CI-CD Integration with Xray

Overview

You can seamlessly integrate JFrog Xray security and compliance scanning in your organization's CI/CD pipeline to make sure that build jobs containing vulnerabilities are stopped early on in the process. As part of a fully automated process, Xray receives information about a build that has just been run by your CI server, it then runs a deep recursive scan on the build down to the deepest level dependency, and if any vulnerabilities are found, Xray will return an indication to the calling CI server.

Failing a build job that includes build artifacts or dependencies with vulnerabilities is an effective way to prevent any infected builds from reaching your production systems. There are organization policies that force developers to scan every build they run and fail them immediately if infected artifacts are found. However, this mode of operation has been found to inhibit developers' creativity and stunt their productivity, and often, developers find a way around this kind of restriction. A better solution is to periodically run this kind of scan once the code of several developers has been merged. For example, during a nightly build run by an organization's CI server.

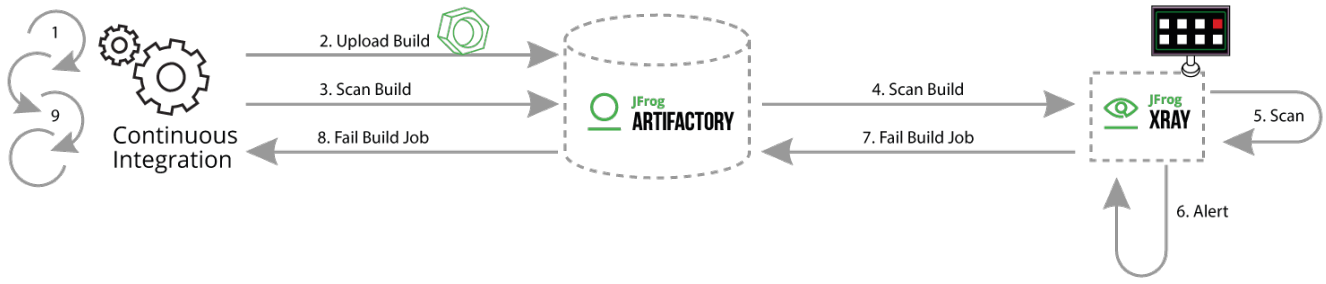
Page contents

How Does Xray Scan Your Builds?

There are three players in this process:

- **Your CI Build**
  Currently supported for Jenkins, Azure DevOps, Bamboo, TeamCity and JFrog CLI.
  The CI build sends a request to Xray, through JFrog Artifactory, for the build to be scanned. If the scan detects a vulnerability, the CI build can take appropriate action.
- **JFrog Artifactory**
  JFrog Artifactory serves as a mediator between the CI server and Xray. It does nothing more than pass information between one and the other.
- **JFrog Xray**
  Upon request, if Xray has defined watches with Actions to fail a build job, it will scan the build, and respond with a message that the build job should fail if a vulnerability is detected in the build artifact or one of its dependencies.

The following workflow shows how Xray scans your builds.



1. The CI build runs.
   Assuming the build is successful, the CI build publishes the build-info to Artifactory.
2. Xray automatically scans your new build artifacts and dependencies..
3. The CI build passes a request to Artifactory to scan the build.
4. Artifactory passes a request to scan the build through Xray's **scanBuild** REST API endpoint.
5. Xray scans the build according to a defined Watch with a **Fail Build Job** action.

> **Multiple watches or no watches?**
>
> You may define multiple Watches with a Fail Build Job Action, each with its own criteria (i.e. Artifact Filters and/or Issue Filters) that should trigger an alert. All of these Watches are applied each time a build is scanned.
>
> If Xray receives a scanBuild request, and there are **no** Watches defined with a Fail Build Job Action, Xray will always respond with an indication to fail the build job, even if no vulnerabilities are found in the build artifacts or their dependencies.

6. If a build artifact or dependency meets the conditions (filters) defined in the Watch, Xray triggers an alert and...
7. Xray responds to the **scanBuild** request indicating that the build job should fail.

> **All Alerts in one response**
>
> The response includes the details of all Alerts generated by all Watches that include a Fail Build Job Action.

8. Artifactory passes the response back to the CI Server.
9. The CI Server fails the build job.

Setting Up Xray CI Integration

Configuring Xray

1. Install Xray.
2. Set up indexing on the resources.
3. For Xray to scan builds upon request by a CI server, you need to configure a Watch with the right filters that specify which artifacts and vulnerabilities should trigger an alert, and set a Fail Build Job Action for that Watch.

Configuring your CI Server

Xray CI/CD integration is supported for Jenkins, Azure DevOps, Bamboo and JFrog CLI.

Jenkins

To **configure a build job** to request a scan, with the Jenkins Artifactory Plug-in (v2.9.0 and above), you need to create a `scanConfig` instance and and pass it to the `xrayScan` method in the Jenkins Pipeline.

Azure DevOps

To scan build artifacts for vulnerabilities in Azure DevOps, you need to add the **Artifactory Xray Scan** task after the Artifactory Publish BuildInfo task.

Bamboo

To **scan build artifacts** for vulnerabilities, with the Bamboo Artifactory Plug-in, you need to add the **Artifactory Xray Scan** task to your plan**.** The task should follow a previous task which publishes the build-info to Artifactory.

TeamCity

To **scan build artifacts and dependencies** for vulnerabilities with the TeamCity Artifactory Plug-in, you need to enable the **Xray scan on build** and **F ail build options**, configured per build.

JFrog CLI

To scan build artifacts for vulnerabilities using JFrog CLI, you need to use the **jfrog rt scan-build** command.

Configuring Artifactory

While Artifactory does not play an active part in this integration, and there is no explicit configuration needed, Artifactory does play a passive role in passing information between your CI server and JFrog Xray.

This feature is supported in **Artifactory from v4.16** and above.

Managing Scanned Builds

Xray's build integration allows you to manage your build jobs and configure them with appropriate actions if build artifacts or dependencies with vulnerabilities are found in your builds. While the default action (in Jenkins) is to simply stop the build, you can actually configure your pipeline to do other things like send email notifications or even run a different build job.

Watch the Screencast

Watch this screencast to learn how to get the best of two worlds - developer productivity and safety, by scanning the results of every build for security vulnerabilities, license compliance issues and more with JFrog Xray.