CTF之PHP代码审计1

原创

bmth666 于 2020-03-10 21:36:23 发布 1978 收藏 15

分类专栏： ctf 文章标签： 安全 php

本文链接：https://blog.csdn.net/bmth666/article/details/104769645

文章目录

弱类型

$a==$b 等于 ture：如果类型转换后$a等于$b

$a===$b 全等 ture：如果$a等于$b，并且他们的类型也相同

如果一个数值和一个字符串比较，那么会将字符串转换为数值

```php
<?php
var_dump(true==2);//true
var_dump(true==0);//false
var_dump(''==0);//true
var_dump(0==false);//true
var_dump(intval(false));//int(0)
var_dump('123'==123);//字符串转换为数字类型,true
var_dump('abc'==0);//true
var_dump('123a'==123);//true
var_dump(intval('123a'));//int(123)
var_dump('a123'==123);//false
var_dump(intval('a123'));//int(0)
```

可以自己测试一下，能看到true不为0时都为真，字母开头的话转换为int型就为0

```
var_dump('0x01'==1);//0x01为16进制，true
var_dump(intval(0x01));//int(1)
var_dump(intval(0x11));//int(17)
var_dump('0e123456789'=='0e987654321');//true
var_dump(intval(0e12345678))//int(0)
var_dump('1e3'==1000);//true
var_dump([0]=='');//false
var_dump([0]==['']);//true
```

