è'r
00000040	A6	DC	40	3C	4A	6F	20	2D	0B	50	F8	BA	20	69	33	A9	¦Ü@<Jo - Pø° i3©
00000050	8D	34	80	63	48	C5	CC	42	08	9C	B7	F8	80	C1	FC	72	4€cHÅÌB œ·ø€Áür
00000060	4C	7C	C5	D2	8F	2F	E1	E0	BC	F3	CE	3B	EF	BC	F3	CE	L|ÅÒ /áà¼óÎ;ï¼óÎ
00000070	DF	E3	AD	DA	50	7F	4D	36	60	D3	6A	56	3E	66	66	36	ßã-ÚP M6`ÓjV>ff6
00000080	3D	51	8F	F3	0F	E6	47	49	D2	02	36	AD	66	8C	CB	66	=Q ó æGIÒ 6-fŒËf
00000090	A4	53	10	10	24	49	7A	CF	7F	B7	1E	E7	1F	CC	AF	35	¤S $IzÏ · ç Ì¯5
000000A0	42	35	C7	0C	E9	14	A4	19	D0	0C	74	81	BD	5B	FD	CE	B5Ç é ¤ Ð t ½[ýÎ
000000B0	DF	B6	E1	D6	8F	06	9B	31	6A	B3	BF	A6	A7	BD	E9	77	ß¶áÖ ›1j³¿¦§½éw
000000C0	FE	6B	BC	99	0D	48	4B	10	AC	37	9C	BF	77	FD	CE	DF	þk¼™ HK ¬7œ¿wýÎß
000000D0	E6	A3	2E	09	39	88	74	CE	E5	2F	9B	00	49	F9	D9	7A	æ£. 9ˆtÎå/› IùÙz
000000E0	9C	7F	0C	5F	43	34	19	00	01	1B	97	17	6C	FC	35	64	œ _C4 — lü5d
000000F0	1B	DF	00	D8	EC	99	7A	9C	7F	2C	5F	FC	7B	2D	B3	82	ß Øì™zœ ,_ü{-³,
00000100	8C	D2	29	20	D6	CD	20	BE	2F	C1	7B	D3	EF	FC	BF	F0	ŒÒ) ÖÍ ¾/Á{Óïü¿ð
00000110	36	AD	43	F9	90	4E	9B	D9	EB	B2	19	E9	5C	B3	B2	4D	6-Cù N›Ùë² é\³²M
00000120	D7	20	DE	A7	7E	E7	EF	5A	1D	6E	A3	04	31	53	2A	B1	× Þ§~çïZ n£ 1S*±
00000130	96	20	CD	A5	26	07	31	2A	D7	11	79	DE	9B	7E	E7	3F	– Í¥& 1*× yÞ›~ç?
00000140	B1	E2	DF	71	81	EE	43	CC	14	AF	42	9B	84	E5	FE	3D	±âßq îCÌ ¯B›„åþ=
00000150	20	5F	EB	6F	3A	0B	4B	E7	3C	00	21	03	5B	F9	B7	6A	_ëo: Kç< ! [ù·j
00000160	7F	15	5A	03	BD	37	FD	CE	7F	62	6D	F9	18	CA	7E	B2	Z ½7ýÎ bmù Ê~²
00000170	25	E4	BA	90	BC	46	B7	C7	EF	91	F9	51	19	B3	B3	64	%ä° ¼F·Çï'ùQ ³³d



