Web application testing using DVWA

What is Damn Vulnerable Web App (DVWA)?

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.

Installation of DVWA:

Step 1: Get DVWA package.

Step 2:

Install essential packages if you do not install LAMP when installing the Ubuntu Server. sudo apt-get install apache2 mysql-server php5 unzip php5-mysql php-pear*

Step 3:

Extract DVWA.

```
sudo cp v1.0.8.zip /var/www/
unzip v1.0.8.zip
```

Step 4:

```
sudo nano /var/www/DVWA/config/config.inc.php
```

Change the "db_password" to the captioned root password, e.g. password.

```
sudo nano /etc/apache2/conf.d/php.ini
```

change "allow_url_include = Off" to "allow_url_inclue = On".

```
sudo chmod -R 777 /var/www/DVWA/hackable/uploads/
```

Step 5:

Point your Firefox to "http://192.168.0.10/DVWA/setup.php" to create/reset database.

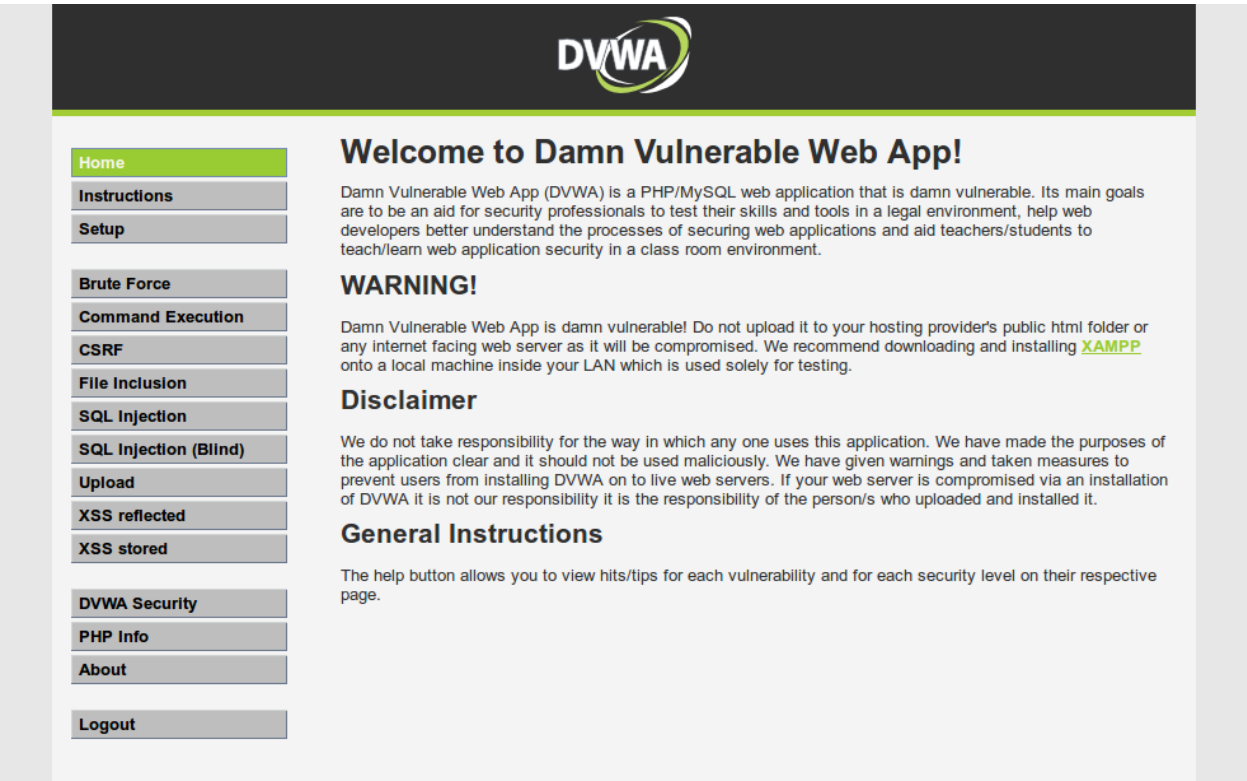
* where 192.168.0.10 is the IP address of the Ubuntu Server or Local Host

Step 6:

Then point your Firefox to "http://192.168.0.10/DVWA/index.php".

