NIST SP 800-218
JFROG PLATFORM COMPLIANCE GUIDE

November 2024 | Ver. 1.3

JFrog

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Abstract
Government agencies must release software reliably, confidently, and on schedule to continually empower public servants and better serve citizens with modern apps and digital services. Simultaneously, they also need to ensure that software is secure and compliant to prevent cyber-attacks and meet IT transformation goals. To accomplish these goals, agencies need DevSecOps – the orchestration and automation of development, security, and operations processes across the software development pipeline.

Product Showcase: JFrog provides an end-to-end Software Supply Chain Platform for automating, managing, securing, distributing, and monitoring your containers, binaries and artifacts, metadata, and configurations - everything that goes into software development - as they advance from source code to production. It unifies developers, operations, and security teams to safeguard the entire software supply chain in a holistic, self-hosted, cloud, multi-cloud, or hybrid platform. Additional information on the JFrog Platform can be found in Appendix A.

KEY CAPABILITIES

- Secured Software Development Framework (SSDF)
- Secure Software Development Lifecycle (SSDLC)

TARGET

- Federal Service Integrators (FSI)
- Government Direct (Federal Civilian, Defense, Intelligence, Critical Infrastructure and other)

The objective of this Compliance Product Guide is to accelerate customer understanding of effective DevSecOps processes and best practices against industry standards and frameworks. As part of this analysis, The JFrog Platform was evaluated against NIST SP 800-218, Secured Software Development Framework (SSDF) 1.1 from a product capability perspective. This document maps specific JFrog functionality to NIST 800-218 recommendations.

SSDF practices can be incorporated into each phase of the software development lifecycle (SDLC), building a security practice that binds people, processes, and technology. This practice helps core software developers, mission application development, and core DevSecOps community to implement a rigid security framework from finding vulnerabilities at early stages (aka shift left) to discovering new undetected vulnerabilities (zero days) and preventing future recurrences.

JFrog plays a critical role in securing each software supply chain, including the SDLC, from code creation to artifacts in the runtime by three core tenets – Core DevOps, Security, and IoT. JFrog's Advanced Security capabilities protect the application, its micro-services, and the cloud infrastructure configurations from causing exposures in production.

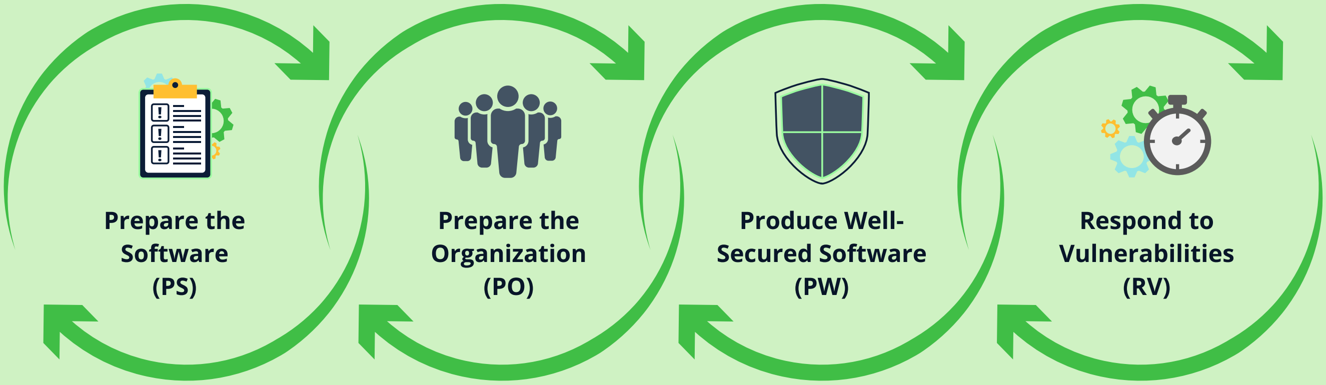
PLATFORM COMPLIANCE MAPPING

The NIST SP 800-218 standard encompasses security, technology, organizational processes, and people. It also supplies the requirements found in other standards, including the CMMC, FEDRAMP, DFARS, and FISMA. Organizations providing DevSecOps capabilities to the US Public Sector need to practice and implement security guidelines recommended in this SSDF Framework. A thorough understanding of security controls and their applicability will accelerate secured software development with the outcome of secured software with fewer vulnerabilities.

