JFROG ADVANCED SECURITY

GUIDED TRIAL IN 15 MINUTES

INTRODUCTION

The JFrog Advanced Security trial provides a fully-functional version of the JFrog Platform (including Artifactory and Distribution). The trial is pre-populated with artifacts (containers) so you can immediately go through security scan results and discover the value of the output, as well as the flow and the experience.

If you want to scan your own artifacts, we recommend creation of new local/remote repositories in Artifactory, and pull/push artifacts (such as Docker containers) for security scanning and observation or examination of the results. For creating new repositories and to upload artifacts, you can access the platform UI, however, it is highly recommended to use the following Guided Trial utility to shorten the process.

The JFrog Guided Trial utility simplifies the onboarding process and will help you quickly:

- Setup a JFrog platform trial instance
- Configure 'local' and 'remote' docker repositories with advanced security scanning
- Assist you with loading Docker images from your local workstation or from Dockerhub (including recommended samples)

```
Welcome to JFrog trial setup
============================
1. Launch and configure a new trial
2. Docker login to existing trial from a new workstation
3. Pull Docker image or select sample docker image
4. Push Docker image from local machine to scan with JAS
5. Exit

Please select an option:
```

HOW TO USE THE GUIDED TRIAL UTILITY

1. Make sure that you have Docker Desktop client and Curl installed.

1. Download Docker Desktop from ***here*** and run the installation file. 2. Download Curl from from the ***official website*** and follow the instructions in the download page.	1. Download Docker Desktop from ***here*** and run the installation file. 2. Type "sudo apt-get install curl" to install the curl package.

2. Download and run the guided trial script for Windows, Linux and Mac.

1. Download the batch script - ***link*** 2. Open cmd.exe 3. Navigate to script path 4. Run command: windows_guided_trial.bat	1. Download the batch script - ***link*** 2. Open terminal 3. Navigate to script path 4. Run command: chmod +x linux_guided_trail.sh 5. Run: ./linux_guided.trial.sh

STEP #1 - SET UP AND CONFIGURE A JFROG ADVANCED SECURITY TRIAL INSTANCE

The script will open the browser so you can launch the instance. Once the instance is up, go back to the terminal and paste in the instance name and credentials.
You would be required to provide the following:

1. The hostname

2. The email address

3. The password or an API token (in case of SSO)

Follow the script, which will open the browser and guide you through the proccess.

Once credentials are being provided, the script will:

- Create a new remote repository, 'docker-hub-remote-repo' for DockerHub, with advanced security scanning.
- Create a new local repository 'local-docker-repo' for DockerHub, with advanced security scanning.
- Configure an Xray Policy. The policy rules are:
  - Create violation for CVEs with Critical CVSS Score
  - Create violation for Exposures with High impact severity
  - Create violation and block download of malicious packages
- Configure an Xray Watch on all repositories.
- Execute "docker login" so you can work with the trial instance.

STEP #2 - DOCKER LOGIN TO EXISTING TRIAL FROM A NEW WORKSTATION

Run this step only on a new workstation.

The script will request the instance name as credentials (as asked in step #1) to configure the Docker client to work with an already-existing trial instance.

STEP #3 - UPLOAD DOCKER TO YOUR NEWLY-CREATED REPOSITORY

Option A - pull your proprietary Docker image or select the sample Docker image

The script will assist you with pulling images from DockerHub to ‘docker-hub-remote-repo’ repository. Option #1 and #2 offer uploading samples of known vulnerable images - WebGoat and Netdata. Option #3 allows uploading your own Docker image.

```
Pull Docker image or select sample docker image:
================================================
1. Pull OWASP WebGoat - Good example of Contextual Analysis value
2. Pull netdata:1.13.0 - Good example of a Docker with 100M+ downloads with an embedded secret
3. Pull custom image from DockerHub via Artifactory to scan with JAS

Please select an option:
```

When the pull process has completed, you will be automatically redirected to the Scans List screen in the platform (in your browser) to review the results. See Appendix A to learn about the Scans List, the security navigation bar and types of findings you can review. If you are working with sample images (WebGoat or Netdata), please see Appendix B.

