Casablanca Maintenance HOLMES Security/Vulnerability Report

This table represents the known exploitable and non-exploitable vulnerabilities in third party packages used in the project.

Repository	Group	Impact Analysis	Action
holmes-common	com.fasterxml.jackson.core	False Positive Explanation: This vulnerability issue only exists if com.fasterxml.jackson.databind.ObjectMapper.setDefaultTyping() is called before it is used for deserialization. holmes-common does not use ObjectMapper for serialization/deserialization of JSON objects. Instead, Holmes uses GSON to avoid the vulnerability issues. The reason this is detected is that jackson-databind is introduced indirectly by msb-java-sdk. Also, the MSB team has declared this to be a false positive.	
holmes-dsa	com.fasterxml.jackson.core	False Positive Explanation: This vulnerability issue only exists if com.fasterxml.jackson.databind.ObjectMapper.setDefaultTyping() is called before it is used for deserialization. holmes-dsa does not use ObjectMapper for serialization/deserialization of JSON objects. Instead, Holmes uses GSON to avoid the vulnerability issues. The reason this is detected is that jackson-databind is introduced indirectly by msb-java-sdk. Also, the MSB team has declared this to be a false positive.	
holmes-engine-management	com.fasterxml.jackson.core	Explanation: This vulnerability issue only exists if com.fasterxml.jackson.databind.ObjectMapper.setDefaultTyping() is called before it is used for deserialization. holmes-engine-management does not use ObjectMapper for serialization/deserialization of JSON objects. Instead, Holmes uses GSON to avoid the vulnerability issues. The reason this is detected is that jackson-databind is introduced indirectly by dropwizard-core. To solve the problem, we have to replace the framework of Holmes or wait for updates from Dropwizard. From Homles perspective, we don't use Jackson for JSON data processing. So this is not a big deal for Holmes.	Need to update Dropwizard to check whether its new version has solved this problem. Otherwise, we have to switch to another framework.
holmes-rule-management	com.fasterxml.jackson.core	Explanation: This vulnerability issue only exists if com.fasterxml.jackson.databind.ObjectMapper.setDefaultTyping() is called before it is used for deserialization. holmes-rule-management does not use ObjectMapper for serialization/deserialization of JSON objects. Instead, Holmes uses GSON to avoid the vulnerability issues. The reason this is detected is that jackson-databind is introduced indirectly by dropwizard-core. To solve the problem, we have to replace the framework of Holmes or wait for updates from Dropwizard. From Homles perspective, we don't use Jackson for JSON data processing. So this is not a big deal for Holmes.	Need to update Dropwizard to check whether its new version has solved this problem. Otherwise, we have to switch to another framework.