1、知识点

知识点	说明
ini_set()	为一个配置选项设置值
error_reporting()	关闭所有PHP错误报告
header()	在header头部显示指定内容
die()	退出程序
strval()	获取变量对应的字符串类型的值
is_numeric()	判断变量是否为数字或数字字符串
intval()	获取变量对应的整数型的值
strrev()	反转字符串

2、源码

```
<?php

$info = "";
$req = [];
$flag="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";

ini_set("display_error", false); //为一个配置选项设置值
error_reporting(0); //关闭所有PHP错误报告

if(!isset($_GET['number'])){
   header("hint:26966dc52e85af40f59b4fe73d8c323a.txt"); //HTTP头显示hint
26966dc52e85af40f59b4fe73d8c323a.txt

   die("have a fun!!"); //die - 等同于 exit()

}

foreach([$_GET, $_POST] as $global_var) {  //foreach 语法结构提供了遍历数组的简单方式
    foreach($global_var as $key => $value) {
        $value = trim($value);  //trim - 去除字符串首尾处的空白字符（或者其他字符）
        is_string($value) && $req[$key] = addslashes($value); // is_string - 检测
变量是否是字符串，addslashes - 使用反斜线引用字符串
    }
}


function is_palindrome_number($number) {
    $number = strval($number); //strval - 获取变量的字符串值
    $i = 0;
    $j = strlen($number) - 1; //strlen - 获取字符串长度
    while($i < $j) {
```

```
        if($number[$i] !== $number[$j]) {
            return false;
        }
        $i++;
        $j--;
    }
    return true;
}


if(is_numeric($_REQUEST['number'])) //is_numeric — 检测变量是否为数字或数字字符串
{

   $info="sorry, you cann't input a number!";

}
elseif($req['number']!=strval(intval($req['number']))) //intval — 获取变量的整数值
{

     $info = "number must be equal to it's integer!! ";

}
else
{

     $value1 = intval($req["number"]);
     $value2 = intval(strrev($req["number"]));

     if($value1!=$value2){
          $info="no, this is not a palindrome number!";
     }
     else
     {

          if(is_palindrome_number($req["number"])){
              $info = "nice! {$value1} is a palindrome number!";
          }
          else
          {
             $info=$flag;
          }
     }

}

echo $info;
```

3、分析

1）传入number=1，在$_GET处下断点，开启Debug。这里首先对GET方法中的参数进行判断，检查是否传入number参数，此时我们传入的是number=1，所以这里不会进入该内容，不会退出程序。

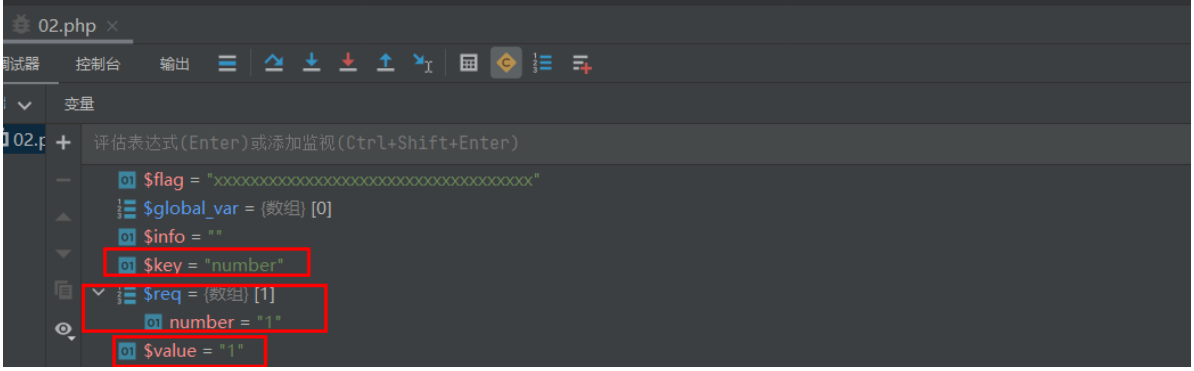
```php
if(!isset($_GET['number'])){   $_GET: {number => "1"}[1]
    header( header: "hint:26966dc52e85af40f59b4fe73d8c323a.txt");

    die("have a fun!!"); //die - 等同于 exit()

}
```

