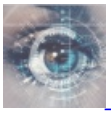
i春秋 Misc Web 爆破-1 wp

原创

Garybr0 于 2021-01-10 11:06:17 发布 107 收藏 2

分类专栏： CTF writeup PHP函数 文章标签： i春秋 web



2021.1.10

~~网安菜鸡今天来划水了！~~

CTF题目属于萌新入门级，写下WP仅供自己总结练习，大佬请自行绕路，另外如果有师傅愿意有**每日轻松一笑**环结，还望不吝赐教。【狗头】

题目提示：
flag就在某六位变量中。

点开链接，是一段PHP源码：

```php
<?php
include "flag.php";
$a = @$_REQUEST['hello'];
if(!preg_match('/^\w*$/',$a )){
    die('ERROR');
}
eval("var_dump($$a);");
show_source(__FILE__);
?>
```


首先第一行 include “flag.php” 表示包含一个flag.php的文件。
第二行$ a = @$ _REQUEST[‘hello’];命名一个变量a，让他接收超全局变量$ _REQUEST[‘hello’] 。这里有一个知识点，就是什么是超全局变量？

超全局变量：

超全局变量 — 超全局变量是在全部作用域中始终可用的内置变量。
PHP 中的许多预定义变量都是“超全局的”，这意味着它们在一个脚本的全部作用域中都可用。在函数或方法中无需执行 global $variable; 就可以访问它们。超全局变量包括：
$GLOBALS
$_SERVER
$_GET
$_POST
$_FILES
$_COOKIE
$_SESSION
$_REQUEST
$_ENV
本题涉及了两个超全局变量，分别是$GLOBALS和$_REQUEST。

$GLOBALS

$GLOBALS — 引用全局作用域中可用的全部变量。一个包含了全部变量的全局组合数组。变量的名字就是数组的键。
具体使用方法：来自PHP Document

范例

示例 #1 *$GLOBALS* 范例

```php
<?php
function test() {
    $foo = "local variable";

    echo '$foo in global scope: ' . $GLOBALS["foo"] . "\n";
    echo '$foo in current scope: ' . $foo . "\n";
}

$foo = "Example content";
test();
?>
```

以上例程的输出类似于：

```
$foo in global scope: Example content
$foo in current scope: local variable
```


$_REQUEST

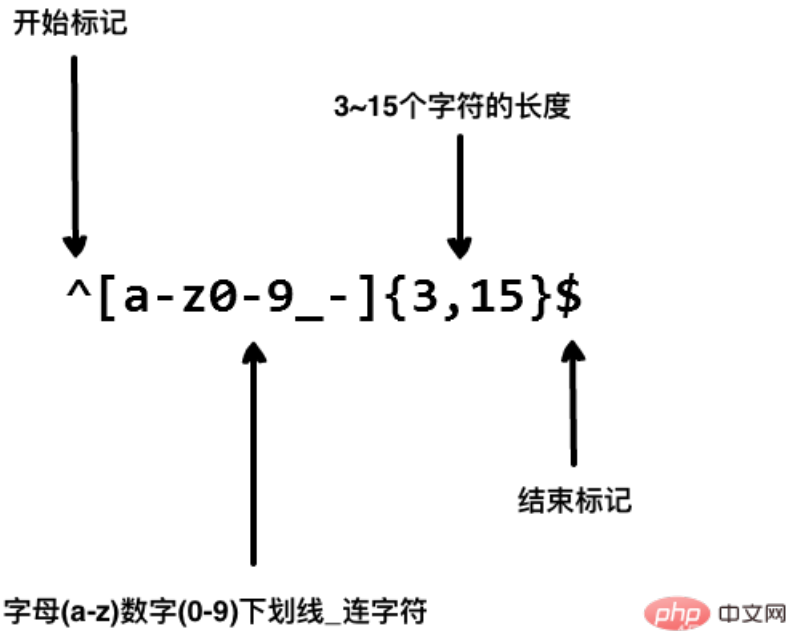
$_REQUEST — HTTP Request 变量。默认情况下包含了 $_GET，$_POST 和 $_COOKIE 的数组。