User name is "admin" and Password is "password".

Screenshots:



Manual SQL injection using DVWA

What is a SQL Injection?

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.

What is SQL Injection Harvesting?

SQL Injection Harvesting is where a malicious user supplies SQL statements to render sensitive data such as usernames, passwords, database tables, and more.

SQL Injection

1) Login to DVWA

1. Start up Firefox
2. Place http://**localhost**/dvwa/login.php in the address bar.
3. Login: admin
4. Password: password
5. Click on Login

2) Set DVWA Security Level

1. Click on DVWA Security, in the left hand menu.
2. Select "low"

SQL Injection Menu

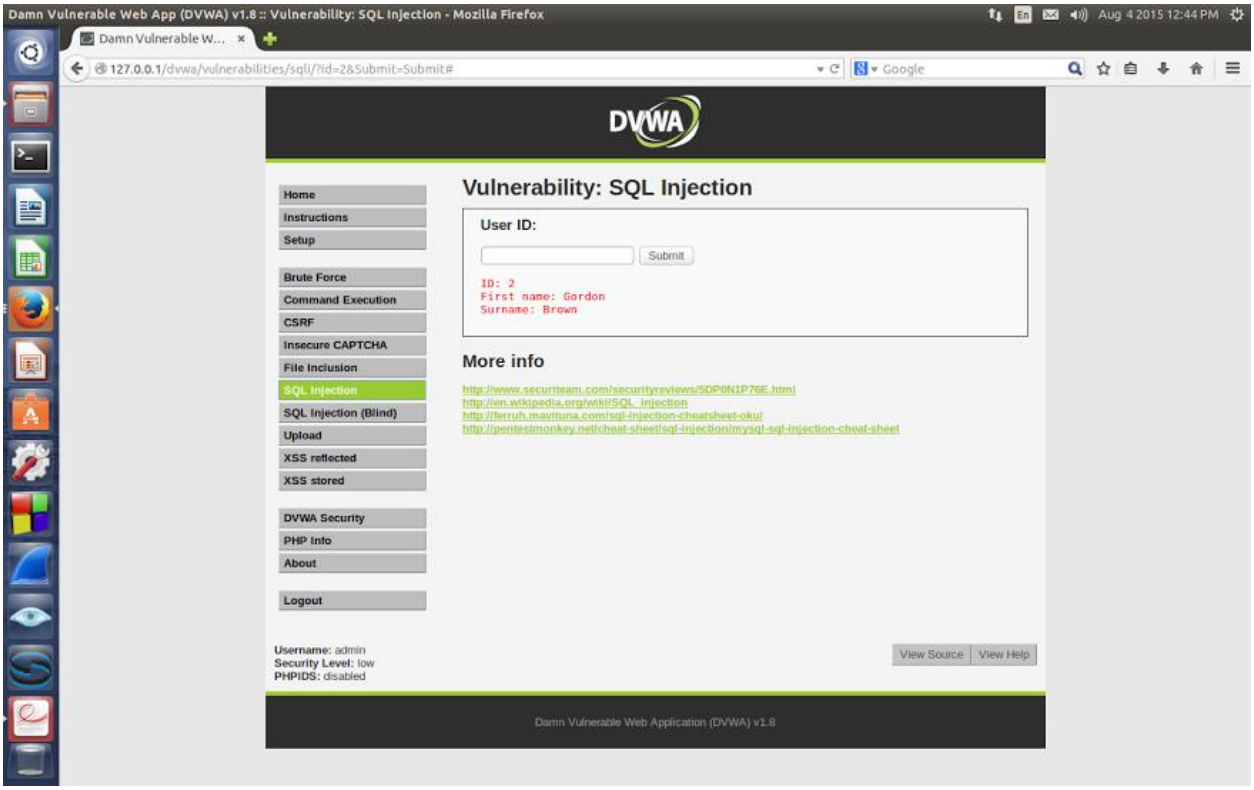
Select "SQL Injection" from the left navigation menu.

