

Lab 2 (15 min)

OVERVIEW: In this lab you will use JFrog Curation, configure policies with it and test it with a maven install command.

EXPECTED OUTCOME: Upon successful completion of this lab you will gain knowledge of how to curate open-source dependencies with the JFrog Platform.

Step by step instructions

Note :

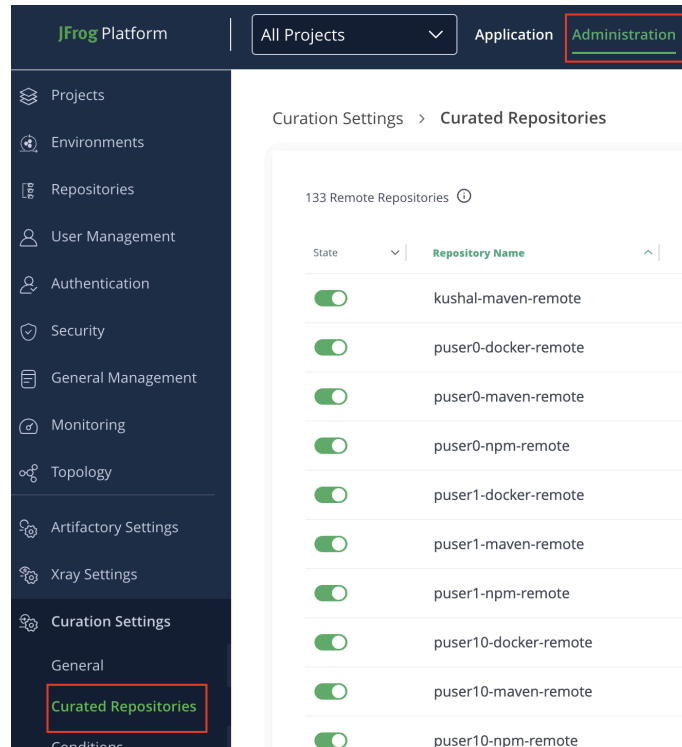
As phase #1 requires admin rights, this has already been done in the training lab environment.

This will be demonstrated by your lab instructor.

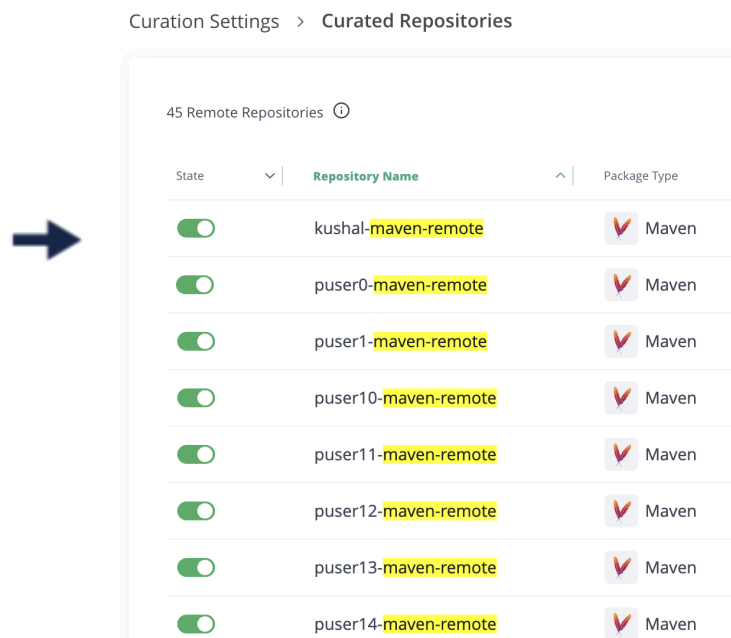
As part of the training, we recommend you read the instructions below, as it will teach you the exact process and steps required to control your 3rd parties dependencies as a JFrog platform's administrator.

