封神台正则表达式题目

原创

炎凰先生 于 2020-07-24 08:15:38 发布 178 收藏

分类专栏： Web安全微专业

本文链接：https://blog.csdn.net/qq_40941912/article/details/107552642

一、原理

1. 正则表达式，又称规则表达式。正则表达式通常被用来检索、替换那些符合某个模式(规则)的文本。
2. PHP中的相关函数：
   preg_match_all();——匹配字符串中所有的符合正则表达式的字符串，返回匹配次数。
   preg_replace();——将符合正则表达式的字符串转换成新的字符串。返回替换后的结果。
   preg_match();——匹配字符串中所有的符合正则表达式的字符串，第一次匹配成功后返回。
3. 要熟悉并掌握正则表达式的字符、关键字、限定符、修饰符。

- 常用转义字符：
  - 数字：\d
  - 非数字：\D
  - 空白字符（空格、制表符、换页符等）：\s
  - 非空白字符：\S
  - 单词字符（26个英文字母+数字+下划线_）：\w
  - 非单词字符：\W

- 自定义字符结合
  - 字符集合：[单个字符或字符区间]，用于匹配集合内字符
    如：
    - [a-z]表示a-z这26个小写字母
    - [0-9a-z]表示0-9这10个数字和a-z26个小写字母
    - [135a-b]表示包含数字1、3、5和字母a-b这8个字母

- 注意：两个不同字符段间请勿使用,隔开。

- 非集：[^单个字符或字符区间]，用于匹配非集合内字符。

  如：

  - [^0-9]表示匹配所有非数字字符。
  - [^a-zA-Z]表示匹配所以非字母字符。

正则表达式的语法-关键字

() => 和数学一样很像，代表这是一个整体。

^ => 匹配输入字符串的开始位置

$ => 匹配输入字符串的结尾位置

. => 通配符[代表任意字符][不匹配换行]

* => 匹配0次或者多次

+ => 匹配1次或者多次

\ => 转义字符

| => 两项之间的一个选择。

限定符：

{n} => 例如: 0{8} 意思是指 只有连起来8个0才会被匹配

{n,} =>例如： 0{2,} 意思是 只要有2个0及其以上的就会被匹配

{n,m} => 例如： 0{2,4} 意思是最少匹配2个0，最多匹配4个0

注：被匹配时，默认匹配最多的次数

修饰符：

/i => 不区分大小写

/^ => 匹配规则必须从头开始匹配

/A => 匹配规则必须从头开始匹配
/s => .将匹配一切字符
/x => 正则表达式中的空白字符会被忽略

二、作业

(一)题目

在靶场中，找到正则表达式题目，并得到题目中的flag
题目代码：

```php
<?php
$key='flag{******************************}';
$Regular= preg_match("/zkaq.*key.{2,9}:\/.*\/(key*key)/i", trim($_GET["id"]), $match);
if( $Regular ){
  die('key: '.$key);
}
```

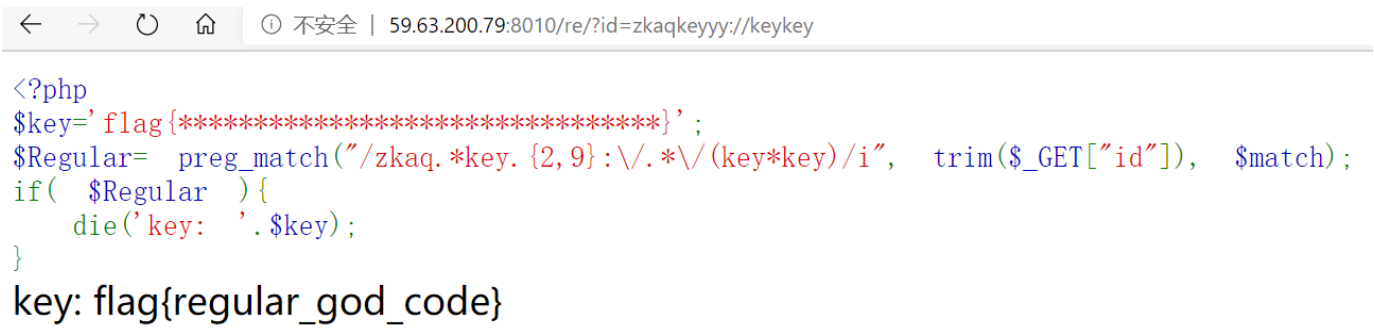
(二)分析

1. 题目最终目的是得到对应的flag。由`$key='flag{******************************}'`和`die('key: '.$key);`;可知我们可以通过控制程序，使程序运行`die('key: '.$key);`,从而得到flag。
2. 然后我们看到if语句的判定条件是$Regular$，因此，需要控制程序使得Regular为真。
3. 分析`$Regular= preg_match("/zkaq.*key.{2,9}:\/.*\/(key*key)/i", trim($_GET["id"]), $match);`，若使$Regular$为真，需要使得$preg\ \ atch$函数返回为1，即传递的id参数‘$trim(_GET[“id”])$中含有符合正则表达式$/zkaq.key.\{2,9\}:./(key*key)/i$`的子串。
4. 分析`/zkaq.*key.{2,9}:\/.*\/(key*key)/i`，设计符合条件的id参数字符串。

- 修饰符`\i`规定匹配时不区分大小写;
- `zkaq`是正常的字符，因此设计的id参数字符串中首先可以是`zkaq`;
- `.*`是匹配任意字符任意次，因此设计为空字符;
- `key`为固定的字符串;
- `.{2,9}`是匹配任意字符串2-9次，可以设计为`yy`;
- `:`为固定字符;
- `\/`为转义匹配`/`;
- `(key*key)`是一个整体，前面匹配字符串`ke`，后面匹配字符串`key`，中间匹配任意次y，我们可以简单的设计为`keykey`;
- 经过上面分析，我们得出字符串`zkaqkeyyy://keykey`。

(三)结果

1. 将页面URL中的id参数修改为设计好的字符串`zkaqkeyyy://keykey`，请求页面，我们就可以得到flag了。



将key对应的值提交，显示成功。