JFROG XRAY

SECURITY AND COMPLIANCE OF THE OPEN SOURCE SOFTWARE DEPENDENCIES YOU RELY ON

May 2021 | www.jfrog.com

TABLE OF CONTENTS

INTRODUCTION

With software now underpinning and fueling all business processes, DevOps teams are aligned with and directly influence their companies' competitive position. Their work -- the continuous creation and enhancement of business-critical applications -- impacts revenue growth, operational efficiency, customer satisfaction, brand reputation and much more.

Because they've become crucial for business success, DevOps teams are under intensifying pressure to streamline and accelerate software development and delivery every day. However, in their eagerness to speed up their software pipelines through automation and collaboration, DevOps teams can't overlook security. If they deliver unsafe applications to employees and customers, DevOps' benefits evaporate and their business suffers.

DON'T OVERLOOK SECURITY

Unfortunately, security is still an afterthought for many enterprise DevOps teams, as GigaOm Analyst Jon Collins observed recently. "Too often, security is brought into the timeline just before deployment, risking last minute headaches and major delays," he wrote in a blog post.

Because DevOps teams remove traditional gates and guardrails, and use more distributed, hybrid and componentized technologies, such as cloud, containers and microservices, they face new security challenges that can't be overcome with legacy tools and processes. "The continuous integration and continuous delivery (CI/CD) process of DevOps is as impactful a change to cybersecurity programs as the changes to the applications and infrastructure that these methodologies manage," Doug Cahill, ESG Research analyst, wrote.

Take the case of containers, whose popularity has revolutionized cloud-native application development, thanks to their portability, scalability and flexibility, but which organizations must secure with new best practices and tools. As Forrester Research analyst Sandy Carielli pointed out in a blog post: "Security pros are brought in later and left with the suboptimal task of applying existing tools and traditional security mindsets to secure containers -- and discovering that those are ill-equipped to the task."

Then there's the growing usage of OSS components, which help quicken application development but often contain security issues, like vulnerabilities. Open source software (OSS) makes up almost half of the code in the applications built by developers surveyed recently by IDC. Clearly, DevOps teams that use these OSS components must properly scan and analyze them for security and compliance issues.

Other security challenges for DevOps teams include the growing use of hybrid cloud environments, along with organizational and logistical obstacles, such as siloed security teams. As these challenges have piled up, many IT teams have resorted to accumulating a growing number of security point products that often do not interoperate and change the defined development workflow to further cloud visibility into DevOps security and compliance practices.

"Organizations are overwhelmed with the amount of and overlap in issues raised from multiple testing tools, complicating prioritization and mitigation, so integrated application security platforms are desired," ESG analyst Dave Gruber wrote.

Unsurprisingly, cyber criminals have taken notice of these challenges. Always on the hunt for new and effective hacking vectors, they're increasingly targeting software development pipelines. For example, they've embraced upstream supply chain attacks, in which they stealthily infect a software provider's code during the development stage. That way, hackers' malware hides in legitimate software and gets shipped to thousands of customers through otherwise official distribution methods. A high-profile example of such an attack was the SolarWinds hack in late 2020, which affected prominent Fortune 500 companies and U.S. federal government agencies.

DEVSECOPS

To protect DevOps pipelines from vulnerabilities, misconfigurations, and other security gaps, what's needed is an approach that automates and embeds security checks across the software development lifecycle (SDLC). This is called DevSecOps.

As GigaOm's Collins explains in his report "GigaOm Radar for Evaluating DevSecOps Tools," DevSecOps' principle is "bringing security best practices as early as possible into DevOps-based software creation, delivery and operation," while the tools should "automate best practices, augment the pipeline and support development activities."

With DevSecOps, organizations can quickly and continuously detect security and compliance issues in their software -- from design to production. That way, when a problem is identified, the DevOps team can immediately "shift left" in their CI/CD process and fix it before the unsafe code in question moves to the next stage.

"Historically, even if companies adopted new DevOps practices, security teams often still existed in silos and did not embrace 'continuous methodologies.' With security becoming an increasing priority, bringing it into the automation fold is rising. DevSecOps is the natural stepping-stone in the digital transformation journey," reads a note from investment bank Cowen.

HOW JFROG CAN HELP YOU

In this white paper, we'll explain how you can implement a DevSecOps strategy using the JFrog DevOps Platform -- in particular JFrog Artifactory, a universal artifact repository manager, and JFrog Xray, a security and compliance management tool for open source software. Read on to learn how to obtain continuous and comprehensive security and compliance, along with full visibility and control of your SDLC with JFrog as your DevSecOps centerpiece.

WHAT IS SOFTWARE COMPOSITION ANALYSIS (SCA)?

As the use of OSS has skyrocketed, cyber criminals have focused on it as an attack vector. "Open source vulnerabilities are especially attractive in attacking scenarios because they represent a type of vulnerability that can be used across applications, a known element of an otherwise custom application," 451 Research analyst Daniel Kennedy noted.

In response, Software Composition Analysis (SCA) tools have emerged to help developers and DevOps teams manage, monitor and secure their OSS usage. SCA tools also help track OSS usage across teams and development sites, while providing automated monitoring of usage.

Popular security products like Static Application Security Testing (SAST) tools only help identify vulnerabilities in your own proprietary code, not in OSS dependencies. Therefore, SAST tools may in some cases only find vulnerabilities in ~10% of your application code.

SCA tools also help to manage and monitor license compliance in OSS dependencies, a growing concern for legal teams and CEOs wary of failed audits, or of expensive intellectual property or license infringement cases. It's critical to know what OSS is being used, its license type, by which team, and in which builds and product releases.

Effective SCA tools provide visibility into each OSS license, its version and the presence and severity of any known vulnerabilities. For automated monitoring and governance, SCA tools need policy enforcement capabilities. These should be configurable to trigger specific actions for identified security or compliance violations, based on what is being scanned and its SDLC stage.

Consider the following when evaluating an SCA tool:

Ecosystem Integration: For maximum effectiveness at monitoring and scanning OSS components, an SCA tool must tightly integrate with your artifact management tools/package managers (i.e. repositories), IDEs, CI servers and more, via out-of-the box integrations or via automation scripts or via open REST APIs or a combination of these

Technology Support: How universal is the SCA tool? Does it support all the programming languages and package types you use? Does it support container scanning? Does it support cloud-native development and deployment?

Vulnerability Intelligence: Security tools are only as good as their vulnerability data. You need comprehensive and up to date license and vulnerability intelligence so you can eliminate the most vulnerabilities.

Deployment Topology Support: Since flexible and dependable deployment methodologies are the backbone of software supply chains, SCA tools must support your day-to-day DevOps deployment topology. It should also support multi-sites and provide high availability, as well as support self-hosted (on-premises), cloud, multi-cloud, and hybrid

WHAT IS JFROG XRAY?

