

Lab 2 (15 min)

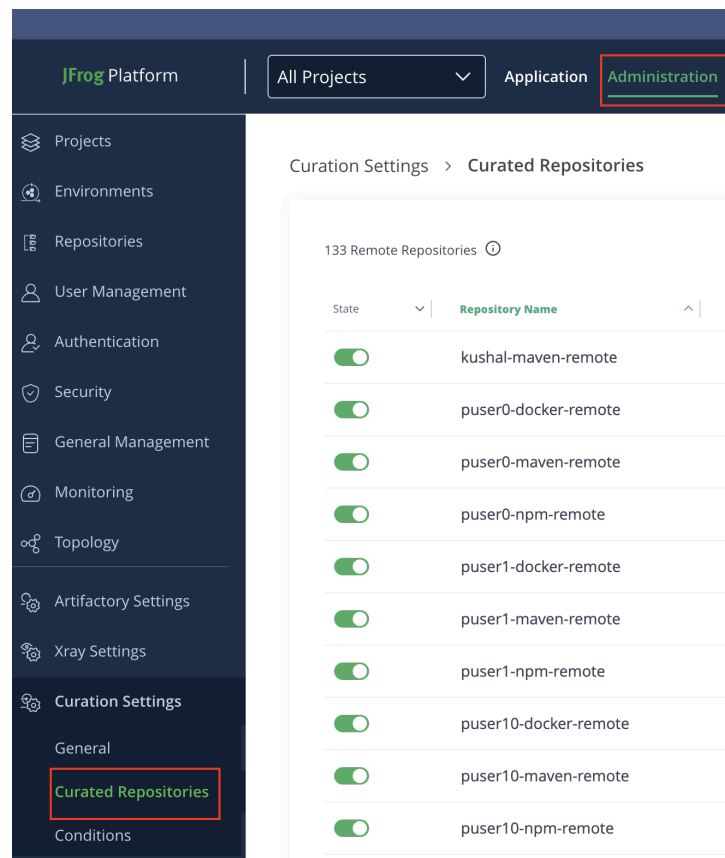
OVERVIEW: In this lab you will use JFrog Curation, configure policies with it and test it with a maven install command.

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

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.

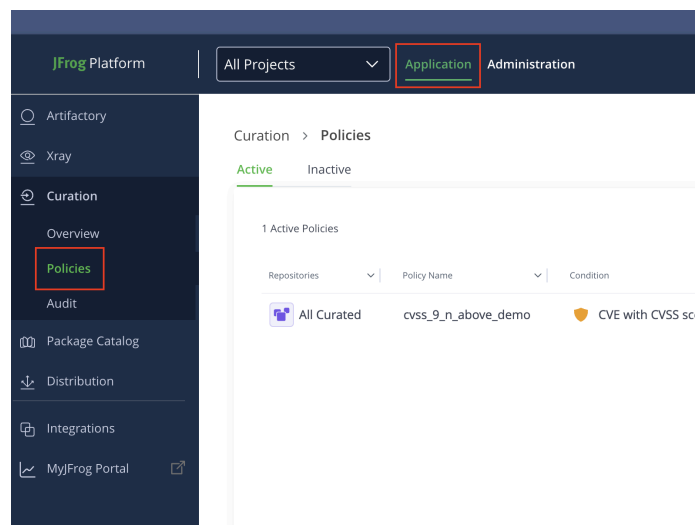
Curation Settings > Curated Repositories

45 Remote Repositories ⓘ

State	Repository Name	Package Type
	kushal-maven-remote	Maven
	puser0-maven-remote	Maven
	puser1-maven-remote	Maven
	puser10-maven-remote	Maven
	puser11-maven-remote	Maven
	puser12-maven-remote	Maven
	puser13-maven-remote	Maven
	puser14-maven-remote	Maven

Phase #2 - Creating a Curation policy

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 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'.

✓

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"/> ...	 ..

- 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.

Policy Condition

Select the policy condition that will indicate a violation

Search condition

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 Type

Condition Type

From Template

Configured Parameters

Security

JFrog

Package has vulnerability with CVSS score in range {range}

CVSS score range

9-10

Fix version dependence

Not dependent

Worse

Supported Type

Description

npm

Py

Java

Go

Docker

Red Hat

Ansible

Blocks 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

- 5

Actions & Notifications

Select the required action if a violation occurs

☒ Block

☐ Dry Run

☐ Notify by Email

Add Email

Next

Policy Condition

Malicious package

Supported

All

Description

Detects 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 Effectiveness

[Covered Repositories List](#)

Cancel

Save Policy

Phase #3 - Testing your Curation policy

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

<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. The left sidebar contains the navigation menu with 'Audit' highlighted. The main content area shows the 'Curation > Audit Event' page. The 'Blocked/approved' tab is selected, showing a table with 1 event. The event details on the right show the package name 'log4j:log4j', version '1.2.17', and the violated policy 'cvss_9_n_above_demo'. The event is marked as 'blocked'.

