Student Full Name: Zachary You Yi Jie
Admin No: 231692C
Tutorial grp: IT2301

I have chosen the following security features for my assignment:

Contents Page:

Web App Feature:

Login Page

Security Feature 1

Account Lockout

This would be implemented by developing a database tracker for failed attempts, helping the admin or system detect suspicion over accounts on grounds of a possible brute force attack. For example, by adding a failed attempt column into the user database table and keeping the tracker after every failed attempt, thus marking each subsequent failure after a prior failure. Once this counter reaches a threshold, such as 5 failed attempts, the account will be locked out for a predefined period of time, such as 15 minutes. The account status will also be updated and set to 'locked' in the database; all attempts after the lockout within that time frame will be rejected. Also, the email notification to the user would explain that his account has been locked temporarily; this is a warning for the user in case of suspicious activity, with details on how the account can be unlocked. Moreover, once the 15 minutes have passed, if it still has 5 more failed attempts after the temporary lockout, the account will be permanently set to locked, and another email will go out to the user stating this and that it will be locked out permanently until they follow the instructions and gets in touch with support. This will help in mitigating brute force attacks, in which attackers will try various combinations of usernames and passwords in order to get unauthorized access by force. In such cases, this mechanism will significantly delay these automated brute force attacks and, in turn, reduce their effectiveness by locking such accounts out after repeated failed attempts. OWASP regards account lockout as one of the most efficient ways of defense against brute force, especially when implemented along with measures like CAPTCHA or multi-factor authentication. This security feature will make sure that valid users are protected from attackers by impeding an attacker's efforts.

Security Feature 2:

Multi-Factor Authentication

Multi-factor authentication is one of the most important security features to be added as an extra layer for protection of the user account. This would mitigate the risk of unauthorized access to the login page, even when a password is compromised, by forcing users to prove their identity in multiple ways. This is implemented by making the users input their password and another form of verification before access to their accounts is given, entailing integration of an OTP system into the login process. The system will then generate a unique OTP and send it to the email address or mobile number via SMS or an authenticator app when a user successfully enters a username and password. The application will use a secure API and generate a six-digit OTP after password verification, which then will be sent to the user's contact. Then, it will ask the user to enter the OTP on the login page within the limited time, say 3 minutes, and if the OTP entered matches the generated OTP within the valid time period, the system will then grant access to the user's account. Otherwise, the verification may fail, and the user can request a new OTP and try to log in again. This feature mitigates the security problem of password compromise that occurs in several ways, including phishing attacks, among others. Since an independent second factor is needed in addition to the password, there is a much lower possibility of an attacker accessing something he has unauthorized access to. Multi-factor authentication will go a long way in helping reduce the cases of account compromise, since it introduces another layer of security that requires access to the second authenticating factor-something an attacker hardly possesses. This solution proves effective since something the user knows, such as their password, will now be combined with something they can possess, say the OTP on their device. So, even if an attacker manages to obtain the password of a user, he will most probably not have access to the user's email account or mobile device for receiving the OTP. This multilayered security approach is supported by OWASP's recommendations for multi-factor authentication as one of the critical controls to prevent unauthorized access. On their website, it says, "MFA is by far the best defense against the majority of password-related attacks, including brute-force, credential stuffing, and password spraying.". It ensures therefore that by putting multi-factor authentication in place, the application enhances security; hence this will help in making user accounts protected against common security attacks that might aim at other normal single-factor authentication systems.

Security Feature 3:

Session Timeout

Session timeout is one of the most important security features for limiting associated risk from session hijacking and unauthorized access due to devices. This feature will automatically log a user out from the application if it does not show any activity beyond a pre-defined period. The approach involves setting a session timeout limit, say 15 minutes, through the application backend's session management system. When the user logs in, a session token will be generated and copied on both the client and server sides; additionally, a timestamp of the last activity will be recorded. On every user interaction-such as clicking a button or navigating a page-the server will update the activity timestamp. If no activity is detected for the preset timeout period, the session automatically terminates and the user is taken to the login page. Then, the session token will be invalidated on the server so that it could not be reused, and when the user reaccesses the application, he has to provide his login credentials all over again for authentication purposes, proving that he is the rightful user. This feature reduces the security issue of session hijacking, whereby attackers take advantage of active sessions to impersonate legitimate users and reach their accounts. By putting in place session timeout, the time attackers can hijack a session will be minimized. The session timeout also covers a user in case he/she forgets to log out on the device if their inactivity exceeds the period stipulated, thereby minimizing the chances of session hijacking. According to OWASP, a session timeout ensures prevention against session-based attacks as most web applications contain sensitive information that a hacker may access. This feature is so effective because it limits the exposure of active sessions; in other words, should a session token be intercepted or left active on an unattended device, the attacker will have a minimal time window to exploit it. This strengthens the security of the application and thereby helps in protecting user accounts from unauthorized access while allowing the same users to re-authenticate as and when necessary. This would reduce all types of risks regarding session hijacking and unauthorized devices.