可以看出能分辨出16进制和科学计数法，并且数组和非数组比较时都为假

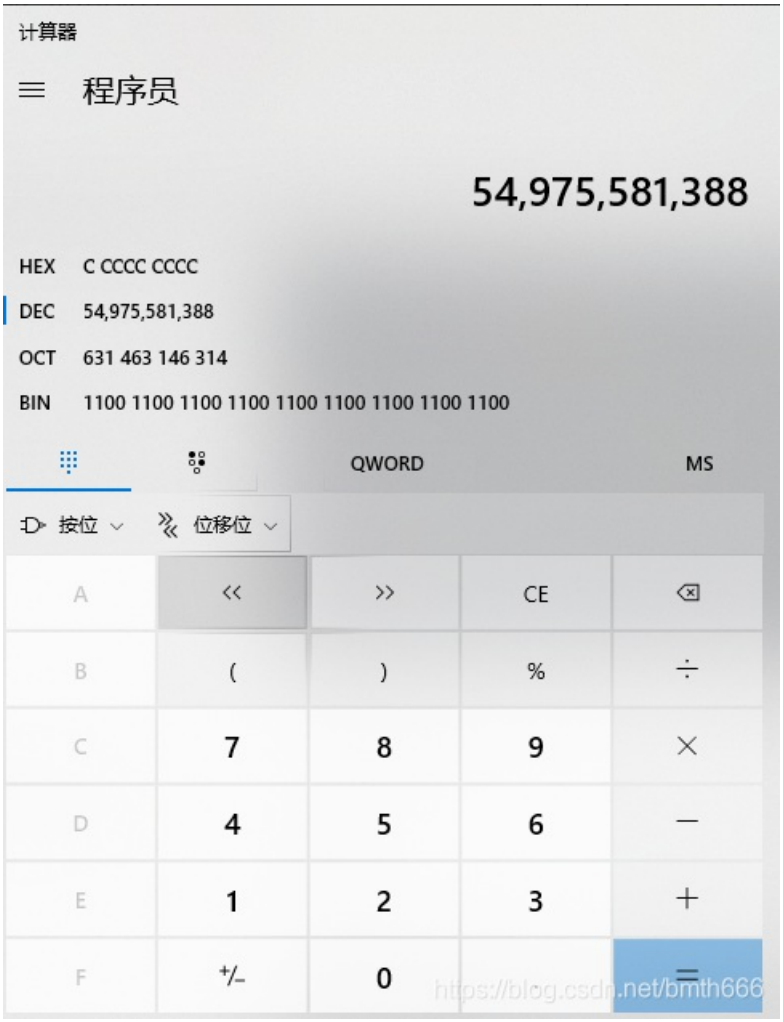
例题1

实验环境：南邮ctf：起名字真难

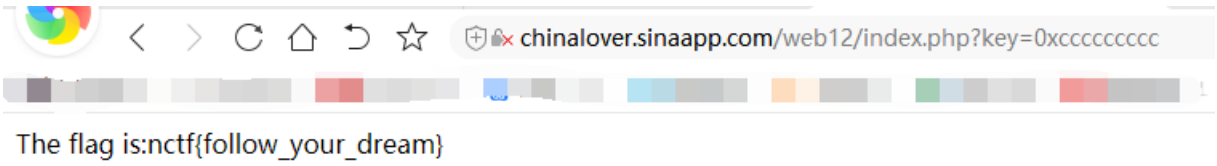
给出的源码如下：

```
<?php
function noother_says_correct($number)
{
       $one = ord('1');
       $nine = ord('9');
       for ($i = 0; $i < strlen($number); $i++)
       {
               $digit = ord($number{$i});
               if ( ($digit >= $one) && ($digit <= $nine) )
               {
                       return false;
               }
       }
          return $number == '54975581388';
}
$flag='*******';
if(noother_says_correct($_GET['key']))
   echo $flag;
else
   echo 'access denied';
?>
```

意思是我们输入的key中不能有数字，但是要和`54975581388`相等，所以使用弱类型比较，使用0x的格式输入，先查看要求的数字的16进制



刚好没有数字，那么我们传入`0xcccccccc`进去就可以得到flag了



The flag is:nctf{follow_your_dream}

例题2

实验环境：bugku：md5 collision(NUPT_CTF)

这里借用师傅的源代码

```php
<?php
$md51 = md5('QNKCDZO');
$a = @$_GET['a'];
$md52 = @md5($a);
if(isset($a)){
if ($a != 'QNKCDZO' && $md51 == $md52) {
    echo "nctf{*****************}";
} else {
    echo "false!!!";
}}
else{echo "please input a";}
?>
```

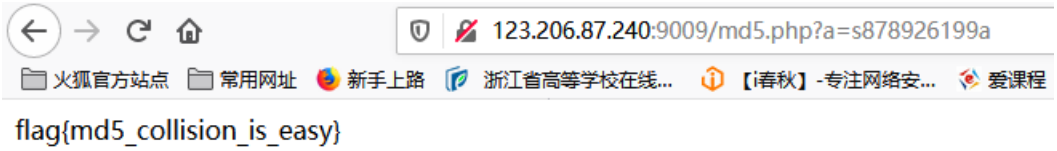
进入题目发现要我们提交一个参数a，随便传一个a的值发现返回false。由于不是`===`，0e开头的md5都为0，我们可以传入一个经过md5加密后为0e的值即可

0e开头的md5和原值：

```
QNKCDZO
0e830400451993494058024219903391
240610708
0e462097431906509019562988736854
s878926199a
0e545993274517709034328855841020
s155964671a
0e342768416822451524974117254469
s214587387a
0e848240448830537924465865611904
s214587387a
0e848240448830537924465865611904
s878926199a
0e545993274517709034328855841020
s1091221200a
0e940624217856561557816327384675
s1885207154a
0e509367213418206700842008763514
s1502113478a
0e861580163291561247404381396064
s1885207154a
0e509367213418206700842008763514
s1836677006a
0e481036490867661113260034900752
s155964671a
0e342768416822451524974117254469
s1184209335a
0e072485820392773389523109082030
```

s1665632922a
0e731198061491163073197128363787
s1502113478a
0e861580163291561247404381396064
s1836677006a
0e4810364908676611132600349007 52
s1091221200a
0e940624217856561557816327384675
s155964671a
0e342768416822451524974117254469
s1502113478a
0e861580163291561247404381396064
s155964671a
0e342768416822451524974117254469
s1665632922a
0e731198061491163073197128363787
s155964671a
0e342768416822451524974117254469
s1091221200a
0e940624217856561557816327384675
s1836677006a
0e4810364908676611132600349007 52
s1885207154a
0e5093672134182067008420087635 14
s532378020a
0e22046309585551150758804120581 5
s878926199a
0e5459932745177090343288558410 20
s1091221200a
0e940624217856561557816327384675
s214587387a
0e8482404488305379244658656119 04
s1502113478a
0e861580163291561247404381396064
s1091221200a
0e940624217856561557816327384675
s1665632922a
0e731198061491163073197128363787
s1885207154a
0e5093672134182067008420087635 14
s1836677006a
0e4810364908676611132600349007 52
s1665632922a
0e731198061491163073197128363787
s878926199a
0e5459932745177090343288558410 20

