JFrog Advanced Security Export

Table of Contents

# Set Up JFrog Advanced Security

Enable Advanced Scans in your environment to start scanning your artifacts.

| # | Task | Description | For more information, see... |
| --- | --- | --- | --- |
| 1 | **Enable Advanced Scans** | Configure Indexing in order to enable advanced scans used in Advanced Security. | [Enable Advanced Scans](#UUID2258b6334d22d0eb8c2e4e3bea1c42b1) |
| 2 | **Create a Security Policy with Advanced Scan Rules** | Configure rules and define security criteria to be used in a Security Policy. | [Create a Security Policy with Advanced Scans Rules](#UUID018f8c6d347030cfa42178814c54a2c8) |
| 3 | **Use Advanced Scans APIs** | Use Advanced Security Rest APIs. | [JFrog Advanced Security REST API SUPPORT](#UUID582dce6f718206910bab4052172f5d68) |

## Enable Advanced Scans

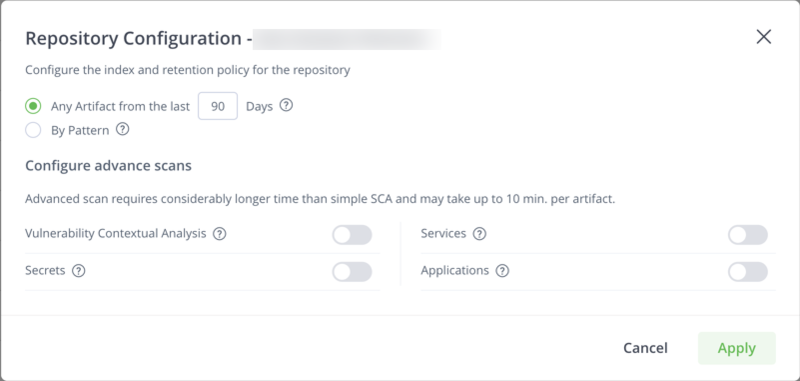
Advanced Scans are configured per repository. The repository must be indexed by Xray, for more information, see [Indexing Resources](https://www.jfrog.com/confluence/display/JFROG/Indexing+Xray+Resources). Advanced Scans are applied on newly scanned artifacts only, and not on existing indexed artifacts. You can also run contextual analysis and exposures scans on an existing artifact. To learn more about it, see [Exposures Scans](https://www.jfrog.com/confluence/display/JFROG/Exposures+Scanning+Categories) and [Contextual Analysis](https://www.jfrog.com/confluence/display/JFROG/Vulnerability+Contextual+Analysis).

If you would like to enable the scanning categories, do the following:

1. In Classic Navigation, navigate to the **Administration** module, go to **Xray** **Settings | General** and click **Indexed Resources**.

* In New Navigation, navigate to the **Administration** module, go to **Xray** **Settings** and click **Indexed Resources**.

1. Select the repository or build and select **Configure**.
2. Select the categories you want to enable:



## Create a Security Policy with Advanced Scans Rules

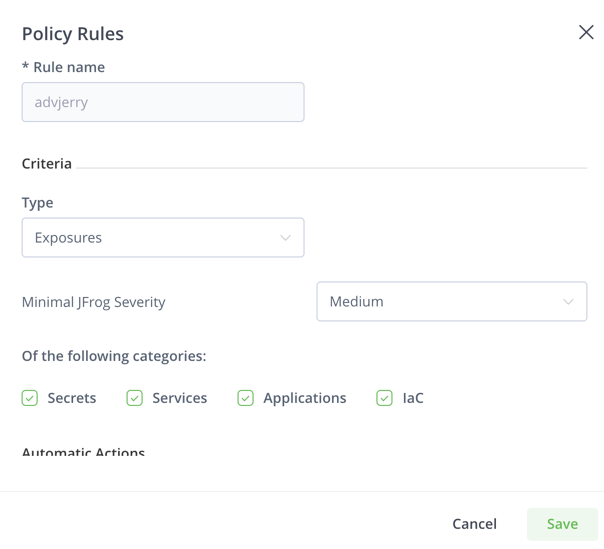
Policies enable you to create a set of rules, in which each rule defines security criteria, with a corresponding set of automatic actions according to your needs. Policies are enforced when applying them to </document/preview/361196#UUID-8827a9ab-1073-844c-095f-17c69daa11e0>. To learn more on how to create a Policy, see </document/preview/625848#UUID-05f92e64-4f1a-a5e3-072b-4e8e108efbd8>.

We recommend creating Policies to get a focused list of violations based on your security criteria.

You can create a security policy with exposures and contextual analysis-specific rules. A violation is issued when the criteria you set are met. You can view issued violations either from [Scans List](/document/preview/551896#UUID-82a35411-384b-423c-8aba-0b6796583862)or [Watch Violations](/document/preview/361938#UUID-d1b510b5-16a4-37c7-6da5-e6e649aecb90)pages.

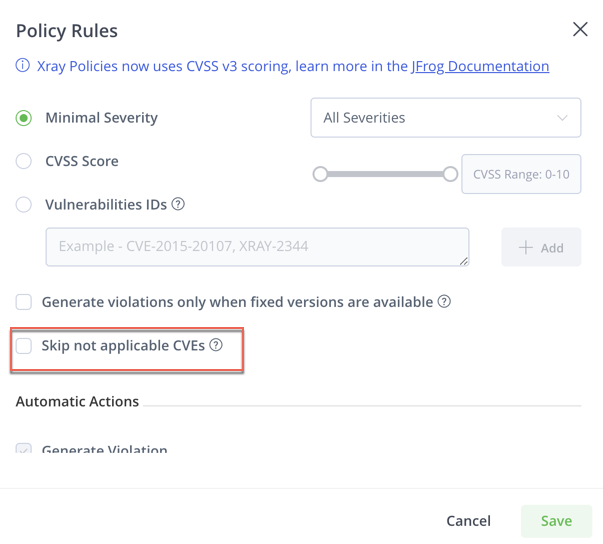
### Create an Exposures Policy

1. In the **Administration** module, under **Xray**, select **Watches** **& Policies** and from the **Policies** tab click **New Policy**.
2. Select the policy rule type **Security**.
3. Click **New Rule**, and from the **Type** drop-down, select **Exposures**.
4. Select the **Minimal JFrog Severity**. The severity given by the JFrog Security Research team after the manual analysis of the CVE by the team. This specifies that a violation is issued only if the selected severity is met.
5. Select one or more exposure categories. This specifies that a violation is issued only for the selected categories.
6. Define the automatic actions that determine the automatic response to a detected Policy violation. For more information, see </document/preview/625848#UUID-05f92e64-4f1a-a5e3-072b-4e8e108efbd8>.

* 

**Create a Contextual Analysis Policy**

1. In the **Administration** module, under **Xray**, select **Watches & Policies** and from the **Policies** tab click **New Policy**.
2. Select the policy rule type **Security**.
3. Check the **Skip not applicable CVEs** checkbox. By selecting this option, the Policy will not issue any violations on CVEs that were found not applicable by the Contextual Analysis scanners.

