sysdig

ソフトウェアセキュリティ大全の全体像

パイプラインの各段階で必要となるもの

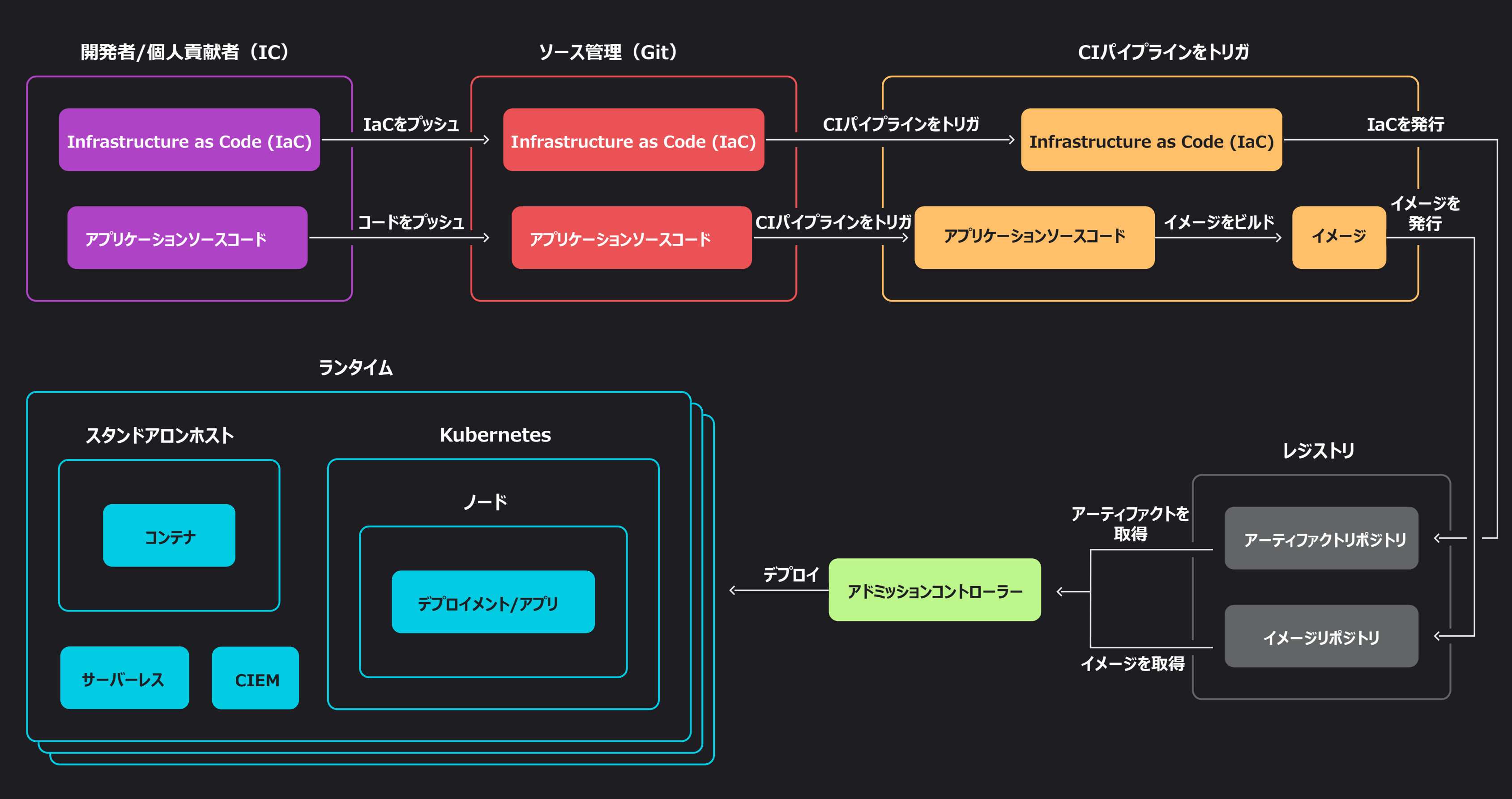
このインフォグラフィックでは、ソフトウェアライフサイクルの各段階を保護する方法を、「**シフトレフト**」アプローチに焦点を当てて紹介します。シフトレフトアプローチを採用することで、早い段階で修正を行うことができ、それにより、リスクの低減とコストの削減が可能となります。Sysdigは、開発パイプライン全体を通じてセキュリティチェックを統合することで、開発者がVSCode、Jenkins、GitHub Actionsなどのツールを使用して問題の早期発見と修正を行えるようにしています。また、セキュリティフィードバックを開発者環境やJiraやGitHub Issuesのようなチケットシステムへと統合することで、脆弱性に迅速かつ効率的に対処できることを確実にします。

