セキュリティの柱

AWS Well-Architected Framework

セキュリティの柱: AWS Well-Architected Framework

Table of Contents

セキュリティの柱 - AWS Well-Architected Framework

公開日: 2023 年 12 月 6 日 (改訂履歴 (p. 137))

このホワイトペーパーは、AWS Well-Architected フレームワークのセキュリティの柱に焦点を当てています。お客様が安全な AWS ワークロードの設計、配信、メンテナンスにベストプラクティスと最新の推奨事項を適用するうえで役立つガイダンスを提供します。

はじめに

AWS Well-Architected フレームワークは、AWS でワークロードを構築するための意思決定におけるトレードオフの理解に役立ちます。このフレームワークを使用すれば、信頼性、安全性、効率性、コスト効率に優れ、持続可能なワークロードを、クラウド内で設計および運用するためのアーキテクチャ上の最新のベストプラクティスを学ぶことができます。このフレームワークにより、ワークロードをベストプラクティスに照らし合わせて一貫的に測定し、改善点を特定することが可能となります。Well-Architected ワークロードを備えることによって、ビジネス成功の可能性が大幅に高まると私たちは確信しています。

このフレームワークは次の 6 つの柱に基づいています。

- オペレーショナルエクセレンス
- セキュリティ
- 信頼性
- パフォーマンス効率
- コスト最適化
- サステナビリティ

このホワイトペーパーは、セキュリティの柱に焦点を当てています。これを理解して最新の AWS の推奨事項に従うことで、ビジネス要件および規制要件を満たすことができます。この内容は、最高技術責任者 (CTO)、最高情報セキュリティ責任者 (CSO/CISO)、設計者、開発者、オペレーションチームメンバーなどの技術担当者を対象にまとめられています。

このホワイトペーパーを読むことで、セキュリティを念頭に置いてクラウドアーキテクチャを設計するための AWS の最新の推奨事項と戦略を理解できます。ここでは、実装の詳細やアーキテクチャのパターンについては説明していませんが、そのような情報に該当するリソースへの参照が記載されています。このホワイトペーパーにある手法を採用すれば、データとシステムを保護し、アクセスをコントロールし、セキュリティイベントに自動的に応答するアーキテクチャを構築できます。

セキュリティの基礎

セキュリティの柱は、クラウドテクノロジーを活用し、セキュリティ体制の向上を可能にするやり方でデータ、システム、資産を保護する方法を表します。このホワイトペーパーでは、AWS で安全なワークロードを設計するための詳細なベストプラクティスガイダンスを提供します。

設計原則

クラウドには、ワークロードのセキュリティ強化に役立つ多くの原則があります:

- 強力なアイデンティティ基盤を実装する: 最小権限の原則を導入し、各 AWS リソースとの通信における適切な認可のもと、役割分担を徹底させます。ID 管理を一元化し、長期間にわたって一つの認証情報を使用し続けないようにします。
- トレーサビリティの維持: ご使用の環境に対して、リアルタイムでモニタリング、アラート、監査のアクション、および変更を行います。ログとメトリクスの収集をシステムに統合して、自動的に調査しアクションを実行します。
- すべてのレイヤーでセキュリティを適用する: 複数のセキュリティコントロールを使用して深層防御アプローチを適用します。ネットワークのエッジ、VPC、ロードバランシング、すべてのインスタンスとコンピューティングサービス、オペレーティングシステム、アプリケーション、コードなど、すべてのレイヤーに適用します。
- セキュリティのベストプラクティスを自動化する: 自動化されたソフトウェアベースのセキュリティメカニズムにより、安全で、より速く、より費用対効果の高いスケーリングが可能になります。バージョン管理されているテンプレートにおいてコードとして定義および管理されるコントロールを実装するなど、セキュアなアーキテクチャを作成します。
- 伝送中および保管中のデータを保護する: データを機密性レベルに分類し、暗号化、トークン分割、アクセスコントロールなどのメカニズムを適宜使用します。
- データに人の手を入れない: データに直接アクセスしたりデータを手動で処理したりする必要を減らしたり、排除したりするメカニズムとツールを使用します。これにより、機密性の高いデータを扱う際の誤処理、改変、ヒューマンエラーのリスクを軽減します。
- セキュリティイベントに備える: 組織の要件に合わせたインシデント管理および調査のポリシーとプロセスを導入し、インシデントに備えます。インシデント対応シミュレーションを実行し、ツールとオートメーションにより、検出、調査、復旧のスピードを上げます。

定義

クラウドのセキュリティには、次の 7 つの領域があります。

責任共有モデル

セキュリティとコンプライアンスは、AWS とお客様との間で共有される責任です。この共有モデルは、ホストオペレーティングシステムや仮想化レイヤーから、サービスが運用されている施設の物理的なセキュリティに至るまで、さまざまなコンポーネントを AWS が運用、管理、およびコントロールするため、お客様の運用負担を軽減することができます。お客様は、AWS が提供するセキュリティグループファイアウォールの設定に加え、ゲストオペレーティングシステム (アップデートおよびセキュリティパッチを含む) およびその他の関連アプリケーションソフトウェアの責任と管理を負うものとします。お客様の責任範囲は、使用するサービス、IT 環境へのサービス統合、適用可能な法律および規制に応じて異なります。したがって、お客様は選択するサービスを注意深く検討する必要があります。また、この責任共有モデルの性質によって柔軟性が得られ、お客様がデプロイを統制できます。下の図に示すように、この責任分担は一般に、クラウド「の」セキュリティとクラウド「内の」セキュリティと呼ばれます。

AWS の責任「クラウドのセキュリティ」 – AWS は、AWS クラウドで提供されるすべてのサービスを実行するインフラストラクチャの保護について責任を負います。このインフラストラクチャは、AWS クラウドサービスを実行するハードウェア、ソフトウェア、ネットワーク、および施設で構成されています。

お客様の責任「クラウド内のセキュリティ」 – お客様の責任は、お客様が選択した AWS クラウドサービスに応じて異なります。これにより、お客様がセキュリティの責任の一部として実行する必要がある設定作業の量が決まります。例えば、Amazon Elastic Compute Cloud (Amazon EC2) などのサービスは、Infrastructure as a Service (IaaS) に分類されるため、お客様は必要なセキュリティ設定と管理タスクをすべて実行する必要があります。Amazon EC2 インスタンスをデプロイするお客様は、ゲストオペレーティングシステムの管理 (アップデートやセキュリティパッチの適用を含む)、お客様が各インスタンスにインストールしたアプリケーションソフトウェアやユーティリティの管理、AWS が提供する各インスタンスのファイアウォール (セキュリティグループ) の設定について責任を負います。Amazon S3 や Amazon DynamoDB などの抽象化されたサービスについては、インフラストラクチャレイヤー、オペレーティングシステム、プラットフォームの運用を AWS が行い、お客様はエンドポイントにアクセスしてデータを保存、取得します。お客様は、データの管理 (暗号化オプションを含む)、アセットの分類、および IAM ツールを使用した適切なアクセス許可の適用について責任を負います。

図 1: AWS 責任共有モデル。

このお客様/AWS の責任分担モデルは IT 統制にも拡張されます。IT 環境を運用する責任がお客様と AWS との間で分担されるのと同様に、IT 統制の管理、運用、および検証も分担されます。AWS は、これまで

お客様が管理していた AWS 環境にデプロイされた物理インフラストラクチャに関連する制御を管理することで、お客様の運用管理の負担を軽減することができます。お客様によって AWS のデプロイ方法は異なるため、特定の IT 統制の管理を AWS に移行することで、（新たな）統制環境の分散化を図るというメリットが得られます。その後、お客様は必要に応じて、利用可能な AWS の統制とおよびコンプライアンス文書を使用して、統制の評価および検証手順を実行することができます。以下は AWS と AWS のお客様、またはその両方によって管理されるコントロールの例です。

継承された統制 – お客様が AWS から完全に継承するコントロール。

- 物理統制と環境統制

共有コントロール – インフラストラクチャ層とお客様層の両方に適用されるコントロール。ただしコンテキストまたは観点は異なる。共有コントロールでは、AWS はインフラストラクチャの要件を提供し、お客様は AWS サービスの使用中に独自のコントロール実装を提供する必要があります。以下に例を示します。

- パッチ管理 – AWS はインフラストラクチャ内の欠陥のパッチ適用と修正に責任を持ちますが、ゲストオペレーティングシステムおよびアプリケーションにパッチを適用する責任はお客様にあります。
- 設定管理 – AWS はインフラストラクチャデバイスの設定の責任を負いますが、ゲストオペレーティングシステム、データベース、およびアプリケーションの設定はお客様ご自身の責任で行います。
- 意識向上およびトレーニング – AWS は AWS の従業員にトレーニングを提供しますが、お客様は自社の従業員をトレーニングする必要があります。

カスタマー特有 – お客様が AWS のサービス内にデプロイするアプリケーションに基づき、お客様が単独で責任を負うべきコントロール。以下に例を示します。

- サービスおよびコミュニケーション保護またはゾーンセキュリティでは、お客様が特定のセキュリティ環境下でデータのルーティングやゾーニングを行わなければならない場合があります。

ガバナンス

セキュリティガバナンスは、全体的なアプローチのサブセットとして、リスク管理を支援するためのポリシーと管理目標を定義することによって、ビジネス目標をサポートすることを目的としています。セキュリティ管理目標に対して階層的アプローチ (各レイヤーが前のレイヤーの上に構築される) を取ることにより、リスク管理を達成します。AWS 共有責任モデルを理解することが基礎のレイヤーとなります。この知識により、お客様側での責任は何か、AWS から何を継承するかが明確になります。有用なリソースは AWS Artifact でAWS のセキュリティおよびコンプライアンスレポートへのオンデマンドアクセスを付与し、オンライン契約を選択します。

コントロールの目的のほとんどは、次のレイヤーで満たします。プラットフォーム全体の機能が備わっているのはここになります。例えば、このレイヤーには AWS アカウントの販売プロセス、AWS IAM Identity Center などの ID プロバイダーとの統合、共通の検出制御などが含まれます。プラットフォームガバナンスプロセスのアウトプットの一部もここにあります。新しい AWS サービスを使って開始する場合、AWS Organizations サービスでサービスコントロールポリシー (SCP) を更新し、サービスの初期使用のためのガードレールを提供します他の SCP を使用して、一般的にセキュリティ不変条件と呼ばれる共通のセキュリティ制御目標を実装できます。これらは、複数のアカウント、組織単位、または AWS 組織全体に適用する管理目標または設定です。典型的な例としては、インフラストラクチャが実行されるリージョンを制限したり、検知コントロールの無効化を防いだりすることが挙げられます。この中間レイヤーには、設定ルールやパイプラインのチェックなど体系化されたポリシーも含まれています。

最上位のレイヤーは、製品チームが管理目標を達成する場所です。これは、製品チームがコントロールするアプリケーションで実装が行われるためです。これは、アプリケーションでの入力確認の実施や、マイクロサービス間で ID が正しく受け渡しされるのを保証することなどが考えられます。設定を担当するのは製品チームですが、中間レイヤーから一部の機能を継承できます。

コントロールを実装する場所がどこであろうと目的は同じで、すなわち「リスク管理」です。一連のリスク管理フレームワークは、特定の業界、リージョン、またはテクノロジーに適用されます。主な目標は「可能性と結果に基づいてリスクを強調する」です。つまり 内在するリスクです.そして、可能性、結果、またはその両方を低減させる管理目標を定義することができます。そして、管理策を実施することで、結果としてどのようなリスクが生じるかを確認することができます。つまり 残存リスクです.管理目標は 1 つまたは複数のワークロードに適用できます。次の図は、典型的なリスクマトリックスを示しています。可能性は過去の発生頻度に基づき、結果はイベントの金銭的、風評的、時間的コストにに基づいています。

可能性	リスクレベル				
非常に 可能性が高い	低	中	高	重要	重要
高い	低	中	中	高	重要
可能性がある	低	低	中	中	高
低い	低	低	中	中	高
非常に 可能性が低い	低	低		中	高
結果	最小限	低	中	高	重大

図 2: リスクレベルの可能性マトリックス

AWS アカウントの管理と分離

AWS では、社内のレポート構造を流用せずに、個別アカウントごとにワークロードを整理し、機能、コンプライアンス要件、共通のコントロールセットに基づいてアカウントをグループ化することを推奨しています。AWS では、アカウントが強固な境界となります。例えば、開発およびテストのワークロードと本番ワークロードを切り離すために、アカウントレベルの分離を強く推奨しています。

アカウントを一元的に管理する: AWS Organizations は、AWS アカウントの作成と管理、およびアカウント作成後の制御を自動化します。AWS Organizations を使用してアカウントを作成する場合、使用する E メールアドレスの検討が重要です。これがパスワードリセットを許可するルートユーザーとなるためです。Organizations は、ワークロードの要件と目的に応じた異なる環境である 組織単位 (OU)でアカウントをグループ化できます。

制御を一括設定する: 特定のサービス、リージョン、サービスアクションのみを適切なレベルで許可することによって、AWS アカウントが実行できる操作を制御します。AWS Organizations では、サービスコントロールポリシー (SCP) を使用して、組織レベル、組織単位、アカウントレベルでアクセス許可ガードレールを適用できます。これは、すべての AWS Identity and Access Management (IAM) ユーザーおよびロールに適用されます。例えば、SCP を適用して、明示的に許可されていないリージョン内のユーザーがリソースを起動することを制限できます。AWS Control Tower では、複数アカウントの効率的な設定と管理が可能です。このサービスを使うと、AWS Organization のアカウント設定の自動化、プロビジョニングの自動化、 ガードレール (予防や検出など) の適用、ダッシュボードによる可視化を実現できます。

サービスとリソースを一括設定する: AWS Organizations では、すべてのアカウントに適用する AWS サービス を設定できます。例えば、組織全体で実行されるすべてのアクションの集中ログ記録を AWS CloudTrailで設定し、メンバーアカウントによるログの無効化を防止できます。また、 AWS Config を使用して定義したルールを基にデータを一元的に集約することもできます。これによってワークロードのコンプライアンス監査と、変更への迅速な対応が可能となります。AWS CloudFormation StackSets を使用す

ると、組織内の複数のアカウントと OU にまたがる AWS CloudFormation スタックを集中管理できます。これによって、新しいアカウントを自動的にプロビジョニングしてセキュリティ要件を満たすことができます。

セキュリティサービスの委任管理機能を使用して、管理に使用されるアカウントを組織の請求 (管理) アカウントから分離します。GuardDuty、Security Hub、AWS Config などいくつかの AWS のサービスでは、管理機能に特定のアカウントを指定するなど、AWS Organizations との統合をサポートします。

ベストプラクティス

SEC01-BP01 アカウントを使用してワークロードを分ける:

マルチアカウント戦略を取り、環境 (本番稼働、開発、テストなど) とワークロードの間に共通ガードレールを構成し、分離を確立します。アカウントレベルの分類は、セキュリティ、請求、アクセスのために強力な分離境界を提供するため、強く推奨されます。

期待される成果: クラウドオペレーション、無関係のワークロード、環境を別々のアカウントに分類し、クラウドインフラストラクチャ全体のセキュリティを向上させるアカウント構造。

一般的なアンチパターン:

- データ重要度レベルの異なる複数の無関係のワークロードを同一アカウントに配置する。
- きちんと定義されていない組織単位 (OU) 構造。

このベストプラクティスを活用するメリット:

- 誤ってワークロードにアクセスした場合の影響範囲が抑えられます。
- AWS サービス、リソース、およびリージョンへのアクセスの一元的ガバナンス。
- ポリシーとセキュリティサービスの一元管理により、クラウドインフラストラクチャのセキュリティを維持する。
- アカウント作成とメンテナンスプロセスの自動化。
- コンプライアンスや規制要件に対応した、インフラストラクチャの集中監査。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

AWS アカウント は、さまざまなデータ重要度レベルで稼働するワークロードまたはリソース間にセキュリティ分離境界を提供します。AWS は、マルチアカウント戦略を通して大規模にクラウドワークロードを管理し、この分離境界を活用するためのツールを提供します。AWS でのマルチアカウント戦略のコンセプト、パターン、および実装に関するガイダンスについては、「Organizing Your AWS Environment Using Multiple Accounts」(複数のアカウントを使用した AWS 環境の組織化) を参照してください。

一元管理下に複数の AWS アカウント がある場合、アカウントを組織単位 (OU) の層によって定義された階層に組織化する必要があります。次に、OU とメンバーアカウントに対してセキュリティ管理を組織化して適用することにより、組織内のメンバーアカウントに対して一貫性のある予防的制御を確立できます。セキュリティ管理は継承されるため、OU 階層の下位レベルにあるメンバーアカウントに対するアクセス許可をフィルタリングすることができます。優れた設計によりこの継承を利用して、各メンバーアカウントに対して望ましいセキュリティ管理を達成するのに必要なセキュリティポリシーの件数と複雑性を軽減します。

AWS Organizations および AWS Control Tower は、AWS 環境でこのマルチアカウント構造を実装および管理するのに使用できる 2 つのサービスです。AWS Organizations では、1 つまたは複数の OU 層 (それぞれに複数のメンバーアカウントを含む) で定義された階層にアカウントを組織化できます。サービス管理ポリシー (SCP) により、組織管理者がメンバーアカウントできめ細やかな予防的コントロールを確立し、AWS Config はメンバーアカウントに関して積極的かつ検出的コントロールを確立するのに使用できます。多くの AWS サービス が AWS Organizations と統合し、組織内のすべてのメンバーアカウントで、委任管理制御とサービス固有のタスク実行を提供します。

AWS Organizations 上の層にある AWS Control Tower は、ランディングゾーンにマルチアカウント AWS 環境向けの ワンクリックベストプラクティスセットアップを提供します。ランディングゾーンは、Control Tower によって確立されるマルチアカウント環境への入口です。Control Tower には、AWS Organizations と比較していくつかの利点があります。アカウントガバナンスを改善する 3 つの利点には次のようなものがあります。

- 組織に対して承認されたアカウントに自動適用される、統合された必須のセキュリティガードレール。
- 所定の OU セットに対してオン/オフと切り替えられるオプションのガードレール。
- AWS Control Tower Account Factory では、事前承認されたベースラインと組織内の構成オプションを含むアカウントを自動的にデプロイできます。

実装手順

1. 組織単位構造を設計する: 組織単位を適切に設計することにより、サービスコントロールポリシーやその他のセキュリティコントロールの作成と保守に必要な管理負担を軽減できます。組織単位構造は、貴社のビジネスニーズ、データ重要度、およびワークロード構造に合致したものである必要があります。
2. マルチアカウント環境向けのランディングゾーンを作成する: ランディングゾーンは一貫したセキュリティとインフラストラクチャ基盤を提供します。そこから組織はワークロードを迅速に開発、立ち上げ、デプロイできます。カスタムビルドのランディングゾーンまたは AWS Control Tower を使用して、環境のオーケストレーションを実行できます。
3. ガードレールを確立する: ランディングゾーンを通して環境に一貫性のあるセキュリティガードレールを実装します。AWS Control Tower は、必須とオプションのコントロールのリストを提供します。必須コントロールは、Control Tower 実装時に自動的にデプロイされます。強く推奨されたコントロールとオプションのコントロールのリストを確認し、ニーズに適したコントロールを実装します。
4. 新しく追加されたリージョンへのアクセスを制限する: 新しい AWS リージョン について、ユーザーやロールなどの IAM リソースは、指定したリージョンのみに伝播されます。このアクションは、Control Tower 使用時はコンソール経由で、または AWS Organizations で IAM アクセス許可を調整することにより実行できます。
5. AWSCloudFormation StackSets を検討する: StackSets を使用すると、IAM ポリシー、ロール、グループなどのリソースをさまざまな AWS アカウント とリージョンに承認されたテンプレートからデプロイしやすくなります。

リソース

関連するベストプラクティス:

- SEC02-BP04 一元化された ID プロバイダーを利用する (p. 28)

関連するドキュメント:

- AWS Control Tower
- AWS セキュリティ監査のガイドライン
- IAM ベストプラクティス
- CloudFormation StackSets を使用して、複数の AWS アカウント とリージョン全体にリソースをプロビジョニングする

- 組織関連の FAQ
- AWS Organizations 用語およびコンセプト
- AWS Organizations マルチアカウント環境のサービスコントロールポリシーのためのベストプラクティス
- AWS アカウント管理リファレンスガイド
- 複数のアカウントを使用した AWS 環境の組織化

関連動画:

- 自動化とガバナンスにより AWS の大規模な採用を可能にする
- Well-Architected の手法によるセキュリティのベストプラクティス
- AWS Control Tower を使って複数のアカウントをビルドおよび統制する
- 既存の組織に対して Control Tower を有効化する

関連ワークショップ:

- Control Tower Immersion Day

SEC01-BP02 セキュアアカウントのルートユーザーおよびプロパティ

ルートユーザーは AWS アカウント で最も権限が高いユーザーであり、アカウント内の全リソースに対する完全な管理者アクセスがあるだけでなく、場合によってはセキュリティポリシーによる制限の対象外となります。ルートユーザーへのプログラムによるアクセスを無効化し、ルートユーザーに対する適切なコントロールを確立し、さらにルートユーザーの定期的使用を避けることにより、ルート認証情報を不用意に曝露するリスク、それによるクラウド環境の侵害を軽減することができます。

期待される成果: ルートユーザーをセキュリティ保護することにより、ルートユーザー認証情報を不正使用した場合の偶発的または意図的な損害が生じる可能性が低減されます。検出コントロールを確立することによっても、ルートユーザーを使ったアクションが取られると適切な担当者にアラートを送信できます。

一般的なアンチパターン:

- ルートユーザー認証情報を必要とする少数以外のタスクに対してもルートユーザーを使用する。
- 緊急時に重要なインフラストラクチャ、プロセス、担当者が正常に機能するかどうかを検証するために、定期的な緊急時対応計画のテストを怠っている。
- 典型的なアカウントログインフローのみを考慮し、代替アカウント回復方法を考慮することも、テストすることもしていない。
- DNS、E メールサーバー、および携帯電話会社がアカウント復旧フローで使用されるにもかかわらず、重要なセキュリティ境界の一部として対処していない。

このベストプラクティスを活用するメリット: ルートユーザーへのアクセスを確保することにより、アカウントでアクションをコントロールおよび監査できるという安心感が向上する。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

AWS は、アカウントを保護するのに役立つ多くのツールを提供しています。ただし、これらの対策の一部は既定では有効になっていないため、実装するには直接的な措置を講じる必要があります。これらの推

奨事項を、AWS アカウント をセキュリティ保護するための基本的なステップと考えてください。これらのステップを実装する際、セキュリティ管理を継続的に評価およびモニタリングすることが重要となります。

AWS アカウント を初めて作成する際は、アカウント内のすべての AWS のサービスとリソースに完全なアクセス許可を持つ 1 つの ID から始めます。この ID は、AWS アカウント のルートユーザーと呼ばれます。アカウント作成に使用した E メールアドレスとパスワードを使用すれば、ルートユーザーとしてログインできます。AWS ルートユーザーに付与されるアクセス許可が昇格したため、特にそれを必要とするタスクを実行する AWS ルートユーザーの使用は制限する必要があります。ルートユーザーのログイン認証情報は注意して保護し、AWS アカウント ルートユーザーに対しては多要素認証 (MFA) を必ず有効にしておく必要があります。

ユーザー名、パスワード、多要素認証 (MFA) デバイスを使用してルートユーザーにログインする通常の認証フローに加えて、アカウントに関連付けられた E メールアドレスと電話番号にアクセスし、AWS アカウント ルートユーザーにログインするためのアカウント復旧フローもあります。そのため、復旧メールを送信するルートユーザーの E メールアカウントと、そのアカウントに関連する電話番号をセキュリティ保護することも同程度に重要となります。また、ルートユーザーに関連付けられた E メールアドレスが、同じ AWS アカウント の E メールサーバーやドメインネームサービス (DNS) リソースでホストされている場合、潜在的な循環依存性についても考慮する必要があります。

AWS Organizations を使用する場合、それぞれにルートユーザーが含まれる AWS アカウント が複数あります。1 つのアカウントを管理アカウントに指定し、その管理アカウントの下に何層ものメンバーアカウントを追加することができます。管理アカウントのルートユーザーのセキュリティ保護を優先してから、メンバーアカウントのルートユーザーに対処してください。管理アカウントのルートユーザーをセキュリティ保護する戦略は、メンバーアカウントのルートユーザーとは異なり、メンバーアカウントのルートユーザーに対しては予防的なセキュリティコントロールを講じることができます。

実装手順

ルートユーザーのコントロールを確立するには、次の実装ステップが推奨されます。該当する場合、推奨事項は CIS AWS Foundations ベンチマークバージョン 1.4.0 に相互参照されます。AWS アカウント およびリソースのセキュリティ保護については、これらのステップに加え、AWS ベストプラクティスガイドラインも参照してください。

予防的コントロール

1. アカウントに対して、正確な 連絡先情報を設定します。
   a. この情報は、紛失したパスワードの復旧フロー、紛失した MFA デバイスアカウントの復旧フロー、およびチームとの重要なセキュリティ関連のコミュニケーションに使用されます。
   b. 企業ドメインによってホストされた E メールアドレスを使用します (ルートユーザーの E メールアドレスとしては、できれば配布リストのほうが望ましい)。個人の E メールアカウントではなく配布リストを使うことにより、長期的にはルートアカウントへのアクセスに対して冗長性と継続性を追加することになります。
   c. 連絡先情報に記載された電話番号は、この目的専用の安全なものである必要があります。この電話番号をどこかに記載したり、誰かと共有したりしないでください。
2. ルートユーザーにはアクセスキーを作成しないでください。アクセスキーが存在する場合は、それを削除します (CIS 1.4)。
   a. ルートユーザーに対する長期保存可能なプログラム認証情報 (アクセスキーとシークレットキー) は排除します。
   b. ルートユーザーのアクセスキーがすでにある場合、それらのキーを使うプロセスを、AWS Identity and Access Management (IAM) ロールからの臨時アクセスキーを使い、次に ルートユーザーのアクセスキーを削除することにより、移行させる必要があります。
3. ルートユーザーの認証情報を保管する必要があるかどうかを決定します。
   a. AWS Organizations を使用して新しいアカウントを作成している場合、新規メンバーアカウントのルートユーザーの初期パスワードはランダムな値に設定され、決して公開されることはありません。

必要に応じ、AWS Organization 管理アカウントからのパスワードリセットフローを使って、メンバーアカウントへのアクセスを獲得することを検討してください。

b. スタンドアロン AWS アカウント または管理 AWS Organization アカウントに対しては、ルートユーザーの認証情報を作成して安全に保管することを検討してください。ルートユーザーの MFA を有効にする

4. AWS マルチアカウント環境のメンバーアカウントのルートユーザーに対しては、予防的コントロールを有効にします。
   a. メンバーアカウントに対して、ルートユーザー向けのルートアクセスキーの作成を許可しない予防的ガードレールの有効化を検討してください。
   b. メンバーアカウントに対して、ルートユーザーとしてのアクションを許可しない予防的ガードレールの有効化を検討してください。
5. ルートユーザーの認証情報が必要な場合:
   a. 複雑なパスワードを使用します。
   b. ルートユーザー、特に AWS Organizations 管理 (支払者) アカウント (CIS 1.5) に対しては多要素認証 (MFA) を有効化します。
   c. 回復力とセキュリティのために、ハードウェア MFA デバイスを検討してください。これは、単回使用デバイスを使用することにより、MFA コードを含むデバイスが他の目的に再使用される可能性が少なくなるためです。電池式のハードウェア MFA デバイスが定期的に交換されていることを検証してください。(CIS 1.6)
      - ルートユーザーに対して MFA を設定するには、仮想 MFA または ハードウェア MFA デバイスのいずれかを有効化する手順に従ってください。
   d. バックアップ用に複数の MFA デバイスを登録することを検討してください。 アカウントごとに最大 8 台の MFA デバイスを登録できます。
      - ルートユーザーに対して複数の MFA デバイスを登録すると、MFA デバイス紛失時にアカウントを復旧するフローが無効になることに注意してください。
   e. パスワードは安全に保管し、電子的にパスワードを保管する際は循環依存関係を検討してください。入手するために同じ AWS アカウント へのアクセスが必要となる方法でパスワードを保管しないでください。
6. オプション: ルートユーザーに対して定期的なパスワードローテーションスケジュールを設定することを検討します。
   - 認証情報管理のベストプラクティスは、規制およびポリシー要件によって異なります。MFA によって保護されるルートユーザーは、認証の単一要素としてパスワードに依存しません。
   - 定期的にルートユーザーパスワードを変更することにより、誤って露出したパスワードが不正使用されるリスクを低減します。

検出コントロール

- ルート認証情報の使用を検出するアラームを作成します (CIS 1.7)。Amazon GuardDuty を有効にすることにより、RootCredentialUsage 所見を使ってルートユーザー API 認証情報の使用をモニタリングおよびアラートを発行します。
- AWS Config 用の AWS Well-Architected セキュリティの柱コンフォーマンスパックに含まれる検出コントロール、またはAWS Control Tower を使用している場合は、Control Tower 内にある強く推奨されるコントロールを評価および実装します。

運用ガイダンス

- 組織で、ルートユーザー認証情報へのアクセスが必要な担当者を決定します。
  - 1 人の担当者がすべての必要な認証情報とルートユーザーアクセスを取得するために MFA にアクセスするのを回避するため、2 人制を採用します。
  - アカウントに関連付けられた電話番号と E メールエイリアス (パスワードリセットと MFA リセットフローに使用される) は、個人ではなく、組織が管理するよう徹底してください。

- ルートユーザーは例外的にのみ使用します (CIS 1.7)。
  - AWS のルートユーザーを、たとえ運営業務であっても日常的なタスクに使用してはなりません。ルートユーザーを必要とする AWS タスクを実行するには、ルートユーザーとしてのみログインしてください。その他すべてのアクションは、適切なロールを持つ他のユーザーが実行しなければなりません。
- ルートユーザーにアクセスできることを定期的にチェックし、ルートユーザー認証情報を使用する必要がある緊急事態の前に手順をテストしておきます。
- アカウントに関連付けられた E メールアドレスと、その他の連絡先に記載された E メールアドレスが有効であることを定期的にチェックします。これらの E メールの受信箱に、 <abuse@amazon.com>から受信したセキュリティ通知が届いていないかどうかモニタリングしてください。また、アカウントに関連付けられた電話番号があれば、それが通じることも確認してください。
- ルートアカウントの不正使用に対処するインシデント対応手順を準備しておきます。AWS アカウントに対するインシデント対応戦略の策定に関する詳細については、AWS セキュリティインシデント対応ガイドと、セキュリティの柱のホワイトペーパーの「インシデント対応」セクションに記載されたベストプラクティスを参照してください。

リソース

関連するベストプラクティス:

- SEC01-BP01 アカウントを使用してワークロードを分ける: (p. 6)
- SEC02-BP01 強力なサインインメカニズムを使用する (p. 20)
- SEC03-BP02 最小特権のアクセスを付与します (p. 36)
- SEC03-BP03 緊急アクセスのプロセスを確立する (p. 39)
- SEC10-BP05 アクセスを事前プロビジョニングする (p. 110)

関連するドキュメント:

- AWS Control Tower
- AWS セキュリティ監査のガイドライン
- IAM ベストプラクティス
- Amazon GuardDuty – ルート認証情報使用アラート
- CloudTrail によるルート認証情報使用モニタリングに関するステップバイステップガイダンス
- AWS での使用が認可された MFA トークン
- AWS に break glass アクセスを実装する
- AWS アカウント を改善するためのトップ 10 セキュリティアイテム
- AWS アカウント の不正なアクティビティに気付いた場合はどうすればよいですか?