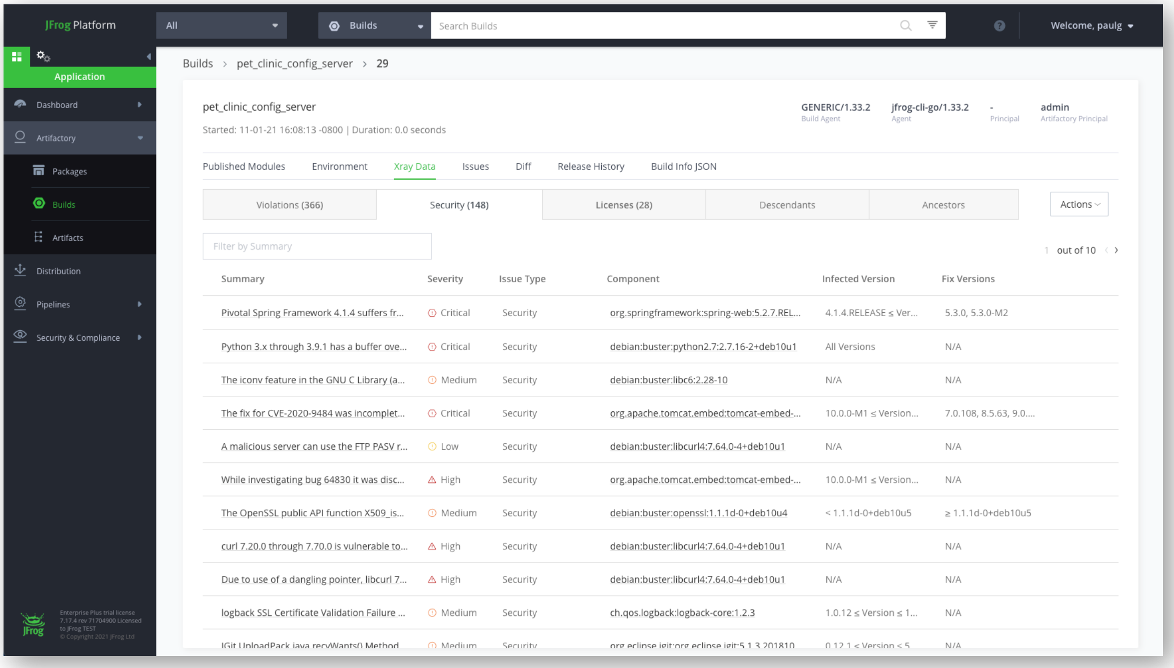
JFrog Xray is a universal SCA solution that helps developers and DevSecOps teams continuously identify OSS vulnerabilities and license compliance violations, before they make it into production releases. Xray is an integral part of the JFrog Platform, which provides a singular and indivisible DevOps experience. Xray also forms a tight unit with Artifactory, combining to create a comprehensive artifact management and SCA DevSecOps solution.

Let’s look at key Xray features that differentiate it from other SCA tools.

NATIVE INTEGRATION WITH ARTIFACTORY

Xray and Artifactory share a data model and metadata, enabling you to create an automated open source security and compliance solution as part of your SDLC. They provide a wealth of metadata about all of your artifacts, builds and binaries.

Artifactory, a system-of-record binary repository, indexes not only standard package metadata but also custom and package properties, exhaustive build information, deploy information, QA status and much more, going beyond stateless metadata on specific binary signatures. This metadata reveals the context of the binary artifact within the organization, and its history in the SDLC.



Builds > pet_clinic_config_server > 29

pet_clinic_config_server

Started: 11-01-21 16:08:13 -0800 | Duration: 0.0 seconds

GENERIC/1.33.2 Build Agent | jfrog-cli-go/1.33.2 Agent | - Principal | admin Artifactory Principal

Summary	Severity	Issue Type	Component	Infected Version	Fix Versions
Pivotal Spring Framework 4.1.4 suffers fr...	Critical	Security	org.springframework:spring-web:5.2.7.REL...	4.1.4.RELEASE ≤ Ver...	5.3.0, 5.3.0-M2
Python 3.x through 3.9.1 has a buffer ove...	Critical	Security	debian:buster:python2.7:2.7.16-2+deb10u1	All Versions	N/A
The iconv feature in the GNU C Library (a...	Medium	Security	debian:buster:libc6:2.28-10	N/A	N/A
The fix for CVE-2020-9484 was incomplet...	Critical	Security	org.apache.tomcat.embed:tomcat-embed-...	10.0.0-M1 ≤ Version...	7.0.108, 8.5.63, 9.0....
A malicious server can use the FTP PASV r...	Low	Security	debian:buster:libcurl4:7.64.0-4+deb10u1	N/A	N/A
While investigating bug 64830 it was disc...	High	Security	org.apache.tomcat.embed:tomcat-embed-...	10.0.0-M1 ≤ Version...	N/A
The OpenSSL public API function X509_is...	Medium	Security	debian:buster:openssl:1.1.1d-0+deb10u4	< 1.1.1d-0+deb10u5	≥ 1.1.1d-0+deb10u5
curl 7.20.0 through 7.70.0 is vulnerable to...	High	Security	debian:buster:libcurl4:7.64.0-4+deb10u1	N/A	N/A
Due to use of a dangling pointer, libcurl 7...	High	Security	debian:buster:libcurl4:7.64.0-4+deb10u1	N/A	N/A
logback SSL Certificate Validation Failure ...	Medium	Security	ch.qos.logback:logback-core:1.2.3	1.0.12 ≤ Version ≤ 1...	N/A
JGit UploadPack.java recvWants() Method	Medium	Security	org.eclipse.jgit:org.eclipse.jgit:5.1.3.201810	0.12.1 ≤ Version ≤ 5	N/A

Through its unique interface with Artifactory, Xray looks deeper than third-party SCA tools into OSS components and their metadata across a variety of package formats including Docker, Maven, Gradle, npm, NuGet and more. Xray's deep recursive scanning combined with the indexed metadata in Artifactory, gives Xray a unique capability to analyze the relationships between binary artifacts and understand the impact that any vulnerability in one component has on any others- including ancestors and descendants.

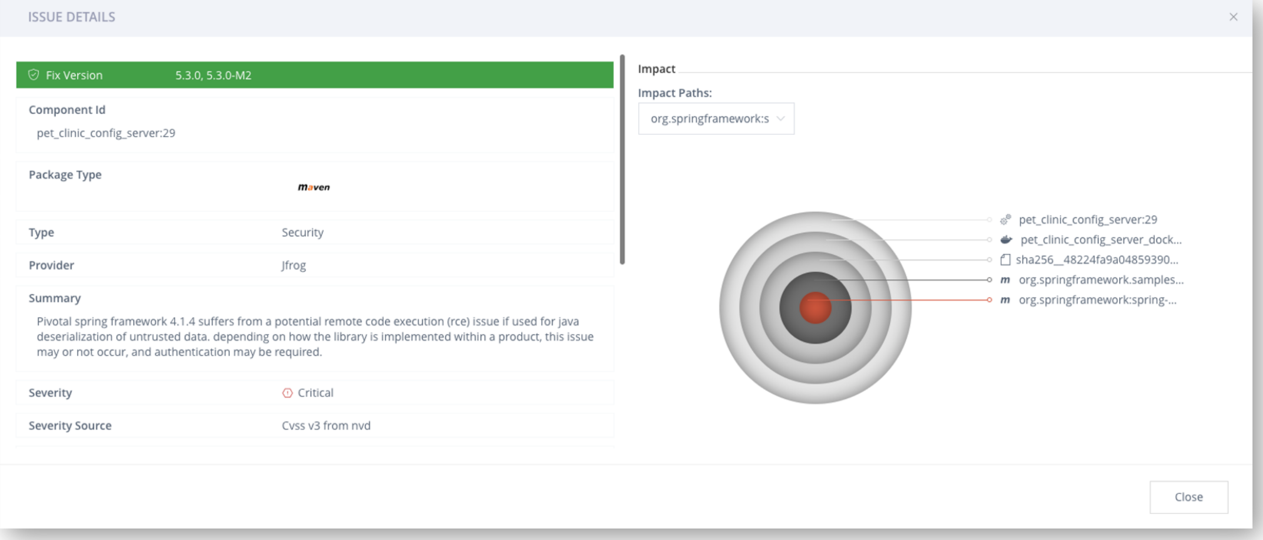
The Xray-Artifactory integration also lets Xray uniquely combine data feeds with Artifactory's exhaustive metadata. This empowers Xray to place security and license compliance information right next to all of your other DevOps and binary metadata.

