JFrog Xray

Universal Component Analysis

Making DevOps Omniscient

White Paper

November 2016 | www.jfrog.com

Contents

Executive Summary

The use of open source components continues to rise at an ever increasing rate, but most companies lack the formal infrastructure to manage OSS usage and do not have adequate security practices in place to mitigate the risks inherent with using open source components. The massive usage of open source components in commercial systems has opened new avenues for cybercrime which may carry an annual cost of over $400 billion globally. But cybercrime is not the only vulnerability that companies should be concerned with. Open source license compliance, outdated and deprecated components, and post-production bugs are just some of the vulnerabilities that must be vigilantly avoided, and immediately remediated when discovered.

What is JFrog Xray?

JFrog Xray works with JFrog Artifactory to perform universal artifact analysis, and reveal a variety of issues and vulnerabilities at any stage of the software application lifecycle. By scanning binary artifacts and their metadata, recursively going through dependencies at any level, JFrog Xray provides unprecedented visibility into vulnerable artifacts lurking anywhere in your organization. Xray's interface with Artifactory gives it the exclusive advantage of combining any number of data feeds with the exhaustive metadata stored within Artifactory to detect different issues and vulnerabilities without needing access to source code. JFrog Xray is also fully automated through a rich REST API that lets it integrate with a CI/CD pipeline and allows other binary analysis tools to build on its unique capabilities.

How does JFrog Xray Protect You?

JFrog Xray is the only tool that protects your development and production systems through the unique combination of:

- **Deep Recursive Scanning:** the ability to recursively drill down and analyze even the smallest binary artifacts that affects your software.
- **Continuous Analysis:** the ability to continuously scan and analyze existing artifacts, even those long since deployed to production, and provide alerts for just-discovered issues and vulnerabilities.
- **Custom API-Driven Automated Analysis:** the ability to add custom automated analyses through an open REST API.

The Xray-Artifactory Edge

As a complementary product to JFrog Artifactory, JFrog Xray has access to the wealth of metadata Artifactory stores which, combined with deep recursive scanning, puts Xray in a unique position to analyze the relationships between binary artifacts and provide radical transparency into your component architecture to reveal the impact that a vulnerability in one component has on any other.

What Does JFrog Xray Give You?

Custom Vulnerability Dashboard: Detailed reports on all vulnerabilities you are interested in.
Impact Analysis: Clear visualization of how a vulnerability in one component affects all others.
Analysis Filters: Focus on the most relevant scope based on different parameters.
Ecosystem Integration: Fully automated vulnerability analysis and management by integrating with your CI/CD pipeline.

JFrog Xray's tight integration with JFrog Artifactory places it in a unique position to take full advantage of the exhaustive metadata Artifactory stores. By identifying the relationships between binary artifacts in an organization's repositories JFrog Xray understands how a vulnerability in one component impacts all the others. As the only tool that combines deep recursive scanning, continuous analysis, and custom, API-driven automation, JFrog Xray offers the most comprehensive treatment of issues and vulnerabilities in open source and commercial software available today.

Introduction

The most recent yearly Future of Open Source survey shows that the use of open source components continues to rise at an ever-increasing rate, but also that most companies lack the formal infrastructure to manage OSS usage and do not have adequate security practices in place to mitigate the risks inherent in using open source components. Docker may be an exception. The company's growth and the corresponding massive adoption of Docker registries has spawned a slew of "Security for Docker" companies mushrooming on the scene. But Docker is virtualization technology built on containers, and containers are only there to provide a runtime environment for your real business logic which may be written using any other technology available on the market. So all those companies using Docker also download over 250,000 npm packages from npmjs.org daily, over half a billion Maven packages from JCenter monthly and billions of other components from the variety of public registries available today for all major packaging formats.

The massive use of open source components in commercial systems has opened new avenues for cybercrime which, according to some reports, carries an annual cost of over $400 billion globally. A good example is Heartbleed which exposed many of the internet's web servers to theft of sensitive data such as private keys and passwords, and incurred costs that, for some companies, ran into millions. But cybercrime is not the only vulnerability in open source components that organizations should be concerned with. Today, a software vendor can attach one of over a hundred different licenses to an open source component published on a public repository for free download. Companies are not always willing to comply with the terms set forth in some of these licenses and must ensure that none of the components in their products use them. On a more basic level, even components that are authorized for use in a company can get outdated as new versions are released. Bugs and performance issues may be reported for specific versions of components long after they have been deployed to production systems, and components may get deprecated as they continue to fester deep down in old legacy products. All of these use cases are issues and vulnerabilities that responsible software development organizations cannot afford to miss. Without vigilant monitoring, production systems are exposed, but no less important, organizations also put their whole development engines at risk of grinding to a halt if these vulnerabilities turn up on the development floor.

