

Started on Monday, 17 March 2025, 3:32 PM**State** Finished**Completed on** Monday, 17 March 2025, 3:40 PM**Time taken** 8 mins 32 secs**Marks** 13.00/15.00**Grade** 86.67 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. Command Injection
- ☐ b. Cross-Site Scripting (XSS)
- ☐ c. CSRF Attack
- ☐ d. SQL Injection

Question 2

Complete

Mark 0.00 out of 1.00

How can Broken Access Control be exploited?

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

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

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

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 && mv /etc/passwd /dev/null
- ☒ c. ABC123 && del C:\Windows\System32
- ☐ d. ABC123; rm -rf /

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 parameterized queries and sanitize input
- ☐ c. Use eval() to process user input
- ☐ d. Use an insecure API to execute shell commands

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. The database gets automatically deleted
- ☐ c. It allows Cross-Site Scripting (XSS)
- ☒ d. Unauthorized users can access restricted information or perform admin actions

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. Using HTTPS instead of HTTP
- ☐ c. Incorrect use of loops in JavaScript
- ☐ d. Poor network security configuration

Question 10

Complete

Mark 1.00 out of 1.00

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

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

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

Question 12

Complete

Mark 0.00 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. parseInt()
- ☐ b. console.log()
- ☐ c. JSON.stringify()
- ☒ d. exec()

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. Validate user roles and permissions before processing requests
- ☐ c. Store JWT tokens in Local Storage without encryption
- ☐ d. Remove authentication from sensitive endpoints

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