当铺密码解一下：

当铺密码

中口由羊口中中大中中井

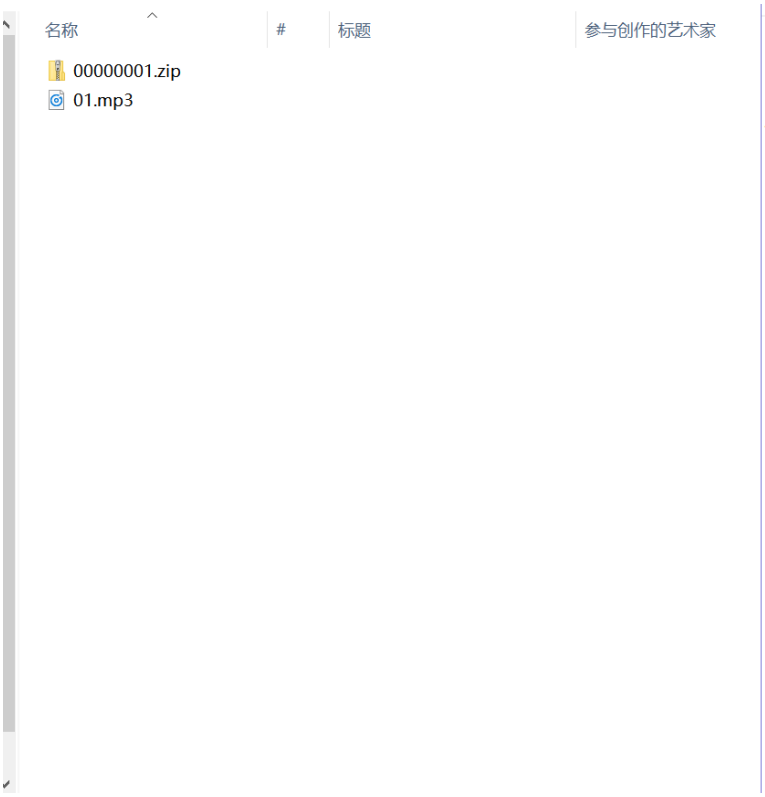
转换密码↓

201902252228

简介：当前汉字有多少笔画出头，就是转化成数字几。
（例：王夫井工夫口 = 678470）

返回

我一开始以为这就是，后来又想了一下，这题应该还有音频，暂且放着不管。foremost继续分离文件，发现"01.mp3"文件：




```
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick MikusAudit FileForemost started at Sat May 18 13:24:16 2019Invocation: foremost Misc-02.png Output directory: /mnt/hgfs/共享文件夹/ISCC/MISC/Misc-02/outputConfiguration file: /etc/foremost.conf
------------------------------------------------------------File: Misc-02.pngStart: Sat May 18 13:24:16 2019Length: 3 MB (3739253 bytes) Num        Name (bs=512)        Size    File Offset    Comment 0:        00000001.zip        3 MB
      701    Finish: Sat May 18 13:24:17 20191 FILES EXTRACTED zip:= 1
------------------------------------------------------------
Foremost finished at Sat May 18 13:24:17 2019
```


然后听了会，嗯还行，继续放到mp3stego下，我一开始没用过这软件，因为之前没遇到过音频隐写，CSDN上搜了一下操作流程，发现有个-P password，第一次没用，啥玩意也没出来，后来想了一下欸，之前的数字好像可以用上了，开始操作：

```
E:\工具合集1\CTF工具合集\隐写\音频隐写\mp3stego-gui>decode -X -P password 01.mp3
MP3StegoEncoder 1.1.15
See README file for copyright info
Input file = '01.mp3'  output file = '01.mp3.pcm'
Will attempt to extract hidden information. Output: 01.mp3.txt
the bit stream file 01.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=0, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=stereo, sblim=32, jsbd=32, ch=2
[Frame 9053]Avg slots/frame = 417.913; b/smp = 2.90; br = 127.986 kbps
[ERROR]Encrypt: unexpected end of cipher message.

E:\工具合集1\CTF工具合集\隐写\音频隐写\mp3stego-gui>decode -X -P 201902252228 01.mp3
MP3StegoEncoder 1.1.15
See README file for copyright info
Input file = '01.mp3'  output file = '01.mp3.pcm'
Will attempt to extract hidden information. Output: 01.mp3.txt
the bit stream file 01.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=0, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=stereo, sblim=32, jsbd=32, ch=2
[Frame 9053]Avg slots/frame = 417.913; b/smp = 2.90; br = 127.986 kbps
Decoding of "01.mp3" is finished
The decoded PCM output file name is "01.mp3.pcm"
```

