

Insights Components

This section introduces you to the Components tab in Kiuwan Insights.

Contents:

- [Components Inventory](#)
 - [Supported languages and resources](#)
 - [Overall Information on Components](#)
 - [List of Components](#)
 - [Component details](#)
 - [Duplicated components](#)

Components Inventory

Kiuwan Insight analyzes your application software, discovering all external dependencies, and builds a **components inventory** that lets you track any external piece of code that could be part of your application.

Go to **Insights > Components** to access the components inventory.

Supported languages and resources

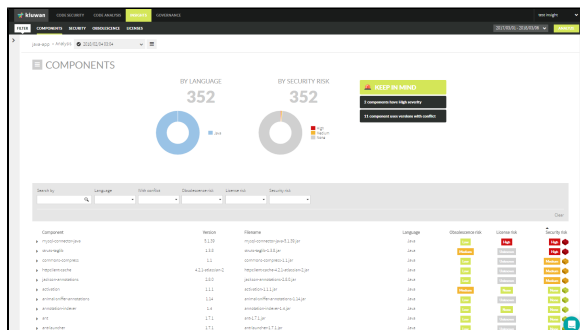
Kiuwan Insights uses the following resources to extract information on 3rd party dependencies.

Supported languages	Supported repositories	Supported build systems	Repositories Used	Database Vulnerabilities Used	Licenses extract from
Go	• GitHub	• go.mod • Gopkg.lock	GitHub: https://github.com/	• NVD: https://nvd.nist.gov/	• GitHub
Java	• Maven • Gradle	• Ant (*.xml files) • Maven (pom.xml files) • Gradle (*.gradle files) • *.jar, *.war, *.ear files	Maven (central or others configured in settings.xml or pom.xml files): https://repo.maven.apache.org/maven2/	• NVD: https://nvd.nist.gov/	• pom.xml • License file into jar file.
Javascript	• Npm • Bower	• Npm (package.json files) • Bower (bower.json files) • Yarn (package.json files)	Npm: https://www.npmjs.com/	• NVD: https://nvd.nist.gov/	• NPM Rest services.
Kotlin	• Maven • Gradle • Ant	• Ant (*.xml files) • Maven (pom.xml files) • Gradle (*.gradle and *.gradle.kts files)	Maven (central or others configured in settings.xml or pom.xml files): https://repo.maven.apache.org/maven2/	• NVD: https://nvd.nist.gov/	• Maven services
.Net	• Nuget	• Nuget (*.csproj, project.json, global.json, *.vbproj files)	Nuget: https://www.nuget.org/	• NVD: https://nvd.nist.gov/	• Nuget Rest services.

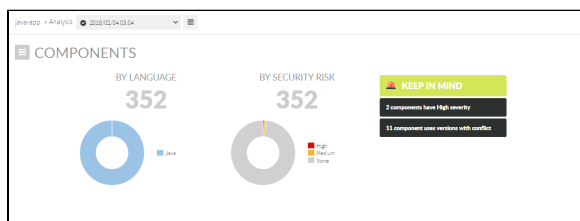
Php	• Packagist	• Composer (composer.json, composer.lock files)	Packagist: https://packagist.org/	• NVD: https://nvd.nist.gov/	• NVD: https://nvd.nist.gov/
Python	• PyPI • GitHub	• PyPI (setup.py files) • Requirements (txt file with declared dependencies)	PyPI: https://pypi.org/	• NVD: https://nvd.nist.gov/	• PyPI Rest services
Ruby	• RubyGems	• Gemfile, Gemfile.lock and *.gemspec files	RubyGems: https://rubygems.org/	• NVD: https://nvd.nist.gov/ • Bundler audit database	• License and obsolescence pending
Scala	• Maven	• SBT (build.sbt)	Maven (central or others configured in settings.xml or pom.xml files): https://repo.maven.apache.org/maven2/	• NVD: https://nvd.nist.gov/	• pom.xml.
Swift	• CocoaPods • GitHub	• Podspec (*.podspec, Podfile.lock files) • Package (Package.swift files)	Repository Podspec in Github: https://github.com/CocoaPods/Specs	• NVD: https://nvd.nist.gov/	• podspec.json of the component.

The **Components Inventory** section shows the following information:

1. Overall Information on Components – aggregated information on number and type of components
2. List of Components – detailed listing of components
3. Component detail – detailed information on selected component



Overall Information on Components



- Number of components by language
- Number of components by Security Risk level (High, Medium, Low and None)
- Alerts :

- Components with High-Security Risk
- Components being used with different versions that might cause conflicts
- Etc.

List of Components

Kiuwan Insights provides a full list of all those components being used by your application.