Web App Feature 2:

Registration Page

Security Feature 1:

Email Verification

Verification via email is one of the important features that are normally used in order to validate that only valid users with appropriate email addresses could complete the registration process. It features token generation during the time of user registration, whereby it embeds the verification link to that unique and secure token and sends it to the provided email address by the user. When any user registers, an application generates a token, stores it in the database with the timestamp for its expiration date by hashing. The generated link is then emailed to the user, who has to click it within a given time, say 24 hours. In case of clicking the link, the system then verifies the token against the one in the database. Upon verification and a match, the system will activate the account by updating the status of the user's account in the database to "verified". If it doesn't work, then the registration isn't complete and the user should be prompted to request a new verification email. This feature helps to mitigate the issue of spam and fake account creation that may overwhelm systems with accounts and lead to abuse. According to OWASP, verification of email addresses ensures that accounts are linked to real users who have access to their emails. This security feature will work because it requires an active email account as well as user interaction to complete registration. This makes it a great deal harder for abuse of the system by either bots or users with malicious intent. It further enhances accountability on the user side by allowing the possibility of recovery methods for password resets and notifications for account activity.

Security Feature 2:

Strong Password Enforcement

Strong password enforcement is one security feature that ensures users create passwords resilient against brute force and dictionary attacks.

It's implemented by the system, where the system enforces strict password rules on the users-like minimum 8 characters in length and at least one uppercase, one lowercase, one number, and one special character. In such cases, when the user submits the password, the application will validate it against the preset criteria set through the backend logic. If the password does not meet these, the user is prompted with an error message to make a stronger password. This helps solve the problem of weak passwords, which are one of the most common causes of account compromise. The weak password is easy to crack or guess, especially with the use of automated tools such as brute force or dictionary attack scripts. Enforcing such strong password policy security features means this security feature will make users create passwords that are more secure and not easy to exploit. As put by OWASP, password complexity requirements form an important part of secure authentication systems in helping reduce the success rate for many credential-based attacks. This security feature will effectively prevent account compromise because a strong password is difficult to guess or crack within reasonable time, even by an attacker who uses advanced tools. By implementing the strong password and adding more layers such as account lockout and multi-factor authentication, it enhances security in providing layered approaches to protect the user's account.

Security Feature 3:

CAPTCHA

CAPTCHA is a form of security preventing bots from creating registrations en masse. This is achieved by providing users with tasks that are usually easy for a human to solve, but difficult or impossible to be solved by an automated system. CAPTCHA works by having the user solve the CAPTCHA challenge at the time of account registration. It would send a backend validation for the response provided by the user, using any of the CAPTCHA validation services, for instance, Google reCAPTCHA. In case the response is correct, it allows form submission to go ahead, thereby processing account registration. In case the response happens to be invalid, block the registration and ask the user to try again. This will help reduce the problem of automated attacks where bots try to either flood systems with a lot of fake accounts or exploit the registration forms for malicious means. These can further lead to resource exhaustion, spamming, or even launching further attacks using the compromised system. OWASP classifies CAPTCHA as one of the most effective mechanisms for blocking bots and making sure that form submissions originate from real users. The potential of this security feature to mitigate this issue is in its ability to make a distinction between humans and bots reliably. CAPTCHA should be integrated with the web application feature so that the application could mitigate the risk of abuse, protect the system resources, and keep the registration processes legitimate. Modern CAPTCHA systems such as invisible CAPTCHA further improve usability by minimizing friction for human users while still managing to maintain reliable and robust security against bots.

References:

Account Lockout

- OWASP

Multi-Factor Authentication (MFA)

- OWASP

Session Timeout

- OWASP

Email Verification

- OWASP

Strong Password Enforcement

- OWASP
- OWASP

CAPTCHA

- OWASP
- OWASP