DEEP RECURSIVE SCANNING

JFrog Xray's deep recursive scanning uniquely peels away and scans every single layer of an image until it has found the absolute lowest base layer. By scanning binary components and their metadata, and recursively going through dependencies at any level, Xray provides radical transparency into your software architecture. It also gives you unprecedented visibility into vulnerable artifacts lurking anywhere in your organization from your development floor to your production datacenter.

JFrog Xray starts with your primary software component, and then recursively drills down to identify its dependencies, and those dependencies' dependencies, and so on, until every single artifact of your software, whether directly or indirectly, has been identified. In fact, as an open, flexible and package-agnostic tool, Xray can accommodate new package formats and perform the same deep recursive scanning.

> "Xray allows us to be able to scan through all the different docker layers and find out what binaries are actually being included in here and that way we have a process in place that we can actually go and notify a team and help them understand that there are vulnerabilities in your build pack."
>
> *Brad Becktell, DevOps Engineer*
> *Kroger*



ISSUE DETAILS

Fix Version	5.3.0, 5.3.0-M2
Component Id pet_clinic_config_server:29	
Package Type	maven
Type	Security
Provider	Jfrog
Summary Pivotal spring framework 4.1.4 suffers from a potential remote code execution (rce) issue if used for java deserialization of untrusted data. depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required.	
Severity	Critical
Severity Source	Cvss v3 from nvd

Impact

Impact Paths: org.springframework:s

×
pet_clinic_config_server:29
pet_clinic_config_server_dock...
sha256__48224fa9a04859390...
m org.springframework.samples...
m org.springframework:spring-...
Close

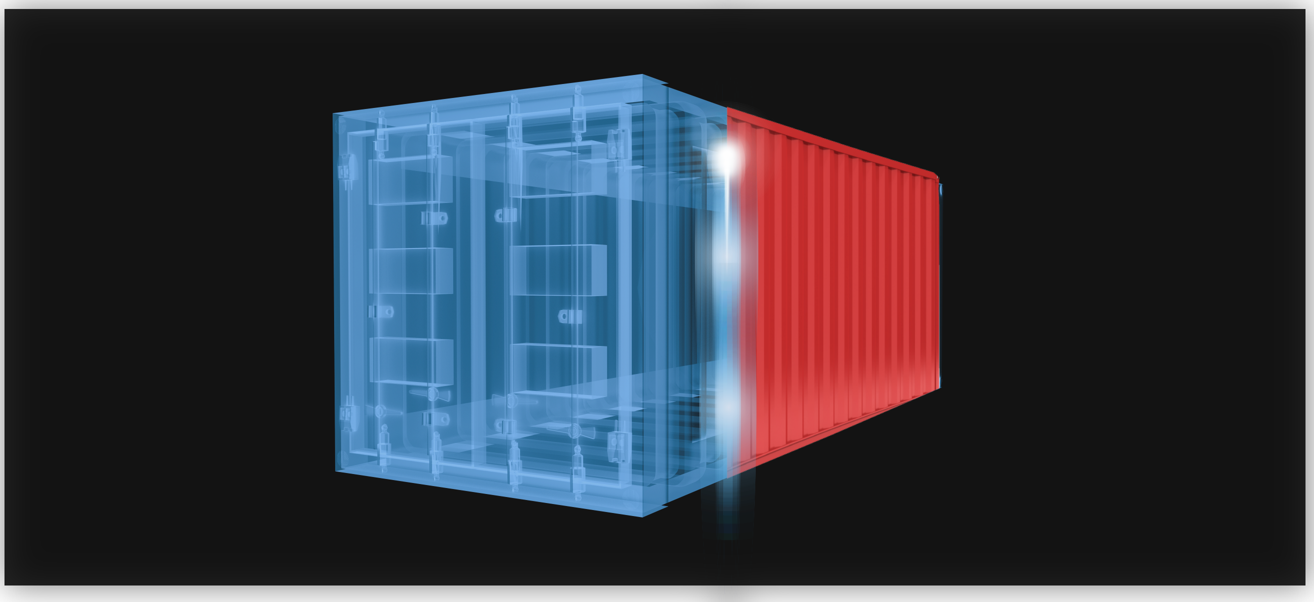
Once all components and dependencies have been identified, Xray cross-references them with its integrated vulnerability and license intelligence, which comes from VulnDB, the U.S. National Vulnerability Database (NVD) and other private and public sources. You can also connect other public or private vulnerability databases to Xray. Once you have set up policies and watches, Xray will automatically alert you to any component or dependency matches.

CONTAINER AND IMAGE SCANNING

The popularity of microservices and of cloud-native development and deployments has boosted the adoption of Docker, containerized applications and Kubernetes. However, this has introduced risk into the software supply chain, requiring new security technology. Enter container scanning. This is now a must-have to ensure container images are free from vulnerabilities in cloud-native software development pipelines. Xray supports the scanning of containerized applications and Docker images for OSS vulnerabilities and license compliance issues through all image layers.

JFrog Xray's deep recursive scanning peels away and scans every single layer of a container image and does not stop at your Docker image base layer. It reveals all of the different layers and their dependencies ensuring that every software artifact that is included in your Docker image has been scanned for vulnerabilities and license issues.

It will identify both OS and application components in the image and all of the dependencies associated with them. Unlike some other SCA tools JFrog Xray will keep scanning no matter how many layers there are. Even when images uploaded to your Docker registries in the Docker repository are given a clean bill of health, Xray continues to securely scan them to make sure they are not infected with any new vulnerabilities.