Phase #1 - Selecting a Repository to curate

1. Click on 'Administration' at the top pane and then Curation Settings -> Curated Repositories on the left pane.



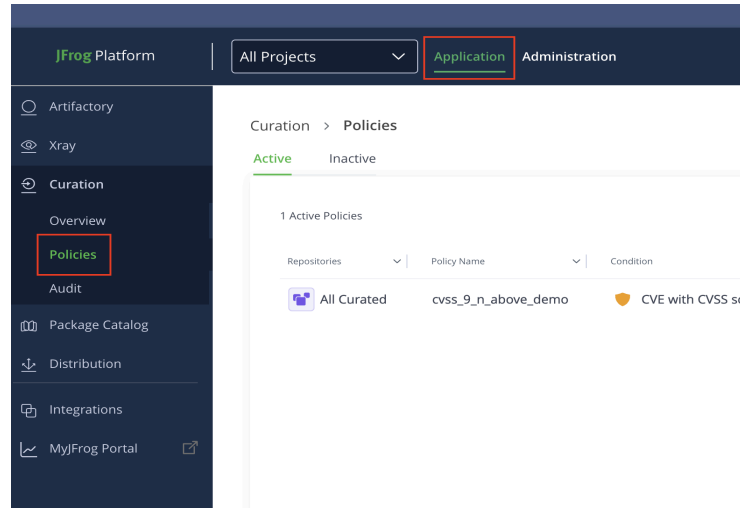
2. Make sure Curation is Turned On for your respective 'puserX-maven-remote' repository. This will enable you to enforce curation policies on this repository.



Phase #2 - Creating a Curation policy

If it is not already done, unselect your Jfrog project “PuserX” and choose “All projects” (see the top nav bar in the screen below).

3. Click on ‘Application’ at the top pane and then on ‘Curation’ -> ‘Policies’ from the left pane.



4. Click on ‘Create Policy’
5. Choose a name (we suggest **DemoUserX**, to avoid name conflicts error on save) for the policy and click ‘Next’

Curation > Policies > **New Curation Policy**

1

Policy Name

What is the name of the policy?

Name

Next >

2

Repositories

3

Policy Condition

6. Choose your specific 'puserX-maven-remote" repository. This means the policy will be enforced only on this repository. Click 'Next'.

Curation > Policies > New Curation Policy

✓

2

3

Policy Name

demo

Repositories

Specify the remote repositories for the policy.

All Curated ⓘ

Specific ⓘ

+

Select Remote Repository









Select a repository in order to continue

Next >

Policy Condition

Select A Remote Repository

45 Remote repositories

Repository Name	Package Type
<input type="checkbox"/> kushal-maven-remote	 Maven
<input type="checkbox"/> puser0-maven-remote	 Maven
<input type="checkbox"/> puser1-maven-remote	 Maven
<input type="checkbox"/> puser10-maven-remote	 Maven
<input type="checkbox"/> puser11-maven-remote	 Maven
<input type="checkbox"/> puser12-maven-remote	 Maven
<input type="checkbox"/> puser13-maven-remote	 Maven
<input type="checkbox"/> ...	 ..

7. Now it's time to choose a condition for the policy. Take a look at the different options, some are security related, some are license related and some are operational.
- Curation enables you to pick the right OSS dependency based on different types of criterias. For this lab, let's choose 'CVE with CVSS score 9 or above (with or without a fix version available)' as the condition. Click 'Next'.

3

Policy Condition

Select the policy condition that will indicate a violation

Conditions (52)

☐ Malicious package

☐ CVE with CVSS score of 9 or above (fix version available)

☒ CVE with CVSS score of 9 or above (with or without a fix version available)

☐ CVE with CVSS score between 7.0 and 8.9 (fix version available)

☐ CVE with CVSS score between 7.0 and 8.9 (with or without a fix version available)

Next >

CVE with CVSS score of 9 or above (with or without a fix version available)

Risk TypeSecurity

Condition TypeJFrog

From TemplatePackage has vulnerability with CVSS score in range {range}

Configured ParametersCVSS score range9-10

Fix version dependence

New vs. Existing risk

Supported Typenpm, Maven, Gradle, SBT, CocoaPods, Composer, NuGet, Docker, Helm, Kubernetes, Terraform, Ansible, Puppet, Chef, SaltStack, Jenkins, GitLab, Travis CI, CircleCI, GitHub Actions, AWS, Azure, Google Cloud, IBM Cloud, Oracle Cloud, SAP, Salesforce, Microsoft Dynamics, SAP SuccessFactors, SAP S/4HANA, SAP Ariba, SAP Fieldglass, SAP Concur, SAP Hybris, SAP Commerce, SAP Customer Data Cloud, SAP Marketing Cloud, SAP Sales Cloud, SAP Service Cloud, SAP SuccessFactors, SAP S/4HANA, SAP Ariba, SAP Fieldglass, SAP Concur, SAP Hybris, SAP Commerce, SAP Customer Data Cloud, SAP Marketing Cloud, SAP Sales Cloud, SAP Service Cloud

DescriptionBlocks 3rd party package versions with a known vulnerability whose NVD CVSS score is 9 or above, regardless of whether a newer version that fixes the vulnerability is available. This policy should be used in cases where you want to block a package due to its vulnerability score based on any CVSS

8. Here we can add a waiver that will exclude specific packages from the policy being created. A waiver can be added also after policy creation. Let's skip and click on 'Next'.