このプロセスに含まれる主要なセキュリティ概念としては、多層防御、最小権限、ゼロトラスト、セキュリティバイデフォルト、継続的コンプライアンスなどが挙げられます。

- **多層防御：**多層化されたセキュリティ管理により、さまざまな段階で脆弱性を確実に捕捉します。たとえ問題が早期の段階でセキュリティ管理をすり抜けたとしても、その問題は、CI/CDパイプラインやランタイムなどのような次の段階で検知される可能性が高くなります。
- **最小権限：**最小権限の原則とは、ユーザーやサービスに、必要としているアクセス権のみを確実に付与することで、リスクを最小限に抑えるものです。Sysdigを使うと、過剰な権限を付与されているロールや構成を特定して修正することにより、アタックサーフェスを縮小できます。
- **ゼロトラスト：**Sysdigは、あらゆる段階でコンポーネントを継続的に検証し、信頼できる要素だけが本番環境に進めるようにします。Sysdigは、ランタイム中においても、異常の発生を監視しており、発生した異常にフラグを立てます。
- **セキュリティバイデフォルト：**早い段階からセキュアなプラクティスを組み込むことで、Sysdigは設定ミスの可能性を減らします。これにより、セキュリティは後付けされるものではなく、開発プロセスに組み込まれるものとなります。
- **継続的コンプライアンス：**定期的な自動スキャンにより、セキュリティポリシーの継続的な遵守を確実にし、新たな脆弱性の出現をいち早く捕捉することで、セキュリティドリフトを防止します。

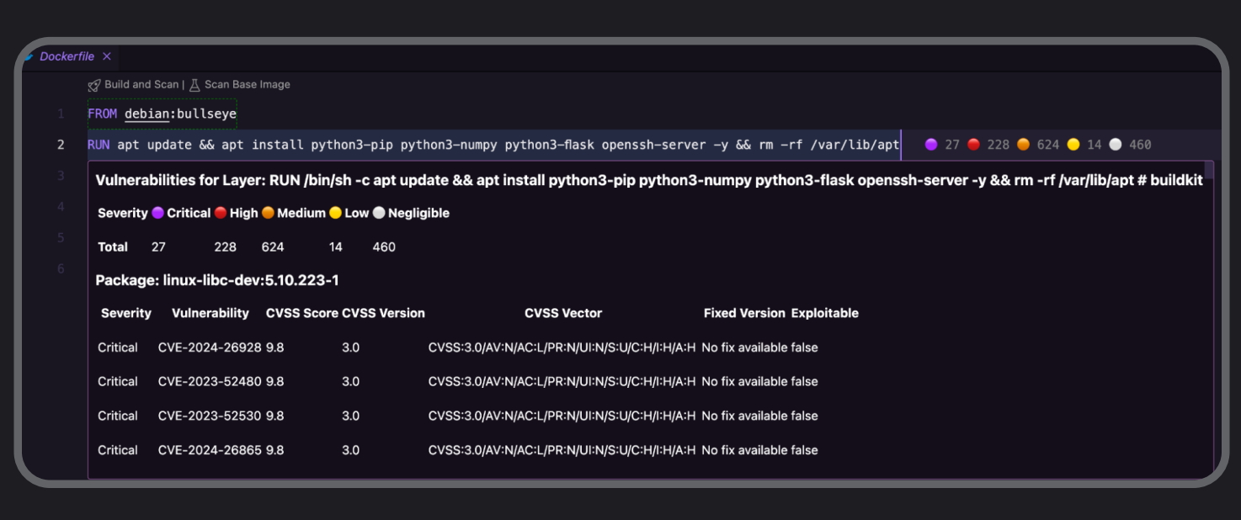
また、Sysdigは、**イミュータブルインフラストラクチャー**と**宣言型IaC**を活用することで、複数の環境間での一貫性を維持しています。これにより、構成のドリフトを減らし、トレーサビリティを向上させています。円滑なセキュリティチェックは、マイクロサービスやクラウドのようなペースの速い環境でも、開発を遅らせることなく、DevOpsワークフローとスムーズに統合できます。

最後に、**ランタイム脅威検知**は、実行中のワークロードにおけるゼロデイ脆弱性や疑わしい振る舞いを特定することで、アクティブな脅威に対する強力な最終防御ラインを提供します。



