# JFrog Security
# 完全ガイド

# 目次

# JFrog Security とは？

JFrog Security は、JFrog ソフトウェア・サプライチェーン・プラットフォームの不可欠な構成要素です。ソフトウェア開発ライフサイクル（SDLC）全体にわたる完全な可視化を組織に提供し、すべての JFrog サービスにおいて、「単一の管理画面」によるシームレスな DevOps およびセキュリティ管理を実現します。

JFrog Security は、以下を含む広範なセキュリティツールを網羅しています。

- OSS キュレーション: オープンソースソフトウェアの導入制限と管理
- SCA（ソフトウェア組成分析）: 依存関係の脆弱性スキャン
- SBOM（ソフトウェア部品構成表）: 透明性の高いコンポーネント管理
- コンテナセキュリティ: イメージの脆弱性および設定ミス対策
- コンテキスト解析（Contextual Analysis）: 実行環境に基づいた真の脅威の特定
- シークレット検出: コードや設定ファイル内の機密情報の漏洩防止
- IaC（Infrastructure as Code）セキュリティ: インフラ構成の安全確保
- ミスコンフィギュレーション（設定ミス）: システム設定上の脆弱性検知
- SAST（静的アプリケーションセキュリティテスト）: 独自開発コードの脆弱性解析
- その他、多岐にわたるセキュリティ機能

JFrog Securityにより、次のことが可能になります：

**Enforce**
SDLC全体にわたり、アーティファクトリポジトリ、CI/CDツールやプロセス、さらには統合開発環境（IDE）に至るまで、一貫したセキュリティ対策を適用します。

**Identify**
依存関係の宣言段階でセキュリティ脆弱性やライセンス違反を早期に検出し、悪意のある、またはリスクの高いオープンソースパッケージが組織内に取り込まれる前にダウンロードをブロックします。

**Block**
High/Critical CVE、悪意のあるパッケージ、シークレット露出、設定不備などのリスクを含むビルドは自動的にブロック。セキュリティポリシーを開発基盤にネイティブに強制します。

JFrog CLI、Docker Desktop、Frogbot、JFrog Platform UIを通じて、開発からリリースまで一貫してセキュリティを統合。
本ガイドでは、JFrog Securityの主要機能とプラットフォームとしての価値を解説します。

今すぐ閲覧

**DevSecOps 事例ウェビナーを見る**

## シフトレフトの実践

「シフトレフト」とは、開発プロセスのできる限り早い段階でセキュリティ上の問題を検出・分析・対処するという考え方です。
開発者が利用するオープンソースソフトウェア（OSS）の多くは、公開リポジトリから取得されています。セキュリティリスクは、開発者がインターネットからライブラリをダウンロードした瞬間から始まります。

そのため、脆弱性をできるだけ早い段階でブロックまたは修正することで、対応コストを抑え、開発効率を高めることができます。

JFrog Securityにより、開発者はシフトレフトを容易に実践でき、ソフトウェアが組織の環境に取り込まれる前の段階から、脆弱性やライセンス違反を包括的にスキャンできます。

# JFrog Security ソリューション

## ソフトウェアパッケージキュレーション

JFrog Curation は、依存関係取得の段階でリスクを制御し、危険なオープンソースが組織に取り込まれる前に自動的に遮断します。サプライチェーンの入口からセキュリティを強制するプラットフォーム機能です。
開発基盤にネイティブ統合されることで、常に信頼できる最新パッケージのみを利用可能にし、セキュリティとリリース速度を両立します。

JFrog Curationの主な機能とメリットは以下の通りです。

**一元的な可視化と統制**
サードパーティ製パッケージのダウンロード状況を一元的に可視化・管理します。
既知および未知の脅威から保護し、信頼できるパッケージのみをSDLCに取り込むことが可能です。

**開発者にとってスムーズなパッケージ利用を実現**
開発者が信頼できるサードパーティ製パッケージのみをダウンロード・利用できる環境を提供します。
高・重大レベルのCVE、悪意のあるパッケージ、運用リスク、ライセンス問題などに対する自動ポリシーにより、不要なコンポーネントがパイプラインへ入り込むのを防止します。
ブロックされたパッケージを可視化し、問題を迅速に解決することが可能です。

**サードパーティパッケージの自動キュレーション**
既存の開発パイプラインにシームレスに統合された仕組みにより、サードパーティ製パッケージを継続的に検証します。
開発者には、信頼できるソフトウェアコンポーネントのみを提供し、安全な開発環境を実現します。

