

Open Source License Compliance Proposal for Dependency-Track

Draft 2018-12-10

Contributors: Henry Haverinen, Kim Paananen, Sami Lempinen,
Juuso Kanner

Introduction

- This presentation is a draft concept for adding open source license compliance functions to Dependency-Track

License policy

- Whether an OSS license is acceptable will depend on the organization and the project.
- Sometimes architecture decisions such as dynamic vs static linking will impact the decision
- Proposed OSS license policy states:
 - **Uncategorized:** license policy has not been defined for this license (default)
 - **Blacklisted:** this license is not acceptable for use
 - **Whitelisted:** this license is acceptable for use
 - **Case-by-case:** whether the license is acceptable needs to be analysed separately per project

The scope of the license policy

- For simplicity, it is proposed to have a single global license policy in Dependency-Track, rather than multiple policies or project specific policies
- Especially if the policy includes the "case-by-case" option, a single global policy is expected to be good enough for most users

Defining the license policy

DEPENDENCY-TRACK Search... About Profile Logout

Name	SPDX License ID	OSI Approved	License Policy
389 Directory Server Ex..	389-exception	No	Uncategorized
3dfx Glide License	Glide	No	Uncategorized
Abstyles License	Abstyles	No	Uncategorized
Academic Free License ...	AFL-1.1	Yes	Uncategorized
...	Whitelisted

OSI Approved options: Yes, No

License Policy options: Uncategorized, Whitelisted, Blacklisted, Case-by-case

"OSI Approved" is not user editable, so we could use "Yes" or "No" instead of a checkbox

The user can define the OSS License policy by selecting one of the options from a drop-down list

Importing and exporting the license policy

- As an enhancement, it would be nice to be able to export the policy to a file and import a policy to the tool from a file

About license audit

- License audit is very similar to vulnerability audit
- License audit can be done per project or per component similarly as a vulnerability audit
- Analysis works similarly to the analysis of vulnerabilities
- Suppressing works similarly to suppressing vulnerabilities. Suppressing will impact the metrics, and it should be used mainly for possible "false positives" and case-by-case approved licenses.

License audit for a project

The screenshot displays the Dependency-Track web application interface. At the top, the 'DEPENDENCY-TRACK' logo is visible. The main header shows 'Test Project' with a dropdown menu set to '1' and a 'test' label. On the right, there are navigation links for 'About', 'Profile', and 'Logout', along with a risk score of '0' and 'Inherited Risk Score'. Below the header, a navigation bar contains tabs for 'Overview', 'Dependencies', 'Audit', and 'License audit'. The 'License audit' tab is currently selected. A search bar and a refresh button are located below the navigation bar. At the bottom, a table header is visible with columns: Component, Version, Group, Vulnerability, CWE, Severity, Analysis, and Suppressed. Two callout boxes provide instructions: one pointing to the 'Audit' tab with the text 'Rename the current Audit as "Security audit" or "Vulnerability audit"', and another pointing to the 'License audit' tab with the text 'Add a new tab "License audit"'.

Rename the current Audit as "Security audit" or "Vulnerability audit"

Add a new tab "License audit"

DEPENDENCY-TRACK

About Profile Logout

Test Project 1

1 test

View Details

Overview Dependencies Audit License audit

Search

Component Version Group Vulnerability CWE Severity Analysis Suppressed


Medium Severity: 0
Low Severity: 0

Inherited Risk Score 0

License analysis options

- License analysis options in license audit:
 - NOT_SET: License audit has not been done
 - APPROVED: the license has been approved
 - REJECTED: the license has been rejected
- For whitelisted licenses, the initial analysis value is APPROVED
- For blacklisted licenses, the initial analysis value is REJECTED
- For uncategorized licenses, the initial analysis value is NOT_SET
- For case-by-case licenses, the initial analysis value is NOT_SET
- For components with an unknown license, the initial analysis value is NOT_SET

License audit

 **DEPENDENCY-TRACK** [About](#) [Profile](#) [Logout](#)

[Overview](#) [Dependencies](#) [Audit](#) **[License audit](#)**

Component	Version	SPDX License ID	License Policy	License Analysis	Suppressed
OpenSSL	1.0.1e	OpenSSL	Uncategorized	APPROVED	<input checked="" type="checkbox"/>
Example tech	1.52.0	LGPL-3.0	Case-by-case	REJECTED	<input type="checkbox"/>

License

Comment

[Add Comment](#)

Audit trail

henry – 10 Dec 2018 at 12:17:31
The application is statically linked, so the license is not acceptable

henry – 10 Dec 2018 at 12:18:18
NOT_SET → REJECTED

Analysis

- Rejected
- Approved

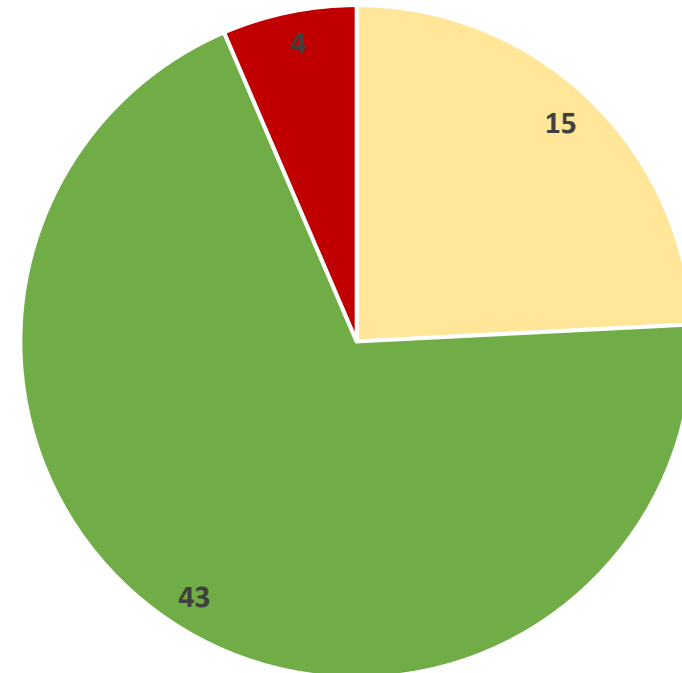
Re-auditing the licenses after the BoM is updated

- When a license issue is audited/suppressed and the BoM is later updated, then the previous manual audit analysis decisions and suppression decisions should still remain, at least for those components whose version does not change
- However, if the version of a component changes, then it is possible that the previous manual approval/suppression decisions may become invalid, especially for components whose BoM didn't include a license. The simplest solution would be to restore the analysis state to the default value as dictated by the policy, and suppression state to unsuppressed. However, there could also be some other softer state that suggests that the user should review the license audit decisions to make sure they are still valid.

License compliance dashboard for a project

- Unprocessed license issues =
of components with an uncategorized license or with an unprocessed case-by-case license (License analysis NOT_SET and license issue is not suppressed)
- Approved licenses =
of components with a whitelisted or an approved case-by-case license (License analysis APPROVED)
- Rejected licenses =
of components with a blacklisted or a rejected case-by-case license (License analysis REJECTED and license issue is not suppressed)

Project X License Compliance



■ Unprocessed license issues ■ Approved licenses ■ Rejected licenses ■

License compliance dashboard for portfolio

- Projects with license issues = number of projects that use a component that has an uncategorized license, a blacklisted license, a rejected case-by-case licenses, or an unprocessed case-by-case licenses and the issue is not suppressed



0

Projects at Risk



0

Projects with license issues