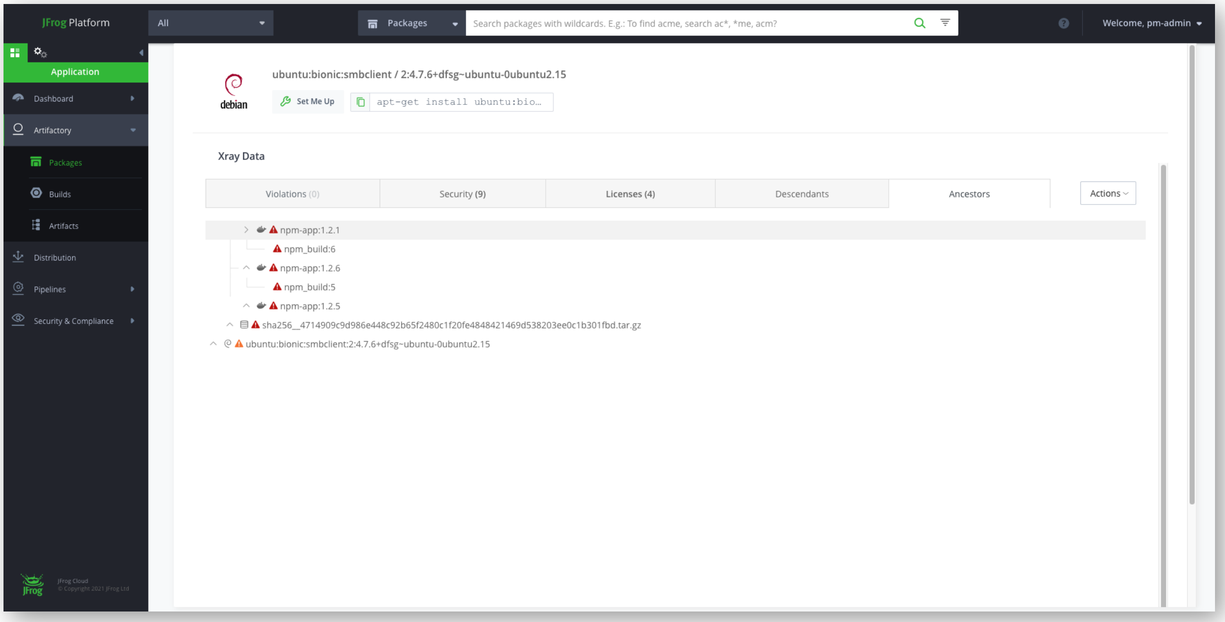
IMPACT AND CONTINUOUS ANALYSIS

With Xray's deep recursive scanning, you get a complete picture of all component relationships in your application. Once Xray detects a license issue or vulnerability in a component, it displays a graph showing the relationship between the component and its impact on all others that are affected.

The relationships may be in the context of binary artifacts, builds, or deployments that are connected to the flagged component. Impact analysis can be triggered manually according to internal policies, or automatically in response to an event.

Issues and vulnerabilities in your software may be identified at any time, from initial development to production. These can range from critical security flaws to a simple deprecation of an open source component but they all can have serious consequences.

Xray knows what open source dependencies are in your production releases and will trigger a violation if any new vulnerabilities are discovered in those components even after production release. This will give you awareness of the need to address any new vulnerabilities in a post production release.



ubuntu:bionic:smbclient / 2:4.7.6+dfsg~ubuntu-0ubuntu2.15

`apt-get install ubuntu:bio…`

Xray Data

```
    > npm-app:1.2.1
        npm_build:6
    ^ npm-app:1.2.6
        npm_build:5
    ^ npm-app:1.2.5
  ^ sha256__4714909c9d986e448c92b65f2480c1f20fe4848421469d538203ee0c1b301fbd.tar.gz
^ ubuntu:bionic:smbclient:2:4.7.6+dfsg~ubuntu-0ubuntu2.15
```

IMPACT PATH

One of the unique features of Xray is the tracing of the impact path of a vulnerable component. A component can appear in multiple places within a build image or in multiple builds. Xray will show you all places where a vulnerable component is impacting your software.

CUSTOM API-DRIVEN AUTOMATION

Xray comes with a set of analyses out-of-the-box, including vulnerability monitoring, open source license compliance, component version change detection, and more. However, your organization may require its own set of parameters to monitor, such as quality criteria, performance criteria or even custom properties assigned to binary artifacts. For example, you may want to receive an alert (or fail a build!) if the quality rating of a component in your staging repository is downgraded.

Through an open API, you can configure Xray to analyze any criteria of your components or their metadata. This ability lets you define any custom analysis to meet the requirements of your organization.

CI/CD PIPELINE INTEGRATION

Xray can be integrated into an organization's CI/CD pipeline to stop build jobs containing vulnerabilities early on in the process. As part of a fully automatable process, Xray receives information about a build run by your CI server, runs a recursive scan down to the build's deepest level dependency, and notifies the CI server about any detected vulnerabilities.

Some organizations force developers to scan all builds they run and fail them immediately if vulnerable artifacts are found. While this prevents infected builds from reaching production, it inhibits developers' creativity. A better solution is to periodically run a scan once the code of several developers has been merged: for example, during a nightly build run by an organization's CI server. While the default action (in Jenkins) is to simply stop the build, you can configure Xray and your pipeline to do other things like send email notifications, IMs or even run a different build job.

The cost of remediating a vulnerability is like the cost of fixing a bug and therefore the earlier you find and remediate a vulnerability in the release cycle, the lower the cost. JFrog Xray is instrumental in flagging vulnerable components across your CI/CD pipeline.

Xray scans large builds in seconds, so you can confidently integrate DevSecOps into your CI process without slowing it down or your releases. Other SCA solutions can add significant delay to the CI process, especially when you're running hundreds of builds concurrently. Such a delay can cause organizations to postpone scanning until their next step, increasing remediation costs if vulnerabilities are found later in the pipeline. With Xray, you get a timely and integrated "shift-left" DevSecOps process.

ECOSYSTEM INTEGRATION

JFrog Xray is a fully automatable tool with CLI support and a rich REST API, enabling easy integration across your SDLC. This allows you to complement Xray's capabilities by integrating it with other security tools, such as DAST and SAST solutions.

Xray can become an integral part of your software supply chain through its out-of-the-box integrations, IDE integrations, and REST API. It allows you to 'Shift-Left' by embedding security and compliance at any stage of your CI/CD pipeline from coding to deployment.

Out-of-the-box Integrations: Utilize Xray to automatically scan builds as part of your CI/CD pipeline, through native integration with leading CI servers including JFrog Pipelines, Jenkins, CircleCI, TeamCity and many more. 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.

REST API: Xray provides a convenient and up-to-date self-descriptive REST API that can be used by various tools/frameworks to automate the creation of REST calls. It provides easy integration and automation with leading DevOps tools. Minimize any manual processes across your software development pipeline and have Xray provide your other DevOps tools the security data they need.

