Module 5- Computer Systems (2022-23)

Project

UNIVERSITY OF TWENTE.

Testing-Security by Design Checklist

Team ID: 29	**Team Members:** Mayank Thakur, Chris Bosman, Louisa Hafferl, Meenakshi Girish Nair, Vithursika Vinasiththamby, Gyum Cho
Project Name: RaspberryPlant	**Mentor(s):** Priya Naguine and Radu Basarabá

Instructions:

1. Refer to the below table. All the mentioned points are mandatory to perform for your application except point no. 4.
2. You should consider at least 2 vulnerabilities for each criteria given in Column 'B', except point no. 4, 6, and 7.
3. The mitigation plan/solution should be considered for every identified vulnerability.
4. Make sure to review the document with your team members and mentor(s) before final submission.
5. This checklist should be in lined and submitted along with the Software Testing document.

Points	Source Code Review, Static and Dynamic Application Testing	Identified Vulnerabilities for testing (Name them)	Put tick ✓ (if you have completed all the points as mentioned in Column 1.	Remarks, if any
1	Application security vulnerabilities (e.g. Access Control, Injection, Authentication, Cross Site scripting, etc.)	Injection – we had textboxes where users could input texts. This was a vulnerability which has been patched by using prepared statements. Authentication – Users are asked to login. Passwords are salted and stored on the database.	✓	
2	Weak security in functions (e.g. old encryption techniques, Hashing, Privileges assigned, Function error, etc.)	We use the Sha512 hashing, which is slightly outdated, but we use salts to improve this. No differences in Privileges are assigned because there are no admins, only users	✓	
3	Duplicate/unnecessary functions	There are not a lot of functions and none of them are duplicates or unnecessary	✓	
4	Analyzing Program (e.g. computation time, power consumption, etc.) **(Optional)**	-		

5	Address the remaining vulnerabilities of your application (manual)	People can read database credentials from the Raspberry Pi and manually write things to the database. Requests on the web-app can be intercepted by reading packets.	✓	
6	Make a mitigation plan/solution by listing down the vulnerabilities	Packaging is used on the Raspberry Pi side to stop users from being able to read database credentials. HTTPS encryption will be used to stop people from intercepting requests on the webapp.	✓	
7	Review with your team members and approve by your mentor(s).	-	✓	

Team members reviewed: Mayank, Vithursika, Meenakshi, Chris, Gyum, Louisa (all yes)

Mentor(s) reviewed and verified: (Mentor 1, Yes), (Mentor 2, Yes), …

Prepared by:

Dipti K. Sarmah (Project Coordinator)