* 

## JFrog Advanced Security REST API SUPPORT

The following REST APIs support JAS in Policies:

* [[Create Policy](/document/preview/514330#UUID-0f4985c8-e364-8d9a-89dd-74ad321ed981)](/document/preview/514290#UUID-12604d28-75fe-7e72-ad07-505c6a804a67)
* [Get Policy](urn:resource:component:514345)

# Repository/Artifact On-Demand Advanced Scan

Repositories can be configured automatically to scan for JAS, as described in [Set Up JFrog Advanced Security](#UUID05d3d0a4f8457d1ade9b24403f088473). You can also run Contextual Analysis and scan for Exposure per repository and artifact on-demand.

Starting from Xray version 3.66.x and above, you can run Contextual Analysis or scan for Exposures on an existing artifact.

Starting from Xray version 3.73.x and above, you can run Contextual Analysis and scan for Exposures per repository when needed.

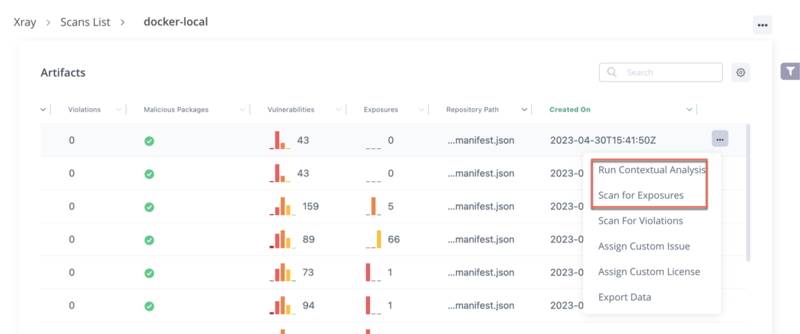
The following topics review Repository / Artifact On-Demand Advanced Scanning:

* [Advanced Scans on Existing Artifact](#UUIDcd221cfc4a223a8c06cdbd698cb4eb51)
* [Advanced Scans per Repository](#UUID48067787fdf9c8e3222d0491a32a7289)
* [Repository / Artifact Advanced Scan REST API Support](#UUIDcfd5d0b17f901847e1eb8d05c4245324)

## Advanced Scans on Existing Artifact

Do the following:

1. From the **Scans List** page, **Repositories** tab, select the repository.
2. Navigate to the artifact you want.
3. Click the **Actions Menu**next to the artifact, and select **Run Contextual Analysis** or **Scan for Exposures**.

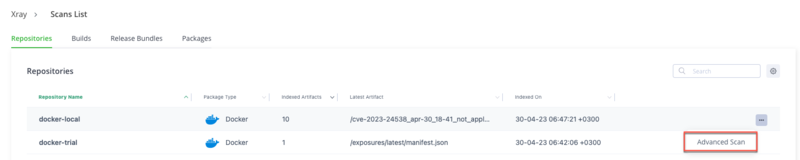
* 

The results appear under Security Issues.

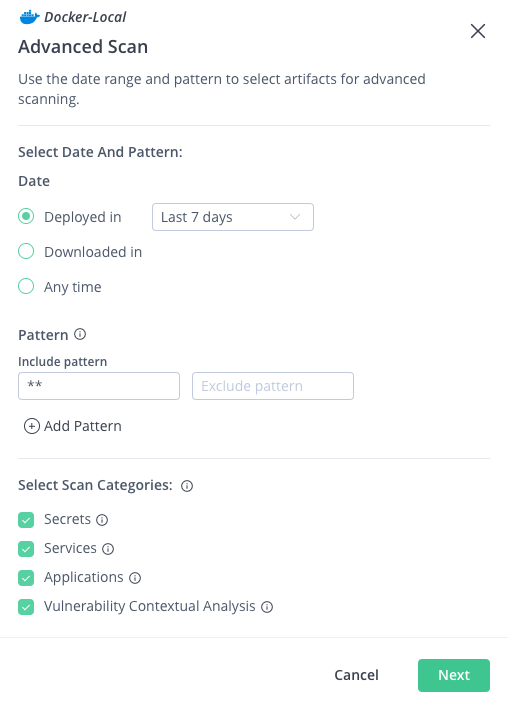


## Advanced Scans per Repository

1. From the **Scans List** page, **Repositories** tab, navigate to the repository you want to scan.
2. Click the **Actions Menu**next to the repository and select **Advanced Scan**.

* 

1. In the **Advanced Scan**dialog box, define the criteria. The criteria specifies which artifacts that are contained in the selected repository are to be scanned. You can select the following:
   1. **Deployed in**: Defines a scope of artifacts that were deployed in Artifactory within a specific time range.
   2. **Downloaded in**: Defines a scope of artifacts that were downloaded from Artifactory within a specific time range.
   3. **Any time**: Any artifacts that exist within the repository.
   4. **Pattern**: Defines a path within the repository. A pattern is required in addition to the date range.
   5. **Categories**: Defines which JAS categories to scan for issues.
2. Click**Next** and follow the Wizard instructions.

* 

## Repository / Artifact Advanced Scan REST API Support

* </document/preview/586629#UUID-89c102f9-5e9b-ac33-f0c6-4c6d91909666>
* </document/preview/586828#UUID-a7e9acd3-cda9-baa7-d2cd-ffe120dc7b78>
* </document/preview/586901#UUID-319eb7e1-2ba3-7958-14e1-b1cf21b001c6>
* </document/preview/551885#UUID-1894543d-8403-a343-52ae-3349b744d86d>
* </document/preview/551624#UUID-ded1b0b1-80db-33f8-141c-5e0644c19627>

# JFrog Advanced Security Scan Results and Post-Scan Actions

The JFrog Advanced Security (JAS) scan results can be viewed from the [Scans List](https://www.jfrog.com/confluence/display/JFROG/Xray+Scans+List) page. See the relevant sections in [Exposures Scans](https://www.jfrog.com/confluence/display/JFROG/Exposures+Scanning+Categories) and [Contextual Analysis](https://www.jfrog.com/confluence/display/JFROG/Vulnerability+Contextual+Analysis) to learn more about results and actions. The following topics review JAS Scan Results and Post-Scan Action:

* [Ignore an Exposures Violation](#UUID09f42f04003df99ab4bf6d07df06b38c)
* [Advanced Security Scan Results REST API Support](#UUID1a8f272ae5456e8431a7ef3e2dfa9612)

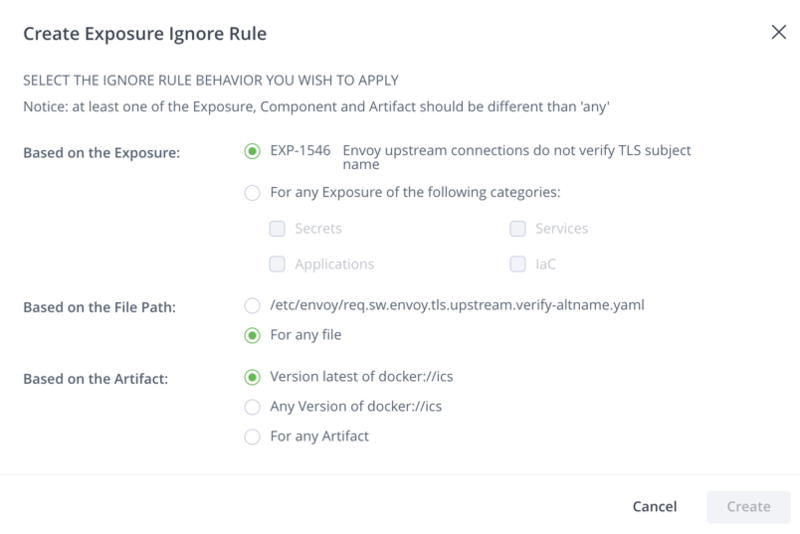
## Ignore an Exposures Violation

Ignore rules allow you to whitelist and ignore security violation rules, in order to filter out unwanted violation noise. For example, you might be running Advanced Scans on a testing repository and don’t want these violations to hinder your testing, or you might have instilled strict actions if a violation is found that is a blocker for continuing your development. You want to ignore the specific violation for the time being.

