Started on	Monday, 2 June 2025, 12:32 PM
State	Finished
Completed on	Monday, 2 June 2025, 12:38 PM
Time taken	6 mins 32 secs
Marks	9.00/12.00
Grade	**75.00** out of 100.00

Question 1

Complete

Mark 1.00 out of 1.00

How can you prevent JWT replay attacks in sensitive RBAC-based applications?

- a. Implement rotating refresh tokens
- b. Use only the frontend to validate roles
- c. Store tokens in localStorage
- d. Use longer expiration time

Question 2

Complete

Mark 0.00 out of 1.00

If a user's role is updated from "editor" to "admin", but their JWT hasn't expired yet, what is a potential risk?

- a. Token becomes invalid immediately
- b. Token size increases
- c. Role update may not reflect until re-login
- d. Signature gets mismatched

Question 3

Complete

Mark 1.00 out of 1.00

In a RBAC model, which principle is crucial for minimizing access privileges?

- a. Token obfuscation
- b. Role inheritance
- c. Time-based access
- d. Least privilege

Question 4

Complete

Mark 0.00 out of 1.00

In a secure RBAC system, where should the logic for role-based route protection ideally reside?

- a. Frontend only
- b. JWT header
- c. Middleware or backend route handlers
- d. Database triggers

Question 5

Complete

Mark 1.00 out of 1.00

What change should be made to the following JWT-based login handler to add RBAC? const token = jwt.sign({ id: user.id }, 'mysecret');

- a. Encrypt the token
- b. Use HS512 algorithm
- c. Add user email to the payload
- d. Add role: user.role to payload

Question 6

Complete

Mark 1.00 out of 1.00

What is a secure way to refresh a short-lived JWT without asking the user to log in again?

- a. Use the same JWT for 1 year
- b. Use a cookie-stored access token
- c. Store token in sessionStorage
- d. Use a secure refresh token mechanism

Question 7

Complete

Mark 0.00 out of 1.00

What is the primary purpose of the JWT signature?

- a. Validates the integrity and authenticity of the token
- b. Prevents cross-site scripting attacks
- c. Stores expiration timestamp
- d. Encrypts the token data

Question 8

Complete

Mark 1.00 out of 1.00

What is the problem with the following code if used in production? const token = jwt.sign({ userId: 1 }, '123', { expiresIn: '2h' });

- a. Token will never expire
- b. The secret is weak and predictable
- c. It uses numeric user ID
- d. Nothing, it's secure

Question 9

Complete

Mark 1.00 out of 1.00

What will happen if the secret key used to sign a JWT is leaked?

- a. Token will become unreadable
- b. JWTs will auto-expire
- c. Any user can generate valid tokens
- d. Signature verification will be stricter

Question 10

Complete

Mark 1.00 out of 1.00

Which claim in a JWT helps enforce token expiration?

- a. exp
- b. sub
- c. aud
- d. iat

Question 11

Complete

Mark 1.00 out of 1.00

Which part of a JWT is typically used to store user roles for implementing RBAC?

- a. Payload
- b. Signature
- c. Token Expiry
- d. Header

Question 12

Complete

Mark 1.00 out of 1.00

Why is storing a JWT in localStorage considered risky in web applications?

a. It expires too quickly
b. It increases backend load
c. It cannot be read by JavaScript
d. It's vulnerable to XSS attacks