アマゾン ウェブ サービス（AWS）クラウドとコンテナを保護するための5つのベストプラクティス

sysdig SECURE EVERY SECOND.

アマゾン ウェブ サービス（AWS）クラウドとコンテナを保護するための5つのベストプラクティス

コンテナとクラウドの導入が加速する中、これらの環境に対する可視性を確保することは、依然として企業にとって大きな課題です。クラウド移行の進展とDevOpsプラクティスの広範な普及によって、コンテナ化は今や主流のトレンドとなりました。Gartnerは、2029年までに世界の組織の95%以上が本番環境でコンテナ化されたアプリケーションを運用すると予測しており、これは2023年の50%未満から大幅に増加する見込みです。この爆発的な成長は、アプリケーションの構築、導入、管理の方法に革命をもたらしています。しかし同時に、潜在的な攻撃対象領域を拡大させ、組織を新たな高度な脅威にさらす可能性も高めています。

また、コンテナは基本的にブラックボックスです。コンテナでは内部で何が起こっているのかを確認するのが困難であり、しかもコンテナの寿命は非常に短期です。実際、当社の調査によると、現在70%のコンテナの寿命が5分未満です。従来のセキュリティツールでは、コンテナの内部を確認することも、Kubernetesの動的な性質を処理することも、AWSクラウド、オンプレミス、またはハイブリッド環境におけるスケーリングも行えません。また、プロプライエタリなセキュリティツールは、オープンソースソフトウェアが持つ標準化と技術革新のスピードについていけません。

アマゾン ウェブ サービス（AWS）環境において、コンテナおよびクラウドサービスに関する効率的なセキュリティ管理とコンプライアンス管理を自動化するにはどうすればよいでしょうか？貴社では、Amazon EKS、Amazon ECS、AWS Fargate上でワークロードを正常に実行するために必要となる可視性とセキュリティ管理を確保できていますか？クラウドネイティブ環境向けに構築された適切な統合ツールセットを活用することで、企業や組織は、AWSアカウント、インフラストラクチャー、ワークロードのすべてにおいて、クラウドとコンテナに関するセキュリティリスクを適切に管理できるようになります。

そのためには、クラウドの設定ミスによるリスクを減らすこと、クラウドとコンテナの脆弱性を継続的にスキャンすること、異常なアクティビティを検知すること、脅威を優先順位付けすること、そしてお使いのアプリケーションがそのライフサイクル全体にわたってセキュアであることを保証することが必要となります。これら5つの重要なワークフローを通じて、最も重要なセキュリティと可視性の要件をカバーできるようになります。その結果、AWS上でのコンテナやKubernetesの実行、そしてクラウドサービス運用を自信を持ってセキュアに実現できるようになります。

01
継続的なクラウドセキュリティの実現

設定ミスや不審な動作を即座に特定するためには、継続的なクラウドセキュリティが必要です。責任共有モデルにおいて、こうしたセキュリティ対策を実装し、運用していくのはAWSユーザーの責務です。下記に示す事項は、自社のクラウドセキュリティ体制を検証するのに役立ちます。

- AWS環境（Amazon VPC、Amazon RDS、Amazon S3、Amazon ECS、Amazon EKS、AWS Fargateといったシステムやアプリケーション、サービス）全体におけるクラウドリソースのインベントリを通じて、可視性を向上させましょう。
- CISのベンチマークに照らして、お使いのクラウドの設定を定期的にチェックし、設定ミス（パブリックなストレージバケット、公開されているセキュリティグループ、アクセス制御など）を特定した上で、違反を是正するための措置を取りましょう。
- さらに、AWS CloudTrailのログやポリシーを利用してサービスの変更、疑わしい挙動、潜在的な脅威に関するアラートを発行することで、クラウドアカウント、ユーザー、サービス全体のアクティビティを継続的に監視できます。
- 環境間のセキュリティ制御を標準化した上で、できれば共有ポリシーモデルで一貫したポリシーを適用しましょう。
- また、脆弱性、リアルタイムでの構成変更、リスクの高いIDの挙動、アクティブな脅威といった、最も重大なリスクにつながる検知結果を組み合わせ、優先順位を付けて対応することが重要です。
- 複数のAWSクラウドドメインから得られるコンテキストを組み込むことで、潜在的な攻撃経路を視覚化し、ラテラルムーブメントを検知することも可能です。
- 最小権限の原則を徹底し、クラウドモデルにおけるゼロトラストに準拠することで、過剰な権限を排除しましょう。
- 本番環境での設定ミスをIaC（Infrastructure as Code）マニフェストへとマッピングすることで、ドリフトを減らしましょう。
- クラウドのアクティビティログを解析することで、すべてのクラウドアカウント、ユーザー、サービスにおける予期せぬ変化や疑わしい活動を検知しましょう。

