CYBERWISER.eu

Cyber Range & Capacity Building in Cybersecurity

SQL INJECTION

Dario Varano

May 2020

SQL Injection

- Introduction
- SQL recap
- Understanding common exploit techniques
- Hands-on session

What is SQL injection?
What does it affect?
How does it work?

SQL Injection

- Introduction
- SQL recap
- Understanding common exploit techniques
- Hands-on session

What is SQL?
SQL language insight

SQL Injection

- Introduction
- SQL recap
- Understanding common exploit techniques
- Hands-on session

Part I - How to bypass authentication?
Part II – Retrieving data using UNION statement

SQL Injection

- Introduction
- SQL recap
- Understanding common exploit techniques
- Hands-on session

Retrieve username and password of subscribed users of a website

INTRODUCTION

Some text here

What is SQL Injection?

It’s the vulnerability that shows up every time you give an attacker the chance to influence the SQL queries that an application executes against a database server.

CYBERWISER.eu
Cyber Range & Capacity Building in Cybersecurity

What does it affect?

Any code that accepts input from an untrusted external source and use it to make dynamic SQL statements could be vulnerable.

How does it work?

The SQL code is injected into application input parameters that are passed to a database server in order to be parsed and executed.

SQL RECAP

A recap section for the Structured Query Language

What is SQL?

A standard language aimed at querying database systems (e.g. MySQL, SQL Server, etc.)

SQL language insight (1)

- `CREATE DATABASE db_name;`
  - Create a new SQL database
- `CREATE TABLE tab_name (column1 datatype, column2 datatype, …);`
  - Create a new SQL table in a database
- `INSERT INTO tab_name (column1, column2) VALUES (value1, value2);`
  - Insert new records in an existing table
- `SELECT column1, column2 FROM tab_name;`
  - Fetch data from an existing SQL table in a database
- `SELECT column1, column2 FROM tab_name WHERE condition;`
  - The WHERE statement is used to filter results

SQL language insight (2)

- `SELECT column1, column2 FROM tab_name WHERE condition1 AND|OR|NOT condition2;`
  - The WHERE statement can be combined with AND, OR and NOT operators to filter results using more conditions
- `UPDATE tab_name SET column1=value1 WHERE condition;`
  - Modify an existing record in a table
- `DELETE FROM tab_name WHERE condition;`
  - Delete existing records in an existing table
- `DROP DATABASE db_name;`
  - Drop an existing SQL database
- `DROP TABLE tab_name;`
  - Drop an existing SQL table in a database

SQL language insight (3)

- SELECT col1_name FROM tab1_name UNION SELECT col1_name FROM tab2_name
  - It combines the result set of two or more SELECT clauses, returning a table with distinct values
- It is possible to select all the records in a table, using *
  - `SELECT * FROM tab_name`
- The standard way for separating SQL statements to be executed is ;
  - `SELECT column1 FROM tab_name; SELECT column2 FROM tab_name`
- SQL keywords are NOT case sensitive
  - `SELECT is equal to select`
- Single line comments starts with -- with a space afterwards
  - `SELECT * FROM Customers -- WHERE City='Pisa';` is equivalent to `SELECT * FROM Customers`

MySQL: INFORMATION_SCHEMA

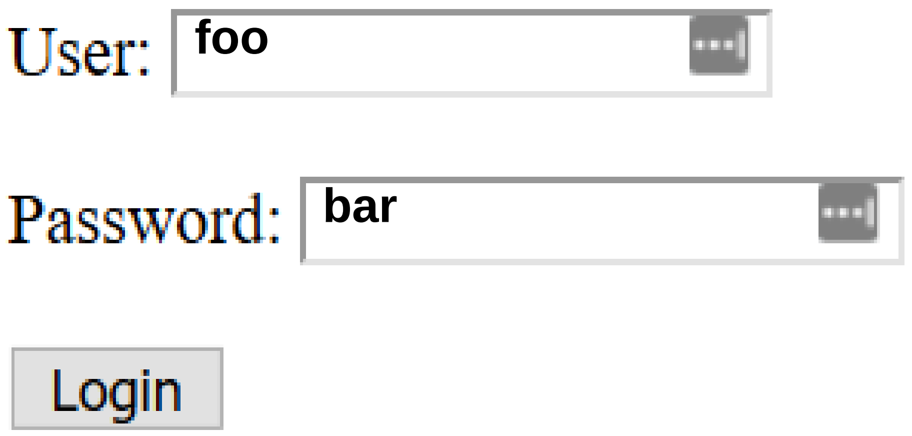
A **database** storing information about all the other databases that the MySQL server maintains. It contains several **read-only tables**:

- TABLES: provides information about tables in databases. Important columns:
  - TABLE_SCHEMA: name of the db to which the table belongs;
  - TABLE_NAME: name of the table;
- COLUMNS: provides information about columns in tables. Important columns:
  - TABLE_SCHEMA: name of the corresponding db;
  - TABLE_NAME: name of the corresponding table;
  - COLUMN_NAME: name of the column;

UNDERSTANDING COMMON EXPLOIT TECHNIQUES

Part I

How to bypass authentication?