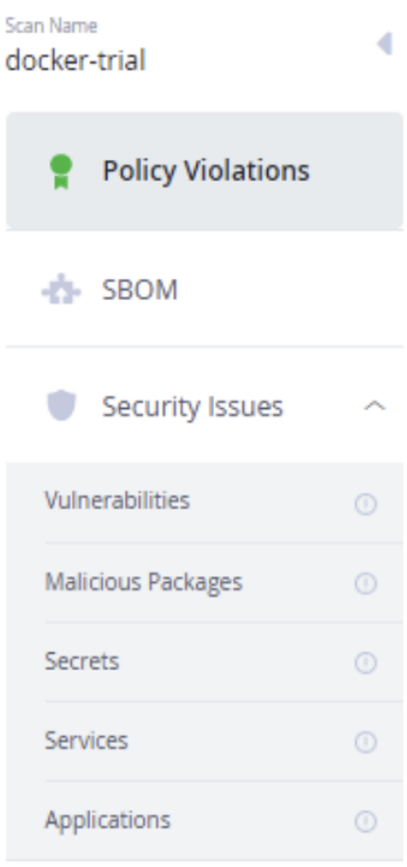
Option B - Push Docker image from local machine to scan with JFrog

The script will assist you with pushing images from your local machine to the ‘local-docker-repo’ repository.

```
Push Docker image from local machine to scan with JAS:
======================================================
Listing available docker images on local machine:

REPOSITORY          TAG       IMAGE ID       CREATED       SIZE
nginx               latest    080ed0ed8312   2 weeks ago   142MB
mysql               latest    4073e6a6f542   5 weeks ago   530MB
circleci/slim-base  latest    be1e44c35321   6 years ago   30.6MB

Enter Docker image name and then its tag:
=========================================
Enter the Docker image name as shown in the REPOSITORY column: circleci/slim-base
Enter the Docker tag: latest

The push refers to repository [[redacted].jfrog.io/local-docker-repo/circleci/slim-base]
```

APPENDIX A - SCAN RESULTS STRUCTURE

The JFrog scan reports are available under the “Security Issues” screen and contain comprehensive information about the different scan results.



The Report contains the following sections:

Policy violations:

- An aggregation of security issues that cause a violation of the policy. In step 1 we have configured the following policy:
  - Trigger a violation upon Critical CVSS Score
  - Trigger a violation upon High impact exposure (of Secrets, Application, Services)
  - Trigger a violation upon detection of a malicious package

SBOM

- List of all detected open source software components that were detected, including the option to export the results to SPDX and CyclonDX format, including support of the new VEX attributes to testify on exploitability of analyzed CVEs.

Security issues → Vulnerabilities:

- Listing of all the CVEs that were found. For Docker containers, you can see the results of “Contextual Analysis” that allows you to prioritize whether the open source software vulnerabilities are actually exploitable (applicable or not) in your application.

Security issues → Malicious Packages:

- Discover and eliminate unwanted or unexpected packages, using JFrog’s unique database of identified malicious packages.

Security issues → Secrets:

- Detect secrets left exposed in any containers stored in JFrog Artifactory to prevent any accidental leak of passwords, API keys, internal tokens, or credentials.

Security issues → Services:

- Discover whether common services (such as Nginx, envoy, Prometheus, Apache, and more) are configured insecurely, causing exposure to attacks.

Security issues → Applications:

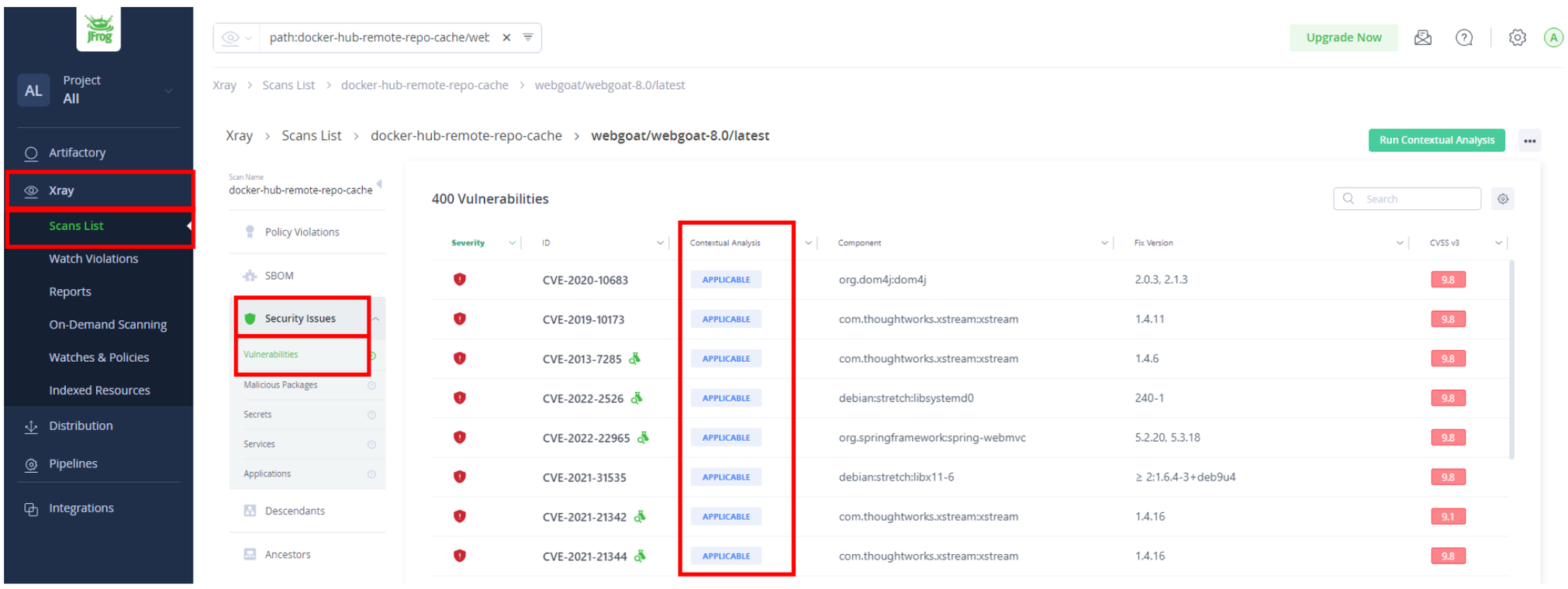
- Discover whether your developers are using OSS libraries insecurely, causing exposure to attacks.

