



Universidad
del Caribe

2000

CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO



CRECIENDO CON EL SIGLO

Ingeniería en Datos e Inteligencia Organizacional

Tópicos selectos en ingeniería de datos

Código vulnerable

Profesor:

Ismael Jimenez Sanchez

Presenta:

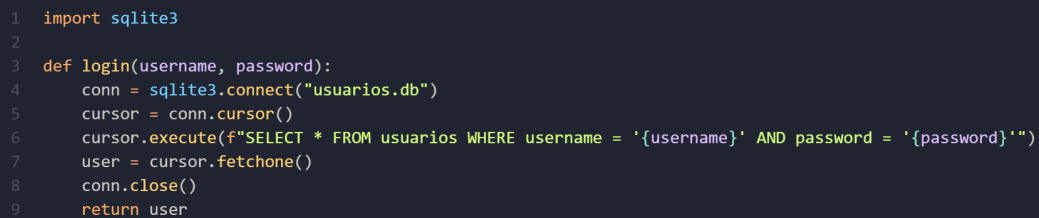
200300608 LUIS ANGEL NOH SANTIAGO

Inyección de SQL en Python

La Inyección de SQL es una vulnerabilidad de seguridad que ocurre cuando un programa o aplicación web permite que un atacante ingrese datos maliciosos en una consulta SQL. Esto puede llevar a que un atacante acceda, modifique o elimine datos en una base de datos de manera no autorizada.

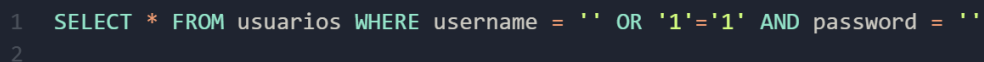
Posible causa

Supongamos que el sitio tiene un formulario de inicio de sesión y el código para verificar las credenciales del usuario es el siguiente

A screenshot of a code editor with a dark background and light-colored text. The code is a Python function named 'login' that takes 'username' and 'password' as arguments. It uses the 'sqlite3' module to connect to a database named 'usuarios.db'. It then executes a SQL query to select a user from the 'usuarios' table where the username and password match the provided inputs. The function returns the user object if found, otherwise it returns an empty string.

```
1 import sqlite3
2
3 def login(username, password):
4     conn = sqlite3.connect("usuarios.db")
5     cursor = conn.cursor()
6     cursor.execute(f"SELECT * FROM usuarios WHERE username = '{username}' AND password = '{password}'")
7     user = cursor.fetchone()
8     conn.close()
9     return user
```

En este código, el usuario proporciona su nombre de usuario y contraseña en el formulario de inicio de sesión. Sin embargo, si un atacante ingresa una contraseña como ' OR '1'='1', la consulta SQL resultante sería:


A screenshot of a code editor showing a SQL query. The query is a SELECT statement that selects all columns from the 'usuarios' table where the username is an empty string and the password is ' OR '1'='1'. This is a classic SQL injection attack designed to bypass password authentication.

```
1 SELECT * FROM usuarios WHERE username = '' OR '1'='1' AND password = ''
2
```

Esto siempre evaluará a verdadero, lo que permitirá que el atacante inicie sesión sin conocer la contraseña.

Corregir la vulnerabilidad

Para evitar la inyección de SQL en Python y en otros lenguajes, se debe utilizar la preparación de consultas o parámetros vinculados. Esto implica el uso de placeholders en las consultas SQL en lugar de insertar directamente datos proporcionados por el usuario.



```
1 import sqlite3
2
3 def login(username, password):
4     conn = sqlite3.connect("usuarios.db")
5     cursor = conn.cursor()
6     cursor.execute("SELECT * FROM usuarios WHERE username = ? AND password = ?", (username, password))
7     user = cursor.fetchone()
8     conn.close()
9     return user
```