関連動画:

- 自動化とガバナンスにより AWS の大規模な採用を可能にする
- Security Best Practices the Well-Architected Way (Well-Architected の手法によるセキュリティのベストプラクティス)
- AWS re:inforce 2022 – Security best practices with AWS IAM からの「Limiting use of AWS root credentials (AWS ルート認証情報の使用を制限する)」

関連する例とラボ:

- ラボ: AWS アカウント and root user (AWS アカウントのセットアップとルートユーザー)

ワークロードを安全に運用する

ワークロードの安全な運用は、設計、ビルド、実行から継続的改善までワークロードのライフサイクル全体が対象です。クラウドで安全に運営する能力を改善する方法の 1 つは、ガバナンスに組織的アプローチを採用することです。ガバナンスとは、関与する担当者の適切な判断のみに依存することなく、意思決定が一貫して導かれるやり方です。ガバナンスモデルとプロセスは、「あるワークロードの管理目標が達成され、そのワークロードに適切であるということはどのように確認すればよいですか?」という質問への回答に当たります。 意思決定へのアプローチが一定していると、ワークロードのデプロイが加速され、組織内のセキュリティ機能の水準を向上させるのに役立ちます。

ワークロードを安全に運用するには、セキュリティのすべての領域に包括的なベストプラクティスを適用する必要があります。組織レベルおよびワークロードレベルにおいて、「運用上の優秀性」で定義した要件とプロセスを抽出し、それらをすべての領域に適用します。AWS や業界のレコメンデーションおよび脅威インテリジェンスを最新に保つことで、脅威モデルと管理目標を進化させることができます。セキュリティプロセス、テスト、検証を自動化することで、セキュリティ運用の規模を拡大することができます。

オートメーションは、プロセスの一貫性と再現性を可能にします。人は多くのことに長けていますが、同じことをミスなく一貫して繰り返し行うことは、得意なことではありません。きちんとしたランブックを作成しても、繰り返し行われる作業を一貫して行うことができないというリスクはあります。特に、担当業務が多岐にわたり、不慣れなアラートに対応しなければならない場合は、その傾向が顕著になります。しかし、オートメーションは毎回同じように反応します。アプリケーションをデプロイする最良の方法は、オートメーションです。デプロイを実行するコードをテストして、それを使ってデプロイを実施することができます。これにより、変更プロセスに対する信頼性が高まり、変更に失敗するリスクが軽減されます。

設定が管理目標を満たしていることを確認するために、まず非運用環境でオートメーションとデプロイされたアプリケーションをテストします。こうすることで、オートメーションをテストして、すべてのステップを正しく実行したことを証明できます。また、開発とデプロイサイクルの早い段階でフィードバックが得られるため、再作業を減らすことができます。デプロイエラーの可能性を減らすため、設定変更は人ではなくコードで行うようにしましょう。アプリケーションを再デプロイする必要がある場合、オートメーションを使用すると、これが非常に簡単になります。追加の管理目標を定義すると、すべてのワークロードのオートメーションに簡単に追加することができます。

個々のワークロード所有者がワークロードに固有のセキュリティに投資する代わりに、共通の機能と共有コンポーネントを使用することで時間を節約することができます。複数のチームが利用できるサービスの例としては、AWS アカウントの作成プロセス、人の ID の一元化、ログの共通設定、AMI やコンテナのベースイメージ作成などがあります。このアプローチにより、ビルダーはワークロードサイクル時間を改善して、セキュリティ管理目標を一貫して達成することができます。チームの一貫性が高まれば、管理目標を検証し、管理態勢とリスクポジションを利害関係者に適切に報告できるようになります。

ベストプラクティス

SEC01-BP03 管理目標を特定および検証する:

脅威モデルから特定されたコンプライアンス要件とリスクに基づいて、ワークロードに適用する必要がある管理目標および管理を導き出し、検証します。管理目標と制御を継続的に検証することは、リスク軽減の効果測定に役立ちます。

このベストプラクティスを活用しない場合のリスクレベル: 高

実装のガイダンス

- コンプライアンス要件を特定する: ワークロードが準拠する必要のある組織要件、法的要件、規制要件を確認します。
- AWS コンプライアンスリソースを特定する: コンプライアンスを支援するために使用できる AWS のリソースを特定します。
  - https://aws.amazon.com/compliance/
  - https://aws.amazon.com/artifact/

リソース

関連するドキュメント:

- AWS セキュリティ監査のガイドライン
- セキュリティ速報

関連動画:

- AWS Security Hub: Manage Security Alerts and Automate Compliance
- Well-Architected の手法によるセキュリティのベストプラクティス

SEC01-BP04 セキュリティ脅威に関する最新情報を入手する:

適切な制御を定義して実装するために、最新のセキュリティ脅威を常に把握して攻撃ベクトルを認識します。AWS Managed Services を利用することで、AWS アカウントにおける予期しない動作や異常な動作の通知を簡単に受けることができます。セキュリティ情報フローの一環として、AWS Partner ツールまたはサードパーティーの脅威情報フィードの利用を検討します。それらの 共通脆弱性識別子 (CVE) リスト には、一般に公開されているサイバーセキュリティの脆弱性が含まれており、最新の情報を入手するために利用することができます。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

- 脅威インテリジェンスソースを購読する: ワークロードで使用しているテクノロジーに関連する、複数のソースからの脅威インテリジェンス情報を定期的に確認します。
  - 共通脆弱性識別子リスト
- 検討 AWS Shield Advanced サービスを検討する: ワークロードがインターネットに接続できる環境であれば、インテリジェンスソースをほぼリアルタイムで可視化することができます。

リソース

関連するドキュメント:

- AWS セキュリティ監査のガイドライン
- AWS Shield

- セキュリティ速報

関連動画:

- Well-Architected の手法によるセキュリティのベストプラクティス

SEC01-BP05 セキュリティに関する推奨事項を常に把握する

AWS と業界の両方のセキュリティの推奨事項を常に最新に保ち、ワークロードのセキュリティ体制を進化させます。AWS セキュリティ速報 は、セキュリティおよびプライバシー通知に関する重要な情報を含みます。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

- AWS のアップデートをフォローする: 購読または定期的にチェックして、新しい推奨事項や ヒント、テクニックを確認しましょう。
  - AWS Well-Architected ラボ
  - AWS セキュリティブログ
  - AWS のサービスドキュメント
- 業界ニュースを購読する: 複数のソースから、ワークロードで使用しているテクノロジーに関連するニュースフィードを定期的に確認します。
  - 例: 共通脆弱性識別子リスト

リソース

関連するドキュメント:

- セキュリティ速報

関連動画:

- Well-Architected の手法によるセキュリティのベストプラクティス

SEC01-BP06 パイプラインのセキュリティコントロールのテストと検証を自動化する

ビルド、パイプライン、プロセスの一環としてテストおよび検証されるセキュリティメカニズムの安全なベースラインとテンプレートを確立します。ツールとオートメーションを使用して、すべてのセキュリティコントロールの継続的なテストと検証を実施します。たとえば、マシンイメージやインフラストラクチャなどの項目をコードテンプレートとしてスキャンして、セキュリティの脆弱性、不規則性、ドリフトを各ステージで確立されたベースラインから確認します。AWS CloudFormation Guard を使用すると、CloudFormation が安全なことを検証し、時間を節約し、設定エラーのリスクを低減するのに役立ちます。

本番環境に取り込まれるセキュリティの誤設定の数を減らすことが非常に重要です。ビルドプロセスでより適切な品質管理をより多く実行し、欠陥の数を減らすことができれば、より優れたものになります。

継続的インテグレーションおよび継続的デプロイ (CI/CD) のパイプラインは、可能な限りセキュリティの問題をテストできるように設計する必要があります。CI/CD パイプラインは、ビルドと配信の各段階でセキュリティを強化する機会を提供します。CI/CD セキュリティツールも更新して、進化する脅威を軽減する必要があります。

ワークロード設定への変更をトラッキングして、監査、変更管理、および該当する可能性がある調査へのコンプライアンスに役立てます。AWS Config を使用して、AWS およびサードパーティーリソースを記録および評価できます。これにより、ルールやコンフォーマンスパックへの全体的なコンプライアンスを継続的に監査および評価できます。コンフォーマンスパックとは、是正措置に関する一連のルールのことです。

変更トラッキングには、組織の変更管理プロセス (MACD-Move/Add/Change/Delete とも呼ばれる) の一部である計画されていた変更、予定外の変更、インシデントなどの予期しない変更を含める必要があります。変更はインフラストラクチャで発生する場合もあれば、コードリポジトリの変更、マシンイメージおよびアプリケーションインベントリの変更、プロセスとポリシーの変更、ドキュメントの変更などの他のカテゴリに関連するものである場合もあります。

このベストプラクティスを活用しない場合のリスクレベル: ミディアム

実装のガイダンス

- 設定管理を自動化する: 設定管理サービスまたはツールを使用して、リモートでアクションを実行し、安全な設定を自動的に適用および検証します。
  - AWS Systems Manager
  - AWS CloudFormation
  - AWS で CI/CD パイプラインを設定する

リソース

関連するドキュメント:

- How to use service control policies to set permission guardrails across accounts in your AWS Organization

関連動画:

- Managing Multi-Account AWS Environments Using AWS Organizations
- Well-Architected の手法によるセキュリティのベストプラクティス

SEC01-BP07 脅威モデルを使用して脅威を特定し、緩和策の優先順位を付ける

脅威のモデル化を実行し、ワークロードの潜在的脅威と関連付けられた緩和策を特定し、最新の状態を維持します。脅威に優先順位を付け、セキュリティコントロール緩和策を調整して防止、検出、対応を行います。ワークロードの内容、および進化するセキュリティ環境の状況に応じてセキュリティコントロールを保持および維持します。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

脅威のモデル化とは何ですか?

「脅威のモデル化は、価値のある対象を保護する文脈で、脅威と緩和策を特定、伝達、理解するためのもの」 – The Open Web Application Security Project (OWASP) Application Threat Modeling

脅威をモデル化すべきなのはなぜですか?

システムは複雑であり、時代とともに次第に複雑かつ高性能となり、提供するビジネス価値は向上し、顧客満足度とエンゲージメントは強化されています。つまり、IT 設計を決定する際は、増え続けるユースケースの件数を考慮する必要があるということです。このような複雑で数が多いユースケースの組み合わせは、非構造化アプローチでは一般に脅威の検出と緩和に効果がありません。代わりに必要となるのは、システムに対する潜在的な脅威を列挙し、緩和策を考案し、その緩和策に優先順位をつけて、組織の限定的なリソースがシステム全体のセキュリティ体制の改善に最大の効果を発揮できるような体系的アプローチです。

脅威のモデル化は、このような体系的アプローチを提供する設計となっており、その狙いは、ライフサイクルの後半と比較すると相対的にコストと労力が低い設計プロセスの早い段階で問題を発見し、対処することです。このアプローチは、「シフトレフト (前倒し)」セキュリティの業界原則と一致しています。最終的に脅威のモデル化は組織のリスク管理プロセスと統合し、脅威駆動型アプローチを使用して、実装するコントロールの決定を促します。

脅威のモデル化は、いつ実行すべきですか?

ワークロードのライフサイクルにおけるできるだけ早い段階で脅威のモデル化を開始することにより、より柔軟に特定した脅威への対策を実施できるようになります。ソフトウェアのバグと同様、脅威を特定するのが早いほど、その対策のコスト効率が向上します。脅威モデルはライブドキュメントであり、ワークロードの変化に応じて進化し続ける必要があります。大きな変化、脅威の状況における変化が生じた場合や、新たな機能またはサービスを採用した場合などを含む、経時的な脅威モデルを保持します。

実装手順

脅威のモデル化の実行方法を教えてください

脅威のモデル化にはさまざまな実行方法があります。プログラミング言語と同様、それぞれに長所と短所があり、自分に最も適した方法を選択する必要があります。1 つのアプローチは、Shostack's 4 Question Frame for Threat Modeling (脅威のモデル化のための Shostack の 4 つの質問フレーム) から始めるやり方です。これは、脅威のモデル化の演習に構造を与える自由形式の質問です。

1. うまくいっているものは何か?

   この質問の目的は、構築しているシステム、さらにはセキュリティに関連するシステムに関する詳細を理解してそれに合意するのを支援することです。構築している対象を視覚化できるため、モデルや図を作成するのが、この質問に対する回答として最も良くある方法です。たとえば、データフロー図などです。システムに関する推測と重要な詳細を書き留めることも、対象範囲を定義するのに役立ちます。これにより、脅威モデルに取り組む担当者全員の目指す方向が合致し、対象範囲外のトピック (システムの古いバージョンなど) に脱線して時間を浪費する事態を回避できます。たとえば、ウェブアプリケーションを構築している場合、ブラウザクライアントのオペレーティングシステムの信頼できるブートシーケンスをモデル化する脅威については、あまり時間をかける価値があるとは思えません。
2. どんな問題が起きる可能性があるでしょうか?

   ここで、システムに対する脅威を特定します。脅威とは、望ましくない影響を生じさせ、システムのセキュリティに悪影響を及ぼす恐れのある、偶発的または意図的なアクションや事象を指します。どのような問題が起きるかをはっきりと理解していなければ、何も対策は打てません。

   何が問題になるのかに関して、定型的なリストは存在しません。このリストを作成するには、チーム内の個人全員と脅威のモデル化に関与する関係担当者間のブレインストーミングとコラボレーションが必要となります。ブレインストーミングは、STRIDE などの脅威を特定するモデルを使用すると実施しやすくなります。これは、評価するためのさまざまなカテゴリ (スプーフィング、改ざん、否認、情報漏洩、サービス拒否、権限昇格) を提案するものです。さらに、既存のリストを見直し、OWASP トップ 10、HiTrust 脅威カタログ、そして組織独自の脅威カタログなどのインスピレーションを調査することもブレインストーミングに役立ちます。

3. それをどうするのですか?

   前の質問と同様、考えられる緩和策について定型的なリストはありません。このステップに対する入力項目は、特定された脅威、アクター、および前のステップからの改善点です。

   セキュリティとコンプライアンスは、AWS とお客様との間で共有される責任です。「それをどうするのですか?」という質問を行うときは、「誰がその責任者なのか?」ということも尋ねていると理解することが重要です。お客様と AWS 間の責任のバランスを理解することにより、お客様のコントロール下にある脅威のモデル化演習の範囲を理解するのに役立ちます。これは通常、AWS サービス設定オプションとお客様独自のシステムごとの緩和策を組み合わせたものです。

   共有責任の AWS 担当部分については、AWS サービスが多くのコンプライアンスプログラムの範囲内であることに気づくと思います。これらのプログラムは、セキュリティとクラウドのコンプライアンスを維持するためにAWS に配置された堅牢なコントロールを理解するのに役立ちます。これらのプログラムからの監査レポートは、AWS 顧客向けに AWS Artifact からダウンロードできます。

   どの AWS サービスを使用していても、必ずお客様の責任となる要素が存在し、これらの責任に合わせた緩和策を脅威モデルに組み込む必要があります。AWS サービス自体のセキュリティコントロール緩和のためには、たとえば、AWS Identity and Access Management (認証と承認)、データ保護 (静止時と転送時)、インフラストラクチャセキュリティ、ログ、モニタリングなどのドメインを含む、さまざまなドメイン全体にセキュリティコントロールの実装を検討することが推奨されます。各 AWS サービスのドキュメントには、専用のセキュリティに関する章が入っており、緩和策とみなされるセキュリティコントロールに関するガイダンスを提供します。重要ですので、記述しているコードとコード依存関係を考慮し、それらの脅威に対応するために設定できるコントロールについて考えてください。これらのコントロールは、入力の検証、セッションの取扱い、および範囲の取り扱いなどが考えられます。多くの場合、脆弱性の大部分はカスタムコードで発生するため、この領域を注視してください。
4. うまくいきましたか?

   狙いは、チームと組織が脅威モデルの質と、脅威のモデル化を行う際の時間的な速さを改善することです。これらの改善は、練習、学習、指導、レビューを組み合わせることで実現します。深く掘り下げて実践的な学習を行うため、お客様とチームが「Threat modeling the right way for builders training course (ビルダー向けの正しい脅威モデル化トレーニングコース)」またはワークショップを終了することが推奨されます。さらに、組織のアプリケーション開発ライフサイクルに脅威モデル化を統合する方法についてガイダンスを求めている場合、AWS セキュリティブログの「How to approach threat modeling (脅威のモデル化にアプローチする方法)」を参照してください。

Threat Composer

脅威のモデル化の実行に役立てるため、Threat Composer ツールの使用を検討してください。脅威のモデル化における価値実現までの時間の短縮を目的としたツールです。このツールは、以下の用途で役立ちます。

- 脅威の文法に沿って、自然な非線形のワークフローに当てはまる有益な脅威の文章を記述する
- 人間が読める脅威モデルを生成する
- 機械可読な脅威モデルを生成して、脅威モデルをコードとして扱えるようにする
- Insights Dashboard を使用して、品質や対象範囲を改善できる分野をすばやく特定する

詳細については、Threat Composer にアクセスして、システム定義の Example Workspace に切り替えてください。

リソース

関連するベストプラクティス:

- SEC01-BP03 管理目標を特定および検証する: (p. 12)
- SEC01-BP04 セキュリティ脅威に関する最新情報を入手する: (p. 13)

- SEC01-BP05 セキュリティに関する推奨事項を常に把握する (p. 14)
- SEC01-BP08 新しいセキュリティサービスと機能を定期的に評価および実装する (p. 18)

関連するドキュメント:

- 脅威モデリングのアプローチ方法 (AWS セキュリティブログ)
- NIST: Guide to Data-Centric System Threat Modeling

関連動画:

- AWS Summit ANZ 2021 - How to approach threat modelling
- AWS Summit ANZ 2022 - Scaling security – Optimise for fast and secure delivery

関連トレーニング:

- Threat modeling the right way for builders – AWS Skill Builder virtual self-paced training
- Threat modeling the right way for builders - AWS ワークショップ

関連ツール:

- Threat Composer

SEC01-BP08 新しいセキュリティサービスと機能を定期的に評価および実装する

ワークロードのセキュリティ体制を進化させることができる、AWS および AWS パートナーのセキュリティサービスと機能を評価および実装します。AWS セキュリティブログは、新しい AWS サービスおよび機能、実装ガイド、および一般的なセキュリティガイダンスを取り上げます。「AWS の最新情報」は、すべての AWS 機能、サービス、および発表に関する最新情報を確認する優れた方法です。

このベストプラクティスを活用しない場合のリスクレベル: 低

実装のガイダンス

- 定期的なレビューを計画する: コンプライアンス要件、AWS の新しいセキュリティ機能とセキュリティサービスの評価、業界の最新ニュースの入手を含むレビューアクティビティのカレンダーを作成します。
- AWS のサービスと機能について調べる: 使用中のサービスで利用可能なセキュリティ機能について調べ、新しい機能がリリースされた時には、それについて確認します。
  - AWS セキュリティブログ
  - AWS セキュリティ速報
  - AWS のサービスドキュメント
- AWS のサービスの導入プロセスを定義する: 新しい AWS サービスの導入プロセスを定義します。新しい AWS のサービスの機能とワークロードのコンプライアンス要件を評価する方法を含めます。
- 新しいサービスと機能をテストする: 新しいサービスと機能がリリースされたら、本稼働環境に近いかたちで複製する本稼働環境ではない環境でテストします。
- その他の防御メカニズムを実装する: ワークロードを保護するための自動化されたメカニズムを実装し、利用可能なオプションを確認します。
  - AWS Config ルール による非準拠 AWS リソースの修復

リソース

関連動画:

- Well-Architected の手法によるセキュリティのベストプラクティス

ID とアクセス管理

AWS のサービスを使用するには、ユーザーとアプリケーションに AWS アカウントのリソースへのアクセス権限を与える必要があります。AWS で実行するワークロードの増加に伴い、適切なユーザーが適切な条件で適切なリソースにアクセスできるようにするためには、強固な ID 管理とアクセス許可が必要です。AWS は、幅広い機能の選択肢を提供することによって、ユーザーとマシンの ID および権限の管理を支援しています。これらの機能のベストプラクティスは、次の 2 つの領域に大きく分類されます。

トピック

ID 管理

AWS ワークロードを安全に運用するには、2 種類の ID を管理する必要があります。

- ユーザー ID: 管理者、開発者、オペレーター、アプリケーションのエンドユーザーは、AWS 環境とアプリケーションにアクセスできる ID が必要です。これらのユーザーは、あなたの組織のメンバー、または共同作業を行う外部ユーザーで、ウェブブラウザ、クライアントアプリケーション、モバイルアプリ、インタラクティブなコマンドラインツールを介して AWS リソースを操作します。
- マシン ID: ワークロードアプリケーション、運用ツール、コンポーネントには、データ読み取りなどのため、AWS のサービスにリクエストを送信できる ID が必要です。このような ID には、Amazon EC2 インスタンスや AWS Lambda 関数など、AWS 環境で実行されているマシンが含まれます。また、アクセスを必要とする外部関係者のマシン ID を管理することもできます。さらに、AWS 環境にアクセスする必要があるマシンが AWS 外にある可能性もあります。

ベストプラクティス

SEC02-BP01 強力なサインインメカニズムを使用する

サインイン (サインイン認証情報を使った認証) は、多要素認証 (MFA) などのメカニズムを使わない場合、特にサインイン認証情報が不用意に開示されたり、容易に推測されたりする場合に、リスクが発生する恐れがあります。MFA や強力なパスワードポリシーを要求することで、これらのリスクを軽減する強力なサインインのメカニズムを使用します。

期待される成果: AWS Identity and Access Management (IAM) ユーザー、AWS アカウント ルートユーザー、AWS IAM Identity Center (AWS シングルサインオンの後継サービス)、およびサードパーティー ID プロバイダー向けに強力なサインインメカニズムを使用することにより、AWS の認証情報に対する意図しないアクセスのリスクを軽減します。これは、MFA が必須となり、強力なパスワードポリシーが適用され、異常なログイン動作が検出されることを意味します。

一般的なアンチパターン:

- 複雑なパスワードや MFA など、自分のアイデンティティに対して強力なパスワードポリシーを適用しない。
- 複数のユーザー間で同一の認証情報を共有する。
- 疑わしいサインインに対して検出コントロールを使用しない。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

人的 ID が AWS にサインインする方法は多数あります。AWS ベストプラクティスは、AWS に認証する際にフェデレーション (直接フェデレーションまたは AWS IAM Identity Center を使用) を使って、一元化された ID プロバイダーに依存する方法です。この場合、ID プロバイダーまたは Microsoft Active Directory を使って、セキュアなサインインプロセスを確立する必要があります。

最初に AWS アカウント を開いたとき、AWS アカウント ルートユーザーから始めます。ユーザー (およびルートユーザーを必要とする タスク) へのアクセスを設定するには、アカウントのルートユーザーのみを使用する必要があります。.AWS アカウント を開いた直後にアカウントのルートユーザーに対して MFA を有効化し、AWS ベストプラクティスガイドを使用してルートユーザーをセキュリティ保護することが重要です。

AWS IAM Identity Center でユーザーを作成する場合、そのサービスでサインインプロセスをセキュリティ保護します。消費者アイデンティティについては、Amazon Cognito user pools を使用して、そのサービスで、またはAmazon Cognito user pools がサポートする ID プロバイダーの 1 つを使ってサインインプロセスをセキュリティ保護します。

AWS Identity and Access Management (IAM) ユーザーを使用している場合、IAM を使ってサインインプロセスをセキュリティ保護することになります。

サインイン方法に関係なく、強力なサインインポリシーを適用することが不可欠です。

実装手順

一般的な強力なサインインに関する推奨事項は次の通りです。実際に行う設定は、貴社のポリシーによって設定するか、または NIST 800-63 のような標準を使います。

- MFA が必要です。人的 ID とワークロードに対しては、MFA を義務付けることがIAM のベストプラクティスです。MFAを有効にすることで、追加のセキュリティ層が提供されます。この層では、ユーザーがサインイン認証情報、ワンタイムパスワード (OTP)、またはハードウェアデバイスから暗号的に検証および生成された文字列を提供することが求められます。
- 最小パスワード文字数を適用します。これは、パスワードの強さにおける主な要素です。
- パスワードの複雑性を適用すると、パスワードを推測しにくくなります。
- ユーザー自身によるパスワードの変更を許可します。
- 共有認証情報ではなく、個別の ID を作成します。個別の ID を作成することで、各ユーザーに固有のセキュリティ認証情報を付与することができます。個別のユーザーを作成することで、各ユーザーのアクティビティを監査する機能が利用できます。

IAM Identity Center レコメンデーション

- IAM Identity Center は、デフォルトディレクトリを使用する際、パスワードの文字数、複雑性、および再使用要件を確立する、事前定義された パスワードポリシーを提供します。
- MFA を有効にし、アイデンティティソースがデフォルトディレクトリ、AWS Managed Microsoft AD、または AD Connector の場合、MFA に対してコンテキストアウェアまたは常時オン設定を行います。
- ユーザーが、自分の MFA デバイスを登録できるようにします。

Amazon Cognito user pools ディレクトリのレコメンデーション:

- パスワードの強さ設定を行います。
- ユーザーに対してMFA を義務付けます。
- 疑わしいサインインをブロックできる適応型認証などの機能に対して、Amazon Cognito user pools上級セキュリティ設定を使用します。

IAM ユーザーのレコメンデーション:

- IAM Identity Center または直接フェデレーションを使用することが理想的です。しかし、IAM ユーザー向けのニーズもあるでしょう。その場合は、IAM ユーザー向けにパスワードポリシーを設定します。パスワードポリシーを使用して、最小文字数、またはアルファベット以外の文字が必要かどうかなどの要件を定義できます。
- IAM ポリシーを作成して、MFA サインインを適用し、ユーザーが自分のパスワードと MFA デバイスを管理できるようにします。

リソース

関連するベストプラクティス:

- SEC02-BP03 シークレットを安全に保存して使用する (p. 24)
- SEC02-BP04 一元化された ID プロバイダーを利用する (p. 28)
- SEC03-BP08 組織内でリソースを安全に共有する (p. 49)

関連するドキュメント:

- AWS IAM Identity Center (AWS シングルサインオンの後継サービス) パスワードポリシー
- IAM ユーザーのパスワードポリシー
- AWS アカウント のルートユーザーのパスワードの設定
- Amazon Cognito パスワードポリシー
- AWS 認証情報
- IAM セキュリティのベストプラクティス

関連動画:

- Managing user permissions at scale with AWS IAM Identity Center (AWS SSO を使用した大規模なユーザー権限の管理)
- Mastering identity at every layer of the cake (すべての層での ID の把握)

SEC02-BP02 一時的な認証情報を使用する

何らかの認証を行う際、認証情報が誤って開示、共有、盗難されたりなどのリスクを軽減または排除するには、長期的認証情報ではなく一時的な認証情報を使うことが推奨されます。

期待される成果: 長期的認証情報のリスクを軽減するには、人的および機械両方の ID にできるだけ一時的な認証情報を使用するようにします。長期的認証情報を使用すると、多くのリスクが生じます。たとえば、パブリックな GitHub リポジトリにコードでアップロードすることができます。一時的な認証情報を使うことにより、認証情報が侵害されるリスクが大幅に減少します。

一般的なアンチパターン:

- 開発者が、フェデレーションを使って CLI から一時的な認証情報を取得するのではなく、IAM users からの長期的なアクセスキーを使用する。
- 開発者がコードに長期的アクセスキーを埋め込んで、そのコードをパブリック Git リポジトリにアップロードする。
- 開発者が、モバイルアプリに長期的アクセスキーを埋め込んで、アプリストアで公開する。
- ユーザーが長期的アクセスキーを他のユーザー、または従業員と共有し、長期的アクセスキーを所有したまま離職する。
- 一時的認証情報を使用できるのに、マシン ID に対して長期的なアクセスキーを使用する。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

すべての AWS API と CLI リクエストに対して、長期的認証情報ではなく一時的なセキュリティ認証情報を使用します。AWS サービスに対する API および CLI リクエストは、ほとんどの場合、AWS アクセスキーを使って署名する必要があります。これらのリクエストの署名に使用する認証情報は、一時的でも長期的でもかまいません。長期的認証情報 (長期的アクセスキー) を使用すべき唯一の状況は、IAM ユーザーまたは AWS アカウント ルートユーザーを使用している場合です。AWS に対してフェデレーションを行うか、または他の方法により IAM ロールを担う場合、一時的認証情報が生成されます。サインイン認証情報を使って AWS Management Console にアクセスしても、AWS サービスへのコールを行うために一時的な認証情報が生成されます。長期的認証情報が必要な状況はほとんどなく、一時的な認証情報でほとんどのタスクを遂行できます。

一時的な認証情報を優先して長期的な認証情報の使用を回避することは、フェデレーションと IAM ロールを優先して IAM ユーザーの使用を減少させる戦略と一致していなければなりません。IAM ユーザーは過去に人的とマシン ID 両方に対して使用されましたが、長期的アクセスキー使用におけるリスクを回避するため、それを使用しないよう推奨しています。

実装手順

従業員、管理者、開発者、オペレーター、および顧客などの人的 ID の場合:

- 一元化された ID プロバイダーに依存して、人間ユーザーが一時的な認証情報を使って AWS にアクセスするには、ID プロバイダーにフェデレーションを使用することを義務付ける必要があります。ユーザーに対するフェデレーションは、各 AWS アカウント の直接フェデレーションで、または AWS IAM Identity Center (AWS IAM Identity Center の後継サービス) および好みの ID プロバイダーを使って行うことができます。フェデレーションは、長期的な認証情報を排除するだけでなく、IAM ユーザーを使用する場合と比較して多数の利点があります。ユーザーは 直接フェデレーション用のコマンド行から、または IAM Identity Center を使用して、一時的な認証情報をリクエストすることができます。つまり、IAM ユーザーまたは、ユーザー向けの長期的認証情報を必要なケースはほとんどないということです。
- Software as a Service (SaaS) などのサードパーティーに、AWS アカウント のリソースへのアクセスを付与する際、クロスアカウントロールおよびリソースベースポリシーを使用できます。
- 消費者や顧客向けのアプリケーションに AWS リソースへのアクセスを許可する必要がある場合、Amazon Cognito アイデンティティ プールまたはAmazon Cognito user pools を使用して、一時的な認証情報を提供できます。認証情報のアクセス許可は、IAM ロールによって設定されます。 認証されていないゲストユーザーには、制限付きのアクセス権限を持つ IAM ロールを個別に定義できます。

マシン ID の場合、長期的認証情報を使用しなければならない場合があります。これらの場合、IAM ロールで AWS にアクセスする際に、ワークロードが一時的な認証情報を使用するよう義務付ける必要があります。

- Amazon Elastic Compute Cloud (Amazon EC2) の場合、Amazon EC2 に対して ロールを使用できます。

- AWS Lambda では、一時的な認証情報を使って AWS アクションを実行するためのサービス権限を付与する Lambda 実行ロールを設定できます。AWS サービスが、IAM ロールを使って一時的な認証情報を付与する類似モデルは多数あります。
- IoT デバイスの場合、AWS IoT Core 認証情報プロバイダーを使って、一時的な認証情報をリクエストできます。
- オンプレミスのシステム、または AWS 外で実行され、AWS リソースへアクセスする必要があるシステムの場合、IAM Roles Anywhere を使用できます。

一時的な認証情報が選択肢として使えず、長期的認証情報を使う必要があるシナリオがあります。これらの状況では、が定期的に認証情報を監査してローテーションし、さらに 長期的認証情報が必要なユースケースに対して定期的にアクセスキーをローテーションします。長期的認証情報が必要となるかもしれない例には、WordPress プラグインやサードパーティーの AWS クライアントなどが考えられます。長期的認証情報を使用すべき状況、またはデータベースログインなどの AWS アクセスキー以外の認証情報については、AWS Secrets Manager など、シークレット管理を処理するために設計されたサービスを使用できます。Secrets Manager は、サポートされているサービスを使用して、暗号化されたシークレットを簡単に管理、ローテーション、安全に保存できます。長期的認証情報のローテーションについては、「アクセスキーのローテーション」を参照してください。

リソース

関連するベストプラクティス:

- SEC02-BP03 シークレットを安全に保存して使用する (p. 24)
- SEC02-BP04 一元化された ID プロバイダーを利用する (p. 28)
- SEC03-BP08 組織内でリソースを安全に共有する (p. 49)

関連するドキュメント:

- 一時的なセキュリティ認証情報
- AWS 認証情報
- IAM セキュリティのベストプラクティス
- IAM ロール
- IAM Identity Center
- ID プロバイダーとフェデレーション
- アクセスキーのローテーション
- Security Partner Solutions: Access and Access Control (セキュリティパートナーソリューション: アクセスおよびアクセスコントロール)
- AWS アカウントのルートユーザー

関連動画:

- Managing user permissions at scale with AWS IAM Identity Center (AWS SSO を使用した大規模なユーザー権限の管理) (AWS IAM Identity Center の後継サービス)
- Mastering identity at every layer of the cake (すべての層での ID の把握)

SEC02-BP03 シークレットを安全に保存して使用する

ワークロードには、データベース、リソース、およびサードパーティーサービスにアイデンティティを証明するための自動機能が必要となります。これは、API アクセスキー、パスワード、および OAuth トークンなどの、シークレットアクセス認証情報を使って実現されます。これらの認証情報を保存、管理、ロー

テーションする専用のサービスを使用することで、認証情報が侵害される可能性を低減することができます。

期待される成果: 次の目標を達成するアプリケーションの認証情報を安全に管理するメカニズムを実装する:

- ワークロードに必要なシークレットを特定する。
- 長期的認証情報を短期的認証情報と置き換える (可能な場合) ことによりその数を減らす。
- 安全なストレージと、残りの長期的認証情報の自動化されたローテーションを確立する。
- ワークロードに存在するシークレットへのアクセスを監査する。
- 開発プロセス中、ソースコードに組み込まれたシークレットがないことを継続的に監視する。
- 認証情報が誤って開示される可能性を減らす。

一般的なアンチパターン:

- 認証情報をローテーションしない。
- ソースコードまたは設定ファイルに長期的認証情報を保管する。
- 認証情報を暗号化せずに保管する。

このベストプラクティスを活用するメリット:

- シークレットが、保管時と転送時に暗号化される。
- 認証情報へのアクセスが、API (認証情報の自動販売機と考える) 経由でゲート化される。
- 認証情報へのアクセス (読み出しと書き込み) が監査およびログ記録される。
- 懸念事項の分離: 認証情報のローテーションは、アーキテクチャの他の部分から分離できる別のコンポーネントによって実行されます。
- シークレットは、ソフトウェアコンポーネントに対してオンデマンドで配布され、中央ロケーションでローテーションが発生する。
- 認証情報へのアクセスは、非常にきめ細やかに制御できます。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

従来、データベースやサードパーティーの API、トークンなどの認証に使用する認証情報は、ソースコードや環境ファイルに埋め込まれている場合がありました。AWS は、これらの認証情報を安全に保管し、自動的にローテーションし、その使用を監査するメカニズムを複数提供しています。

シークレット管理に対する最善のアプローチは、削除、置換、ローテーションのガイダンスに従うことです。最も安全な認証情報は、保管、管理、処理が不要なものです。認証情報によっては、ワークロードの機能にとって不要となった、安全に削除できるものもあります。

ワークロードの正常な機能に依然として必要な認証情報については、長期的認証情報を一時的または短期的な認証情報と置換する機会があるかもしれません。たとえば、AWS シークレットアクセスキーをハードコーディングする代わりに、IAM ロールを使って長期的認証情報を一時的認証情報と置換することを検討してみてください。

存続期間の長いシークレットによっては、削除も置換もできないものがあります。これらのシークレットは、AWS Secrets Manager などのサービスに保管して、一元的に保管、管理したり、定期的にローテーションしたりすることができます。

ワークロードのソースコードと設定ファイルの監査を行うと、さまざまなタイプの認証情報が明らかになる可能性があります。次の表は、一般的なタイプの認証情報を取り扱うための戦略をまとめたものです。

Credential type	Description	Suggested strategy
IAM access keys	AWS IAM access and secret keys used to assume IAM roles inside of a workload	Replace: Use IAM ロール assigned to the compute instances (such as Amazon EC2 or AWS Lambda) instead. For interoperability with third parties that require access to resources in your AWS アカウント, ask if they support AWS クロスアカウントアクセス. For mobile apps, consider using temporary credentials through Amazon Cognito ID プール (フェデレーティッドアイデンティティ). For workloads running outside of AWS, consider IAM Roles Anywhere or AWS Systems Manager ハイブリッドアクティベーション.
SSH keys	Secure Shell private keys used to log into Linux EC2 instances, manually or as part of an automated process	Replace: Use AWS Systems Manager or EC2 Instance Connect to provide programmatic and human access to EC2 instances using IAM roles.
Application and database credentials	Passwords – plain text string	Rotate: Store credentials in AWS Secrets Manager and establish automated rotation if possible.
Amazon RDS and Aurora Admin Database credentials	Passwords – plain text string	Replace: Use the Amazon RDS との Secrets Manager 統合 or Amazon Aurora. In addition, some RDS database types can use IAM roles instead of passwords for some use cases (for more detail, see IAM データベース認証).
OAuth tokens	Secret tokens – plain text string	Rotate: Store tokens in AWS Secrets Manager and configure automated rotation.
API tokens and keys	Secret tokens – plain text string	Rotate: Store in AWS Secrets Manager and establish automated rotation if possible.

