

# 小白首次打CTF，思路详解记录

原创

雾土  于 2019-10-11 15:30:31 发布  956  收藏 3

文章标签: [CTF Brup](#)

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/weixin\\_37990514/article/details/102502233](https://blog.csdn.net/weixin_37990514/article/details/102502233)

版权

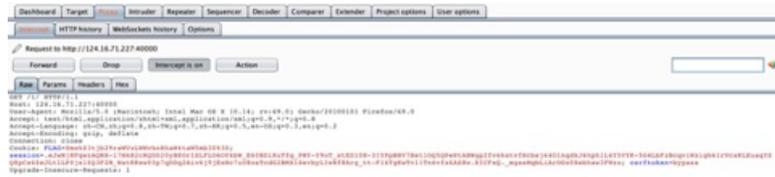
## 引言

小编第一次接触CTF, 有点儿刺激啊/捂脸 一脸懵逼的那种刺激, 但是在好心人的帮助下终于完美解决了三个问题!

## 正文

题一:

访问http://124.16.71.227:40000/1/, 看到题目“找flag”，用brup查看报文



可以看到cookie项包含FLAG信息:

=ZmxhZ3tjb29raWVzLWNVbnRhaW4taW5mb30%3D

这是一串base64编码的加密信息，先用url解码，将末尾的%3D转换为=，再用base64解码刚刚url解码得到的字符串，找到了flag。



题二:

访问http://124.16.71.227:40000/2/, 看到页面中有登录输入框，于是随便输入用户名1，点击login，看到:

# Hello, 1

No flag here. Unless you are root.

Go back.

于是换成root登录，发现仍然不合法，



于是再次使用用户名1登录，使用brup拦截请求报文，手动将请求报文中的登录名改为root，



再次尝试发现仍然存在错误，提示hostname不对，

# Hello, root

Your host is invalid, localhost(127.0.0.1) only.

[Go back.](#)

[https://blog.csdn.net/weixin\\_37990514](https://blog.csdn.net/weixin_37990514)

于是手动将请求报文中的name改为root, host改为127.0.0.1,

Request to http://124.16.71.227:40000

Forward Drop Intercept is on Action

Raw Params Headers Hex

Name	Value
POST	/2/index.php HTTP/1.1
Host	127.0.0.1
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:69.0) Gecko/20100101 Firefox/69.0
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
Content-Type	application/x-www-form-urlencoded
Content-Length	9
Connection	close
Referer	http://124.16.71.227:40000/2/index.php
Cookie	session=.ejwNj8FqwzAQRH-l7NkH2cRQDD20yBEOriSLFLO6OYk
Upgrade-Insecure-Requests	1

name=root

[https://blog.csdn.net/weixin\\_37990514](https://blog.csdn.net/weixin_37990514)

最终得到了flag。

## Hello, root

flag{client-side-and-server-side-bypass}

题三:

访问http://124.16.71.227:40001/csrf, 题目是说下面表单被django csrf中间件保护, 尝试点击get flag按钮, 发现:

## Django website

The form below is protected by *Django CSRF MIDDLEWARE*  
can you bypass it?

Get flag

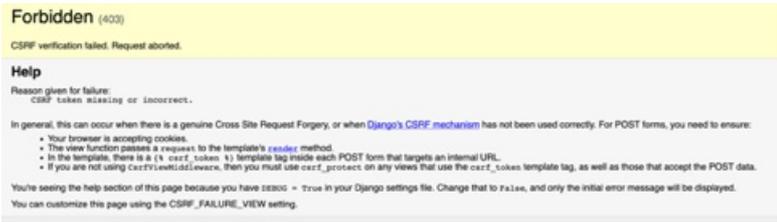
Your token is wrong

没有思路, 于是查看源代码, 发现了注释信息:

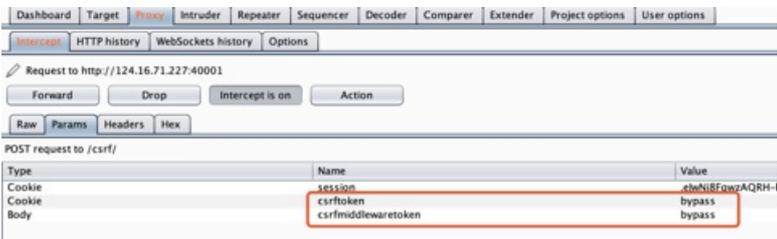
```
[root@eshop-cache01 6379]# cat appendonly.aof
*2
$6
SELECT
$1
0
*3
$3
set
$2
k1
$2
v1
*3
$3
set
```

\$2  
k2  
\$2  
v2

于是将请求报文中的csrfmiddlewaretoken的值改成bypass，点击get flag按钮，发现禁止访问，



通过阅读禁止信息，发现，csrftoken值缺失或者不正确，因为之前修改了csrfmiddlewaretoken的值，猜测这两个值应该一样，于是手动将请求报文中的csrftoken改为bypass，



发送报文，得到了flag。

## Django website

The form below is protected by *Django CSRF MIDDLEWARE*  
can you bypass it?

flag{django\_csrf\_bypassed}

[https://blog.csdn.net/weixin\\_37990514](https://blog.csdn.net/weixin_37990514)