There are many reasons why you might want to ignore a violation, you can read more about it here [Ignore Rules](https://www.jfrog.com/confluence/display/JFROG/Ignore+Rules).

**Ignoring an Exposures Violation**:

1. In Scans List, select the repository and artifact you want to view violations for.
2. In the **Policy Violations** tab, select the Exposures violation you want to ignore. Exposures violations ID is indicated with an EXP prefix.
3. Click on the menu on the right side of the violation, and select **Ignore Violation**.

* 
* The create Exposure Ignore Rule window appears.
* 

1. Choose a combination of the ignore criteria depending on your needs.

| Ignore Rule | Description |
| --- | --- |
| **Based on the Exposure** |  |
| Exposure Scanner | Ignores all violations for the specific scanner. Take note, if this is checked, all exposure violations related to this scanner will be ignored. |
| For any Exposure of the following categories | Ignores all violations of the specific exposures category. If all categories are selected, no violations will be created for Exposures for the specified scope in the ignore rule. |
| **Based on file path** |  |
| Specific File path | The rule will be applied on the specific path within the specified artifact scope. |
| For any file | The rule will be applied for any file path within the specified artifact scope. |
| **Based on Artifact** |  |
| Artifact name selected current version | The rule will be applied on the specific artifact for that specific version of the Docker image. |
| Artifact name selected any version | The rule will be applied on the specific artifact for all versions of the Docker image. |
| For any Artifact | The rule will be applied on all artifacts that contain that violation in the Docker image. |

Managing, deleting and other Ignore Rules related actions are described in [Ignore Rules](/document/preview/514594#UUID-dcb8d4a3-bb33-8f0f-7c8b-bc917b82cb9d).

## Advanced Security Scan Results REST API Support

You can create an ignore rule for Exposures violation using the following REST API:

* [Create Ignore Rule](/document/preview/514435#UUID-9e299865-7cb3-8415-bda6-87047e05cb09)

# Vulnerability Contextual Analysis

**Subscription Information**

This feature is supported with the **Enterprise X** or **Enterprise+** license, with the Advanced Security Add-on.

JFrog Security and the JFrog research team's continuous effort to enhance security is introducing an additional capability: **Vulnerability Contextual Analysis**. JFrog Xray previously released a powerful capability, the [JFrog Security CVE Research and Enrichment](/document/preview/361955#UUID-83e1c6f5-9695-a9b9-87c4-4cbecb840d24) feature, that helps you with enhanced analysis on CVE findings in a way that allows you to focus on the most important issues with the capability of finding the best resources invested in fixing them. Vulnerability Contextual Analysis is an extension to that capability, ensuring Xray's analysis findings are as focused as possible.

**The Issue**

When Xray scans your packages, it can potentially find thousands of vulnerabilities. Thus, developers will have to sift through these long lists of vulnerabilities to identify their relevance and in some cases, it can be hard to pinpoint where to start, as many of these vulnerabilities may not affect your artifacts. This process is erroneous and time-consuming.

**The Solution**

Vulnerability Contextual Analysis uses the artifact context to eliminate false positive reports on vulnerabilities that are not applicable. This process involves automated scanners running on top of the container to find reachable paths for the analyzed vulnerabilities. Xray automatically validates some high and very high-impact vulnerabilities, such as vulnerabilities that have prerequisites for exploitations, and provides contextual analysis information for these vulnerabilities, to assist you in figuring out which vulnerabilities are applicable to a specific artifact.

**What are the Benefits of Vulnerability Contextual Analysis?**

* Analyzes the finished code the way an attacker would. Know what issues are exploitable and their potential impact.
* Tests an issue in the context of the complete artifact, also within a build or Release Bundle.
* Enables action and remediation in the context of the actual artifact, build or Release Bundle.

**Note**

Important details regarding the Contextual Analysis feature:

Supported Repositories:

* Docker
* Maven
* Gradle

Supported languages inside a container:

* Java
* Go
* Python
* JavaScript
* TypeScript
* Rust with Cargo auditable build
* .Net binaries
* Native binaries (C, C++ ELF)
* Kotlin

Supported languages in source code analysis:

* Java
* Go
* Python
* JavaScript
* TypeScript

Covers More than 1400 high-profile CVEs

* Maven repo is supported in Xray version 3.77.4 and above (Java supports only JAR-of-JARs Uber JAR. See the example [here](https://jfrog.com/help/r/jfrog-security-documentation/create-an-uber-jar-for-contextual-analysis)for creating an Uber JAR)
* Rust binaries in Docker containers are supported in Xray version 3.79.x and above
* .Net binaries in Docker containers are supported in Xray version 3.95.4 and above

See also [Contextual Analysis in JFrog Applications Doc Hub](https://docs.jfrog-applications.jfrog.io/jfrog-security-features/contextual-analysis)

## Work with Contextual Analysis

This section reviews how Contextual Analysis can be enabled and disabled, and how to view statuses and results. It includes the following:

| # | Task | Description | For more information, see... |
| --- | --- | --- | --- |
| 1 | **Enable Contextual Analysis** | Disabled by default, reviews how to enable Contextual Analysis. | [Enable/Disable Contextual Analysis](#UUIDe3ca99a240477da97f3ab84bbae1dbf9) |
| 2 | **Contextual Analysis Statuses and Results** | Provides details about Contextual Analysis results and the different statues that may be displayed. | [Contextual Analysis Statuses and Results](#UUIDd8d95e62b9e12bd53ab71b1f22efb533) |
| 3 | **Run Contextual Analysis on an Existing Artifact** | Describes how to run Contextual Analysis on existing artifacts to discover vulnerabilities. | [Run Contextual Analysis on an Existing Artifact](#UUIDfe1c3a059f13e259985df66aba936c29) |
| 4 | **Create an Uber Jar** | Use “spring-boot-maven-plugin” for creating the Uber JAR. | [Create an Uber JAR for Contextual Analysis](#UUID9d12c33db06630c0ff39e79a728177af) |

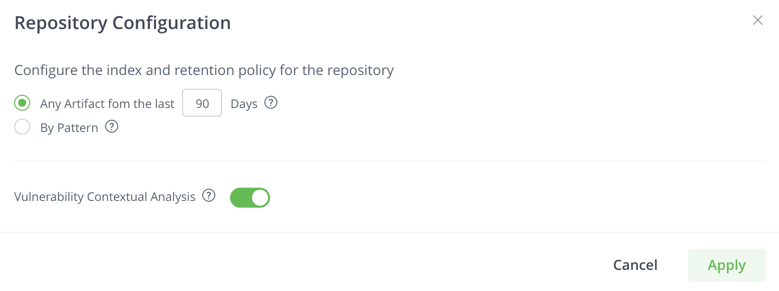
### Enable/Disable Contextual Analysis

Vulnerability Contextual Analysis is disabled by default for new artifacts in all resources that are marked for indexing by Xray. To enable, do the following:

1. In Classic Navigation, navigate to the **Administration** module, go to **Xray** **Settings | General** and click **Indexed Resources**.