一般的なアンチパターンは、ソースコード、設定ファイル、またはモバイルアプリ内に IAM アクセスキーを埋め込むことです。IAM アクセスキーが AWS サービスと通信する必要がある場合、一時的 (短期的) セキュリティ認証情報を使用します。これらの短期的な認証情報は、EC2 インスタンス用の IAM ロール、Lambda 関数の実行ロール、モバイルユーザーアクセスのための Cognito IAM ロール、および IoT デバイス用の IoT Core ポリシーを通して提供できます。サードパーティー向けの場合は、IAM ユーザーをサーバーして、サードパーティーにそのユーザー向けのシークレットアクセスキーを送信するよりも、アカウントのリソースへの必要なアクセス権を持つ IAM ロールにアクセスを委譲する方法を優先します。

ワークロードに、他のサービスやリソースとの相互運用に必要なシークレットの保管が必要となるケースが多数あります。AWS Secrets Manager は、これらの認証情報の安全な管理、さらには API トークン、パスワード、およびその他の認証情報の保管、使用、ローテーション専用です。

AWS Secrets Manager は、機密性の高い認証情報を確実かつ安全に保管して取扱うための主な機能を 5 つ提供しています: 保管時の暗号化、転送中の暗号化、総合的な監査、きめ細やかなアクセスコントロール、および拡張可能な認証情報のローテーション。AWS パートナーによるその他のシークレット管理サービス、または類似の機能や保証を提供するローカルで開発されたソリューションも使用できます。

実装手順

1. Amazon CodeGuru などの自動化ツールを使用して、ハードコード化された認証情報を含むコードパスを特定します。
   - Amazon CodeGuru を使って、コードリポジトリをスキャンします。レビューが完了したら、CodeGuru で `Type=Secrets` をフィルターして、問題のあるコードの行を突き留めます。
2. 削除または置換できる認証情報を特定します。
   a. すでに不要な認証情報を特定して、削除用にマークします。
   b. ソースコードに埋め込まれた AWS シークレットキーについては、必要なリソースに関連付けられた IAM ロールと置換します。ワークロードの一部が AWS 外であるにもかかわらず AWS リソースにアクセスする IAM 認証情報が必要な場合、IAM Roles Anywhere または AWS Systems Manager ハイブリッドアクティベーションを検討してください。
3. ローテーション戦略を使用すべきその他のサードパーティー、存続期間の長いシークレットについては、Secrets Manager をコードに統合して、ランタイムにサードパーティーのシークレットを取得します。
   a. CodeGuru コンソールは、検出された認証情報を使って Secrets Manager を作成できます。
   b. Secrets Manager から取得したシークレットをアプリケーションコードに統合します。
      - サーバーレス Lambda 関数では、言語に依存しない Lambda 拡張子を使用できます。
      - EC2 インスタンスまたはコンテナに対しては、AWS が複数のよく使用されるプログラミング言語で、Secrets Manager からシークレットを取得するためのクライアント側コードの例を提供しています。
4. 定期的にコードベースをレビューして再スキャンすることで、コードに新たなシークレットが追加されていないことを確認します。
   - git-secrets などのツールを使って、ソースコードリポジトリに新しいシークレットがコミットされるのを防止することを検討してください。
5. 予想外の使用、不適切なシークレットへのアクセス、またはシークレットの削除試行がないかどうか、Secrets Manager アクティビティをモニタリングします。
6. 認証情報に対する人的曝露を減少させます。この目的に特化した IAM ロールに対する認証情報を読み出し、書き込み、および変更するためのアクセスを制限し、一部の運用ユーザーにのみ、その役割を担うためのアクセスを提供します。

リソース

関連するベストプラクティス:

- SEC02-BP02 一時的な認証情報を使用する (p. 22)
- SEC02-BP05 定期的に認証情報を監査およびローテーションする (p. 31)

関連するドキュメント:

- Getting started with AWS Secrets Manager (AWS シークレットマネージャーの開始方法)
- ID プロバイダーとフェデレーション

- Amazon CodeGuru Introduces Secrets Detector (Amazon CodeGuru がシークレットディテクターを提供)
- How AWS Secrets Manager uses AWS Key Management Service (AWS Secrets Manager が AWS Key Management Service を使用する方法について)
- Secret encryption and decryption in Secrets Manager (Secrets Manager におけるシークレット暗号化と復号化)
- Secrets Manager ブログエントリ
- Amazon RDSと AWS Secrets Manager の統合を発表

関連動画:

- Best Practices for Managing, Retrieving, and Rotating Secrets at Scale (シークレットを大規模に管理、取得、変更するためのベストプラクティス)
- Find Hard-Coded Secrets Using Amazon CodeGuru Secrets Detector (Amazon CodeGuru Reviewer Secrets Detector を使ってハードコード化されたシークレットを見つける)
- Securing Secrets for Hybrid Workloads Using AWS Secrets Manager (AWS re:Inforce 2022 - AWS Secrets Manager を使用したハイブリッドワークロードのシークレットの保護)

関連ワークショップ:

- Store, retrieve, and manage sensitive credentials in AWS Secrets Manager (AWS Secrets Manager で機密性の高い認証情報を保存、取得、管理する)
- AWS Systems Manager ハイブリッドアクティベーション

SEC02-BP04 一元化された ID プロバイダーを利用する

ワークフォースユーザー ID (従業員と契約社員) の場合、ID を一元管理できる ID プロバイダーを利用します。一つの場所から権限の作成、割り当て、管理、取り消し、監査を行うため、複数のアプリケーションおよびシステムにまたがる権限を効率的に管理できます。

期待される成果: 一元化された ID プロバイダーを使用して、ワークフォースユーザー、認証ポリシー (多要素認証 (MFA) の要求など)、システムやアプリケーションへの承認 (ユーザーのグループメンバーシップや属性に基づくアクセスの割り当てなど) を一元管理します。ワークフォースユーザーは一元化された ID プロバイダーにサインインし、内部アプリケーションと外部アプリケーションにフェデレーション (シングルサインオン) します。これにより、ユーザーは複数の認証情報を覚えておく必要がなくなります。ID プロバイダーは人事 (HR) システムと統合されているため、人事上の変更は ID プロバイダーと自動的に同期されます。例えば、誰かが組織を離れた場合、フェデレーションされたアプリケーションやシステム (AWS を含む) へのアクセスを自動的に取り消すことができます。ID プロバイダーで詳細な監査ログを有効にし、これらのログでユーザーの異常な行動がないか監視します。

一般的なアンチパターン:

- フェデレーションとシングルサインオンを使用しない。ワークフォースユーザーが、複数のアプリケーションやシステムで個別のユーザーアカウントと認証情報を作成する。
- ID プロバイダーを人事システムに統合するなど、ワークフォースユーザーのアイデンティティのライフサイクルを自動化していない。ユーザーが組織を離れたり、役割を変更したりした場合に、複数のアプリケーションやシステムのレコードを手動のプロセスで削除または更新する。

このベストプラクティスを活用するメリット: 一元化された ID プロバイダーを使用することで、ワークフォースユーザーのアイデンティティとポリシーを 1 か所で管理でき、ユーザーやグループにアプリケーションへのアクセス権を割り当てたり、ユーザーのサインインアクティビティを監視したりできます。人

事 (HR) システムと統合することで、ユーザーの役割が変更された場合は、これらの変更が ID プロバイダーと同期され、ユーザーに割り当てられたアプリケーションと権限が自動的に更新されます。ユーザーが組織を離れると、そのユーザーのアイデンティティは ID プロバイダーで自動的に無効になり、フェデレーションアプリケーションおよびシステムへのアクセス権が取り消されます。

このベストプラクティスを活用しない場合のリスクレベル: 高

実装のガイダンス

AWS にアクセスするワークフォースユーザー向けのガイダンス

組織内の従業員や契約社員などのワークフォースユーザーは、AWS Management Console または AWS Command Line Interface (AWS CLI) を使って職務を遂行するため、AWS へのアクセス権を必要とする場合があります。一元化された ID プロバイダーから 2 つのレベルで AWS にフェデレーションすることで、ワークフォースユーザーに AWS へのアクセス権を付与できます。1 つは各 AWS アカウント への直接フェデレーション、もう 1 つは AWS 組織内の複数のアカウントへのフェデレーションです。

- ワークフォースユーザーをそれぞれの AWS アカウント と直接フェデレーションするには、一元化された ID プロバイダーを使用して、そのアカウントの AWS Identity and Access Management にフェデレーションできます。IAM の柔軟性により、 SAML 2.0 または Open ID Connect (OIDC) という別々の ID プロバイダーを各 AWS アカウント で有効にして、アクセスコントロールにはフェデレーションユーザー属性を使用することができます。ワークフォースユーザーはウェブブラウザを使用し、認証情報 (パスワードや MFA トークンコードなど) を入力して ID プロバイダーにサインインします。ID プロバイダーは、AWS Management Console のサインイン URL に送信される SAML アサーションをユーザーのブラウザに発行して、 IAM ロールを引き受けることで、ユーザが AWS Management Console にシングルサインオンできるようにします。ユーザーは、ID プロバイダーからの SAML アサーションを使用して、AWS ロールを引き受けることで、 AWS CLI の AWS や AWS STS SDK で使用する 一時的な IAM API 認証情報を 取得することもできます。
- ワークフォースユーザーを AWS 組織内の複数のアカウントにフェデレーションするには、 AWS IAM Identity Center を使用して、AWS アカウント やアプリケーションへのワークフォースユーザーのアクセスを一元管理できます。組織のアイデンティティセンターを有効にし、ID ソースを設定します。IAM Identity Center は、ユーザーやグループの管理に使用できるデフォルトの ID ソースディレクトリを提供します。または、 SAML 2.0 を使用して 外部 ID プロバイダーに接続し、 SCIM を使用してユーザーとグループを 自動的にプロビジョニングするか、または AWS Directory Service を使用して Microsoft AD Directory に接続することで、 外部 ID ソースを選択することもできます。ID ソースを設定したら、アクセス許可セットで最小権限ポリシーを定義して、ユーザーとグループに AWS アカウント へのアクセス権を 割り当てることができます。ワークフォースユーザーは一元化された ID プロバイダーを通じて認証を行い、 AWS アクセスポータル にサインインして、自分に割り当てられた AWS アカウント とクラウドアプリケーションにシングルサインオンします。ユーザは AWS CLI v2 を設定して、アイデンティティセンターで認証を行い、AWS CLI コマンドを実行するための認証情報を取得できます。アイデンティティセンターでは、AWS アプリケーション (Amazon SageMaker Studio や AWS IoT Sitewise Monitor ポータル) へのアクセスにシングルサインオンも使用できます。 。

前述のガイダンスに従うと、ワークフォースユーザーは AWS でワークロードを管理する際、通常の操作で IAM users およびグループを使用する必要がなくなります。代わりに、ユーザーとグループは AWS 外部で管理され、ユーザーはフェデレーション ID として AWS リソースにアクセスできます。フェデレーション ID では、一元化された ID プロバイダーで定義されたグループを使用します。AWS アカウント で不要になった IAM グループ、IAM users、および永続的なユーザー認証情報 (パスワードとアクセスキー) を特定して削除する必要があります。ま た 、 IAM 認証情報レポートを使用して、未使用の認証情報を検索して、 該当する IAM users や IAM グループを削除できます。 組織に サービスコントロールポリシー (SCP) を適用して、新しい IAM users やグループが作成されないようにし、フェデレーション ID を介した AWS へのアクセスを強制できます。

アプリケーションのユーザー向けガイダンス

モバイルアプリなどのアプリケーションのユーザーの ID を管理するには、一元化された ID プロバイダーとして Amazon Cognito を使用できます。Amazon Cognito は、ウェブアプリやモバイルアプリの認証、

承認、ユーザー管理を可能にします。Amazon Cognito は数百万人のユーザーにスケール可能な ID ストアを備え、ソーシャル ID フェデレーションとエンタープライズ ID フェデレーションをサポートし、ユーザーとビジネスの保護に役立つ高度なセキュリティ機能を提供します。カスタムのウェブまたはモバイルアプリケーションを Amazon Cognito と統合すると、アプリケーションへのユーザー認証とアクセスコントロールを数分で追加できます。SAML や Open ID Connect (OIDC) などのオープン ID 標準に基づいて構築された Amazon Cognito は、さまざまなコンプライアンス規制に対応し、フロントエンドおよびバックエンドの開発リソースと統合します。

実装手順

ワークフォースユーザーの AWS へのアクセス手順

- 以下のいずれかの方法を使用し、一元化された ID プロバイダーを使用して、ワークフォースユーザーを AWS にフェデレーションします。
  - IAM Identity Center を使用し、ID プロバイダーとフェデレーションすることで、AWS 組織内の複数の AWS アカウント へのシングルサインオンを有効にします。
  - IAM を使用して、ID プロバイダーを各 AWS アカウント に直接接続し、フェデレーションによるきめ細かいアクセスを可能にします。
- フェデレーション ID で置き換えられた IAM users とグループを特定して削除します。

アプリケーションのユーザー向けの手順

- アプリケーション用の一元化された ID プロバイダーとして Amazon Cognito を使用します。
- OpenID Connect と OAuth を使用して、カスタムアプリケーションを Amazon Cognito と統合します。認証のための Amazon Cognito など、さまざまな AWS サービスと統合するためのシンプルなインターフェイスを提供する Amplify ライブラリを使用して、カスタムアプリケーションを開発できます。

リソース

関連する Well-Architected のベストプラクティス:

- SEC02-BP06 ユーザーグループと属性を活用する (p. 33)
- SEC03-BP02 最小特権のアクセスを付与します (p. 36)
- SEC03-BP06 ライフサイクルに基づいてアクセスを管理する (p. 46)

関連するドキュメント:

- Identity federation in AWS
- IAM でのセキュリティのベストプラクティス
- AWS Identity and Access Management Best practices
- Getting started with IAM Identity Center delegated administration
- How to use customer managed policies in IAM Identity Center for advanced use cases
- AWS CLI v2: IAM Identity Center credential provider

関連動画:

- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) deep dive
- AWS re:Invent 2022 - Simplify your existing workforce access with IAM Identity Center
- AWS re:Invent 2018: Mastering Identity at Every Layer of the Cake

関連する例:

- Workshop: Using AWS IAM Identity Center to achieve strong identity management
- Workshop: Serverless identity

関連ツール:

- AWS セキュリティコンピテンシーパートナー: ID およびアクセスの管理
- saml2aws

SEC02-BP05 定期的に認証情報を監査およびローテーションする

認証情報を定期的に監査およびローテーションして、リソースへのアクセスに認証情報を使用できる期間を制限します。長期的認証情報を使用すると多くのリスクが生じ、これらのリスクは長期的認証情報を定期的にローテーションすることにより軽減できます。

期待される成果: 認証情報のローテーションを実装することにより、長期的認証情報の使用に関連するリスクを軽減します。認証情報ローテーションポリシーの不遵守を定期的に監査して、是正します。

一般的なアンチパターン:

- 認証情報の使用を監査しない。
- 必要がないのに、長期的認証情報を使う。
- 長期的認証情報を使用して、定期的にローテーションしない。

このベストプラクティスを確立しない場合のリスクレベル: 中

実装のガイダンス

一時的な認証情報に頼れず、長期的な認証情報が必要な場合は、認証情報を監査して、多要素認証 (MFA) などの定義された管理方法が実施され、定期的にローテーションされ、アクセスレベルが適切であることを確認する必要があります。

正しい制御が実施されていることを確認するには、定期的な検証、できれば自動化されたツールによる検証が必要です。ユーザー ID の場合、ユーザーにはパスワードの定期的な変更と、一時的な認証情報を優先したアクセスキーの廃止を要求する必要があります。 AWS Identity and Access Management (IAM) ユーザーから一元化された ID に移行すると、認証情報レポートを生成してユーザーを監査できます。

また、ID プロバイダーで MFA を実施およびモニタリングすることをお勧めします。AWS Config ルールを設定するか、またはAWS Security Hub セキュリティスタンダードを使って、ユーザーの MFA が有効になっているかどうかをモニタリングできます。IAM Roles Anywhere を使って、マシン ID の一時的な認証情報を提供することを検討してください。IAM ロールと一時的な認証情報の使用が不可能なときは、アクセスキーの監査および更新の頻度を高めることが重要です。

実装手順

- 認証情報を定期的に監査する: ID プロバイダーと IAM で設定されている ID を監査することで、承認された ID のみがワークロードにアクセスできるようになります。こういった ID には、IAM ユーザー、AWS IAM Identity Center ユーザー、Active Directory ユーザー、またはさまざまなアップストリーム ID プロバイダーのユーザーを含みますが、これらに限定されません。たとえば、組織を離れた人を削除したり、不要になったクロスアカウントのロールを削除したりします。IAM エンティティがアクセスするサービスへのアクセス許可を定期的に監査するプロセスを用意します。これにより、未使用のアクセス許可を削除するために変更する必要があるポリシーを特定できます。認証情報レポートと

AWS Identity and Access Management Access Analyzer を使用して、IAM 認証情報とアクセス許可を監査します。Amazon CloudWatch を使って、AWS 環境内で呼び出される特定の API コールのアラームを設定します。Amazon GuardDuty は、想定外のアクティビティがあるとアラートを発動します。これは、IAM 認証情報に対する過度に寛容なアクセスまたは意図しないアクセスを示している可能性があります。

- 認証情報を定期的にローテーションする: 一時的な認証情報を使用できない場合、長期的 IAM アクセスキーを定期的にローテーションしてください (最大 90 日ごと)。知らない間にアクセスキーが開始された場合でも、これによりその認証情報を使ってリソースにアクセスされる期間を制限できます。IAM ユーザーのアクセスキーのローテーションについては、「アクセスキーのローテーション」を参照してください。
- IAM アクセス許可を確認する: AWS アカウント のセキュリティを改善するには、各 IAM ポリシーを定期的に確認してモニタリングします。ポリシーが最小特権の原則に準拠していることを確認します。
- IAM リソース作成および更新の自動化を検討する: IAM Identity Center は、ロールやポリシー管理など多くの IAM タスクを自動化します。または、AWS CloudFormation を使用すると、テンプレートを検証してバージョンを管理できるため、ロールやポリシーを含む IAM リソースのデプロイを自動化して、人為的ミスが生じる可能性を減らすことができます。
- IAM Roles Anywhere を使用して、マシン ID の IAM ユーザーを置換する: IAM Roles Anywhere を使用すると、オンプレミスサーバーなど、従来は不可能であった領域でロールを使用できるようになります。IAM Roles Anywhere は、信頼された X.509 証明書を使って AWS を認証し、一時的な認証情報を受け取ります。IAM Roles Anywhere を使用することにより、長期的認証情報がオンプレミス環境に保管されなくなるため、これらの認証情報をローテーションする必要がなくなります。X.509 証明書の有効期限が近づいたら、モニタリングとローテーションが必要となることに注意してください。

リソース

関連するベストプラクティス:

- SEC02-BP02 一時的な認証情報を使用する (p. 22)
- SEC02-BP03 シークレットを安全に保存して使用する (p. 24)

関連するドキュメント:

- Getting started with AWS Secrets Manager (Amazon SQS の開始方法)
- IAM ベストプラクティス
- ID プロバイダーとフェデレーション
- Security Partner Solutions: Access and Access Control (セキュリティパートナーソリューション: アクセスおよびアクセスコントロール)
- 一時的なセキュリティ認証情報
- AWS アカウント アカウントの認証情報レポートの取得

関連動画:

- Best Practices for Managing, Retrieving, and Rotating Secrets at Scale (シークレットを大規模に管理、取得、変更するためのベストプラクティス)
- Managing user permissions at scale with AWS IAM Identity Center (AWS SSO を使用した大規模なユーザー権限の管理)
- Mastering identity at every layer of the cake (すべての層での ID の把握)

関連する例:

- Well-Architected ラボ - IAM ユーザーの自動クリーンアップ

- Well-Architected ラボ - IAM グループおよびロールの自動デプロイ

SEC02-BP06 ユーザーグループと属性を活用する

管理対象のユーザー数が増えるにつれて、大規模な管理ができるユーザー管理方法が必要となります。一般的なセキュリティ要件を持つユーザーを ID プロバイダーで定義したグループに分け、アクセスコントロールに使用される可能性のあるユーザー属性 (部署や場所など) を最新で正確な状態に保つメカニズムを導入します。アクセス制御には、個々のユーザーではなくこのグループと属性を使用します。こうすると、アクセス許可セットを使用してユーザーのグループメンバーシップや属性を一度変更するだけで アクセスを一元管理でき、ユーザーのアクセスに変更が必要なときに多数のポリシーを個別に更新せずに済みます。ユーザーグループや属性の管理に AWS IAM Identity Center (IAM Identity Center)を使用できます。IAM Identity Center は、一般的に使用されている属性に対応しています。ユーザー作成時の手動入力も、クロスドメイン ID 管理システム (SCIM) 仕様などで定義された同期エンジンを使用した自動プロビジョニングも可能です。

一般的なセキュリティ要件を持つユーザーを ID プロバイダーで定義したグループに分け、アクセスコントロールに使用される可能性のあるユーザー属性 (部署や場所など) を最新で正確な状態に保つメカニズムを導入します。アクセスを制御するには、個々のユーザーではなくこれらのグループと属性を使用します。これにより、ユーザーのアクセスニーズが変化したときに多くの個別のポリシーを更新することなく、ユーザーのグループメンバーシップや属性を 1 回変更することで、アクセスを一元管理できます。

このベストプラクティスが確立されていない場合のリスクレベル: 低

実装のガイダンス

- AWS IAM Identity Center (IAM Identity Center)を使用している場合、グループを設定します: IAM Identity Center では、ユーザーのグループを設定し、必要なレベルのアクセス許可をグループに割り当てることができます。
  - AWS シングルサインオン - アイデンティティの管理
- 属性ベースのアクセスコントロール (ABAC) について学ぶ: ABAC は、属性に基づいてアクセス許可を定義する認証戦略です。
  - AWS の ABAC とは
  - ラボ: EC2 の IAM タグベースのアクセスコントロール

リソース

関連するドキュメント:

- AWS Secrets Manager の開始方法
- IAM のベストプラクティス
- ID プロバイダーとフェデレーション
- AWS アカウントのルートユーザー

関連動画:

- Best Practices for Managing, Retrieving, and Rotating Secrets at Scale (シークレットを大規模に管理、取得、変更するためのベストプラクティス)
- Managing user permissions at scale with AWS IAM Identity Center (AWS IAM Identity Center を使用した大規模なユーザー権限の管理)
- すべての層での ID の把握

関連する例:

- ラボ: EC2 の IAM タグベースのアクセスコントロール

Permissions management

アクセス許可を管理して、AWS とワークロードへのアクセスを必要とするユーザー ID やマシン ID へのアクセスを制御します。権限を分けることで、どのような条件で誰が何にアクセスできるかを制御します。特定のユーザー ID およびマシン ID にアクセス権限を設定し、必要とするリソースに対するサービスアクションへのアクセスのみを許可します。さらに、アクセスを取得するために満たすべき条件を指定します。例えば、特定のリージョンのみで新しい Lambda 関数を作成することをデベロッパーに許可できます。大規模な AWS 環境を管理する場合、以下のベストプラクティスに従って、それぞれのアイデンティティに必要なアクセスのみを許可し、必要以上に設定しないようにします。

さまざまなタイプのリソースにアクセスを付与する方法は多数あります。その 1 つは、異なるポリシータイプを使用する方法です。

IAM での アイデンティティベースのポリシーは、マネージドまたはインラインで、ユーザー、グループ、ロールなどの IAM アイデンティティにアタッチされます。これらのポリシーでは、そのアイデンティティができる内容 (そのアクセス許可) を指定できます。アイデンティティベースのポリシーはさらに分類できます。

マネージドポリシー – スタンドアロンのアイデンティティベースのポリシーで、AWS アカウントで複数のユーザー、グループ、およびロールにアタッチできます。マネージドポリシーには 2 つのタイプがあります。

- AWS マネージドポリシー – AWS によって作成および管理されるマネージドポリシー。
- カスタマー管理ポリシー – AWS アカウントで作成および管理するマネージドポリシー。カスタマー管理ポリシーでは、AWS マネージドポリシーよりも正確にポリシー管理できます。

アクセス許可を付与するには、マネージドポリシーのほうが好ましい方法です。ただし、単一のユーザー、グループ、ロールに直接追加するインラインポリシーを使用することもできます。インラインポリシーでは、ポリシーとアイデンティティ間に厳格な 1 対 1 の関係を維持します。アイデンティティを削除すると、インラインポリシーは削除されます。

ほとんどの場合、最小特権の原則に従って独自のカスタマー管理ポリシーを作成する必要があります。

リソースベースのポリシーは、リソースにアタッチされます。例えば、Amazon S3 バケットポリシーはリソースベースのポリシーです。これらのポリシーでは、リソースと同じアカウントまたは別のアカウントにあるプリンシパルにアクセス許可を付与します。リソースベースのポリシーをサポートするサービスの一覧については、「IAM と連携する AWS のサービス」を参照してください。

アクセス許可の境界は、マネージドポリシーを使用して、管理者が設定できるアクセス許可の上限を設定できます。これによって、IAM ロール作成などのアクセス許可の作成および管理の権限を開発者に委任しながらも、付与できるアクセス許可を制限して、自分でそのアクセス許可の範囲を拡大できないように制限できます。

属性ベースのアクセスコントロール (ABAC) では、属性に基づいてアクセス許可を付与することができます。AWS では、これをタグと呼びます。タグは、IAM プリンシパル (ユーザーまたはロール) と AWS リソースにアタッチできます。IAM ポリシーを使うと、管理者は再利用可能なポリシーを作成して IAM プリンシパルの属性に基づいたアクセス許可を適用できます。例えば、管理者は 1 つの IAM ポリシーを使用して、開発者のプロジェクトタグに一致する AWS リソースへのアクセス権を組織内の開発者に付与できます。開発者チームがプロジェクトにリソースを追加すると、属性に基づきそれに対するアクセス許可が自動的に適用されます。このため、リソースが追加されるたびにポリシーを更新する必要はありません。

Organizations サービスコントロールポリシー (SCP) は、組織または組織単位 (OU) のアカウントメンバーの最大許容を定義します。SCP は、アイデンティティベースのポリシーやリソースベースのポリシーがア

カウント内のエンティティ (ユーザーやロール) に付与する許可を制限するものですが、許可を付与するものではありません。

セッションポリシーは、ロールまたはフェデレーションユーザーを引き受けます。AWS CLI または AWS API セッションポリシーを使ってロールまたはユーザーのアイデンティティベースのポリシーがセッションに付与する許可を制限する際、セッションポリシーを渡します。これらのポリシーは、作成されたセッション許可を制限するものですが、許可を付与するものではありません。詳細については「セッションポリシー」を参照してください。

ベストプラクティス

SEC03-BP01 アクセス要件を定義する

ワークロードの各コンポーネントまたはリソースには、管理者、エンドユーザー、またはその他のコンポーネントからアクセスする必要があります。各コンポーネントにアクセスできるユーザーや内容を明確に定義し、適切な ID タイプと認証および承認の方法を選択します。

一般的なアンチパターン:

- シークレットをハードコーディングする、またはアプリケーション内に格納する
- 各ユーザーにカスタムのアクセス許可を付与する
- 永続的な認証情報を使用する

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

ワークロードの各コンポーネントまたはリソースには、管理者、エンドユーザー、またはその他のコンポーネントからアクセスする必要があります。各コンポーネントにアクセスできるユーザーや内容を明確に定義し、適切な ID タイプと認証および承認の方法を選択します。

組織内の AWS アカウントへの通常のアクセスは、フェデレーションアクセス または一元化された ID プロバイダーを使用して提供する必要があります。また、アイデンティティ管理を一元化し、AWS へのアクセスを従業員のアクセスライフサイクルに統合するための確立されたプラクティスを整備する必要があります。例えば、従業員がアクセスレベルの異なる職種に異動するときは、そのグループメンバーシップも新しいアクセス要件を反映するように変更される必要があります。

非人間アイデンティティのアクセス要件を定義するときは、どのアプリケーションとコンポーネントがアクセスを必要としているか、またアクセス許可をどのように付与するかを決定します。お勧めのアプローチは、最小特権アクセスモデルで構築された IAM ロールを使用する方法です。AWS マネージドポリシーは、最も一般的なユースケースをカバーする定義済みの IAM ポリシーを提供します。

AWS のサービス (AWS Secrets Manager や AWS Systems Manager パラメータストア) を使用すると、IAM ロールの使用が不可能なケースで、シークレットをアプリケーションやワークロードから安

全に切り離すことができます。Secrets Manager では、認証情報の自動ローテーションを確立できます。Systems Manager でパラメータの作成時に指定した一意の名前を使用することで、スクリプト、コマンド、SSM ドキュメント、設定、オートメーションワークフロー内のパラメータを参照できます。

AWS Identity and Access Management Roles Anywhere を使用すると、 IAM 内の一時的なセキュリティ認証情報を取得して、 AWS の外部で実行されるワークロードに使用できます。ワークロードに使用できる IAM ポリシー および IAM ロール は、AWS アプリケーションが AWS リソースにアクセスするために使用するものと同じです。

可能な場合は、長期の静的な認証情報よりも、短期の一時的な認証情報を優先します。IAM ユーザーに、プログラムによるアクセスと長期の認証情報を付与する必要がある場合は、 最後に使用されたアクセスキーの情報を使用し、 アクセスキーのローテーションと削除を行います。

リソース

関連するドキュメント:

- Attribute-based access control (ABAC)
- AWS IAM Identity Center
- IAM Roles Anywhere
- AWS Managed policies for IAM Identity Center (IAM アイデンティティセンター用の AWS マネージドポリシー)
- AWS IAM policy conditions (AWS IAM ポリシー条件)
- IAM ユースケース
- 必要でない認証情報を削除する
- 「IAM ポリシーを管理する」
- How to control access to AWS resources based on AWS アカウント, OU, or organization (AWS アカウント、OU、または組織に基づいて AWS リソースへのアクセスを制御する方法)
- Identify, arrange, and manage secrets easily using enhanced search in AWS Secrets Manager (AWS Secrets Manager の拡張検索を使用してシークレットを容易に特定、調整、管理する)

関連動画:

- Become an IAM Policy Master in 60 Minutes or Less (60 分以内に IAM ポリシーマスターになる)
- Separation of Duties, Least Privilege, Delegation, and CI/CD (職務分離、最小特権、委任、および CI/CD)
- Streamlining identity and access management for innovation (アイデンティティとアクセスの管理を合理化してイノベーションを実現)

SEC03-BP02 最小特権のアクセスを付与します

特定の条件下で特定のリソースに対する特定のアクションを実行するために ID が必要とするアクセス許可のみを付与するのがベストプラクティスです。グループと ID 属性を使用して、個々のユーザーのアクセス許可を定義するのではなく、規模に応じてアクセス許可を動的に設定します。例えば、開発者のグループに、扱うプロジェクトのリソースのみを管理することを許可できます。これにより、開発者がプロジェクトから離れると、基盤となるアクセスポリシーに変更を加えることなく、その開発者のアクセスは自動的に取り消されます。

期待される成果: ユーザーは、ジョブの実行に必要なアクセス許可のみを持つ必要があります。ユーザーには、限られた時間内に特定のタスクを実行するためだけに本番環境へのアクセスが与えられ、タスクが完了したらアクセスを取り消す必要があります。アクセス許可は、ユーザーが別のプロジェクトまたは職務に移った場合を含め、不要になったときに取り消す必要があります。管理者権限は、信頼できる管理者の

少数のグループのみに付与する必要があります。アクセス許可の変化を避けるため、アクセス許可は定期的にレビューする必要があります。マシンまたはシステムアカウントには、タスクを完了するために必要な最小セットのアクセス許可を付与する必要があります。

一般的なアンチパターン:

- デフォルトでユーザーに管理者アクセス許可を付与する
- ルートユーザーを日常業務に使用する
- 過度に寛容でありながら、完全な管理者権限がないポリシーを作成する。
- アクセス許可をレビューして、最小特権アクセスを許可するかどうかを把握しない。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

最小特権の原則には、特定のタスクの遂行に必要な最小セットのアクションを実行する許可のみを ID に付与する必要があると記載されています。これは、ユーザビリティ、効率性、セキュリティのバランスを取ります。この原則の下で運用すると、意図しないアクセスを制限し、誰がどのリソースにアクセスできるかを追跡するのに役立ちます。デフォルトでは、IAM ユーザーとロールにはアクセス許可はありません。ルートユーザーにはデフォルトでフルアクセスがあり、厳格に制御、監視する必要があり、ルートアクセスが必要なタスクにのみ使用する必要があります。

