| | |
|---|---|
| **Started on** | Thursday, 13 March 2025, 3:25 PM |
| **State** | Finished |
| **Completed on** | Thursday, 13 March 2025, 3:31 PM |
| **Time taken** | 5 mins 38 secs |
| **Marks** | 13.00/15.00 |
| **Grade** | **86.67** out of 100.00 |

**Question 1**

Complete

Mark 1.00 out of 1.00

How does the fixed version of the XSS Demo prevent XSS?

- a. By blocking all comments containing <script>
- b. By encoding all data before storing it
- c. By using DOMPurify to sanitize both input and output
- d. By only allowing administrators to post comments

**Question 2**

Complete

Mark 0.00 out of 1.00

What does the dangerouslySetInnerHTML property in React do?

- a. Encrypts JavaScript code
- b. Prevents all forms of user input
- c. Blocks XSS automatically
- d. Allows raw HTML to be inserted into the page

**Question 3**

Complete

Mark 1.00 out of 1.00

Which of the following SQL queries is vulnerable to SQL Injection?

- a. SELECT * FROM users WHERE username = ? AND password = ?;
- b. SELECT * FROM users WHERE username = '" + user_input + "' AND password = '" + pass_input + "';
- c. SELECT * FROM users WHERE username = 'admin' AND password = 'admin';
- d. PREPARE stmt FROM 'SELECT * FROM users WHERE username = ? AND password = ?';

**Question 4**

Complete

Mark 1.00 out of 1.00

How can a developer protect against XSS attacks in a MERN stack app?

- a. Sanitize user input using DOMPurify or server-side escaping
- b. Use inline JavaScript to filter malicious code
- c. Allow script execution inside innerHTML
- d. Use dangerouslySetInnerHTML without sanitization

**Question 5**

Complete

Mark 1.00 out of 1.00

Why does this React component not execute XSS payloads?

- a. React automatically escapes input to prevent script execution
- b. The comment data is stored in a secure database
- c. dangerouslySetInnerHTML is required to execute scripts
- d. The browser blocks all inline scripts by default

**Question 6**

Complete

Mark 1.00 out of 1.00

Which of the following XSS payloads is most likely to bypass filtering?

- a. <iframe src=javascript:alert('XSS')>
- b. <script>alert('XSS')</script>
- c. All of the above
- d. <img src=x onerror=alert('XSS')>

**Question 7**

Complete

Mark 1.00 out of 1.00

What additional security measures can prevent XSS attacks?

- a. Content Security Policy (CSP)
- b. All of the above
- c. Escaping special characters (<, >)
- d. Sanitizing user input before storing it

**Question 8**

Complete

Mark 1.00 out of 1.00

What is the main difference between SQL Injection and XSS?

- a. Both are prevented using the same techniques
- b. XSS can modify SQL databases
- c. SQL Injection is always more dangerous than XSS
- d. SQL Injection targets databases, while XSS targets browsers

**Question 9**

Complete

Mark 1.00 out of 1.00

What is the most common way to fix XSS vulnerabilities?

- a. Using JavaScript's eval() function
- b. Using DOMPurify to sanitize user input
- c. Allowing only admin users to enter scripts
- d. Storing scripts in the database

**Question 10**

Complete

Mark 1.00 out of 1.00

Which of the following best describes Stored XSS?

- a. A script is executed immediately when injected
- b. A script is embedded in a URL and executed when the victim clicks the link
- c. XSS that only works on outdated browsers
- d. The malicious script is stored in the database and executed when loaded by a user

**Question 11**

Complete

Mark 1.00 out of 1.00

What is the best way to protect a Node.js MySQL database from SQL Injection?

- a. Use eval() to sanitize user input
- b. Store passwords in plain text for easy authentication
- c. Validate user input with client-side JavaScript only
- d. Use prepared statements and parameterized queries

**Question 12**

Complete

Mark 1.00 out of 1.00

What is SQL Injection?

a. A method to securely access a database
b. A technique used to bypass authentication and manipulate database queries
c. A tool to optimize database performance
d. A way to encrypt SQL queries

**Question 13**

Complete

Mark 1.00 out of 1.00

Which payload can be used to bypass authentication in an SQL injection attack?

a. ' OR username='admin' AND password='admin';
b. '; SELECT * FROM passwords;
c. ' AND DROP TABLE users;
d. ' OR 1=1 --

**Question 14**

Complete

Mark 0.00 out of 1.00

Which of the following is NOT a way to prevent SQL Injection?

a. Using Prepared Statements
b. Escaping user input before using it in a query
c. Using ORMs like Sequelize or Mongoose
d. Using DOMPurify for sanitization

**Question 15**

Complete

Mark 1.00 out of 1.00

What does `--` in an SQL Injection attack do?

a. Increases query execution speed
b. Encrypts user input
c. Adds an additional condition to the query
d. Comments out the rest of the SQL query