App security analysis

Earn It Group 5

I. Validation of the Form:

It was applied through form validation to each user input field to make sure that only accurate and appropriate data is accepted. Input data should be checked for length, format, and data type. Both client-side and server-side validation were used to stop client-side checks from being skipped. By doing so, common attacks like SQL injection and cross-site scripting (XSS) will be defended against.

II. HTTP Session:

To maintain user authentication and state information, secure and properly managed HTTP sessions were implemented. Session tokens must be securely generated, one-of-a-kind, and random. Implement session timeout mechanisms to instantly invalidate sessions following a lapse in activity. To prevent unauthorized access or tampering, session data was encrypted.

III. Avoiding Query Parameters:

Query parameters are readily intercepted or logged because they are visible in URLs. Sensitive data are sent from secure channels instead, as HTTP headers or encrypted request bodies. If query parameters are required, they don't contain any sensitive data, validated, and sanitized to guard against injection attacks.

IV. Validation of User Input:

To avoid common security flaws like cross-site scripting (XSS) and command injection attacks, strict input validation for all user-provided data was implemented. User input was verified and cleaned up on the server side to make sure it follows the desired format and doesn't contain

malicious code or unauthorized characters. Depending on the unique input requirements, input validation techniques were used, such as whitelisting, blacklisting, and regular expressions.

Validating the input data is essential in order to ensure its accuracy and guard against potential security flaws.

The usage of the following additional validation checks was taking place:

- Email validation entails making sure the user-provided email address is formatted correctly, such as by using regular expressions to check for proper syntax.
- Verify the KVK number entered by the company is in the correct format and length in accordance with the established standards.
- Validate the BTW number entered by the company to ensure that it follows the correct format and length requirements outlined by tax regulations.
- Increase security by enforcing strict password policies and increasing password complexity.
- Think about requiring a combination of capital and lowercase letters, numbers, and special characters.
- A password strength meter can also be added to give users feedback on the security of the password they have chosen.

V. The URI Filters:

To enforce access control policies and stop unauthorized access to restricted resources, URI filters were used. Depending on user roles, the proper access levels and permissions were defined and enforced. To make sure that the URI filters meet the application's security requirements, they are reviewed and updated frequently.

VI. Hashed Passwords with Salting:

To safely store user passwords, strong password hashing algorithms like SHA256 were used. To thwart precomputed attacks and rainbow table attacks, random salt to each password before

hashing was added. Used enough iterations to sluggish the hashing operation and make it computationally challenging for attackers. Passwords and salts were securely stored to be kept out of the hands of unauthorized people and out of logs and error messages.