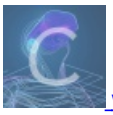
Bugku CTF 代码审计 writeup（未完待续）

原创

KRDecad3 于 2018-05-21 22:34:41 发布 1643 收藏 3

分类专栏： writeup 文章标签： Bugku writeup

本文链接：https://blog.csdn.net/KRDecad3/article/details/80399137



Bugku CTF 代码审计 writeup

0x01extract变量覆盖

http://120.24.86.145:9009/1.php

```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()函数功能：从数组中将变量导入到当前的符号表。使用数组键名作为变量名，使用数组键值作为变量值。
extract(array,extract_rules,prefix)
题中给出extract($_GET)，相当于：
$shiyan = $_GET['shiyan']
$flag = $_GET['flag']。
另外，isset()函数判断$shiyan变量是否设置，

```php
$var = '';
// 结果为 TRUE，所以后边的文本将被打印出来。
if (isset($var)) {
    echo "This var is set so I will print.";
}
```


像这样，即使变量为空，isset()函数也为true。
`payload:?shiyan=&flag=`
或者利用php://伪协议：`payload:?shiyan=&flag=php://input`

0x02strcmp比较字符串

http://120.24.86.145:9009/6.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';
}
?>
```


strcmp(str1, str2)比较两个字符串大小，若是非字符串（例如数组）比较，则会出错。在5.3之前的php中，显示了报错的警告信息后，将return 0。

`payload:?a[]=1`

0x03urldecode二次编码绕过

http://120.24.86.145:9009/10.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

";
}
?>
```


eregi()在一个字符串中搜索指定的模式的字符串，搜索不区分大小的正则匹配。
urldecode()解码以编码的URL字符串，但是$_GET[]会将参数解码一次。

> **Warning** 超全局变量 `$_GET` 和 `$_REQUEST` 已经被解码了。对 `$_GET` 或 `$_REQUEST` 里的元素使用 **urldecode()** 将会导致不可预计和危险的结果。
>
>

若本题将“hackerDJ”进行两次URL编码，在eregi()中比较”hackerDJ”与$_GET[id]是否相同，$_GET会解码一次，从二次URL编码变成一次URL编码；urldecode()会将$_GET[id]从二次URL编码变成一次URL编码，赋值给$_GET[id]，当$_GET[id]与“hackerDJ”比较时，$_GET[id]再从一次URL编码解码，最后比较相等得到flag。
将“hackerDJ”进行两次URL编码。

```
payload：?id=%25%36%38%25%36%31%25%36%33%25%36%62%25%36%35%25%37%32%25%34%34%25%34%61
```

0x04md5()函数

http://120.24.86.145:9009/18.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';
}
?>
```


md5()计算字符串的MD5散列值。
md5()函数不能处理数组，使用数组绕过，md5(array)会返回null。
`payload1:?username[]=1&password=2`

0x05数组返回NULL绕过

http://120.24.86.145:9009/19.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';
}
?>
```


strpos(string, find[, start])查找字符串在另一字符串中第一次出现的位置。
ereg()正则匹配，需使password中只含有英文字母和数字，又因为strpos()需要匹配“–”才能得到flag，可以使用数组绕过ereg()和strpos()。
ereg()只能处理字符，而password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!==false。
strpos()的参数同样不能够是数组，所以返回的依旧是null，null!==false也是正确。
`payload:?password[]=1`
思路二：ereg()可以进行%00截断，绕过正则匹配。
`payload:?password=1%00--`

0x06弱类型整数大小比较绕过

http://120.24.86.145:9009/22.php

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


is_numeric(var)检测变量是否为数字或数字字符串，是则返回true，否则返回false。

is_numeric()对于空字符%00，无论%00放在前面还是后面都可以判断为非数值，而空格%20只能放在数值后面，实质上都是弱类型转换。

`payload1:?password=1337%00`

`payload2:?password=1337%20`

`payload3:?password=1337a`

0x07sha()函数比较绕过

http://120.24.86.145:9009/7.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!

';
?>
```


同md5()一样，sha1()函数也无法处理数组，因此可以构造数组绕过。

`payload:?name[]=1&password=2`

0x08md5加密相等绕过

http://120.24.86.145:9009/13.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";}
?>
```


md5生成的以“0e”开头的哈希值都解释为0，所以PHP在判断时会认为相同。

`payload:?a=240610708`

0x09十六进制与数字比较

http://120.24.86.145:9009/20.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);
?>
```


ord()返回字符串中首个字符的ASCII值。
题目中会要求输入的password中不能有0~9数字，并且还需要判断$number==$_GET[password]。所以将$number=3735929054转换为16进制“deadc0de”，再在前面加上0x表示16进制。

```
payload:?password=0xdeadc0de
```

0x10ereg正则%00截断

http://120.24.86.145:9009/5.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

';
}
}
```


题目通过ereg()函数判断password中是否只含有字母或数字，进而限制password长度小于8，并且值小于9999999；在此前提下用strpos()函数搜索password中是否含有“-”，若是则输出flag。

解法一，利用数组绕过strpos()函数。

`payload:?password[]=9999999999`

解法二，利用ereg()%00截断漏洞，在用科学记数法来构造1e9满足strlen($_GET[‘password’]) < 8 && $_GET[‘password’]> 9999999这个条件，再加上“-”来满足strpos()的条件。

`payload:?password=1e9%00*-*`

0x11strpos数组绕过

http://120.24.86.145:9009/15.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~';
}
?>
```


构造数组绕过，并且值含有数字。

`payload:?ctf[]=1`

0x12数字验证正则绕过

http://120.24.86.145:9009/21.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;
}
}
?>
```


直接POST一个password就可以，并且小于12个字符。

`payload: POST:password=1`