IAM ポリシーは、IAM ロールまたは特定のリソースに明示的にアクセス許可を付与するために使用されます。例えば、アイデンティティベースのポリシーは、IAM グループにアタッチでき、S3 バケットはリソースベースのポリシーで制御できます。

IAM ポリシーの作成では、AWS がアクセスを許可または拒否するために必要なサービスアクション、リソース、条件を指定できます。AWS では、アクセスを最小限にするために役立つさまざまな条件を用意しています。例えば、依頼者が AWS 組織に属していない場合、`PrincipalOrgID` 条件キーを使用して、アクションを拒否できます。

AWS のサービスがユーザーに代わって行うリクエスト (AWS CloudFormation による AWS Lambda 関数の作成など) を制御するには、`CalledVia` 条件キーを使用します。異なるポリシータイプを層にして、深層防御を確立し、ユーザーの全体的なアクセス許可を制限する必要があります。どのアクセス許可がどのような条件の下で付与できるかも制限できます。例えば、アプリケーションチームが独自の IAM ポリシーを作成することは許可できますが、アクセス許可の境界 を適用して、チームが受け取る最大のアクセス許可を制限する必要もあります。

実装手順

- 最小特権ポリシーを実装する: IAM グループおよびロールに最小特権のアクセスポリシーを割り当てて、定義したユーザーのロールまたは機能を反映します。
  - API 使用状況に関するベースポリシー: 必要なアクセス許可を判断する 1 つの方法は、AWS CloudTrail ログをレビューすることです。このレビューでは、ユーザーが AWS 内で実際に実行するアクションに合わせてカスタマイズされたアクセス許可を作成できます。IAM Access Analyzer は、アクティビティに基づいて IAM ポリシーを自動生成できます。IAM Access Advisor を組織またはアカウントレベルで使用して、特定のポリシーの最終アクセスの情報を追跡できます。
- 職務に応じた AWS マネージドポリシーの使用を検討してください。 きめ細かいアクセス許可ポリシーの作成を開始するとき、どこから始めればよいかわからない場合があります。AWS には、例えば請求、データベース管理者、データサイエンティストなど、一般的な職種に対するマネージドポリシーがあります。これらのポリシーは、最小特権ポリシーの実装方法を判断している間、ユーザーの持つアクセスを絞り込むことができます。
- 必要でないアクセス許可を削除する: 必要でないアクセス許可を削除し、過度に寛容なポリシーを削ります。IAM Access Analyzer ポリシー生成は、アクセス許可ポリシーの微調整に役立ちます。
- ユーザーの本番環境へのアクセスが制限されるようにする: ユーザーは、有効なユースケースのある本番環境にのみアクセスできるようにする必要があります。ユーザーが、本番稼働アクセスが必要な特定

のタスクを実行した後は、アクセスを取り消す必要があります。本番環境へのアクセスを制限することは、本番に影響する意図しないイベントを回避するのに役立ち、意図しないアクセスの影響範囲を狭めます。

- アクセス許可の境界を考慮する: アクセス許可の境界は、アイデンティティベースのポリシーが IAM エンティティに付与できるアクセス許可の上限を設定する管理ポリシーを使用するための機能です。エンティティのアクセス許可の境界では、アイデンティティベースのポリシーとそのアクセス許可の境界の両方で許可されているアクションのみを実行できます。
- アクセス許可のリソースタグを検討する: リソースタグを使用する属性ベースのアクセスコントロールモデルでは、リソースの目的、所有者、環境、またはその他の基準に基づいてアクセスを付与できます。例えば、リソースタグを使用して、開発と本番環境を区別することができます。これらのタグを使用して、開発者を開発環境に制限することができます。タグ付けとアクセス許可ポリシーを組み合わせることで、きめ細かいリソースアクセスを達成でき、すべての職務に複雑な、カスタムポリシーを定義する必要がなくなります。
- AWS Organizations のサービスコントロールポリシーを使用します。 サービスコントロールポリシーは、組織のメンバーアカウントで利用できる最大のアクセス許可を一元管理します。重要なのは、サービスコントロールポリシーでは、メンバーアカウントでルートユーザーのアクセス許可を制限できることです。AWS Organizations を強化する規範的マネージドコントロールを提供する、AWS Control Tower の使用も検討してください。Control Tower 内で独自のコントロールを定義できます。
- 組織のユーザーライフサイクルポリシーを確立する: ユーザーライフサイクルポリシーは、ユーザーが AWS にオンボードされたとき、ジョブロールまたはスコープを変更したとき、または AWS へのアクセスが不要になったときに実行するタスクを定義します。アクセス許可レビューは、ユーザーのライフサイクルの各ステップで実行し、アクセス許可が適切に制限されていることを検証して、アクセス許可の変化を回避します。
- アクセス許可をレビューする定期的スケジュールを確立し、不要なアクセス許可を削除する: ユーザーアクセスを定期的にレビューして、過度に寛容なアクセスがないことを検証する必要があります。AWS Config および IAM Access Analyzer は、ユーザーアクセス許可を監査するときに役立ちます。
- 職種マトリックスを確立する: 職種マトリックスは、AWS フットプリント内で必要なさまざまなロールとアクセスレベルを可視化します。職種マトリックスを使用して、組織内でのユーザーの責任に基づいてアクセス許可を定義し、分離できます。個々のユーザーまたはロールにアクセス許可を直接適用する代わりに、グループを使用します。

リソース

関連するドキュメント:

- 最小特権を付与する
- IAM エンティティのアクセス許可の境界
- Techniques for writing least privilege IAM policies (最小特権の IAM ポリシーを作成するテクニック)
- IAM Access Analyzer makes it easier to implement least privilege permissions by generating IAM policies based on access activity (IAM Access Analyzer は、アクセスアクティビティに基づいて IAM ポリシーを生成することにより、最小特権のアクセス許可の実装を容易にする)
- IAM アクセス許可の境界を使用して開発者にアクセス許可管理を委任する
- 最終アクセス情報を使用した AWS のアクセス許可の調整
- IAM でのポリシータイプと使用する状況
- IAM ポリシーシミュレーターを使用した IAM ポリシーのテスト
- AWS Control Tower のガードレール
- Zero Trust architectures: An AWS perspective (ゼロトラストアーキテクチャ: AWS の視点)
- How to implement the principle of least privilege with CloudFormation StackSets (CloudFormation StackSets を使用して最小特権の原則を実装する方法)
- Attribute-based access control (ABAC)

- ユーザーアクティビティを確認してポリシーの範囲を削減する
- ロールアクセスを表示する
- タグ付けを使用して環境を整理しアカウンタビリティを促進
- AWS のタグ付け戦略
- AWS リソースのタグ付け

関連動画:

- Next-generation permissions management (次世代のアクセス許可管理)
- Zero Trust: An AWS perspective (ゼロトラスト: AWS の視点)
- How can I use permissions boundaries to limit users and roles to prevent privilege escalation? (アクセス許可の境界を使用して IAM ユーザーとロールの範囲を限定し、権限の昇格を防ぐにはどうすればよいですか?)

関連する例:

- ラボ: ロールの作成を委任する IAM アクセス許可の境界
- ラボ: EC2 の IAM タグベースのアクセスコントロール

SEC03-BP03 緊急アクセスのプロセスを確立する

一元化された ID プロバイダーで万一問題が発生した場合に備え、ワークロードへの緊急アクセスを許可するプロセスを作成します。

緊急事態につながる可能性のあるさまざまな障害モードに対応するプロセスを設計する必要があります。例えば、通常の状況では、従業員ユーザーはクラウドへのフェデレーションに一元化された ID プロバイダー (SEC02-BP04) を使用して、ワークロードを管理します。ただし、一元化された ID プロバイダーに障害が発生した場合や、クラウドのフェデレーションの設定が変更された場合、従業員ユーザーはクラウドにフェデレーションできなくなる可能性があります。緊急アクセスのプロセスでは、権限を持つ管理者がフェデレーション設定やワークロードの問題を解決するために、代替手段 (代替のフェデレーションやユーザーの直接アクセスなど) を通じてクラウドリソースにアクセスすることを許可します。緊急アクセスのプロセスは、通常のフェデレーションメカニズムが復旧するまで使用されます。

期待される成果:

- 緊急事態と見なされる障害モードを定義して文書化します。通常の状況と、ユーザーがワークロードの管理に使用するシステムを考慮してください。それぞれの依存関係でどのように障害が発生しうるか、またその障害がどのように緊急事態を引き起こすかを検討します。障害モードを特定し、障害の可能性を最小限に抑える高い回復力を備えたシステムを構築するうえで役立つ質問とベストプラクティスは、信頼性の柱 で確認できます。
- 障害が緊急事態であると確認する際に従うべき手順を文書化します。例えば、ID 管理者がプライマリ ID プロバイダーとスタンバイ ID プロバイダーのステータスを確認すること、両方とも使用できない場合は、ID プロバイダーに障害が発生した場合の緊急事態を宣言することを要求できます。
- 各種の緊急モードまたは障害モードに固有の緊急アクセスプロセスを定義します。具体的に定義することで、ユーザーが緊急事態の種類にかかわらず一般的なプロセスを使いすぎる状況を減らすことができます。緊急アクセスのプロセスで、各プロセスを使用すべき状況、逆にそのプロセスを使用すべきでない状況、適用される可能性のある代替プロセスを説明します。
- 詳細な指示とプレイブックを含めてプロセスが十分に文書化されており、迅速かつ効率的に実行できます。緊急事態はユーザーにとってストレスの多い時間であり、ユーザーは極度の時間的プレッシャーにさらされる可能性があるため、プロセスはできるだけシンプルに設計してください。

一般的なアンチパターン:

- 緊急アクセスプロセスの文書化およびテストが不十分である。ユーザーは緊急事態への備えができておらず、緊急事態が発生しても即席のプロセスに従う。
- 緊急アクセスのプロセスで、通常のアクセスメカニズムと同じシステム (一元化された ID プロバイダーなど) を使用する。これにより、このようなシステムの障害が、通常および緊急の両方のアクセスメカニズムに影響を及ぼし、障害からの回復能力が損なわれる可能性がある。
- 緊急アクセスのプロセスが、緊急ではない状況で使用される。例えば、パイプラインを通じて変更を送信するよりも直接変更を加える方が簡単だと感じるため、ユーザーが頻繁に緊急アクセスプロセスを誤用する。
- 緊急アクセスのプロセスで、プロセスを監査するための十分なログが生成されていない、またはログが監視されておらずプロセスの誤用の可能性についてアラートされない。

このベストプラクティスを活用するメリット:

- 緊急アクセスのプロセスを十分に文書化し、十分にテストすることで、ユーザーが緊急事態に対応して解決するための時間を短縮できます。これにより、ダウンタイムが減少し、顧客に提供するサービスの可用性が高まります。
- 緊急アクセスのリクエストをそれぞれ追跡し、緊急事態以外の場合にプロセスを誤用しようとする不正な試みを検出してアラートすることができます。

このベストプラクティスを活用しない場合のリスクレベル: 中

実装のガイダンス

このセクションでは、AWS 上にデプロイされたワークロードに関連する複数の障害モードに対する緊急アクセスプロセス作成のためのガイダンスを提供します。すべての障害モードに適用される共通のガイダンスから始め、次に障害モードのタイプに基づいた具体的なガイダンスを示します。

すべての障害モードに共通のガイダンス

障害モードに対する緊急アクセスプロセスを設計する際は、次の点を考慮してください。

- プロセスの前提条件と仮定事項 (プロセスを使用すべき場合と使用すべきでない場合) を文書化します。障害モードを詳しく説明し、他の関連システムの状態などの仮定事項を文書化しておくと役立ちます。例えば、障害モード 2 に対するプロセスでは、ID プロバイダーは使用可能だが、AWS の設定が変更されているか、有効期限が切れていることを仮定しています。
- 緊急アクセスプロセスで必要となるリソースを事前に作成しておきます (SEC10-BP05)。例えば、IAM users とロールを持つ緊急アクセス用の AWS アカウント と、すべてのワークロードアカウントでのクロスアカウントの IAM ロールを事前に作成します。これにより、緊急事態の発生時にリソースが準備され使用可能である状態を確保することができます。リソースを事前に作成しておくことで、緊急時に利用できなくなる可能性のある AWS コントロールプレーン API (AWS リソースの作成と変更に使用) に依存する必要がなくなります。さらに、IAM リソースを事前に作成しておくと、 結果整合性のための遅延の可能性を考慮する必要がなくなります。
- インシデント管理計画に緊急アクセスプロセスを含めます (SEC10-BP02)。緊急事態の追跡方法、同僚チームやリーダーシップなど組織内の他のメンバーや、該当する場合は外部の顧客やビジネスパートナーへの伝達方法を文書化します。
- 既存のサービスリクエストワークフローシステム (ある場合) で緊急アクセスリクエストプロセスを定義します。通常、このようなワークフローシステムでは、リクエストに関する情報を収集する受付フォームを作成したり、ワークフローの各段階でリクエストを追跡したり、自動および手動の承認ステップを追加したりできます。各リクエストを、インシデント管理システムで追跡される、対応する緊急イベントに関連付けます。緊急アクセス用の統一されたシステムがあると、こうしたリクエストを単一のシステムで追跡し、使用傾向を分析して、プロセスを改善できます。
- 緊急アクセスプロセスは権限を持つユーザーのみが開始できることと、必要に応じてそのユーザーの同僚または管理層の承認が必要であることを確認します。承認プロセスは、営業時間の内外で効果的に実

施される必要があります。承認リクエストについて、一次承認者が不在の場合はどのように二次承認者を許可するのか、承認を受けるまで、どのように一連の管理層にリクエストをエスカレーションするのかを定義します。

- 緊急アクセスに成功した場合と失敗した場合の両方について、詳細な監査ログとイベントが生成されることを確認します。リクエストプロセスと緊急アクセスメカニズムの両方を監視して、誤用や不正アクセスを検出します。インシデント管理システムから進行中の緊急事態とアクティビティを関連付け、想定時間外に障害が発生した場合にアラートを発します。例えば、緊急アクセス AWS アカウント でのアクティビティを監視してアラートする必要があります。こうしたアクティビティは通常の操作では使用されるべきではありません。
- 緊急アクセスプロセスを定期的にテストして、手順が明確であること、適切なレベルのアクセス権を迅速かつ効率的に付与できることを確認します。緊急アクセスプロセスは、インシデント対応シミュレーション (SEC10-BP07) およびディザスタリカバリテスト (REL13-BP03) の一環としてテストする必要があります。

障害モード 1: AWS へのフェデレーションに使用する ID プロバイダーが使用できない

「 SEC02-BP04 一元化された ID プロバイダーを利用する」で説明したとおり、ワークフォースユーザーをフェデレーションして AWS アカウント へのアクセス権を付与するには、一元化された ID プロバイダーを利用することが推奨されます。IAM Identity Center を使用して AWS 組織内の複数の AWS アカウント にフェデレーションするか、IAM を使用して個別の AWS アカウント にフェデレーションすることができます。いずれの場合も、ワークフォースユーザーは、シングルサインオンのために AWS へのサインインエンドポイントにリダイレクトされる前に、一元化された ID プロバイダーで認証されます。

万一、一元化された ID プロバイダーが利用できなくなった場合、ワークフォースユーザーは AWS アカウント にフェデレーションすることも、ワークロードを管理することもできなくなります。こうした緊急事態には、AWS アカウント にアクセスするための緊急アクセスプロセスを少数の管理者に提供し、一元化された ID プロバイダーのオンライン復帰を待つ余裕のない重要なタスクを実行できるようにします。例えば、ID プロバイダーが 4 時間利用できない場合、その間、顧客トラフィックの想定外の急増に対応するために、本番稼働用アカウントの Amazon EC2 Auto Scaling グループの上限を変更する必要が生じたとします。その場合、緊急管理者は、緊急アクセスプロセスに従って特定の本番稼働用 AWS アカウント へのアクセス権を取得し、必要な変更を加える必要があります。

緊急アクセスプロセスでは、事前に作成されている緊急アクセス用 AWS アカウント を使用します。このアカウントは緊急アクセスの目的でのみ使用され、緊急アクセスプロセスに対応するための AWS リソース (IAM ロールや IAM users など) が設定されています。通常の操作中は、誰も緊急アクセスアカウントにアクセスしてはならず、このアカウントの誤用については監視してアラートする必要があります (詳細については、前述の「共通のガイダンス」セクションを参照してください)。

緊急アクセス用アカウントには、緊急アクセスを必要とする AWS アカウント でクロスアカウントロールを引き受ける権限を持つ、緊急アクセス IAM ロールがあります。これらの IAM ロールは事前に作成され、緊急アカウントの IAM ロールを信頼する信頼ポリシーで設定されています。

緊急アクセスプロセスでは、次のいずれかの方法を使用できます。

- 緊急アクセスアカウントで、緊急管理者用の強力なパスワードと MFA トークンを設定した一連の IAM users を事前に作成できます。これらの IAM users には、IAM ロールを引き受け、緊急アクセスが必要な AWS アカウント へのクロスアカウントアクセスを許可する権限があります。このようなユーザーはできるだけ少人数にし、各ユーザーを 1 人の緊急管理者に割り当てることが推奨されます。緊急時には、緊急管理者ユーザーがパスワードと MFA トークンコードを使用して緊急アクセス用アカウントにサインインし、緊急アカウントで緊急アクセス IAM ロールに切り替え、最後にワークロードアカウントで緊急アクセス IAM ロールに切り替えて、緊急の変更アクションを実行します。この方法の利点は、それぞれの IAM user が 1 人の緊急管理者に割り当てられるため、CloudTrail イベントを確認することで、どのユーザーがサインインしたかを把握できることです。欠点は、複数の IAM users と、それぞれに関連付けられた永続的なパスワードと MFA トークンを管理しなければならないことです。
- 緊急アクセス用 AWS アカウント ルートユーザー を使用して緊急アクセスアカウントにサインインし、緊急アクセスの IAM ロールを引き受け、ワークロードアカウントでクロスアカウントロールを引き受

けることができます。ルートユーザーには強力なパスワードと複数の MFA トークンを設定することが推奨されます。また、パスワードと MFA トークンは、強力な認証と承認を実行する安全なエンタープライズ認証情報ボールトに保管することをお勧めします。パスワードと MFA トークンのリセット要因を確保する必要があります。アカウントの E メールアドレスを、クラウドセキュリティ管理者が監視するメール配布リストに設定し、アカウントの電話番号は、同様にセキュリティ管理者が監視する共有電話番号に設定します。この方法の利点は、管理するルートユーザーの認証情報が 1 セットだけであることです。欠点は、これが共有ユーザーであるため、複数の管理者がルートユーザーとしてサインインできてしまうことです。エンタープライズボールトのログイベントを監査して、どの管理者がルートユーザーのパスワードをチェックアウトしたかを特定する必要があります。

障害モード 2: AWS の ID プロバイダー設定が変更された、または有効期限が切れている

ワークフォースユーザーが AWS アカウント にフェデレーションできるようにするには、外部 ID プロバイダーを使用して IAM Identity Center を設定するか、IAM ID プロバイダーを作成します (SEC02-BP04)。通常、これらを設定するには、ID プロバイダーが提供する SAML メタデータ XML ドキュメントをインポートします。メタデータ XML ドキュメントには、ID プロバイダーが SAML アサーションの署名に使用するプライベートキーに対応する X.509 証明書が含まれています。

AWS 側でのこれらの設定は、管理者が誤って変更または削除する可能性があります。もう 1 つのシナリオとして、AWS にインポートされた X.509 証明書の有効期限が切れ、新しい証明書を含む新しいメタデータ XML が AWS にインポートされていない場合があります。いずれの場合も、ワークフォースユーザーの AWS へのフェデレーションが失敗し、緊急事態が発生する可能性があります。

こうした緊急事態には、フェデレーションの問題を解決するための AWS へのアクセスを ID 管理者に提供できます。例えば、ID 管理者は緊急アクセスプロセスを使用して緊急アクセス用 AWS アカウント にサインインし、アイデンティティセンター管理者アカウントのロールに切り替え、ID プロバイダーから提供された最新の SAML メタデータ XML ドキュメントをインポートして外部 ID プロバイダーの設定を更新することで、フェデレーションを再有効化することができます。フェデレーションが修正されたら、ワークフォースユーザーは引き続き通常の操作プロセスに従ってワークロードアカウントにフェデレーションします。

前述の「障害モード 1」で説明した方法に従って、緊急アクセスプロセスを作成できます。アイデンティティセンター管理者アカウントだけにアクセスし、そのアカウントでアイデンティティセンター上でのアクションを実行するための最小権限のアクセス権を、ID 管理者に付与できます。

障害モード 3: ID センターの中断

万一 IAM Identity Center または AWS リージョン の中断が発生した場合に備えて、AWS Management Console への一時的なアクセスを提供するための構成を設定しておくことが推奨されます。

緊急アクセスプロセスでは、ID プロバイダーから緊急アカウントの IAM への直接フェデレーションを使用します。プロセスと設計上の考慮事項の詳細については、 Set up emergency access to the AWS Management Consoleを参照してください。

実装手順

すべての障害モードで共通の手順

- 緊急アクセスプロセス専用の AWS アカウント を作成します。IAM ロールや IAM users、IAM ID プロバイダー (オプション) など、アカウントで必要となる IAM リソースを事前に作成しておきます。さらに、緊急アクセスアカウントで対応する IAM ロールとの信頼関係を持つ、ワークロード AWS アカウント でクロスアカウントの IAM ロールを事前に作成します。この場合、 AWS CloudFormation StackSets と AWS Organizations を使用して、組織内のメンバーアカウントでこうしたリソースを作成できます。

- AWS Organizations サービスコントロールポリシー (SCP) を作成して、メンバー AWS アカウント のクロスアカウント IAM ロールの削除と変更を拒否します。

- 緊急アクセス AWS アカウント の CloudTrail を有効にし、ログ収集 AWS アカウント の中央の S3 バケットに証跡イベントを送信します。AWS Control Tower を使用して AWS マルチアカウント環境を設定・管理している場合は、AWS Control Tower を使用して作成した、または AWS Control Tower に登録したすべてのアカウントではデフォルトで CloudTrail が有効になっており、専用のログアーカイブ AWS アカウント の S3 バケットに送信されます。
- 緊急 IAM ロールごとのコンソールログインと API アクティビティに一致する EventBridge ルールを作成して、緊急アクセスアカウントのアクティビティを監視します。インシデント管理システムで追跡されている進行中の緊急事態以外でアクティビティが発生した場合は、セキュリティオペレーションセンターに通知を送信します。

「障害モード 1: AWS へのフェデレーションに使用する ID プロバイダーが使用できない」、「障害モード 2: AWS の ID プロバイダー設定が変更された、または有効期限が切れている」の追加手順

- 緊急アクセス用に選択したメカニズムに応じて、リソースを事前に作成します。
  - IAM users を使用する: 強力なパスワードと関連付けられた MFA デバイスを持つ IAM users を事前に作成します。
  - 緊急アカウントのルートユーザーを使用する: ルートユーザーに強力なパスワードを設定し、そのパスワードをエンタープライズ認証情報ボールトに保存します。複数の物理 MFA デバイスをルートユーザーに関連付け、緊急管理チームのメンバーがすぐにアクセスできる場所に保管します。

「障害モード 3: ID センターの中断」の追加手順

- 「Set up emergency access to the AWS Management Console」で説明されているとおり、緊急アクセス AWS アカウント で IAM の ID プロバイダーを作成して、ID プロバイダーからの直接 SAML フェデレーションを有効にします。
- ID プロバイダーでメンバーのいない緊急オペレーショングループを作成します。
- 緊急アクセスアカウントで緊急オペレーショングループに対応する IAM ロールを作成します。

リソース

関連する Well-Architected のベストプラクティス

- SEC02-BP04 一元化された ID プロバイダーを利用する
- SEC03-BP02 最小特権のアクセスを付与します
- SEC10-BP02 インシデント管理計画を作成する
- SEC10-BP07 ゲームデーを実施する

関連するドキュメント:

- Set up emergency access to the AWS Management Console
- SAML 2.0 フェデレーティッドユーザーが AWS Management Console にアクセス可能にする
- Break glass access

関連動画:

- AWS re:Invent 2022 - Simplify your existing workforce access with IAM Identity Center
- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) deep dive

関連する例:

- AWS Break Glass Role

- AWS customer playbook framework
- AWS incident response playbook samples

SEC03-BP04 アクセス許可を継続的に削減する

チームと必要とするアクセスを決定したら、不要になったアクセス許可を削除し、最小特権のアクセス許可を達成するためのレビュープロセスを確立します。人間とマシンアクセス両方について使用しないアイデンティティとアクセス許可を継続的にモニタリングして削除します。

期待される成果: アクセス許可ポリシーは、最小特権原則に準拠する必要があります。職務やロールの定義がはっきりしてくるにつれ、アクセス許可ポリシーを見直し、必要でないアクセス許可を削除する必要があります。このアプローチにより、不注意による認証情報漏洩や不正アクセスによる影響を軽減することができます。

一般的なアンチパターン:

- デフォルトでユーザーに管理者アクセス許可を付与する
- 過度に寛容でありながら、完全な管理者権限がないポリシーを作成する。
- 不要になった後もアクセス許可ポリシーを保持する。

このベストプラクティスを確立しない場合のリスクレベル: 中

実装のガイダンス

チームやプロジェクトが始まったばかりの場合、革新とアジリティを刺激するために、寛容な許可ポリシーが使われる可能性があります。たとえば、開発またはテスト環境であれば、開発者にはさまざまなAWS サービスへのアクセスを付与できます。継続的にアクセスを評価し、アクセスを、現在のジョブを完了するために必要なサービスおよびサービスアクションのみに制限することが推奨されます。この評価は、人的およびマシン ID 両方にお薦めします。マシン ID は、システムまたはサービスアカウントと呼ばれることもありますが、AWS にアプリケーションまたはサービスへのアクセスを付与するアイデンティティです。このアクセスは、本稼働環境で特に重要です。ここでは、過剰に寛容なアクセス許可を使うと影響が大きく、顧客データを開示してしまう可能性があるためです。

AWS は、使用されていないユーザー、ロール、アクセス許可、および認証情報を特定するための方法を複数提供しています。AWS は、Amazon S3 バケットのオブジェクトなど AWS リソースへの関連付けられたアクセスキー、およびアクセスを含む、IAM ユーザーとロールのアクセス活動を分析するのにも役立ちます。AWS Identity and Access Management Access Analyzer ポリシー生成により、プリンシパルが実際にやりとりするサービスやアクションに基づいて、限定的な許可ポリシーを作成することができます。Attribute-based access control (ABAC) は、各ユーザーに直接権限ポリシーをアタッチするのではなく、ユーザーの属性を利用してアクセス許可を与えることができるため、アクセス権限管理の簡素化に役立ちます。

実装手順

- AWS Identity and Access Management Access Analyzerを使用する: IAM Access Analyzer は、組織内のリソースや、Amazon Simple Storage Service (Amazon S3) バケットや IAM ロールなど、外部エンティティと共有しているアカウントを特定するのに役立ちます。
- IAM Access Analyzer ポリシー生成を使用する: IAM Access Analyzer ポリシー生成は、IAM ユーザーまたはロールのアクセスアクティビティに基づいて、きめ細やかなアクセス許可ポリシーを作成するのに役立ちます。
- IAM ユーザーとロールに対して許容可能な期間と使用ポリシーを決定する: 最終アクセスタイムスタンプを使って、使用されていないユーザーとロールを特定し、それを削除します。サービスとアクションの最終アクセス時間情報を確認し、特定のユーザーおよびロールのアクセス許可を特定してスコープを

決定できます。たとえば、最終アクセス時間情報を使用して、アプリケーションロールが必要とする特定の Amazon S3 アクションを特定し、それらのアクションのみにアクセスを制限できます。最終アクセス時間情報は、AWS Management Console およびプログラムで使用でき、インフラストラクチャワークフローや自動化ツールに組み込むことができます。

- AWS CloudTrail にデータイベントを口グ記録することを検討する: デフォルトで、CloudTrail は Amazon S3 オブジェクトレベルアクティビティ (たとえば、`GetObject` および `DeleteObject`) または Amazon DynamoDB テーブルアクティビティ (たとえば、`PutItem` および `DeleteItem`) などのデータイベントを口グ記録しません。これらのイベントの口グ記録を有効にして、特定の Amazon S3 オブジェクトまたは DynamoDB テーブルアイテムにアクティビティする必要があるユーザーとロールを決定します。

リソース

関連するドキュメント:

- 最小特権を付与する
- 必要でない認証情報を削除する
- AWS CloudTrailとは?
- IAM ポリシーを管理する
- DynamoDB の口グ記録とモニタリング
- Amazon S3 バケットとオブジェクトの CloudTrail イベント口ギングの有効化
- AWS アカウント アカウントの認証情報レポートの取得

関連動画:

- Become an IAM Policy Master in 60 Minutes or Less (60 分以内に IAM ポリシーマスターになる)
- Separation of Duties, Least Privilege, Delegation, and CI/CD (職務分離、最小特権、委任、および CI/CD)
- AWS re:Inforce 2022 - AWS Identity and Access Management (IAM) deep dive (ディープダイブ)

SEC03-BP05 組織のアクセス許可ガードレールを定義する

組織内のすべての ID へのアクセスを制限する共通コントロールを確立します。例えば、特定の AWS リージョン へのアクセスを制限したり、中央セキュリティチームが使用する IAM ロールなどの一般的なリソースをオペレータが削除できないようにしたりできます。

一般的なアンチパターン:

- ワークロードを組織の管理者アカウントで実行する
- 本番稼動のワークロードと非本番稼動のワークロードを同じアカウントで実行する

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

AWS で管理するワークロードの増加に伴い、アカウントを使用してワークロードを分離し、AWS Organizations を使用してそのアカウントを管理する必要があります。組織内のすべての ID へのアクセ

スを制限するために、共通のアクセス許可ガードレールの確立を推奨しています。例えば、特定の AWS リージョンへのアクセスを制限したり、中央セキュリティチームが使用する IAM ロールなどの共通リソースをチームのメンバーが削除できないようにしたりできます。

これを実行するには、ユーザーによるキーサービスの無効化を防止するなどの、サービスコントロールポリシーの例を実装します。SCP は IAM ポリシー言語を使用し、すべての IAM プリンシパル (ユーザーとロール) が遵守するコントロールを確立します。特定の条件に基づいて、特定のサービスアクションおよびリソースへのアクセスを制限することによって、組織のアクセスコントロールのニーズを満たすことができます。ガードレールには、必要に応じて例外を定義できます。例えば、アカウント内の特定の管理者ロールを除くすべての IAM エンティティに対して、サービスアクションを制限します。

管理アカウントでのワークロードの実行は避けることをお勧めします。管理アカウントは、メンバーアカウントに影響を及ぼすセキュリティガードレールを統制およびデプロイするために使用する必要があります。一部の AWS サービスでは、委任された管理者アカウントの使用がサポートされています。この委任アカウントを使用できる場合は、管理アカウントの代わりに使用する必要があります。組織の管理者アカウントへのアクセスは、厳しく制限する必要があります。

マルチアカウント戦略を用いると、ワークロードにガードレールを適用する際の柔軟性が大幅に向上します。AWS Security Reference Architecture では、アカウント構造の設計方法に関する規範ガイダンスが提供されます。AWS Control Tower などの AWS サービスは、組織全体で予防的コントロールと発見的コントロールの両方を一元管理する機能を提供します。組織内の各アカウントまたは OU の明確な目的を定義し、その目的に沿ってコントロールを制限します。

リソース

関連するドキュメント:

- AWS Organizations
- サービスコントロールポリシー (SCP)
- Get more out of service control policies in a multi-account environment (マルチアカウント環境でサービスコントロールポリシーをさらに活用する)
- AWS Security Reference Architecture (AWS SRA)

関連動画:

- Enforce Preventive Guardrails using Service Control Policies (サービスコントロールポリシーを使用して予防的ガードレールを適用する)
- Building governance at scale with AWS Control Tower (AWS Control Tower を使用したガバナンスの大規模な構築)
- AWS Identity and Access Management deep dive (AWS Identity and Access Management の深堀り)

SEC03-BP06 ライフサイクルに基づいてアクセスを管理する

アクセスコントロールをオペレーター、アプリケーションのライフサイクル、一元化されたフェデレーションプロバイダーと統合します。たとえば、ユーザーが組織を離れるとき、またはロールを変更するときに、ユーザーのアクセス権を削除します。

複数のアカウントでワークロードを管理する場合、それらのアカウント間でリソースを共有するケースがあります。リソースの共有には、 AWS Resource Access Manager (AWS RAM) を使用することをお勧めします.このサービスを使用すると、AWS Organizations 組織および組織単位内で AWS リソースを簡単かつ安全に共有できます。AWS RAM を使用すると、共有されている組織または組織単位内外へのアカウン

トの移動に伴い、共有リソースへのアクセスの許可または取り消しが自動的に行われます。これで、意図したアカウントのみとのリソースの共有を確実に行えます。

このベストプラクティスを活用しない場合のリスクレベル: 低

実装のガイダンス

ユーザーアクセスのライフサイクル: 新しいユーザーの参加、職務の変更、退職するユーザーに対するユーザーアクセスライフサイクルポリシーを実装して、現在のユーザーのみがアクセスできるようにします。

リソース

関連するドキュメント:

- Attribute-based access control (ABAC)
- 最小権限を付与する
- IAM Access Analyzer
- 必要でない認証情報を削除する
- 「IAM ポリシーを管理する」

関連動画:

- Become an IAM Policy Master in 60 Minutes or Less
- Separation of Duties, Least Privilege, Delegation, and CI/CD

SEC03-BP07 パブリックおよびクロスアカウントアクセスの分析

パブリックおよびクロスアカウントアクセスに焦点を当てた結果を継続的にモニタリングします。パブリックアクセスとクロスアカウントアクセスを減らして、このアクセスを必要とする特定のリソースのみへのアクセスに限定します。

