

Casablanca MSB 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
msb-apigateway	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. msb-apigateway doesn't invoke this method, and a concrete java type is explicitly specified when deserializing the JSON objects, so this vulnerability issue has no impact on msb-apigateway. https://github.com/FasterXML/jackson-docs/wiki/JacksonPolymorphicDeserialization msb-apigateway codes using ObjectMapper: https://gerrit.onap.org/r/gitweb?p=msb/apigateway.git;a=blob;f=apiroute/apiroute-service/src/main/java/org/onap/msb/apiroute/wrapper/util/ConfigUtil.java;h=e0318ba32b5c02f5d2a6ed3287560a0bccd057eb;hb=refs/heads/master https://gerrit.onap.org/r/gitweb?p=msb/apigateway.git;a=blob;f=apiroute/apiroute-service/src/main/java/org/onap/msb/apiroute/wrapper/util/Jackson.java;h=1e5abd148f41003fc46370157fe6dc671b124340;hb=refs/heads/master https://gerrit.onap.org/r/gitweb?p=msb/apigateway.git;a=blob;f=apiroute/apiroute-service/src/main/java/org/onap/msb/apiroute/wrapper/util/JacksonJsonUtil.java;h=1e2fe6d39d9a2d6b3ad3db9c984e22fea1ae2d66;hb=refs/heads/master	Raised a ticket to LF: https://rt.linuxfoundation.org/Service/Display.html?id=53375
msb-apigateway	org.springframework	This vulnerability issue is an indirect dependency introduced by msb-swagger-sdk, which is a false positive. msb-apigateway doesn't use Common-collection lib in its codes.	
msb-apigateway	commons-collections	This vulnerability issue is an indirect dependency introduced by msb-swagger-sdk, which is a false positive. msb-apigateway doesn't use Common-collection lib in its codes.	Raised a ticket to LF: https://rt.linuxfoundation.org/Service/Display.html?id=53375
msb-apigateway	commons-beanutils	This vulnerability issue is an indirect dependency introduced by msb-swagger-sdk, which is a false positive. msb-apigateway doesn't use Common-collection lib in its codes.	Raised a ticket to LF: https://rt.linuxfoundation.org/Service/Display.html?id=53375
msb-apigateway	org.webjars.bootstrap	The bug has been fixed in upstream project, but not included in its latest release yet, it will be updated once the upstream project release the new version.	
msb-apigateway	org.webjars.bootstrap	The bug has been fixed in upstream project, but not included in its latest release yet, it will be updated once the upstream project release the new version.	
msb-apigateway	org.webjars.bootstrap	The bug has been fixed in upstream project, but not included in its latest release yet, it will be updated once the upstream project release the new version.	
msb-discovery	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. msb-discovery doesn't invoke this method, and a concrete java type is explicitly specified when deserializing the JSON objects, so this vulnerability issue has no impact on msb-discovery. https://github.com/FasterXML/jackson-docs/wiki/JacksonPolymorphicDeserialization msb-discovery codes using ObjectMapper: https://gerrit.onap.org/r/gitweb?p=msb/discovery.git;a=blob;f=sdclient/discovery-service/src/main/java/org/onap/msb/sdclient/wrapper/consul/Consul.java;h=a8851fcca579023e1192502c0dbcb17276197ba9;hb=refs/heads/master https://gerrit.onap.org/r/gitweb?p=msb/discovery.git;a=blob;f=sdclient/discovery-service/src/main/java/org/onap/msb/sdclient/wrapper/consul/util/Jackson.java;h=28cfb5b9fe20c009a007bc097e0c27d69553a99d;hb=refs/heads/master https://gerrit.onap.org/r/gitweb?p=msb/discovery.git;a=blob;f=sdclient/discovery-service/src/main/java/org/onap/msb/sdclient/wrapper/consul/util/ObjectMapperContextResolver.java;h=69fea06fdf4d6a12ebe0c7a7b72c957194365e0b;hb=refs/heads/master https://gerrit.onap.org/r/gitweb?p=msb/discovery.git;a=blob;f=sdclient/discovery-service/src/main/java/org/onap/msb/sdclient/wrapper/util/JacksonJsonUtil.java;h=acbb17d8af9e4e9dcff886045786fac252cb08c4;hb=refs/heads/master	Raised a ticket to LF: https://rt.linuxfoundation.org/Service/Display.html?id=53375

msb-discovery	com. smoketurne r.dropwizard	It's an example brought by indirect dependency. Because the codes are not called by MSB, so it has no impact to MSB and it's downstream project. However, it should and will be deleted from MSB dependencies in the next release.	
