

[ExtAPI] Security/Vulnerability Threat Template

This template is intended to be used to document the outcome of the impact analysis related to the known vulnerability reported by Nexus-IQ ([CLM tab in Jenkins](#)). Nexus-IQ can identify the known vulnerabilities contained in the components use by onap components.

Report in the template below only the vulnerabilities that Nexus-IQ is reporting as **"Critical"** (Level 7 to 10).

This table will be presented to the TSC at Code Freeze milestone (M4).

It is recommended to **first update to the latest version** of the third party components available. In case the latest third party components still reports some vulnerabilities, you must provide an impact analysis as illustrated in the example below.

In the case where you have nested third party components (a third party component embedding another third party component) and there is **NO CVE** number for the upstream third party component (meaning the third party component you are embedding), it is recommended to open a vulnerability issue on the upstream third party component.



Usage

Please make a **Copy** of this template into your project wiki space. Be sure to make a Copy (not a Move) by using the ... on the top right corner of this page

Within the M4 checklist create a link toward your copy of this template

Once this template has been copied into your project wiki space, you can delete this "Tip" section as well as the "Sample of CLM Report" screenshot. This screenshot is just an example.

The following table is addressing 2 different scenarios:

- Confirmation of a vulnerability including an action
- False Positive

The information related to Repository, Group, Artifact, Version and Problem Code are extracted from the CLM report (see the below screenshot)

Repository	Group	Impact Analysis	Action
externalapi-nbi	ch.qos.logback	Project is impacted. Change to 1.2.3	blocked URLEXTAPI-71 - Security: logback-classic in externalapi-nbi DONE https://geritt.onap.org/r/#/c/42691/
externalapi-nbi	com.fasterxml.jackson.core	False Positive. The exploit primarily is about enabling polymorphic type handling with the object mapper and writing class specifics into the JSON object. There are two ways of doing this: 1. ObjectMapper.enableDefaultTyping() 2. @JsonTypeInfo for marshalling / unmarshalling an object By default the ObjectMapper does not enableDefaultTyping, the sparky/data-router code bases are not using either approach, so the possibility of the exploit vector does not apply.	
externalapi-nbi	commons-beanutils	Project is impacted by this vulnerability.	blocked URLEXTAPI-72 - Security: commons-beanutils in externalapi-nbi DONE https://geritt.onap.org/r/#/c/42273/

CLM Report

You can check the latest CLM report for ExternalAPI [here](#)