期待される成果: AWS リソースのうちどれが、誰と共有されるのかを把握します。共有されたリソースを継続的にモニタリングおよび監査し、認証されたプリンシパルとのみ共有されていることを確認します。

一般的なアンチパターン:

- 共有されたリソースのインベントリを保持しない。
- リソースへのクロスアカウントまたはパブリックアクセスの承認のためのプロセスを遵守しない。

このベストプラクティスが確立されていない場合のリスクレベル: 低

実装のガイダンス

アカウントが AWS Organizations にある場合、リソースへのアクセスを、組織全体、特定の組織単位、または個別のアカウントに付与することができます。アカウントが組織のメンバーでない場合、個別のアカウントとリソースを共有することができます。直接クロスアカウントアクセスを付与するには、Amazon Simple Storage Service (Amazon S3) バケットポリシーなどのリソースベースのポリシーを使用するか、別のアカウントのプリンシパルがアカウントの IAM ロールを引き受けることを許可します。リソースポリシーを使用している場合、アクセスが認証済みのプリンシパルにのみ付与されていることを確認してください。パブリックアクセス可能にする必要があるすべてのリソースを承認するプロセスを定義します。

AWS Identity and Access Management Access Analyzerは、証明可能セキュリティを使用して、アカウントの外部からリソースへのすべてのアクセスパスを識別します。また、リソースポリシーの継続的な確認と、パブリックおよびクロスアカウントアクセスの結果の報告により、広範囲なアクセス権の分析を単純化します。IAM Access Analyzer を AWS Organizations で設定して、すべてのアカウントが表示されることを確認します。IAM Access Analyzer では、リソースのアクセス許可をデプロイする前に、検出結果をプレビューすることもできます。これにより、ポリシー変更によって、意図されたパブリックアクセスおよびクロスアカウントアクセスのみがリソースに付与されていることを検証できます。マルチアカウントアクセスを設計する際、信頼ポリシーを使用して、ロールを引き受けるケースを制御できます。たとえば、PrincipalOrgId 条件キーを使用して、AWS Organizations 外からのロールを引き受けようとするのを拒否できます。

AWS Config は、設定が誤っているリソースを報告し、AWS Config ポリシーチェックを通して、パブリックアクセスが設定されたリソースを検出できます。AWS Control TowerやAWS Security Hubなどのサービスでは、AWS Organizations 全体でチェックとガードレールのデプロイが簡素化され、公開されたリソースを特定および修復します。たとえば、AWS Control Tower にはマネージド型のガードレールが含まれており、Amazon EBS スナップショットのうち AWS アカウント によって復元できるものがあるかどうかが検出されます。

実装手順

- AWS Organizations に対して AWS Config を有効化することを検討する: AWS Config では、AWS Organizations 内の複数アカウントからの検出結果を、委任された管理者アカウントに集計することができます。これにより、全体像が把握でき、アカウント全体に AWS Config ルール をデプロイして、パブリックにアクセス可能なリソースを特定できます。
- AWS Identity and Access Management Access AnalyzerIAM Access Analyzer を設定すると、外部エンティティと共有している、組織やアカウント内のリソース (Amazon S3 バケットや IAM ロールなど) を特定するのに役立ちます。
- AWS Config で自動修復を使用し、Amazon S3 バケットのパブリックアクセス設定の変更に対応します: Amazon S3 バケットに対して、パブリックアクセスのブロック設定を自動的に再有効化することができます。
- モニタリングを実装して、Amazon S3 バケットがパブリックになった場合はアラートを発動する: Amazon S3 パグリックアクセスブロックが無効な場合と、Amazon S3 バケットがパブリックになったかどうかを特定するために、モニタリングとアラートを設定しておく必要があります。さらに、AWS Organizations を使用している場合、Amazon S3 パブリックアクセスポリシーへの変更を防ぐサービスコントロールポリシーを作成する必要があります。AWS Trusted Advisor は、オープンアクセス権限がある Amazon S3 バケットをチェックします。誰にでもアクセスを付与、アップロード、削除するバケット権限は、バケットのアイテムを誰でも追加、変更、または削除できるようにすることで、セキュリティ関連の問題の原因となることがあります。Trusted Advisor のチェックは、バケットの明示的なアクセス許可を検証します。また、バケットに関連付けられたポリシーで、バケットのアクセス許可を上書きする可能性があるものについても検証します。また、AWS Config を使って、Amazon S3 バケットにパブリックアクセスがないかモニタリングできます。詳細については、「AWS Config を使って、パブリックアクセスを許可する Amazon S3 バケットをモニタリングおよび対応する」を参照してください。アクセスをレビューする間、どのようなタイプのデータが Amazon S3 バケットに含まれているかを考慮することが重要です。Amazon Macie は、PII、PHI などの機密性の高いデータ、およびプライベートまたは AWS キーなどの認証情報を検出して保護します。

リソース

関連するドキュメント:

- AWS Identity and Access Management Access Analyzer を使用する
- AWS Control Tower コントロールライブラリ
- AWS Foundational Security Best Practices 標準
- AWS Config マネージドルール

- AWS Trusted Advisor チェックリファレンス
- Amazon EventBridge を使って AWS Trusted Advisor チェック結果をモニタリングする
- 組織内のすべてのアカウントで AWS Config ルールを管理する
- AWS Config および AWS Organizations

関連動画:

- Best Practices for securing your multi-account environment (マルチアカウント環境を守るためのベストプラクティス)
- Dive Deep into IAM Access Analyzer(IAM Access Analyzer を深堀りする)

SEC03-BP08 組織内でリソースを安全に共有する

ワークロードの数が増えるにつれて、それらのワークロードのリソースへのアクセスを共有したり、複数のアカウントでリソースを複数回プロビジョニングしたりする必要が生じます。開発環境、テスト環境、本番環境などの環境を区分けするための構造があるかもしれません。ただし、分離構造があっても、安全に共有する能力は制限できません。重複するコンポーネントを共有することにより、運用諸経費を削減し、同一リソースを複数回作成する間に見逃したものを推測しなくても、一貫したエクスペリエンスを実現できます。

期待される成果: 安全な方法を使用して組織内のリソースを共有することにより、意図しないアクセスを最小限に抑え、データ損失防止イニシアチブに役立てます。個々のコンポーネントを管理するのと比較して、運用諸経費を削減し、同じコンポーネントを何度も手動で作成することによるエラーを減らし、ワークロードのスケーラビリティを向上させることができます。削減できた時間を活用して、マルチポイント障害シナリオを解決し、自信を持ってコンポーネントが不要になる時を判断できるようになります。外部共有リソースの分析に関する規範的ガイダンスについては、「SEC03-BP07 パブリックおよびクロスアカウントアクセスの分析 (p. 47)」を参照してください。

一般的なアンチパターン:

- 継続的にモニタリングして、予定外の外部共有が生じたときに自動的にアラートを発動するプロセスがない。
- 共有すべき/すべきでない内容に関する基準がない。
- 必要な時点で明示的に共有するのではなく、広く開かれたポリシーをデフォルトとしている。
- 必要に応じて重複する基本的リソースを手動で作成する。

このベストプラクティスを確立しない場合のリスクレベル: 中

実装のガイダンス

アクセスコントロールとパターンを構築し、信頼できるエンティティとのみ共有リソースの消費を安全に管理します。共有リソースをモニタリングして、継続的に共有リソースアクセスをレビューし、不適切なまたは予想外の共有があればアラートを発動します。パブリックおよびクロスアカウントアクセスを分析するをレビューして、外部からのアクセスを必要なリソースのみに限定し、継続的にモニタリングし、自動的にアラートを出すプロセスを確立するためのガバナンスを確立するのに役立ちます。

AWS Organizations 内のクロスアカウント共有は、多数の AWS サービス (AWS Security Hub、Amazon GuardDuty、および AWS Backup) によってサポートされています。これらのサービスを使用すると、中央アカウントでデータを共有し、中央アカウントからアクセス可能、あるいは中央アカウントからリソースとデータを管理できます。例えば、AWS Security Hub は個別アカウントから中央アカウントに検出結果を送信するため、すべての検出結果を確認することができます。AWS Backup は、リソースのバックアップを取り、アカウント全体で共有します。AWS Resource Access Manager (AWS RAM) を使用して、VPC

サブネットおよび Transit Gateway 添付ファイル、AWS Network Firewall、または Amazon SageMaker パイプラインなど他の一般的なリソースを共有できます。

リソースを組織内のリソースのみと共有するよう制限するには、サービスコントロールポリシー (SCP) を使って、外部プリンシパルへのアクセスを防止します。リソースを共有する際、アイデンティティベースのコントロールとネットワークコントロールを組み合わせて、組織のデータ境界を作成すると、意図しないアクセスから保護するのに役立ちます。データ境界とは、信頼できるアイデンティティのみが、期待されるネットワークから信頼できるリソースにアクセスするよう徹底するのに役立つ予防的な一連のガードレールです。これらのコントロールは、どのリソースが共有可能かについて適切な制限を設け、共有や公開が許可されるべきでないリソースについてはそれを禁止する必要があります。例えば、データ境界の一部として、VPC エンドポイントポリシーと `AWS:PrincipalOrgId` 条件を使用することで、Amazon S3 バケットにアクセスするアイデンティティが組織に確実に属するようにできます。SCP は、サービスにリンクされたロール (LSR) または AWS サービスプリンシパルに適用されないことに注意してください。

Amazon S3 を使用する際、 は Amazon S3 バケットに対して ACL を無効化し、IAM ポリシーをし応してアクセスコントロールを定義します。Amazon CloudFront から Amazon S3 オリジンにアクセスされることを制限するには、オリジンアクセスアイデンティティ (OAI) からオリジンアクセスコントロール (OAC) に移行します。これは、AWS Key Management Service のサーバー側暗号化を含む追加機能をサポートします。

場合によっては、組織外のリソースを共有したり、リソースにサードパーティーのアクセスを付与したりするかもしれません。リソースを外部で共有する権限管理に関する規定的なガイダンスについては、「Permissions management」を参照してください。

実装手順

1. AWS Organizations を使用します。

   AWS Organizations は、組織を作成して一元管理するときに、複数の AWS アカウント を統合できるアカウント管理サービスです。アカウントを組織単位 (OU) にグループ化し、OU ごとに異なるポリシーをアタッチすることにより、予算、セキュリティ、コンプライアンスのニーズに対応できます。また、AWS 人工知能 (AI) と機械学習 (ML) サービスがどのようにデータを収集して保管するかをコントロールし、Organizations と統合された AWS サービスのマルチアカウント管理を使用できます。
2. AWS Organizations を AWS サービスと統合します。

   組織のメンバーアカウントを代理してタスクを実行する AWS サービスを有効にすると、AWS Organizations が各メンバーアカウントでそのサービスに対して IAM サービスがリンクされたロールを作成します。AWS Management Console、AWS API、または AWS CLI を使用して、信頼できるアクセスを管理する必要があります。信頼されたアクセスを可能にするための規範的ガイダンスについては、「AWS Organizations を他の AWS サービスおよび Organizations と併用できる AWS サービスの使用」を参照してください。
3. データ境界を確立します。

   AWS 境界は、AWS Organizations によって管理される組織として表現されるのが普通です。オンプレミスネットワークとシステムとともに、AWS リソースへのアクセスは、My AWS の境界としてみなしているものです。この境界の目標は、アイデンティティが信頼され、リソースが信頼され、そしてネットワークが予想されている場合にそのアクセスが許可されていることを検証することにあります。

   a. 境界を定義および実装します。

      各認証条件について、AWS ホワイトペーパーの「境界の構築」の境界の実装に記載されたステップに従います。ネットワーク層に関する規範的ガイダンスについては、「ネットワークの保護」を参照してください。

   b. 継続的にモニタリングとアラートを行います。

      AWS Identity and Access Management Access Analyzer は、組織内のリソースや、外部エンティティと共有しているアカウントを特定するのに役立ちます。IAM Access Analyzer を AWS Security Hub と統合して、IAM Access Analyzer から Security Hub へリソースの検出結果を送信および集計し、環境のセキュリティ体制を分析するのに役立てます。統合を有効にするには、各アカウントの各

リージョンでIAM Access Analyzer と Security Hub の両方を有効にします。また、AWS Config ルール を使用して、設定を監査し、AWS Chatbot と AWS Security Hub を併用する適切な当事者にアラートを出すことができます。次に、AWS Systems Manager の自動化ドキュメントを使用して、非準拠のリソースを修復できます。

c. 外部で共有するリソースを継続的にモニタリングおよびアラート通知するための規範的ガイダンスについては、「パブリックおよびクロスアカウントアクセスを分析する」を参照してください。

4. AWS サービスのリソース共有を使用し、それに従って制限します。

   多くの AWS サービスを使用すると、別のアカウントとリソースを共有したり、別アカウントにある Amazon マシンイメージ (AMI) および AWS Resource Access Manager (AWS RAM) などのリソースをターゲットとすることができます。`ModifyImageAttribute` API を制限して、AMI を共有する信頼できるアカウントを指定します。AWS RAM を使用している場合、`ram:RequestedAllowsExternalPrincipals` 条件を指定し、共有を自組織にのみ限定して、信頼されていないアイデンティティからのアクセスを防止します。規範的ガイダンスおよび注意事項については、「リソース共有および外部ターゲット」を参照してください。

5. AWS RAM を使用して、自分のアカウント内または他の AWS アカウント と安全に共有します。

   AWS RAM は、自分のアカウントおよび他の AWS アカウント アカウントのロールやユーザーで作成したリソースを安全に共有するのに役立ちます。マルチアカウント環境の場合、AWS RAM ではリソースを作成したら、それを他のアカウントと共有できます。このアプローチにより、運用諸経費を削減し、Amazon CloudWatch および AWS CloudTrail との統合を通じて、一貫性、可視性、監査可能性を提供することができます。これは、クロスアカウントアクセスを使用している場合は享受できません。

   リソースベースポリシーを使って過去に共有したリソースがある場合、`PromoteResourceShareCreatedFromPolicy` API または同等のコマンドを使って、リソース共有を完全な AWS RAM リソース共有に昇格させることができます。

   場合によっては、リソースを共有するための追加ステップが必要かもしれません。たとえば、暗号化されたスナップショットを共有するには、AWS KMS キーを共有する必要があります。

リソース

関連するベストプラクティス:

- SEC03-BP07 パブリックおよびクロスアカウントアクセスの分析 (p. 47)
- SEC03-BP09 サードパーティーとリソースを安全に共有する (p. 52)
- SEC05-BP01 ネットワークレイヤーを作成する (p. 65)

関連するドキュメント:

- バケット所有者が所有権のないオブジェクトへのクロスアカウントアクセス許可を付与する
- How to use Trust Policies with IAM (IAM ロールと信頼ポリシーを使用する方法)
- Building Data Perimeter on AWS (AWS でのデータ境界の構築)
- AWS リソースへのアクセス権を第三者に付与するときに外部 ID を使用する方法
- AWS Organizations と使用できる AWS サービス
- AWS でデータ境界を確立する: 信頼できるアイデンティティのみが会社データにアクセスできるようにします

関連動画:

- Granular Access with AWS Resource Access Manager (AWS Resource Access Manager を使用したきめ細かいアクセス)
- Securing your data perimeter with VPC endpoints (VPC エンドポイントを使用したデータ境界の保護)

- Establishing a data perimeter on AWS (AWS でのデータ境界の確立)

関連ツール:

- Data Perimeter Policy Examples (データ境界ポリシーの例)

SEC03-BP09 サードパーティーとリソースを安全に共有する

クラウド環境のセキュリティは、組織内にとどまりません。組織が、データの一部を管理するのにサードパーティーに依存することもあります。サードパーティー管理システムの権限管理は、一時的な認証情報を使用する最小特権の原則を用いたジャストインタイムアクセスの実践に従う必要があります。サードパーティーと密に連携することにより、意図しないアクセスの影響が及ぶ範囲とリスクをともに縮小することができます。

期待される成果: ユーザーと関連付けられた長期的AWS Identity and Access Management (IAM) 認証情報、IAM アクセスキー、およびシークレットキーは、認証情報が有効かつアクティブである限り、誰でも使用できます。IAM ロールと一時的な認証情報を使うと、そういった機密性の高い詳細の管理と運用間接費など、長期的認証情報を維持するための業務を減らすことにより、総合的なセキュリティスタンスが改善されます。IAM 信頼ポリシーの外部 ID に対して汎用一意識別子 (UUID) を使用し、IAM ロールにアタッチされた IAM ポリシーを制御かに置くことにより、サードパーティーに付与されたアクセスを監査して、過度に寛容でないことを確認できます。外部共有リソースの分析に関する規範的ガイダンスについては、「SEC03-BP07 パブリックおよびクロスアカウントアクセスの分析 (p. 47)」を参照してください。

一般的なアンチパターン:

- 条件なしでデフォルトの IAM 信頼ポリシーを使用する。
- 長期的 IAM 認証情報とアクセスキーを使う。
- 外部 ID を再使用する。

このベストプラクティスを確立しない場合のリスクレベル: 中

実装のガイダンス

AWS Organizations 外のリソースを共有したり、アカウントにサードパーティーのアクセスを付与したりする場合があります。たとえば、サードパーティーが提供する監視ソリューションが、貴社のアカウント内のリソースにアクセスする必要があるかもしれません。そのような場合、サードパーティーにとって必要な権限のみを含む IAM クロスアカウントロールを作成します。さらに、外部 ID 条件を使って信頼ポリシーを定義します。外部 ID を使用すると、自分またはサードパーティーが各顧客、サードパーティー、またはテナンシーに対して一意の ID を生成できます。一意の ID を作成後は、自分以外の人物によってコントロールできなくなります。サードパーティーは、外部 ID を安全に、監査可能かつ再現可能な方法で顧客に関連付けるプロセスを実装する必要があります。

また、IAM Roles Anywhere を使用すると、AWS を用いる AWS 外のアプリケーションに対して IAM ロールを管理できます。

サードパーティーが貴社の環境にアクセスする必要がなくなった場合は、ロールを削除します。サードパーティーに長期的な認証情報を提供することは避けてください。共有をサポートする他の AWS サービスを継続的に把握しておきます。たとえば、AWS Well-Architected Tool では ワークロードを他の AWS アカウント と共有でき、AWS Resource Access Manager は所有する AWS リソースを安全に他のアカウントと共有するのに役立ちます。

実装手順

1. クロスアカウントロールを使って、外部アカウントへのアクセスを提供します。

クロスアカウントロールを使うと、顧客にサービスを提供するために外部アカウントとサードパーティーによって保存されている機密情報の量が減ります。クロスアカウントロールがあると、アクセスを管理および監査する能力を維持しながら、AWS Partner または組織内の他のアカウントなど、アカウントの AWS リソースへのアクセスをサードパーティーに安全に付与できます。

サードパーティーが、ハイブリッドインフラストラクチャからサービスを提供したり、またはオフサイトロケーションにデータをプルする場合があります。IAM Roles Anywhere を使用すると、サードパーティーのワークロードが AWS ワークロードと安全に対話し、長期的認証情報のニーズをさらに軽減することができます。

外部アカウントアクセスを提供するために、ユーザーと関連付けられた長期的認証情報、またはアクセスキーを使用しないでください。かわりに、クロスアカウントロールを使ってクロスアカウントアクセスを提供します。

2. サードパーティーに外部 ID を使用します。

   外部 ID を使用することにより、IAM 信頼ポリシーで誰がロールを担うかを指定できます。信頼ポリシーでは、ロールを引き受けるユーザーが、操作を行う条件とターゲットを実施する必要があります。またこの方法により、アカウントの所有者が特定の状況下でのみロールを引き受けることを許可する方法も提供できます。外部 ID の主な機能は、代理人の混乱問題に対処して防止することにあります。

   外部 ID は、AWS アカウント 所有者であり、自分のものに加えて他の AWS アカウント にアクセスするサードパーティーに対してロールを設定した場合、または異なる顧客に代わってロールを引き受けるポジションにある場合に使用します。サードパーティーまたは AWS Partner と連携して、IAM 信頼ポリシーに含める外部 ID 条件を確立します。

3. 汎用一意識別子を使用します。

   汎用一意識別子 (UUID) など、外部 ID に対してランダムな一意の値を生成するプロセスを実装します。サードパーティーが異なる顧客間で外部 ID を再使用しても、「代理人の混乱」問題に対処できません。これは、顧客 A が、顧客 B のロール ARN と重複した外部 ID を使用することで、顧客 B のデータを表示できる可能性があるためです。マルチテナント環境では、サードパーティーが異なる AWS アカウント で複数の顧客をサポートするため、サードパーティーが各 AWS アカウント に対する外部 ID として異なる一意の ID を使用する必要があります。サードパーティーは、重複した外部 ID を検出して、各顧客をそれぞれの外部 ID に安全にマッピングする責任があります。サードパーティーは、外部 ID を指定する際にのみロールを引き受けることができることをテストして検証する必要があります。サードパーティーは、外部 ID が必要となるまで、顧客ロール ARN と外部 ID を保存することを控える必要があります。

   外部 ID はシークレットとして取り扱われませんが、外部 ID は電話番号、氏名、アカウント ID など推測しやすい値であってはなりません。外部 ID を読み取り専用にすることで、設定のなりすましを目的として外部 ID が変更されないようにします。

   ご自身またはサードパーティーが外部 ID を生成できます。ID 生成に責任がある担当者を決定するプロセスを定義します。外部 ID を作成するエンティティにかかわらず、サードパーティーは顧客間で一貫した一意性とフォーマットを適用します。

4. 顧客が提供する長期的認証情報を廃止します。

   長期的認証情報の使用を廃止して、クロスアカウントロールまたは IAM Roles Anywhere を使用します。長期的認証情報を使用する必要がある場合、ロールベースのアクセスに移行する計画を立ててください。キーの管理に関する詳細については、「ID 管理」を参照してください。また、AWS アカウントチームおよびサードパーティーと連携して、リスク軽減ランブックを確立します。セキュリティインシデントの潜在的影響に対する反応と軽減の規範的ガイダンスについては、「インシデント対応」を参照してください。

5. セットアップに規範的ガイダンスがあるか、または自動化されていることを確認します。

   アカウントのクロスアカウントアクセス向けに作成されたポリシーは、least-privilege principle に従う必要があります。サードパーティーは、ロールポリシードキュメント、または AWS CloudFormation テンプレートまたは同等のものを使用する自動化されたセットアップメカニズムを提供する必要がありま

す。これにより、手動ポリシー作成に関連してエラーが発生する可能性が減り、監査可能証跡を提供します。AWS CloudFormation テンプレートを使用したクロスアカウントロール作成の詳細については、「クロスアカウントロール」を参照してください。

サードパーティーは、自動化され、監査可能なセットアップメカニズムを提供する必要があります。ただし、必要なアクセスを概説したロールポリシードキュメントを使用することにより、ロールのセットアップを自動化する必要があります。AWS CloudFormation テンプレートまたは同等のものを使用して、監査業務の一環として、ドリフト検出で変化をモニタリングする必要があります。

6. 変更するアカウント。

アカウント構成、サードパーティーのニーズ、または提供されるサービスオファリングは変わる可能性があります。変更と障害を予想し、それに応じて適切な人材、プロセス、テクノロジーを計画する必要があります。定期的に提供するアクセスのレベルを監査し、予想外の変更があった場合にアラートを出す検出方法を実装します。外部 ID のロールとデータストアをモニタリングおよび監査します。予想外の変更やアクセスパターンの結果、サードパーティーのアクセスを一時的または恒久的に取り消す準備をしておく必要があります。また、実行にかかる時間、関与する人材、コスト、および他のリソースの影響など、取り消し操作の影響を測定します。

検出方法に関する規範的ガイダンスについては、「検出のベストプラクティス」を参照してください。

リソース

関連するベストプラクティス:

- SEC02-BP02 一時的な認証情報を使用する (p. 22)
- SEC03-BP05 組織のアクセス許可ガードレールを定義する (p. 45)
- SEC03-BP06 ライフサイクルに基づいてアクセスを管理する (p. 46)
- SEC03-BP07 パブリックおよびクロスアカウントアクセスの分析 (p. 47)
- SEC04 の検出

関連するドキュメント:

- バケット所有者が所有権のないオブジェクトへのクロスアカウントアクセス許可を付与する
- How to use Trust Policies with IAM (IAM ロールと信頼ポリシーを使用する方法)
- AWS アカウント 間の IAM ロールを使用したアクセスの委任
- IAM を使用して別の AWS アカウントのリソースにアクセスするにはどうすればよいですか?
- IAM でのセキュリティのベストプラクティス
- クロスアカウントポリシーの評価論理
- AWS リソースへのアクセス権を第三者に付与するときに外部 ID を使用する方法
- Collecting Information from AWS CloudFormation Resources Created in External Accounts with Custom Resources (カスタムリソースで外部アカウントに作成された AWS CloudFormation のリソースから情報を収集する)
- Securely Using External ID for Accessing AWS Accounts Owned by Others (他人が所有する AWS アカウントへのアクセスに外部 ID を安全に使用する)
- Extend IAM roles to workloads outside of IAM with IAM Roles Anywhere (IAM Roles Anywhere で AWS IAM ロールを AWS 外のワークロードに拡張する)

関連動画:

- How do I allow users or roles in a separate AWS アカウント access to my AWS アカウント? (別の AWS アカウントのユーザーやロールに、私の AWS アカウントへのアクセスを許可するにはどうすればよいですか?)

- AWSre:Invent 2018: Become an IAM Policy Master in 60 Minutes or Less (60 分以内に IAM ポリシーマスターになる)
- AWS Knowledge Center Live: IAM Best Practices and Design Decisions (ベストプラクティスと設計の決定)

関連する例:

- Well-Architected Lab - Lambda cross account IAM role assumption (Lambda クロスアカウント IAM ロール引き受け) (レベル 300)
- Configure cross-account access to Amazon DynamoDB (Amazon DynamoDB へのクロスアカウントアクセスを設定する)
- AWS STS Network Query Tool (AWS STS ネットワーククエリツール)

検知

検知は、予期しないか望ましくない設定の変更の検知と、不要な動作の検知という 2 つの部分に分かれています。1 番目は、アプリケーション配信ライフサイクルの複数個所で発生する可能性があります。Infrastructure as Code (例えば CloudFormation テンプレート) を使用し、CI/CD パイプラインやソース管理でチェックを実施することにより、ワークロードをデプロイする前に不要な設定をチェックできます。次に、ワークロードを非本番環境と本番環境にデプロイする際、ネイティブの AWS、オープンソース、または AWS Partner ツールを使って設定をチェックできます。これらのチェックは、セキュリティプリンシパルまたはベストプラクティスに合致しない設定や、テストされた設定とデプロイされた設定間で行われた変更に対して実行できます。実行中のアプリケーションの場合、既知のデプロイ以外や自動化されたスケーリングイベントなど、設定が予期しないやり方で変更されたかどうかをチェックできます。

2 番目の予期しない動作の検知については、ツールを使うか、または特定のタイプの API コールの増加のアラートを送信することができます。Amazon GuardDuty を使用すると、予期しない、さらに潜在的に不正または悪意のあるアクティビティが AWS アカウントで発生した場合、アラートを受け取ることができます。また、ワークロードでの使用が予想されていない変異型 API コールや、セキュリティ体制を変更する API コールを明示的にモニタリングする必要があります。

検出により、潜在的なセキュリティ設定の誤り、脅威、予期しない動作を特定できます。検出はセキュリティライフサイクルの最重要部分であり、品質管理プロセス、法的義務またはコンプライアンス義務、脅威の特定とその対応をサポートします。検出メカニズムにはさまざまなタイプがあります。たとえば、ワークロードのログは、使用された脆弱性を分析できます。ワークロードに関連する検出メカニズムを定期的に見直し、内部および外部のポリシーと要件を満たしていることを確認する必要があります。自動化されたアラートと通知は、チームやツールが調査できるように、定義された条件に基づいて行う必要があります。これらのメカニズムは、組織が異常なアクティビティの範囲を特定し把握するのに役立つ重要な対応機能です。

AWS には、検出メカニズムに対処する際に使用できる多くのアプローチがあります。以下の各セクションでは、こうしたアプローチの使用方法を説明します。

ベストプラクティス

- SEC04-BP01 サービスとアプリケーションのログ記録を設定する (p. 56)
- SEC04-BP02 ログ、結果、メトリクスを一元的に分析する (p. 59)
- SEC04-BP03 イベントへの応答を自動化する (p. 61)
- SEC04-BP04 実用的なセキュリティイベントを実装する (p. 62)

SEC04-BP01 サービスとアプリケーションのログ記録を設定する

サービスとアプリケーションからセキュリティイベントログを保持します。これは、監査、調査、運用のユースケースにおけるセキュリティの基本原則であり、ガバナンス、リスク、コンプライアンス (GRC) の標準、ポリシー、手順によって推進される共通のセキュリティ要件です。

期待される成果: 組織は、セキュリティインシデント対応など、内部プロセスまたは義務を遂行する必要がある場合、AWS サービスおよびアプリケーションからのセキュリティイベントログを、確実かつ一貫して迅速に取得できるようにする必要があります。運用側の成果を改善するために ログの一元化を検討してください。

一般的なアンチパターン:

- ログが永久に保存される、またはすぐに削除される。
- 誰でもログにアクセスできる。

- ログガバナンスと使用について、手動プロセスのみに依存する。
- 必要な場合に備えて、あらゆるタイプのログを保存する。
- 必要な場合にのみログ整合性をチェックする。

このベストプラクティスを活用するメリット: セキュリティインシデントの根本原因分析 (RCA) メカニズムを導入し、ガバナンス、リスク、コンプライアンス義務のための証拠資料とします。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

セキュリティ調査または要件に基づいた他のユースケース中、インシデントの全容とタイムラインを記録して理解するために関連ログをレビューできる必要があります。ログはまた、関心のある特定のアクションが発生したことを示すアラート生成にも必須です。クエリと取得メカニズムとアラートを選択、有効化、保存、設定することが非常に重要となります。

実装手順

- ログのソースを選択して有効にします。 セキュリティ調査の前に、関連するログを取得し、過去にさかのぼって AWS アカウント でアクティビティを再構築する必要があります。ワークロードに関連するログソースを選択して、有効化します。

  ログソース選択条件は、ビジネスで必要なユースケースに基づいたものである必要があります。AWS CloudTrail または AWS Organizations 証跡を使って各 AWS アカウント に証跡を確立し、そのための Amazon S3 バケットを設定します。

  AWS CloudTrail は、AWS のサービスアクティビティをキャプチャする AWS アカウント に対して API コールをトラッキングするログサービスです。これは、デフォルトで有効になっており、管理イベントは 90 日間保持され、AWS Management Console、AWS CLI、または AWS を使用して CloudTrail イベント履歴から検索することが可能です。データイベントをより長く保持および確認するには、CloudTrail 証跡を作成して、Amazon S3 バケットと、そしてオプションで Amazon CloudWatch ロググループと関連付けます。または、CloudTrail Lake を作成できます。これは、CloudTrail ログを最長 7 年間保持し、SQL ベースのクエリ施設を提供します。

  AWS は、VPC を使用している顧客は、VPC フローログとAmazon Route 53 リゾルバーのクエリログをそれぞれ使用して、ネットワークトラフィックと DNS ログを有効化し、それらを Amazon S3 バケットまたは CloudWatch ロググループにストリーミングすることを推奨しています。VPC、サブネット、またはネットワークインターフェイス向けに VPC フローログを作成できます。VPC フローログについては、コストを削減するためにどこでどのようにフローログを使用するかを選択できます。

  AWS CloudTrail ログ、VPC フローログ、および Route 53 リゾルバーのクエリログは、AWS でセキュリティ調査をサポートするための基本的なログ記録ソースです。また、Amazon Security Lake を使用して、このログデータを収集、正規化、そして Apache Parquet フォーマットと Open Cybersecurity Schema Framework (OCSF) に保存することもできます。Security Lake は、他の AWS ログ、およびサードパーティーソースからのログもサポートします。

  AWS のサービスは、Elastic Load Balancing ログ、AWS WAF ログ、AWS Config レコーダーログ、Amazon GuardDuty 検出結果、Amazon Elastic Kubernetes Service (Amazon EKS) 監査ログ、および Amazon EC2 インスタンスのオペレーティングシステムおよびアプリケーションログなど、基本ログソースによってキャプチャされないログを生成できます。ログ記録とモニタリングオプションの詳しいリストについては、『AWS セキュリティインシデント対応ガイド』の「付録 A: クラウド機能の定義 – ログとイベント」を参照してください。
- 各 AWS サービスとアプリケーションの調査ログ機能: 各 AWS サービスとアプリケーションは、ログストレージのオプションを提供し、それぞれが独自の保持とライフサイクル機能が備わっています。2 つの最も良く使用するログストレージサービスは、Amazon Simple Storage Service (Amazon S3) と Amazon CloudWatch です。保持期間が長い場合、費用対効果と柔軟なライフサイクル機能のために Amazon S3 を使用することが推奨されます。プライマリログ記録オプションが Amazon CloudWatch ロ

グの場合、オプションとして、アクセス頻度の低いログを Amazon S3 にアーカイブすることを検討する必要があります。

- ログストレージを選択する: ログストレージの選択肢は、通常、使用するクエリツール、保持機能、精通度、そしてコストに関連しています。ログストレージの主なオプションは、Amazon S3 バケットまたは CloudWatch ロググループです。

  Amazon S3 バケットは、ライフサイクルポリシーがオプションで備わっている、費用対効果に優れ、耐久性の高いストレージを提供します。Amazon S3 バケットに保存されているログは、Amazon Athena などのサービスを使ってクエリすることができます。