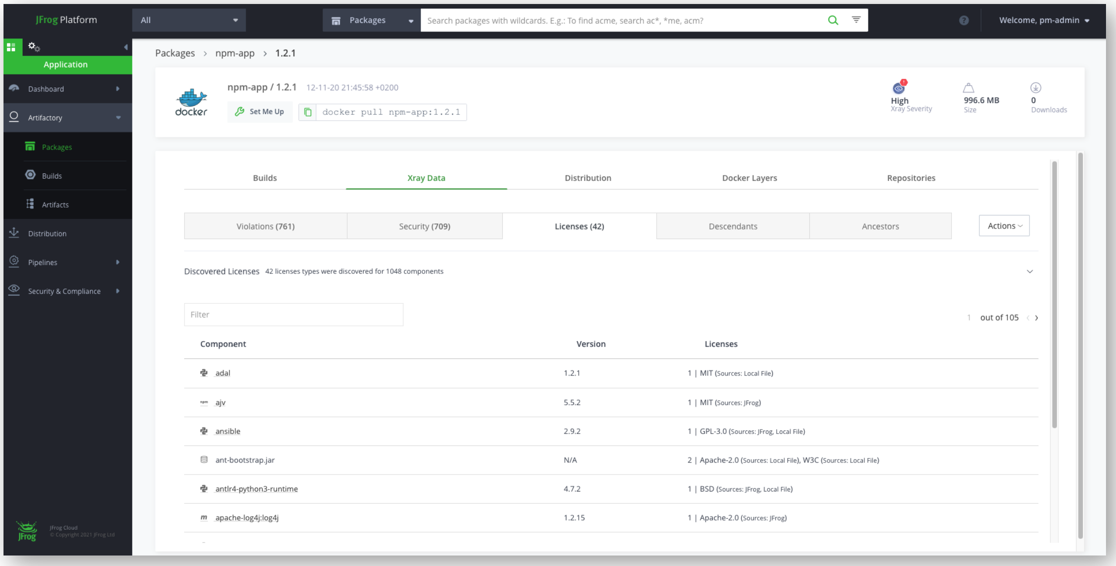
IDE Integration: Xray's IDE integration completes the CI/CD process, by bringing Xray's issue discovery one step earlier in the SDLC, to development time. With integrations for industry leading IDEs - IntelliJ IDEA, Eclipse, Visual Studio and Visual Studio Code - it provides developers with critical insights as early as possible in the development phase making it less likely for vulnerable components to ever reach production. Built into the IDE integration are details of the vulnerable dependency and the suggested remediations to solve the problem discovered.

"Too often, security is brought into the timeline just before deployment, risking last minute headaches and major delays."

Jon Collins, GigaOm Analyst

AUTOMATED LICENSE COMPLIANCE

Xray provides the ability to define and automate enforcement of your organization's license compliance policies. Different mitigation behavior can be set based on the context of the licence type and where the component is being used. Upon detection of license violations, Xray can notify users in several different ways including: emails, Slack, Jira tickets, 3rd party incident response platforms like PagerDuty or any other system via Webhooks. Besides defining violations and generating notifications, the system lets you set up enforcement actions, including blocking the download of a vulnerable binary, failing a build, and preventing the distribution of a vulnerable release bundle.



Packages > npm-app > 1.2.1

npm-app / 1.2.1 12-11-20 21:45:58 +0200

docker pull npm-app:1.2.1

High Xray Severity | 996.6 MB Size | 0 Downloads

Discovered Licenses 42 licenses types were discovered for 1048 components

Component	Version	Licenses
adal	1.2.1	1 | MIT (Sources: Local File)
ajv	5.5.2	1 | MIT (Sources: JFrog)
ansible	2.9.2	1 | GPL-3.0 (Sources: JFrog, Local File)
ant-bootstrap.jar	N/A	2 | Apache-2.0 (Sources: Local File), W3C (Sources: Local File)
antlr4-python3-runtime	4.7.2	1 | BSD (Sources: JFrog, Local File)
apache-log4j:log4j	1.2.15	1 | Apache-2.0 (Sources: JFrog)

CONAN AND C/C+ SUPPORT

Xray scans Conan packages, as well as C and C++ builds, deployed to Artifactory. Conan is a dependency and package manager for C and C++ languages. It is free and open-source, and it works on all OS platforms. It integrates with all build systems like CMake and Visual Studio, as well as proprietary ones. A powerful Conan feature is its ability to create and manage pre-compiled binaries for any possible platform and configuration.

Xray supports these four main use cases for Conan and C/C++:

- Xray scans packages downloaded from ConanCenter to Artifactory
- Xray scan packages built with Conan that are uploaded to Artifactory
- If you're building a Conan package and integrating Xray into your CI process, Xray will scan those Conan builds
- Even if you're not using Conan, Xray will scan your C++ builds.

CVSS v3 SUPPORT

The Common Vulnerability Scoring System (CVSS) is an open industry standard for assessing the severity of software security vulnerabilities. The scoring algorithm assigns severity scores to security vulnerabilities using several metrics that endeavour to approximate the ease of the exploit and the impact of the exploit. Xray collects the scores and severities from two different sources:

NVD: The National Vulnerability Database which contains known vulnerabilities with their respective CVSS score.

OS Package Security Advisory: Some open source operating systems have their own security trackers with further analysis of the vulnerability inside the operating system package.

SCORE RANGE AND SEVERITY LEVELS

The goal is to allow you to prioritize responses and resources according to the level of the threat. Scores range from 0 to 10, with 10 being the highest severity. CVSS v3 also provides a severity description as follows:

- Critical (CVSS score 9.0-10.0)
- High (CVSS score 7.0-8.9)
- Medium (CVSS score 4.0-6.9)
- Low (CVSS score 0.9-3.9)
- Unknown

Xray still supports CVSS v2 scoring, but will only use it if the CVSS v3 score is not available. Security rules when set are measured against the CVSS v3 score and severity level for triggering violations, as described in Creating Xray Policies and Rules.

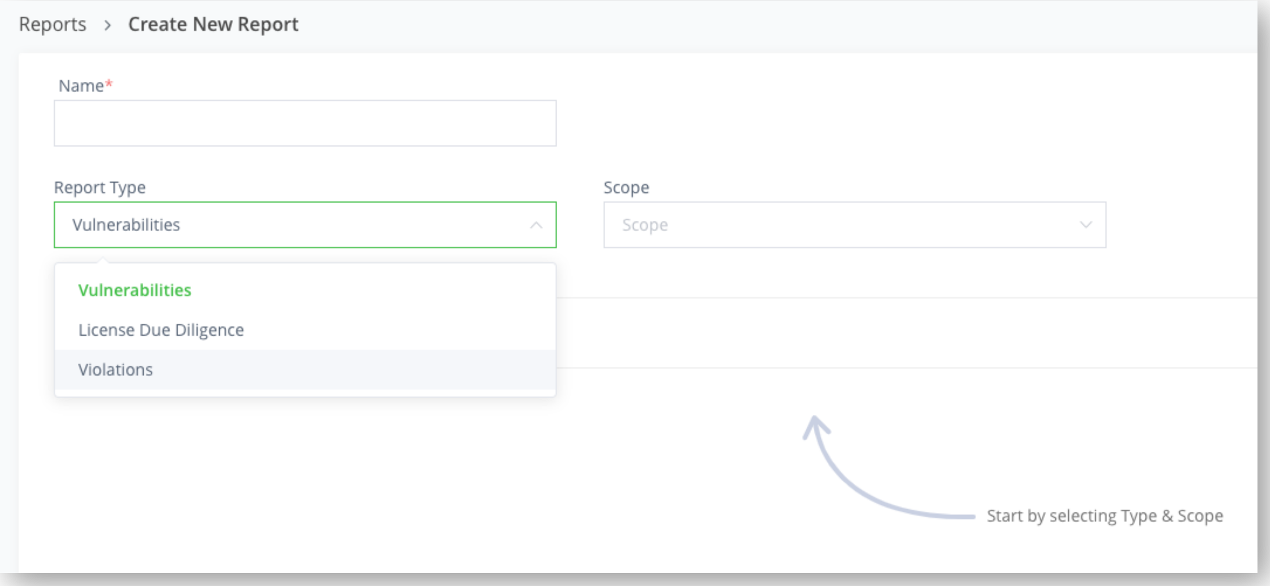
SET PERMISSIVE POLICIES FOR MULTIPLE LICENSE COMPONENTS

A component with multiple licenses may have one or more that fail to comply with a policy rule. Some SCA tools will always trigger a violation in a case like this. However, Xray gives you more flexibility and can be configured to be more or less restrictive in this scenario.