2）继续向下执行，使用了foreach函数对接收到的参数进行遍历，并赋值给$global_var。

随后通过$global_var，将键值分别赋给$key和$value，那么此时$key="number"，$value="1"。

接着对$value进行了首尾去空，并使用addslashes对$value中SQL注入的关键词进行转义赋值给数组$req，此时$req={"number"="1"}。

```php
foreach([$_GET, $_POST] as $global_var) {  //foreach 语法结构提供了遍历数组的简单方式    $global_var: [0]
    foreach($global_var as $key => $value) {    $key: "number"    $value: "1"
        $value = trim($value);  //trim - 去除字符串首尾处的空白字符（或者其他字符）
        is_string($value) && $req[$key] = addslashes($value); // is_string - 检测变量是否是字符串，addslash
    }
}
```



3）继续向下执行，使用is_numeric函数判断number参数是否为数字或者纯数字类型的字符串，由于我们传入的是1，那么这里就会被直接过滤掉，并且返回报错信息。

```php
if(is_numeric($_REQUEST['number'])) //is_numeric - 检测变量是否为数字或数字字符串
{

    $info="sorry, you cann't input a number!";

}
```

```php
echo $info;   $info: "sorry, you cann't input a number!"
```

4）目前已知两个判断，需要传入number参数，并且number参数的值不能是数字或者数字型字符串。这里我们继续看代码逻辑，继续总结：

- 46行的判断跟上面类似，直接忽略这个判断条件。

```
elseif($req['number']!=strval(intval($req['number']))) //intval - 获取变量的整数值
{

    $info = "number must be equal to it's integer!! ";

}
```

5）进入到else中，将$req数组中number的值转成整型，并赋值给$value1；将$req数组中number的值反转顺序然后转成整型，并赋值给$value2；如果两个值不等，就报错。

```
else
{

    $value1 = intval($req["number"]);
    $value2 = intval(strrev($req["number"]));

    if($value1!=$value2){
        $info="no, this is not a palindrome number!";
    }
```

6）进入下一个else中，其中调用is_palindrome_number函数对$req中的number数组进行处理，并返回boolean值，只有当值为false时，才能拿到flag。

```
else
{

    if(is_palindrome_number($req["number"])){
        $info = "nice! {$value1} is a palindrome number!";
    }
    else
    {
       $info=$flag;
    }
}
```

7、跟进到is_palindrome_number函数。这里对number值的长度进行判断，只要长度-1之后还比0大，也就是长度大于1，就可以返回false。

```php
function is_palindrome_number($number) {
    $number = strval($number); //strval - 获取变量的字符串值
    $i = 0;
    $j = strlen($number) - 1; //strlen - 获取字符串长度
    while($i < $j) {
        if($number[$i] !== $number[$j]) {
            return false;
        }
        $i++;
        $j--;
    }
    return true;
}
```

8、这里总结一下：

1) is_numeric($_REQUEST['number'])，也就是不能为纯数字，可以以%00开头进行绕过。

2) $req['number'] == strval(intval($req['number']))，也就是转换后的值要想等，这里传入值为纯数字即可。

3) intval($req['number']) == intval(strrev($req['number']))，意思时传入的值顺序反转过后，还是相等，这里传入的两个数字一样即可。

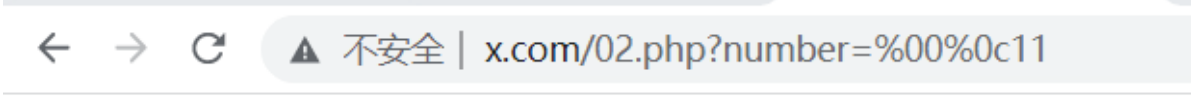
4) is_palindrome_number()返回false，这里在数字前加一个字符就行。

5) 也就说传入一个%00开头，然后加一个字符（不会被转义并且转成字符串或者整型，值都为空的字符），最后加上两个一样的字符就行。

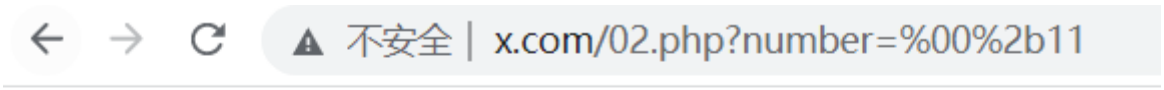
6) 可以是这样：

?number=%00%0b11

?number=%00%0c11



XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

4、利用

```
?number=%00%2b11
?number=%00%0c11
```