Started on	Monday, 2 June 2025, 12:32 PM
State	Finished
Completed on	Monday, 2 June 2025, 12:37 PM
Time taken	5 mins 28 secs
Marks	10.00/12.00
Grade	**83.33** 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. Store tokens in localStorage
- d. Use only the frontend to validate roles

Question 2

Complete

Mark 1.00 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. Signature gets mismatched
- b. Role update may not reflect until re-login
- c. Token size increases
- d. Token becomes invalid immediately

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. Token obfuscation
- c. Role inheritance
- d. Time-based access

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. Database triggers
c. Middleware or backend route handlers
d. JWT header

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

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

Question 7

Complete

Mark 1.00 out of 1.00

What is the primary purpose of the JWT signature?

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

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. Nothing, it's secure
- c. It uses numeric user ID
- d. The secret is weak and predictable

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

Question 10

Complete

Mark 1.00 out of 1.00

Which claim in a JWT helps enforce token expiration?

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

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