For example, you can set up Xray so that it allows a component to go through and not trigger a violation if one or more of the component's multiple licenses are non-compliant with a policy rule. Xray can even send you a notification that one or more of the component's licenses fail to comply, without necessarily triggering a violation.

XRAY REPORTS

Xray enables you to take action from the results of its scans of your open source packages, builds and artifacts. Each report presents a snapshot of your OSS risk from a particular point-in-time, and displays information in a friendly way. The scope of each report is configured using filters, which you select based on your requirements.



Reports > Create New Report

Name*

Report Type: Vulnerabilities

Scope: Scope

- Vulnerabilities
- License Due Diligence
- Violations

Start by selecting Type & Scope

TYPES OF REPORT

The currently available types of reports are:

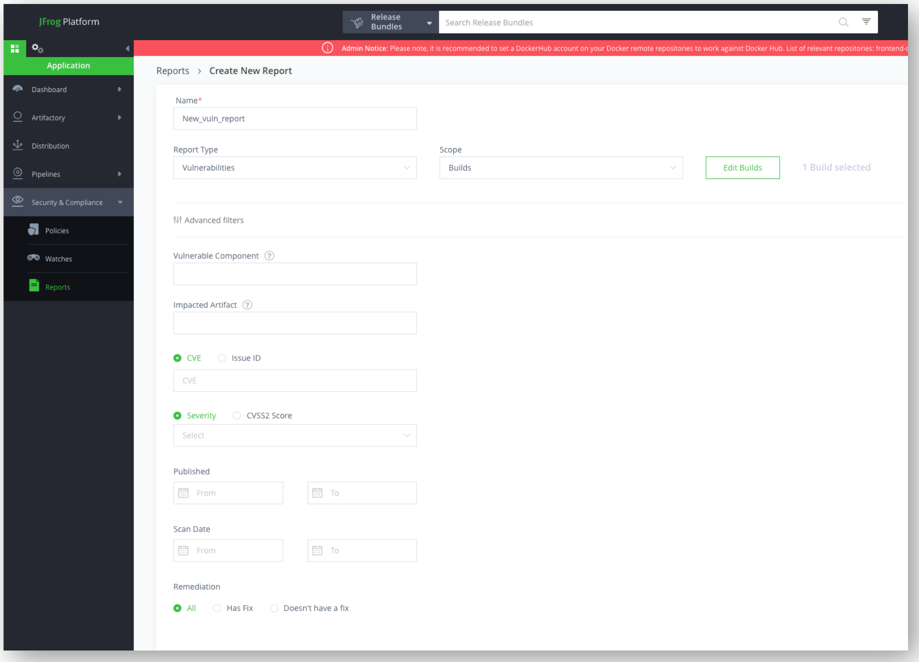
Vulnerabilities Report
Provides information about vulnerabilities in your artifacts, builds, and release bundles (software releases), as well as criteria such as the vulnerable component, CVE record, CVSS score, and severity.

License Due Diligence Report
Provides you with a list of components and artifacts, and their relevant software licenses, enabling you to verify that the components and artifacts you're using comply with your corporate license guidelines. It lists all of the license types associated with each component, as well as unknown and unrecognized licenses.

Violations Report
Provides you with information on security and license violations for each component found in the selected scope. Information includes type of violation, impacted components and artifacts, and severity.

Reports are configurable and you can customize your views through advanced filtering options. You can also create reports using patterns. For example you can capture a set of builds that all start with the same prefix 'docker_build_'. You can also choose to scan the most recent build only; or 3, 5, 10 or however many of the previous versions of the build.

The scope of a report can be configured by filtering by vulnerable components, impacted artifacts, scan dates, CVE ID or CVSS severity score. For remediation purposes, you can also configure reports to display ‘all vulnerabilities’, ‘ones with a fix’ or ‘doesn’t have a fix’.



Admin Notice: Please note, it is recommended to set a DockerHub account on your Docker remote repositories to work against Docker Hub. List of relevant repositories: frontend-

Reports > Create New Report

Name*: New_vuln_report

Report Type: Vulnerabilities

Scope: Builds

1 Build selected

Vulnerable Component:

Impacted Artifact:

● CVE ○ Issue ID

CVE

● Severity ○ CVSS2 Score

Select

Published: From To

Scan Date: From To

Remediation: ● All ○ Has Fix ○ Doesn't have a fix

You can view generated reports in the JFrog Platform UI. They are very detailed and contain a wealth of information and metadata about the discovered vulnerability, including:

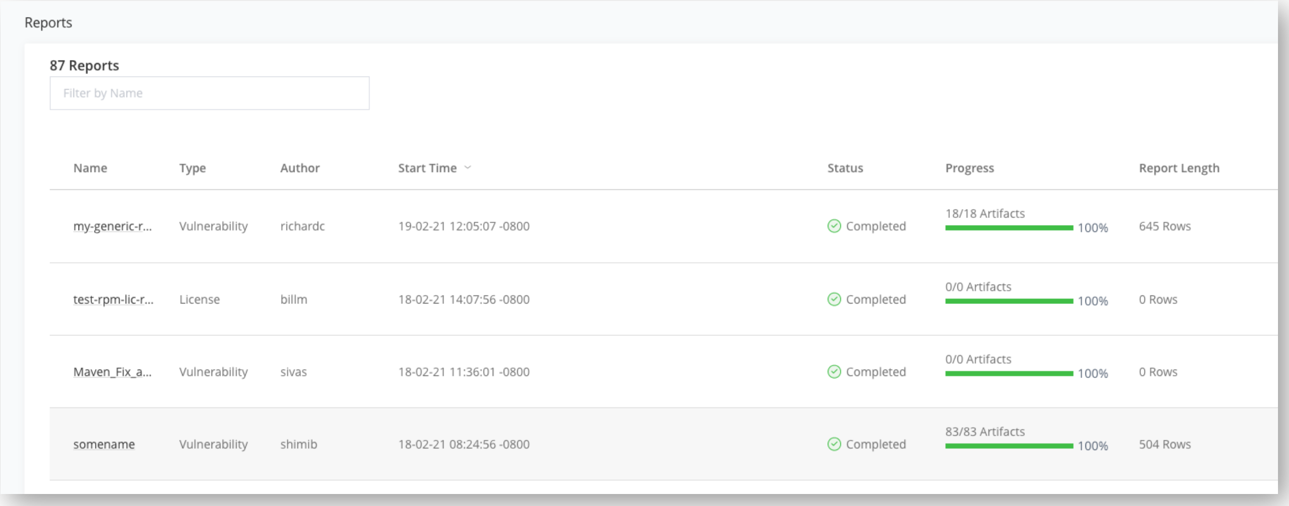
- the path to the vulnerability
- the fix version number (if any)
- granular metadata detailing package type, severity, CVSS score breakdown, publish date, scan date and more...

Reports can also be exported as PDF, JSON, or CSV files, or via a REST API to be further analyzed or used by other applications or tools in your organization.