随便选一个传入得到flag



flag{md5_collision_is_easy}

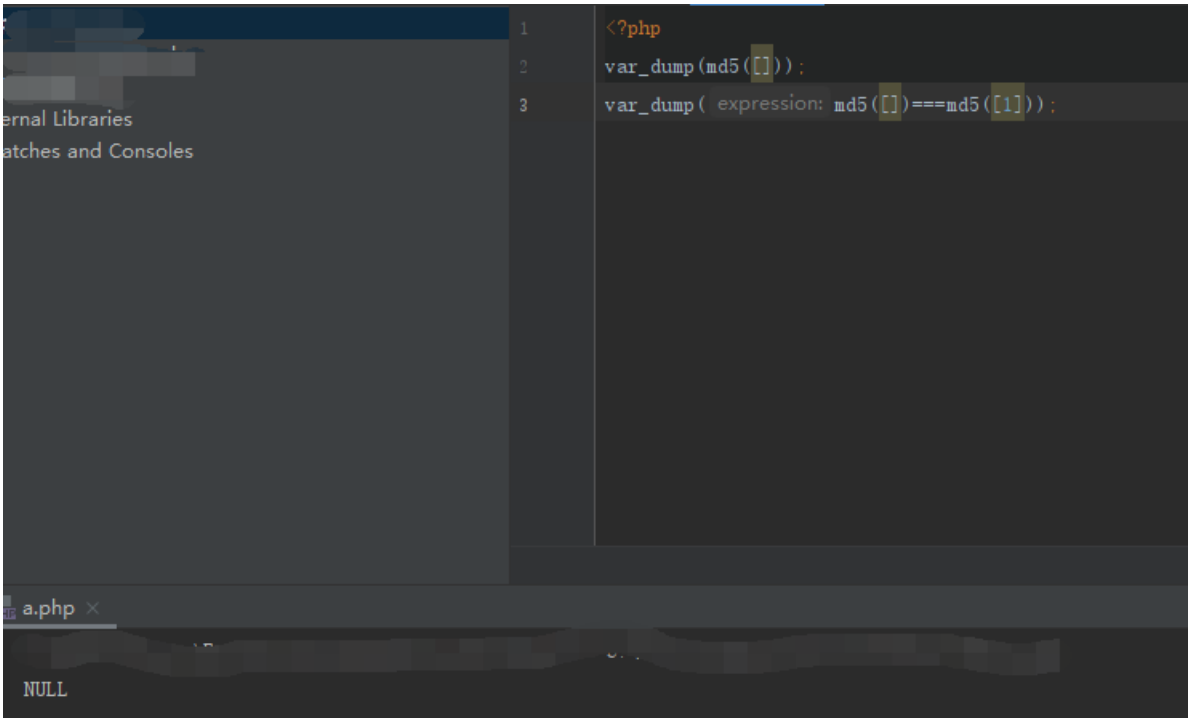
可参考bugku(md5 collision(NUPT_CTF))

例题3

实验环境：bugku：各种绕过

```php
<?php
highlight_file('flag.php');
$_GET['id'] = urldecode($_GET['id']);
$flag = 'flag{xxxxxxxxxxxxxxxxx}';
if (isset($_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!';

}
?>
```