  CloudWatch ロググループは、CloudWatch Logs Insights により、耐久性の高いストレージとビルトインクエリ施設を提供します。

- 適切なログ保持を識別する: Amazon S3 バケットまたは CloudWatch ロググループを使ってログを保存する場合、各ログソースに対して適切なライフサイクルを確立して、ストレージと取得コストを最適化する必要があります。顧客のログは通常 3 ヶ月 ~ 1 年間で、すぐにクエリでき、最長 7 年間保持されます。可用性と保持の選択は、セキュリティ要件と、法令、規制、およびビジネス上の義務の組み合わせに合わせるべきです。
- 適切な保持とライフサイクルポリシーを使って各 AWS サービスとアプリケーションのログを有効にする: 組織の各 AWS サービスまたはアプリケーションについて、特定のログ設定ガイダンスを確認してください。
  - AWS CloudTrail 証跡を設定する
  - VPC フローログを設定する
  - Amazon GuardDuty 検出結果エクスポートを設定する
  - AWS Config 記録を設定する
  - AWS WAF Web ACL トラフィックを設定する
  - AWS Network Firewall ネットワークトラフィックログを設定する
  - Elastic Load Balancing アクセスログを設定する
  - Amazon Route 53 リゾルバーのクエリログを設定する
  - Amazon RDS ログを設定する
  - Amazon EKS コントロールプレーンログを設定する
  - Amazon EC2 インスタンスとオンプレミスサーバーに対して Amazon CloudWatch エージェントを設定する
- ログのクエリメカニズムを選択して実装する: ログのクエリについては、CloudWatch ログ分析情報 を CloudWatch ロググループに保存されたデータに、Amazon Athena と Amazon OpenSearch Service を Amazon S3 に保存されたデータに使用できます。また、セキュリティ情報とイベント管理 (SIEM) サービスなど、サードパーティーのクエリツールを使用することもできます。

  ログクエリツールを選択するためのプロセスは、セキュリティオペレーションの人材、プロセス、およびテクノロジー側面を考慮する必要があります。オペレーション、ビジネス、セキュリティの要件を満たし、長期的にアクセスとメンテナンスが可能なツールを選択します。ログクエリツールは、スキャンするログの数がツールの制限内に収まっている場合、動作が最適であることに注意してください。コストや技術的な制約から、複数のクエリツールを所有することも珍しくありません。

  たとえば、過去 90 日間のデータにはサードパーティーのセキュリティ情報とイベント管理 (SIEM) ツールを使用しながらも、SIEM のログインジェストコストが原因で 90 日以前のデータをクエリする際は Athena を使用するとった場合です。どのような実装であっても、必要なツールの数を最小限に抑えることで、特にセキュリティイベントの調査時に、運用効率が最大となるアプローチであることを確認してください。
- アラートにログを使用する: AWS は、複数のセキュリティサービスを使ってアラートを提供しています:
  - AWS Config では、AWS リソース構成のモニタリングと記録が行われ、目標の構成に対する評価と修復が自動化できます。
  - Amazon GuardDuty は脅威検出サービスです。悪意のある動作や不正な動作を継続的にモニタリングし、AWS アカウント とワークロードを保護できるようにします。GuardDuty は、AWS CloudTrail

管理およびデータイベント、DNS ログ、VPC フローログ、および Amazon EKS 監査ログなどのソースからの情報をインジェスト、集計、および分析します。GuardDuty は、CloudTrail、VPC フローログ、DNS クエリログ、および Amazon EKS から直接独立したデータストリームをプルします。Amazon S3 バケットポリシーを管理したり、ログを収集および保存する方法を変更したりする必要はありません。独自の調査やコンプライアンス目的で、これらのログを保持することは引き続き推奨されています。

- AWS Security Hub では、複数の AWS のサービスや任意のサードパーティー製品からのセキュリティアラートまたは検出結果の集約、整理、優先順位付けが一元的に行われ、セキュリティアラートとコンプライアンスステータスを包括的に把握できます。

また、これらのサービスの対象外となるセキュリティアラートや、自分の環境に関連する特定なアラートについては、カスタムアラート生成エンジンを使用することもできます。これらのアラートや指示を構築するたの情報については、『AWS セキュリティインシデント対応ガイド』の「検出」を参照してください。

リソース

関連するベストプラクティス:

- SEC04-BP02 ログ、結果、メトリクスを一元的に分析する (p. 59)
- SEC07-BP04 データのライフサイクル管理を定義する (p. 84)
- SEC10-BP06 ツールを事前デプロイする (p. 112)

関連するドキュメント:

- AWS Security Incident Response Guide (AWS セキュリティインシデント対応ガイド)
- Amazon Security Lake の開始方法
- 開始方法: Amazon CloudWatch Logs
- Security Partner Solutions: Logging and Monitoring (セキュリティパートナーのソリューション: ログ記録とモニタリング)

関連動画:

- AWS re:Invent 2022 - Introducing Amazon Security Lake (Amazon Security Lake の紹介)

関連する例:

- Assisted Log Enabler for AWS (AWS 向けの Assisted Log Enabler)
- AWS Security Hub Findings Historical Export (AWS セキュリティハブ検出結果履歴レポート)

関連ツール:

- Snowflake for Cybersecurity

SEC04-BP02 ログ、結果、メトリクスを一元的に分析する

セキュリティ運用チームは、ログを収集し、検索ツールを使用することによって、不正なアクティビティや意図しない変更の可能性がある、潜在的に関心のあるイベントを発見します。ただし、収集されたデー

タを分析して手動で情報を処理するだけでは、複雑なアーキテクチャから流れる大量の情報に対応するには不十分です。分析とレポートだけでは、適切なリソースを割り当てて、イベントをタイミング良く実行する作業が容易になる訳ではありません。

熟練したセキュリティオペレーションチームを構築するには、セキュリティイベントと調査結果の流れを、チケットシステム、バグまたは問題システム、その他のセキュリティ情報とイベント管理 (SIEM) システムなどの、通知およびワークフローシステムに深く統合することをお勧めします。これにより、メールや静的レポートからワークフローが排除され、イベントや調査結果のルーティング、エスカレート、管理が可能になります。多くの組織はセキュリティアラートをチャットまたはコラボレーションや開発者の生産性プラットフォームに統合しています。自動化に着手している組織は、API 主導の、低レイテンシーのチケット発行システムによって、「何を最初に自動化するか」を計画する際にかなりの柔軟性が得られます。

このベストプラクティスは、ユーザーアクティビティやネットワークイベントを示すログメッセージから生成されたセキュリティイベントだけでなく、インフラストラクチャ自体で検出された変更から生成されたセキュリティイベントにも適用できます。変更による悪影響が小さく、AWS Identity and Access Management (IAM) と AWS Organizations の設定の組み合わせではその実行を阻止できないような状況では、変更を検出し、変更が適切かどうかを判断し、その情報を正しい修復ワークフローにルーティングする機能が、安全なアーキテクチャを維持、検証するうえで不可欠です。

Amazon GuardDuty と AWS Security Hub は、他の AWS のサービスでも利用できるログレコードの集約、重複排除、分析メカニズムを提供します。GuardDuty は、AWS CloudTrail 管理やデータイベント、VPC DNS ログ、および VPC Flow Logs などのソースからの情報を取込み、集計し、分析します。Security Hub は、GuardDuty、AWS Config、Amazon Inspector、Amazon Macie、AWS Firewall Manager、および AWS Marketplace で利用できるかなりの数のサードパーティーセキュリティ製品、そして適切にビルドした場合は独自のコードからの出力を取込み、集計、分析できます。GuardDuty と Security Hub のどちらにも、複数のアカウントにわたって調査結果とインサイトを集約できるマスターメンバーモデルがあります。Security Hub は、オンプレミスの SIEM を導入しているお客様に AWS 側のログ/アラートのプリプロセッサ/アグリゲータとしてよく使用され、お客様はそこから AWS Lambda ベースのプロセッサとフォワーダーを介して Amazon EventBridge を取り込むことができます。

このベストプラクティスを活用しない場合のリスクレベル: 高

実装のガイダンス

- ログ処理機能を評価する: ログの処理に使用できるオプションを評価します。
  - Amazon OpenSearch Service を使用して (ほぼ) あらゆる対象をログ記録およびモニタリングする
  - ログ記録および分析ソリューションを専門とするパートナーを探す
- CloudTrail ログの分析の最初の作業として Amazon Athena をテストする
  - CloudTrail ログを分析するように Athena を設定する
- AWS での集中ロギングを実装する: 複数のソースからのログ記録を一元化する次の AWS のサンプルソリューションを参照してください。
  - 集中ロギングソリューション
- パートナーで集中ロギングを実装する: APN パートナーは、ログを一元的に分析するためのソリューションを提供しています。
  - ログ記録とモニタリング

リソース

関連するドキュメント:

- AWS の回答: 集中ログ記録
- AWS Security Hub
- Amazon CloudWatch

- Amazon EventBridge
- 開始方法: Amazon CloudWatch Logs
- セキュリティパートナーのソリューション: ログ記録とモニタリング

関連動画:

- リソースの設定とコンプライアンスを一元的にモニタリングする
- Amazon GuardDuty および AWS Security Hub の調査結果の修復
- クラウドにおける変更管理: Amazon GuardDuty および AWS Security Hub

SEC04-BP03 イベントへの応答を自動化する

自動化を使用してイベントを調査および修正することで、人為的な労力やエラーが軽減され、調査機能をスケールできます。定期的なレビューは、自動化ツールを調整するのに役立ちます。継続して繰り返し行います。

AWS では、Amazon EventBridge を使用して、関心のあるイベントと予期しない変更についての情報の調査を自動化されたワークフローに組み込むことができます。このサービスには、スケーラブルなルールエンジンが備わっており、ネイティブの AWS イベント形式 (AWS CloudTrail イベントなど)と、独自のアプリケーションから生成できるカスタムイベントの両方を仲介できます。Amazon GuardDuty では、インシデントレスポンスシステムを構築するワークフローシステム (AWS Step Functions) や、中央のセキュリティアカウントにイベントをルーティングできます。また、バケットにルーティングして詳細分析を実行することもできます。

変更を検出してこの情報を正しいワークフローにルーティングするには、AWS Config ルール と コンフォーマンスパックを使用できます。AWS Config は、(EventBridge より高いレイテンシーを通して) スコープ内サービスへの変更を検出し、AWS Config ルール ルールを使用して分析できるイベントを生成します。分析されたイベントは、ロールバックやコンプライアンスポリシーの適用のほか、変更管理プラットフォームや運用チケット発行システムなどのシステムに対する情報の転送に使用できます。AWS Config イベントに対応する独自の Lambda 関数を作成するだけでなく、 AWS Config ルール 開発キット および オープンソースライブラリの AWS Config ルール も利用できます。コンフォーマンスパックとは、YAML テンプレートとして作成される単一エンティティとしてデプロイする一連の AWS Config ルール および修正アクションです。A サンプルコンフォーマンスパックテンプレートは Well-Architected セキュリティの柱で利用できます。

このベストプラクティスを活用しない場合のリスクレベル: ミディアム

実装のガイダンス

- GuardDuty で自動アラートを実装する: GuardDuty は、脅威検出サービスです。悪意のあるアクティビティや不正な動作を継続的にモニタリングし、AWS アカウント とワークロードを保護します。GuardDuty を有効にし、自動アラートを設定します。
- 調査プロセスを自動化する: 時間を節約するため、イベントを調査して情報を管理者に報告する自動化されたプロセスを開発します。
  - ラボ: Amazon GuardDuty ハンズオン

リソース

関連するドキュメント:

- AWS の回答: 集中ログ記録
- AWS Security Hub

- Amazon CloudWatch
- Amazon EventBridge
- 開始方法: Amazon CloudWatch Logs
- セキュリティパートナーのソリューション: ログ記録とモニタリング
- Amazon GuardDuty の設定

関連動画:

- リソースの設定とコンプライアンスを一元的にモニタリングする
- Amazon GuardDuty および AWS Security Hub の調査結果の修復
- クラウドにおける変更管理: Amazon GuardDuty および AWS Security Hub

関連する例:

- ラボ: 発見的統制の自動デプロイ

SEC04-BP04 実用的なセキュリティイベントを実装する

チームに送信され、チームによるアクションが可能なアラートを作成します。チームがアクションを実行するための関連情報がアラートに含まれていることを確認します。使用する検知メカニズムごとに、 ランブック または プレイブック形式の調査プロセスも用意する必要があります。例えば、 Amazon GuardDutyを有効にすると、 さまざまな調査結果が生成されます。。調査結果タイプごとにランブック エントリが必要です。例えば、 トロイの木馬 が検出された場合、調査して修復するよう指示する簡単な説明をランブックに記載する必要があります。

このベストプラクティスが確立されていない場合のリスクレベル: 低

実装のガイダンス

- AWS のサービスで利用可能なメトリクスを検出する: Amazon CloudWatch で利用可能な、利用中のサービスのメトリクスを確認することができます。
  - AWS のサービスドキュメント
  - Using Amazon CloudWatch Metrics
- Amazon CloudWatch アラームを設定します。
  - Amazon CloudWatch でのアラームの使用

リソース

関連するドキュメント:

- Amazon CloudWatch
- Amazon EventBridge
- Security Partner Solutions: Logging and Monitoring (セキュリティパートナーのソリューション: ログ記録とモニタリング)

関連動画:

- Centrally Monitoring Resource Configuration and Compliance (リソースの設定とコンプライアンスを一元的にモニタリングする)
- Remediating Amazon GuardDuty and AWS Security Hub Findings (Amazon GuardDuty および AWS Security Hub の調査結果の修復)
- Threat management in the cloud: Amazon GuardDuty and AWS Security Hub (クラウドにおける変更管理: Amazon GuardDuty および AWS Security Hub)

インフラストラクチャ保護

インフラストラクチャ保護には、ベストプラクティスと組織の義務または規制上の義務に準拠するために必要な、多層防御などの制御手法が含まれています。クラウドでの継続的な運用を成功させるには、このような手法を使用することが非常に重要です。

インフラストラクチャ保護は、情報セキュリティプログラムの重要な部分です。意図しない不正アクセスや潜在的な脆弱性から、ワークロード内のシステムとサービスを保護できます。たとえば、信頼境界 (ネットワークとアカウントの境界など)、システムセキュリティの設定とメンテナンス (強化、最小化、パッチ適用など)、オペレーティングシステムの認証と承認 (ユーザー、キー、アクセスレベルなど)、その他の適切なポリシー適用ポイント (ウェブアプリケーションファイアウォールや API ゲートウェイなど) を定義します。

リージョン、アベイラビリティーゾーン、AWS Local Zones、および AWS Outposts

リージョン、アベイラビリティーゾーン、AWS ローカルゾーン、および AWS Outpostsは AWS セキュアグローバルインフラストラクチャのコンポーネントですが、必ず精通しておいてください。

AWS にはリージョンという概念が存在します。これは、データセンターが集積されている世界中の物理的ロケーションのことです。論理的データセンターの各グループは、アベイラビリティーゾーン (AZ) と呼ばれます。各 AWS リージョンは、1 つの地理的領域内にある、複数の、隔離され、物理的にも分かれている AZ で成り立っています。データレジデンシー要件がある場合は、目的の場所の近くにある AWS リージョンを選択できます。データが物理的に配置されているリージョンに対する完全な制御と所有権を保持することで、地域のコンプライアンス要件やデータレジデンシー要件を満たすのに役立ちます。各 AZ には、独立した電源、冷却、および物理的セキュリティが備わっています。アプリケーションを AZ 間でパーティショニングすると、停電、落雷、竜巻、地震などの問題から、よりよく隔離され保護されます。各 AZ はそれぞれ他の AZ から物理的に意味のある距離、つまり数キロメートル離れていますが、互いにすべて 100 km (60 マイル) 以内に配置されています。AWS リージョン内のすべての AZ は、AZ 間に高スループットかつ低レイテンシーのネットワーキングを提供する、完全に冗長性を持つ専用メトロファイバー上に構築された、高帯域幅、低レイテンシーのネットワーキングで相互接続されています。AZ 間のすべてのトラフィックは暗号化されています。高度な可用性の実現にフォーカスしている AWS のお客様は、複数の AZ で実行するようにアプリケーションの設計をすることで、より強力な障害耐性を実現できます。AWS リージョンは、最高レベルのセキュリティ、コンプライアンス、データ保護を実現します。

AWS Local Zones では、コンピューティング、ストレージ、データベース、およびその他の選択された AWS のサービスをンドユーザーから近い場所に配置します。AWS Local Zones では、メディアエンターテインメントのコンテンツ制作、リアルタイムゲーミング、貯水池のシミュレーション、電子自動設計、機械学習など、エンドユーザーに対するレイテンシーが 10 ミリ秒未満であることが要求される高性能なアプリケーションを簡単に実行できます。各 AWS Local Zone ロケーションは AWS リージョンを拡張したものであり、Amazon EC2、Amazon VPC、Amazon EBS、Amazon File Storage、および Elastic Load Balancing などの AWS のサービスを使用して、地理的にエンドユーザーと近い場所で、レイテンシーの影響を受けやすいアプリケーションを実行できます。AWS Local Zones は、ローカルと AWS リージョンでそれぞれ実行中のワークロード間で高帯域幅かつ安全な接続が利用できます。同じ API とツールセットを介してすべてのリージョン内サービスにシームレスに接続します。

AWS Outposts により、ネイティブの AWS のサービス、インフラストラクチャ、運用モデルをほぼすべてのデータセンター、コロケーションスペース、オンプレミスの施設で利用できるようになります。同じ AWS の API、ツール、インフラストラクチャをオンプレミス全体と AWS クラウドで使用できるため、真に一貫したハイブリッドエクスペリエンスが提供されます。AWS Outposts はコネクテッド環境向けに設計されたものです。低レイテンシー、もしくはローカルでデータを処理する必要があるためにオンプレミスに残されているワークロードをサポートできます。

AWS では、いくつかの方法でインフラストラクチャを保護できます。以下の各セクションでは、こうしたアプローチの使用方法を説明します。

トピック

ネットワークの保護

従業員も顧客も、ユーザーはどこにでも存在する可能性があります。ネットワークにアクセスできる人なら誰でも、何でも信用するという従来のモデルから脱却する必要があります。すべてのレイヤーでセキュリティを適用するという原則に従えば、 ゼロトラスト アプローチを採用することになります。ゼロトラストセキュリティは、アプリケーションのコンポーネントやマイクロサービスを互いに分離して考え、どのコンポーネントやマイクロサービスも他のものを信用しないというモデルです。

ネットワーク設計を慎重に計画し管理することで、ワークロード内のリソースを分離し境界を作るための基礎が形成されます。ワークロードのリソースの多くは VPC 内で動作し、セキュリティのプロパティを継承するため、自動化によって支えられた検査および保護メカニズムで設計をサポートすることが重要です。同様に、純粋なエッジサービスやサーバーレスを使用して VPC の外部で動作するワークロードの場合は、よりシンプルなアプローチでベストプラクティスを適用します。ウェブアプリケーションのバックエンドに関する推奨事項については、 「AWS Well-Architected サーバーレスアプリケーションレンズ」でサーバーレスセキュリティに関する具体的なガイダンスを参照してください。

ベストプラクティス

SEC05-BP01 ネットワークレイヤーを作成する

機密度要件を共有するコンポーネントを階層化し、不正アクセスによる潜在的な影響範囲を最小化します。たとえば、インターネットアクセスを必要としない仮想プライベートクラウド (VPC) 内のデータベースクラスターは、インターネットへのルート、またはインターネットからのルートがないサブネットに配置する必要があります。トラフィックは、隣接する次に最も機密度が低いリソースからのみ流れる必要があります。ロードバランサーの背後にある Web アプリケーションを考慮します。ロードバランサーからデータベースに直接アクセスできてはいけません。データベースに直接アクセスすべきなのは、ビジネスロジックまたは Web サーバーのみです。

期待される成果: 階層型ネットワークを作成する。階層型ネットワークを使用すると、類似のネットワーキングコンポーネントを論理的にグループ化できます。また、不正ネットワークアクセスの影響の潜在的範囲が縮小されます。適切に階層化されたネットワークでは、不正なユーザーが AWS 環境内で追加リソースをピボットするのが困難になります。内部ネットワークパスをセキュリティ保護するだけでなく、ウェブアプリケーションや API エンドポイントなどのネットワークエッジも保護する必要があります。

一般的なアンチパターン:

- 単一 VPC またはサブネットですべてのリソースを作成する。
- 過度に寛容なセキュリティグループを使用する。
- サブネットを使用しない。
- データベースなどのデータストアに直接アクセスを許可する。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

共通の達成可能要件を持つ Amazon Elastic Compute Cloud (Amazon EC2) インスタンス、Amazon Relational Database Service (Amazon RDS) データベースクラスター、AWS Lambda 関数などのコンポーネントは、サブネットで形成されるレイヤーにセグメント化できます。Lambda 関数などのサーバーレスワークロードを、VPC 内または Amazon API Gateway の背後にデプロイすることを検討してください。インターネットアクセスが不要な AWS Fargate (Fargate) タスクは、ルートがインターネットとの経路がないサブネットに配置する必要があります。この階層的なアプローチは、意図しないアクセスを許可する可能性がある単一レイヤーの誤設定の影響を軽減します。AWS Lambda の場合は、VPC 内で関数を実行して、VPC ベースのコントロールを利用できます。

数千の VPC、AWS アカウント、オンプレミスネットワークを含むネットワーク接続の場合は、AWS Transit Gateway を使用する必要があります。Transit Gateway は、スポークのように機能するすべての接続されたネットワーク間でトラフィックがどのようにルーティングされるかを制御するハブとして機能します。Amazon Virtual Private Cloud (Amazon VPC) と Transit Gateway 間のトラフィックは AWS プライベートネットワークに留まりますが、これにより不正ユーザーへの外的露出やセキュリティ上の問題を軽減することができます。Transit Gateway のリージョン間ピアリングはまた、リージョン間トラフィックを単一障害点や帯域幅のボトルネックなしで暗号化します。

実装手順

- Reachability Analyzer を使用して、設定に基づく送信元と送信先間のパスを分析する: Reachability Analyzer では、VPC に接続されたリソースに対する接続の検証を自動化できます。この分析は設定を確認することで行われます (分析を行う際にネットワークパケットは送信されません)。
- Amazon VPC Network Access Analyzer を使用して、リソースへの意図しないネットワークアクセスを特定する: Amazon VPC Network Access Analyzer を使用すると、ネットワークアクセス要件を指定して、潜在的なネットワークパスを特定できます。
- リソースがパブリックサブネットにあるべきかどうか考慮する: VPC のパブリックサブネットには、パブリックソースからのインバウンドネットワークトラフィックを絶対に受信しなければならない場合を除き、リソースを配置しないでください。
- VPC に サブネットを作成する: (複数のアベイラビリティーゾーンを含むグループで) 各ネットワークレイヤーのサブネットを作成し、マイクロセグメンテーションを強化します。正しい ルートテーブルをサブネットと関連付けて、ルーティングとインターネット接続を制御できていることも確認します。
- AWS Firewall Manager を使用して、VPC セキュリティグループを管理する: AWS Firewall Manager は、複数のセキュリティグループを使用する管理上の負担を軽減します。
- AWS WAF を使って一般的な Web 脆弱性を保護する: AWS WAF は、SQL インジェクションなど一般的な Web 脆弱性がトラフィックにないか点検することにより、エッジセキュリティを強化するのに役立ちます。また、特定の国や地域から発信される IP アドレスからのトラフィックを制限することもできます。
- Amazon CloudFront をコンテンツ配信ネットワーク (CDN) として使用する: Amazon CloudFront は、データをユーザーの近くに保管することにより、Web アプリケーションのスピードを向上させるのに役立ちます。また、HTTPS の適用、地域へのアクセス制限、ネットワークトラフィックが CloudFront を経由した場合にのみリソースへのアクセスを許可することで、エッジセキュリティを改善することもできます。
- アプリケーションプログラミングインターフェイス作成時に Amazon API Gateway を使用する (API): Amazon API Gateway は、セキュアな REST、HTTPS、および WebSocket APIs を公開、モニタリング、およびセキュリティ保護するのに役立ちます。

リソース

関連するドキュメント:

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC セキュリティ
- Reachability Analyzer
- Amazon VPC Network Access Analyzer

関連動画:

- AWS Transit Gateway reference architectures for many VPCs (多くの VPC 用の AWS Transit Gateway リファレンスアーキテクチャ)
- Application Acceleration and Protection with Amazon CloudFront, AWS WAF, and AWS Shield (Amazon CloudFront、AWS WAF、AWS Shield によるアプリケーション高速化と保護)
- AWS re:Inforce 2022 - Validate effective network access controls on AWS(AWS で効果的なネットワークアクセスコントロールを検証する)
- AWS re:Inforce 2022 - Advanced protections against bots using AWS WAF (AWS WAF を使用したボットからの高度な保護)

関連する例:

- Well-Architected ラボ - Automated Deployment of VPC (VPC の自動デプロイ)
- ワークショップ: Amazon VPC Network Access Analyzer

SEC05-BP02 すべてのレイヤーでトラフィックを制御する

ネットワークトポロジを設計する際には、各コンポーネントの接続要件を調べる必要があります。たとえば、コンポーネントがインターネットアクセス (インバウンドおよびアウトバウンド) や、VPC、エッジサービス、外部データセンターへの接続を必要とする場合です。

VPC では、設定したプライベート IPv4 アドレス範囲または AWS によって選択された IPv6 アドレス範囲を使用して、AWS リージョン にまたがるネットワークトポロジを定義できます。インバウンドトラフィックとアウトバウンドトラフィックの両方に、多層防御アプローチを用いた複数のコントロールを適用する必要があります。これには、セキュリティグループ (ステートフルインスペクションファイアウォール)、ネットワーク ACL、サブネット、ルートテーブルの使用などが含まれます。VPC 内では、アベイラビリティーゾーンにサブネットを作成できます。各サブネットには、トラフィックがサブネット内でたどるパスを管理するためのルーティングルールを定義するルートテーブルを関連付けることができます。インターネットまたは VPC にアタッチされた NAT あるいは他の VPC ゲートウェイを経由するルートを設定することで、インターネットルーティングが可能なサブネットを定義できます。

インスタンス、Amazon Relational Database Service (Amazon RDS) データベース、またはその他のサービスが VPC 内で起動されると、ネットワークインターフェイスごとに独自のセキュリティグループが設定されます。このファイアウォールはオペレーティングシステムレイヤーの外側にあり、許可されるインバウンドトラフィックとアウトバウンドトラフィックのルールを定義するために使用できます。また、セキュリティグループ間の関係も定義できます。たとえば、データベース層のセキュリティグループ内のインスタンスは、関連するインスタンスに適用されるセキュリティグループを参照して、アプリケーション層のインスタンスからのトラフィックのみを受け入れます。TCP 以外のプロトコルを使用している場合を除き、ロードバランサーや CloudFront なしでインターネットから Amazon Elastic Compute Cloud (Amazon EC2) インスタンスに直接アクセスできるようにする必要はありません (セキュリティグループによって制限されているポートでも)。 CloudFront.これにより、オペレーティングシステムやアプリケーションの問題による意図しないアクセスから保護できます。サブネットには、ステートレスファイアウォールとして機能する、サブネットにアタッチされたネットワーク ACL を設定することもできます。

レイヤー間で許可されるトラフィックの範囲を絞り込むようにネットワーク ACL を設定する必要があります。インバウンドルールとアウトバウンドルールの両方を定義する必要があることに注意してください。

一部の AWS サービスは、インターネットにアクセスして API 呼び出しをする (AWS API エンドポイント がある) ためのコンポーネントが必要です。その他の AWS サービスは VPC エンドポイント を Amazon VPC 内で使用します。Amazon S3 や Amazon DynamoDB を含む多くの AWS サービスは VPC エンドポイントをサポートしており、このテクノロジーは次で一般化されています。 AWS PrivateLink.AWS のサービス、サードパーティーのサービス、および他の VPC セキュリティでホストされる独自のサービスにアクセスするには、このアプローチを使用することが推奨されます。AWS PrivateLink のすべてのネットワークトラフィックは、グローバルな AWS バックボーンにとどまり、インターネットにトラバースすることはありません。接続を開始できるのは、サービスのプロバイダーではなくサービスのコンシューマーのみです。外部サービスアクセスに AWS PrivateLink を使用することにより、インターネットなしでエアギャップ VPC を作成することができるため、外部の脅威ベクトルから VPC を保護するのに役立ちます。サードパーティーのサービスは AWS PrivateLink を使用して、プライベート PI アドレス経由で顧客が VPC からサービスに接続できるようにします。インターネットへのアウトバウンド接続を必要とする VPC アセットでは、これらは、AWS が管理する NAT ゲートウェイ、アウトバウンド専用のインターネットゲートウェイ、ユーザーが作成して管理するウェブプロキシを経由するアウトバウンド (一方向) でのみ可能です。

このベストプラクティスを活用しない場合のリスクレベル: 高

実装のガイダンス

- VPC 内のネットワークトラフィックを制御する: VPC ベストプラクティスを実装してトラフィックを制御する
  - Amazon VPC セキュリティ
  - VPC エンドポイント
  - Amazon VPC セキュリティグループ
  - ネットワーク ACL
- エッジでのトラフィックを制御する: Amazon CloudFront などのエッジサービスを実装して、追加の保護レイヤーやその他の機能を提供します。
  - Amazon CloudFront ユースケース
  - AWS Global Accelerator
  - AWS Web Application Firewall (AWS WAF)
  - Amazon Route 53
  - Amazon VPC Ingress Routing
- プライベートネットワークトラフィックを制御する: ワークロードのプライベートトラフィックを保護するサービスを実装します。
  - Amazon VPC ピアリング
  - Amazon VPC エンドポイントサービス (AWS PrivateLink)
  - Amazon VPC トランジットゲートウェイ
  - AWS Direct Connect
  - AWS サイト間 VPN
  - AWS クライアント VPN
  - Amazon S3 Access Points

リソース

関連するドキュメント:

- AWS Firewall Manager
- Amazon Inspector

- AWS WAF の開始方法

関連動画:

- AWS Transit Gateway reference architectures for many VPCs
- Application Acceleration and Protection with Amazon CloudFront, AWS WAF, andAWS Shield

関連する例:

- Lab: Automated Deployment of VPC

SEC05-BP03 ネットワーク保護を自動化する

保護メカニズムを自動化し、脅威インテリジェンスと異常検出に基づく自己防御型ネットワークを提供します。たとえば、現在の脅威に適応し、その影響を軽減できる侵入検知および防止ツールなどです。ウェブアプリケーションファイアウォールは、ネットワーク保護を自動化できる例の 1 つです。たとえば、AWS WAF セキュリティの自動化ソリューション (https://github.com/awslabs/aws-waf-security-automationsを使用して、既知の脅威アクターに関連付けられた IP アドレスからのリクエストを自動的にブロックします。

このベストプラクティスが確立されていない場合のリスクレベル: ミディアム

実装のガイダンス

- ウェブベースのトラフィックの保護を自動化する: AWS では、AWS CloudFormation を使用して、一般的なウェブベースの攻撃をフィルタリングするために設計された AWS WAF ルールセットを自動的にデプロイするソリューションを提供しています。ユーザーは、AWS WAF ウェブアクセスコントロールリスト (ウェブ ACL) に含まれるルールを定義する、あらかじめ設定された保護機能から選択することができます。
  - AWS WAF のセキュリティオートメーション
- AWS Partner ソリューションを検討する: AWS パートナーは、お客様のオンプレミス環境にある既存のコントロールと同等または統合された、業界をリードする何百もの製品を提供しています。これらの製品は、既存の AWS サービスを補完し、包括的なセキュリティアーキテクチャの導入と、クラウドとオンプレミス環境におけるよりシームレスなエクスペリエンスを実現します。
  - インフラストラクチャのセキュリティ

リソース

関連するドキュメント:

- AWS Firewall Manager
- Amazon Inspector
- Amazon VPC のセキュリティ
- AWS WAF の開始方法

関連動画:

- AWS Transit Gateway reference architectures for many VPCs (多くの VPC 用の AWS Transit Gateway リファレンスアーキテクチャ)
- Application Acceleration and Protection with Amazon CloudFront, AWS WAF, and AWS Shield (Amazon CloudFront、AWS WAF、AWS Shield によるアプリケーションの高速化と保護)

関連する例:

- ラボ: VPC の自動デプロイ

SEC05-BP04 検査と保護を実装する

各レイヤーでトラフィックを検査し、フィルタリングします。VPC の設定に潜在的な意図しないアクセスの可能性がないかを検査するには、 VPC Network Access Analyzer を使用できます。.ネットワークアクセス要件を指定して、それを満たさない潜在的なネットワークパスを特定できます。HTTP ベースのプロトコルを介してトランザクションを実行するコンポーネントの場合、一般的な攻撃からの保護にはウェブアプリケーションファイアウォールが役立ちます。 AWS WAF は、Amazon API Gateway API、Amazon CloudFront、または Application Load Balancer に転送される設定可能なルールに一致する HTTP リクエストを監視してブロックできるウェブアプリケーションファイアウォールです。AWS WAF の使用を開始するには 、 AWS マネージドルール を独自のルールと組み合わせて使用するか、既存の パートナー統合を使用できます。。

AWS Organizations 全体にわたって AWS WAF、AWS Shield Advanced による保護、Amazon VPC セキュリティグループを管理するには、AWS Firewall Manager を使用できます。AWS Firewall Manager を使用すると、アカウントとアプリケーション全体にわたってファイアウォールルールを一元的に設定および管理できるため、一般的なルールの適用を簡単に拡張できます。また、 AWS Shield Advanced、または ソリューション を使用して、攻撃に迅速に対応できます。これらは、ウェブアプリケーションへの不要なリクエストを自動的にブロックします。Firewall Manager は、 AWS ネットワークファイアウォールとも併用できます。.AWS ネットワークファイアウォールは、ルールエンジンを使用して、ステートフルとステートレスの両方のネットワークトラフィックを細かくコントロールするマネージドサービスです。ルールに対しては Suricata 対応の オープンソース侵入防止システム (IPS) 仕様がサポートされており、ワークロードの保護に役立ちます。

