AI Security Code Reviewer

Generated: 2025-06-15 19:28:23

Technical Security Analysis

Security Score: 50/100

Detailed Findings:

- SQL Injection (Critical)

Line: N/A

Description: User input directly concatenated into SQL query

Explanation: This allows attackers to inject malicious SQL commands

Fix: Use parameterized queries with placeholders

CWE: N/A

- OS Command Injection (Medium)

Line: N/A

Description: User input used in os.system() call

Explanation: This allows attackers to execute arbitrary commands

Fix: Use subprocess.run() with shell=False and proper argument lists

CWE: N/A

Code Analysis:

```
import sqlite3

import os


def get_user_data(user_id):
```

```
conn = sqlite3.connect('users.db')

query = f"SELECT * FROM users WHERE id = '{user_id}'"

result = conn.execute(query).fetchone()

os.system(f"echo 'User: {user_id}'")

return result
```