RESPOSTAS MISSÃO 9

DESAFIO 1

1) Explique por que este código é vulnerável a SQL Injection.

Quando o código é executado, a injeção SQL (' OR '1'='1) é utilizada para obter acesso ao banco de dados sem uma senha válida. Com isso, colocando qualquer senha iria ter acesso ao usuário.

2)

```
import sqlite3
# Conectando ao banco de dados em memória
connection = sqlite3.connect(':memory:')
cursor = connection.cursor()

# Criando uma tabela e inserindo dados
cursor.execute('''CREATE TABLE users (id INTEGER PRIMARY KEY, username TEXT, password TEXT)''')
cursor.execute("INSERT INTO users (username, password) VALUES ('admin', 'admin123')")
cursor.execute("INSERT INTO users (username, password) VALUES ('user', 'user123')")
connection.commit()

# Função de login insegura
def login(username, password):
    query = "SELECT * FROM users WHERE username = ? AND password = ?"
    cursor.execute(query, (username, password))
    return cursor.fetchone()

# Testando o login com SQL Injection
user = login("admin", "admin123")
print("Usuário encontrado:", user)

connection.close()
```

```
Usuário encontrado: (1, 'admin', 'admin123')
```

DESAFIO 2

1) Substituímos uma entrada de teste que comprometia a clareza da lógica de busca por uma entrada válida, mantendo a segurança das consultas padronizadas, de forma que a função opere conforme esperado.

2)

```
import sqlite3
# Conectando ao banco de dados em memória
connection = sqlite3.connect(':memory:')
cursor = connection.cursor()

# Criando uma tabela e inserindo dados
cursor.execute('''CREATE TABLE products (id INTEGER PRIMARY KEY, name TEXT, price REAL)''')
cursor.execute("INSERT INTO products (name, price) VALUES ('Notebook', 2000.0)")
cursor.execute("INSERT INTO products (name, price) VALUES ('Smartphone', 1500.0)")
connection.commit()

# Função de busca de produtos insegura
def search_product(product_name):
    query = "SELECT * FROM products WHERE name LIKE ?"
    cursor.execute(query, ('%' + product_name + '%',))
    return cursor.fetchall()

# Testando a busca com SQL Injection
products = search_product("Notebook")
print("Produtos encontrados:", products)
connection.close()
```

```
Produtos encontrados: [(1, 'Notebook', 2000.0)]
```