The JFrog Platform is a unified DevSecOps solution that provides universal artifact management, advanced security and compliance, fast and trusted releases, end-to-end automation and IoT device management. These solutions support 800-218 and are delivered as cloud, multi-cloud, self hosted, air gapped and hybrid deployments. Specific mapping is shown below and in more detail in Appendix A.

NIST
National Institute of
Standards and Technology

NIST SP 800-218



Prepare the Software (PS)

Prepare the Organization (PO)

Produce Well-Secured Software (PW)

Respond to Vulnerabilities (RV)

NIST SP 800-218 (SSDF) OVERVIEW

WHAT IS NIST SP 800-218?

NIST 800-218 describes a set of fundamental, sound practices for secure software development called the Secure Software Development Framework (SSDF). Organizations should integrate the SSDF throughout their existing software development practices, express their secure software development requirements to third-party suppliers using SSDF conventions, and acquire software that meets the practices described inthe SSDF.

HOW CAN 800-218 BE IMPLEMENTED?

The SSDF defines only a high-level subset of what organizations may need to do, so organizations should consult the references and other resources for additional information on implementing the practices. Not all practices apply to all use cases; organizations should adopt a risk-based approach to determine what practices are relevant, appropriate, and effective to mitigate the threats to their software development practices.

NIST 800-218 AND EO14028

The President's Executive Order (EO) on "Improving the Nation's Cybersecurity (14028)" issued on May 12, 2021 [EO14028], charged multiple agencies – including NIST – with enhancing cybersecurity through a variety of initiatives related to the security and integrity of the software supply chain.

Section 4 of the EO directed NIST to solicit input from the private sector, academia, government agencies, and others and to identify existing or develop new standards, tools, best practices, and other guidelines to enhance software supply chain security. Many subsections from Section 4e of the EO map to SSDF practices and tasks that can help address each subsection as part of a risk-based approach.

PRODUCT OVERVIEW

PRODUCT SUMMARY

The JFrog Platform is an end-to-end Software Supply Chain Platform to control and secure software development from source code to runtime. This creates a single source of truth for secure application software development.

JFrog provides an end-to-end Software Supply Chain Platform for automating, managing, securing, distributing, and monitoring your containers, binaries and artifacts, metadata, and configurations - everything that goes into software development - as they advance from source code to production. It unifies developers, operations, and security teams to safeguard the entire software supply chain in a holistic, hybrid, multi-cloud platform.

CORE DEVSECOPS

- With JFrog Artifactory at the core, curate a secure, single source of record for DevSecOps and the entire software supply chain. Update software across any environment, with OOTB support for over 35 technology types.

- Next Gen CI/CD JFrog Pipeline, Automate, orchestrate, and attest to your pipeline's integrity across cloud and self-managed environments.

- Updates from Code to Devices: Create your own private, fast, secure, hybrid distribution model for updates all the way to the edge.

- Support tool choices and deployment targets across self-hosted, cloud, and multi-cloud environments without sacrificing speed or availability.

JFROG SECURITY

- JFrog Xray provides a Software Composition Analysis (SCA) solution for Identifying and resolving security vulnerabilities and license compliance issues in your open-source dependencies.

- Malicious Package Detection: Discover and eliminate unwanted or unexpected packages, using JFrog's unique database of identified malicious packages. The database is sourced with thousands of packages identified by our research team in common repositories alongside continuously aggregated malicious package information from global sources.

- Operational risk policies to block undesired packages: Discover and eliminate unwanted or unexpected packages, using JFrog's unique database of identified malicious packages. The database is sourced with thousands of packages identified by our research team in common repositories alongside continuously aggregated malicious package information from global sources.

