Analyzing Resource Scan Results

Overview

JFrog Xray scans and displays Xray data in the JFrog Platform providing radical transparency about any infected components or license breaches in your software. Xray data is located within each of your resource pages allowing you to quickly review the status of for your scanned resources - Packages, Package Versions, Builds, Artifacts or Release Bundles.

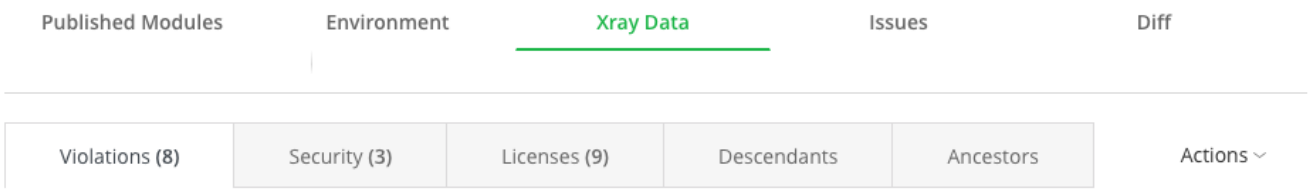
If need be you can drill down to gain radical transparency about any infected components or license breaches in your software as described in the following section.

Page Contents

- Overview
- Analyzing Detailed Scanned Data on Resources
  - Violations
  - Security
  - Determining the Issue Severity Level for Operating Systems Packages
  - Licenses
  - Descendents
  - Ancestors
- Xray Actions
  - Scanning for Violations
  - Assigning Custom Issues
  - Assigning Custom Licenses
  - Exporting Xray Data

Analyzing Detailed Scanned Data on Resources

Each of the scanned resources - packages, builds, artifacts and Release Bundles contains the following set of Xray sub tabs and a list of actions.



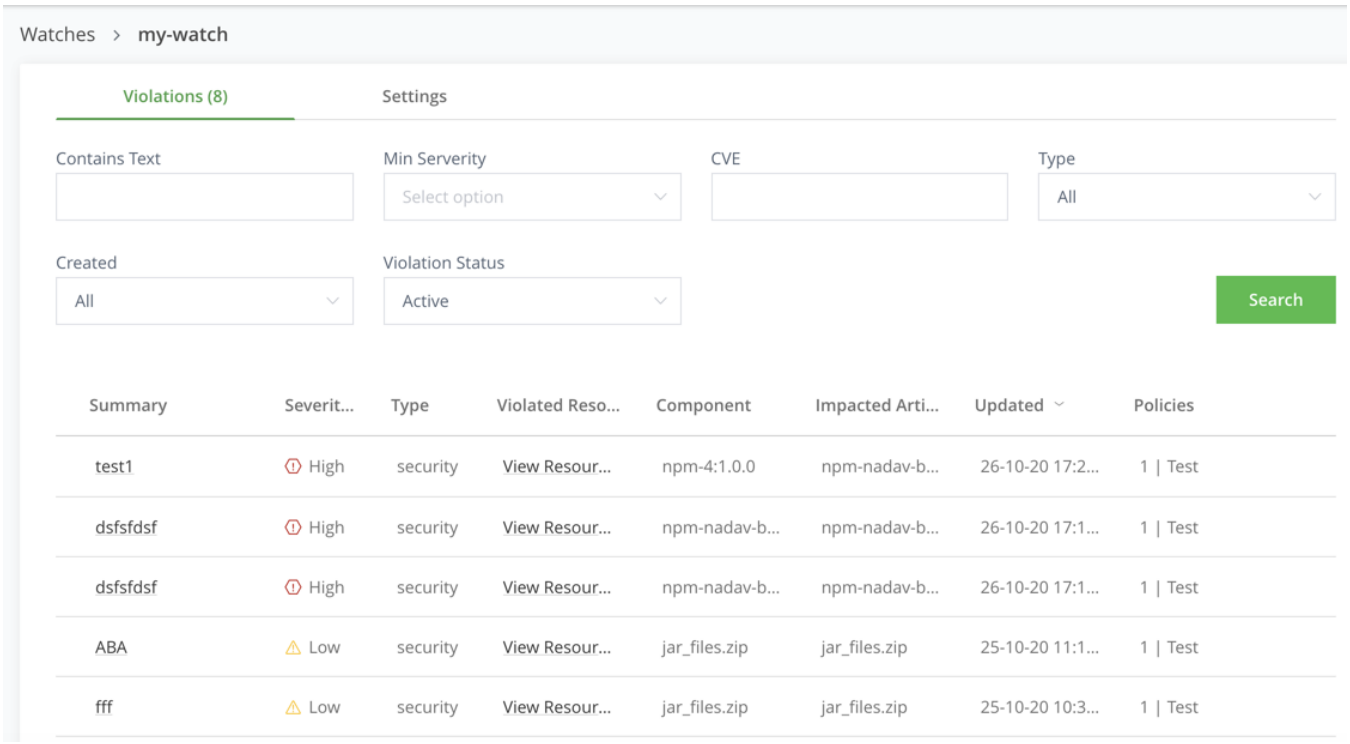
The Xray Data sub tabs are:

- Violations: These are violations to filters defined on a watch. They are only reported for the root component, not for its dependencies.
- Security: Known security vulnerabiliites for the selected component.
- Licenses: OSS licenses used by the component.
- Decedents: Components that the selected component includes (depends on).
- Ascendants: Components that include (depend on) the selected component.