**DevSecOpsの体験を向上**
ソフトウェアサプライチェーンの入口で不要な依存関係をブロックする「シフトレフト」アプローチにより、サードパーティパッケージ検証のコストを削減します。
透明性と責任の明確化により、開発チームが利用するサードパーティコンポーネントの品質を確保します。

## JFrog Catalog

JFrogは、数十万件におよぶソフトウェアパッケージに関する豊富で構造化されたデータを継続的に収集し、JFrog Catalog を通じて提供しています。
Catalogはパッケージの“検索エンジン”として機能し、主要なOSSパッケージに関する詳細情報を迅速に確認できます。
Curationポリシーの基盤となるナレッジベースとして、Catalogはビジネスポリシーに基づいた、より正確で安全な意思決定を支援します。

JFrog Curation と JFrog Catalog は、JFrog Artifactory、Xray、Advanced Security と連携することで、ソフトウェア開発ライフサイクル全体のセキュリティを強化します。
その結果、JFrog Platform は「コードからエッジまで」をカバーする、可視化・統制・セキュリティの単一基盤として機能します。

**JFrog Curation と Catalog の詳細はこちら**

## 静的アプリケーションセキュリティテスト（SAST）

SAST（静的アプリケーションセキュリティテスト）は長年利用されてきましたが、いくつかの大きな課題も抱えてきました。例えば、誤検知（false positive）が多いこと、スキャンに時間がかかること、そして単一のソースファイルを超えた分析ができない点などです。さらに、SASTツールは統合が難しく、エンドツーエンドのパイプラインプロセスと適切に連携できていないケースも多く見られます。

JFrogのSASTは、Advanced Securityユーザーに対し、バイナリから開発者が記述するコード（生成AIコードを含む）までをカバーする包括的なセキュリティを提供します。さらに、軽量であるため、開発スピードを損なうことなく利用できます。

JFrogのSASTで実現できる主な機能は以下の通りです：

**開発初期での脆弱性の早期発見**

開発の初期段階でコード内のセキュリティリスクを特定することで、後工程での修正に伴うコストや工数を大幅に抑えることが可能になります。

**攻撃対象領域（アタックサーフェス）の縮小**

コードの脆弱性を特定・排除することで、攻撃対象となり得る領域を減らし、本番環境におけるコードの弱点が悪意ある攻撃者に悪用されるリスクを低減します。

**監査対応とコンプライアンス遵守**

SASTは、規制違反につながる可能性のある脆弱性を特定することで、データセキュリティやプライバシー要件への準拠を支援します。

**コスト効率の向上**

ソフトウェアがCI/CDパイプラインを経て後工程に進んだ後に脆弱性を修正するのは、開発段階で対応する場合に比べてはるかに高コストになります。JFrogのSASTは、開発初期に脆弱性を特定・修正することで、こうしたコストを大幅に削減します。

**ブランドと信頼の強化**

セキュリティインシデントは企業の評価に深刻な影響を与えます。SASTの導入により、コードセキュリティへのコミットメントを明確に示し、顧客の信頼とロイヤルティを高めることができます。

**JFrogのSASTの詳細はこちら**

## ソフトウェア構成分析 (SCA)

Software Composition Analysis（SCA）は、開発チームが依存関係に含まれるセキュリティ脆弱性を迅速にスキャンするためのアプリケーションセキュリティ手法です。

現代のアプリケーションは、自社開発のコードだけで構築されることはほとんどありません。現在、本番ソフトウェアの70〜90%がオープンソースコードで構成されているといわれています。

オープンソースパッケージの普及により開発スピードは大きく向上しましたが、それに伴いセキュリティリスクも高まっています。

JFrog Securityは、バイナリレベルで脆弱性をスキャンすることで、コードから本番環境に至るまで、すべてのソフトウェアを安全かつコンプライアンスに準拠した状態に保ちます。

JFrogのSCAが提供する主なセキュリティ機能とメリットは以下の通りです：

**脆弱性の検出**

JFrogは、ソフトウェアコンポーネントおよび依存関係をスキャンし、National Vulnerability Database（NVD）やCommon Vulnerabilities and Exposures（CVE）をはじめとする、公開・非公開のさまざまな情報源に掲載された既知の脆弱性を特定します。さらに、脆弱性の修正に向けた具体的なステップを伴う実践的なインサイトも提供します。

