SQL INJECTION

Submitted by Susani Antony

ID:2217152

TO PROFESSOR GRACE

CONTENTS

- INTRODUCTION
- EXAMPLES OF SQL INJECTION
- IMPACTS OF SQL INJECTION
- HOW TO DETECT SQL INJECTION
- CODE
- DEMONSTRATION
- CONCLUSION

INTRODUCTION

What is SQL injection(SQLi)

Is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.

This Enable an attacker to view data that they are normally able to retrieve.

Hence an attacker can modify or delete this data causing persistent changes to the application's content or behaviour.

In some situations an attacker can escalate a SQL injection attack to compromise the underlying server or other back-end infrastructure and it can enable them to perform denial of service attacks.

EXAMPLES OF SQL INJECTION

- Retrieving hidden data, where you can modify a SQL query to return additional results.
- Subverting application logic, where you can change a query to interfere with the application's logic
- UNION attacks, Where you can retrieve data from different database tables.
- Blind SQL injection, where you can retrieve data from different database

IMPACTS OF SQL INJECTION ATTACKS

Unauthorized access to sensitive data such as

- Passwords
- Credit card details
- Personal user information

HOW TO DETECT SQL INJECTION

You can detect SQL injection manually using systematic set of test against every entry point of the application

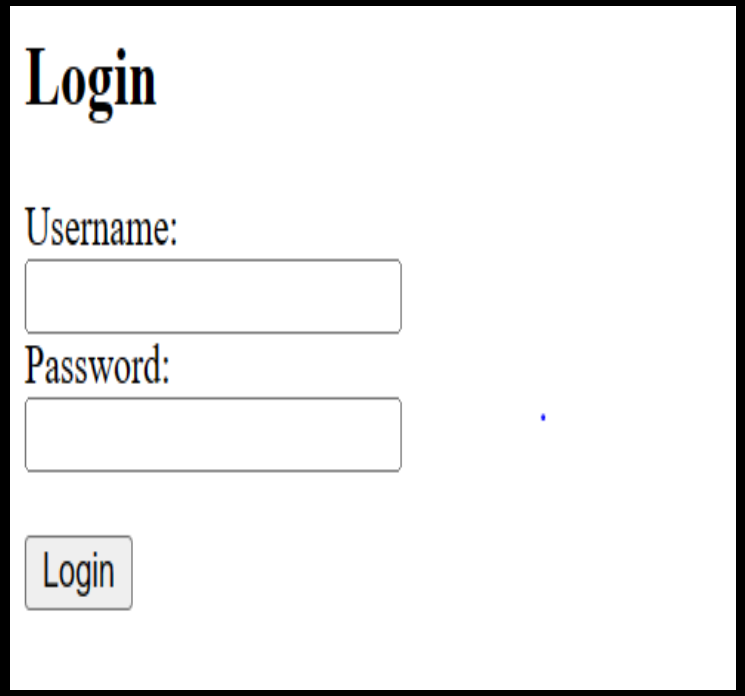
- The single quote character ' and look for errors or other anomalities
- Boolean conditions such as OR 1=1 and OR 1=2 and look for systematic differences in the application's responses
- Also you can use a scanner like Burp Scanner.

CODE

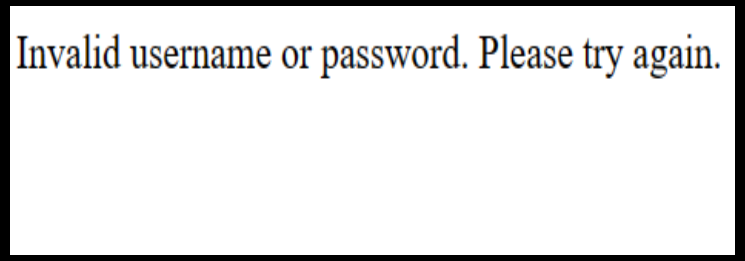
```python
from flask import Flask, render_template, request
import sqlite3

app = Flask(__name__)

conn = sqlite3.connect('vulnerable.db')
cursor = conn.cursor()


cursor.execute('''CREATE TABLE IF NOT EXISTS users
                  (id INTEGER PRIMARY KEY, username TEXT, password TEXT)''')

cursor.execute("INSERT INTO users (username, password) VALUES ('user1', 'password1')")
cursor.execute("INSERT INTO users (username, password) VALUES ('user2', 'password2')")

conn.commit()

@app.route('/')
def index():
    return render_template('login.html')

@app.route('/login', methods=['POST'])
def login():
    username = request.form['username']
    password = request.form['password']


    conn = sqlite3.connect('vulnerable.db')
    cursor = conn.cursor()

    # Vulnerable to SQL injection
    query = "SELECT * FROM users WHERE username='%s' AND password='%s'"%(username,password)

    cursor.execute(query)
    user = cursor.fetchone()

    conn.close()

    if user:
        return f"Welcome, {username}!"
    else:
        return "Invalid username or password. Please try again."

if __name__ == '__main__':
    app.run(debug=True)

```

DEMONSTRATION

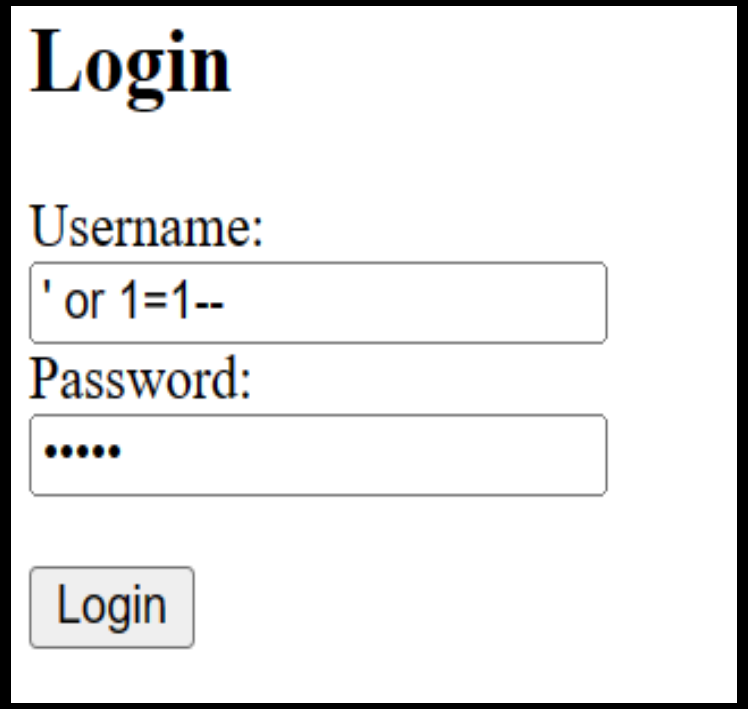
homepage



Sign in with wrong details



Sql injection



Welcome, ' or 1=1--!

THANK YOU