What is JFrog Xray?

JFrog Xray works with JFrog Artifactory to perform universal artifact analysis, and reveal a variety of issues and vulnerabilities at any stage of the software application lifecycle. Through its unique interface with Artifactory, JFrog Xray can look deep into software components and their metadata in a variety of package formats including Docker, npm, Debian, NuGet, JAR and more. By scanning binary components and their metadata, recursively going through dependencies at any level, JFrog Xray provides radical transparency into your software architecture showing how any one artifact affects another, and gives you unprecedented visibility into vulnerable artifacts lurking anywhere in your organization from your development floor to your production datacenter.

JFrog Xray's interface with JFrog Artifactory gives it the exclusive advantage of combining any number of data feeds with the exhaustive metadata stored within Artifactory such as build information, deployment information, QA status and more. This combination empowers JFrog Xray with unique capabilities for detecting security vulnerabilities, outdated component and dependency versions, license monitoring and more, through the analysis of binary artifacts without needing access to source code.

Unlike traditional binary analysis tools, JFrog Xray is a fully automated product with a rich REST API. This enables integration of JFrog Xray with your CI/CD pipeline and allows other binary analysis tools, seeking to perform security audits and other analyses, to build on Xray and its unique capabilities.

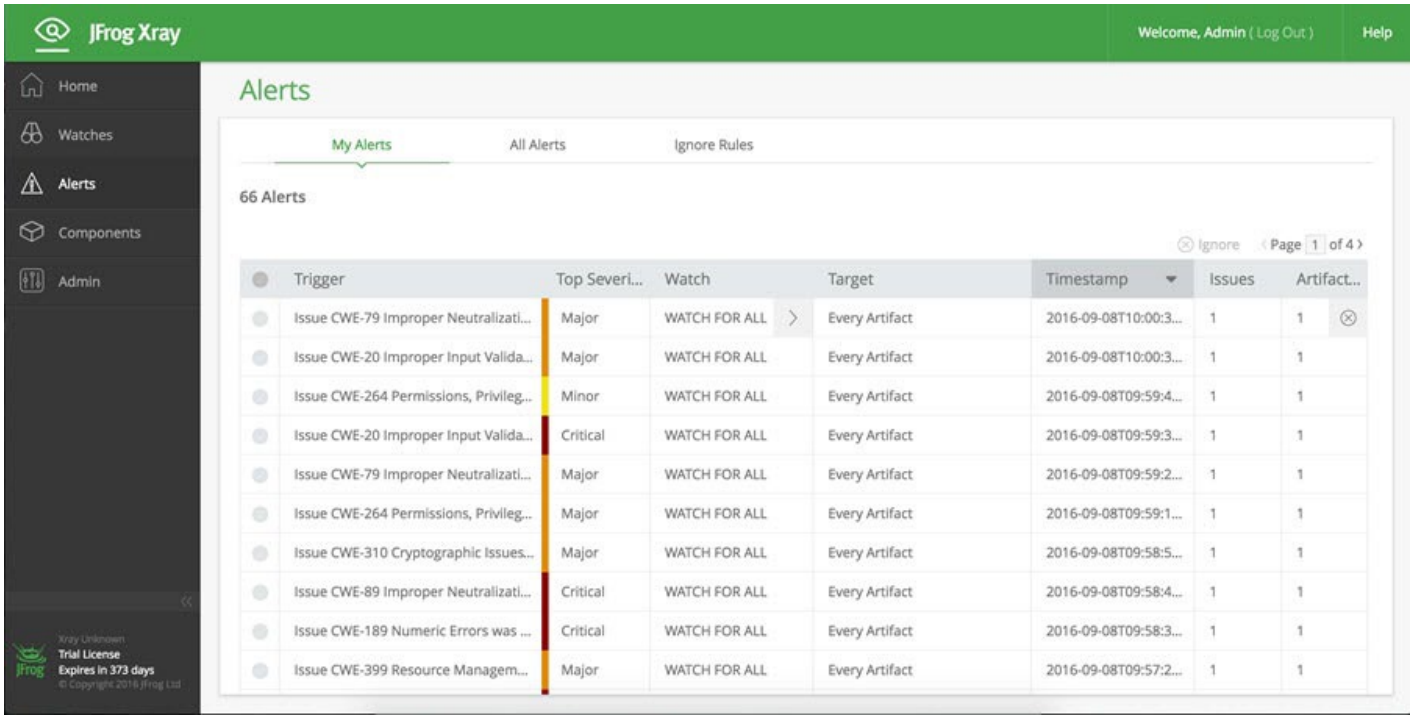
How Does JFrog Xray Protect You?