- Shift as far left as possible with JFrog developer tools: Scan your packages for security vulnerabilities and license violations early in your SDLC with developer-friendly tools. See vulnerabilities with remediation options and applicability context right inside your IDE. Automate your pipeline with our CLI tool and perform dependency, container, and on-demand vulnerability scans. Scan early to minimize threats, reduce risk, fix faster, and save costs.

- SBOM and Regulatory Compliance: Generation of SPDX, CycloneDX, and VEX Standard-format SBOMs. Comprehensive SBOM accuracy with binary analysis going well beyond standard metadata regulatory requirements by fully monitoring and controlling vulnerabilities across the SDLC. Keep Malicious package database and automate the publication of SBOM and associated CVEs when needed.

- FOSS Compliance and License Clearing: Reduce risk by monitoring, controlling, and validating shipping of approved licenses. Automate license-clearing processes, ensuring development teams are using fully approved licenses.

- Efficiently find and fix security issues across your entire DevSecOps pipeline, including exposed secrets, OSS vulnerabilities, IaC and container security, and open-source license issues, with automation, contextual analysis, and enhanced remediation.

Federal Agencies and other enterprises continue to face security threats such as cyberattacks, ransomware, security vulnerabilities in legacy software, and more. NIST SP 800-218 guides on building strong secured software using the practices and guidelines specified in the SSDF framework.

Adopting the JFrog DevSecOps Platform helps Federal Agencies and other enterprises achieve compliance with the NIST SP 800-218 and other security regulations. JFrog Security helps these companies with advanced security protection in place to defend against ever-evolving cyber threats.

The Mission Critical Piece of Your Development Infrastructure

Technology Agnostic

Create across public clouds and self hosted data centers using any technology you need.

Ultimate Scalability

Resilient and performant even with thousands of users and petabytes of data, with HA and failover capabilities.

Secure Automation

Enterprise IAM and role based access controls meet built-in application security and anti-tampering.

APPENDIX A: SECURITY REQUIREMENTS MAPPING

REQ.ID	Security Requirement	Platform Capability
PO.1	Define Security Requirements for Software Development	The JFrog Platform manages all the component vulnerability policies, FOSS licensing policies, and FOSS operational policies. Policies can be applied to every SDLC process across the organization, from providing early insights to developers (in the IDE) up to the extent of blocking releases, all in a fully automated fashion.
PO.2	Implement Roles and Responsibilities	The JFrog Platform has a flexible Roles mechanism to support the central administration of security policies with full delegation to teams/projects where needed. It also features Role Based Access Controls (RBAC) ensuring access to certain capabilities only resides with the correct teams & roles.
PO.3	Implement Supporting Toolchains	The JFrog Platform allows for the automation of security functions across every stage of the SDLC. These include other Security and DevOps tools ensuring automated implementation of security practices across tool chains and processes. Open APIs, built-in integrations, support for custom plugins, and the JFrog CLI ensure automation securely across tool chains for all SDLC phases.
PO.4	Define and Use Criteria for Software Security Checks	The JFrog Platform provides the ability to define security policies to evaluate artifacts, builds, or container images across all stages of the SDLC. These policies can be in the form of security vulnerabilities, operational risk, license compliance and malicious packages. There is also the ability to curate open-source components even before they enter the development environment. During the development lifecycle, evidence is gathered and stored providing auditability and security at each stage of the SDLC.