開発者/個人貢献者（IC）

コードベースの記述、テスト、保守を行う個人です。彼らはツールや環境を利用して、**Infrastructure as Code（IaC）**や**アプリケーションのソースコード**を開発します。



```
Dockerfile ×
Build and Scan | Scan Base Image
1  FROM debian:bullseye
2  RUN apt update && apt install python3-pip python3-numpy python3-flask openssh-server -y && rm -rf /var/lib/apt   ● 27 ● 228 ● 624 ● 14 ● 460
3  Vulnerabilities for Layer: RUN /bin/sh -c apt update && apt install python3-pip python3-numpy python3-flask openssh-server -y && rm -rf /var/lib/apt # buildkit
4  Severity ● Critical ● High ● Medium ● Low ● Negligible
5  Total    27      228    624      14     460
6  Package: linux-libc-dev:5.10.223-1
   Severity   Vulnerability    CVSS Score CVSS Version          CVSS Vector                                  Fixed Version  Exploitable
   Critical   CVE-2024-26928 9.8         3.0            CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H  No fix available false
   Critical   CVE-2023-52480 9.8         3.0            CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H  No fix available false
   Critical   CVE-2023-52530 9.8         3.0            CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H  No fix available false
   Critical   CVE-2024-26865 9.8         3.0            CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H  No fix available false
```

Sysdigを導入すると、**Visual Studio Code拡張機能**を利用するか、または**Sysdig CLI Scanner**を実行することで、開発中に設定ミスや脆弱性をスキャンできるようになります。

Infrastructure as Code

Terraformや**Helmチャート**のようなツールを使ってInfrastructure as Code（IaC）を開発する場合、セキュリティおよび規制コンプライアンスを確保するために、CISベンチマークやNIST SP 800-53のようなコンプライアンス標準を遵守することが不可欠となります。

開発者は、最小権限の原則、適切なシークレット管理、定期的なコードレビューなどのベストプラクティスを実施することで、強固なセキュリティポスチャーを維持する必要があります。

アプリケーションソースコード

開発者は、自分が書いたコードや、使用しているライブラリやパッケージに含まれている可能性のある脆弱性を積極的に見つけようとする必要があります。問題を早期に捕捉することで、それらを迅速かつ効率的に修正できるようになるため、小さな欠陥が将来大きなセキュリティリスクになるのを防ぐことができます。

ソース管理（Git）

コードリポジトリ管理を通じて、バージョン管理、ブランチ、マージ、そして開発者間のコラボレーションを可能にします。

Sysdig内部での**Gitとの統合**を有効にすることで、お使いのリポジトリ内にある設定ミスを自動的に検知し、Sysdigからの推奨パッチを含むプルリクエストをオープンできるようになります。

Infrastructure as Code

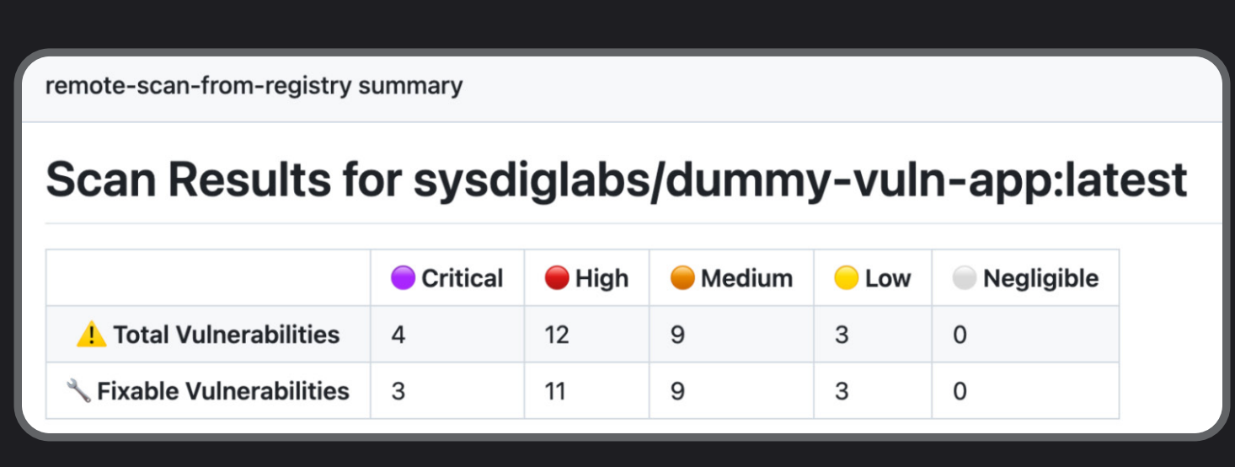
この段階で修正を適用すると、開発者はパッチを認識できるようになり、解決が迅速化されます。ここでセキュリティチェックを統合すると、問題が最も簡単に修正できる開発プロセスで検出されます。また、早期検出により、開発者間でセキュリティを第一に考える考え方が育まれ、コーディングプラクティスの改善が促進されます。

