Introduction
We have made a forum-like application for this assignment which have been implemented in a secure way.

Misuse Cases

1. **Login (code injection)**
    - Misuser tries to log in to the application with a existing username and password. The username contains some code injection (sql or xss).
    - The system checks the whitelist if the username contains allowed characters, if not an error message is shown.
    - The system then allows the user to try again.
2. **Login (Brute force)**
    - A misuser tries to log in to the system by using a brute force software together with a registered user's username.
    - The misuser will have a couple of tries to enter the correct password and check the recaptcha checkbox. If a couple of tries have been made the checkbox will start to ask for you to enter characters that are shown in an image or sometimes ask you to solve a task.
    - The user will have to enter the right recaptcha for a successful login.
3. **Login (Session hijacking)**
    - The misuser tries to use a stolen session cookie to gain access to the application.
    - The system validates the session cookie as not valid. The system then redirects the misuser to the login form.
4. **Registration (code injection)**
    - The misuser tries to register a user to the application with a username and password. The username or password contains code injection(sql or xss).
    - The system checks the whitelist for characters that are allowed, if a character that is not allowed is found an exception will the thrown.
    - The application then allows the user to try again with another username or password.
5. **Edit user information (code injection)**
    - A logged in misuser tries to change the password. The old or new passwords contain sql injection code or xss.
    - The system checks the passwords to a whitelist to make sure that ….
6. **Create topic (code injection)**
    - A logged in misuser tries to create a new topic. The topic name or it's content contains some sql or xss.
    - The system checks the name and content with a whitelist. If dangerous characters are found an error message will be shown.
    - The misuser will get another try to create a new topic.

7. **Edit topic (code injection)**
    - The misuser will try to edit the topic. The new topic name or content contains sql or xss code.
    - The system checks the name and content with a whitelist. If dangerous characters are found an error message will be shown.
    - The misuser will get another try to edit topic.
8. **Create comment (code injection)**
    - The misuser will try to create a comment on a topic. The comment contains sql or xss code.
    - The system checks the comment with a whitelist. If dangerous characters are found an error message will be shown.
    - The misuser will get another try to create a comment.
9. **Edit comment (code injection)**
    - The misuser will try to edit the comment. The new comment contains sql or xss code.
    - The system checks the name and content with a whitelist. If dangerous characters are found an error message will be shown.
    - The misuser will get another try to edit comment.
10. **topicId, commentId in URL (Code injection)**
    - The misuser tries to inject sql injection in the url before or after the id's.
    - The data that is inserted into the database is parameterized and the table will only accept a number, if not an exception is thrown.
    - Typing an id will either take you to the thread/comment or take you to an empty page if it do not exist.

Confidentiality
User credentials are encrypted in the database.
Integrity
Data can not be modified or altered by unauthorized users and all data is validated before it is posted or stored.
Availability
The system should be up and running.
Authentication
The system checks if the logged in user are authenticated in a correct way by checking the session id, password and so on.

Authorization
Users who are not authorized should not have access to the forum or any of its functions. The authorized users have access to create a new topic, read topics and comment on topics. The users can edit their own posts and topics and delete their own comment. Only the admin can delete topics and delete or edit everything else.

Accountability

The system will store logs of user actions. Things that will be logged is:

- Login and logout
- Exceptions
- Create new topics, edit topic, create comment, edit comment, delete topic, delete comment
- Password was changed