The following sections describe the Xray Data sub tabs displaying the Packages resource as an example. Please note the tabs are identical for builds, artifacts and Release Bundles.

Violations

Displays the violations detected on the package version based on the watches and associated policies set by the users. You can view the vulnerability severity, type and the associated policies. To view a components and its dependencies, click on the Component icon. In some cases, when violations are detected, as security or legal personnel, you would like to accept or to add some of these violations to an Allow List. For more information, see Ignore Rules.



Violation Details

CVE: CVE-2021-3807
Type: Violation

Violation Type:	security
CVEs:	CVE-2021-3807
CVSS V2:	7.8
CVSS V3:	7.5
Fixed Versions:	5.0.1 6.0.1
Impacted Artifact:	generic://sha256:8cf923fec8a60f4db57fbf7ca79cf0b471b3f0fb57616dc796051d07aaa153f6/jfrog-artifactory-pro-7.27.10-darwin.tar.gz
Project Names:	
Issue ID:	XRAY-185075
Infected Component:	npm://ansi-regex:4.1.0
Package Type:	npm
Policy Names:	p1
Provider:	JFrog
Vulnerability Published Date:	2021-09-19T15:11:59.79+03:00
Component Physical Path:	artifactory-pro-7.27.10/app/frontend/bin/server/dist/node_modules/qrcode/package.json/node_modules/ansi-regex/package.json
References:	https://github.com/chalk/ansi-regex/commit/8d1d7cdb586269882c4bdc1b7325d0c58c8f76f9