REQ.ID	Security Requirement	Platform Capability
PO.5	Separate and protect each environment involved in software development	The JFrog Platform can be deployed in its own environmental boundary (Network, access, etc.). providing uniform access control and authorization to services deployed on the platform. It is implemented as microservices guaranteeing a level of trust and boundaries defining access & authorization for inter-services, and also provides a uniform access and authorization model across the services. The Platform has “Projects” that isolate resources such as artifacts, builds, and repositories with role based access control (RBAC). There is a request router that routes the request to the right component - routing requests to services such as Artifactory, Access, Replicator, Event, Observability, Integration, Metadata and JFrog Connect. There are multiple router services that connect to JFrog Xray and JFrog Distribution. JFrog protects each deployment environment to ensure support, build, test, dev, production are protected by access such as user authentication - multi-factor, SSO, SAML - to support backend identity manager and provides fine grained permission and role based access control to resources (Artifacts, builds) deployed to end users and groups. Each instance has well defined network segmentation that isolates deployment (Dedicated VPC, network adapter) with well-defined ingress and egress ports, security groups (CSP) defined and configured access control. It can also be deployed into an air gapped network with complete zero ingress and egress. Each platform access (ingress) can be protected using signed certificates, and are protected using TLS certificates for each service described above.

REQ.ID	Security Requirement	Platform Capability
PO.5	Separate and protect each environment involved in software development	When you enable TLS, all communications to the JFrog Platform are required to use TLS including service-to-service communication within the platform. In the Platform, Access acts as the CA and signs the TLS certificates used by all the different JFrog Platform services. Deployments provide a service account to minimize the human interaction when running CI pipelines to build an artifact from a tool chain perspective. The JFrog Platform also provides continuous monitoring and uniform log and metrics for each deployment. Log aggregation and metrics emits could be sent to external engines for building dashboards and for day 2 operations. Audit trail log records are produced which registers all operations related to users, groups and permissions to allow auditing and tracking capabilities that allow you to enforce security policies in your organization. JFrog ensures the authenticity of your builds/binaries as they progress through the SDLC. JFrog Xray & Advanced Security protects artifacts continuously by monitoring for security vulnerabilities and deep contextual scanning capabilities. (More examples on other controls are defined below.) The Distribution service provides the ability to create an immutable "Release Bundle (RB)" which is comprised of resources (artifacts, metadata, etc.) which is signed/encrypted using GPG keys and then distributed to the target instance. Once distributed, upon receiving the RB, the target JFrog Platform verifies the atomicity of the bundle by comparing the hashes and thus preserves/secures the resources.

REQ.ID	Security Requirement	Platform Capability
PS.1	Protect All Forms of Code from Unauthorized Access and Tampering	The JFrog Platform allows organizations to create immutable release assets immediately from the output of the build process. In addition, it stores artifact metadata, configuration files and creates buildInfo comprising the metadata required to recreate any build. Any changes to artifacts or repositories are stored and noted for auditability. It also features RBAC ensuring access to certain capabilities only resides with the correct teams and roles. JFrog Artifactory uniquely stores artifacts using checksum-based storage. This avoids storing duplicate files, saves storage and optimizes the management and protecting of binaries. The Distribution service provides the ability to create an immutable “Release Bundle” (RB) which is comprised of resources (artifacts, metadata, etc.) which is signed/ encrypted using GPG keys and then distributed to the target JFrog Platform. Once distributed, upon receiving the RB, the target JFrog Platform verifies the atomicity of the bundle by comparing the hashes and thus preserves/secures the resources.
PS.2	Provide a Mechanism for Verifying Software Release Integrity	The JFrog Platform ensures the integrity of software releases by defining a potential release as an immutable asset early in the SDLC, capturing evidence of all actions taken against the potential release as it matures, and leveraging policy and quality gates to automate and/ or block advancement of the release asset toward consumption. Additionally, JFrog ensures the authenticity of your binaries as they progress through the SDLC. JFrog Artifactory enables signing capabilities to certain package types such as RPM , Maven and Debian repositories. The keys are generated using GPG and this enables package Release file has not been manipulated in any way from the generated metadata.