Time	Action	Reason
09-09-24 09:27:14 -0500	blocked	Policy violation

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

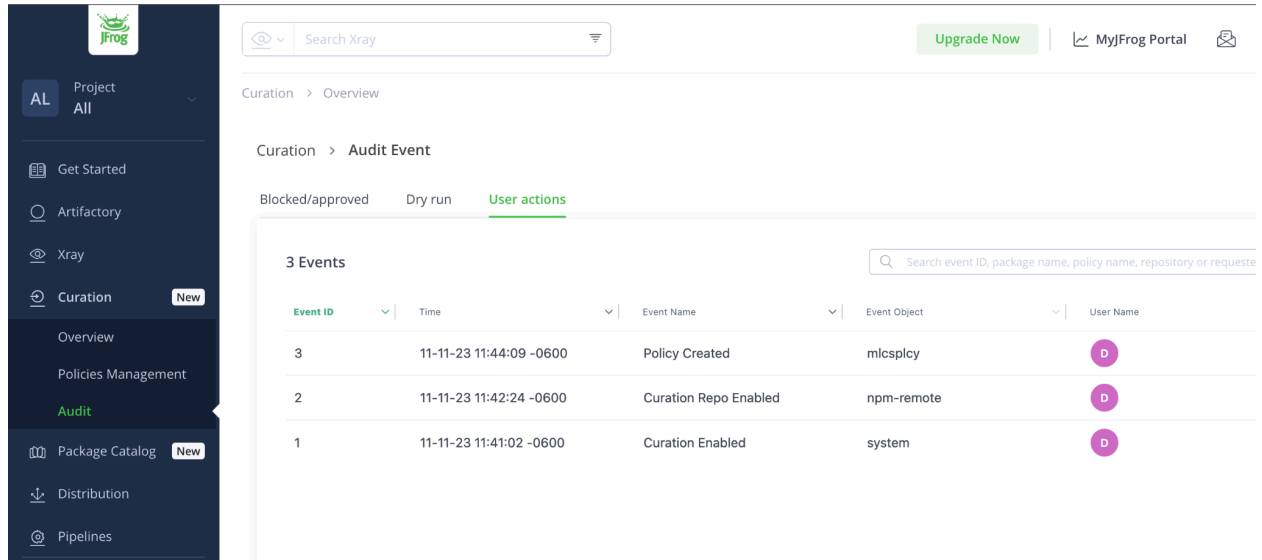
Violated Policies (1)

Condition	Severity
CVE with CVSS score of 9 or above (with or without a fix version available)	9.8
Package version contains the following vulnerability(s):	9.8
CVE-2019-17571	No fixed version were found
CVE-2022-23305	No fixed version were found

Recommendations: We could not find a version that resolves all the violating vulnerabilities. Please 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.



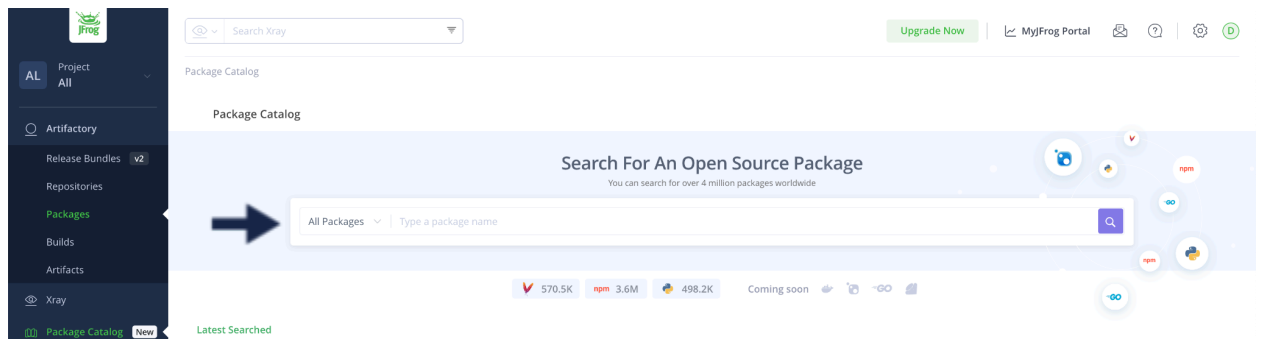
The screenshot shows the JFrog Xray interface. The left sidebar has a dark blue background with the JFrog logo at the top. The main content area is white. At the top, there's a search bar labeled 'Search Xray' and a green 'Upgrade Now' button. Below the search bar, the breadcrumb 'Curation > Overview' is visible. The main heading is 'Curation > Audit Event'. There are three tabs: 'Blocked/approved', 'Dry run', and 'User actions' (which is highlighted in green). Below the tabs, there's a table titled '3 Events'. The table has columns: '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 has a dark blue background with the JFrog logo at the top. The main content area is white. At the top, there's a search bar labeled 'Search Xray' and a green 'Upgrade Now' button. Below the search bar, the breadcrumb 'Package Catalog' is visible. The main heading is 'Package Catalog'. There's a large blue banner with the text 'Search For An Open Source Package' and 'You can search for over 4 million packages worldwide'. Below the banner, there's a search bar with a dropdown menu set to 'All Packages' and a text input field labeled 'Type a package name'. A blue arrow points to the search bar. At the bottom, there's a row of statistics: '570.5K', 'npm 3.6M', and '498.2K'. To the right of these statistics, there's a 'Coming soon' label and a 'GO' button.

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