CM2010 Software Design and Development: Mid-term Coursework 1 submission [001]

Part 3 by Maryam Zaman

Secure Programming Recommendation 1: The first **Problem** with the web application is of passwords being stored in the database as plain text. This is a huge security risk as it means that anyone who gains access to the database can read the passwords. The **Change** that should be implemented to improve security is password hashing. Instead of storing the plain text passwords, the application should store the hash of the passwords. When a user logs in, the application should compare the hash of the entered password with the stored hash.

Secure Programming Recommendation 2: The second **Problem** is that the client sends everything as text without using any encryption algorithm to the server using the GET method. This is insecure as it means that the data is sent in clear text over the network, making it vulnerable to interception attack. The **Change** that should be made to improve this is to use HTTPS (HTTP Secure) instead of HTTP. HTTPS encrypts the data sent between the client and server, protecting it from eavesdroppers. It is recommended to consistently utilize HTTPS rather than switching between HTTP to HTTPS.

Secure Programming Recommendation 3: The third **Problem** is that the application uses the GET method to send sensitive data.

In the GET method, data is appended to the URL which is not secure. The **Change** that should be made to improve this is to use the POST method instead of GET for sending sensitive data.

Secure Programming Recommendation 4: The fourth **Problem** is that the application does not have any mechanism to protect against SQL Injection attacks. SQL Injection is a common web application vulnerability where an attacker can manipulate or control SQL queries by sending specially crafted input. The **Change** that should be made to improve security and prevent SQL Injection attacks is to use strongly typed parameterized queries or prepared statements. They ensure that the parameters (values) passed into SQL queries are treated safely, thus preventing malicious input from being executed as part of the SQL commands.

Secure Programming Recommendation 5: The final **Problem** is that the application does not have any mechanism to protect against Cross-Site Scripting (XSS) attacks. XSS is a type of attack where an attacker can inject malicious scripts into web pages viewed by other users. The **Change** that should be made to improve security and prevent XSS attacks is to implement and utilize output encoding (also known as output escaping). Output encoding converts untrusted input into a safe form where the input is displayed as data to the user without being executed as code in the browser.

Word limit: 430 (Excluding the resource & the title at top i.e. module title and my name)

Resources:

OWASP’s secure coding guidelines - Secure Coding Practices - Quick Reference Guide (owasp.org)

and

OWASP Secure Coding Practices - Quick Reference Guide | Secure Coding Practices | OWASP Foundation