REQ.ID	Security Requirement	Platform Capability
PS.2	Provide a Mechanism for Verifying Software Release Integrity	The Distribution service provides the ability to create an immutable "Release Bundle" (RB) which is comprised of resources (artifacts, metadata, etc.) which is signed/ encrypted using GPG keys and then distributed to the target JFrog Platform. Once distributed, upon receiving the RB, the target JFrog Platform verifies the atomicity of the bundle by comparing the hashes and thus preserves/secures the resources.
PS.3	Archive and Protect Each Software Release	JFrog Artifactory protects artifacts stored in repositories with role based access control and also fine grade permission model to protect unauthorized access to artifacts such as binaries, builds and configuration & metadata files. The Distribution service provides ability to create "Release Bundle" (RB) which comprises of resources - artifacts, metadata, etc. - which is signed/encrypted using GPG keys. JFrog Xray generates Software Bill of Materials (SBOM) - enabling DevSecOps engineers to understand and analyze the open-source components and dependencies of their builds - thus providing the provenance data. This SBOM report can be exported in SPDX and Cyclone DX formats and thus it is updated on every build. Xray when configured with security policies with fail build/ blocking, security posture with continuous scanning, traceability with ascendants and descendants. JFrog ensures the authenticity of your builds/binaries as they progress through the SDLC. JFrog's long term archive optimizes the storage of Artifacts or builds in an archived manner for regulatory purposes. It has the ability to bring back those artifacts to JFrog Artifactory preserving paths, repository, etc.

REQ.ID	Security Requirement	Platform Capability
PW.1	Design Software to Meet Security Requirements and Mitigate Security Risks	JFrog provides a single platform to define and apply application security and compliance policies across the SDLC with the ability to allow for exceptions to those policies if required. JFrog provides detailed vulnerability reports and insights to enable security risk assessments. It also identifies the impact of vulnerabilities with traceability. JFrog Curation can be used to block malicious or risky packages from even entering your software supply chain - radically reducing risk in development. JFrog provides detailed remediation guidance to developers in their IDE, CLI, or platform UI. JFrog identifies OSS vulnerabilities, exposed secrets, misconfigurations, and poor coding that could lead to many different types of exposures. JFrog's Distribution service provides the ability to create "Release Bundles" (RB)" which are comprised of artifacts, metadata, etc. which is signed/encrypted using GPG keys. JFrog Xray generates an SBOM enabling DevSecOps engineers to understand and analyze the open-source components and dependencies of their builds - thus providing provenance. JFrog ensures the authenticity of your builds/binaries across the SDLC...
PW.2	Review the Software Design to Verify Compliance with Security Requirements and Risk Information	JFrog Artifactory protects artifacts stored in repositories with role-based access control and also fine grade permission model to protect unauthorized access to artifacts such as binaries, configuration and metadata files.

REQ.ID	Security Requirement	Platform Capability
PW.2	Review the Software Design to Verify Compliance with Security Requirements and Risk Information	JFrog's Distribution service provides the ability to create "Release Bundles" (RB)" which are comprised of artifacts, metadata, etc. which is signed/encrypted using GPG keys. JFrog Xray generates an SBOM - enabling DevSecOps engineers to understand and analyze the open-source components and dependencies of their builds - thus providing provenance. data. This SBOM report can be exported in SPDX and Cyclone DX (VEX) format and thus it is updated on every build. Xray when configured with security policies with fail build/ blocking, security posture with continuous scanning, traceability with ascendants and descendants. JFrog ensures the authenticity of your builds/binaries as they progress through the SDLC.
PW.4	Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality	JFrog Curation policies when defined will enable the following: Control & prevent malicious & risky 3rd party packages entering your supply chain. Package download filter against defined set of policy templates In tandem with Catalog - "the LinkedIn of OSS" Transparent developer experience. Automate Curation workflows for a seamless developer experience. Curate OSS libraries in the Artifactory instance (in DMZ network segmentation) thus isolating packages in sanctioned repositories. Promote the libraries back to developer Artifactory instance using JFrog Federated repositories or JFrog Distribution in case of Air gapped environments. The JFrog Catalog will be the source of truth for developers and organizations for approved packages (OSS and proprietary) with private information (metadata) for use within the organization.

