KF7013 – Website Development and Deployment – Final Assessment

Academic Security Report – w22037955

Introduction

Website security is a fundamental aspect that needs to be consider at the development, deploying and functioning stages of a website. This includes the protection of the website, its data and its users from data breaches, cyber-attacks, and any other type of online threats. There are many measures that can be taken to protect against these attacks and threats at the all the stages of the website. Specially, in the development stage, it is mandatory to take necessary measures using necessary tools and techniques of programming languages.

SQL Injection Attacks

Structured Query Language injection attack is a common type of cyber-attack which allows attackers to read and manipulate data in databases by injecting SQL queries into input fields of websites. With this kind of an attack to a vulnerable website, attackers can gain access to read, change, or delete databases and sometimes they may be able to breach login systems with unauthorized logins.

Most popular method for avoiding SQL injection attack is running SQL queries with prepared statements (parameterized queries). Parameterized statements ensure that the parameters such as user inputs, passed into SQL statements are processed and executed in a safe manner, using multiple safeguards (Rankothge W.H.; Randeniya M. and Samaranayaka V. (2020)). The ‘ Tourist Excursion Booking ’ website developed for this assignment is coded with this prepared/ parameterized statements at all instances where it reads or inserts data to the database. Then those prepared statements and user data inputs go through inbuilt php functions mentioned below which avoid the vulnerability of SQL injection attacks.

- mysqli_prepare()
- mysqli_query()
- mysqli_stmt_bind_param()
- mysqli_stmt_execute()
- mysqli_stmt_get_result()
- mysqli_fetch_assoc()

Cross-site scripting attacks (XSS)

Cross-site scripting attack is an attack to websites which allows attackers to inject malicious scripts to the website by inserting programming codes to website’s input fields. With this attacks, attackers may gain access to sensitive information of users, unauthorized logins and redirecting users to malware, spam or phishing links. Sometimes attackers may be able to change the content of web pages as well.

XSS threats can be avoided by validating and checking data that are provided by users to ensure consistence with the required format for web applications (Taha T.A. and Karabatak M. (2018)). In this

website, all the data inputs from HTML forms are validated and verified using several inbuilt php functions. Mainly all the data inputs go through a function called sanitizeInput() which sanitizes the data using 3 php inbuilt functions as below.

- trim() - function removes whitespace and other predefined characters from both sides of a string. (www.w3schools.com PHP trim() Function)
- htmlspecialchars() - function converts some predefined characters to HTML entities.
    - & (ampersand) becomes &
    - " (double quote) becomes "
    - ' (single quote) becomes '
    - < (less than) becomes <
    - > (greater than) becomes > (www.w3schools.com PHP htmlspecialchars() Function)
- mysqli_real_escape_string() - unction escapes special characters in a string for use in an SQL query, taking into account the current character set of the connection. (www.w3schools.com PHP mysqli real_escape_string() Function)

Then, the “E-mail” input fields in both “signup.php” and “login.php” validate the user email inputs by checking the entered email is in valid format of an email by using the inbuilt php function mentioned below.

- filter_var($email, FILTER_VALIDATE_EMAIL)

This function restricts users inserting any statement to the website other than a valid email address. Apart from that, closing database connections is also an important because leaving database connections open can make the website more vulnerable to cyber-attacks. Therefore, all the php scripts in this website which used a database connection to read or insert data to the database closed the database connections by using inbuilt php function mentioned below.

- mysqli_close()

hashing of user passwords also implemented in this website to prevent leaking user password at a data breach event. This is important because most of real-world users of websites tend to use similar passwords to most of their online user accounts. This can lead to a severe matter going beyond from this website for users if user passwords are leaked.

Security measures to be taken in future

There are several security measures that can be implemented in future for this website.

Google reCAPTCHA

This is a service provided by google to filter real users entering data to the website forms. This can prevent brut force attacks from bots which basically a software or a script continuously submitting data to website forms.

Locating user restricted files into an upper level of public_html folder

Accessing the script files which uses for functionality but nothing to display, through a url can make the website more vulnerable to cyber-attacks. As an example, all the php scripts in KF7013/content/includes can move to an upper-level folder to the public_html folder in the server to prevent accessing them by HTTP requests.

Cloudflare Website Protection

Cloudflare is a popular cloud-based website security service that can be used to protect websites from cyber-attacks. This is solid protection against all kind of unauthorized attempts of gaining access on the target website, customer data, compromise and abusive bots. This type of protection also includes features like under attack mode, where the client should be able to answer to the java script challenge. (Tasevski I. and Jakimoski K. (2020)) Driving the traffic through the Cloudflare can prevent specially the Distributed Denial of Service (DDoS) attacks to the website. Furthermore, Cloudflare provides SSL/TSL encryptions which makes the website more secured.

2-Factor Authentication for users

it is recommended to implement a 2-Factor Authentication system for logins of the users by using a mobile application such as 'google Authenticator' or 'Duo Mobile' to prevent unauthorized logins to the system.

References

- Rankothge W.H.; Randeniya M. and Samaranayaka V. (2020) Identification and Mitigation Tool for Sql Injection Attacks (SQLIA) Available at: https://ieeexplore.ieee.org/document/9342703 (Accessed: 06.01.2023)

- Taha T.A. and Karabatak M. (2018) A proposed approach for preventing cross-site scripting. Available at: https://ieeexplore.ieee.org/document/8355356 (Accessed: 06.01.2023)

- www.w3schools.com PHP trim() Function. Available at: https://www.w3schools.com/php/func_string_trim.asp (Accessed: 06.01.2023)

- www.w3schools.com PHP htmlspecialchars() Function. Available at: https://www.w3schools.com/php/func_string_htmlspecialchars.asp (Accessed: 06.01.2023)

- www.w3schools.com PHP mysqli real_escape_string() Function. Available at: https://www.w3schools.com/php/func_mysqli_real_escape_string.asp (Accessed: 06.01.2023)

- Tasevski I. and Jakimoski K. (2020). Overview of SQL Injection Defense Mechanisms. Available at: https://ieeexplore.ieee.org/document/9306676 (Accessed: 06.01.2023)