**継続的なセキュリティ監視**

JFrogは、開発ライフサイクル全体にわたりコンポーネントや依存関係を常時監視し、脆弱性やポリシー違反を検出した際に即座に通知します。

**アーティファクトスキャンの詳細を見る**

**JFrog Artifactoryとのネイティブ統合**

JFrog Artifactoryは、すべてのアーティファクトとリポジトリを一元管理し、組織における単一の信頼できる基盤（Single Source of Truth）を提供します。JFrogのセキュリティソリューションはすべてネイティブに統合されており、リポジトリへのアップロード時点でコンポーネントや依存関係を自動的にスキャンします。これにより、セキュリティは後付けではなく、開発プロセスに自然に組み込まれます。そしてその基盤は、単なる「事実の源」から「信頼の源」へと進化します。

**CI/CDツールとのシームレスな統合**

JFrogは、Jenkins、TeamCity、Bambooなどの主要なCI/CDツールと連携し、CIプロセスに組み込まれた形で、脆弱性やポリシー違反の自動検出を実現します。

## SBOM（ソフトウェア部品表）の可視化

### ソフトウェア部品表 (SBOM)は、

ソフトウェアセキュリティとサプライチェーンリスク管理の中核を担う重要な要素です。

SBOMは、アプリケーションを構成するすべてのコンポーネントおよび依存関係（オープンソースとプロプライエタリの両方）を可視化する包括的なリストです。

リスクを的確に管理するためにはSBOMの整備が不可欠であり、各コンポーネントのバージョン情報も含まれることで、脆弱性の特定と対応をより正確かつ迅速に行うことが可能になります。

SBOMは、アプリケーションを構成するコンポーネントと依存関係を体系的に可視化することで、サプライチェーンリスクの把握と迅速な対応を可能にします。

また、コンプライアンス対応を支援すると同時に、セキュリティへの取り組みを顧客やパートナーに明確に示し、信頼の構築にもつながります。

SBOMの生成は、専用ツールを活用することで効率的かつ自動化できます。

これらのツールは、コンポーネントや依存関係、バージョン情報を自動的に追跡し、CycloneDXやSPDX、SWIDなどの標準フォーマットで可視化します。さらに、最新のVEXフォーマットにも対応しています。

JFrogのSBOMツールをCI/CDに組み込むことで、SBOMの生成は開発プロセスに自然に組み込まれ、継続的に更新されます。新しいビルドやパッケージごとに自動スキャンが行われるため、常に最新の状態を維持できます。

JFrogのSBOMツールとSCAセキュリティ機能により、アプリケーションを構成するすべてのコンポーネントを完全に可視化し、脆弱性やポリシー違反を迅速に検出できます。さらに、リスク評価から継続的な監視までを一貫して実現し、ソフトウェア開発ライフサイクル全体のセキュリティを強化します。

## シークレット（機密情報）の検出

シークレットは、機密システムへのアクセスを支える重要な情報です。
APIキーやパスワードなどの認証情報は、開発プロセス全体のセキュリティを維持する上で不可欠です。

JFrog Securityは、Artifactoryに格納されたアーティファクトやビルドをスキャンし、露出したシークレットを検出することで、トークンや認証情報の漏洩リスクを未然に防ぎます。
JFrogは、設定・テキスト・バイナリファイル全体を対象に、平文の認証情報や秘密鍵、トークンなどのシークレットを検出します。

150種類以上の認証情報タイプに対応した継続的に更新されるリストに加え、独自のシークレット検出エンジンにより、高精度かつ広範なカバレッジを実現します。さらに、証明書のスキャンも行い、有効期限切れや脆弱な証明書といったリスクも見逃しません。

以下は検出される可能性のある一例です：

有効期限が切れた証明書の使用


平文のAPIキーや秘密鍵が
含まれている状態

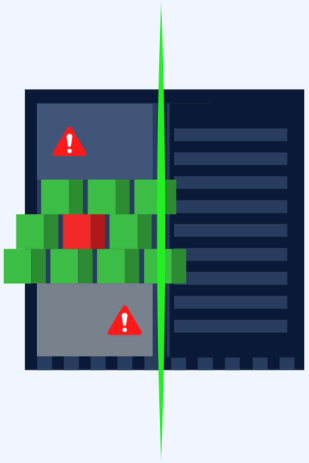
JFrog Securityは、シークレットの露出を検出するだけでなく、迅速な対応につながる具体的なインサイトを提供します。 検出箇所の行番号、想定されるリスク、推奨される修正方法までを提示し、確実な対処を支援します。
どれほど高度なセキュリティ対策も、鍵情報が露出していては無意味です。