1) print all user's name:

-Input "2" into the text box.

Then Click Submit.

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



5.1 SQL Injection Screen

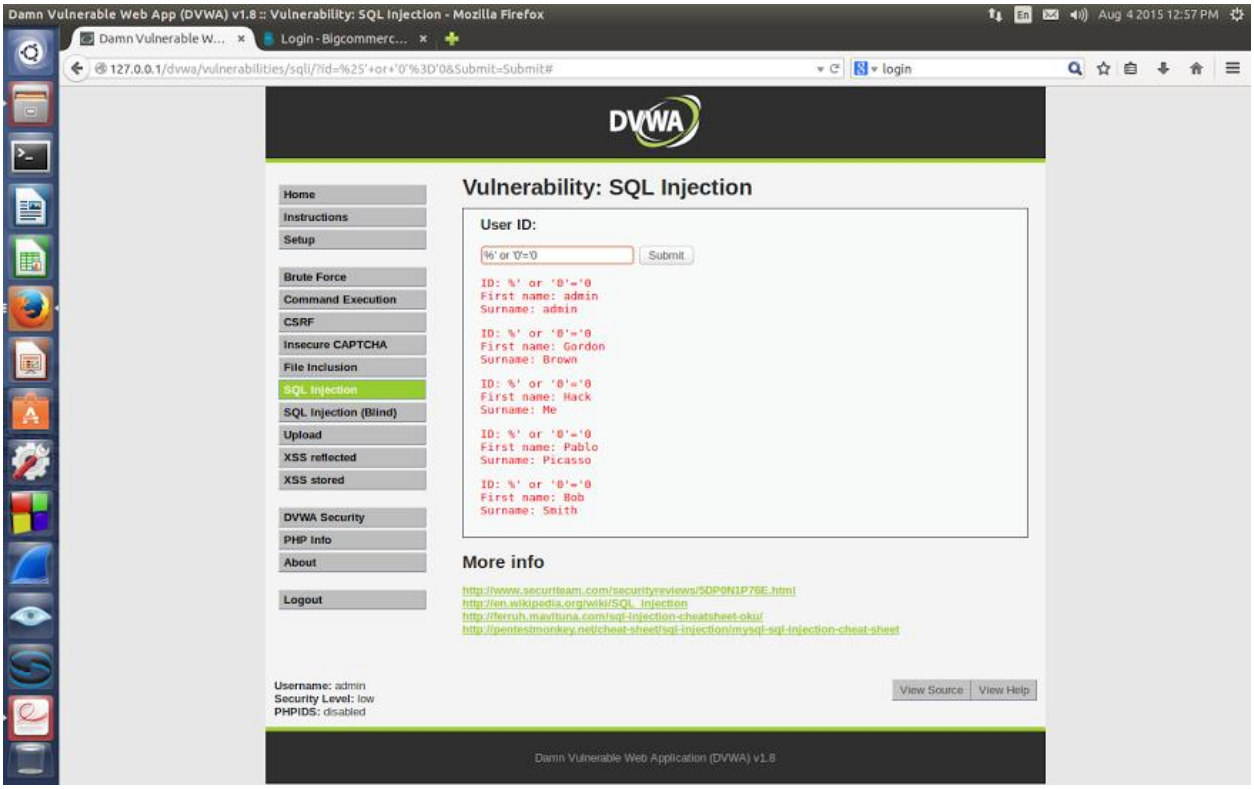
2) Input the below text into the User ID Textbox

```
%' or '0'='0
```

In this scenario, we are saying display all record that are **false** and all records that are **true**.