REQ.ID	Security Requirement	Platform Capability
PW.4	Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality	JFrog Promotion of Artifacts could be achieved using build promotion from one environment to another (example development to staging) or using immutable Release Bundle from one instance to another (for example: promotion from unclassified environment to classified environment). The Distribution service provides the ability to create “Release Bundles” (RB) which are comprised of resources - artifacts, metadata, etc. - which is signed/encrypted using GPG keys which is then distributed to the target instance. Once distributed, upon receiving the RB, target instance verifies the atomicity of the bundle by comparing the hashes and thus preserves/ secures the resources. JFrog Xray generates Software Billd of Materials (SBOM) - enabling DevSecOps engineers to understand and analyze the dependencies of their components thus providing the provenance data. JFrog Xray also has malicious package detection, vulnerability detection, OSS license check and operational risk for the detected software components. Xray operational risk provides a score for the detected software components and the calculation of it takes into account several factors such as the version age that is in use, EOL projects detection and the level of maintenance of the component, by analyzing public data available in open source projects repositories. JFrog's Advanced Security offering aligns with managing the application’s exposures including: Static Application Security Testing (SAST), Infrastructure as Code (IaC) security, exposed secrets, service misconfigurations, OSS library misconfigurations and determining CVE applicability - in the context of the application. It scans for hard coded secrets, tokens and passwords in code, text and configuration files and inside scanned binaries.

REQ.ID	Security Requirement	Platform Capability
PW.4	Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality	It supports Terraform modules and plan files for the IaC security analysis. For the security findings discovered JFrog provides detailed remediation advice derived by our security research team. These often include simple code changes for fixing the security issue. The Advanced Security CVE contextual analysis feature analyzes code and configurations to detect the actual applicability of the reported vulnerabilities in the scanned application. This feature allows you to eliminate false positives that are not applicable and lets you focus on the ones that actually affect your products. It scans the container image, to see if the vulnerable function is even called and checks reachable paths and tests for configuration for the vulnerabilities. JFrog Cold Storage Archival will archive digital records (artifacts) that are not in use anymore and cannot be deleted due to archiving regulations. JFrog ensures the authenticity of your builds/binaries as they progress through the SDLC.
PW.5	Create Source Code by Adhering to Secure Coding Practices	JFrog SAST enables development teams to write and commit trusted source code. Security-focused engines deliver scans that detect zero-day security vulnerabilities. JFrog SAST checks code against unsafe data flow, functions & calls, and other known bad coding practices. JFrog's IDE plugin provides the ability to warn developers of source code vulnerabilities at development time in the IDE as the software is being built. It can also look at source code in your Git repository - a great shift left practice.

REQ.ID	Security Requirement	Platform Capability
PW.6	Configure the Compilation, Interpreter, and Build Processes to Improve Executable Security	JFrog's IDE plugin provides the ability to warn developers of source code, open-source & configuration vulnerabilities at development time as the software is being built. Knowing the open-source and source code are trusted at compilation time helps ensure executable (binary) security. JFrog also does security, license, configuration, and secrets scans on binaries throughout the SDLC, ensuring binaries are secure before deployment and when in production.
PW.7	Review and/or Analyze Human-Readable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements	JFrog SAST enables development teams to write and commit trusted source code. Security-focused engines deliver scans that detect zero-day security vulnerabilities. JFrog SAST checks code against unsafe data flow, functions & calls, and other known bad coding practices. It checks code in the IDE and Git repositories.
PW.8	Test Executable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements	JFrog's security scans binaries for proprietary, open-source, configuration and services vulnerabilities throughout the software supply chain. Automate security and compliance policies to detect and remediate from your Git repository to production. This approach validates the security of the binaries that will execute in runtime. Detect and remediate vulnerabilities in runtime with continuous scanning.

