Started on	Monday, 2 June 2025, 12:32 PM
State	Finished
Completed on	Monday, 2 June 2025, 12:37 PM
Time taken	5 mins 6 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 longer expiration time
- c. Use only the frontend to validate roles
- d. Store tokens in localStorage

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

Question 3

Complete

Mark 1.00 out of 1.00

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

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

Question 4

Complete

Mark 1.00 out of 1.00

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

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

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. Add role: user.role to payload
- c. Add user email to the payload
- d. Use HS512 algorithm

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. Store token in sessionStorage
- c. Use a secure refresh token mechanism
- d. Use a cookie-stored access token

Question 7

Complete

Mark 0.00 out of 1.00

What is the primary purpose of the JWT signature?

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

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

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. Signature verification will be stricter
- d. Any user can generate valid tokens

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 0.00 out of 1.00

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

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

Question 12

Complete

Mark 1.00 out of 1.00

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

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