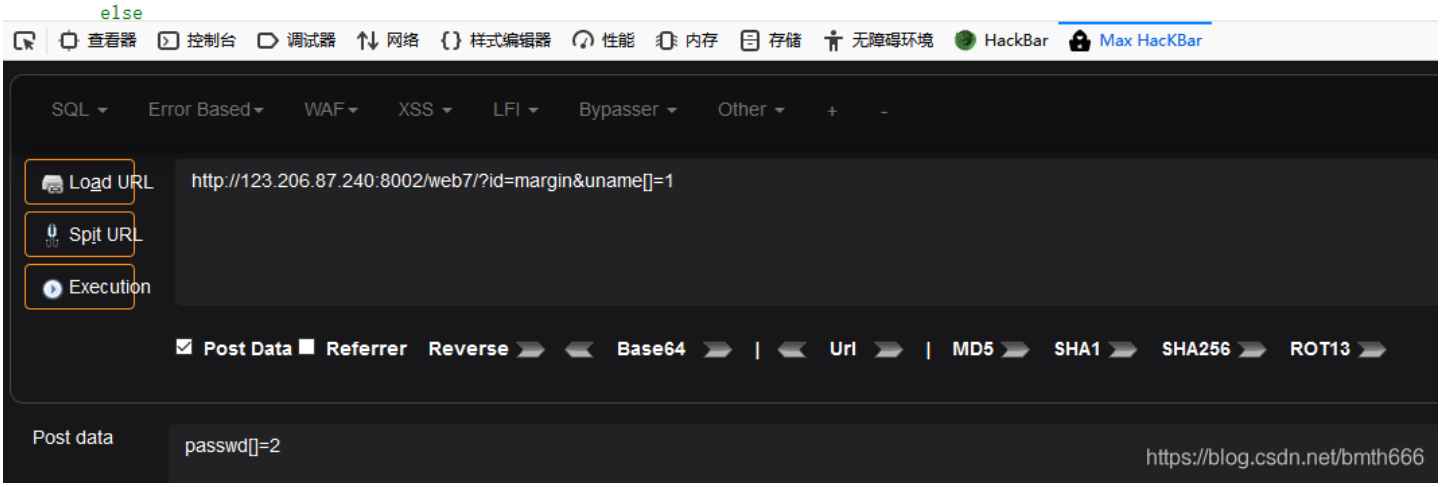

uname的值不能为passwd的值，但他们的哈希值要全相等，并且当id传值为margin，满足这三个条件，就返回flag，全相等就不能用0e绕过了。这里还有一个类型：数组。



```
bool(true)

Process finished with exit code 0
```


当传入数组时返回NULL，但两个的值还是相等的，传入

```php
<?php
highlight_file('flag.php');
$_GET['id'] = urldecode($_GET['id']);
$flag = 'flag{xxxxxxxxxxxxxxxxx}';
if (isset($_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
```

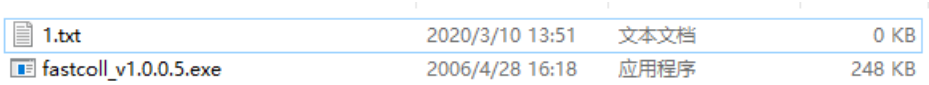


得到`Flag: flag{HACK_45hhs_213sDD}`

md5强碰撞

```php
if((string)$_POST['param1']!==(string)$_POST['param2']
&& md5($_POST['param1'])===md5($_POST['param2'])){
        die("success!");
        }
```

强网杯的一个md5强碰撞的题目，只能用两个值的md5相等来解，这里有一个工具可以生成，在参考文章可以下载，我就简单演示一遍。



首先创建一个空的1.txt文件，然后进入cmd

```
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Allowed options:
  -h [ --help ]           Show options.
  -q [ --quiet ]          Be less verbose.
  -i [ --ihv ] arg        Use specified initial value. Default is MD5 initial
```

```
                        value.
  -p [ --prefixfile ] arg Calculate initial value using given prefixfile. Also
                          copies data to output files.
  -o [ --out ] arg        Set output filenames. This must be the last option
                          and exactly 2 filenames must be specified.
                          Default: -o msg1.bin msg2.bin


[redacted]fastcoll_v1.0.0.5.exe -p 1.txt -o 2.txt 3.txt
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: '2.txt' and '3.txt'
Using prefixfile: '1.txt'
Using initial value: 0123456789abcdeffedcba9876543210

Generating first block: ..
Generating second block: S01...
Running time: 0.986 s
```


生成了2.txt和3.txt，判断一下md5值是否相等

```
[redacted]>certutil -hashfile "2.txt" MD5
MD5 的 2.txt 哈希:
696a78e7ba54396b4b28032b63faa83f
CertUtil: -hashfile 命令成功完成。

[redacted]>certutil -hashfile "3.txt" MD5
MD5 的 3.txt 哈希:
696a78e7ba54396b4b28032b63faa83f
CertUtil: -hashfile 命令成功完成。
```