REQ.ID	Security Requirement	Platform Capability
PW.9	Configure Software to Have Secure Settings by Default	JFrog's advanced security focuses on binary scanning to detect and remediate security, configuration, micro-services, secrets and Infrastructure as code (IaC) vulnerabilities. JFrog checks IaC, OSS libraries and services configurations to make sure that they follow best practices and don't pose a security risk. Scans are performed in source code, configuration files and inside binaries across the stages of the SDLC.
RV.1	Identify and Confirm Vulnerabilities on an Ongoing Basis	JFrog scans from code to runtime. JFrog's security research team continuously updates the vulnerability database through its own research and inclusion of public (NVD) and private 3rd-party sources. Their attack vector and CVE research yields easy-to-use remediation, for speedy fixes by developers or security teams. Artifacts including binaries are continuously scanned for known vulnerabilities with automated policies on an ongoing basis.
RV.2	Assess, Prioritize, and Remediate Vulnerabilities	JFrog's advanced security solution (Contextual analysis) assesses and examines the applicability of identified CVEs by analyzing their context and attributes. It checks if first party code calls the vulnerable function, and if CVEs are in reachability paths as well as checking additional configurations and file attributes to ascertain applicability - and enabling prioritization. JFrog's database coupled with Contextual analysis also recommends concrete, actionable and cost-effective remediation steps that take into account the specific attributes and configurations of the application and recommends developer friendly fixes - sometimes a simple code change. JFrog's security insights dashboards enable easy high-level prioritization of vulnerabilities at a build/project/application level. Simple click through in the user interface enables efficient focus on high or critical security vulnerabilities.

REQ.ID	Security Requirement	Platform Capability
RV.2	Assess, Prioritize, and Remediate Vulnerabilities	JFrog's security insights dashboards enable easy high-level prioritization of vulnerabilities at a build/project/application level. Simple click through in the user interface enables efficient focus on high or critical security vulnerabilities. Enable applicability analysis right in the hands of your development teams with shift left integration in all the popular IDEs, JFrog's command line (CLI) tool, and JFrog Frogbot - which supports Git repository integration and scanning. Prioritize and remediate vulnerabilities as soon as they arise in the early stages of development. Save time and remediate promptly increasing development efficiency. JFrog analyzes infrastructure as code (IaC), misconfiguration and secrets security. For high severity findings JFrog provides a detailed remediation advice from our security research team. JFrog Runtime enables Security and DevOps teams to monitor Kubernetes clusters in real time, identify, prioritize, and remediate security incidents based on actual risk, verify image integrity, and meet compliance requirements.
RV.3	Analyze Vulnerabilities to Identify Their Root Causes	JFrog's Advanced Security solution (Contextual analysis) assesses and examines the applicability of identified CVEs by analyzing their context and attributes. It checks if 1st-party code calls the vulnerable function, and if CVEs are in reachability paths as well as checking other configurations and file attributes to ascertain applicability - and enabling prioritization. JFrog's database with Contextual analysis also recommends concrete, actionable and cost effective remediation steps that take into account the specific attributes and configurations of the application and recommends developer friendly fixes - sometimes a simple code change. A security insights dashboards enable easy prioritization of vulnerabilities at a build, project, or application level. Simple click through in the user interface enables efficient focus on high or critical security vulnerabilities.

REQ.ID	Security Requirement	Platform Capability
RV.3	Analyze Vulnerabilities to Identify Their Root Causes	Enable applicability analysis right in the hands of your development teams with shift left integration in all the popular IDEs, JFrog's command line (CLI) tool, and JFrog Frogbot - which supports Git repository integration and scanning. Prioritize and remediate vulnerabilities as soon as they arise in the early stages of development. Save time and remediate promptly increasing development efficiency. JFrog Security checks IaC Security, library and services configurations and if we there are exposed secrets. For high severity findings JFrog provides detailed remediation advice from our security research team.

ABOUT JFROG

JFrog empowers thousands of DevOps organizations globally to build, secure, distribute, and connect any software artifact to any environment using the universal, hybrid, multi-cloud JFrog Platform.

LEGAL STATEMENT

www.jfrog.com

www.twitter.com/jfrog

www.facebook.com/artifrog/

www.linkedin.com/company/jfrog-ltd

Start for free at https://jfrog.com/start-free/