開発者向けの 6 つの DevSecOps ベストプラクティス

1 アプリケーションの依存関係リスクを軽減する

管理が必要なアプリケーションの構築に使用されたソフトウェアコンポーネントに脆弱性が含まれていることがあります。ソフトウェア・コンポジション分析 (SCA) ツールを使用して、ソフトウェア・サプライチェーンのリスクを軽減できます。オープンソースソフトウェア・コンポーネントを使用している場合は特に有用です。以下の機能を持つ SCA ツールを探しましょう。

- アプリケーションの依存関係をスキャンして、既知の脆弱性がないことを確認する
- コンポーネントとそのライセンスを特定し、準拠していない可能性があるライセンスを洗い出して、ソフトウェアライセンスのコンプライアンスの自動化を支援する
- アプリケーションの依存関係が最新で、今もアクティブな状態でアップデートを作成しているコミュニティから発生していることを確認する
- アプリケーション構築プロセスの自動化された部分、および開発環境の部分となる。これにより、開発者は問題が統合される前に解決でき、アプリケーション構築エラーの数が減少します。

2 コードと構成管理を統一する

Kubernetes やコンテナ化環境では一般的である GitOps パラダイムには、開発から始めて、セキュリティ体制を大幅に向上できる手法が含まれています。

- ソースコード管理 (SCM) の開発ベストプラクティスを構成に適用します。チェックイン、マージ、承認に同じコントロールを使用すると、インフラストラクチャの構成変更を特定の人物や時間にまで追跡できます。
- 運用担当者に頼るのではなく、開発者はプロセスの早い段階で構成を検討し、アプリケーションの意図されたプロダクション環境に対するビジョンを確立しなければなりません。同じタイプの環境とセキュリティコントロールを、開発、テスト、プロダクションに使用すると、ライフサイクル全体を通じて構成の管理が容易になります。
- 自動化された構築パイプラインを使用して、継続的インテグレーション/継続的デリバリー (CI/CD) のコンテナイメージおよびバイナリーアーティファクトを構築します。これらのイメージをプロダクションにデプロイするとき、場当たり的な変更は不要になります。
- 機密データを SCM システムに保存してはいけません。ツールを使用して構成とコンテナイメージをスキャンし、埋め込みシークレットが含まれていないことを確認します。

3 アプリケーション・シークレットを保護する

ID や、パスワード、トークン、キーなどのシークレットを、アプリケーションのライフサイクル全体で管理することが重要です。SCM システム、コンテナレジストリ、バイナリーリポジトリへのアクセスを制御する必要があります。アプリケーションがデータベースやサービスへのアクセスに使用する認証情報や、自動化されたビルドおよびテストプロセスに必要な認証情報も、セキュリティ保護する必要があります。シークレットを SCM システムや設定ファイルに保存していると、偶発的に開示されてしまうことがあります。アプリケーション・シークレットを保護するには、以下の対策を講じます。

- ID 管理およびアクセス制御インフラストラクチャをライフサイクルの早期段階で確立します。
- シークレットボールト、またはハードウェア・セキュリティ・モジュール (HSM) を使用して、保管時および移動時にシークレットを管理および保護することを検討します。シークレットボールトは通常ソフトウェア・ソリューションで、HSM は特殊なハードウェアを使用して保護レベルを強化します。このいずれかを ID 管理インフラストラクチャに統合します。

4 信頼できるベースイメージを使用する

コンテナベースイメージは、最小限に凝縮された Linux® ディストリビューションです。数百のパッケージをプリインストールでき、脆弱性が含まれている可能性があります。コンテナイメージのリスクを軽減するには、以下の対策を講じます。

- 十分にテスト済みで信頼性のあるアップデートが定期的に提供される、信頼されるイメージを選択します。イメージのソースと利用できるサポートオプションを確認します。
- イメージツールを使用して既知の脆弱性をチェックします。イメージをスキャンして、構成がセキュアで、埋め込みシークレットがないことも確認する必要があります。
- エクスプロイトに使用される可能性がある、オペレーティングシステム (OS) ツールなどの不要なバイナリーを削除して、攻撃ベクトルを削減します。

5 コンプライアンスおよび監査の問題に対して早期に対処する

プロダクションに移行する際の遅延を低減するため、開発の早期段階で必要となるコンプライアンス・フレームワークと技術的制御を理解することが重要です。コンプライアンスとセキュリティ要件を施行するための自動化チェックを構築パイプラインに挿入できます。

あらかじめ文書化に取りかかります。手順とポリシーの文書化は監査の 50% 以上を占めます。ポリシーの文書には、アクセス制御、変更管理、バックアップ、データ保持を含めます。アプリケーションのセキュリティテストや SCA などのセキュリティチェックを、手順を文書化するときに対象として含めます。

6 強力なプラットフォームとエコシステムから始める

セキュリティの脅威が増え続ける中、統合されたサポート付きソリューションを提供する、包括的なセキュリティエコシステムを備えたプラットフォームを使用することが重要です。Red Hat® OpenShift® は、開発のみならず運用もサポートする多彩な機能を備えた、エンタープライズグレードの Kubernetes プラットフォームです。Red Hat OpenShift における強力な構築およびデプロイパイプラインは、自動化されたセキュリティチェックとコントロールを実装する最適な環境を実現します。セキュリティチェックは、ソースコードのイメージへの構築からプロダクションのデプロイまで、プロセスのどの段階にも挿入できます。

Red Hat にはセキュリティパートナーのエコシステムがあり、Red Hat OpenShift のセキュリティ機能を強化し、拡張します。これらのパートナーは Red Hat と連携して、Red Hat OpenShift と統合されるサポート付きソリューションを提供します。多様なソリューションの中から、お客様固有のセキュリティおよび組織上の要件に適合するソリューションを選択できます。

コンテナベースのアプリケーションの開発を迅速化するため、 Red Hat OpenShift 上で実行される Red Hat CodeReady Workspaces は Kubernetes ネイティブの開発環境です。Red Hat Universal Base Images と Red Hat Runtimes は、アプリケーションの信頼できるソースから強力な基盤を提供します。

Red Hat DevSecOps フレームワーク

セキュリティのライフサイクルの全体的なビューを把握し、開発セキュリティ機能が Red Hat DevSecOps フレームワークに適合することを確認するには、red.ht/DevSecOps にアクセスしてください。

開発セキュリティ・ソリューションを見つける

Red Hat および Red Hat のセキュリティパートナーが提供する Web セミナーをご覧になり、アプリケーションのライフサイクル全体を通じてセキュリティを組み込む方法をご確認ください。

Red Hat について

Red Hat は、受賞歴のあるサポート、トレーニング、コンサルティングサービスをお客様に提供し、複数の環境にわたる標準化、クラウドネイティブ・アプリケーションの開発、複雑な環境の統合、自動化、セキュリティ保護、運用管理を支援します。

アジア太平洋
+65 6490 4200
apac@redhat.com

オーストラリア
1 800 733 428

インド
+91 22 3987 8888

インドネシア
001 803 440 224

日本
03 4590 7472

韓国
080 708 0880

マレーシア
1 800 812 678

ニュージーランド
0800 450 503

シンガポール
800 448 1430

中国
800 810 2100

香港
800 901 222

台湾
0800 666 052

fb.com/RedHatJapan
twitter.com/RedHatJapan
linkedin.com/company/red-hat

jp.redhat.com
#F29682_0921