アプリケーションソースコード

Gitリポジトリでソースコードの段階からアプリケーションを保護することは、脆弱性を早期に発見する鍵となります。なぜなら、これにより、パイプラインやランタイムに影響を与える前に、問題を確実に修正できるようになるからです。Gitワークフローにおける自動化されたチェックを使うと、開発を遅らせることなくセキュリティを維持できるようになります。

CI/CDパイプライン

CI/CD（継続的インテグレーションおよび継続的デプロイメント）のパイプラインは、コード変更の統合、テストの実行、そしてアプリケーションのデプロイを行うプロセスを自動化します。



remote-scan-from-registry summary

Scan Results for sysdiglabs/dummy-vuln-app:latest

	● Critical	● High	● Medium	● Low	● Negligible
⚠ Total Vulnerabilities	4	12	9	3	0
Fixable Vulnerabilities	3	11	9	3	0

お客様がSysdigを使ってCI/CDパイプラインをスキャンできる環境の例としては、**GitHub**、**GitLab**、**Jenkins**、**Azure Pipelines**などが挙げられますが、これらはほんの一例にすぎません。

Infrastructure as Code

CI/CDパイプラインにおけるセキュリティチェックは、デプロイ前に残っているIaCの設定ミスを検知し、セキュアでないインフラ設定が本番環境に到達するのを阻止します。また、パイプライン内での自動スキャンは、異なる環境間でのセキュリティポリシーの一貫した実施を確実にします。

アプリケーションソースコード

この時点で、アプリケーションはビルドされますが、依存関係に関するバージョン固定が行われていない場合、新たなセキュリティ上の懸念が生じます。なぜなら、その場合、アプリケーションは再現性に失敗し、開発時には存在しなかった新しい脆弱性を導入してしまう可能性があるためです。通常、アプリケーションは、コンテナイメージへとパッケージ化されます。

イメージ

現在では、アプリケーションの脆弱性だけでなく、コンテナイメージのビルド手順における悪しきプラクティスも問題になっています。これには、OSレベルの依存関係、公開されたシークレット、ルート（root）ユーザーとしての不適切なデフォルト設定、あるいはベースイメージから継承されたさらなるセキュリティ違反などが含まれます。イメージがセキュリティポリシーを満たしていない場合、パイプラインは失敗し、その結果、そのようなイメージのパブリックレジストリへの到達は阻止されます。

レジストリ

レジストリとは、ビルドアーティファクトとコンテナイメージ向けの一元化されたストレージシステムであり、取り出しとデプロイメントを容易にします。この段階において、たった1つの脆弱性や設定ミスが、デプロイメント時に数千から数百万件もの影響を受けるインスタンスにつながる可能性があることを肝に銘じておいてください。これこそが、前段階を保護することが重要となる理由です。