Vulnerability Details

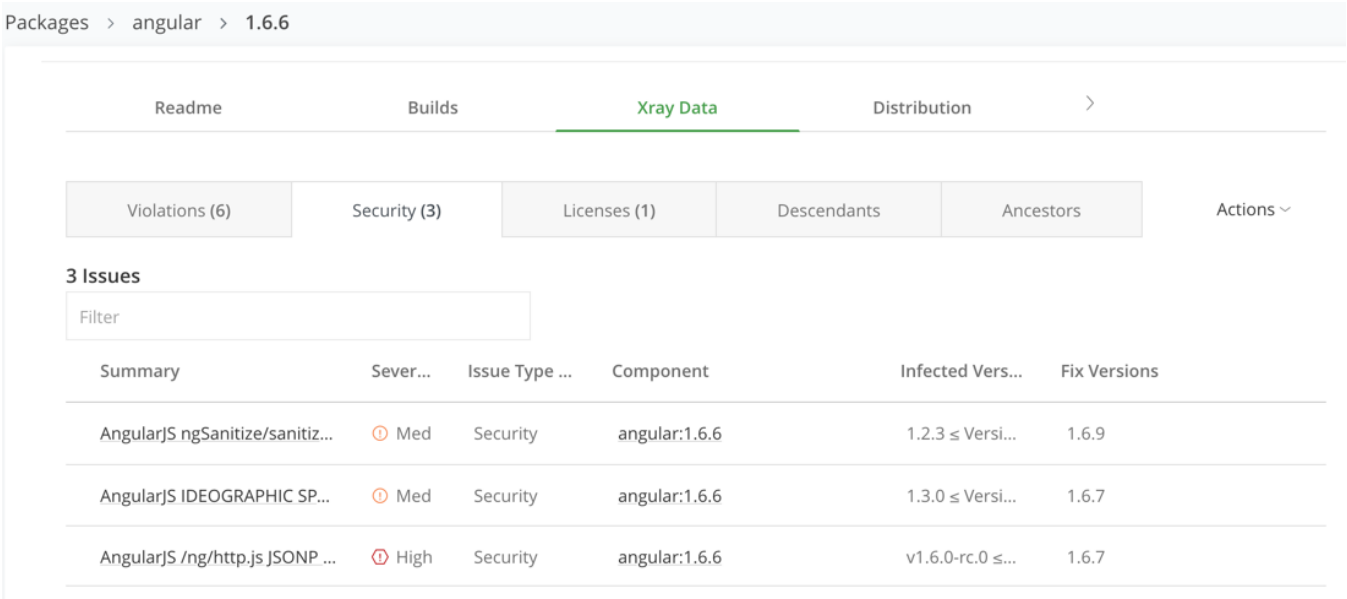
CVE: CVE-2021-31684
Type: Vulnerability

Summary:	A vulnerability was discovered in the indexOf function of JSONParserByteArray in JSON Smart versions 1.3 and 2.4 which causes a denial of service (DOS) via a crafted web request.
Severity	⚠ High
Vulnerable Component:	gav://net.minidev:json-smart:2.4.2
Component Physical Path:	sha256__358cfb7bd06c8eacda26affd1f4aeb815028e2d7171626294ce0b05bb4144c73.tar.gz/var/lib/neo4j/labs/apoc-4.4.0.2-core.jar/META-INF/maven/net.minidev/json-smart/pom.xml
Impacted Artifact:	docker://neo4j:latest
Path:	docker-local/neo4j/latest/
Project Names:	
Fixed Versions:	1.3.3 2.4.5
Published:	2021-06-03T17:02:16+03:00
Artifact Scan Time:	2022-02-27T14:55:05+02:00
Issue ID:	XRAY-177224
CVSS V2:	5
CVSS V3:	7.5
Package Type:	maven
Provider:	JFrog

Physical Path of Vulnerable Component

Security

Displays the known security vulnerabilities for the selected package version and the effected versions and fixed versions that do not contain the vulnerability.



To examine the details of a violation, click the violation in the list to display the Issues Details popup.

Determining the Issue Severity Level for Operating Systems Packages

Xray initially populates data about vulnerabilities and licenses from the Xray global database server managed by JFrog. After the initial database synchronisation, Xray is then continuously synchronized with the central database for new updates on a daily basis.

When analyzing the vulnerabilities for open source operating systems packages, Xray fetches data regarding the severity of the vulnerability from two sources:

- **NVD**: The National Vulnerability Database which contains known vulnerabilities each with their CVSS score. For more information on CVSS scoring in Xray, see CVSS Scoring in Xray.
- **Security Advisory**: Some open source operating systems have their own security trackers with further analysis of the vulnerability inside the operating system package.

In the case where the Operating System Security Advisory contains data about the vulnerability in a package, Xray will compute the severity of the vulnerability based on this data instead of the CVSS score of the vulnerability.

The reason for that, is that the Security Advisory team of the Operating System had done further analysis to come to a more precise conclusion regarding the priority/urgency/severity of the vulnerability inside the operating system package.

To help you understand how Xray maps the information from each, we have outlined each operating system's severity/priority and how it is presented in Xray.

Take note, that if a vulnerability's severity is unknown in the security advisory, the CVSS score will be calculated from the NVD.

Ubuntu

Vulnerabilities source: Ubuntu CVE Tracker

Severity mapped from: Priority

Priority to Xray Severity mapping:

Ubuntu Priority	Xray Severity
Critical	High
High	High
Medium	Medium
Low	Low
Negligible	Low
Untriaged	Unknown (will use CVSS score from NVD)

Priority to Xray Severity mapping: CVSS v3

Ubuntu Priority	Xray Severity
Critical	Critical
High	High
Medium	Medium
Low	Low
Negligible	Low
Untriaged	Unknown (will use CVSS score from NVD)

Debian

Vulnerabilities source: Debian Security Bug Tracker

Severity mapped from: Urgency

Urgency to Severity mapping:

Debian Urgency	Xray Severity
High	High
Medium	Medium
Low	Low
Unimportant	Low

End of Life	Unknown (will use CVSS score from NVD)

RPM

Vulnerabilities source: Red Hat Security Advisories and CVE database

Severity mapped from: Severity Rating

Red Hat Severity to Severity mapping:

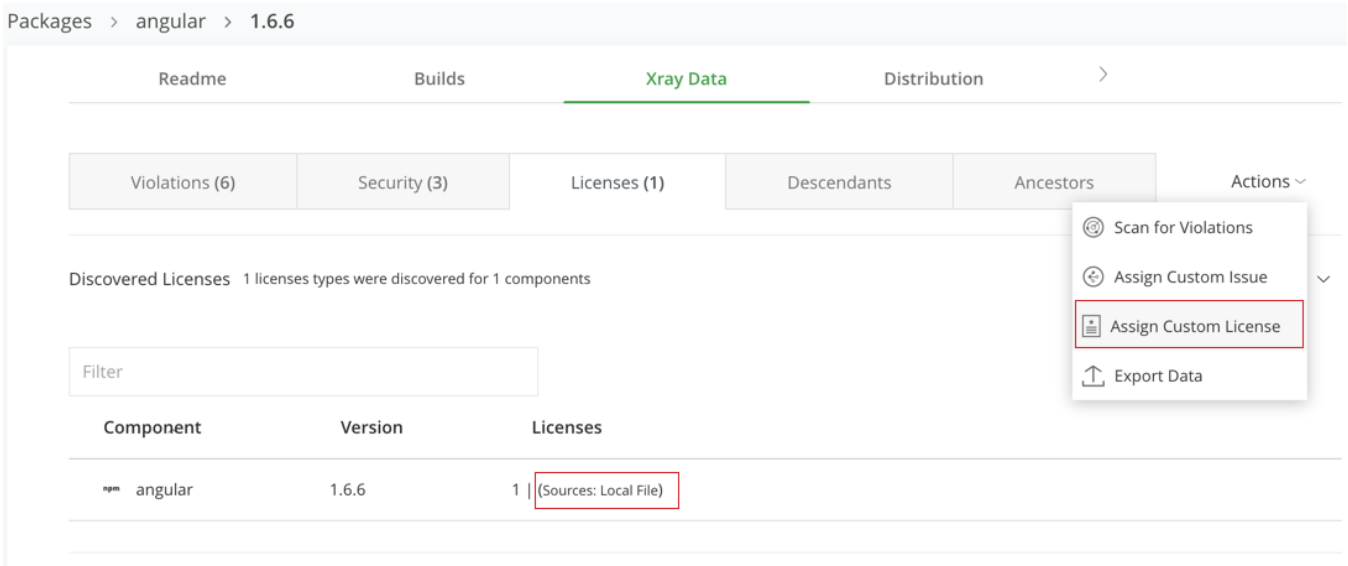
Red Hat Severity	Xray Severity
Critical	High
Important	High
Moderate	Medium
Low	Low