02
ランタイムインテリジェンスに基づいて脆弱性を優先順位付けする

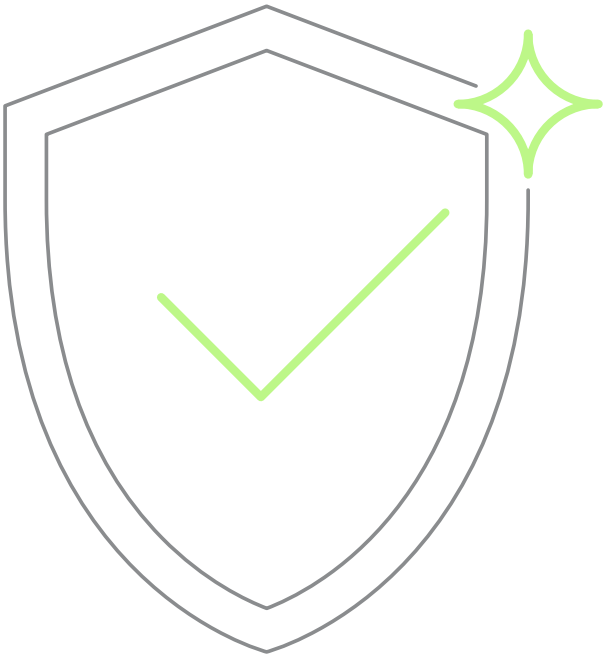
CI/CDパイプラインの普及やコンテナ構築におけるオープンソースソフトウェアの活用が進むにつれ、アプリケーション開発はかつてないスピードで加速しています。その一方で、報告される脆弱性の件数は急増しており、コンテナイメージの増加と本番環境におけるコンテナ稼働は新たな課題を生み出しています。その結果、組織がセキュリティリスクを制御できなくなる可能性が高まっています。特に、効果的な優先順位付けを行えない場合、セキュリティチームは対応すべき課題の多さに圧倒され、開発者は優先度の低い脆弱性への対応に貴重な時間を費やしてしまう恐れがあります。これを防ぐには、脆弱性管理を脆弱性管理をアプリケーションライフサイクル全体にシームレスに組み込むことが不可欠です。以下に、脆弱性がもたらすリスクを制御するための主要な手順に示します。

- 実際に稼働しているパッケージを正確に把握し、その情報に基づいて脆弱性の優先順位付けを自動化することで、対応の効率を大幅に向上させます。
- CI/CDパイプライン（AWS CodePipelineなど）やレジストリ（Amazon ECRなど）にスキャン処理を組み込む、リスクのあるコンテナイメージが本番環境に導入される前にブロックします。
- 起動時の指示、ユーザー権限、シークレットの有無、ラベル情報などをチェックし、ポリシーに準拠した安全なイメージのみを許可します。
- 本番環境に導入されたコンテナに影響を与えるような新しい脆弱性を特定しましょう。
- コンテナだけでなく、ベアメタル、VM、Amazon EC2、クラウドインスタンスといったホスト環境に存在する脆弱性も定期的にスキャンします。
- サーバーレス実行環境であるAWS Fargateに対してイメージスキャンを自動化し、脆弱なコンテナが実行されるリスクを軽減します。
- レイヤード分析を実装することで、脆弱性が導入された具体的なコンテナイメージレイヤーを特定できます。これにより、修正ワークフローの効率化や、責任の所在を明確にすることが可能なります。
- 個々の問題に関して適切なチームにアラートを発行し、CI/CDツールの内部で対応を統合しておきましょう。
- このプロセスにセキュリティ分析とコンプライアンス検証を組み込んでおきましょう。これにより、より早い段階で問題に対処できるようになり、導入のスピードを落とすこともなくなります。これは、「セキュリティのシフトレフト」と呼ばれます。

03
脅威を検知し、対応する

サイバー犯罪は、クラウドネイティブのワークロードやクラウドサービスがもたらす、複雑で拡大し続けるアタックサーフェスにおいて猛威を振るっています。脅威アクターはクラウドの自動化を武器化し、わずか10分以内に攻撃を完了できるケースもあります。インシデントが侵害へと発展するのを防ぐには、攻撃チェーンの初期段階で脅威をリアルタイムで検知する必要があります。これを行うには、コンテキストリッチなイベントデータ、自動化されたアクション、そして高精度なインシデント情報の検索機能が必要です。さらに、コンテナが削除された後でも調査を継続できる仕組みを整えておくことが不可欠です。