发现是相等的，那么我们就将数据转换为url编码的形式

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from urllib import parse
>>> a = parse.quote(open('2.txt','rb').read())
>>> print(a)
%90/%0B%11%0D%1A2%C8%04%C5%F4%14%D7%8D%AA%02vC%1F%0Fs%B4%0D%06%24%BE~M%97%22%92%DFd%F1%CB%F9L%2B%3BA%CB%05Dy%166%7D0%94~
4g%5E%F0%8DZ%3Fu%CA%A4%CD%F09D%27%E8L%D1Z%40%B0%A8g%A4%C4%DCM%7D%EE%0A%82%8E%85L%11%86%16i%D1Z~G%EC%07%FEo%26e%C6%15%F2%
CC%07%CE%A8km7%98%B8%85%2CD%29%2C%18%05V%96%20W%E4%A3%1C%D1%F3%15%CD
>>> b = parse.quote(open('3.txt','rb').read())
>>> print(b)
%90/%0B%11%0D%1A2%C8%04%C5%F4%14%D7%8D%AA%02vC%1F%8Fs%B4%0D%06%24%BE~M%97%22%92%DFd%F1%CB%F9L%2B%3BA%CB%05Dy%16%B6%7D0%9
4~4g%5E%F0%8DZ%3Fu%CA%24%CD%F09D%27%E8L%D1Z%40%B0%A8g%A4%C4%DCM%7D%EE%0A%82%8E%85%CC%11%86%16i%D1Z~G%EC%07%FEo%26e%C6%15
%F2%CC%07%CE%A8km7%988%85%2CD%29%2C%18%05V%96%20W%E4%A3%9C%D1%F3%15%CD
>>>
```


也可以用师傅的脚本生成url编码的结果

```php
<?php
function  readmyfile($path){
    $fh = fopen($path, "rb");
    $data = fread($fh, filesize($path));
    fclose($fh);
    return $data;
}
echo '二进制hash '. md5( (readmyfile("2.txt")));
echo "<br><br>\r\n";
echo  'URLENCODE '. urlencode(readmyfile("2.txt"));
echo "<br><br>\r\n";
echo 'URLENCODE hash '.md5(urlencode (readmyfile("2.txt")));
echo "<br><br>\r\n";
echo '二进制hash '.md5( (readmyfile("3.txt")));
echo "<br><br>\r\n";
echo  'URLENCODE '.  urlencode(readmyfile("3.txt"));
echo "<br><br>\r\n";
echo 'URLENCODE hash '.md5( urlencode(readmyfile("3.txt")));
echo "<br><br>\r\n";
```

可参考文章：如何用不同的数值构建一样的MD5 - 第二届强网杯 MD5碰撞 writeup

弱类型相关函数：

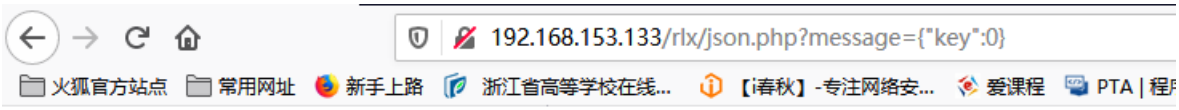
json绕过

json_decode — 对 JSON 格式的字符串进行解码

代码如下：

```php
<?php
highlight_file(__FILE__);
if (isset($_GET['message'])) {
    $message = json_decode($_GET['message']);
    $key ="admin********";
    if ($message->key == $key) {
        echo "flag";
    }
    else {
        echo "fail";
    }
}
else{
     echo "~~~~";
}
?>
```

由于是`==`我们可以传入0使`0==admin********`一样成立的




```php
<?php
highlight_file(__FILE__);
if (isset($_GET['message'])) {
        $message = json_decode($_GET['message']);
        $key ="admin********";
        if ($message->key == $key) {
                echo "flag";
        }
        else {
                echo "fail";
        }
}
else{
          echo "^^^^^";
}
?>
```

flag

switch绕过

如果switch是数字类型的case的判断时，switch会将其中的参数转换为int类型。可以看到显示为flag