Red Hat Severity to Severity mapping: CVSS v3

Red Hat Severity	Xray Severity
Critical	Critical
Important	High
Moderate	Medium
Low	Low

Licenses

Displays the licenses is assigned to a specific version and triggers violations in case it matches criteria of any existing Watches. Click on the License to view the license attached to the components.

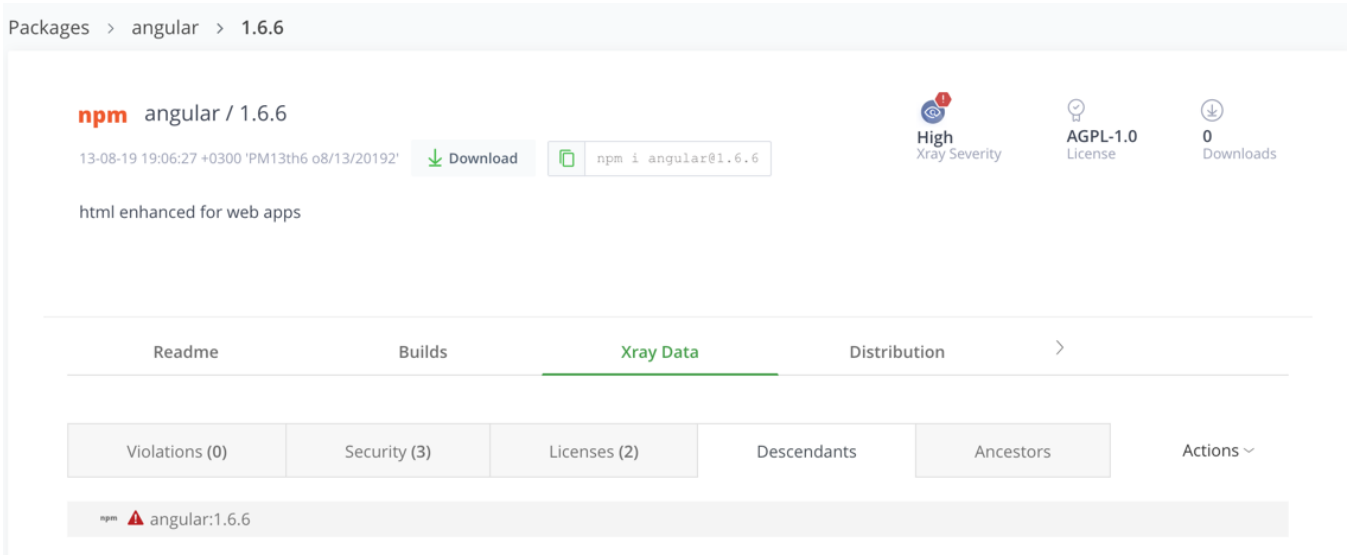


Descendents

Displays the components that the selected component includes (depends on).

