RMIT UNIVERSITY

IT Infrastructure and Security (COSC2737)

Industrial focused project

Option 1

Design and development of a web-based library management systems for community library, include:

1. Design and implement a user-friendly interface (UI) for users to secure login (register and log in) (8 marks)
    a. Design and develop a simple user sign up and secure login interfaces (UIs), to collect the username and password and store into the database (details in 2a).
    b. Design and implement password reset functionality, in case user may want to reset the password if s/he is forgotten.
2. Implement a role-based access to the back-end database. (12 marks)
    a. Design and implement secure password storage mechanism. Keep the input of username and hashed password (collected in 1a) into the backend database of your choice, e.g, SQLite.
    b. Assume that the books (digital copy of books) are already in the system, to implement.
        - Registered users can search, access and download books anytime.
        - Design and implement secure communication between client and server when user searching, accessing and downloading books.
3. Technical documentation includes: (16 marks)
    a. Draw a diagram of the overall architecture design of the security features (i.e., secure login, secure communication, and secure search/access/download) in the Library management system.
    b. Explain through a diagram (provided in 3. a) how users can securely communicate (including secure search/access/download) through your secure client-server implementation.
    c. Explain systematically (i.e., step by step) how the Diffie-Hellman protocol produces the shared key between a client and a server.
    d. Explain through an example, how a digital signature is used to guarantee the integrity during secure communication between client and the server.

Option 2

Design of a web-based Secure Chat Application, include:

1. Create a QR code. (6 marks)
    a. Embed the URL http://titan.csit.rmit.edu.au/~e73581/itis/index.html to a QR code generator https://www.qrcode-monkey.com/ to create a QR code.
    b. Design a web-based Secure Chat Application (provided 2), and link it to the QR code.

       You are encouraged to implement the URL and add an item of "student ID", which can trade 1c.
2. Design a user-friendly interface (UI) for users to secure login (register and log in). (8 marks)
    a. Design intuitive user interfaces (UI) for user registration, login, profile management, and chat messaging window.
    b. Design security best practices (i.e., secure login) for user authentication.
    c. Design secure password storage mechanism.
    d. Design password recovery functionality, in case user may want to recover the password.
3. Authentication and Encryption (8 marks)
    a. Design secure authentication mechanisms, such as multi-factor authentication (MFA)
    b. Design end-to-end encryption using strong cryptographic algorithms to protect message content between two chat entities.
4. Technical documentation includes: (14 marks)
    a. Draw a diagram of the overall architecture design of the Secure Chat Application (i.e., secure login, secure communication, and secure password storage) in the Library management system.
    b. Explain through a diagram (provided in 3. a) how users can adhere to strong passwords through your implementated Secure Chat Application.
    c. Explain through a diagram (provided in 3. a) how users can store strong passwords through your implementated Secure Chat Application.
    d. Explain through a diagram (provided in 3. a) how users can securely communicate (chat messages) through your implementated Secure Chat Application.
    e. Explain systematically (i.e., step by step) how the Diffie-Hellman protocol produces the shared key between a client and a server.

f. Explain how novel and secure your developed Chat Application is and who (e.g., industry or businesses) will benefit from having your secure chat application.

g. Explain how resilient your developed secure chat application is against common security threats, such as SQL injection and cross-site scripting (XSS).