* In New Navigation, navigate to the **Administration** module, go to **Xray** **Settings** and click **Indexed Resources**.

1. Select the repository or build and select**Configure**.
2. Enable the **Vulnerability Contextual Analysis** option.

* 

**Note**

Contextual Analysis is applied on new scans only, and not on existing scans. The analysis will run on indexed resources, however, it will not run on the Index Artifacts History. For more information, see </document/preview/625776#UUID-edc92457-f62c-e873-a822-b6989c7c6d95>.

Starting from Xray version 3.66.x and above, you can scan an existing artifact for Vulnerability Contextual Analysis.

Take note that in some cases, as deep scanning is involved, the scan might take longer to complete.

### Contextual Analysis Statuses and Results

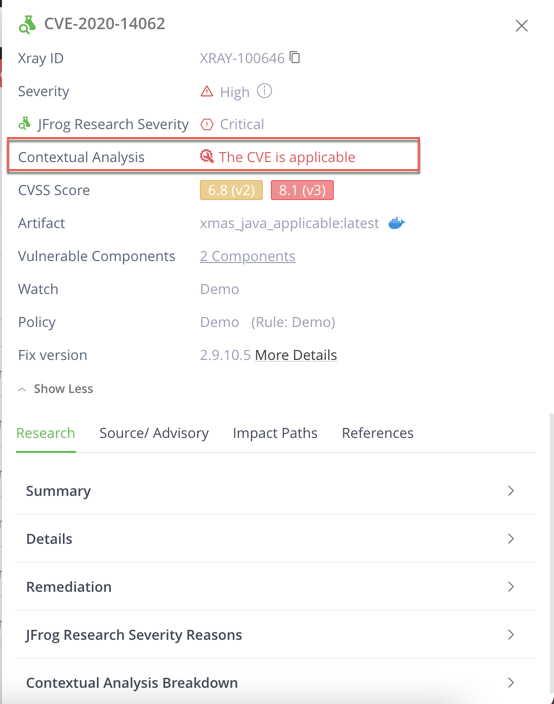
Once an artifact is indexed in Xray as part of a single upload, build, or Release Bundle, Xray will validate if the artifact contains vulnerabilities that are considered to have a very high impact. If such vulnerabilities are found, Xray will run the contextual analysis and retrieve the contextual analysis results. The results consist of the following:

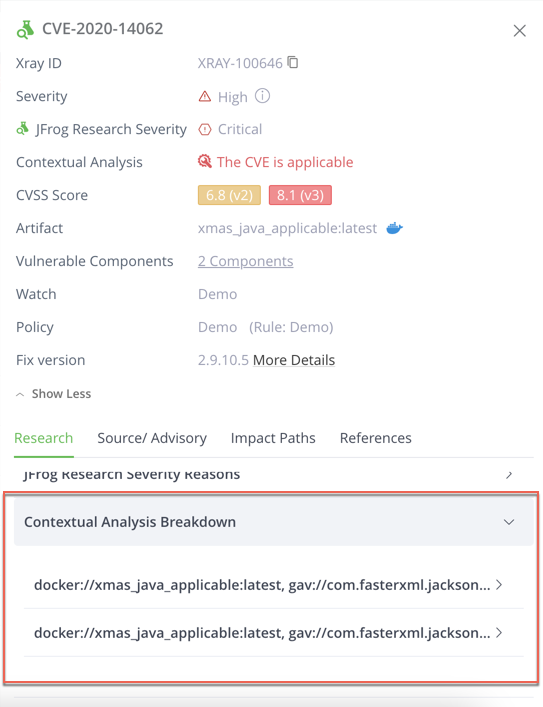
**Vulnerability Contextual Analysis Statuses**

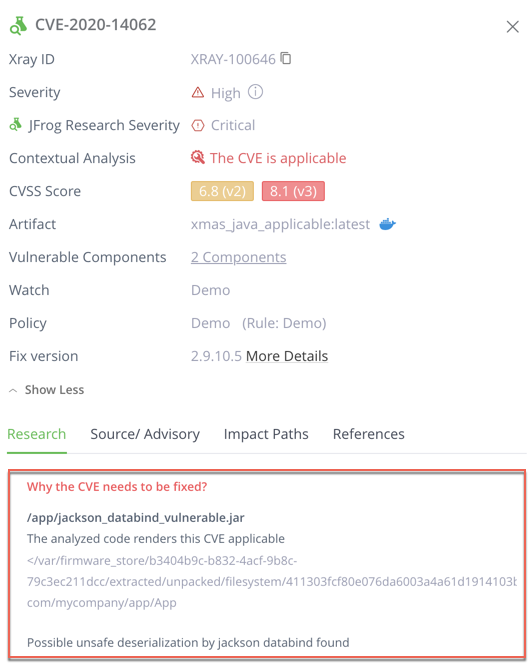
* **Not Scanned**: Initial state, the scan wasn't invoked for the CVE.
* **Applicable**: The vulnerability can be exploited in the context of the scanned artifact.
* **Not Applicable**: The vulnerability cannot be exploited in the context of the scanned artifact.
* **Undetermined:** The applicability cannot be determined by static analysis (e.g. the exploitation requires user interaction).
* **Rescan Required**: A new scanner for this CVE is available, you need to rescan to retrieve applicability results.
* **Upgrade Required**: (Self-Hosted only) The Xray version needs to be updated to receive a new scanner for this CVE. Rescan is required after the upgrade.
* **Not Covered**: Scanner isn't available.
* **Technology Unsupported**: The vulnerability’s package type is currently not supported.
* **Missing Context**: Reachability analysis cannot determine the vulnerability’s applicability. Applicability can be determined by scanning the artifact in a Docker repository in the JFrog Platform.

**Vulnerability Contextual Analysis Results**

The contextual analysis results can be accessed from [Scans List](/document/preview/551896#UUID-82a35411-384b-423c-8aba-0b6796583862).



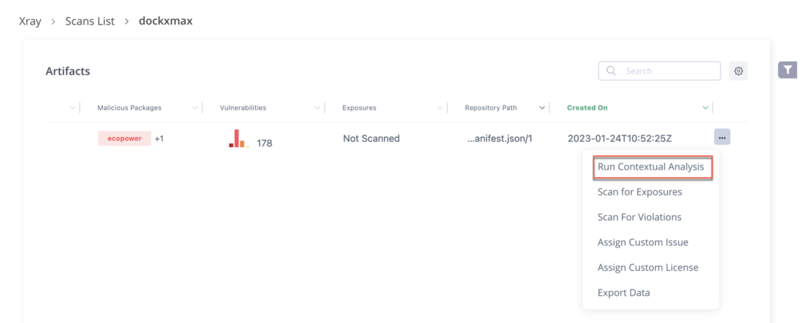




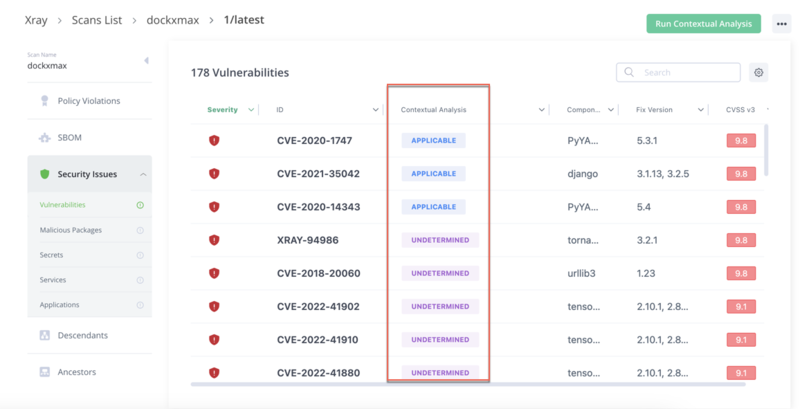
### Run Contextual Analysis on an Existing Artifact