%' - Will probably not be equal to anything, and will be false.

'0'='0' - Is equal to true, because 0 will always equal 0.



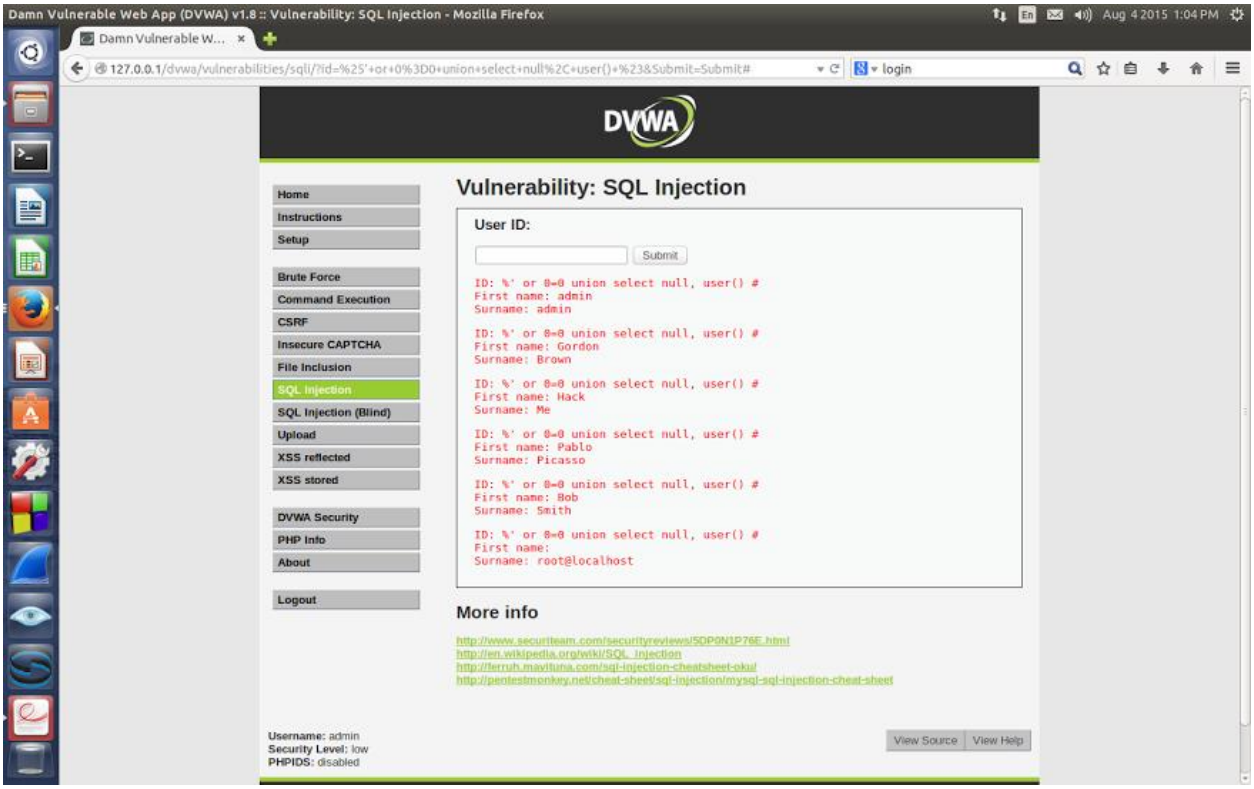
5.2 SQL Injection Screen

3) Display Database User

Input the below text into the User ID Textbox

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

