

Started on	Monday, 2 June 2025, 4:39 PM
State	Finished
Completed on	Monday, 2 June 2025, 4:48 PM
Time taken	8 mins 52 secs
Marks	16.00/16.00
Grade	100.00 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. Use longer expiration time
- ☐ b. Use only the frontend to validate roles
- ☐ c. Store tokens in localStorage
- ☒ d. Implement rotating refresh tokens

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. Token becomes invalid immediately
- ☒ b. Role update may not reflect until re-login
- ☐ 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. Time-based access
- ☐ b. Role inheritance
- ☐ c. Token obfuscation
- ☒ d. Least privilege

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. Middleware or backend route handlers
- ☐ b. Frontend only
- ☐ c. Database triggers
- ☐ 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. Add role: `user.role` to payload
- ☐ b. Encrypt the token
- ☐ 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 a secure refresh token mechanism
- ☐ b. Use a cookie-stored access token
- ☐ c. Store token in `sessionStorage`
- ☐ d. Use the same JWT for 1 year

Question 7

Complete

Mark 1.00 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. Encrypts the token data
- ☐ 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. The secret is weak and predictable
- ☐ b. Nothing, it's secure
- ☐ c. Token will never expire
- ☐ d. It uses numeric user ID

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. Token will become unreadable
- ☐ 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. Signature
- ☐ b. Token Expiry
- ☒ c. Payload
- ☐ 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 cannot be read by JavaScript
- ☐ b. It expires too quickly
- ☒ c. It's vulnerable to XSS attacks
- ☐ d. It increases backend load

Question 13

Complete

Mark 1.00 out of 1.00

Given the following code, which statement is true?

```
const MyComponent = React.memo(({ onClick }) => {  
  console.log("Rendered");  
  return <button onClick={onClick}>Click</button>;  
});
```

What must be true for React.memo to prevent re-renders when parent re-renders?

- ☒ a. onClick must be stable across renders (e.g., memoized using useCallback)
- ☐ b. React.memo always skips rendering regardless of prop types
- ☐ c. onClick must be declared outside the parent component
- ☐ d. onClick must be memoized using useMemo

Question 14

Complete

Mark 1.00 out of 1.00

In which of the following scenarios is useMemo most beneficial?

- ☒ a. To optimize expensive computations based on stable inputs
- ☐ b. To memoize functions used as event handlers
- ☐ c. To store global constants across modules
- ☐ d. To prevent unnecessary re-renders of pure components

Question 15

Complete

Mark 1.00 out of 1.00

Consider the following component:

```
const List = React.memo(({ items }) => {  
  return items.map(item => <div key={item.id}>{item.name}</div>);  
});
```

If the parent re-renders but passes the same array reference for items, what happens?

- ☐ a. React.memo deep compares array values
- ☐ b. React.memo skips rendering only if keys are stable
- ☐ c. React.memo causes List to re-render
- ☒ d. React.memo skips rendering because the array reference is unchanged

Question 16

Complete

Mark 1.00 out of 1.00

Why might excessive use of useMemo lead to performance degradation rather than improvement?

- ☐ a. React re-renders the component regardless of useMemo
- ☐ b. useMemo causes stale closures
- ☐ c. useMemo increases memory usage permanently
- ☒ d. Creating memoized values and comparing dependencies has computational cost