```php
<?php
highlight_file(__FILE__);
$i ="3name";
switch ($i) {
case 0:
case 1:
case 2:
          echo "this is two";
          break;
case 3:
          echo "flag";
break;
}
?>
```

flag

strcmp绕过

strcmp — 二进制安全字符串比较

如果 str1 小于 str2 返回 < 0； 如果 str1 大于 str2 返回 > 0；如果两者相等，返回 0。strcmp函数比较字符串的本质是将两个变量转换为ascii，然后进行减法运算，根据运算结果来决定返回值。

```php
<?php
highlight_file(__FILE__);
$password = "***************";
if(isset($_GET['password'])){
    if (strcmp($_GET['password'], $password) == 0) {
        echo "Right!!!login success";
        exit();
    }else {
        echo "Wrong password..";
          }
}
?>
```

要使`strcmp($_GET['password'], $password) == 0`成立，我们又不知道password的值，只能从strcmp入手，发现：

```
php > echo strcmp('aaaa',[]);
PHP Warning:  strcmp() expects parameter 2 to be string, array given in php shel
l code on line 1
php > echo var_dump(strcmp('aaaa',[]));
PHP Warning:  strcmp() expects parameter 2 to be string, array given in php shel
l code on line 1
NULL
php > var_dump(0==NULL);
bool(true)
php >
```


传入数组的时候为NULL，而`0==NULL`的值又为真，成立。


```php
<?php
highlight_file(__FILE__);
$password = "***************";
if(isset($_GET['password'])){
        if (strcmp($_GET['password'], $password) == 0) {
                echo "Right!!!login success";
                exit();
        }else {
                echo "Wrong password..";
                    }
}
?>
```

Right!!!login success

in_array绕过

in_array

(PHP 4, PHP 5, PHP 7)
in_array — 检查数组中是否存在某个值

说明

```
in_array ( mixed $needle , array $haystack [, bool $strict = FALSE ] ) : bool
```

大海捞针，在大海（**haystack**）中搜索针（**needle**），如果没有设置 **strict** 则使用宽松的比较。

参数

needle
待搜索的值。

> **Note:**
> 如果 **needle** 是字符串，则比较是区分大小写的。

haystack
待搜索的数组。

strict
如果第三个参数 **strict** 的值为 **TRUE** 则 **in_array()** 函数还会检查 **needle** 的类型是否和 **haystack** 中的相同。

返回值

如果找到 **needle** 则返回 **TRUE**，否则返回 **FALSE**。

由于判断是以 `==` 来判断的，所以返回的值都为true

```php
<?php
highlight_file(__FILE__);
$array=[0,1,2,'3'];
var_dump(in_array('abc',$array));
var_dump(in_array('1bc',$array));
var_dump(in_array(3,$array));
?>
```

bool(true) bool(true) bool(true)

第一个 `'abc'` 为0，第二个 `'1bc'` 为1，第三个为3

array_search绕过

array_search — 在数组中搜索给定的值，如果成功则返回首个相应的键名
此函数可能返回布尔值 FALSE，但也可能返回等同于 FALSE 的非布尔值。

```php
<?php
highlight_file(__FILE__);
if(!is_array($_GET['test'])){exit();}
$test=$_GET['test'];
for($i=0;$i<count($test);$i++){
        if($test[$i]==="admin"){
                echo "error";
                exit();
        }
        $test[$i]=intval($test[$i]);
}
if(array_search("admin",$test)===0){
        echo "flag";
}
else{
        echo "false";
}
?>
```


这个题先判断是不是数组，然后再把数组中的内容一一遍历，所有内容都不能等于admin，类型也必须相同，然后转化成int型，然后再进行比较如果填入值与admin相同，则返回flag
我们在数组的第一个传0，就可以绕过了，值为：`array(1) { [0]=> int(0) }`


```php
<?php
highlight_file(__FILE__);
if(!is_array($_GET['test'])){exit();}
$test=$_GET['test'];
for($i=0;$i<count($test);$i++){
        if($test[$i]==="admin"){
                echo "error";
                exit();
        }
        $test[$i]=intval($test[$i]);
}
var_dump($test);
if(array_search("admin",$test)===0){
        echo "flag";
}
else{
        echo "false";
}
?>
```

array(1) { [0]=> int(0) } flag