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



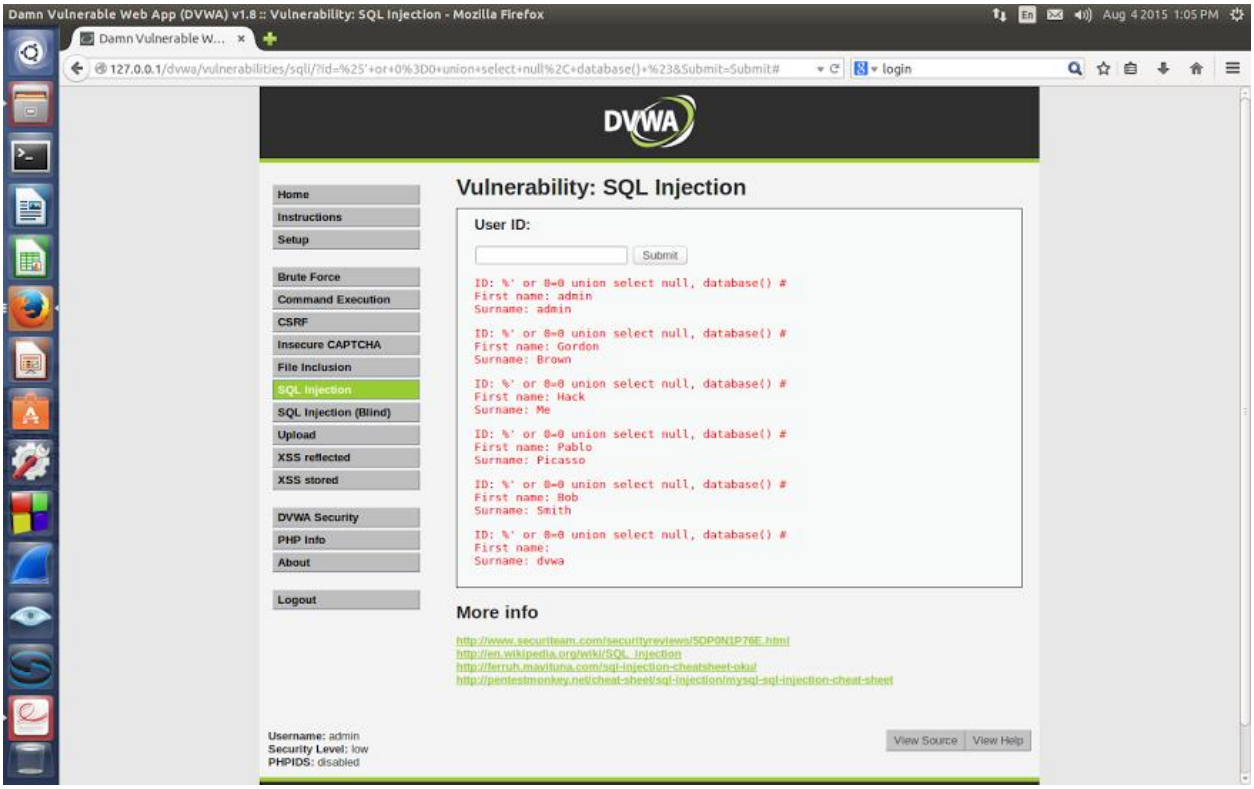
5.3 SQL Injection Screen

4) Display Database Name

Input the below text into the User ID Textbox

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

Notice in the last displayed line, dvwa is displayed in the surname.