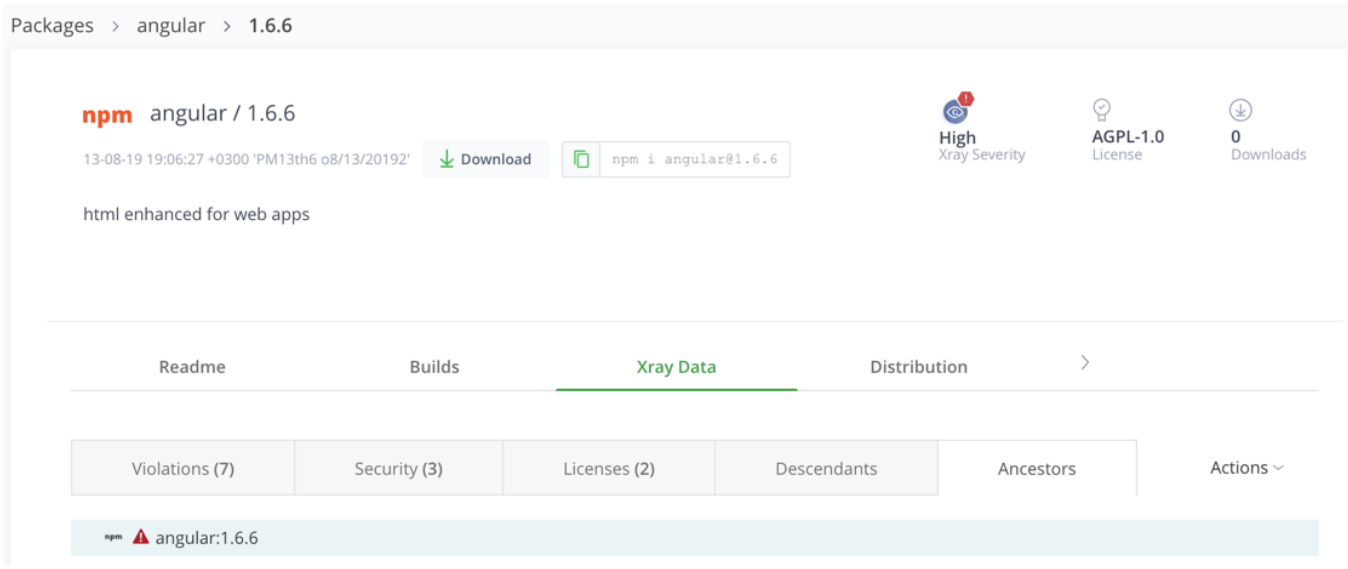
Displays only dependencies that are present within the component. Referenced dependencies that are not included in the package but are referenced in a metadata file present within the package or along side it will not be presented.
For example:

- A Maven `pom.xml` located in the package or/and along side the package jar.
- An NPM `package.json` which can be found inside the package).



Ancestors

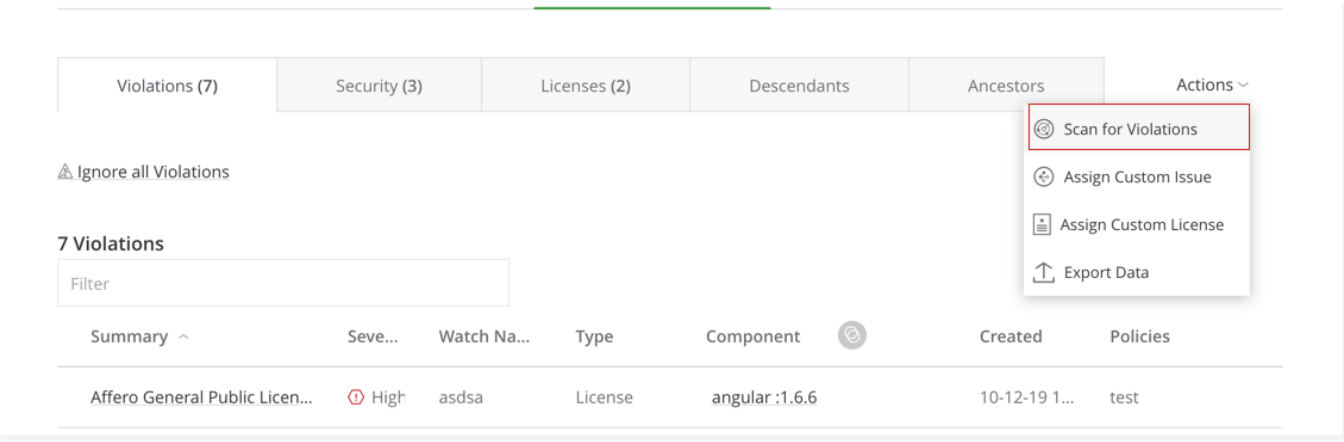
Displays components that include (depend on) the selected component.