JFrog Xray is the only tool that protects your development and production systems through the unique combination of:

- **Deep Recursive Scanning:** the ability to recursively drill down and analyze even the smallest binary component that affects your software.
- **Continuous Analysis:** the ability to continuously scan and analyze existing artifacts, even those long since deployed to production, and provide alerts for just-discovered issues and vulnerabilities.
- **Custom API-Driven Automated Analysis:** the ability to add custom automated analyses through an open API.

Deep Recursive Scanning

JFrog Xray starts with your primary software component, and then recursively drills down to identify its dependencies, and then the dependencies' dependencies, and so on down to any level, until every single artifact that is a part of your software, whether directly or indirectly, has been identified. Xray supports most major package format in use today including Docker, Debian, RPM, NuGet, JAR files, Npm, PyPI and Bower. In fact, as an open and flexible package-agnostic tool, Xray can accommodate new formats that may come on the scene from time to time and provide the same level of deep recursive scanning as with currently available package formats.



Once all components and dependencies have been identified, Xray cross-references them with any number of feeds and databases of known vulnerabilities, and alerts you if any component compromises your software.

Continuous Analysis

Issues and vulnerabilities in your software may be identified at any time, from initial development phases through to its lifecycle in your production systems. Whether these are security flaws that make headlines, or a simple deprecation of some open source component that you are using (either directly or through one of your dependencies), they can have serious consequences for your business. Developer builds may start failing, your CI/CD pipeline may break, or you may realize that sensitive information in your production systems is exposed.

One of the outcomes of JFrog Xray's deep recursive scanning is a map matching the components and dependencies of your software to the databases and live feeds to which Xray is connected. The continuous analysis of live feeds being matched up with your components means that any change in a component, or a new report of a vulnerability or other issue can generate an immediate alert to the right authority within your company regarding the suspicious component. Whether the alert is issued as an indication on a dashboard or an email to the right person, or a notification in your internal messaging system, this virtually immediate response lets you manage the issue with the minimal impact possible.

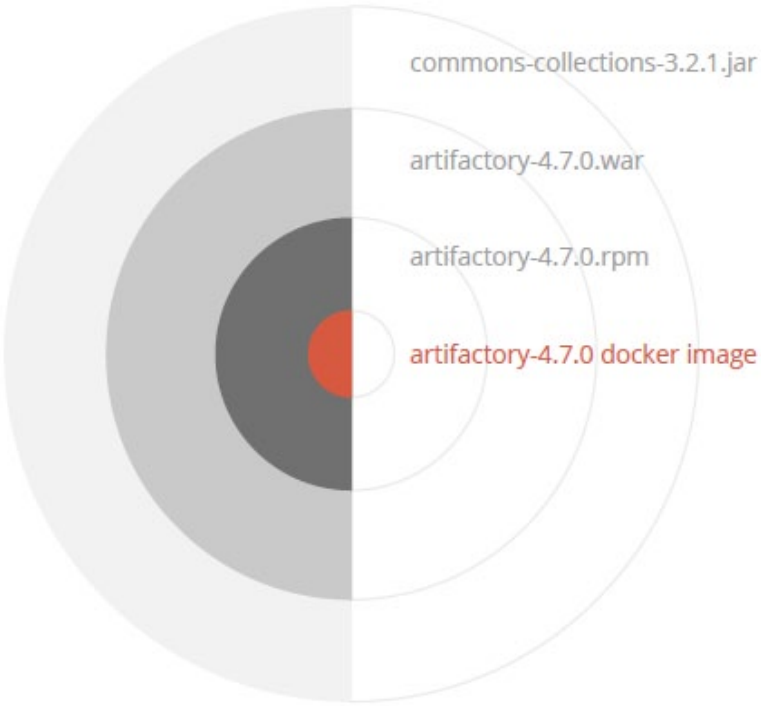
Custom API-driven Automated Analysis

JFrog Xray comes with a set of analyses built-in including monitoring for security vulnerabilities, open source license compliance, component version changes and more. However, different organizations may require their 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 JFrog 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.