Starting from Xray version 3.66.x and above, you can run contextual analysis on an existing artifact. Do the following:

1. From the **Scans List** page, **Repositories** tab, select the repository.
2. Navigate to the artifact you want.
3. Click the Actions Menu next to the artifact, and select **Run Contextual Analysis**.



The results appear under **Security Issues** > **Vulnerabilities**.



### Create an Uber JAR for Contextual Analysis

You can use “spring-boot-maven-plugin” for creating the Uber JAR. Here is an example of how to use it in a pom.xml:

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example</groupId>  
 <artifactId>MavenTest</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>11</maven.compiler.source>  
 <maven.compiler.target>11</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.apache.tomcat.embed</groupId>  
 <artifactId>log4j-core</artifactId>  
 <version>2.14.0</version>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <finalName>my-project-name</finalName>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <version>2.7.14</version>  
 <executions>  
 <execution>  
 <goals>  
 <goal>repackage</goal>  
 </goals>  
 </execution>  
 </executions>  
 </plugin>   
 </plugins>  
 </build>  
  
</project>

## Contextual Analysis REST API Support

The following REST APIs are supported for the Contextual Analysis feature:

* [Get Contextual Analysis per Vulnerability](/document/preview/514377#UUID-00e06ac2-7e82-93d5-a26a-cc03a574e418)
* [Artifact Summary](/document/preview/514424#UUID-9014965c-7895-9083-e723-0490dd12fe5f) - Applicability information was added to each issue.
* [Build Summary](/document/preview/514423#UUID-ef0d5caf-dbb3-59a4-5a96-16aabb944c87) - Applicability information was added to each issue.
* [Get Violations](https://jfrog.com/help/r/xray-rest-apis/get-violations) - Applicability information was added to each violation.
* [List Ignored Violations](/document/preview/514439#UUID-f0ddd0eb-aef8-fbbb-2ff2-516caf185704) - Applicability information was added to each violation.
* [Scan Build V1](/document/preview/514363#UUID-d15877af-6a8b-6d0d-52c0-bf6970c20106) - Applicability information was added to each alert.
* [Get Repositories Configurations](/document/preview/514360#UUID-087c285b-3fb9-b97a-4a94-227c98ceae78): Added a new parameter vuln\_contextual\_analysis: true or false. Only if feature is enabled and it is possible to enable or disable it per repository.
* [Update Repositories Configurations](/document/preview/514359#UUID-dfc289b8-5668-ed3d-c054-0c71a1a30b08) - Added the option to enable or disable Contextual Analysis per repository with the parameter vuln\_contextual\_analysis: true or false.

# Exposures Scans

**Subscription Information**

This feature is supported with the **Enterprise X** or **Enterprise+** license, with the Advanced Security Add-on.

In addition to Xray's software composition analysis and scanning for vulnerabilities in packages, Xray now enables you to perform scans for multiple categories that cover security issues in your configurations and the usage of open source libraries in your code. Along with other Xray capabilities, such as CVE Enrichment, ContextualAnalysis, and Operational Risk, Xray now provides end-to-end supply chain security to cover different forms of software supply chain attacks.

**The Issue**

When it comes to non-code-related security issues, they are often overlooked in an organization as a potential security threat, since they are the smallest and easiest issues to fix. This leaves your software potentially exposed to security threats due to security malpractices (e.g., missing authentication), insecure configurations (e.g., excessive privileges), weak authentication, and so on.

**The Solution**

Xray conducts an automated security scan to detect these potential security exposures in the analyzed artifact. The scan is performed via automated scanning of the artifact using static analysis scanners, which are continuously enhanced by the JFrog research team. The following sections describe the scanning categories in detail.

**Note**

Exposures supports the following package types:

* Docker
* OCI (Xray version 3.59.4)
* Maven (Xray version 3.78.9)
* npm (Xray version 3.78.9)
* PyPI (Xray version 3.78.9)

## Working with Exposure Scans

This section reviews how to work with Exposure Scans. For more information on Exposure Scans, see [Exposures Scans](#UUIDc8c366caddf0d2369f4b7c66a34bbbcc).

| # | Task | Description | For more information, see... |
| --- | --- | --- | --- |
| 1 | **Enable Scanning Categories** | Disabled by default, reviews how to enable Contextual Analysis. | [Enable/Disable Scanning Categories](#UUIDc8050d576a4598fc96664faa6a386cdf) |
| 2 | **View Exposure Scan Results** | View the results of an exposure scan. | [View Exposure Scan Statuses and Results](#UUIDe6886acb956e64ad9cfc18a426cd1c3c) |
| 3 | **Run an Exposure Scan on an existing artifact** | Starting from Xray version 3.66.x, run a scan for Exposures on an existing artifact | [Run an Exposure Scan on an Existing Artifact](#UUID9651e0090e321a2545a227d6eedd9b5b) |

### Enable/Disable Scanning Categories

The scanning categories are disabled by default. You can enable or disable each category separately as desired.

**Note**

The scanning categories are applied on new scans only, and not on existing indexed artifacts. The scan will run on indexed resources, however, it will not run on the Index Artifacts History. For more information, see [Indexing Xray Resources](https://www.jfrog.com/confluence/display/JFROG/Indexing+Xray+Resources).

Note that in some cases, because deep scanning is involved, the scan might take longer to complete.

If you would like to enable the scanning categories, do the following:

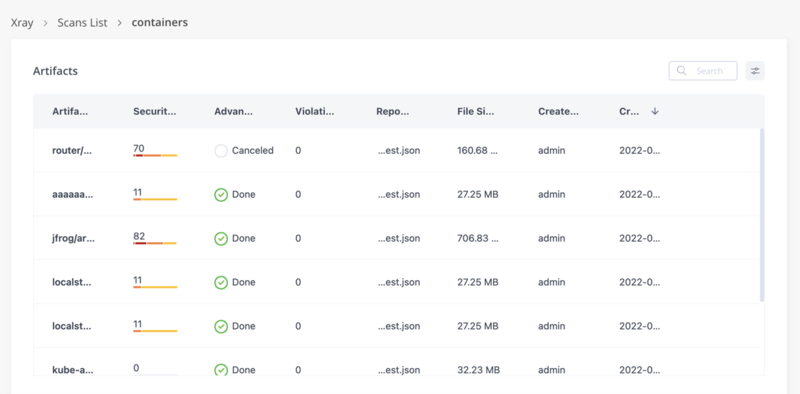
1. Go to the**Administration** module, go to **Xray** **| Settings | General,** and click **Indexed Resources.**
2. Select the repository or build and select**Configure**.
3. Select the categories you want to enable: Vulnerability Contextual Analysis, Services, Secrets, and Applications.