Xray Actions

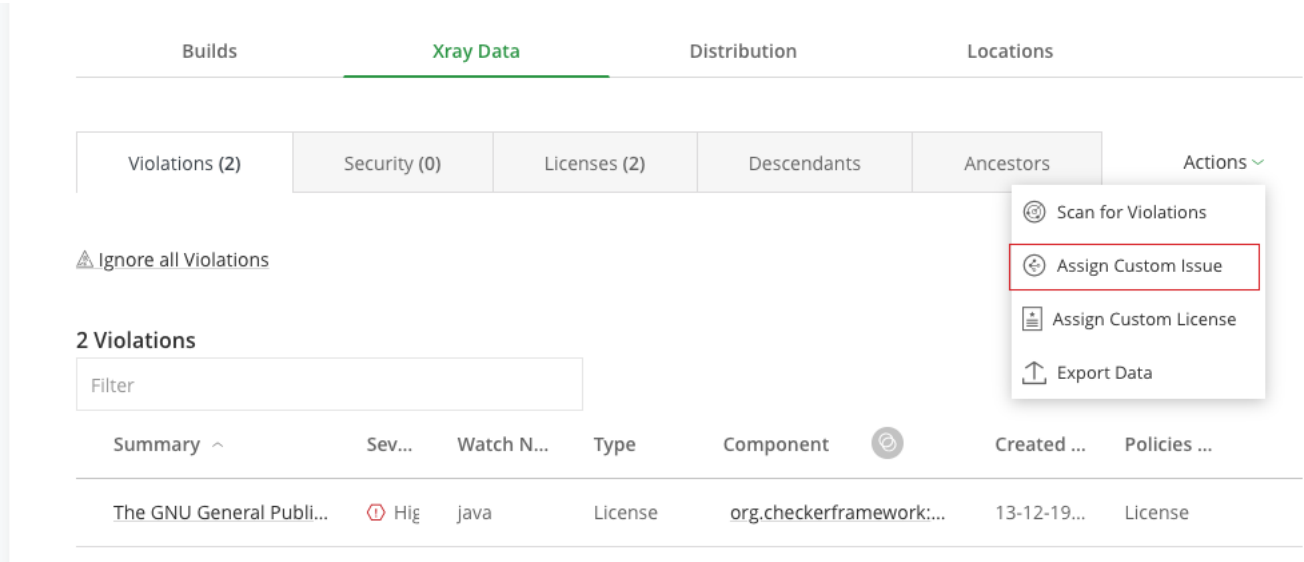
Scanning for Violations

To initiate a manual scan on your package version, select **Scan for Violations** from the Actions list.



Assigning Custom Issues

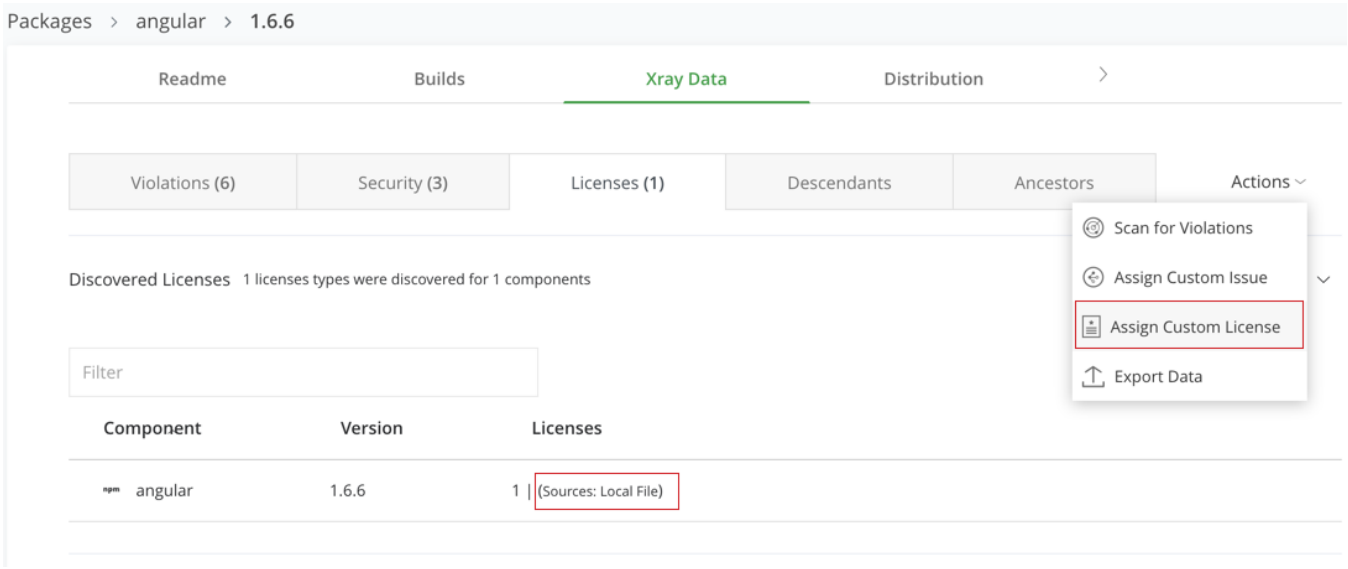
A security vulnerability created by a user is tagged as a Custom issue and can be deleted by users assigned with the Manage Xray Metadata permission.