The Xray-Artifactory Edge

JFrog Xray is tightly coupled with JFrog Artifactory, and as a complementary product, has access to the wealth of metadata Artifactory stores. Artifactory indexes not only standard package metadata (such as those found in *maven-metadata.xml* or *.nuspec* files), but also custom and package properties, exhaustive build information, deploy information and more. This is much more than stateless metadata on specific binary signatures; it is metadata that reveals the context of the binary artifact within the organization, and its history in the software development lifecycle. JFrog Xray's deep recursive scanning combined with the indexed metadata in JFrog Artifactory, as a system-of-record binary repository, puts Xray in a unique position to analyze the relationships between binary artifacts in an organization and understand the impact that a vulnerability in one component has on any other.



Through tight integration with Artifactory, and access to the exhaustive metadata that it indexes, Xray is in a unique position to analyze the relationships between binary artifacts in an organization and understand the impact that a vulnerability in one component has on any other.

What Does JFrog Xray Give You?

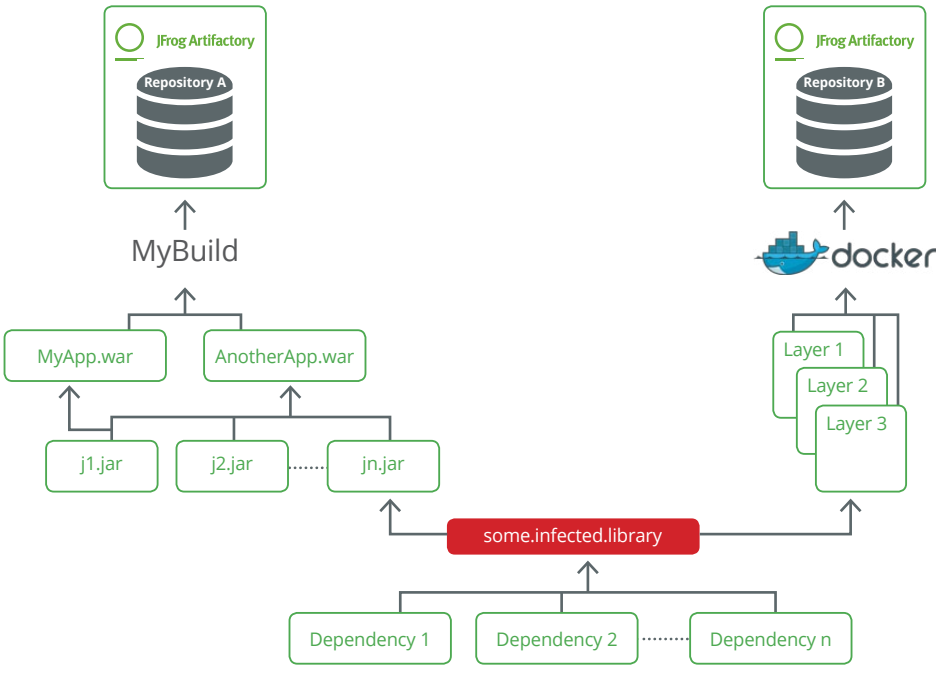
What you really get from JFrog Xray is peace of mind, knowing that Xray is continuously monitoring and analyzing your software to make sure it is safe from cyber-attack, outdated or deprecated components and performance issues, alerting you according to any triggers relevant to your organization.

Custom Vulnerability Dashboard

Get detailed reports on all vulnerabilities you are interested in. Customize your dashboard to display security vulnerabilities, OSS license infringements, quality or performance issues, new and deprecated version alerts and more. You can even define alerts that are triggered according to a custom set of criteria to display on the dashboard.