第三行，是用正则表达式过滤，如果传入的参数不符合格式要求，则die(‘error’);

@写在变量前面，是忽略错误提示的意思。@用于抑制警告输出，通常用在PHP数据库连接数据库或接收网络传参。

正则表达式

PHP中文网



正则表达式在匹配字符串时，遵循以下2个基本原则：

1.最左原则：正则表达式总是从目标字符串的最左侧开始，依次匹配，直到匹配到符合表达式要求的部分，或直到匹配目标字符串的结束。
2.最长原则：对于匹配到的目标字符串，正则表达式总是会匹配到符合正则表达式要求的最长的部分；即贪婪模式。

那么通用原子和元字符有哪些呢？

- \d匹配一个数字字符。等价于 [0-9]。
- \D匹配一个非数字字符。等价于 [^0-9]。
- \f匹配一个换页符。等价于 \x0c 和 \cL。
- \n匹配一个换行符。等价于 \x0a 和 \cJ。
- \r匹配一个回车符。等价于 \x0d 和 \cM。
- \s匹配任何空白字符，包括空格、制表符、换页符等等。等价于 [\f\n\r\t\v]。
- \S匹配任何非空白字符。等价于 [^ \f\n\r\t\v]。
- \t匹配一个制表符。等价于 \x09 和 \cI。
- \v匹配一个垂直制表符。等价于 \x0b 和 \cK。
- \w匹配包括下划线的任何单词字符。等价于'[A-Za-z0-9_]'。
- \W匹配任何非单词字符。等价于 '[^A-Za-z0-9_]'。
- \xn匹配 n，其中 n 为十六进制转义值。十六进制转义值必须为确定的两个数字长。例如，'\x41' 匹配 "A"。'\x041' 则等价于 '\x04' & "1"。正则表达式中可以使用 ASCII 编码。
- \nm标识一个八进制转义值或一个向后引用。如果 \nm 之前至少有 nm 个获得子表达式，则 nm 为向后引用。如果 \nm 之前至少有 n 个获取，则 n 为一个后跟文字 m 的向后引用。如果前面的条件都不满足，若

n 和 m 均为八进制数字 (0-7)，则 • \nm 将匹配八进制转义值 nm。

- \nml如果 n 为八进制数字 (0-3)，且 m 和 l 均为八进制数字 (0-7)，则匹配八进制转义值 nml。
- \un十六进制数字表示的 Unicode 字符。例如， \u00A9 匹配版权符号(?)。
- . 匹配除 “\n” 之外的任何单个字符

使用命名子组

```php
<?php

$str = 'foobar: 2008';

preg_match('/(?P<name>\w+): (?P<digit>\d+)/', $str, $matches);

/* 下面例子在php 5.2.2(pcre 7.0)或更新版本下工作，然而，为了后向兼容，上面的方式是推荐写法. */
// preg_match('/(?<name>\w+): (?<digit>\d+)/', $str, $matches);

print_r($matches);

?>
```

执行结果如下所示:

```
Array
(
    [0] => foobar: 2008
    [name] => foobar
    [1] => foobar
    [digit] => 2008
    [2] => 2008
)
```


正则表达式是一块非常重要的内容，稍后要多加练习以熟练运用。
本题目中传入$*REQUEST就会ERROR，因为\w匹配包括下划线的任何单词字符。等价于'[A-Za-z0-9]*'。

接下来是eval(“var_dump($$a);”);
代表执行var_dump这个函数：
var_dump() 函数用于输出变量的相关信息。

实例

实例

```php
<?php
$a = array(1, 2, array("a", "b", "c"));
var_dump($a);
?>
```

输出结果为：

```
array(3) {
  [0]=>
  int(1)
  [1]=>
  int(2)
  [2]=>
  array(3) {
    [0]=>
    string(1) "a"
    [1]=>
    string(1) "b"
    [2]=>
    string(1) "c"
  }
}
```


var_dump() 等价于 print_r。
最后一行show_source(__FILE__);代表能把PHP代码打印到网页上，同样能实现这个功能的还有highlight_file()。

解题过程：把全局变量GLOBALS通过hello赋值给$a,就变成了$GLOBALS，然后打印全局变量，就能找到flag：