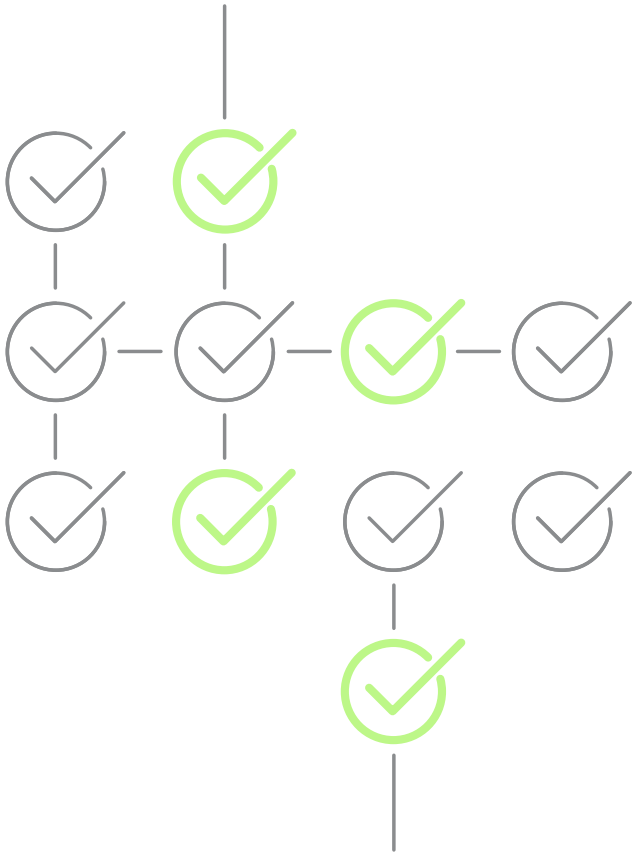
- コンテナ、AWSクラウドサービス、サーバー、そしてユーザーアクティビティ全体にわたって、統一的な脅威検知機能を導入しましょう。
- AWS Fargate上では、サーバーレスアプリケーションに対して、インストルメンテーションを自動化することで、短命なタスクであっても、脅威をリアルタイムで可視化できるようにします。



- さらに利用可能な場合は、厳選されたポリシーを活用し、導入初日から強力な保護を実現して新たな脅威に常に備える状態を維持します。
- 検知ルール、行動分析、機械学習といった複数の検知レイヤーを組み合わせることで、ゼロデイ脅威に対するカバレッジを強化することも重要です。
- コンテナドリフトを防止すること。これにより、攻撃をブロックし、不変性の原則を実施できるようになります。
- 加えて、IDに関連する振る舞いを分析することで、アカウント侵害や権限昇格の初期兆候をいち早く特定できます。このようなアプローチを取ることで、企業や組織は、コンテナやクラウドワークロードに関する脅威に対し、常に一歩先を行くセキュリティ体制を築くことが可能となります。
- システムコールデータ、コマンド、監査ログ、その他のアクティビティをキャプチャし、それらをAmazon EKSやAWSクラウドのコンテキストでリッチ化して活用しましょう。
- AWS CloudTrailのログフィルタリング機能を自動化することで、予期しないサービスアクティビティや構成変更を検知できます。
- さらに、Amazon EKSで利用可能なKubernetesネイティブのコントロールを活用することで、コンテナワークロードのランタイム保護を実現し、最小権限の原則に基づいたネットワークポリシーを実装することが可能です。
- コンテナ関連のセキュリティインシデントについて重要な質問に迅速に回答できる仕組みを整えることで、コンテナが削除された後でもインシデントへの対応や分析、根本原因の特定が行えます。
- セキュリティイベントをIDやその他の重要な調査結果と相互に関連付けることで、調査を加速し、インシデント対応を効率化することも重要です。
- また、AIを戦略的に導入すれば、セキュリティイベントを分析し、コンテキストアウェアネスを通じて人間による対応を迅速化できます。

04
コンプライアンスの継続的な検証

コンテナ、Kubernetes、AWSクラウド環境全体におけるコンプライアンス基準（CIS、SOC2、PCI、NIST800-53など）を満たすためのコンプライアンスチェックを実施しましょう。また、クラウドサービスを継続的に監視することで、コンプライアンスに影響を与える可能性のある「構成ドリフト」が発生していないかどうかを調べます。さらに、定期的な評価と詳細なレポートを通じて、コンプライアンスの進捗状況を測定します。