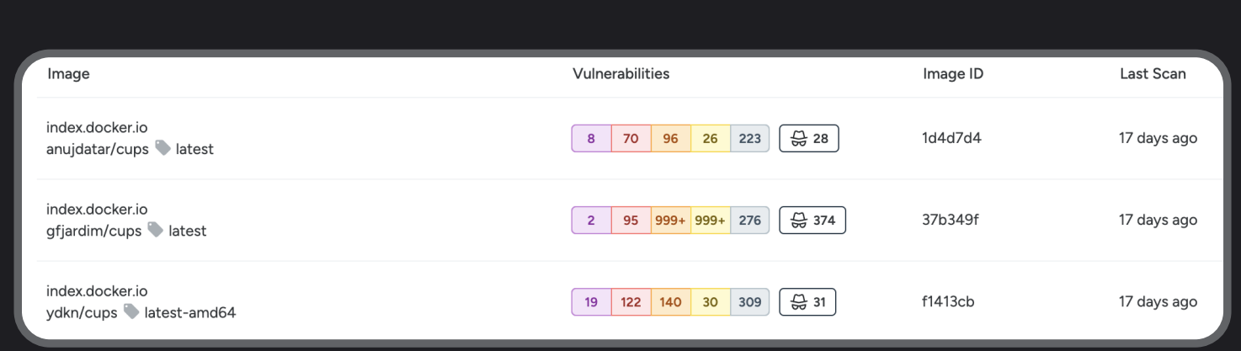


Image	Vulnerabilities	Image ID	Last Scan
index.docker.io anujdatar/cups latest	8 70 96 26 223 28	1d4d7d4	17 days ago
index.docker.io gfjardim/cups latest	2 95 999+ 999+ 276 374	37b349f	17 days ago
index.docker.io ydkn/cups latest-amd64	19 122 140 30 309 31	f1413cb	17 days ago

Sysdig Registry Scannerのチャートを使うと、包括的なリストに記載されているサポートされているベンダーのレジストリをスキャンできます。

アーティファクトリポジトリ

リポジトリ内のアーティファクトは、保存される前にセキュリティチェックを通過する必要があります。これにより、セキュアなコンポーネントだけを利用可能とすることを確実にできます。また、定期的なスキャンにより、新たに発見された脆弱性を捕捉します。これにより、将来のビルドにおいて、安全でありかつコンプライアンスに準拠したリポジトリを維持できるようになります。

イメージリポジトリ

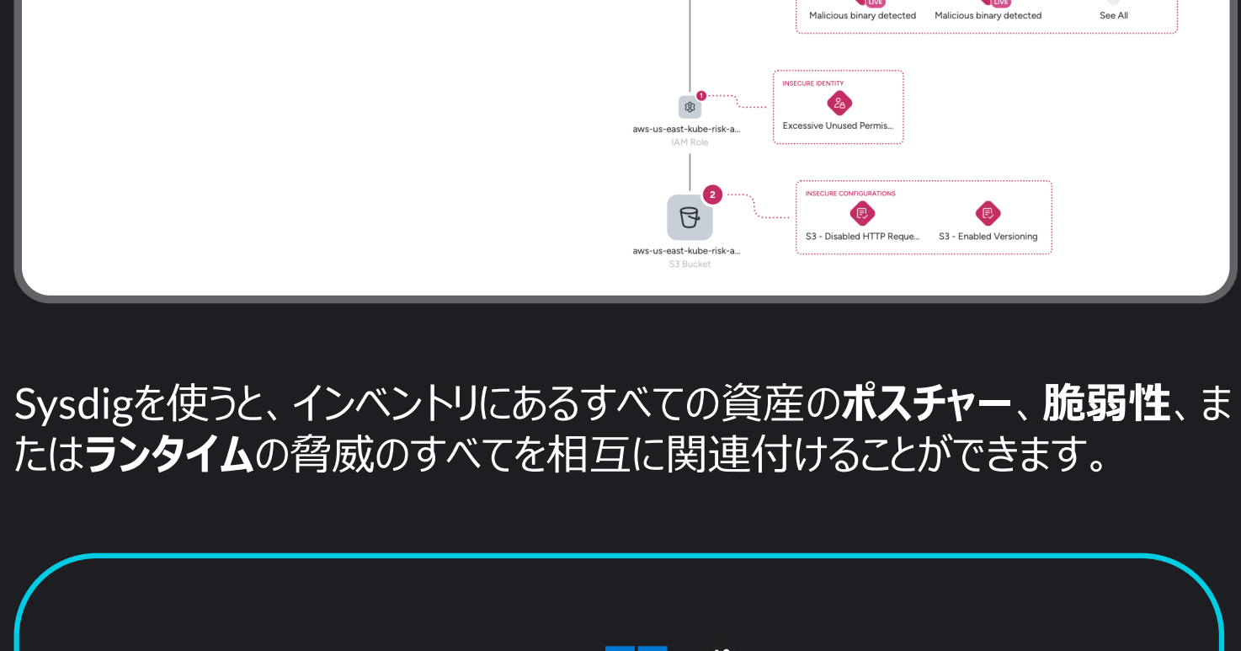
イメージを継続的にスキャンすることで、イメージが保存された後に出現した脆弱性を検知する必要があります。信頼できるセキュアなイメージのみが、デプロイメントを許可されるべきです。また、定期的な再スキャンにより、保存されているイメージも最新のセキュリティ基準を満たしていることを確実にできます。