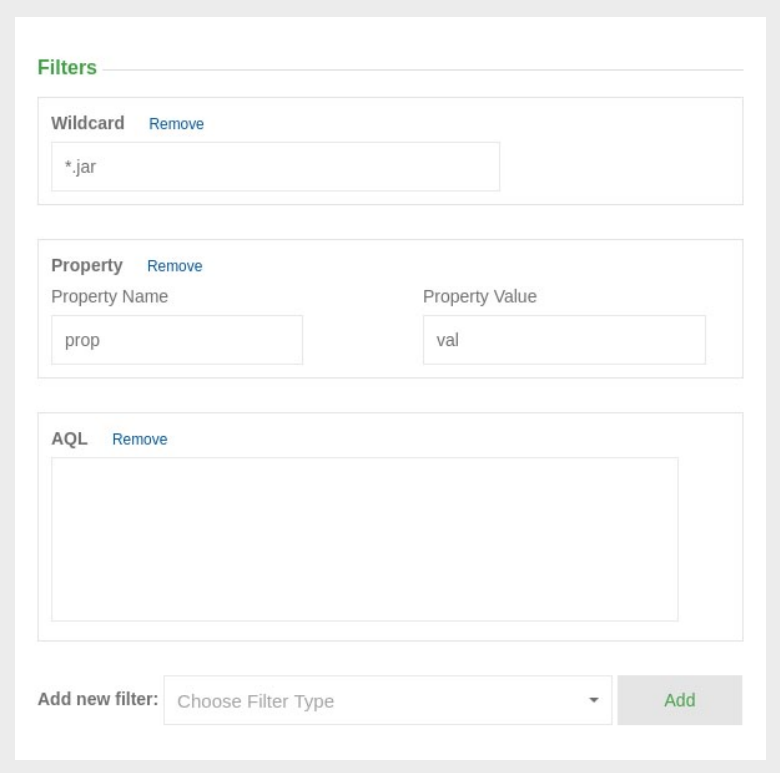
Impact Analysis

Xray's deep recursive scanning gives it a complete picture of the relationships between all the components in your product. Once a component is identified to suffer from an issue or vulnerability, Xray understands the impact it has on other components and displays a graph showing the relationship between the vulnerable component and all others that are affected. The component relationships may be in the context of binary artifacts, builds, or deployments that are connected to the infected component. Impact analysis can be triggered manually according to internal policies, or automatically in response to an event. Xray connects to a configurable set of data sources which can trigger an impact analysis run based on vulnerabilities reported by any of the feeds.



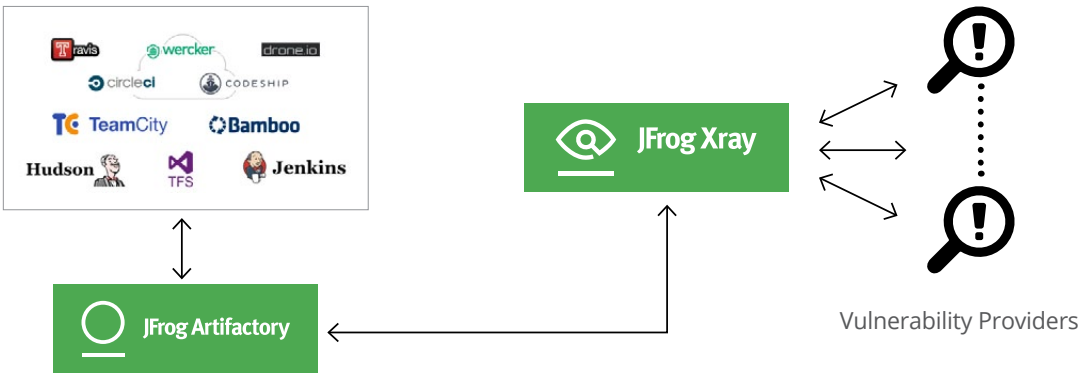
Analysis Filters

An organization's artifact repositories can reach terabytes and even petabytes in size. Running a complete analysis may be resource intensive and, in any case, may generate an impact analysis graph that is too big and detailed to be really useful. To help an administrator focus on more important artifacts, impact analysis can be narrowed down to the relevant scope based on parameters such as package type, repository and path, specific property values, artifact age, build parameters and more. For example, you can configure JFrog Xray to focus on Docker registries in Artifactory and find all images in the production repository with a "deployed" property set to "true" that are impacted by any security vulnerability.



Ecosystem Integration

Through its REST API, Xray is an integral part of your CI/CD pipeline. Configure different alerts depending on the outcome of analyses, or even stop your build if any components are flagged with vulnerabilities.



Summary

The combination of massive usage of open source components, a wide variety of issues and vulnerabilities, and the eagerness of malicious parties to exploit security flaws, places a heavy burden on responsible software development organizations to be constantly vigilant. While many tools are available on the market to mitigate these vulnerabilities, most of them are either restricted to a limited set of package formats or to periodic scanning of components, neither of which provide an adequate solution. Vulnerabilities may be discovered in components of any package format, and at any time, both in development and production systems. JFrog Xray's tight integration with JFrog Artifactory places it in a unique position to take full advantage of the exhaustive metadata stored in Artifactory. By identifying the relationships between binary artifacts in an organization's repositories JFrog Xray understands what impact a vulnerability in one component will have on the others. As the only tool that combines deep recursive scanning, continuous analysis, and custom, API-driven automated analysis, JFrog Xray offers the most comprehensive treatment of issues and vulnerabilities in open source and commercial software available today.