### View Exposure Scan Statuses and Results

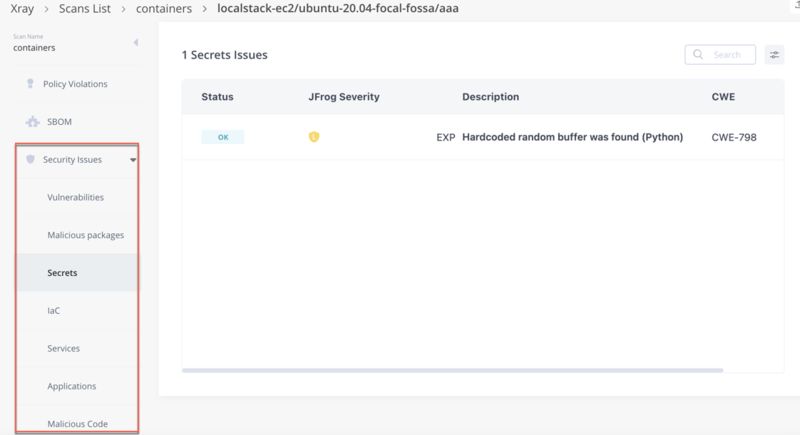
Once an artifact is indexed, Xray will validate if the artifact contains any security issues in any of the categories you have enabled for scanning.

To view scan results, go to**Application | Xray | Scans List.**

1. Select the resource type **Repositories**. Note that, in this version, the new scan categories are only supported for Repositories.
2. Select the resource from the list.

* Each scan contains an overview of the results such as how many vulnerabilities were found, the scan status, and so on. It is important to note that each category has a set of scanners that will search for specific issues. To provide you with full visibility as to what Xray scanned for, the results will show all scanners including items that were scanned and are OK.
* 

1. Select the scan you want to view.

* The scan results are displayed under Security Issues.
* 

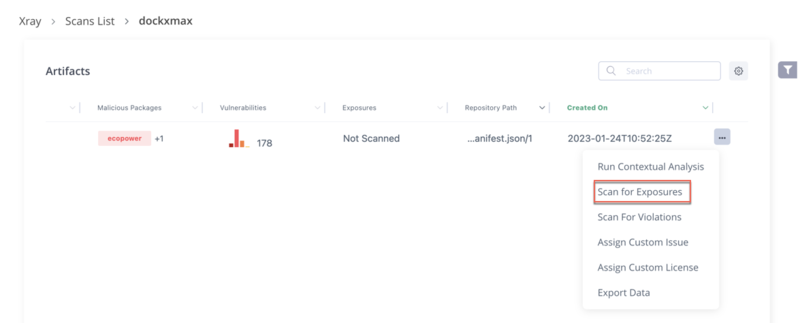
1. Select the issue to view more details. Each issue contains the following information:

| * Field | * Description |
| --- | --- |
| * JFrog Severity Badge | * The severity of the issue that was determined by the JFrog Security Research Team:   + Critical   + High   + Medium   + Low |
| * Status | * There are two possible statues:   + **To Fix**: An issue that was found and should be fixed   + **OK**: An issue Xray scanned for and verified is okay (i.e., no security issues were found) |
| * ID | * Issue identifier |
| * CWE | * The Common Weakness Enumeration (CWE) identifier for the weakness type this issue is associated with. |
| * Fix Cost | * Estimate for the effort involved in fixing the suggested resolution:   + **High effort**: Substantial effort is required from the software developer. Examples include building code from source and applying broad configuration changes.   + **Medium effort:**A medium-level action is required from the software developer. Examples include making changes to existing configurations.   + **Low effort:**Minimal effort is required from the software developer. Examples include removing a file, and making minor changes to existing configurations |
| * Findings | * Provides information on the issue in terms of exactly what and where was found, the security impact of the issue, and what needs to be done to fix it. |
| * Outcomes | * Possible consequences of an attack utilizing this issue. |

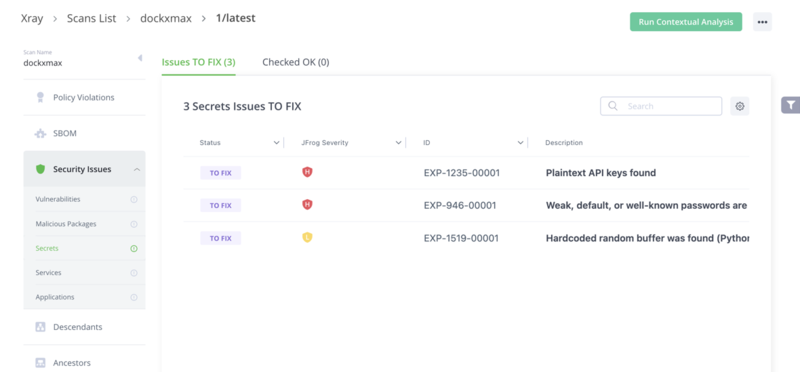
### Run an Exposure Scan on an Existing Artifact

Starting from Xray version 3.66.x and above, you can scan for Exposures on an existing artifact. Do the following:

1. From the **Scans List** page, **Repositories** tab, select the repository.
2. Navigate to the artifact you want to scan.
3. Click the Actions Menu next to the artifact, and select **Scan for Exposures**.



The results appear under **Security Issues**.



## Exposure Scanning Categories

There are a number of categories of Exposure scans, they include the following:

**Secrets Scans - Secrets Detection**

Detects any secret left exposed in the artifacts stored in Artifactory to stop any accidental leak of internal tokens or credentials. For more detailed information, see [Secrets Scans](#UUIDf780f7a64776ff093ef1b60e8908dd36).

**Services Scans - Services Configuration Security**

Detects whether common OSS libraries and services are configured securely, so an application can be easily hardened by default.

Xray scans for configuration issues and security malpractices for specific services and daemons included in your artifacts, such as web servers, database services, proxies, logging daemons, and so on.

**Note**

Supported Services:

* Envoy
* Etcd
* Prometheus
* NGINX
* Apache
* Insecure use of credentials (NGINX credential in config file, credential stored insecurely)
* Enforcement of secure communication (redirecting HTTP to HTTPS, enforcing TLS, TLS version)
* Allowing weak crypto algorithms
* Externally exposing Admin interface
* Un-authenticated access to resources

**Applications Scans - Application Libraries Misuse**

Detects whether common OSS libraries and services are used securely by the application.

Xray scans for configuration issues, security malpractices, and insecure usage of common OSS libraries in your application framework, including the use of excessive privileges, insecure communication methods, insufficient authorization mechanisms, or unsafe cryptographic operations.

**Note**

In this version, only Python and Node-JS applications are supported.

**Examples**:

* Insecure use of credentials (insecure key storage)
* Enforcement of secure communication (redirecting HTTP to HTTPS, enforcing TLS, verifying the TLS certificates of all servers in Python scripts, enforcing TLS version, using secure HTTP headers)
* Use of weak crypto keys
* Throttle logins to prevent brute-force attacks (Throttle Node.js logins to prevent brute-force attacks)
* Invoking Node.js exec functionality with user-provided input

**IaC Scans - IaC Security Analysis**

Scans IaC files stored in Artifactory for early detection of cloud and infrastructure misconfigurations to prevent attacks and data leak.