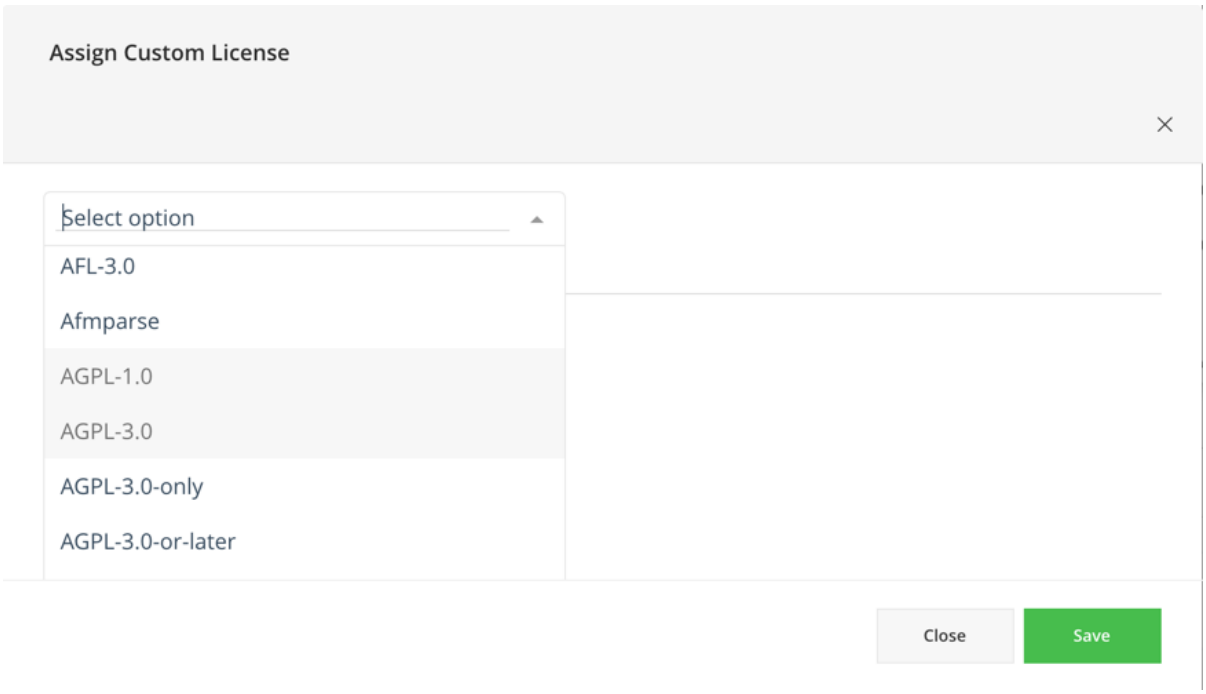
Assigning Custom Licenses

A license created by a user is tagged as a Custom license and can be deleted by

From the **Actions** list, select **Assign a Custom License** to assign a Custom licences on a component in your version.



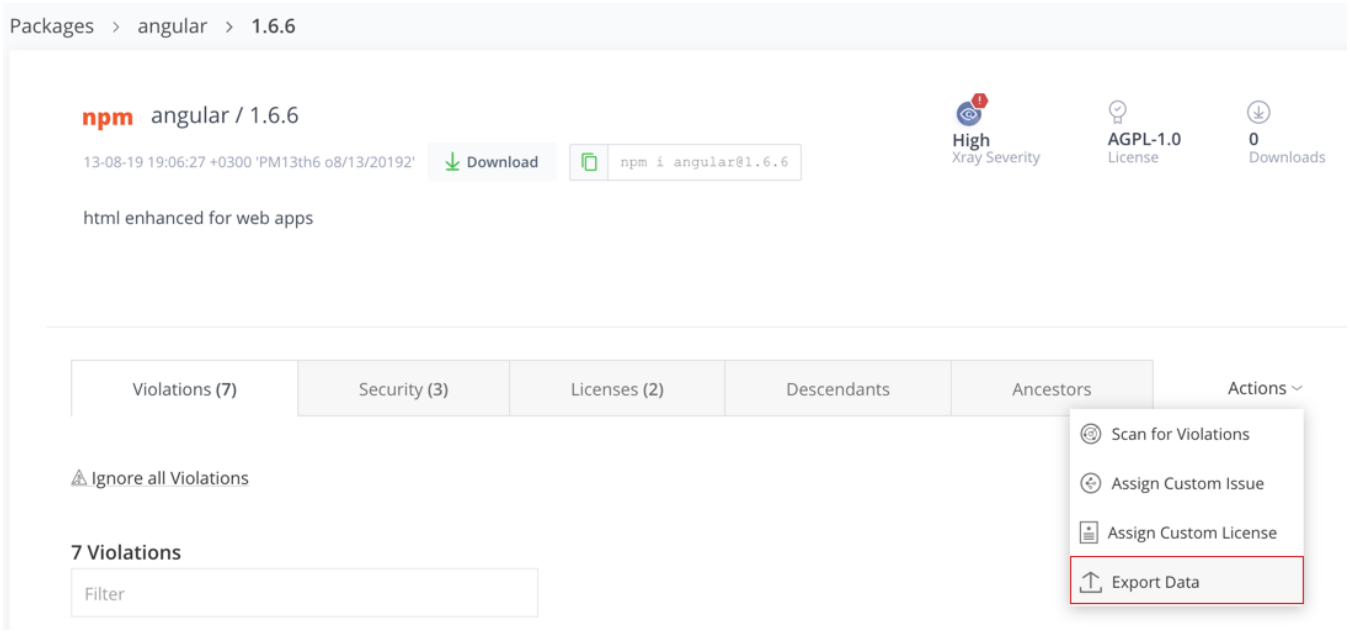
Select a license from a predefined list of licenses.