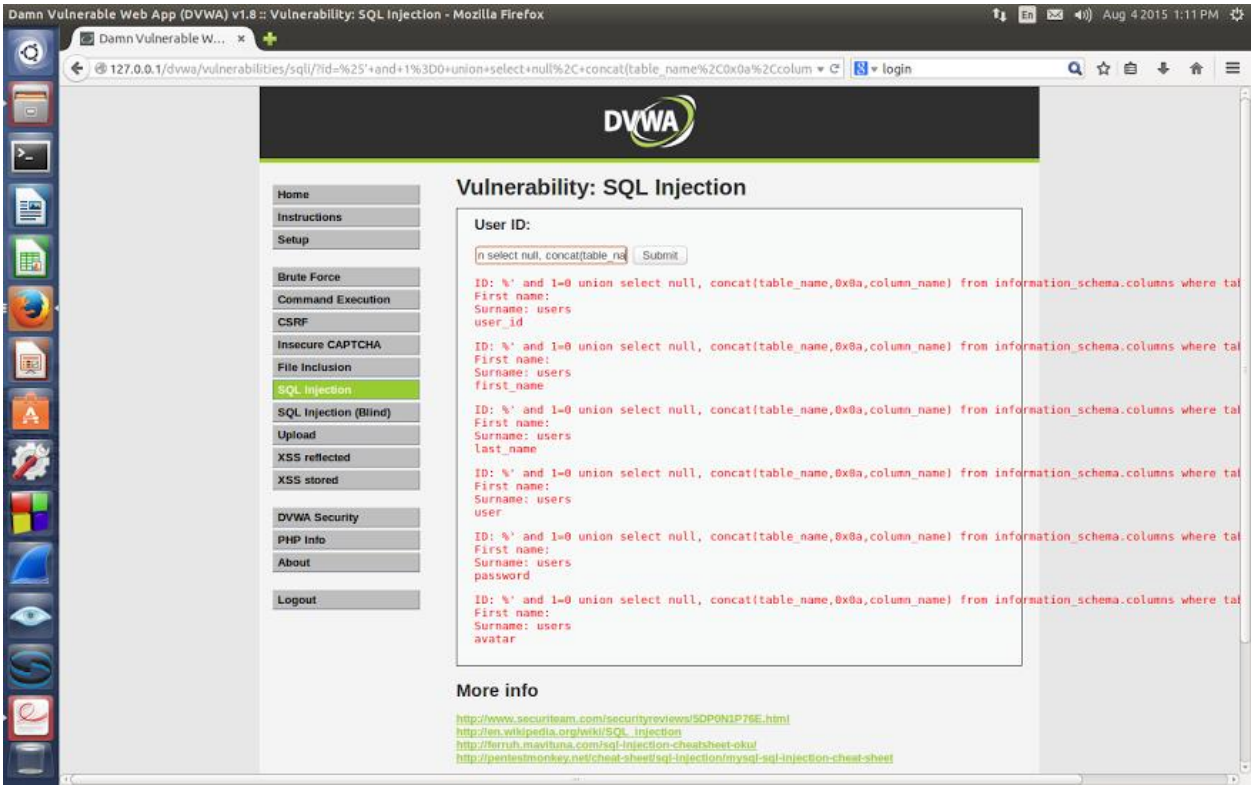
5.4 SQL Injection Screen

5) Display all the columns fields in the information_schema user table

Input the below text into the User ID Textbox

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

it will display all the columns in the **users** table.