Xray scans your Terraform state in Artifactory for Cloud services configuration issues such as the following examples. Xray scans Terraform states for AWS, Azure and GCP cloud services.

**Examples**:

* Insufficient access restrictions to services (public access to repositories, publicly accessible clusters, globally readable/deletable/writeable buckets, use of admin roles in ECS services, IAM users with privileged access to all resources, enforce authorization for all API Gateway methods)
* Insecure use of credentials (use of hardcoded credentials)
* Allowing weak crypto algorithms (use of weak cipher suites)
* Running batches in privileged mode
* Enforcement of secure communication (listening to HTTP, unencrypted communications)
* Wildcard actions in Glue policies
* Missing logging (e.g., found CloudTrail trails with logging disabled)
* Disabled upgrades (e.g., RDS database instance with disabled minor engine upgrades)
* Data at rest encryption enablement for Kinesis streams

## Secrets Scans

Detects any secret left exposed in the artifacts stored in Artifactory to stop any accidental leak of internal tokens or credentials. Xray scans your configuration, text, and binary files for plaintext credentials, private keys, tokens, and similar secrets. Xray uses a constantly updated list of more than 150 specific types of credentials. In addition, Xray uses a proprietary generic secrets matcher, for the best coverage possible. Xray also scans for issues in the certificates used in the software, such as expired or weak certificates.

**Supported Scenarios**

JFrog Secret detection currently supports the following scenarios:

* JFrog Platform: Docker, Maven, npm, PyPI, NuGet, Gradle and Generic. Additional supported Package types are coming soon.
* Developer tools: [IDE](https://docs.jfrog-applications.jfrog.io/jfrog-applications/ide), [CLI](https://docs.jfrog-applications.jfrog.io/jfrog-applications/jfrog-cli/cli-for-jfrog-security), [Git (Frogbot)](https://docs.jfrog-applications.jfrog.io/jfrog-applications/frogbot).

**Note**

In the JFrog Platform both binaries and text files are being scanned for secrets.

In the IDE and Frogbot only text files are scanned.

In the CLI there are commands such as ‘jf audit’ that scan source code and look for secrets in text files, and other commands such as ‘jf docker scan’ that scan both binary and text files.

**Supported Secrets Types**

JFrog Secrets detection can detect the following types of secrets:

* [Access tokens (keys)](#X1a89f93cd3909b125e96a5e2db2c7b69c969063)
* [Certificates/private keys](#Xc8891010291e9b32d874514c059acfceb34adf8)
* [High entropy secrets](#Xa15a62d17c86b65a9922ae488c544aa2ee81f08)
* [URL secrets](#X3faf14da32b1b0c296c459d41e4eea2115f390d)

**Access Tokens**

This scanner detects access tokens with a well-defined structure, in either text or binary files. For example:

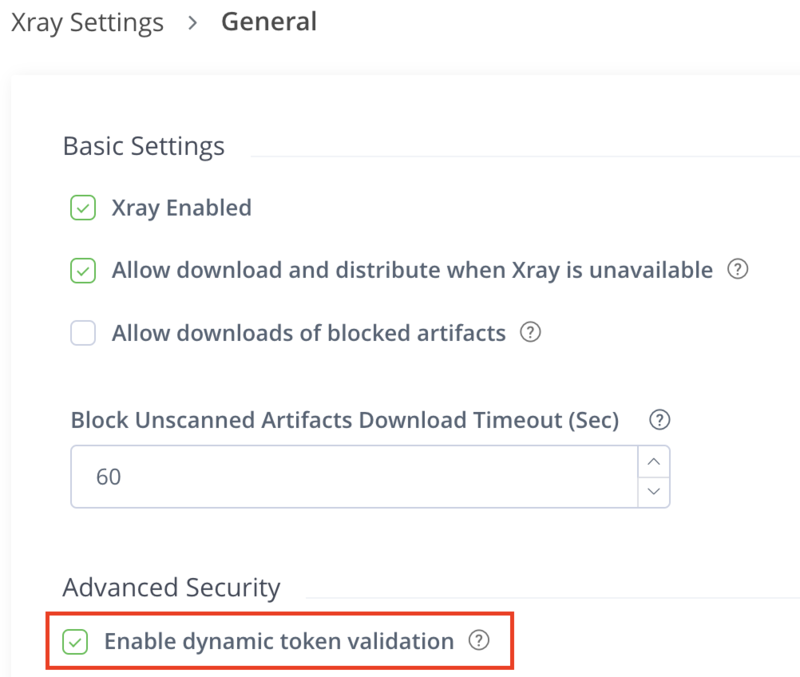
| Platform | Example Token |
| --- | --- |
| AWS | AKIAIOSFODNN7EXAMPLE |
| GitHub | gho\_16C7e42F292c6912E7710c838347Ae178B4a |
| GitLab | gplat-234hcand9q289rba89dghqa892agbd89arg2854 |
| npm | npm\_1234567890abcdefgh |
| Slack | xoxp-123234234235-123234234235-123234234235-adedce74748c3844747aed48499bb |

**Token Validation**

Verify the validity of detected tokens by enabling Token Validation. This feature allows you to distinguish between active and inactive tokens by authenticating against the token provider.

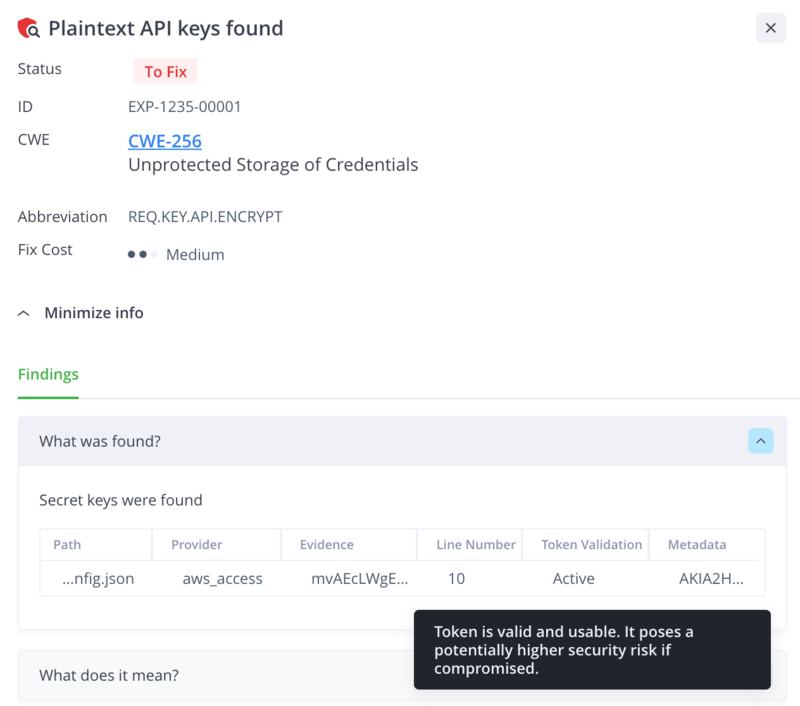
The Dynamic Token Validation feature can be enabled both through the JFrog Platform and through Xray’s REST API.