アドミッションコントローラー

アドミッションコントローラーは、デプロイメント前の最後のセキュリティチェックポイントとして機能するものであり、適合したワークロードのみに実行を許可されることを確実にします。アドミッションコントローラーは、インフラストラクチャーとアプリケーションコンポーネントの両方が、事前に定義されたセキュリティポリシーを満たしていることを検証します。問題が検知された場合、アドミッションコントローラーはデプロイメントを阻止するか、あるいはレビューのためにフラグを立てます。この最終レイヤーは、初期段階で見落とされたものを捕捉し、ゼロトラストアプローチを強化するのに役立ちます。**Sysdig Admission Controller**を使うと、指定されたポリシーに合格したデプロイメントのみを許可できるようになります。

ランタイム

ランタイム環境は、クラウドまたはオンプレミスでアプリケーションをホスティングする環境を指します。これには、スタンドアロンホスト、Kubernetesクラスター、サーバーレスプラットフォーム、CIEMシステムなどが含まれます。ランタイム環境は、アプリケーションをセキュアにかつ効率的に実行し管理するためのインフラを提供します。アプリケーションは、テストとポリシーチューニングのためにQA/ステージングへとデプロイされます。これにより、アプリケーションがユーザーにとってアクセス可能となる本番環境へと移行する前に、チームがセキュリティ設定を改良できるようになります。



Sysdigを使うと、インベントリにあるすべての資産の**ポスチャー**、**脆弱性**、または**ランタイム**の脅威のすべてを相互に関連付けることができます。

スタンドアロンホスト

実行時に、スタンドアロンホストは、脆弱性や不審な動作を継続的に監視する必要があります。定期的なセキュリティチェックを行うことで、ホストが古い設定やセキュアでない設定を実行していないことを確実にできます。ホスト上で実行されている**Sysdigエージェント**は、これらのコンテナでの疑わしい活動をすべて特定します。

コンテナ

コンテナは、実行中に脆弱性がないかどうかを調べるために継続的に監視する必要があります。また、コンテナの寿命はわずか数秒間である場合もあるため、セキュリティには迅速なアクションが必要となります。Sysdigの**Container Drift**は、外部バイナリを検知し、それが実行されるのを防ぎます。

Kubernetes

Kubernetesのセキュリティは、コンテナとその構成における脆弱性の管理を通じて、ワークロードを保護することに重点を置いています。適切なリソース制限とロールベースのアクセス制御（RBAC）は、エクスポージャーを減らすために不可欠です。インシデントは、**Kubernetesの監査ログ**を分析することで発見されます。.

ノード

スタンドアロンホストと同様に、Kubernetesは（DaemonSet経由で）各ノードにSysdig Agentをデプロイすることで、すべてのワークロードの予期せぬアクティビティを検知した上で、脅威アクターによるマルウェアの実行を防止します。Sysdigの**キャプチャ**と**Sysdig Inspect**を使うことで、インシデントを詳細に分析できるようになります。

コンテナ

Sysdigにより生成されたネットワークポリシーを使用して、強力なコンテナネットワーキングを実現します。

サーバーレス

サーバーレス環境では、不正アクセスや設定ミスを防ぐために厳格なセキュリティポリシーが必要となります。コンプライアンスを徹底することで、承認された機能のみがデプロイされ実行されるようになります。機能の隔離は、アタックサーフェスを縮小し、潜在的な侵害を制限するために不可欠です。

CIEM

CIEMは、セキュリティにとって極めて重要です。それは過剰な権限を管理して最小化することで、重要なクラウドリソースへの不正アクセスのリスクを低減するためです。CIEMは最小権限の原則を実施することで、ユーザーとサービスが真に必要とするアクセス権のみを持つことを保証し、権限昇格攻撃の可能性を低減します。Sysdigは、さまざまな活動を通じてCIEMを支援します。これには、クラウドのアクセス許可を継続的に分析すること、誤った設定や過度に許可されたロールを特定すること、そしてリスクを低減するための実用的なインサイトを提供することなどが含まれます。

sysdig

インフォグラフィック

ING-011-JA REV. A 11/24

Sysdigについて

クラウド環境では、1秒1秒が重要となります。Sysdigは、ランタイムインサイトとオープンソースのFalcoを使用してリスクの変化を即座に検出することにより、クラウド攻撃をリアルタイムで阻止します。Sysdigは、ワークロード、アイデンティティ、サービスを横断してシグナルを相関させ、隠れた攻撃経路を発見し、最も重要なリスクに優先順位を付けます。

デモを依頼 →