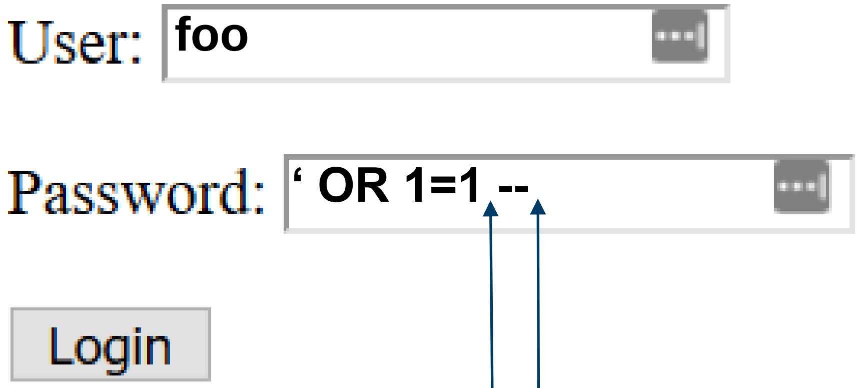


The query that is going to be executed is:

```
SELECT *
FROM users
WHERE user=‘foo’
AND password=‘bar’
```

Hint: What if you change the meaning of the SQL query to always return **true**?

Injection



User: foo

Password: ‘ OR 1=1 --

A space **is required** after --

A space **is not required** before --

The query that is going to be executed is:

```
SELECT *
FROM users
WHERE user=‘foo’
AND password=‘’ OR 1=1 --
```

UNDERSTANDING COMMON EXPLOIT TECHNIQUES

Part II

Retrieving data using UNION statements (1)

Normal usage: combining result-set of two or more SELECT into a single result set:

```
SELECT column1 FROM tab1_name
UNION
SELECT column1 FROM tab2_name
```

The above query returns a table including **distinct** values coming from both SELECT statements

By injecting a UNION, followed by another arbitrary query, it is possible to retrieve any table accessible to the database user

Retrieving data using UNION statements (2)

The result columns names will be the ones of the first SELECT statements

Conditions for using the UNION statement properly:

1. Each SELECT statement **MUST** have the **<u>same</u>** number of columns
2. The columns **MUST** have **<u>similar</u>** data types

HANDS-ON SESSION

A dive into the cyber range platform

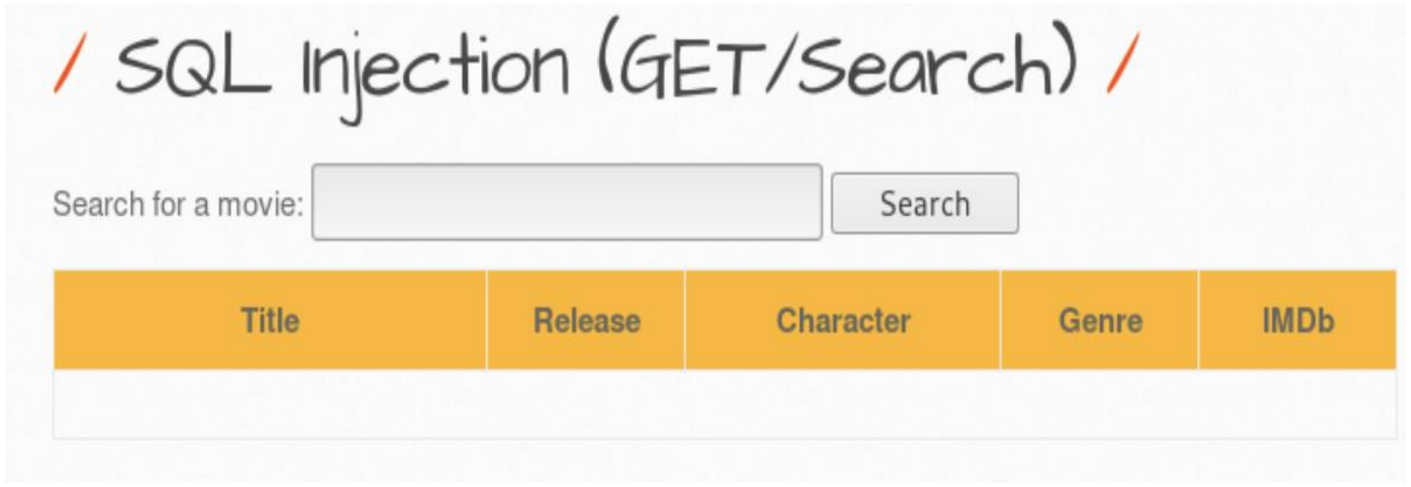
CYBERWISER.eu
Cyber Range & Capacity Building in Cybersecurity

Instructions (1)

- Login into the cyber range
- A VM workstation will be at your disposal:
  - Click on to open the VNC console
  - Click on to open the VNC console in a new browser tab
  - Login into the workstation and launch the web browser
  - Visit the following URL: http://ip_address/bWAPP
- Login into bWAPP using the following credentials:
  - Username: **bee**
  - Password: **bug**

Instructions (2)

- Choose “**SQL Injection GET/Search**” from the selection button, then click “**Hack**”
- You can now search for a movie:



Title	Release	Character	Genre	IMDb

- Objective: retrieve **username** and **password** of all registered users

CYBERWISER.eu
Cyber Range & Capacity Building in Cybersecurity

CYBERWISER.eu

Cyber Range & Capacity Building in Cybersecurity