9. Currently, Curation has two different actions: 'Block', which will block the download request and return a proper error message, or 'Dry Run', which will only simulate the curation flow. Let's choose 'Block' and click on 'Save Policy' on the bottom right.

5

Actions & Notifications

Select the required action if a violation occurs

☒ Block

☐ Dry Run

☐ Notify by Email

Next

Policy ConditionMalicious package

SupportedAll

DescriptionDetects 3rd party packages that have been identified by the JFrog Security Research team as malicious. The JFrog Security Research group created scanners that continuously scan 3rd party packages for indications of malicious intent. Our detectors look for indications of infection methods (e.g. typosquatting, dependency confusion) suspicious payload actions (e.g. download and execute, dynamic code evaluation), obfuscation techniques and more.

Policy EffectivenessCovered Repositories List

Save Policy

Congratulations, you've created your first curation policy!

Phase #3 - Testing your Curation policy

10. Open a new browser Tab and paste this URL for your respective 'puserX-maven-remote' repository.

Note : don't click on this link, replace puserX with your matching user number before !!

<https://dsodsec.jfrog.io/artifactory/puserX-maven-remote/log4j/log4j/1.2.17/log4j-1.2.17.jar>

11. You should see a 403 Forbidden with a message explaining the Blocked download.

```
swampup17242481111.jfrog.io/ui/native/puser0-maven-remote/log4j/log4j/1.2.17/log4j-1.2.17.jar
{
  "errors": [
    {
      "status": 403,
      "message": "package log4j:log4j:1.2.17 download was blocked by jfrog packages curation service due to the following policies violated {cvss_9_n_above_demo,
    }
  ]
}
```

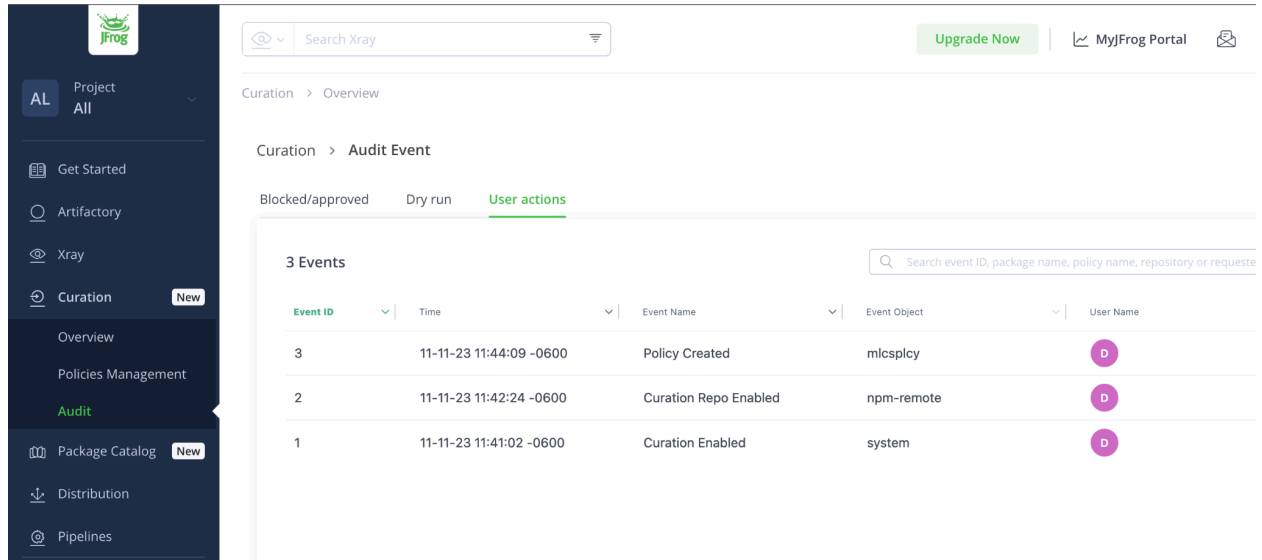
Phase #4 - Inspecting the Curation event

12. Go back to the JPD UI, and click on 'Curation' -> 'Audit'. Here you should be able to see the last event, which is a rejection of your maven download request.