Reports will always keep a record of the filter settings you chose, that way, you can audit what you see in the report against the scope that you defined, allowing you to maintain consistency across reports.

PERMISSION MANAGEMENT

A permissions role is included in the JFrog Platform to support setting user permissions. It enables permission restrictions on who can create, edit and share reports, and can be assigned at an individual user or group team level.



Reports

87 Reports

Name	Type	Author	Start Time	Status	Progress	Report Length
my-generic-r...	Vulnerability	richardc	19-02-21 12:05:07 -0800	Completed	18/18 Artifacts 100%	645 Rows
test-rpm-lic-r...	License	billm	18-02-21 14:07:56 -0800	Completed	0/0 Artifacts 100%	0 Rows
Maven_Fix_a...	Vulnerability	sivas	18-02-21 11:36:01 -0800	Completed	0/0 Artifacts 100%	0 Rows
somename	Vulnerability	shimib	18-02-21 08:24:56 -0800	Completed	83/83 Artifacts 100%	504 Rows

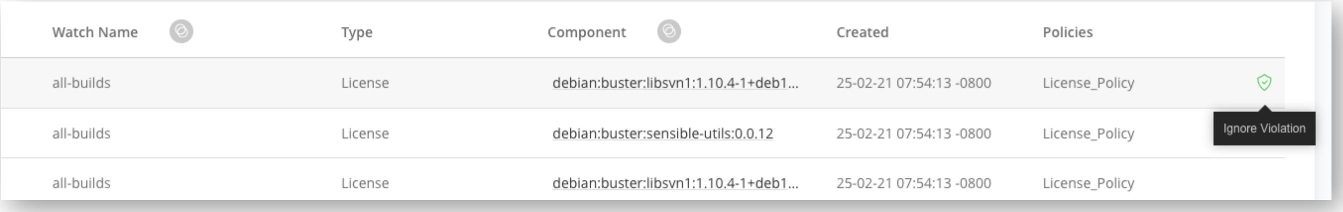
IGNORE RULES

> “We develop software for the (U.S.) Government and we use open source software… one of the downsides to open source is that there is not a lot of trust built into it. How do you go through and validate open source technologies? This is where JFrog Xray shines! Building Trust for the Government.”
>
> *Thomas Hastings,*
> *Software Engineer,*
> *Polaris Alpha*

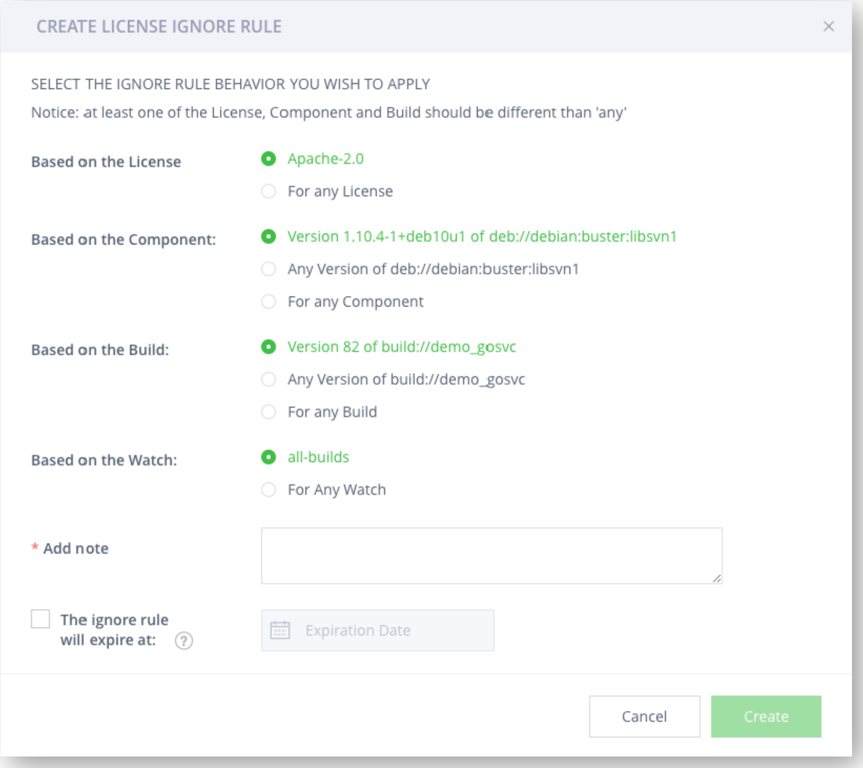
JFrog Xray Ignore Rules feature allows you to whitelist, ignore or accept security violation rules, in order to filter out unwanted violation noise. The rules can be approved to be ignored by different teams and users for reasons such as:

- You’re aware of the vulnerability and can protect against it.
- Your environment doesn’t meet the requirements for this violation.
- That vulnerability is not a show stopper and you’ll handle it later.
- Stop unimportant violations to fail your build or block downloads.

In the UI there is an icon that appears at the end of the data rows of builds and artifacts. Clicking the icon enables you to set up an ignore rule for that specific build or artifact.



Watch Name	Type	Component	Created	Policies
all-builds	License	debian:buster:libsvn1:1.10.4-1+deb1...	25-02-21 07:54:13 -0800	License_Policy
all-builds	License	debian:buster:sensible-utils:0.0.12	25-02-21 07:54:13 -0800	License_Policy
all-builds	License	debian:buster:libsvn1:1.10.4-1+deb1...	25-02-21 07:54:13 -0800	License_Policy



CREATE LICENSE IGNORE RULE ×

SELECT THE IGNORE RULE BEHAVIOR YOU WISH TO APPLY

Notice: at least one of the License, Component and Build should be different than 'any'

Based on the License

● Apache-2.0

○ For any License

Based on the Component:

● Version 1.10.4-1+deb10u1 of deb://debian:buster:libsvn1

○ Any Version of deb://debian:buster:libsvn1

○ For any Component

Based on the Build:

● Version 82 of build://demo_gosvc

○ Any Version of build://demo_gosvc

○ For any Build

Based on the Watch:

