Lifecycle Scanning

Quick start for supported languages and where to scan. For more information, see the Comprehensive Guide to Lifecycle Scanning.

Java

Application Types:
jar, war, ear, tar.gz, zip, tgz, bz2

Scan the application files using either the Maven plugin or the CLI.

A scan will pick up all the dependencies packaged into that file unless it's an uber jar.

Maven scans provide 100% package hash matching.

JavaScript

Application Types:
js, zip, tar.gz, tgz

If using webpack to build, use `copy-modules-webpack-plugin`.

Otherwise scan in the CLI.

NuGet / .Net

Application Types:
nupkg, dll, zip, tar.gz

Best option is to use the Visual Studio Plugin.

Scanning the deployment artifact (zip file usually) can also be useful at the release stage to catch anything not directly brought in through nuget.

NOTE: scanning `.nupkg` files provides the best results.

Python

Application Types:
requirements.txt

Use only requirements using the "==" operator and version without wildcards. .

To scan in the CLI:
- Use `pipfreeze` to create the requirements file.
- Add optional environmental markers.
- Run a scan.

Docker

Application Types:
tar

Scans the application layer of your containers, and provides precise component intelligence for Java, JavaScript, Nuget, and Python.

To scan a Docker image, first save it as a `tar` file, and then run a scan in the CLI, Web UI, or as a CI build step.

C / C++

Application Types:
CMake files

Scan in the CLI using XC:
Use the `-xc, --expanded-coverage` parameter to run an XC scan.

NOTE: XC results come from unverified public sources, and do not include any Sonatype enriched information.

Android

Application Types:
jar, war, ear, tar.gz, zip, tgz, bz2

Scanning APK directly is not supported.
- Compilers convert source code into DEX (`Dalvik Executable`) file
Maven Build
- Use the maven plugin
Gradle Build
- Maven copy dependencies
Scan folder using the CLI

YUM

Application Types:
.rpm / tgz

Best option is to scan in a CI build before it is packaged into a deb.

To scan in the CLI using XC:
- Use the `-xc, --expanded-coverage` parameter to run an XC scan.

Ruby

Application Types:
gemfile.lock

Firewall and XC only, not available in Lifecycle.

Scan in the CLI using XC:
Use the `-xc, --expanded-coverage` parameter to run an XC scan.

PHP

Application Types:
composer.lock

Scan in the CLI using XC:
Use the `-xc, --expanded-coverage` parameter to run an XC scan.

Node / node.js

Application Types:
js, zip, tar.gz, tgz

To scan in the CLI:
- Scan the `node_modules` folder generated from npm install.
- Delete the `node_modules` and run `npm install --production` to exclude dev dependencies in the `package.json`.

Swift

Application Types:
Package.Swift (for Swift)
.podspec (for CocoaPods)

Scan in the CLI using XC:
Use the `-xc, --expanded-coverage` parameter to run an XC scan.

NOTE: This guide will be updated as more language support is added to the Nexus IQ Server.
Please check back often for updates.

sonatype