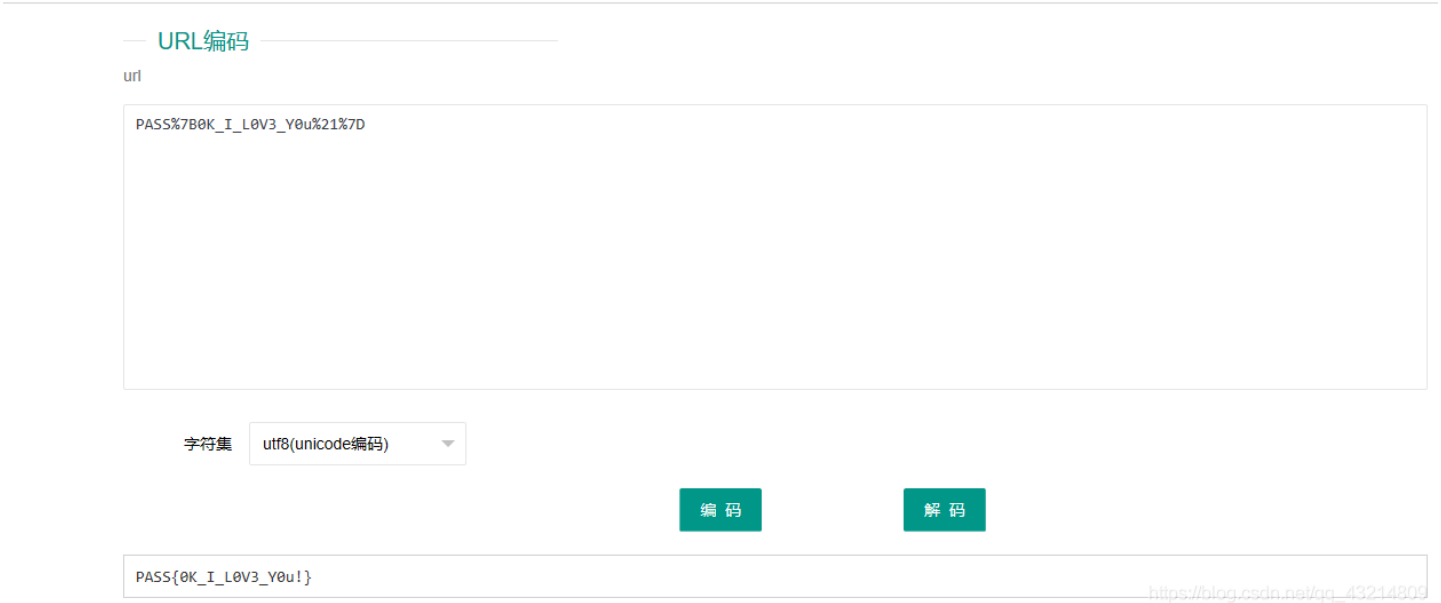

最后出来txt文件，打开直接转ASCII码，得到flag：

```
&#102;&#108;&#97;&#103;&#123;&#80;&#114;&#69;&#116;&#84;&#121;&#95;&#49;&#83;&#99;&#67;&#57;&#48;&#49;&#50;&#95;&#103;&#79;&#48;&#100;&#125;

flag{PrEtTy_1ScC9012_gO0d}
```