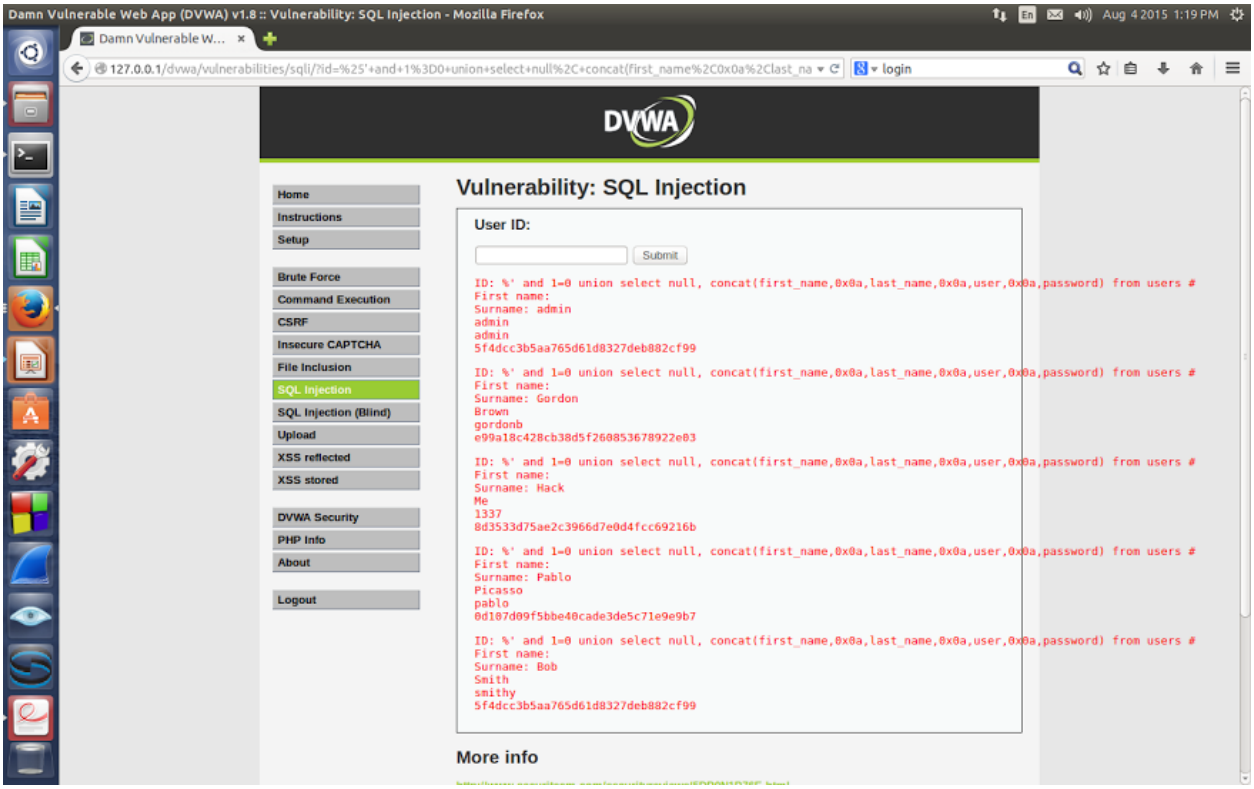
5.5 SQL Injection Screen

6) Display all the columns field contents in the information_schema user table

Input the below text into the User ID Textbox

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

Successfully displayed all the necessary authentication information into this database.



5.6 SQL Injection Screen

XSS using DVWA

What is Cross Site Scripting?

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in Web applications.

XSS enables attackers to inject client-side script into Web pages viewed by other users.

A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy.

In Addition, the attacker can send input (e.g., username, password, session ID, etc) which can be later captured by an external script.

The victim's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site.

How to perform it using DVWA!

1) Login to DVWA.

Username: admin

Password: password

2) Set Security to Low.

3) Select "XSS Stored" from the left navigation menu.

4) XSS Test Name: Test 1

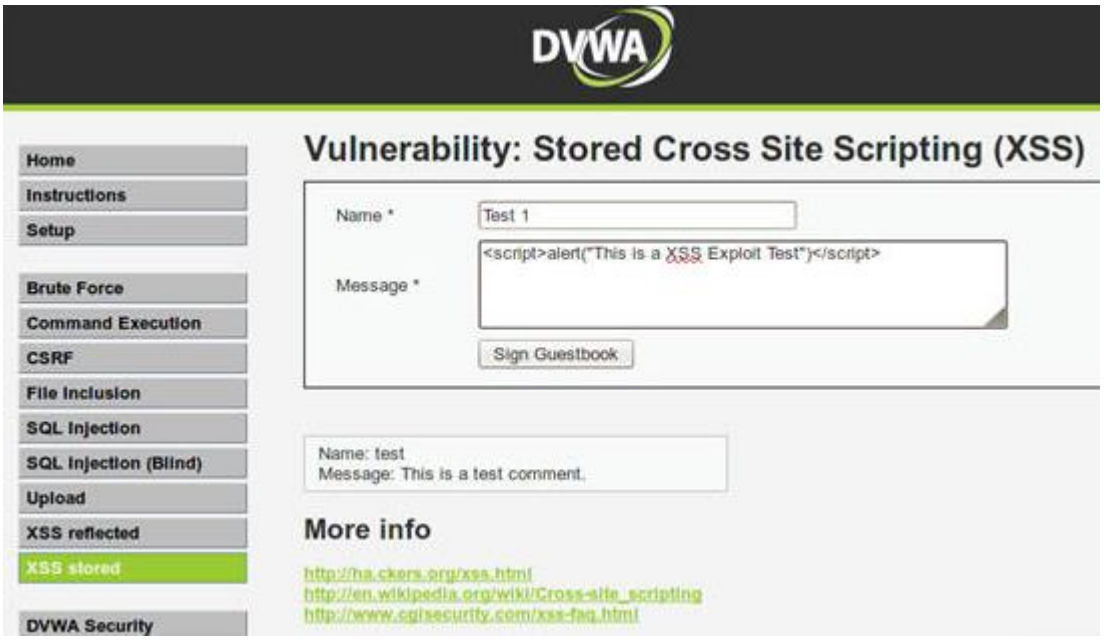
Message: <script>alert("This is a XSS Exploit Test")</script>