APPENDIX B - WEBGOAT AND NETDATA/1.13.0 EXAMPLES

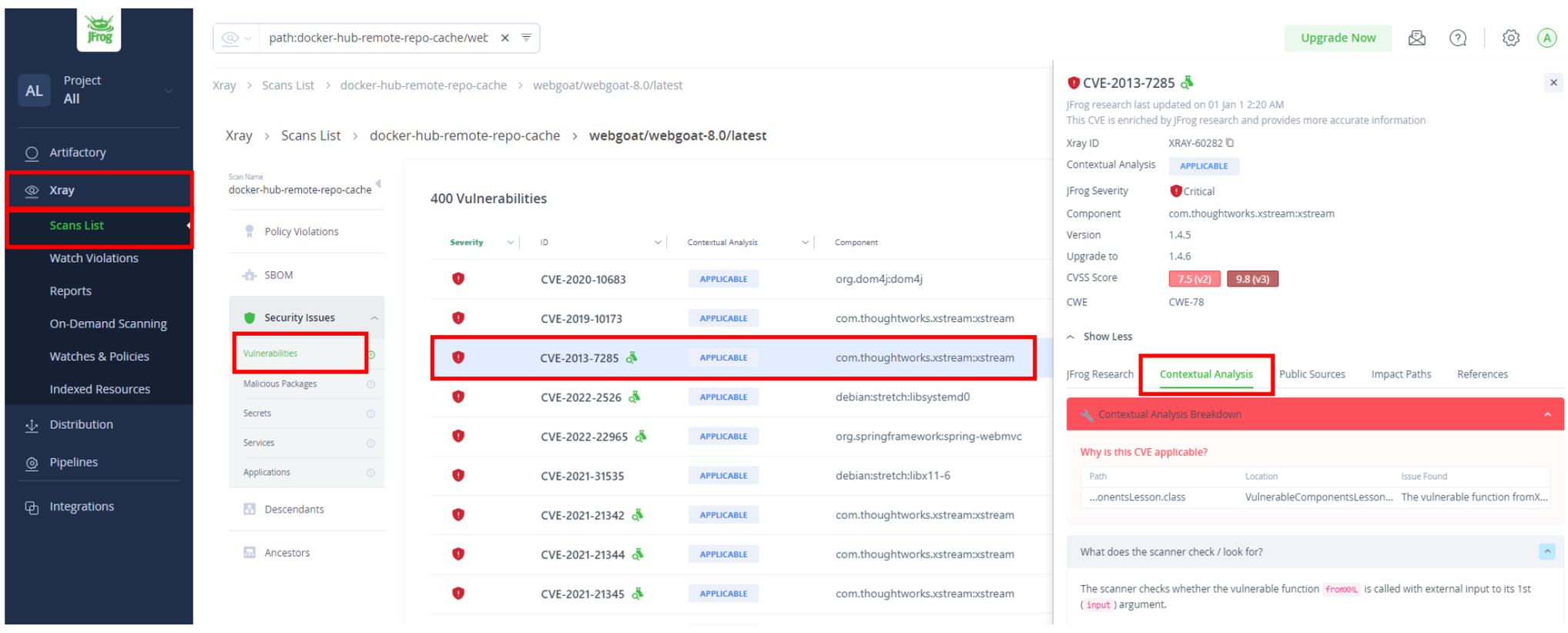
Example #1 | OWASP Webgoat

OWASP WebGoat is a deliberately insecure application that has vulnerabilities commonly found in Java-based applications that use common and popular open-source components.