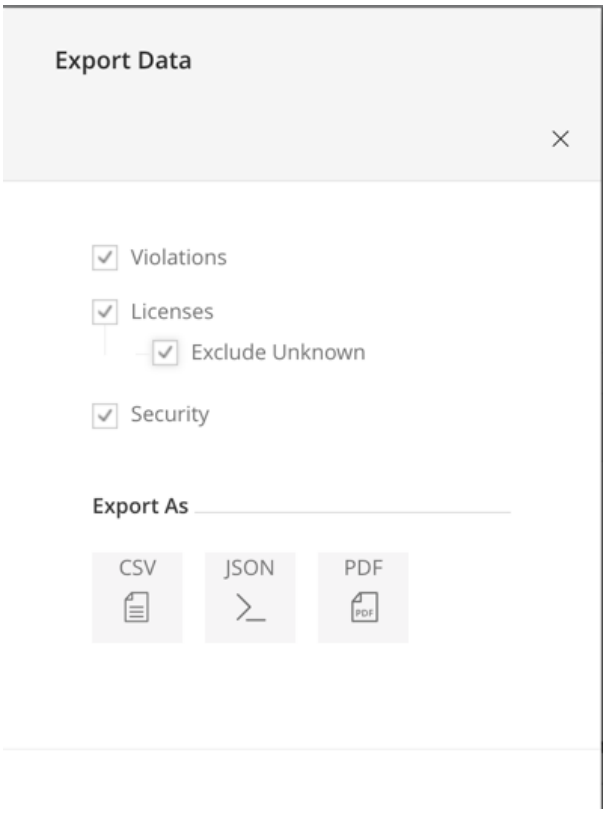
Click **Save**. A manual scan is triggered to update the license list.

Exporting Xray Data

Using the Actions menu, you can export full details for the selected component and version including violations, security issues and licenses. From the Xray Data tab on the package versions page, select **Export Data** from the **Actions** list.



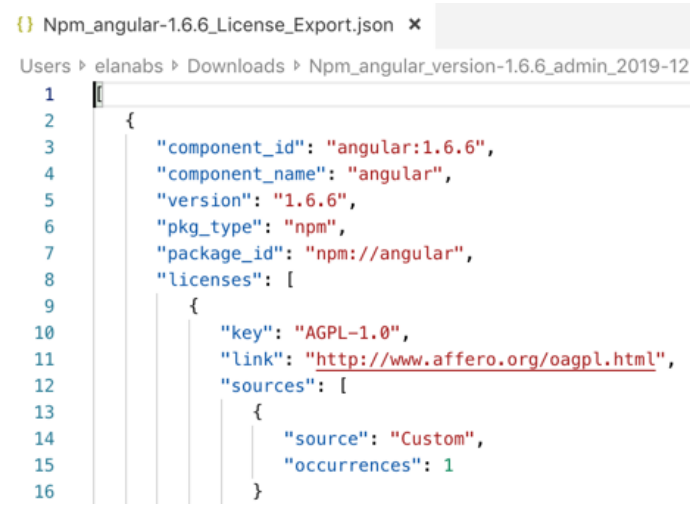
In the following **Export data** popup, specify if you want to export violation, licenses or security parameters that should be exported and the export format.



The file is downloaded to your local drive.

Below are some examples of exported files in different formats.

	A	B	C
1	Component Name	Licenses	Licenses Links
2	angular:1.6.6	AGPL-1.0,	http://www.affero.org/oagpl.html,http://www.affero.org/oagpl.html



{} Npm_angular-1.6.6_License_Export.json ×

Users ▸ elanabs ▸ Downloads ▸ Npm_angular_version-1.6.6_admin_2019-12


```
[
  {
    "component_id": "angular:1.6.6",
    "component_name": "angular",
    "version": "1.6.6",
    "pkg_type": "npm",
    "package_id": "npm://angular",
    "licenses": [
      {
        "key": "AGPL-1.0",
        "link": "http://www.affero.org/oagpl.html",
        "sources": [
          {
            "source": "Custom",
            "occurrences": 1
          }
```


You can also automate exporting component details using the Export Component Details REST API endpoint.