**シークレット検出ワークショップについて詳しく見る**

## コンテナのセキュリティスキャン

コンテナスキャンは、コンテナ内の全レイヤーを分析し、イメージやコンポーネントに潜む脆弱性を特定するプロセスです。
コンテナアプリケーションのセキュリティにおいて中核となる機能であり、開発段階で問題を解消することで、安全なデプロイを実現します。

JFrogのコンテナスキャンの主な機能とメリットをご紹介します：



**マルチレイヤースキャン**
JFrogは、ベース、アプリケーション、OSを含むすべてのレイヤーを横断的にスキャンし、コンテナスタック全体にわたる脆弱性やコンプライアンスリスクを可視化します。
どれだけ多層構造であっても、すべてのオープンソースコンポーネントを分析し、既知の脆弱性を確実に検出します。


**コンテナのコンテキスト分析（Container Contextual Analysis）**
業界初のこの機能は、コンテナの構成や属性を深く分析し、検出されたCVEが実際に悪用可能かどうか（適用性）を判断します。
設定やファイル属性まで含めて検証することで、真に対応が必要なリスクを見極めます。
さらに、コンテナの特性に基づいた実践的かつコスト効率の高い修正方法を提示します。

SCAツールの課題は、過剰な検出によるノイズです。
実際にはリスクのない脆弱性まで大量に検出されることで、開発者は不要な対応に時間を費やしてしまいます。JFrogの調査では、DockerHubの主要イメージにおけるCVEの78%が実際には悪用不可能であることが判明しています。

これは、構成やセキュリティ設定といったコンテキストが考慮されていないためです。
JFrogのコンテナスキャンは、この課題を解決し、開発者が「すべてを直す」のではなく「本当に必要なものだけを直す」ことを可能にします。

## IaC（Infrastructure as Code）セキュリティ

IaCセキュリティは、クラウド構成の問題をデプロイ後ではなく、インフラコードの段階で解決するためのベストプラクティスです。
従来は、クラウドリソースのデプロイ後にセキュリティ対策を講じるのが一般的であり、手動設定によるリスクを後追いで管理する形でした。

JFrogのIaCセキュリティは、スケーラブルで一貫性のあるクラウドセキュリティを実現し、問題を早期に検出することで実行時リスクを最小化します。
アーティファクトリポジトリ、CI/CDプロセス、さらにはIDEまで、SDLC全体にわたってセキュリティを組み込み、組織全体での統制を可能にします。JFrogにより、IaCファイルを通じてクラウドおよびインフラの設定ミスを早期に検出し、セキュリティを強化できます。

## 高度なセキュリティスキャン機能

JFrog Advanced Securityは、JFrog Xrayの機能を拡張し、SCAの範囲を超えた包括的なソフトウェアサプライチェーンセキュリティを実現します。

前述の通り、革新的な機能を備えており、その主な特長とメリットは以下の通りです。

### SAST: 信頼できるビルドは信頼できるコードから

シームレスで開発者フレンドリーな体験により、信頼できるコードの開発とコミットを支援します。
高速かつ高精度なセキュリティエンジンにより、誤検知を最小化しながら、開発スピードを維持したままスキャンを実現します。

### コンテキストに基づくCVE優先度付け：本当に重要な脆弱性だけに集中

大量の脆弱性に圧倒されていませんか？その多くは実際にはリスクではありません。
JFrogのコンテキスト分析エンジンは、コードとその属性をもとに、CVEが本当に影響を及ぼすかを判断します。
脆弱な関数が実際に呼び出されているかを検証し、さらに設定やファイル属性まで分析することで、悪用可能性を正確に評価します。

### シークレット検出：重要な認証情報の漏洩を未然に防ぐ

コンテナやアーティファクト内に、認証情報が露出している可能性を見逃していませんか？
JFrogのシークレット検出は、既知の構造に加え、不審な変数パターンも解析することで、ランダムな認証情報まで検出します。これにより、誤検知を抑えつつ高精度な検出を実現します。

### IaCセキュリティ：デプロイ前にクラウドインフラのリスクを防ぐ

IaCファイルの利用が拡大する中で、人的ミスによるリスクは増大しています。
クラウドの安全な運用には、IaCの設定を事前に検証し、適切に保護することが不可欠です。
JFrogのIaCセキュリティスキャナーは、これらのリスクに対して包括的かつ先回りした対策を提供します。