This image contains many known vulnerabilities, so it is very useful for testing JFrog's Contextual Analysis capabilities, and to enjoy its value. For each identified vulnerability, the system will indicate if it's applicable or not in the specific context of the entire artifact (e.g., Docker container). When clicking on applicable CVE, or none applicable one, you will be provided with a proprietary CVE article, explaining and demonstrating how the system determined its applicability. This will significantly reduce the noise that arises with non-applicable CVEs, and focus you only on applicable CVEs, thus allowing your developers to save precious time while triaging vulnerabilities.



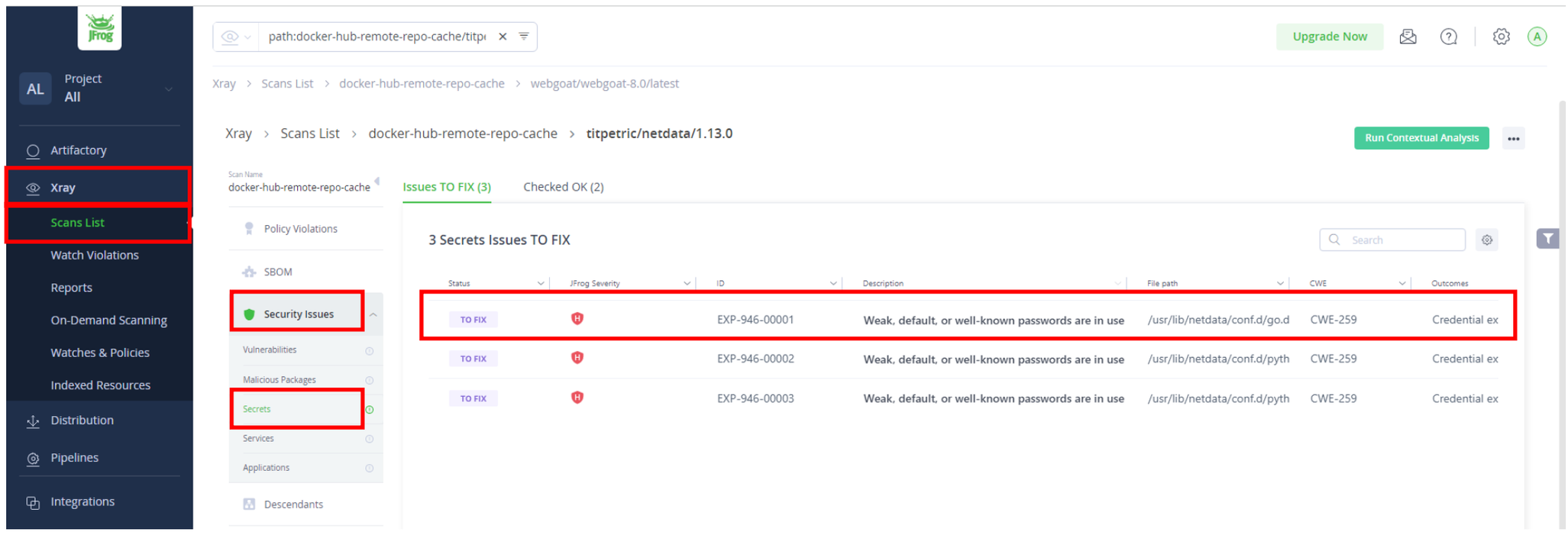
Under "Xray" > "Scans List" > "Security issues" > "Vulnerabilities" you will find the complete list of identified CVEs. Please choose "Applicable" or "Not Applicable" to expand it and review the additional information in the right pane. See the evidence that proves the CVE applicability and the following remediation steps to understand how the issue could be mitigated.



Example #2 | Netdata/1.13.0

Netdata provides distributed, real-time, performance and health monitoring for systems and applications. It is a highly-optimized monitoring agent you install on all your systems and containers. The Netdata Docker image has more than 100M+ downloads!

In the case of this specific version, the JFrog Secret Detection engine that scans for keys, credentials and token, revealed default weak password. When choosing one of the findings, you will be provided with information to point you accurately where the secret was found. It will also provide information about the risk, its implication, and remediation paths.



Under “Xray” > “Scans List” > “Security Issues“ > “Secrets” you will find the complete list of revealed secrets. Please choose one of the findings to expand it and review the additional information in the right pane. See the evidence that proves the secret is exposed and the following remediation steps to understand how the issue could be mitigated.