7.他们能在一起吗？：

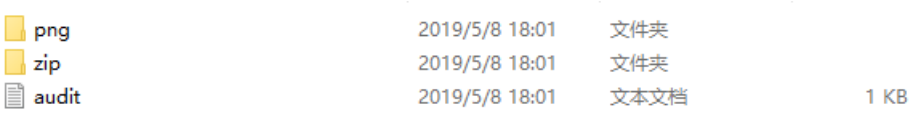
直接一个二维码，扫码得到一串base64文件，解码：

url再解一下：

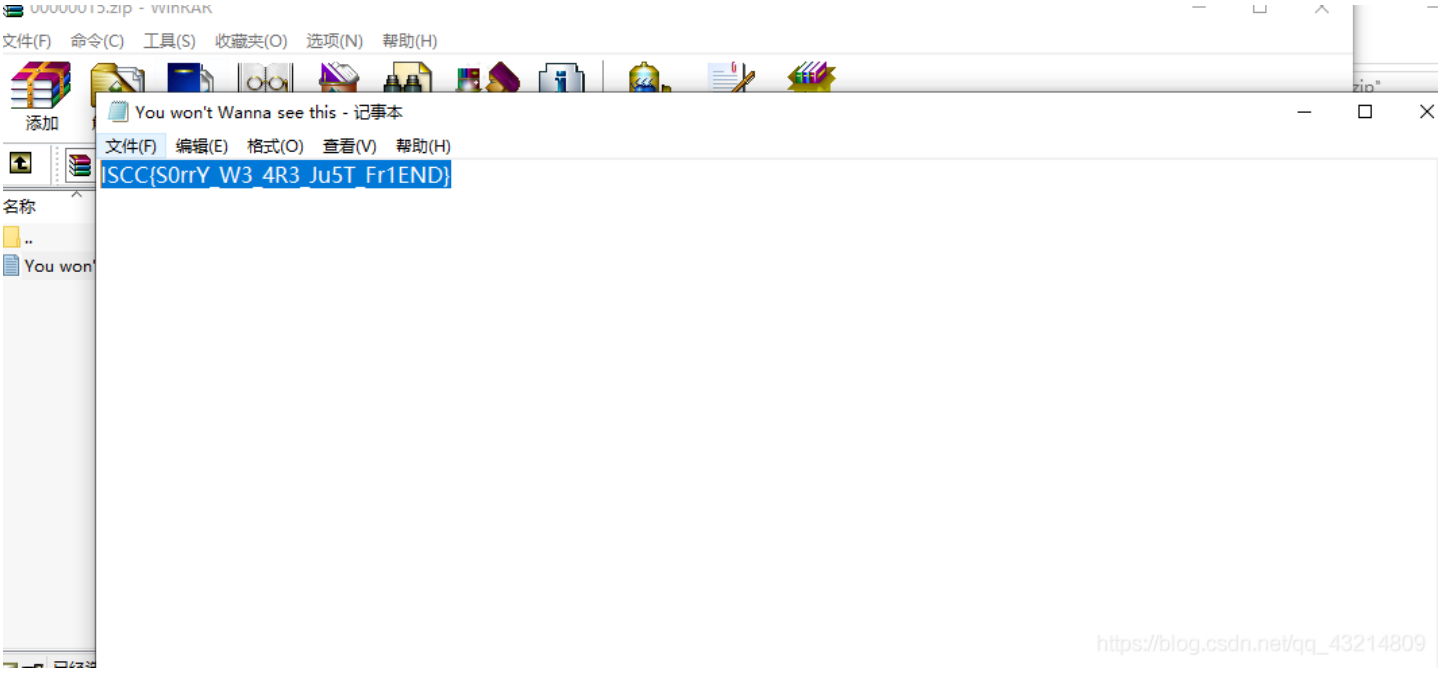


出来的东西先留着，因为二维码里还有东西，同样foremost分离出来：

出来的东西先留着，因为二维码里还有东西，同样foremost分离出来：

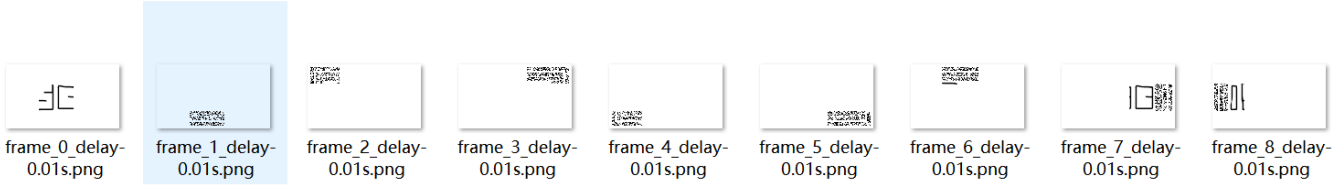


在zip文件夹里有我们想要的东西，但是需要输入密码，直接用前面的那个，放进去得到flag：



8.Aesop's secret:

打开是一个gif动态图，直接按帧数分离：



一开始以为是个二维码拼接起来发现不是，是个ISCC，还是不知道有啥用，那就stegsolve看一下，发现一段代码：

```
4a383771795a4c35 6b416631666d4148
344f653133497534 333562665242755a
5748706e526a5442 6e352b787344484f
4e69523374302b4f 613879472f744f4b
4a4d4e5561756564 764d794e34763451
4b6946756e773d3d 0d0a
Ascii:
U2FsdGVk X19QwGkc
gD0fTjZx gijRzQOG
oCWALh4s RDec2w6x
sY/ux53V uj/AMZBD
J87qyZL5 kAf1fmAH
4Oe13Iu4 35bfRBuZ
gHpnRjTB n5+xsDHO
NiR3t0+O a8yG/tOK
JMNUaued vMyN4v4Q
KiFunw== ..
```


一开始想着是不是base64，试了一下不是，后来又想了一下应该是有密码之类的，得用上"ISCC"，结合题目名称，确定是AES，解码开始，连结两次码，得到flag：

9.Keyes' secret:

这题蛮狠的，考的键盘编码，看完眼睛都快废掉了，给了一大长串密码，最后的落脚点就是在"{WSXIUYHNBVTRFVBTRFVBQWERTYQAZSCEWSXCDEEFVTYHNMKJTGBNMJUYGRDXCVBMNBVCDRTGHUWSXCFEQWERTYTRFVBWSXNBVCXSWERFRFVGYHNWSXCDEMNBVCDRTGHU}"上面，按照我自己的理解尝试了许多次，可能下面的也存在错的，但是最后提交对了？

1.“WSX”:I; 2.“IUYHNBV”:S; 3.“TRFVB”:C; 4.与3相同; 5."QWERTY"对应的是空格; 6.“QAZSCE”:K; 7.“WSXCDE”:E; 8.“EFVT”:Y; 9.“YHNMKJ+GRDXCVB”:B; 10.“TGBNMJUY”: O; 11.“MNBVCDRTGHU”:R; 12.“WSXCFE”: D; 13.同5; 14.与3相同:C; 15.与1相同:I; 16.“NBVCXSWERF”: P; 17.“RFVGYHN”:H; 18.与7相同:E; 19.与11相同:R;

总结一下{ISCC KEYBOARD CIPHER}，提交的格式我记不清了。