- 加えて、CISベンチマークやAWS、Docker、Kubernetesなどの業界のベストプラクティス（に照らしてクラウドコントロールプレーンやコンテナ化されたアプリケーション、プラットフォームの構成を継続的にチェックしましょう。
- ビルド時にコンプライアンスを検証しましょう。これは、コンテナイメージのスキャンポリシーを標準規格（NIST、PCI、SOC2、HIPAAなど）や社内コンプライアンスポリシー（ブラックリストに載ったイメージ、パッケージ、ライセンスなど）にマッピングすることで可能となります。
- さらに、AWS CloudTrailを使用してAWSクラウドサービス全体における構成やポリシーの変更を監視するこおも欠かせません。
- セキュリティ標準に対応した豊富なFalcoルールのセットを通じて、実行時にコンプライアンスを管理しましょう。
- 重要なシステムファイルやディレクトリの改竄、および不正な変更を検知するためにFileintegrity Monitoring（FIM）を実装しましょう。
- 自動化により、手動プロセスを排除し、自動修復によりコンプライアンスを実現しましょう。これは、本番環境での設定ミスをIaC（Infrastructure as Code）マニフェストへとマッピングすることで可能となります。
- クラウド監査ログとコンテナのフォレンジックデータを使用して、クラウドおよびコンテナのコンプライアンスに関する証明を提示しましょう。

05
コンテナ、Amazon EKS、AWSクラウドサービス、のモニタリングとトラブルシューティング

コンテナは短命で動的に変化し続けるため、コンテナが消失すると内部のすべての情報も同時に失われます。その結果、ユーザーによるセキュアシェル（SSH）の利用やログの確認は不可能となり、モノリシックなアプリケーションで使用されてきた従来型のツールの多くは問題発生時に機能しなくなります。AWSワークロードとインフラストラクチャの健全性やパフォーマンスを可視化することは、クラウドアプリケーションの可用性を確保する上で不可欠です。

- オープンソースのPrometheusやAWS CloudWatchといったクラウド監視の標準を活用することで、AWS Fargate、Amazon S3、Amazon RDS、AWS Lambdaなどのサービスに対する可観測性を実現できます。これらを組み合わせることで、より深いインサイトを得ることが可能になります。

- コンテナベースのアプリケーションの動的な性質をモニタリングすることは、クラウドサービスの高可用性とパフォーマンスにとって不可欠です。
- マイクロサービスベースのアプリケーションは複数のインスタンスに分散できるほか、コンテナはAWSクラウド、オンプレミス、ハイブリッドインフラストラクチャー全体を移動できます。
- Kubernetesオーケストレーションの状態をモニタリングすることは、Kubernetesがすべてのサービスインスタンスを稼働させ続けているかどうかを理解する上で極めて重要です。
  - さらに、インフラストラクチャー、サービス、アプリケーションにわたる詳細な可視性を通じて健全性とパフォーマンスを監視しましょう。Kubernetesのオーケストレーション監視機能を活用すれば、クラスター全体の運用状況を把握できます。
  - コンテナやクラウドのコンテキストを利用して、オーナーを即座に特定でき、問題解決に結びつけます。過剰なリソースを消費しているPodを特定し、キャパシティ制限を監視しましょう。オートスケーリングの挙動をモニタリングすることで、予期せぬ課金、アプリケーションのロールアウト、デプロイメントのロールバックを制御します。
  - クラスター全体におけるキャパシティを最適化することで、コストを削減します。
- コンテナに関するディープな可視性と、Kubernetesとクラウドのコンテキストでリッチ化されたきめ細かなメトリクスにより、アプリケーションのパフォーマンスを改善し、問題を迅速に解決しましょう。これにより、所定のセキュリティインシデントがサービスの可用性に与える影響をモニタリングできるようになります。

クラウド環境では一秒一秒が重要です。Sysdigは、ランタイムインサイトとオープンソースのFalcoを活用し、リスクの変化を瞬時に検知してクラウド攻撃をリアルタイムで阻止します。さらに、ワークロード、ID、サービス間のシグナルを相互に関連付けることで、隠れた攻撃経路を可視化し、最も重要なリスクに優先順位を付けることができます。

Sysdig. Secure Every Second.

デモを依頼 →

sysdig

アマゾン ウェブ サービス（AWS）クラウドとコンテナを保護するための5つのベストプラクティス

COPYRIGHT © 2021-2024 SYSDIG,INC.
ALL RIGHTS RESERVED.
CL-010-JA REV. F 09/25