msb-java-sdk	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. msb-discovery doesn't invoke this method, and a concrete java type is explicitly specified when deserializing the JSON objects, so this vulnerability issue has no impact on msb-discovery. https://github.com/FasterXML/jackson-docs/wiki/JacksonPolymorphicDeserialization msb-java-sdk codes using ObjectMapper: https://gerrit.onap.org/r/gitweb?p=msb/java-sdk.git;a=blob;f=src/main/java/org/onap/msb/sdk/discovery/Util/JacksonJsonUtil.java;h=aac256a847853721e7caffd8e876dc145fc4e16b;hb=refs/heads/master https://gerrit.onap.org/r/gitweb?p=msb/java-sdk.git;a=blob;f=src/main/java/org/onap/msb/sdk/httpclient/convert/jackson/JacksonConverterFactory.java;h=4170cbdd88376e8471443c17c81b10c7c1df84df;hb=refs/heads/master https://gerrit.onap.org/r/gitweb?p=msb/java-sdk.git;a=blob;f=src/main/java/org/onap/msb/sdk/httpclient/convert/jackson/JacksonConverterFactoryBuilder.java;h=e0477ad9ebd197d68b25b91f0823a3c695a48ab6;hb=refs/heads/master	Raised a ticket to LF:https://rt.linuxfoundation.org/SelfService/Display.html?id=53375
msb-java-sdk	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. msb-discovery doesn't invoke this method, and a concrete java type is explicitly specified when deserializing the JSON objects, so this vulnerability issue has no impact on msb-discovery. https://github.com/FasterXML/jackson-docs/wiki/JacksonPolymorphicDeserialization msb-java-sdk codes using ObjectMapper: https://gerrit.onap.org/r/gitweb?p=msb/java-sdk.git;a=blob;f=src/main/java/org/onap/msb/sdk/discovery/Util/JacksonJsonUtil.java;h=aac256a847853721e7caffd8e876dc145fc4e16b;hb=refs/heads/master https://gerrit.onap.org/r/gitweb?p=msb/java-sdk.git;a=blob;f=src/main/java/org/onap/msb/sdk/httpclient/convert/jackson/JacksonConverterFactory.java;h=4170cbdd88376e8471443c17c81b10c7c1df84df;hb=refs/heads/master https://gerrit.onap.org/r/gitweb?p=msb/java-sdk.git;a=blob;f=src/main/java/org/onap/msb/sdk/httpclient/convert/jackson/JacksonConverterFactoryBuilder.java;h=e0477ad9ebd197d68b25b91f0823a3c695a48ab6;hb=refs/heads/master	
msb-swagger-sdk	com. fasterxml. jackson. core	False Positive It does not cause any issues to the services and service does not expose functionality directly	LF already waived it
msb-swagger-sdk	com. fasterxml. jackson. core	False Positive It does not cause any issues to the services and service does not expose functionality directly	LF already waived it
msb-swagger-sdk	com. fasterxml. jackson. dataformat	False Positive It does not cause any issues to the services and service does not expose functionality directly	LF already waived it
msb-swagger-sdk	commons-beanutils	False Positive It does not cause any issues to the services and service does not expose functionality directly	LF already waived it
msb-swagger-sdk	commons-collections	False Positive It does not cause any issues to the services and service does not expose functionality directly	LF already waived it