For every 3rd party component, you will have access to detailed component information such as:

Name	Description
Component name	Name of the component
Version	The version(s) in use
Filename	The physical container (.jar, .dll, .js, etc)
Language	The programming language it is written in.
Obsolescence risk	A component's Obsolescence Risk is a measure of the risk level relative to: - the antiquity of your version respect to the latest version, and - how active is the component Both values are combined in the Obsolescence Risk to provide a value of the risk associated with using outdated or "dead" components. Please visit Obsolescence Risk for further information.
License risk	A component's License Risk is a measure of the risk level relative to the legal implications of used components' licenses. Please visit Insights Licenses for further information.
Security risk	A component's Security Risk is based on CVSS v2 Base Scores (Severities) of its vulnerabilities: - If the selected component has more than one vulnerability, Kiuwan will label the component with the highest severity value of all the vulnerabilities of the component. - If the selected component has only one vulnerability, the Severity of that vulnerability will be the Security Risk of the component.

Component	Version	Filename	Language	Obsolescence risk	License risk	Security risk
mysql-connector-java	5.1.39	mysql-connector-java-5.1.39.jar	java	Low	High	High
stax-api	1.3.3	stax-api-1.3.3.jar	java	Medium	Medium	Medium
commons-compress	1.1	commons-compress-1.1.jar	java	Low	Medium	Medium
httpclient	4.2.1-alpha2	httpclient-core-4.2.1-alpha2.jar	java	Low	Medium	Medium
jackson-core	2.8.0	jackson-core-2.8.0.jar	java	Low	Medium	Medium
activation	1.1.1	activation-1.1.1.jar	java	Medium	Low	Low
animal-sniffer	1.14	animal-sniffer-annotations-1.14.jar	java	Low	Medium	Medium
annotations	1.4	annotations-1.4.jar	java	Low	Medium	Medium
ant	1.7.1	ant-1.7.1.jar	java	Low	Medium	Medium
ant-launcher	1.7.1	ant-launcher-1.7.1.jar	java	Low	Medium	Medium

Component details

By clicking on a component, you will have access to the following information:

- The description of the component
- The license of the component
- Found vulnerabilities of the selected component:
 - CVE identifier, and link to NIST National Vulnerability Database desc page
 - CWE type, and link to MITRE Common Weakness Enumeration desc page
 - Vulnerability description
 - Severity (more on this at [Security Risk](#))

struts-taglib		1.3.8	struts-taglib-1.3.8.jar	Java	Unknown	Struts2	Info
Description	License						
Vulnerabilities							
CVE	CWE	Description					Severity
CVE-2016-1151		AsteriskServlet.java in Apache Struts 1.1.x through 1.3.10 mishandles multithreaded access to an ActionForm instance, which allows remote attackers to execute arbitrary code or cause a denial of service (unreleased memory access) via a multipart request, a related issue to CVE-2016-0999.					Medium
CVE-2016-0999	CVE-20	The ValidatorServlet implementation in Apache Struts 1.1.1 through 1.3.10 allows remote attackers to bypass intended access restrictions via a modified page parameter.					Medium
CVE-2016-1152	CVE-20	AsteriskServlet.java in Apache Struts 1.1.x through 1.3.10 does not properly restrict the validator configuration, which allows remote attackers to conduct response crafting (VDD) attacks to cause a denial of service via crafted input, a related issue to CVE-2016-0999.					Medium
CVE-2014-0114	CVE-20	Apache Commons Beanutils, as distributed in itscommons-beanutils-1.9.0.jar in Apache Struts 1.x through 1.3.10 and in other products requiring commons-beanutils through 1.9.0, does not suppress the class property, which allows remote attackers to "manipulate" the classloader and execute arbitrary code via the class parameter, as demonstrated by the passing of this parameter to the getClass method of the ActionForm object in Struts 1.					High

Duplicated components

With Kiuwan Insights you can identify different versions of the same component used by your application.

The below example shows that the analyzed application is incorporating two different versions of ZKoss common library: 8.0.1 and 6.0.0

zkoss-common		8.0.1	zk-8.0.1.jar	Java	Java	None	Info
Description							
2x normal							
License							
GNU Lesser General Public License v3.0 only							
Vulnerabilities							
CVE	CWE	Description					Severity
No records to display							
zkoss-common		6.0.0	zk-6.0.0.jar	Java	Java	None	Info
Description							
2x normal							
License							
GNU Lesser General Public License v3.0 only							
Vulnerabilities							
CVE	CWE	Description					Severity

Most probably, this component duplication is not intended, and it's something that would produce maintainability headaches when upgrading to a newer version of the library.