ClandBAC-Quiz

Started on	Monday, 17 March 2025, 3:32 PM
State	Finished
Completed on	Monday, 17 March 2025, 3:39 PM
Time taken	6 mins 49 secs
Marks	15.00/15.00
Grade	**100.00** out of 100.00

Question 1
Complete
Mark 1.00 out of 1.00
Flag question

Given the following vulnerable code, what type of attack can be performed?
exec(`ping ${req.body.host}`, (error, stdout, stderr) => { ... });

- a. CSRF Attack
- b. SQL Injection
- c. Cross-Site Scripting (XSS)
- d. Command Injection ✓

Question 2
Complete
Mark 1.00 out of 1.00
Flag question

How can Broken Access Control be exploited?

- a. By logging in with the wrong password
- b. By modifying JWT tokens or accessing restricted APIs ✓
- c. By using a strong password
- d. By making too many API requests

Question 3
Complete
Mark 1.00 out of 1.00
Flag question

How can the following function be exploited?

```
app.post('/track-vehicle', (req, res) => {
  const { plateNumber } = req.body;
  exec(`echo Tracking vehicle ${plateNumber}`, (error, stdout, stderr) => { ... });
});
```

- a. By making multiple requests at the same time
- b. By sending an empty request body
- c. By injecting shell commands in the plateNumber field ✓
- d. By using a VPN

Question 4
Complete
Mark 1.00 out of 1.00
Flag question

If a user inputs `ABC123 && rm -rf /`, what will happen on a Linux server?

- a. The vehicle tracking system will show an error
- b. The server will shut down immediately
- c. The entire file system could be deleted ✓
- d. Nothing will happen

Question 5
Complete
Mark 1.00 out of 1.00
Flag question

What command could an attacker enter in the `/track-vehicle` endpoint to delete files on a Windows system?

- a. ABC123 && mv /etc/passwd /dev/null
- b. ABC123 && shutdown -h now
- c. ABC123; rm -rf /
- d. ABC123 && del C:\Windows\System32 ✓

Question 6
Complete
Mark 1.00 out of 1.00
Flag question

What is the best way to prevent command injection attacks?

- a. Use eval() to process user input
- b. Use parameterized queries and sanitize input ✓
- c. Allow user input directly in system commands
- d. Use an insecure API to execute shell commands

Question 7
Complete
Mark 1.00 out of 1.00
Flag question

What is the correct way to restrict access to admin users only?

- a. if (decoded.id === 1) return res.status(403).json({ error: 'Forbidden' });
- b. if (decoded.role !== 'user') return res.status(403).json({ error: 'Forbidden' });
- c. if (!decoded.role) return res.status(403).json({ error: 'Forbidden' });
- d. if (decoded.role !== 'admin') return res.status(403).json({ error: 'Forbidden' }); ✓

Question 8
Complete
Mark 1.00 out of 1.00
Flag question

What is the impact of Broken Access Control on an application?

a. Attackers can execute arbitrary commands on the server
b. It allows Cross-Site Scripting (XSS)
c. Unauthorized users can access restricted information or perform admin actions
d. The database gets automatically deleted

Question 9
Complete
Mark 1.00 out of 1.00
Flag question

What is the primary cause of command injection vulnerabilities in applications?

a. Poor network security configuration
b. Incorrect use of loops in JavaScript
c. Using HTTPS instead of HTTP
d. Lack of input validation when executing system commands

Question 10
Complete
Mark 1.00 out of 1.00
Flag question

What is the safest way to execute system commands in Node.js?

a. Using exec() with user input
b. Using eval()
c. Concatenating user input into system commands
d. Using execFile() with sanitized input

Question 11
Complete
Mark 1.00 out of 1.00
Flag question

What security flaw exists in the following `/users` endpoint?

```
app.get('/users', (req, res) => {
  const token = req.headers.authorization;
  jwt.verify(token, SECRET_KEY, (err, decoded) => {
    db.query('SELECT id, username, role FROM users', (err, results) => {
      res.json({ users: results });
    });
  });
});
```

a. It does not store passwords securely
b. It does not verify the user's role before returning data
c. It does not return JSON data
d. It is vulnerable to SQL injection

Question 12
Complete
Mark 1.00 out of 1.00
Flag question

What would happen if an attacker modified a JWT token to escalate their privileges?

a. The server would detect the modification and reject the request
b. They would get logged out
c. The token would expire immediately
d. They could access admin-only features

Question 13
Complete
Mark 1.00 out of 1.00
Flag question

Which function is the most dangerous when handling user input in Node.js?

a. console.log()
b. parseInt()
c. exec()
d. JSON.stringify()

Question 14
Complete
Mark 1.00 out of 1.00
Flag question

Which of the following is an effective way to prevent Broken Access Control?

a. Remove authentication from sensitive endpoints
b. Validate user roles and permissions before processing requests
c. Store JWT tokens in Local Storage without encryption
d. Allow users to modify their own JWT tokens

Question 15
Complete
Mark 1.00 out of 1.00
Flag question

Why is the following endpoint a security risk?

```
app.get('/users', (req, res) => {
  db.query('SELECT id, username, role FROM users', (err, results) => {
    res.json({ users: results });
  });
});
```

a. It is vulnerable to CSRF
b. It allows SQL Injection
c. It exposes all users' details without authentication
d. It uses HTTPS instead of HTTP