Started on	Monday, 17 March 2025, 3:32 PM
State	Finished
Completed on	Monday, 17 March 2025, 3:37 PM
Time taken	4 mins 50 secs
Marks	15.00/15.00
Grade	**100.00** out of 100.00

Question 1

Complete

Mark 1.00 out of 1.00

Given the following vulnerable code, what type of attack can be performed?

```
exec(`ping ${req.body.host}`, (error, stdout, stderr) => { ... });
```

○ a. SQL Injection

○ b. CSRF Attack

● c. Command Injection

○ d. Cross-Site Scripting (XSS)

Question 2

Complete

Mark 1.00 out of 1.00

How can Broken Access Control be exploited?

○ a. By logging in with the wrong password

○ b. By making too many API requests

● c. By modifying JWT tokens or accessing restricted APIs

○ d. By using a strong password

Question 3

Complete

Mark 1.00 out of 1.00

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 injecting shell commands in the plateNumber field

○ b. By making multiple requests at the same time

○ c. By sending an empty request body

○ d. By using a VPN

Question 4

Complete

Mark 1.00 out of 1.00

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

○ a. Nothing will happen

● b. The entire file system could be deleted

○ c. The vehicle tracking system will show an error

○ d. The server will shut down immediately

Question 5

Complete

Mark 1.00 out of 1.00

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

○ a. ABC123 && shutdown -h now

● b. ABC123 && del C:\Windows\System32

○ c. ABC123; rm -rf /

○ d. ABC123 && mv /etc/passwd /dev/null

Question 6

Complete

Mark 1.00 out of 1.00

What is the best way to prevent command injection attacks?

○ a. Allow user input directly in system commands

○ b. Use an insecure API to execute shell commands

○ c. Use eval() to process user input

● d. Use parameterized queries and sanitize input

Question 7

Complete

Mark 1.00 out of 1.00

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

● a. if (decoded.role !== 'admin') return res.status(403).json({ error: 'Forbidden' });

○ b. if (decoded.role !== 'user') return res.status(403).json({ error: 'Forbidden' });

○ c. if (decoded.id === 1) return res.status(403).json({ error: 'Forbidden' });

○ d. if (!decoded.role) return res.status(403).json({ error: 'Forbidden' });

Question 8

Complete

Mark 1.00 out of 1.00

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

○ a. Attackers can execute arbitrary commands on the server

● b. Unauthorized users can access restricted information or perform admin actions

○ c. The database gets automatically deleted

○ d. It allows Cross-Site Scripting (XSS)

Question 9

Complete

Mark 1.00 out of 1.00

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

● a. Lack of input validation when executing system commands

○ b. Poor network security configuration

○ c. Incorrect use of loops in JavaScript

○ d. Using HTTPS instead of HTTP

Question 10

Complete

Mark 1.00 out of 1.00

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

○ a. Using eval()

○ b. Using exec() with user input

● c. Using execFile() with sanitized input

○ d. Concatenating user input into system commands

Question 11

Complete

Mark 1.00 out of 1.00

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 return JSON data

● c. It does not verify the user’s role before returning data

○ d. It is vulnerable to SQL injection

Question 12

Complete

Mark 1.00 out of 1.00

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

○ a. The token would expire immediately

○ b. They would get logged out

● c. They could access admin-only features

○ d. The server would detect the modification and reject the request

Question 13

Complete

Mark 1.00 out of 1.00

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

● a. exec()

○ b. parseInt()

○ c. console.log()

○ d. JSON.stringify()

Question 14

Complete

Mark 1.00 out of 1.00

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

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

Question 15

Complete

Mark 1.00 out of 1.00

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