The screenshot shows the JFrog Platform UI. On the left is a sidebar with navigation links: JFrog Platform, All Projects, Application, Administration, Artifactory, Xray, Curation, Overview, Policies, Audit (highlighted with a red box), Package Catalog, Distribution, Integrations, and MyJFrog Portal. The main content area is titled 'Curation > Audit Event' and has tabs for 'Blocked/approved', 'Dry run', and 'User actions'. Below the tabs, it shows '1 Events' with a table listing the event. The event details on the right include: Event ID | 9, 09-09-24 09:27:14 -0500, Action: blocked, Reason: Policy violations, Package Name: log4j:log4j, Package Type: Maven, Package Version: 1.2.17, Package URL: https://repo1.maven.org/maven2/log4j/log4j/1.2.17/log4j-1.2.17.jar, Origin Server: b0gokuy6jjcem, Origin Repository: puser0-maven-remote, Requester: pstrainenv, and Violated Policies (1): cvss_9_n_above_demo. The policy details show a condition of 'CVE with CVSS score of 9 or above (with or without a fix version available)' and an explanation that the package version contains the following vulnerability(s): CVE-2019-17571 (Severity 9.8) and CVE-2022-23305 (Severity 9.8). Recommendations state: 'We could not find a version that resolves all the violating vulnerabilities. Ple consult your Curation admin.'

13. In addition, if you click on 'User Actions', you should be able to see the audit trail of the policy you created in Phase #2.

Inspect the information provided on this action.



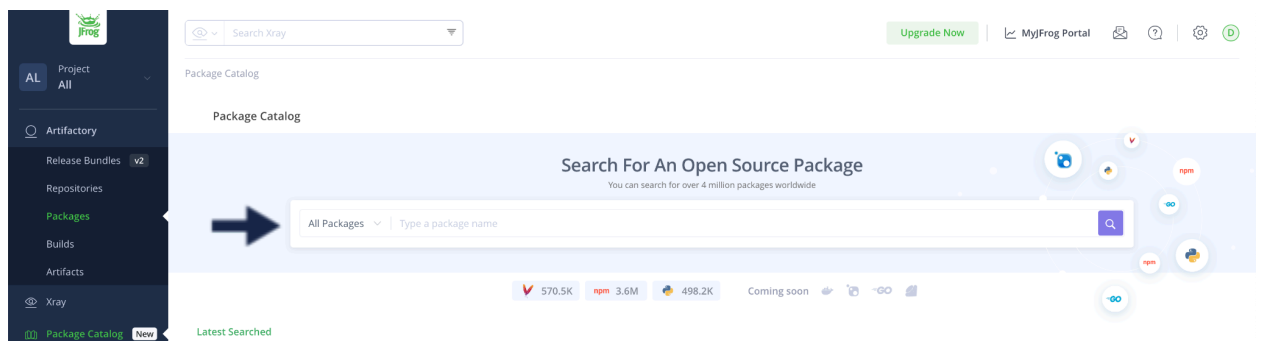
The screenshot shows the JFrog Xray interface. The left sidebar contains navigation links: Project All, Get Started, Artifactory, Xray, Curation (highlighted with a 'New' badge), Overview, Policies Management, Audit (highlighted), Package Catalog (highlighted with a 'New' badge), Distribution, and Pipelines. The main content area is titled 'Curation > Overview' and 'Curation > Audit Event'. It features three tabs: 'Blocked/approved', 'Dry run', and 'User actions' (selected). Below the tabs is a table titled '3 Events' with a search bar. The table has columns for Event ID, Time, Event Name, Event Object, and User Name. The events listed are:

Event ID	Time	Event Name	Event Object	User Name
3	11-11-23 11:44:09 -0600	Policy Created	mlcsplcy	D
2	11-11-23 11:42:24 -0600	Curation Repo Enabled	npm-remote	D
1	11-11-23 11:41:02 -0600	Curation Enabled	system	D

Congratulations! You have completed Lab 2

Phase #5 - BONUS - Inspecting the package in JFrog Catalog

14. Go back to the UI, and click on 'Package Catalog'. Search for the 'cors.js' package.



The screenshot shows the JFrog Package Catalog interface. The left sidebar contains navigation links: Project All, Artifactory, Release Bundles (v2), Repositories, Packages (highlighted), Builds, Artifacts, Xray, and Package Catalog (highlighted with a 'New' badge). The main content area is titled 'Package Catalog' and features a search bar with the text 'Search For An Open Source Package' and 'You can search for over 4 million packages worldwide'. A blue arrow points to the search bar. Below the search bar, there are statistics: 570.5K, npm 3.6M, 498.2K, and 'Coming soon'.

15. What kind of information does the catalog provide on cors.js? What is the core issue and what is the remediation?