● all-builds

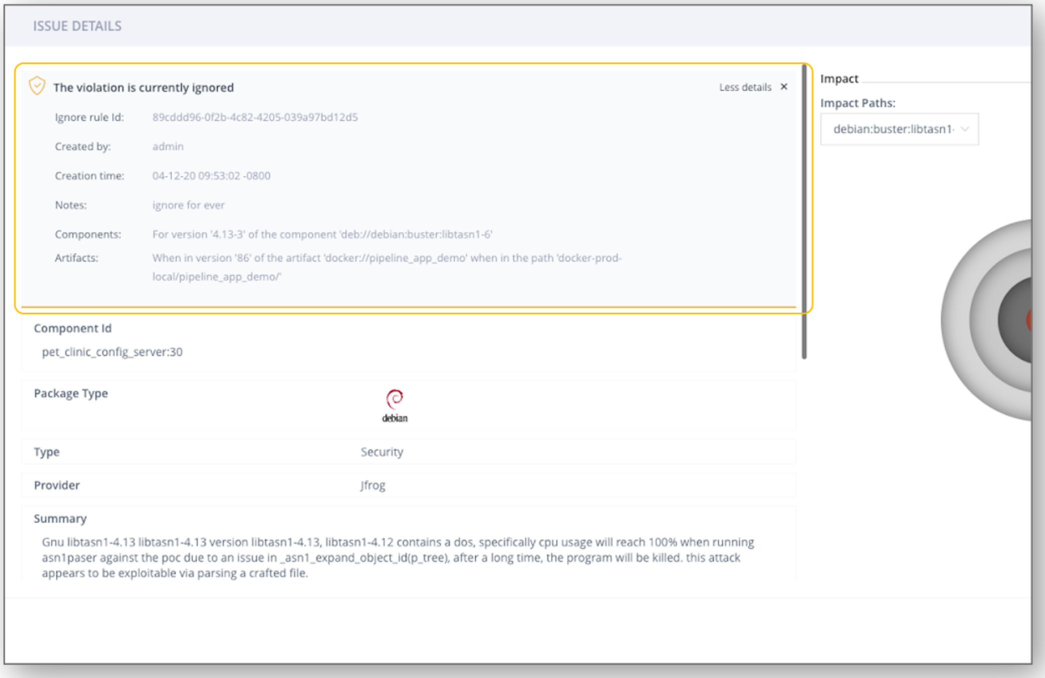
○ For Any Watch

* Add note

□ The ignore rule will expire at: Expiration Date

GRANULARITY OF IGNORE RULES

The Ignore Rules feature offers you broad flexibility and granularity, allowing you to ignore violations based on vulnerability/license, component, artifact or watch. As such, you can get very specific about what you want to ignore. For example, this can be set to a specific component, a specific license or a specific component version number.



ISSUE DETAILS

The violation is currently ignored

Ignore rule Id:	89cddd96-0f2b-4c82-4205-039a97bd12d5
Created by:	admin
Creation time:	04-12-20 09:53:02 -0800
Notes:	ignore for ever
Components:	For version '4.13-3' of the component 'deb://debian:buster:libtasn1-6'
Artifacts:	When in version '86' of the artifact 'docker://pipeline_app_demo' when in the path 'docker-prod-local/pipeline_app_demo/'

Component Id

pet_clinic_config_server:30

Package Type	debian
Type	Security
Provider	Jfrog

Summary

Gnu libtasn1-4.13 libtasn1-4.13 version libtasn1-4.13, libtasn1-4.12 contains a dos, specifically cpu usage will reach 100% when running asn1paser against the poc due to an issue in _asn1_expand_object_id(p_tree), after a long time, the program will be killed. this attack appears to be exploitable via parsing a crafted file.

Impact

Impact Paths: debian:buster:libtasn1-

TIME BASED IGNORE

Ignore rules can be set up to run for a certain period of time, meaning, for example, that you can ignore certain violations for 3 weeks while you ramp up development, after which they will again be enforced, at a time when you're moving to the next stage of development or a new environment.

PREMIUM VULNERABILITY DATABASE

VulnDB

A key component of an SCA tool is its vulnerability database. Many SCA tools only utilize the NVD database as their source for vulnerabilities. This can limit their effectiveness, because newly discovered vulnerabilities often aren't immediately recorded in the NVD. This could cause an application to ship to production with a known vulnerability that hackers are actively trying to exploit.

That's why SCA tools must have a premium vulnerability database that includes newly discovered vulnerabilities the moment they're disclosed. JFrog Xray features such a vulnerability and license database: Risk Based Security's VulnDB.

VulnDB stands apart not only for its timeliness, but also for going above and beyond what's expected from a vulnerability database. For example, as of June 2021, VulnDB had all NVD vulnerabilities, plus more than 83,000 additional ones. Its entries are data-rich, including detailed vulnerability source information, extensive references, links to proof of concept code, and solutions. VulnDB also offers vendor and product ratings, so you can assess their risk, and it monitors more than 2,000 third-party libraries for vulnerabilities.

VulnDB's breadth, depth, and timeliness of vulnerability intelligence coupled with JFrog Artifactory's extensive metadata knowledge and database of software packages offer a powerful solution for quickly identifying and mitigating vulnerabilities across any DevOps pipeline.

"Software Composition Analysis Tools were voted #1 most important DevSecOps tools for ensuring application security."

Jim Mercer, Research Director DevOps & DevSecOps, IDC 2021-04-13 - DevSecOps Adoption, Techniques, and Tools Survey

CUSTOMIZE XRAY WITH PRIVATE VULNERABILITY AND LICENSE DATA

Some organizations need to add their own custom vulnerability data to their SCA tool database. Xray has that covered as well. It lets you specify your own data to detect additional issues in your binaries before they can reach production. With this capability, Xray covers use cases such as these:

- You may consider something to be a vulnerability that Xray's data sources do not
- You may have access to information about software components not included in Xray's data sources
- You may have private (not OSS) libraries for which you have tracked vulnerabilities and want to include them in your Xray security scans

With a custom integration, Xray can look at any external source for additional information about vulnerabilities and licenses. Just like it does for the NVD and VulnDB databases, Xray will apply the security and license compliance policies you have specified to your software components.

RED HAT SECURITY SCANNING CERTIFICATION

JFrog Xray has been certified by Red Hat as part of their Red Hat Partner Vulnerability Scanner Certification Program. Being certified ensures that the security vulnerability and license compliance data identified by JFrog Xray is accurate and consistent with expected results for Red Hat packages, enabling accurate risk assessment based on trusted, certified sources.

This means that enterprises using RPM packages can confidently use the JFrog Platform as their DevSecOps platform. In addition to the Vulnerability Scanner Certification for Xray, JFrog has also been certified for:

Red Hat Certified OpenShift Operator (for JFrog Artifactory and JFrog Xray) to enhance customer installation and automation

Red Hat Certified UBI Container Image (for JFrog Artifactory, the industry's only universal package manager and container registry) for additional assurance of greater reliability, security and performance of the underlying operating system that Artifactory runs on

Red Hat

Certified Technology

Vulnerability Scanner

SUMMARY

The combination of massive usage of open source components, a wide variety of security issues and vulnerabilities, and the eagerness of malicious hackers to exploit security flaws, places a heavy burden on DevOps teams in the organization.

As the only Software Composition Analysis solution that provides native integration with JFrog Artifactory and the JFrog Platform, deep recursive scanning, continuous and detailed impact analysis, and custom API-driven automation, JFrog Xray offers the most comprehensive treatment of security vulnerabilities and license compliance issues in open source and commercial software available today.

ABOUT JFROG

JFrog is on a mission to enable continuous updates through liquid software that helps empower developers to deliver high-quality applications that securely flow to end-users with zero downtime. Our solutions meet your business model needs and support on-premises, cloud, multi-cloud and hybrid deployments.

JFrog solutions are used by more than 5,800 customers—from startups to large enterprises—who depend on JFrog to manage their binaries for their mission-critical applications. This includes more than 75% of the Fortune 100 companies such as Amazon, Facebook, Google, Netflix, Uber, VMware, and Spotify who put their trust in JFrog.

TRANSFORM YOUR DEVOPS INTO A FULLY AUTOMATED AND INTEGRATED DEVSECOPS PLATFORM

Webinar: DevSecOps Best Practices with the JFrog Platform
Webinar: DevSecOps for Kubernetes-based Applications
Get Started with JFrog Xray For FREE