To enable this feature from the JFrog Platform navigate to **Administration > Xray Settings > General > Advanced Security**and check the **Enable dynamic token validation** checkbox.



To enable through REST API, run the following REST API: </document/preview/733722#UUID-8dd54c7b-d9d2-8dda-8894-d468d91e6007>.

Once enabled, the results are presented in the violation's details under the **Findings** tab. To learn more about results, see [View Exposure Scan Statuses and Results](#UUIDe6886acb956e64ad9cfc18a426cd1c3c).



The **Token Validation** column statuses are:

* **Active**: Token is valid and usable. It poses a potentially higher security risk if compromised.
* **Inactive**: Token is inactive. For tokens associated with Self Hosted providers, our validation cannot determine their activity status.
* **Unsupported**: Token provider is currently unsupported for validation.
* **Unavailable**: Cannot validate due to unknown reasons.

The **Metadata** column:

For several providers, additional identification is coupled to the secret token (AWS has a Token ID associated with the Token Secret). Our scanners detect this additional identification and output it to the Metadata column.

**Supported Providers**

These are all the providers currently supported by this scanner. Some providers also support JFrog’s Token Validation.

| Supported Provider | Token Validation Support |
| --- | --- |
| Adobe | Yes |
| Age File Encryption | No |
| Alibaba | Yes |
| Artifactory | No |
| AWS | Yes |
| Azure | Yes |
| Clojars | No |
| Cloudflare | No |
| Contentful | Yes |
| Databricks | No |
| DigitalOcean | Yes |
| Doppler | Yes |
| Dropbox | Yes |
| Duffel | Yes |
| Dynatrace | No |
| Elastic | No |
| Fastly | Yes |
| Finicity | No |
| Flutterwave | Yes |
| GitHub | Yes |
| GitLab | Yes |
| Google | Yes |
| Heroku | Yes |
| Hubspot | No |
| Intercom | No |
| Jenkins | No |
| Linear | Yes |
| Mailchimp | Yes |
| Mailgun | Yes |
| Mapbox | Yes |
| Messagebird | No |
| New Relic | No |
| npm | Yes |
| NuGet | No |
| Okta | No |
| PayPal | Yes |
| Planetscale | No |
| Postman | Yes |
| Postmark | Yes |
| Pulumi | Yes |
| PyPI | No |
| Rubygems | Yes |
| Sauce | No |
| Searchgaurd | No |
| Sendgrid | Yes |
| Sendinblue | Yes |
| Sentry | No |
| Shippo | Yes |
| Shodan | Yes |
| Shopify | No |
| Slack | Yes |
| Snyk | Yes |
| Sonar | No |
| Square | No |
| StackHawk | Yes |
| Stripe | Yes |
| Telegram | Yes |
| Terraform | Yes |
| Travis CI | No |
| Trello | No |
| Twilio | Yes |
| Typeform | Yes |
| Ubidots | Yes |

**Detection Exceptions**

The following access keys will not raise an alert by the scanners

* Public example keys (AWS example -  ANPAI65L554VRJ33ECQS6)
* Keys in files residing in documentation directories (ex. "/usr/local/share/")
* Keys that match any of the following case-insensitive patterns
  + .\*xxxx.\*
  + .\*test.\*
  + .\*sample.\*
  + .\*example.\*
  + .\*token.\*
  + \*123456.\*
  + .\*abcdef.\*
  + .\*foobar.\*
  + .\*deadbeef.\*
  + .\*xxxxxx.\*
  + .\*1111111.\*
  + .\*0000000.\*

**Certificates/Private keys**

JFrog Secret detection can detect issues in X.509 PEM (textual) and DER (binary) certificates and private keys :

* Certificates containing private keys or standalone PEM/DER private keys
* Expired certificates
* Self-signed certificates

**High Entropy Textual Secrets**

JFrog Secret detection can detect secrets in the general form of  “key = value” in textual files (incl. source files and configuration files) where:

* “key” is a variable name indicative of a secret (for example “secret” or “password”)
* “value” is a string with high entropy/randomness (ex. “d#@B2xN,Y}” and not “123123123”)

For example:

my\_password: "CorrectHorseBatteryStaple123!"

**Detection exceptions**

JFrog employs several heuristics to avoid false positive results. The following textual secrets will not raise an alert by the scanners:

* The key completely matches the string “key” (ex. “key = d#@B2xN,Y}”)
* The value matches any of the following case-insensitive patterns
  + .\*xxxx.\*
  + .\*test.\*
  + .\*sample.\*
  + .\*example.\*
  + .\*123456.\*
  + .\*abcdef.\*
  + AKIA.\*
* The value contains a 6-character long sequence of the same character or consecutive characters (ex. “123456”)
* Secrets in files residing in documentation directories (ex. "/usr/local/share/")
* Secrets in 3rd-party files (ex. A file belonging to a 3rd-party npm package)
* Secrets in files where the file path matches any of the following case-insensitive patterns
  + .\*example.\*
  + .\*sample.\*

**URL Secrets**

JFrog Secret detection can detect secrets in textual files where the secret is embedded in a URL (for example; https://myuser:mypass@somedomain.com)

**Detection exceptions**

The following URLs will not raise an alert by the scanners:

* URLs that match any of the following case-insensitive patterns
  + .\*example.\*
  + .\*foo.\*
  + .\*bar.\*
  + .\*example.\*
  + .\*sample.\*
  + .\*some.host.\*
  + .\*some.url.\*
  + .\*site.com.\*
  + .\*google.com.\*
  + .\*url.com.\*
  + .\*xyz.\*
  + .\*abc.\*
  + .\*domain.com.\*
  + .\*host.com.\*
  + .\*host.page.\*
  + .\*codecov.\*
  + .\*deepsource.\*
  + .\*shields.io.\*
* URLs where the authentication part matches any of the following case-insensitive patterns
  + .\*user:pass.\*
  + .\*username.\*
  + .\*anonymous.\*
  + .\*test.\*
  + .\*foo.\*
  + .\*bar.\*
  + .\*example.\*
  + .\*myuser.\*
  + .\*baz.\*
* URLs that reference an image file (ex. URL ends with “.jpg”)

## Exposure Scans REST API Support

The following REST APIs are supported for the Exposures Scanning Categories feature:

* [Exposures Scanning-Get Results List](/document/preview/514540#UUID-1eb2c456-4ad6-e124-4c74-bc522497f004)
* [Exposures Scanning-Get Results Details](/document/preview/514541#UUID-077c67ab-1301-860b-7e0f-4a86f1004c67)
* [Exposures Scanning-Get Findings](/document/preview/514542#UUID-a2450728-32c1-6ec5-2d91-1bbd671d7a2b)
* [Exposures Scanning-Get Evidence](/document/preview/514543#UUID-9ef8a3da-f629-d4cd-61ec-9a24f809d6fc)
* [Exposures Scanning-Get Rows](/document/preview/514544#UUID-3b796a13-c273-bb3c-2b3f-6db2ade879e9)
* [Get Repositories Configurations](/document/preview/514522#UUID-17cacaed-1c94-02e8-c1f7-7206fa2f543d) - Added new Exposure data, under the Exposure parameter.
* [Update Repositories Configurations](/document/preview/514521#UUID-c764573b-2406-2877-2979-799a2264e3fa): Added new Exposure configuration parameters.