ctf php正则截断,BugkuCTF—代码审计—WriteUp(持续更新)

转载

伟星啊 于 2021-03-17 08:22:24 发布 89 收藏
文章标签： ctf php正则截断

BugkuCTF—代码审计—WriteUp(持续更新)

extract变量覆盖

首先分析下代码：

extract函数：

image.png

http://123.206.87.240:9009/1.php

```
$flag='xxx';
extract($_GET);
if(isset($shiyan)) //shiyan这个变量不能为空
{
$content=trim(file_get_contents($flag)); //trim为去除多余的空格
if($shiyan==$content)
{
echo'flag{xxx}';
}
else
{
echo'Oh.no';
}
}
?>
```

根据以上代码，

_content 才能得到flag，进行构造下：

payload：

emmm~~~~~

image.png

strcmp比较字符串

http://123.206.87.240:9009/6.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';

}

?>

if (strcmp(

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。

所以必须让他们两个相等才可以，用数组绕过试下

payload：

image.png

urldecode二次编码绕过

http://123.206.87.240:9009/10.php

if(eregi("hackerDJ",$_GET[id])) { //eregi为正则匹配前面的$get

echo("

not allowed!

");

exit();

}

$_GET[id] = urldecode($_GET[id]);

if($_GET[id] == "hackerDJ")

{

echo "

Access granted!

";

echo "

flag

";

}

?>

首先：

if(eregi("hackerDJ",$_GET[id])) 不能让他匹配成功

其次

_GET[id]);

if($_GET[id] == "hackerDJ")

进行了urldecode解码，解码后为hackerDJ

$get 进行传参的时候一般都进行了一次解码，下面又进行了一次解码。本题目也提示了为为二次解码

所以我们将 hackerDJ 中的一个字母进行二次编码即可

用J吧

j = %4a

% = %25

hackerD%254a

payload如下:

image.png

md5()函数

http://123.206.87.240:9009/18.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';

}

?>

if (

[redacted]_GET['password'])

username不能等于password

if (md5(

[redacted]_GET['password'])

username 的md5 必须等于 password 的md5

php有一个弱类型漏洞，以下0e开头的md5值全部相等

失败。。。刚忘了一个事情

image.png

他是全等 所以这个漏洞没法用

所以直接用数组绕过吧！！

进行payload构造：

image.png

QNKCDZO

0e830400451993494058024219903391

QNKCDZO

240610708

s878926199a

0e545993274517709034328855841020

s155964671a

0e342768416822451524974117254469

s214587387a

0e848240448830537924465865611904

s214587387a

0e848240448830537924465865611904

s878926199a

0e545993274517709034328855841020

s1091221200a

0e940624217856561557816327384675

s1885207154a

0e509367213418206700842008763514

数组返回NULL绕过

http://123.206.87.240:9009/19.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';
}
?>
```

又是数组绕过。。。

payload：

image.png

弱类型整数大小比较绕过

http://123.206.87.240:9009/22.php

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

is_numeric($temp)?die("no numeric"):NULL;

这一句中 要求不能是数字

if($temp>1336)

还要是数字大于1336

php弱类型漏洞

构造payload：

image.png

sha()函数比较绕过

```
$flag = "flag";
if (isset($_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!
';
?>
```

首先

if (

_GET['password'])

两个不能相等

然后

if (sha1(

_GET['password']))

两个必须全等才能得到flag

试下数组绕过。。。

果然 payload：

image.png

md5加密相等绕过

$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";}

?>

好像是0e开头那个漏洞

试一下吧

image.png

十六进制与数字比较

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);
?>
(
                                                    _temp) 返回flag
if ( (
                                                      _one) && (
                                                       _nine) )
```

大于等1 小于等于9 都不可以

要想temp = number =3735929054

用十六进制绕过

0xdeadc0de = 3735929054

payload：

image.png

变量覆盖

题目挂了！！

ereg正则%00截断

```
$flag = "xxx";
if (isset ($_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

';

}

}

?>

①：if (ereg ("^[a-zA-Z0-9]+

_GET['password']) === FALSE)

password匹配必须 a-zA-Z0-9 之中

这个可以用%00截断

②：if (strlen(

Tex parse error!_GET['password'] > 9999999)

password小于8位数 且 大于9999999

这个用数组绕过

三：if (strpos ($_GET['password'], '-') !== FALSE)

数组绕过同时也可以绕过这个

构造payload：

image.png

strpos数组绕过

$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~';

}

?>

if (strpos ($_GET['ctf'], '#biubiubiu') !== FALSE)

可以用数组绕过

if (@ereg ("^[1-9]+

$_GET['ctf']) === FALSE)

这个正则匹配试一下

绕过了 emmmmm~~~

image.png

数字验证正则绕过

```
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;
}
}
?>
```

题目的代码倒是挺长。但是while (TRUE) 后面貌似没有用到

满足：

```
if (0 >= preg_match('/^[[:graph:]]{12,}
                                    .password)) //preg_match — 执行一个正则表达式匹配
{
echo 'flag';
exit;
}
```

即可得到flag

preg_match()返回 pattern 的匹配次数。 它的值将是0次(不匹配)或1次，因为preg_match()在第一次匹配后 将会停止搜索。preg_match_all()不同于此，它会一直搜索subject 直到到达结尾。 如果发生错误preg_match()返回FALSE

也就是随便构造不匹配就行了吧。。。试一下

噗！！

image.png

简单的waf

还是打不开的状态！！