### サービスおよびOSSライブラリの露出対策：アプリケーションを攻撃から守る

従来のセキュリティ対策では見落とされがちな領域にも、JFrogは踏み込みます。
DjangoやFlask、Apache、Nginxといった主要なOSSライブラリやサービスの設定や利用方法まで深く分析し、誤用や設定ミスを検出。
これにより、潜在的な攻撃リスクを事前に排除します。

JFrog Advanced Securityは、ソフトウェアサプライチェーン全体のセキュリティを強化し、侵害リスクを最小限に抑えながら、組織のセキュリティ体制を次のレベルへと引き上げます。

# JFrog 製品と幅広い エコシステムとの連携

JFrog Securityは、JFrog Artifactoryとネイティブに統合され、ソフトウェアサプライチェーン全体を統合的に管理するプラットフォームを形成します。

包括的なアーティファクト管理と一体化したセキュリティスキャンにより、他にはない統合型のアプリケーションセキュリティを実現します。

豊富なメタデータ、ディープなバイナリ解析、革新的なセキュリティ機能を組み合わせることで、開発者はコンポーネント間の関係性を可視化し、影響範囲を正確に把握できます。これにより、脆弱性の影響を横断的に理解し、より迅速かつ適切な対応が可能になります。

エコシステムとの連携について詳しく見る

# JFrog Security ユースケース

JFrog Securityが対応する主なユースケースには、SBOMおよび規制対応、コンテナセキュリティ、IaCセキュリティ、脆弱性管理、FOSSコンプライアンスおよびライセンス対応などがあります。

## SBOMと規制コンプライアンス対応

- SPDX、CycloneDX、VEXなどの標準フォーマットに対応したSBOMを効率的に生成
- メタデータにとどまらないバイナリ解析により、高精度で包括的なSBOMを実現
- SDLC全体にわたる脆弱性の継続的な監視と管理により、規制要件に対応
- 大規模な悪意あるパッケージデータベースを活用し、不正なパッケージの混入を防止
- 必要に応じてSBOMおよび関連するCVE情報を自動的に公開

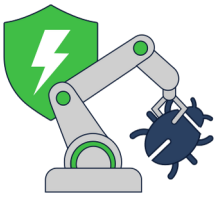
## IaC セキュリティ

- IaC構成を事前に検証し、潜在的なリスクを早期に可視化
- 安全性を確保しながら、デプロイに伴うリスクを最小限に抑制

## コンテナのセキュリティ対策

- バイナリレベルでの分析により、「バイナリ内のバイナリ」やコンテナの全レイヤーを含めたリスクを可視化
- 設定情報やファーストパーティコードとオープンソースソフトウェアの相互作用まで考慮した包括的なスキャンにより、盲点を最小化し、より正確なコンテキストを把握
- 高度なスキャン機能により、悪用可能な脆弱性にフォーカスすることで、セキュリティスキャンにかかる時間を削減

## 脆弱性管理



- SDLC全体を通じて発生するセキュリティ課題を追跡・管理し、インシデントに迅速に対応
- 検出されたCVEに対処し、リスクの深刻度を低減
- 自動化により悪意あるパッケージを特定・ブロックし、AppSecチームの負荷を軽減

## FOSSコンプライアンスとライセンス管理

- プロジェクト、チーム、顧客、提供先に応じて、許可されたライセンスのみで製品出荷を管理・統制
- 従来手動で行っていたライセンスクリアランスプロセスを自動化し、開発チームが適切に承認されたライセンスのみを利用できるようにすることで、法的リスクを回避

# まとめ

JFrog Securityは、開発プロセスで扱うすべてのソフトウェアパッケージを一元管理し、最高レベルのセキュリティとコンプライアンスを実現する基盤です。
JFrogにより、ソフトウェアサプライチェーン全体を単一のプラットフォームで可視化・統制し、安全に運用することができます。

デモを見る

**JFrog Security を無料で始めてみませんか？**

JFrog Securityは、オンプレミスからクラウド、マルチクラウド、ハイブリッドまで、あらゆる環境に柔軟に対応します。
JFrog Platformの一部として、Amazon Web Services、Google Cloud Platform、Microsoft Azureといった主要クラウド上で利用可能です。
トライアル ＆ 価格ページ をご確認ください。

**無料トライアル**