Click Sign Guestbook

5) XSS Test Name: Test 2

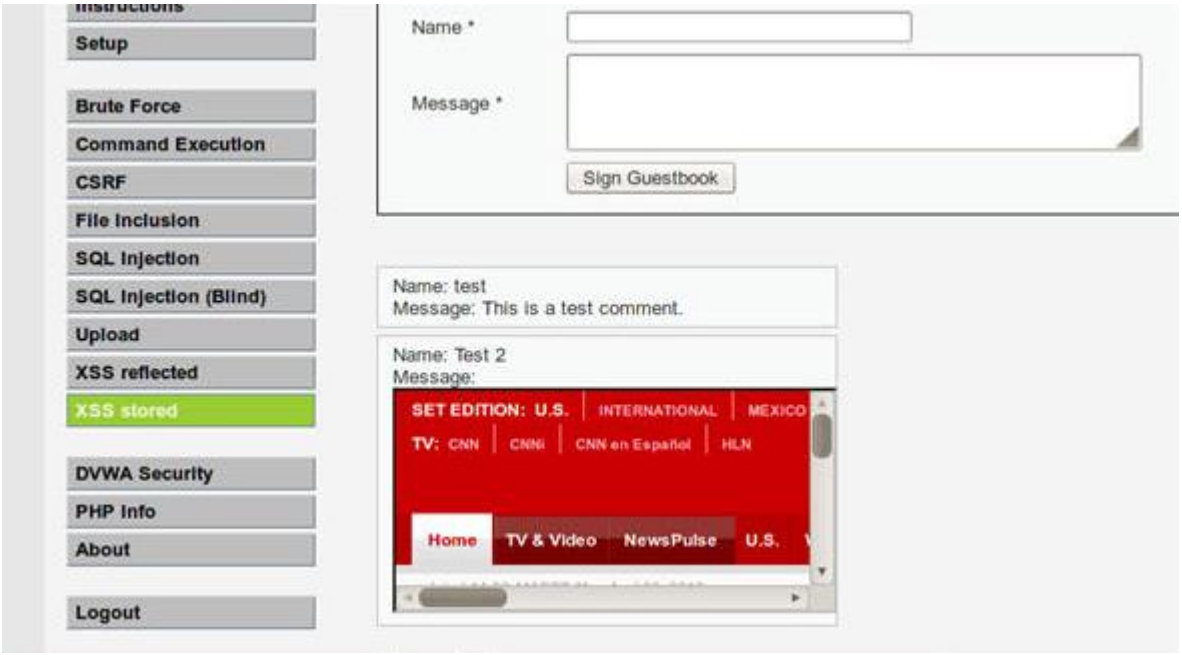
Message: <iframe src="http://www.cnn.com"></iframe>

***<u>Screenshots:</u>**



6.1 Screen 1 XSS Stored

6.2 Script one Executed Test 1



6.3 Test 2 Executed