このベストプラクティスが確立されていない場合のリスクレベル: 低

実装のガイダンス

- Amazon GuardDuty を設定する: GuardDuty は、脅威検出サービスです。悪意のあるアクティビティや不正な動作を継続的にモニタリングし、AWS アカウント とワークロードを保護します。GuardDuty を有効にし、自動アラートを設定します。
  - Amazon GuardDuty
  - ラボ: 発見的統制の自動デプロイ
- 仮想プライベートクラウド (VPC) フローログを設定する: VPC フローログは、VPC のネットワークインターフェイス間を行き来する IP トラフィックに関する情報をキャプチャできるようにする機能です。フローログデータは Amazon CloudWatch Logs および Amazon Simple Storage Service (Amazon S3) にパブリッシュできます。フローログを作成した後、選択した送信先でデータを取得したり表示したりできます。
- VPC トラフィックのミラーリングを検討する: トラフィックミラーリングは、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスの Elastic Network Interface からネットワークトラフィックをコピーし、コンテンツ検査、脅威のモニタリング、トラブルシューティングのために帯域外セキュリティおよびモニタリングアプライアンスに送信するために使用できる Amazon VPC の機能です。
  - VPC トラフィックミラーリング

リソース

関連するドキュメント:

- AWS Firewall Manager
- Amazon Inspector

- Amazon VPC セキュリティ
- AWS WAF の開始方法

関連動画:

- AWS Transit Gateway reference architectures for many VPCs (多くの VPC 用の AWS Transit Gateway リファレンスアーキテクチャ)
- Application Acceleration and Protection with Amazon CloudFront, AWS WAF, and AWS Shield (Amazon CloudFront、AWS WAF、AWS Shield によるアプリケーションの高速化と保護)

関連する例:

- ラボ: VPC の自動デプロイ

コンピューティングの保護

コンピューティングリソースには、EC2 インスタンス、コンテナ、AWS Lambda 関数、データベースサービス、IoT デバイスなどがあります。これらのコンピューティングリソースタイプには、それぞれ異なるアプローチでセキュリティを確保する必要があります。たしかし、深層防御、脆弱性管理、アタックサーフェスの縮小、設定と運用の自動化、遠隔操作など、検討すべき戦略は共通しています。このセクションでは、主要なサービスのためのコンピューティングリソースを保護するための一般的なガイダンスを紹介します。使用される各 AWS サービスについて、サービス文書に記載されている具体的なセキュリティ推奨事項を確認することが重要です。

ベストプラクティス

- SEC06-BP01 脆弱性管理を実行する (p. 71)
- SEC06-BP02 攻撃対象領域を縮小する (p. 73)
- SEC06-BP03 マネージドサービスを活用する (p. 75)
- SEC06-BP04 コンピューティング保護を自動化する (p. 75)
- SEC06-BP05 ユーザーがリモートからアクションを実行できるようにする (p. 77)
- SEC06-BP06 ソフトウェアの整合性を検証する (p. 77)

SEC06-BP01 脆弱性管理を実行する

コード、依存関係、インフラストラクチャ内の脆弱性のスキャンとパッチ適用を頻繁に実施し、新しい脅威から保護します。

期待される成果: 脆弱性管理プログラムを作成して維持する。Amazon EC2 インスタンス、Amazon Elastic Container Service (Amazon ECS) コンテナ、および Amazon Elastic Kubernetes Service (Amazon EKS) ワークロードなどのリソースを定期的にスキャンしてパッチを適用する。Amazon Relational Database Service (Amazon RDS) データベースなど、AWS マネージドリソースのメンテナンスウィンドウを設定する。静的コードスキャンを使って、アプリケーションソースコードに一般的な問題がないかどうか検査する。組織に必要なスキルがあるかどうか、または外部のアシスタンスを雇用できるかどうか調べるために、Web アプリケーションペネトレーションテストを検討します。

一般的なアンチパターン:

- 脆弱性管理プログラムがない。
- 重大度またはリスク回避を考慮せずに、システムパッチ適用を実施する。
- ベンダーが提供する耐用年数 (EOL) を過ぎたソフトウェアを使用する。

- セキュリティの問題を分析する前に、本番環境にコードをデプロイする。

このベストプラクティスを活用するメリット:

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

脆弱性管理プログラムには、セキュリティ評価、問題の特定、優先順位付け、問題解決の一環としてのパッチ適用の実施などが含まれます。オートメーションは、ワークロードの問題や意図しないネットワークへの露出を継続的にスキャンし、修復を実行するための鍵となります。リソースの作成と更新を自動化することにより時間の節約となり、それ以上の問題を生じさせる設定エラーのリスクを低減します。優れた設計の脆弱性管理プログラムでは、ソフトウェアライフサイクルの開発およびデプロイ段階における脆弱性テストも考慮する必要があります。開発とデプロイ中に脆弱性管理を実装することにより、脆弱性が本番環境に入り込む可能性を低減させます。

脆弱性管理プログラムを実装するには、AWS 責任共有モデルと、それが特定のワークロードにどのように関連するかを理解する必要があります。責任共有モデルでは、AWS に AWS クラウド のインフラストラクチャを保護する責任があります。このインフラストラクチャは、AWS クラウド クラウドサービスを実行するハードウェア、ソフトウェア、ネットワーク、および施設で構成されています。ユーザーには、実績データ、セキュリティ設定、Amazon EC2 インスタンスの管理タスクなどクラウド内のセキュリティ、さらにはAmazon S3 オブジェクトが適切に分類・設定されていることを確認する責任があります。脆弱性管理へのアプローチは、利用するサービスによっても異なります。たとえば、AWS はマネージド型のリレーショナルデータベースサービス Amazon RDS に対するパッチ適用を管理しますが、自己ホスト型データベースのパッチ適用はユーザーの責任となります。

AWS には、脆弱性管理プログラムに役立つ様々なサービスがあります。Amazon Inspector は、ソフトウェアの問題と意図しないネットワークアクセスを検出するために、継続的に AWS ワークロードをスキャンします。AWS Systems Manager Patch Manager を使うと、Amazon EC2 インスタンス全体のパッチ適用を管理できます。Amazon Inspector と Systems Manager は、AWS Security Hub で表示できます。これは、AWS セキュリティチェックを自動化して、セキュリティアラートを一元化するのに役立つクラウドセキュリティ体制管理サービスです。

Amazon CodeGuru を使うと、静的コード分析を使って、Java および Python アプリケーションの潜在的問題を特定できます。

実装手順

- Amazon Inspector を設定する: Amazon Inspector は新たに起動された Amazon EC2 インスタンス、Lambda 関数、および Amazon ECR にプッシュされた適格なコンテナイメージを自動的に検出し、ソフトウェア問題、潜在的な欠陥、および意図しないネットワーク露出がないかスキャンします。
- ソースコードをスキャンする: ライブラリと依存関係をスキャンして、問題と欠陥がないか調べます。Amazon CodeGuru は、Java と Python アプリケーションの両方について 一般的なセキュリティ問題を修復するための推奨事項を伝えます。The OWASP Foundation は、Source Code Analysis Tools (SAST ツール) を公開しています。
- 既存環境をスキャンしてパッチを適用するメカニズム、さらには CI/CD パイプラインのビルドプロセスの一環としてスキャンするメカニズムを導入する: 新しい脅威からの保護を強化するため、依存関係や OS の問題をスキャンしてパッチを適用するメカニズムを実装します。そのメカニズムを定期的に実行します。ソフトウェア脆弱性管理は、パッチを適用したりソフトウェア問題に対処したりする状況を理解するのに不可欠です。継続的インテグレーション/継続的デリバリー (CI/CD) パイプラインの早期に脆弱性評価を組み込むことで、潜在的なセキュリティ脆弱性の問題修復の優先度を決定します。アプローチは、利用する AWS サービスによっても異なります。Amazon EC2 インスタンスで実行するソフトウェアの潜在的問題をチェックするには、Amazon Inspector をパイプラインに追加して、問題や潜在的欠陥が検出されたらアラートを発動して、ビルドプロセスを停止します。Amazon Inspector は継続的にリソースをモニタリングします。脆弱性管理には、OWASP Dependency-Check、Snyk、OpenVAS、パッケージマネージャー、および AWS Partner ツールを使うこともできます。

- AWS Systems Manager を使用する: Amazon Elastic Compute Cloud (Amazon EC2) インスタンス、Amazon マシンイメージ (AMI)、およびその他多くのコンピューティングリソースなど、AWS リソースのパッチ管理を行う責任があります。AWS Systems Manager Patch Manager は、セキュリティ関連および他のタイプの更新の両方を使用して、マネージドインスタンスにパッチを適用するプロセスを自動化します。Patch Manager は、Microsoft アプリケーション、Windows sellable サービスパック、およびLinux ベースインスタンスのマイナーアップグレードなど、オペレーティングシステムとアプリケーション両方の Amazon EC2 インスタンスに対するパッチ適用に使用できます。Amazon EC2 に加え、Patch Manager はオンプレミスサーバーへのパッチ適用にも使用できます。

  サポート対象であるオペレーティングシステムの一覧については、Systems Manager ユーザーガイドの「Supported operating systems (サポートされるオペレーティングシステム) 」で確認してください。インスタンスをスキャンして、不足しているパッチのレポートのみを表示したり、不足しているすべてのパッチをスキャンして自動的にインストールしたりできます。
- AWS Security Hub を使用する: Security Hub AWS の総合的なセキュリティ状態を把握できます。複数の AWS サービス全体のセキュリティデータを収集して、標準化されたフォーマットでそれらの検出結果を提供し、AWS サービス全体のセキュリティ検出結果の優先順位を決定できます。
- AWS CloudFormation を使用する: AWS CloudFormation は、複数のアカウントや環境にまたがるリソースデプロイの自動化やリソースアーキテクチャの標準化により、脆弱性管理を支援する Infrastructure as code (IaC) サービスです。

リソース

関連するドキュメント:

- AWS Systems Manager
- Security Overview of AWS Lambda (AWS Lambda のセキュリティ概要)
- Amazon CodeGuru
- Improved, Automated Vulnerability Management for Cloud Workloads with a New Amazon Inspector (改善、自動化された新しい Amazon Inspector のクラウドワークロードの脆弱性管理)
- Automate vulnerability management and remediation in AWS using Amazon Inspector and AWS Systems Manager – Part 1 (Amazon Inspector と AWS Systems Manager を使って AWS の脆弱性管理と修復を自動化する - パート 1)

関連動画:

- Securing Serverless and Container Services (サーバーレスおよびコンテナサービスを保護する)
- Security best practices for the Amazon EC2 instance metadata service (Amazon EC2 インスタンスメタデータサービスにおけるセキュリティベストプラクティス)

SEC06-BP02 攻撃対象領域を縮小する

オペレーティングシステムを強化し、使用するコンポーネント、ライブラリ、外部から利用可能なサービスを最小限に抑えることで、意図しないアクセスへの露出を減らします。まずオペレーティングシステムパッケージやアプリケーション (Amazon Elastic Compute Cloud (Amazon EC2) ベースのワークロード)、あるいはコード内の外部ソフトウェアモジュールなどの、未使用のコンポーネント (すべてのワークロード) を減らします。一般的なオペレーティングシステムやサーバーソフトウェア向けの強化およびセキュリティ設定ガイドが多数あります。例えば、 Center for Internet Security から始めて、反復できます。

Amazon EC2 では、パッチしたり強化したりした自身の Amazon マシンイメージ (AMI) を作成して、組織の具体的なセキュリティ要件を満たすのに役立てることができます。AMI に適用するパッチやその他のセキュリティコントロールは、作成された時点では効果的です。起動後、例えば AWS Systems Manager で変更しない限り動的ではありません。

EC2 Image Builder を使って、安全な AMI をビルドするプロセスを簡素化できます。EC2 Image Builder は、自動化を記述して維持せずにゴールデンイメージを作成して維持するための作業を大幅に軽減します。ソフトウェアアップデートが利用可能になると、ユーザーがイメージビルドを手動で開始しなくても、新しいイメージが自動作成されます。EC2 Image Builder では、AWS 提供のテストと自分のテストの本番で使用する前に、イメージの機能とセキュリティを簡単に検証できます。また AWS 提供のセキュリティ設定を適用して、イメージをさらにセキュリティ保護し、内部セキュリティ条件を満たすことができます。例えば、AWS を使って、セキュリティテクニカル実装ガイド (STIG) に準拠するイメージを作成できます。

サードパーティー製の静的コード分析ツールを使用して、チェックされていない関数入力境界や、該当する共有脆弱性および露出 (CVE) などの一般的なセキュリティ問題を特定できます。専用のインフラストラクチャで Amazon CodeGuru を、サポートされる言語に対して使用できます。コードがリンクしているライブラリが最新バージョンであるかどうか、ライブラリ自体に CVE が含まれていないかどうか、ライブラリにソフトウェアポリシー要件を満たすライセンス条件があるかどうかを判断するために依存関係チェックツールを使用することもできます。

Amazon Inspectorを使用すると、インスタンスに対する設定評価を実行して既知の CVE を確認したり、セキュリティベンチマークに対して評価したり、欠陥の通知を自動化したりすることができます。Amazon Inspector は本番環境インスタンス上またはビルドパイプライン上で実行され、調査結果があるとデベロッパーとエンジニアに通知します。調査結果にはプログラムを使用してアクセスし、バックログやバグ追跡システムにチームを誘導することができます。 EC2 Image Builder は、自動パッチ適用、AWS が提供するセキュリティポリシーの適用、その他のカスタマイズにより、サーバーイメージ (AMI) を保持するために使用できます。コンテナを使用する場合は、ビルドパイプラインの ECR イメージスキャン をイメージリポジトリに対して定期的に実行し、コンテナ内の CVE を探します。

Amazon Inspector やその他のツールは、設定や CVE の有無を特定するのには効果的ですが、アプリケーションレベルでワークロードをテストするには他の方法が必要になります。 ファジング は、オートメーションを使用して不正な形式のデータを入力フィールドやアプリケーションの他の領域に挿入するバグを見つけるためのよく知られた手法です。

このベストプラクティスを活用しない場合のリスクレベル: 高

実装のガイダンス

- オペレーティングシステムを強化する: ベストプラクティスを満たすようにオペレーティングシステムを設定します。
  - Amazon Linux のセキュリティ保護
  - Microsoft Windows Server のセキュリティ保護
- コンテナ化されたリソースを強化する: セキュリティのベストプラクティスを満たすよう、コンテナ化されたリソースを設定します。
- AWS Lambda のベストプラクティスを導入する
  - AWS Lambda のベストプラクティス

リソース

関連するドキュメント:

- AWS Systems Manager
- 要塞ホストを Amazon EC2 Systems Manager と置換する
- AWS Lambda のセキュリティ概要

関連動画:

- Amazon EKS で高セキュリティワークロードを実行する

- サーバーレスおよびコンテナサービスを保護する
- Amazon EC2 インスタンスメタデータサービスのセキュリティに関するベストプラクティス

関連する例:

- ラボ: ウェブアプリケーションファイアウォールの自動デプロイ

SEC06-BP03 マネージドサービスを活用する

Amazon Relational Database Service (Amazon RDS)、AWS Lambda、Amazon Elastic Container Service (Amazon ECS) などのリソースを管理するサービスを実装し、共有責任モデルの一部としてのセキュリティメンテナンスタスクを減らします。例えば、Amazon RDS は、リレーショナルデータベースのセットアップ、運用、スケーリングを支援し、ハードウェアのプロビジョニング、データベースのセットアップ、パッチ適用、バックアップなどの管理タスクを自動化します。つまり、「AWS Well-Architected フレームワーク」で説明されているその他の方法でアプリケーションを保護することに集中できる時間が増加します。Lambda では、サーバーのプロビジョニングや管理を行わずにコードを実行できるため、インフラストラクチャやオペレーティングシステムではなく、コードレベルの接続、呼び出し、セキュリティに集中するだけで済みます。

このベストプラクティスを活用しない場合のリスクレベル: ミディアム

実装のガイダンス

- 利用可能なサービスを調べる: Amazon RDS、AWS Lambda、Amazon ECS などのリソースを管理するサービスを調査、テスト、実装します。

リソース

関連するドキュメント:

- AWS ウェブサイト
- AWS Systems Manager
- Replacing a Bastion Host with Amazon EC2 Systems Manager
- Security Overview of AWS Lambda

関連動画:

- Running high-security workloads on Amazon EKS
- サーバーレスおよびコンテナサービスを保護する
- Security best practices for the Amazon EC2 instance metadata service

関連する例:

- Lab: AWS Certificate Manager Request Public Certificate

SEC06-BP04 コンピューティング保護を自動化する

脆弱性管理、攻撃対象領域削減、リソース管理などのコンピューティング保護メカニズムを自動化します。自動化により、ワークロードの他の側面の保護に時間を使えるようになり、人為的ミスを犯すリスクを軽減できます。

このベストプラクティスを活用しない場合のリスクレベル: ミディアム

実装のガイダンス

- 設定管理を自動化する: 設定管理サービスまたはツールを使用して、リモートでアクションを実行し、安全な設定を自動的に適用および検証します。
  - AWS Systems Manager
  - AWS CloudFormation
  - ラボ: VPC の自動デプロイ
  - ラボ: EC2 ウェブアプリケーションの自動デプロイ

- Amazon Elastic Compute Cloud (Amazon EC2) インスタンスのパッチを自動化する: インスタンスの場合、AWS Systems Manager Patch Manager は、セキュリティ関連および他のタイプの更新の両方を使用して、マネージドインスタンスにパッチを適用するプロセスを自動化します。Patch Manager を使用して、オペレーティングシステムとアプリケーションの両方にパッチを適用できます。
  - AWS Systems Manager パッチマネージャーを使用して
  - AWS Systems Manager オートメーションを使った一元化されたマルチアカウントおよびマルチリージョンパッチ適用

- 侵入検知と防止ツールを実装する: 侵入検知と防止ツールを実装することで、インスタンス上の悪意のあるアクティビティをモニタリングし、停止できます。
- AWS Partner ソリューションを検討する: AWS パートナーは、オンプレミス環境の既存のコントロールと同等、同一、またはそれらと統合される、業界をリードする多くの製品を提供しています。これらの製品は、AWS の既存のサービスを補完し、クラウド環境とオンプレミス環境にわたって包括的なセキュリティアーキテクチャと、よりシームレスなエクスペリエンスをデプロイできるようにします。
  - インフラストラクチャのセキュリティ

リソース

関連するドキュメント:

- AWS CloudFormation
- AWS Systems Manager
- AWS Systems Manager パッチマネージャーを使用して
- AWS Systems Manager オートメーションを使った一元化されたマルチアカウントおよびマルチリージョンパッチ適用
- インフラストラクチャのセキュリティ
- 要塞ホストを Amazon EC2 Systems Manager と置換する
- AWS Lambda のセキュリティ概要

関連動画:

- Amazon EKS で高セキュリティワークロードを実行する
- サーバーレスおよびコンテナサービスを保護する
- Amazon EC2 インスタンスメタデータサービスのセキュリティに関するベストプラクティス

関連する例:

- ラボ: ウェブアプリケーションファイアウォールの自動デプロイ

- ラボ: EC2 ウェブアプリケーションの自動デプロイ

SEC06-BP05 ユーザーがリモートからアクションを実行できるようにする

インタラクティブアクセスの機能を排除すると、人為的ミスのリスクが軽減され、設定や管理が手動で行われる可能性が低くなります。たとえば、直接アクセスや踏み台ホスト経由のアクセスを許可する代わりに、infrastructure-as-codeを使って Amazon Elastic Compute Cloud (Amazon EC2) インスタンスをデプロイし、次に AWS Systems Manager などのツールを使って Amazon EC2 を管理します。AWS Systems Manager は、 オートメーション ワークフロー、io1ドキュメント (プレイブック)、Run Commandなどの機能を使用して、さまざまなメンテナンスおよびデプロイタスクを自動化できます。AWS CloudFormation スタックは、パイプラインから構築され、AWS Management Console や API を直接使用することなく、インフラストラクチャのデプロイおよび管理タスクを自動化できます。

このベストプラクティスを活用しない場合のリスクレベル: 低

実装のガイダンス

- コンソールアクセスを置き換える: インスタンスへのコンソールアクセス (SSH または RDP) を AWS Systems Manager Run Command に置き換えて、管理タスクを自動化します。

- AWS Systems Manager Run Command

リソース

関連するドキュメント:

- AWS Systems Manager
- AWS Systems Manager Run Command
- Replacing a Bastion Host with Amazon EC2 Systems Manager
- Security Overview of AWS Lambda

関連動画:

- Running high-security workloads on Amazon EKS
- サーバーレスおよびコンテナサービスを保護する
- Security best practices for the Amazon EC2 instance metadata service

関連する例:

- Lab: Automated Deployment of Web Application Firewall

SEC06-BP06 ソフトウェアの整合性を検証する

ワークロードで使用されるソフトウェア、コード、ライブラリが信頼できるソースからのものであり、改ざんされていないことを検証するメカニズム (コード署名など) を実装します。たとえば、バイナリとスクリプトのコード署名証明書を検証して作成者を確認し、作成者が作成してから改ざんされていないことを確認する必要があります。AWS Signer は、署名証明書や パブリックキー、プライベートキーを含むコード署名のライフサイクルを一元管理することで、お客様のコードの信頼性と完全性を確保することがで

きます。コード署名のための高度なパターンとベストプラクティスは、以下で学ぶことができます: AWS Lambda.さらに、ダウンロードするソフトウェアのチェックサムをプロバイダーからのチェックサムと比較し、改ざんされていないことを確認できます。

このベストプラクティスが確立されていない場合のリスクレベル: 低

実装のガイダンス

- メカニズムを検証する: コード署名は、ソフトウェアの整合性を検証するために使用できるメカニズムの 1 つです。
  - NIST: Security Considerations for Code Signing (コード署名の考慮事項)

リソース

関連するドキュメント:

- AWS Signer
- New – Code Signing, a Trust and Integrity Control for AWS Lambda (New – コード署名、AWS Lambda の信頼性および整合性のコントロール)

データ保護

ワークロードを設計する前に、セキュリティに影響を与える基本的なプラクティスを用意しておく必要があります。たとえば、データ分類は機密性のレベルに基づいてデータを分類する方法を提供し、暗号化では、不正なアクセスに対し、データを判読できなくすることでデータを保護します。これらは、規制義務への対応ミスを回避したり、規制義務を順守したりする目的の達成に役立つ重要な方法です。

AWS にはデータ保護対策として使用できる多くのさまざまな方法があります。以下のセクションでは、こうしたアプローチの使用方法を説明します。

トピック

データ分類

データ分類方法を確立すると、重要度と機密性に基づいて組織データをカテゴリ別に分類して、各カテゴリに適した保護と保持方法でデータを管理できるようになります。

ベストプラクティス

SEC07-BP01 ワークロード内のデータを特定する

ワークロードが処理するデータの型と分類、関連するビジネスプロセス、データの保管場所、データ所有者を理解することは非常に重要です。また、ワークロードに適用する法的要件やコンプライアンス要件、そしてどのようなデータコントロールを適用すべきなのかを理解する必要があります。データの特定は、データ分類作業の最初のステップです。

このベストプラクティスを活用するメリット:

データ分類により、ワークロード所有者は、機密データを保存する場所を特定し、そのデータへのアクセス方法や共有方法を決定できます。

データ分類は、次の質問への回答となります:

- どのようなタイプのデータを持っていますか?

  次のようなデータが考えられます:
  - 企業秘密、特許、または契約合意などの知的財産 (IP)。
  - 個人と結びついた医療履歴情報を含む医療記録などの、保護対象医療情報 (PHI)。
  - 氏名、住所、生年月日、国民 ID または登録番号などの個人を特定できる情報 (PII)。
  - 会員番号 (PAN)、カード会員名、有効期限、サービスコード番号などクレジットカードのデータ。
  - 機密性の高いデータはどこに保存しますか?
  - データにアクセス、変更、削除できる人は誰ですか?
  - データの誤った取扱いを防ぐためには、ユーザーのアクセス許可を把握することが不可欠です。

- 作成、読み取り、更新、削除 (CRUD) 操作を実行できるのは誰ですか?
  - 誰がデータに対するアクセス許可を管理できるかを理解することにより、アクセス許可が昇格する可能性を考慮します。
- データが意図せずに開示されたり、変更または削除された場合、ビジネスに対してどのような影響が生じますか?
  - データが変更、削除、または誤って開示された場合のリスク結果を理解します。

これらの質問に対する回答を把握することにより、次のようなアクションを取ることができます。

- 機密データの範囲 (機密データの場所の数など) を縮小し、機密データへのアクセスを承認済みユーザーのみに限定します。
- 暗号化、データ紛失防止、およびアイデンティティやアクセス管理など、適切なデータ保護メカニズムとテクニックを実装できるよう、さまざまなデータ型について理解を深めます。
- データの正しいコントロール目的を提供することにより、コストを最適化します。
- データの型や量、機密度の異なるデータをどのように隔離しているかなど、規制当局や監査人からの質問に自信を持って答えることができます。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

データ分類は、データの機密性を識別する行為です。タグ付けを行って、データを簡単に検索し、追跡できるようにすることもあります。また、データ分類を行うと、データの重複を減らし、ストレージやバックアップのコストを削減すると同時に、検索プロセスを高速化できます。

Amazon Macie などのサービスを使用して、機密データの検出と分類両方を大規模に自動化します。Amazon EventBridge や AWS Config など他のサービスは、暗号化されていない Amazon Simple Storage Service (Amazon S3) バケットや Amazon EC2 EBS ボリュームまたはタグが付いていないデータリソースなどのデータセキュリティの問題に対する修復を自動化するために使用できます。AWS サービス統合の完全なリストについては、「EventBridge ドキュメント」を参照してください。

顧客のメール、サポートチケット、製品レビュー、およびソーシャルメディアなどの構造化されていないデータで PII を検出することは、Amazon Comprehend を使うことにより実現します。これは、機械学習 (ML)を用いて構造化されていないテキストで人、場所、感情、話題などのインサイトや関係性を見つけ出す自然言語処理 (NLP) サービスです。データ識別に役立つ AWS サービスのリストについては、「AWS サービスを使って PHI と PII データを検出する一般的テクニック」を参照してください。

データ分類と保護をサポートするもう 1 つの方法は、AWS リソースのタグ付けです。タグ付けを行うと、リソースの管理、特定、整理、検索、およびフィルタリングに使用できる AWS リソースにメタデータを割り当てることができます。

場合によっては、特定のワークロードやサービスが既知のデータ分類のプロセスまたは伝送を保存することが期待されている場合など、リソース全体 (S3 バケットなど) にタグを付けるよう選択することがあります。

適切な場合、管理業務とセキュリティメンテナンスを行いやすくするため、個別のオブジェクトではなく S3 バケットにタグ付けをすることもできます。

実装手順

Amazon S3 内の機密データを検出する:

1. 開始する前に、Amazon Macie コンソールと API オペレーションにアクセスするための適切なアクセス権限があることを確認してください。さらに詳しい情報については、「Amazon Macie の開始方法」を参照してください。

2. Amazon Macie を使用して、機密データが Amazon S3 内にある場合は自動化されたデータ検出を実行します。
   - Amazon Macie の開始方法ガイドを使って、機密データ検出結果のリポジトリを設定し、機密データの検出ジョブを作成します。
   - How to use Amazon Macie to preview sensitive data in S3 buckets (Amazon Macie を使って、S3 バケットで機密データをプレビューする方法)

     デフォルトでは、Macie が、自動化された機密データの検出に推奨されるマネージドデータ識別子のセットを使用して、オブジェクトを分析します。分析は、アカウントまたは組織に対して自動化された機密データ検出を実行する際に Macie が特定のマネージドデータ識別子、カスタムデータ識別子、許可リストを使うように設定することにより、カスタマイズすることができます。特定のバケット (たとえば、通常 AWS ログデータを保存する S3 バケットなど) を除外することにより、分析の対象範囲を調整できます。
3. 自動化された機密データ検出を設定するには、「Performing automated sensitive data discovery with Amazon Macie」(Amazon Macie を使って自動化された機密データ検出を実行する) を参照してください。
4. また、Automated Data Discovery for Amazon Macie (Amazon Macie の自動化されたデータ検出) も検討してください。

Amazon RDS 内の機密データを検出する:

Amazon Relational Database Service (Amazon RDS) データベース内のデータ検出の詳細については、「Enabling data classification for Amazon RDS database with Macie」(Macie で Amazon RDS データベースのデータ分類を有効化する) を参照してください。

DynamoDB 内の機密データを検出する:

- Detecting sensitive data in DynamoDB with Macie (Macie を使って DynamoDB 内で機密データを検出する) では、Amazon Macie を使って、Amazon S3 にエクスポートしてスキャンすることで、 Amazon DynamoDB テーブル内で機密データを検出する方法を説明しています。

AWS パートナーのソリューション

- 広範囲の AWS Partner Network の使用を検討してください。AWS パートナーには、AWS サービスと直接統合する広範囲のツールとコンプライアンスフレームワークがあります。パートナーは、組織のニーズを満たすのに役立つカスタマイズされたガバナンスとコンプライアンスのソリューションを提供します。
- データ分類におけるカスタマイズされたソリューションについては、「Data governance in the age of regulation and compliance requirements」(規制およびコンプライアンス要件時代のデータガバナンス) を参照してください。

AWS Organizations を使用してポリシーを作成およびデプロイすることにより、組織が採用するタグ付け標準を自動的に適用できます。タグポリシーを使用すると、有効なキー名と各キーに対して有効な値を指定できます。モニタリングのみの選択も可能で、既存のタグを評価し、クリーンアップする機会を提供します。タグが選択した標準を満たすと、タグポリシーで適用をオンにして、非準拠のタグが作成されるのを防ぐことができます。詳細については、Securing resource tags used for authorization using a service control policy in AWS Organizations (AWS Organizations のサービスコントロールポリシーを使用して、認可に使用するリソースタグを保護する) および preventing tags from being modified except by authorized principals (タグを許可されたプリンシパル以外が変更できないようにする) のサンプルポリシーを参照してください。

- AWS Organizations でタグポリシーの使用を開始するには、より高度なタグポリシーに移行する前に、「Getting started with tag policies」(タグポリシーの開始方法) を参照してください。組織単位 (OU) または組織全体に拡大する前に、単純なタグポリシーを単一のアカウントにアタッチした場合の効果を理解することにより、タグポリシーの遵守を適用する前にタグポリシーの効果を確認することができま

す。Getting started with tag policies (タグポリシーの開始方法) には、より高度なポリシー関連タスクの実行方法へのリンクが記載されています。

- データ分類をサポートする他の AWS サービスおよび機能を評価することを検討してください。これらは、データ分類ホワイトペーパーに列挙されています。

リソース

関連するドキュメント:

- Getting started with Amazon Macie (Amazon SQS の開始方法)
- Automated data discovery with Amazon Macie (Amazon Macie を使用した自動データ検出)
- Getting started with tag policies (タブポリシーの開始方法)
- Detecting PII entities (PI エンティティの検出)

関連ブログ:

- How to use Amazon Macie to preview sensitive data in S3 buckets (Amazon Macie を使って、S3 バケットで機密データをプレビューする方法)
- Performing automated sensitive data discovery with Amazon Macie (Amazon Macie を使って自動化された機密データ検出を実行する)
- Common techniques to detect PHI and PII data using AWS Services (AWS のサービスを使って PHI と PII データを検出する一般的なテクニック)
- Detecting and redacting PII using Amazon Comprehend (Amazon Comprehend を使用した PII の検出と再編集)
- Securing resource tags used for authorization using a service control policy in AWS Organizations (AWS Organizations のサービスコントロールポリシーを使用して、認可に使用するリソースタグを保護する)
- Enabling data classification for Amazon RDS database with Macie (Macie で Amazon RDS データベースのデータ分類を可能にする)
- Detecting sensitive data in DynamoDB with Macie (Macie を使った DynamoDB の機密データの検出)
-

関連動画:

- Event-driven data security using Amazon Macie (Amazon Macie を使用したイベント駆動型データセキュリティ)
- Amazon Macie for data protection and governance (データ保護とガバナンスのための Amazon Macie)
- Fine-tune sensitive data findings with allow lists (許可リストで機密データ検出を微調整する)

SEC07-BP02 データ保護コントロールを定義する

分類レベルに従ってデータを保護します。たとえば、関連するレコメンデーションを使用してパブリックとして分類されたデータを保護すると同時に、追加のコントロールで機密データを保護します。

リソースタグ、機密性ごと (および注意事項、エンクレーブ、関心のあるコミュニティごと) の個別の AWS アカウント 、IAM ポリシー、AWS Organizations SCP、AWS Key Management Service (AWS KMS)、AWS CloudHSM を使用することで、暗号化によるデータ分類と保護のポリシーを定義および実装できます。たとえば、非常に重要なデータを含む S3 バケット、または、秘密データを処理する Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを含むプロジェクトがある場合、それらに `#Project=ABC#` を付けることができます。直属のチームのみがこのプロジェクトコードの意味を知って

いて、属性ベースのアクセス統制手段を使用する方法を提供します。キーポリシーと許可を使用して AWS KMS 暗号化キーへのアクセスレベルを定義し、安全なメカニズムを通じて適切なサービスだけが機密コンテンツにアクセスできるようにします。タグに基づいて承認決定を判断する場合、AWS Organizations 内のタグポリシーを使用して、タグの許可が適切に定義されていることを確認する必要があります。

このベストプラクティスを活用しない場合のリスクレベル: 高

実装のガイダンス

- データの識別および分類スキーマを定義する: データの識別と分類は、保存するデータの潜在的な影響とタイプ、およびデータにアクセスできるユーザーを評価するために実行されます。
  - AWS ドキュメント
- 利用可能な AWS のコントロールを確認する: 使用しようとしているか、使用を計画している AWS サービスについて、セキュリティコントロールを確認します。多くのサービスには、ドキュメントにセキュリティセクションがあります。
  - AWS ドキュメント
- AWS コンプライアンスリソースを特定する: 支援のために使用できる AWS のリソースを特定します。
  - https://aws.amazon.com/compliance/

リソース

関連するドキュメント:

- AWS ドキュメント
- データ分類に関するホワイトペーパー
- Amazon Macie の開始方法
- 欠落テキスト

関連動画:

- Introducing the New Amazon Macie

SEC07-BP03 識別および分類を自動化する

データの識別と分類を自動化すると、適切な統制を実装するのに役立ちます。人が直接アクセスするよりも自動化した方が、人為的ミスや開示リスクは小さくなります。など、 Amazon Macieなどの、機械学習を使用して AWS の機密データを自動的に検出、分類、保護するツールの利用を評価する必要があります。Amazon Macie は個人識別情報 (PII) や知的財産などの機密データを 認識し、このデータへのアクセスや移動の状況を可視化するダッシュボードやアラートを提供します。

このベストプラクティスが確立されていない場合のリスクレベル: ミディアム

実装のガイダンス

- Amazon Simple Storage Service (Amazon S3) インベントリを使用する: Amazon S3 インベントリは、オブジェクトのレプリケーションと暗号化ステータスの監査とレポートに使用できるツールの 1 つです。
  - Amazon S3 インベントリ
- Amazon Macie を検討する: Amazon Macie は、機械学習を使用して Amazon S3 内に保存されているデータを自動的に検出、分類します。
  - Amazon Macie

リソース

関連するドキュメント:

- Amazon Macie
- Amazon S3 インベントリ
- データ分類に関するホワイトペーパー
- Amazon Macie の開始方法

関連動画:

- Introducing the New Amazon Macie (新しい Amazon Macieの紹介)

SEC07-BP04 データのライフサイクル管理を定義する

定義されるライフサイクル戦略は、機密性レベル、また法的および組織の要件に基づいている必要があります。データを保持する期間、データ破壊プロセス、データアクセス管理、データ変換、データ共有などの側面を考慮する必要があります。データ分類方法を選択するときは、可用性とアクセスのバランスを取ります。また、各レベルにとって安全でありながら使いやすい方式を採用するために、複数レベルのアクセスと微妙な差異も実装する必要がります。常に多層防御方式を採用し、データおよびデータの変換、削除、コピーのメカニズムに人間がアクセスする機会を減らします。例えば、アプリケーション認証を厳格にし、遠距離操作を実行するために必要なアクセス許可をユーザーでなくアプリケーションに付与します。さらに、ユーザーが信頼できるネットワークパスからアクセスしていることを確認して、復号鍵へのアクセスを要求します。ユーザーにデータへの直接アクセス権を付与するのではなく、ダッシュボードや自動レポートなどのツールを使用して、データからの情報をユーザーに提供します。

このベストプラクティスを活用しない場合のリスクレベル: 低

実装のガイダンス

- データタイプを識別する: ワークロードに保存または処理するデータのタイプを特定します。そのデータは、テキスト、イメージ、バイナリデータベースなどが考えられます。

リソース

関連するドキュメント:

- データ分類に関するホワイトペーパー
- Amazon Macie の開始方法

関連動画:

- 新しい Amazon Macie の導入

保管中のデータの保護

保管中のデータ とは、ワークロードの任意の期間に永続的ストレージに保持されるすべてのデータを指します。たとえば、ブロックストレージ、オブジェクトストレージ、データベース、アーカイブ、IoT デバイス、データが保持されているその他のストレージ媒体などがあります。暗号化と適切なアクセスコン

トロールが実装されている場合は、保管中のデータを保護することで不正アクセスのリスクを軽減できます。

暗号化とトークン分割は、共に重要なデータ保護スキームですが、異なる特徴を持ちます。

トークン分割 とは、機密情報を表すトークン (顧客のクレジットカード番号を表すトークンなど) を定義するためのプロセスです。トークンはそれ自体に意味があってはいけません。また、トークン化中のデータから派生してはいけません。このため、暗号化ダイジェストはトークンとしては使用できません。トークン分割方式を慎重に計画することで、コンテンツの保護を強化し、コンプライアンス要件を確実に満たすことができます。たとえば、クレジットカード番号の代わりにトークンを利用すると、クレジットカード処理システムに関するコンプライアンスの範囲を狭めることができます。

暗号化 とは、プレーンテキストに復号化するために必要な秘密鍵がないとコンテンツを読めないように変換する方法です。必要に応じてトークン分割と暗号化の両方を使用して、情報の安全を確保し、保護することができます。この他に、マスキング手段を使用すると、残りのデータが機密とみなされないポイントまでデータの一部を編集できます。たとえば、PCI-DSS では、カード番号の最後の4桁をコンプライアンススコープの境界外に保持して、インデックスを作成できます。

暗号化キーの使用を監査する: 暗号化キーの使用方法を理解したうえで、監査を実施し、キーのアクセス制御メカニズムが適切に実践されていることを検証します。例えば、AWS KMS キーを使用するすべての AWS サービスが、毎回の使用を AWS CloudTrail に記録します。その後、Amazon CloudWatch Insights などのツールを使用して AWS CloudTrail にクエリを実行し、キーの使用がすべて有効であることを確認できます。

ベストプラクティス

SEC08-BP01 安全なキー管理を実装する

安全なキー管理には、ワークロード用に保管中のデータを保護するために必要な、キーマテリアルの保管、ローテーション、アクセス制御、監視が含まれます。

期待される成果: スケーラブルで反復可能な、自動化されたキー管理メカニズム。このメカニズムは、キーマテリアルへの最小特権アクセス権を強制する能力を提供し、キーの可用性、機密性、完全性の適切なバランスを実現できるものでなければなりません。キーへのアクセスは監視され、キーマテリアルは自動化されたプロセスでローテーションされる必要があります。キーの内容は、決して人的 ID にアクセス可能なものであってはなりません。

一般的なアンチパターン:

- 暗号化されていないキーマテリアルに人間がアクセスする。
- カスタム暗号化アルゴリズムを作成する。
- キーマテリアルへのアクセス許可の範囲が広すぎる。

このベストプラクティスを活用するメリット: ワークロード用の安全なキー管理メカニズムを確立することで、不正アクセスからコンテンツを保護することができます。さらに、データの暗号化を要求する規制要件の対象となる場合があります。効果的なキー管理ソリューションがあれば、それらの規制に合わせた技術的メカニズムを提供して、キーマテリアルを保護することができます。

このベストプラクティスを活用しない場合のリスクレベル: 高

実装のガイダンス

多くの規制要件やベストプラクティスには、基本的なセキュリティ制御として保管中のデータの暗号化が含まれています。この制御に準拠するには、ワークロードに、保管中のデータの暗号化に使用されるキーマテリアルを安全に保存および管理するメカニズムが必要です。

AWS は AWS Key Management Service (AWS KMS) を使用して AWS KMS キー用の高い耐久性と安全性を備えた冗長ストレージを提供します。 AWS の多くのサービスがデータの暗号化に対応するため AWS KMS と統合しています。AWS KMS は、FIPS 140-2 レベル 3 検証済みのハードウェアセキュリティモジュールを使用してキーを保護します。AWS KMS キーをプレーンテキストでエクスポートするメカニズムはありません。

マルチアカウント戦略を使用してワークロードをデプロイする場合、 ベストプラクティス として、AWS KMS キーを、そのキーを使用するワークロードと同じアカウントに保持することが考えられます。この分散型モデルでは、AWS KMS キーの管理責任はアプリケーションチームにあります。他のユースケースでは、組織は AWS KMS キーを一元管理されたアカウントに保存することもできます。この一元化された構造では、ワークロードアカウントが統合アカウントに保存されているキーにアクセスするために必要なクロスアカウントアクセスを可能にする追加のポリシーが必要ですが、単一のキーを複数の AWS アカウント で共有するユースケースにより適している可能性があります。

キーマテリアルを保管する場所にかかわらず、キーへのアクセスは キーポリシーと IAM ポリシーを使用して厳重に管理する必要があります。キーポリシーは、AWS KMS キーへのアクセスを制御する主な方法です。さらに、AWS KMS キーの付与によって、ユーザーに代わってデータを暗号化および復号する AWS のサービスへのアクセスを提供できます。時間を取り、 AWS KMS キーへのアクセス制御に関するベストプラクティスを確認してください。

暗号化キーの使用状況を監視して、異常なアクセスパターンを検出するのがベストプラクティスです。AWS マネージドキーと AWS KMS に保存されているカスタマーマネージドキーを使用して実行される操作は AWS CloudTrail で口グインできるため、定期的に確認する必要があります。キー破壊イベントの監視には特に注意する必要があります。キーマテリアルの偶発的または悪意のある破壊を防ぐため、キー破壊イベントによってキーマテリアルがすぐに削除されることはありません。AWS KMS でキーを削除した場合に、デフォルトで 30 日間の 待機期間が設けられおり、管理者はこれらのアクションを確認し、必要に応じてリクエストをロールバックする時間を確保できます。

AWS の多くのサービスはわかりやすい方法で AWS KMS を使用します。唯一必要なのは、AWS マネージドキーとカスタマーマネージドキーのどちらを使用するかを決定することです。ワークロードでデータを暗号化または復号するために直接 AWS KMS を使用する必要がある場合は、データを保護するため エンベロープ暗号化 を使用するのがベストプラクティスです。この AWS Encryption SDK は、アプリケーションにクライアント側の暗号化プリミティブを提供し、エンベロープ暗号化を実装して AWS KMS と統合することができます。

実装手順

1. キーに適切な キー管理オプション を決定します (AWS マネージドまたはカスタマーマネージド)。
   - 使いやすさを考慮して、AWS はほとんどのサービスにおいて AWS 所有キーと AWS マネージドキーを提供しています。これにより、キーマテリアルやキーポリシーを管理しなくても保管時の暗号化が可能になります。
   - カスタマーマネージドキーを使用する場合は、俊敏性、セキュリティ、データ主権、可用性の最適なバランスを実現するデフォルトのキーストアを検討してください。他のユースケースでは、 AWS CloudHSM や 外部キーストアでカスタムキーストアを使用する必要があるかもしれません。
2. ワークロードに使用しているサービスのリストを確認して、AWS KMS がサービスとどのように統合されているかを理解します。例えば、EC2 インスタンスは暗号化された EBS ボリュームを使用できます。これにより、そのボリュームから作成された Amazon EBS スナップショットもカスタマーマネージドキーを使用して暗号化されていることを確認し、暗号化されていないスナップショットデータが誤って開示されるのを防ぐことができます。
   - AWS のサービスで AWS KMS を使用する方法

    - AWS のサービスが提供する暗号化オプションの詳細については、サービスのユーザーガイドまたは開発者ガイドの「保管時の暗号化」トピックを参照してください。
3. AWS KMS の実装: AWS KMS を使用すると、キーの作成と管理が簡単になり、幅広い AWS のサービスやアプリケーションでの暗号化の使用を制御できます。
    - 開始方法: AWS Key Management Service (AWS KMS)
    - AWS KMS キーのアクセス制御については、 ベストプラクティスを確認してください。
4. AWS Encryption SDK の検討: アプリケーションがクライアント側でデータを暗号化する必要がある場合は、AWS KMS 統合済みの AWS Encryption SDK を使用してください。
    - AWS Encryption SDK
5. を 有効にし、 過度に広範な AWS KMS キーポリシーがないかどうかを自動的に確認し、通知を受け取るようにします。
6. Security Hub を 有効にし、 キーポリシーの設定ミス、削除予定のキー、自動ローテーションが有効になっていないキーがある場合に通知を受け取るようにします。
7. AWS KMS キーに適したログ記録レベルを決定します。AWS KMS への呼び出し (読み取り専用イベントを含む) はログに記録されるため、AWS KMS に関連する CloudTrail ログが膨大になる可能性があります。
    - 組織によっては、AWS KMS のログ記録アクティビティを別の証跡に分けた方がよい場合があります。詳細については、AWS KMS デベロッパーガイドの CloudTrail による AWS KMS API コールのログ記録 セクションを参照してください

リソース

関連するドキュメント:

- AWS Key Management Service
- AWS 暗号化サービスとツール
- 暗号化を使用して Amazon S3 データを保護する
- エンベロープ暗号化
- デジタル主権に関する誓約
- Demystifying AWS KMS key operations, bring your own key, custom key store, and ciphertext portability
- AWS Key Management Service 暗号化の詳細説明

関連動画:

- How Encryption Works in AWS
- Securing Your Block Storage on AWS
- AWS data protection: Using locks, keys, signatures, and certificates

関連する例:

- Implement advanced access control mechanisms using AWS KMS

SEC08-BP02 保管中に暗号化を適用する

保管中のデータには暗号化の使用を適用する必要があります。暗号化は、不正なアクセスや偶発的な開示が発生した場合、機密性の高いデータの機密を保持します。

期待される成果: プライベートデータが保管中にデフォルトで暗号化される。暗号化を行うと、データの機密性を維持し、意図的または不注意によるデータの開示や流出に対する保護層を追加して強化できます。

暗号化されたデータは、まずそれを解除しないと読み出すこともアクセスすることもできません。暗号化されずに保管されたデータは、インベントリに入れて制御する必要があります。

一般的なアンチパターン:

- デフォルトで暗号化する設定を使用しない。
- 複合キーに過度に寛容なアクセスを提供する。
- 暗号化および復号化キーの使用をモニタリングしない。
- データを暗号化せずに保管する。
- データの使用、タイプ、分類に関係なく、すべてのデータに同じ暗号化キーを使う。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

暗号化キーとワークロード内のデータ分類をマッピングします。このアプローチは、データに単一の暗号化キーまたは非常にわずかな暗号化キーを使用する場合、過度に寛容なアクセスから保護するのに役立ちます (「(ワークロード内のデータを特定する) (p. 79) を参照してください)。

AWS Key Management Service (AWS KMS) は、多くの AWS サービスと統合し、保管中のデータを暗号化しやすくします。例えば、Amazon Simple Storage Service (Amazon S3) では、新しいオブジェクトが自動的に暗号化されるように、バケットの暗号化をデフォルトで設定できます。AWS KMS を使用する際は、どの程度厳格にデータを制限すべきかを検討してください。デフォルトでサービス制御型の AWS KMS キーは、AWS がユーザーに変わって管理および使用します。基盤となる暗号化キーにへのアクセスを細かく管理すべき機密データの場合、カスタマーマネージドキー (CMK) を検討してください。キーポリシーを使用することで、ローテーションやアクセス管理など、CMK を完全に制御できます。

さらに、Amazon Elastic Compute Cloud (Amazon EC2) および Amazon S3 は、デフォルト暗号化を設定することにより、暗号化の適用をサポートしています。AWS Config ルール を使用して、Amazon Elastic Block Store (Amazon EBS) ボリューム、Amazon Relational Database Service (Amazon RDS) インスタンス、および Amazon S3 バケットなどに対して暗号化を使用していることを自動的に確認します。

AWS はまた、クライアント側の暗号化も提供するため、クラウドにアップロードする前にデータを暗号化できます。AWS Encryption SDK は、エンベロープ暗号化を使ってデータを暗号化する方法を提供します。ラッピングキーを提供すると、AWS Encryption SDK が暗号化する各データオブジェクトに対して固有のデータキーを生成します。マネージド単一テナントハードウェアセキュリティモジュール (HSM) が必要な場合は、AWS CloudHSM を検討します。AWS CloudHSM では、FIPS 140-2 レベル 3 検証済み HSM で暗号化キーを生成、インポート、管理できます。AWS CloudHSM のユースケースには、認証局 (CA) 発行用プライベートキーの保護、Oracle データベースに対する Transparent Database Encryption (TDE) の有効化などが挙げられます。AWS CloudHSM Client SDK は、データを AWS にアップロードする前に、AWS CloudHSM 内に保管されたキーを使って、クライアント側でデータを暗号化できるソフトウェアを提供します。Amazon DynamoDB Encryption Client ではまた、DynamoDB テーブルにアップロードする前のアイテムを暗号化および署名することもできます。

実装手順

- Amazon S3 に対して保管中に暗号化を適用する: Amazon S3 バケットのデフォルト暗号化を実施します。

  新しい Amazon EBS ボリュームのデフォルトの暗号化を設定する: 新しく作成したすべての Amazon EBS ボリュームを暗号化形式で作成することを指定します。AWS が提供するデフォルトキーを使用するか、作成したキーを使用するかを選択できます。

  暗号化された Amazon Machine Image (AMI) を設定する: 暗号化を有効化して既存の AMI をコピーすると、自動的にルートボリュームとスナップショットが暗号化されます。

  Amazon RDS 暗号化を設定する: 暗号化オプションを使用して、保管中の Amazon RDS データベースクラスターとスナップショットに対して暗号化を設定します。

各データ分類に対する適切なプリンシパルへのアクセスを制限するポリシーを使って AWS KMS キーを作成および設定する: 例えば、本番環境データの暗号化のために AWS KMS キーを 1 つ、開発またはテストデータの暗号化のためにもう 1 つ作成します。他の AWS アカウント に対してキーアクセスを提供することもできます。開発環境と本番環境のアカウントは別にすることを検討してください。本番環境で開発アカウントのアーティファクトを復号化する必要がある場合、開発アーティファクトを暗号化するのに使用する CMK ポリシーを編集し、本番アカウントにアーティファクトを復号化する機能を付与できます。次に、本番環境が本番で使用するために復号化されたデータをインジェストできます。

追加の AWS サービスで暗号化を設定する: 他の AWS サービスを使用する場合は、サービスの暗号化オプションを決定するために、そのサービスの「セキュリティドキュメント」を参照してください。

リソース

関連するドキュメント:

- AWS Crypto Tools
- AWS ドキュメント
- AWS Encryption SDK
- AWS KMS Cryptographic Details Whitepaper (AWS KMS 暗号化の詳細についてのホワイトペーパー)
- AWS Key Management Service
- AWS cryptographic services and tools (AWS 暗号化サービスとツール)
- Amazon EBS Encryption (Amazon EBS 暗号化)
- Default encryption for Amazon EBS volumes (Amazon EBS ボリュームのデフォルトの暗号化)
- Encrypting Amazon RDS Resources (Amazon RDS リソースの暗号化)
- Amazon S3 バケットに対してデフォルトの暗号化を有効にするにはどうすればよいですか。
- 暗号化を使用して Amazon S3 データを保護する

関連動画:

- How Encryption Works in AWS (AWS の暗号化の仕組み)
- Securing Your Block Storage on AWS (AWS でブロックストレージをセキュリティ保護する)

SEC08-BP03 保管時のデータの保護を自動化する

自動化ツールを使用して保管中のデータの制御を継続的に検証し、強化します。例えば、すべてのストレージリソースが暗号化されていることを確認します。また、 すべての EBS ボリューム が AWS Config ルール。AWS Security Hub は、セキュリティ標準に対する自動チェック機能を通じて、いくつかの制御を検証することもできます。さらに AWS Config ルール は、自動的に 非準拠のリソースを修復できます.

このベストプラクティスを活用しない場合のリスクレベル: ミディアム

実装のガイダンス

保管中のデータ とは、ワークロードの任意の期間に永続的ストレージに保持されるすべてのデータを指します。たとえば、ブロックストレージ、オブジェクトストレージ、データベース、アーカイブ、IoT デバイス、データが保持されているその他のストレージ媒体などがあります。暗号化と適切なアクセスコントロールが実装されている場合は、保管中のデータを保護することで不正アクセスのリスクを軽減できます。

保管中に暗号化を適用する: データを保存する唯一の方法は、暗号化を使用することだということを確実にする必要があります。AWS KMS は、保管中のすべてのデータをより簡単に暗号化できるように、多数

の AWS のサービスとシームレスに統合します。例えば、Amazon Simple Storage Service (Amazon S3) では、 デフォルトの暗号化を バケットに設定して、すべての新しいオブジェクトが自動的に暗号化されるようにすることができます。さらに、Amazon EC2 および Amazon S3 は、デフォルト暗号化を設定することにより、暗号化の適用をサポートしています。専用のインフラストラクチャで AWS マネージド Config ルール を使用して、例えば次の項目に対して暗号化を使用していることを自動的に確認できます: EBS ボリュームio1Amazon Relational Database Service (Amazon RDS) インスタンス、および Amazon S3 バケット.

リソース

関連するドキュメント:

- AWS Crypto Tools
- AWS Encryption SDK

関連動画:

- AWS での暗号化のしくみ
- AWS でブロックストレージを保護する

SEC08-BP04 アクセスコントロールを適用する

保管中のデータを保護するには、分離やバージョニングなどのメカニズムを使ってアクセス制御を実施し、最小特権の原則を適用してください。データへパブリックアクセスが付与されるのを防止します。

期待される成果: 「知る必要」に基づき、認証されたユーザーのみがデータへアクセスできるようにします。定期的なバックアップとバージョニングでデータを保護し、意図しない、または不注意によるデータの改ざんや削除を防止します。重要なデータを他のデータから分離して、機密性とデータ整合性を保護します。

一般的なアンチパターン:

- 機密度要件と分類の異なるデータを一緒に保管する。
- 復号化キーに、過度に寛容なアクセス許可を使用する。
- データを不適切に分類する。
- 重要なデータの詳細なバックアップを保持しない。
- 本番データへの永続的なアクセスを提供する。
- データアクセスを監査することも、定期的にアクセス許可を審査することもしていない。

このベストプラクティスが確立されていない場合のリスクレベル: 低

実装のガイダンス

アクセス (最小特権を使用)、分離、バージョニングなど、複数のコントロールによって保管中のデータを保護できます。データへのアクセスは、AWS CloudTrail などの探査メカニズムと、Amazon Simple Storage Service (Amazon S3) アクセスログなどのサービスレベルログを使用して監査する必要があります。パブリックにアクセス可能なデータをインベントリし、時間の経過とともにパブリックで利用可能なデータ量の削減します。

Amazon S3 Glacier のボールトロックと Amazon S3 オブジェクトロックは、Amazon S3 のオブジェクトに対して必須のアクセス制御を提供します。ボールトポリシーがコンプライアンスオプションを使用してロックされると、ロックの有効期限が切れるまではルートユーザーでも変更できません。

実装手順

- アクセスコントロールを適用する: 暗号キーへのアクセスを含め、最小特権を用いたアクセスコントロールを適用します。
- さまざまな分類レベルに基づいてデータを分離する: データ分類レベルには異なる AWS アカウント を使用し、それらのアカウントの管理には AWS Organizations を使用します。
- AWS Key Management Service (AWS KMS) ポリシーをレビューする: AWS KMS ポリシーで付与されるアクセスのレベルを確認します。
- Amazon S3 バケットとオブジェクトアクセス許可をレビューする: S3 バケットのポリシーで付与されるアクセスのレベルを定期的に確認します。ベストプラクティスは、バケットを公開で読み取ったり書き込んだりできないようにすることです。AWS Config を使用して公開されているバケットを検出し、Amazon CloudFront を使用して Amazon S3 からコンテンツを提供することを検討します。パブリックアクセスを許可してはならないバケットが、パブリックアクセスを防ぐように正しく構成されていることを確認します。デフォルトでは、すべての S3 バケットはプライベートであり、明示的にアクセスが許可されたユーザーのみがアクセスできます。
- AWS IAM Access Analyzer を有効にする: IAM Access Analyzer は、Amazon S3 バケットを分析して、S3 ポリシーが外部エンティティにアクセスを付与した時点で検出結果を生成します。
- Amazon S3 バージョニングとオブジェクトロックを有効にします (該当する場合)。
- Amazon S3 インベントリを使用する: Amazon S3 インベントリは、S3 オブジェクトのレプリケーションと暗号化ステータスの監査とレポートに使用できます。
- Amazon EBS および AMI 共有アクセス許可をレビューする: 共有アクセス許可は、イメージとボリュームをワークロード外の AWS アカウント に共有することを可能にします。
- AWS Resource Access Manager Shares を定期的にレビューして、リソースを共有し続けるかどうかを決定します。 Resource Access Manager では、AWS Network Firewall ポリシー、Amazon Route 53 リゾルバールール、およびサブネットなど、Amazon VPC 内のリソースを共有できます。定期的に共有リソースを監査し、共有が不要になったリソースは共有を停止します。

リソース

関連するベストプラクティス:

- SEC03-BP01 アクセス要件を定義する (p. 35)
- SEC03-BP02 最小特権のアクセスを付与します (p. 36)

関連するドキュメント:

- AWS KMS Cryptographic Details Whitepaper (AWS KMS 暗号化の詳細についてのホワイトペーパー)
- Introduction to Managing Access Permissions to Your Amazon S3 Resources (Amazon S3 リソースへのアクセス許可の管理の導入)
- Overview of managing access to your AWS KMS resources (AWS KMS リソースへのアクセス管理の概要)
- AWS Config ルール
- Amazon S3 + Amazon CloudFront: A Match Made in the Cloud (理想的な組み合わせ)
- Using versioning (バージョニングの使用)
- Locking Objects Using Amazon S3 Object Lock (Amazon S3 Object Lock を使ってオブジェクトをロックする)
- Sharing an Amazon EBS Snapshot (Amazon EBS スナップショットの共有)
- 共有 AMI
- Hosting a single-page application on Amazon S3 (Amazon S3 でのシングルページアプリケーションのホスティング)

関連動画:

- Securing Your Block Storage on AWS (AWS でブロックストレージをセキュリティ保護する)

SEC08-BP05 人をデータから遠ざけるメカニズムを使用する

通常の運用状況で、すべてのユーザーが機密データおよびシステムに直接アクセスできないようにします。たとえば、変更管理ワークフローを使用して、直接アクセスや踏み台ホストを許可する代わりに、ツールを使用して Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを管理します。これは、タスクを実行する手順を含むオートメーションドキュメントを使用する AWS Systems Manager Automation をトリガーして を通じて 達成できます 。これらのドキュメントはソース管理に保存し、実行前にピアレビューを行い、シェルアクセスと比較してリスクを最小限に抑えるために徹底的にテストできます。ビジネスユーザーは、データストアに直接アクセスする代わりにダッシュボードを使用し、クエリを実行できます。CI/CD パイプラインを使用しない場合は、通常無効になっている特権アクセスメカニズムを適切に提供するために必要な制御とプロセスを決定します。

このベストプラクティスを活用しない場合のリスクレベル: 低

実装のガイダンス

- 人をデータから遠ざけるメカニズムを実装する: メカニズムには、Amazon QuickSight などのダッシュボードを使用して、直接クエリを実行する代わりにユーザーにデータを表示することが含まれます。
  - Amazon QuickSight
- 設定管理を自動化する: 設定管理サービスまたはツールを使用して、リモートでアクションを実行し、安全な設定を自動的に適用および検証します。踏み台ホストを使用したり、EC2 インスタンスに直接アクセスしたりすることを回避します。
  - AWS Systems Manager
  - AWS CloudFormation
  - AWS の AWS CloudFormation テンプレートの CI/CD パイプライン

リソース

関連するドキュメント:

- AWS KMS 暗号化の詳細についてのホワイトペーパー

関連動画:

- How Encryption Works in AWS
- Securing Your Block Storage on AWS

伝送中のデータの保護

送信中のデータ とは、システム間で送信されるすべてのデータを指します。これには、ワークロード内のリソース間での通信や他のサービスとエンドユーザーとの通信が含まれます。転送中のデータに適切なレベルの保護を提供することにより、ワークロードのデータの機密性と整合性を守ることができます。

VPC またはオンプレミスロケーション間でデータを安全に保護します。 専用のインフラストラクチャで AWS PrivateLink を使用して、Amazon Virtual Private Cloud (Amazon VPC) またはオンプレミス接続と AWS でホストされるサービスとの間にセキュアなプライベートネットワーク接続を作成します。AWS

サービス、サードパーティーサービス、および他の AWS アカウント のサービスを、あたかも自分のプライベートネットワークにあるかのように利用することができます。AWS PrivateLink を使うと、インターネットゲートウェイまたは NAT を使わずに、IP CIDR が重複するアカウント間のサービスにアクセスすることができます。また、ファイアウォールルール、パス定義、またはルートテーブルを設定する必要もありません。トラフィックは Amazon のバックボーンにとどまり、インターネットを横断しないため、お客様のデータは保護されます。HIPAA および EU/US プライバシーシールドなどの業界特有のコンプライアンス規制に対する準拠を維持できます。AWS PrivateLink はサードパーティーソリューションとシームレスに連携し、簡素化されたグローバルネットワークを作成するため、クラウドへの移行を加速させて利用可能な AWS のサービスを活用できます。

ベストプラクティス

- SEC09-BP01 安全な鍵および証明書管理を実装する (p. 93)
- SEC09-BP02 伝送中に暗号化を適用する (p. 95)
- SEC09-BP03 意図しないデータアクセスの検出を自動化する (p. 97)
- SEC09-BP04 ネットワーク通信を認証する: (p. 97)

SEC09-BP01 安全な鍵および証明書管理を実装する

Transport Layer Security (TLS) 証明書は、ネットワーク通信を保護し、インターネットやプライベートネットワーク上のウェブサイト、リソース、ワークロードの ID を確立するために使用されます。

期待される結果: 公開鍵基盤 (PKI) で証明書をプロビジョニング、デプロイ、保存、更新できる、安全な証明書管理システム。安全な鍵と証明書の管理メカニズムは、証明書のプライベートキーの内容が漏洩するのを防ぎ、自動的に証明書の定期更新を行い、他のサービスと統合して、ワークロード内のマシンリソースに安全なネットワーク通信と ID を提供します。キーの内容は、決して人的 ID にアクセス可能なものであってはなりません。

一般的なアンチパターン:

- 証明書のデプロイまたは更新プロセス中に手動で手順を実行する。
- プライベート認証機関 (CA) を設計する際、CA 階層に十分な注意を払わない。
- 公共リソースに自己署名証明書を使用する。

このベストプラクティスを確立するメリット:

- 自動デプロイと自動更新により証明書管理を簡素化する
- TLS 証明書を使用して転送中のデータの暗号化を奨励する
- 認証機関による証明書アクションのセキュリティと可監査性を向上させる
- CA 階層のさまざまなレイヤーにおける管理業務を整理する

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

最新のワークロードでは、Transport Layer Security (TLS) などの公開鍵基盤 (PKI) プロトコルを使用して暗号化されたネットワーク通信が広く利用されています。PKI 証明書の管理は複雑になる場合がありますが、証明書のプロビジョニング、デプロイ、更新を自動化することで、証明書管理に伴う手間を軽減できます。

AWS は、汎用 PKI 証明書を管理する 2 つのサービス、AWS Certificate Manager と AWS Private Certificate Authority (AWS Private CA) を提供しています。AWS Certificate Manager は、パブリックとプライベートの AWS ワークロードの両方で使用するための証明書のプロビジョニング、管理、およびデプロイに使用できる主要なサービスです。AWS Certificate Manager は AWS のパブリック認証機関を使用し

て証明書を発行し、他の多くの AWS マネージドサービスと統合して、ワークロード用の安全な TLS 証明書を提供します。

AWS Private Certificate Authority (プライベート CA) では、独自のルート認証機関または下位認証機関を確立し、API を通じて TLS 証明書を発行できます。こうした種類の証明書は、TLS 接続のクライアント側で信頼チェーンを制御し管理するシナリオで使用できます。TLS ユースケースに加えて、AWS Private CA は、Kubernetes ポッドへの証明書の発行、Matter デバイス製品認証、コード署名、およびカスタムテンプレートを使用したその他のユースケースにも使用できます。IAM Roles Anywhere を使用して、プライベート CA によって署名された X.509 証明書が発行されたオンプレミスのワークロードに、一時的な IAM 認証情報を提供することもできます。

AWS Certificate Manager と AWS Private Certificate Authority の他、AWS IoT Core は、モノのインターネット (IoT) デバイスへの PKI 証明書のプロビジョニング、管理、およびデプロイに特化したサポートを提供します。AWS IoT Core は、公開鍵基盤に大規模に IoT デバイスをオンボーディングするための特殊なメカニズムを提供します。

プライベート CA 階層を確立する際の考慮事項

プライベート CA を確立する必要がある場合、特別な注意を払って事前に CA 階層を適切に設計しておくことが重要です。プライベート CA 階層を作成する場合は、CA 階層の各レベルを個別の AWS アカウント にデプロイすることがベストプラクティスです。この意図的な手順により、CA 階層内の各レベルへの外部からのアクセスが減り、CloudTrail ログデータ内の異常をより簡単に発見できるようになります。また、いずれかのアカウントに不正アクセスがあった場合、アクセス範囲と影響が小さくなります。ルート CA はそれぞれ別のアカウントに保存し、1 件以上の中間 CA 証明書の発行にのみ使用すべきです。

次に、ルート CA のアカウントとは別のアカウントに 1 つ以上の中間 CA を作成し、エンドユーザー、デバイス、または他のワークロードに証明書を発行します。最後に、ルート CA から中間 CA に証明書を発行します。これにより、エンドユーザーまたはデバイスに証明書が発行されます。回復力の計画、クロスリージョンレプリケーション、組織全体での CA の共有など、CA デプロイの計画と CA 階層の設計の詳細については、Planning your AWS Private CA deployment を参照してください。

実装手順

1. ユースケースに必要となる適切な AWS サービスを判断します。
   - 多くのユースケースでは、AWS Certificate Manager を使用して AWS の既存の公開鍵基盤を活用できます。AWS Certificate Manager は、ウェブサーバー、ロードバランサー、または公的に信頼されている証明書で AWS のマネージド認証機関を利用できるその他の用途に TLS 証明書をデプロイするために使用できます。
   - 独自のプライベート認証機関階層を確立する必要がある場合や、エクスポート可能な証明書へのアクセスが必要な場合は、AWS Private Certificate Authority を検討してください。これにより、AWS Certificate Manager を、AWS Private Certificate Authority を使用する さまざまな種類のエンドエンティティ証明書の発行に使用できます。
   - 組み込み型モノのインターネット (IoT) デバイス向けに、証明書を大規模にプロビジョニングする必要があるユースケースについては、AWS IoT Core を検討してください。
2. 可能な限り、証明書の自動更新を実装してください。
   - ベストプラクティスは、AWS Certificate Manager が発行した証明書にAWS Certificate Manager のマネージド型更新と、統合された AWS のマネージドサービスを使用することです。
3. ログ記録と監査証跡を確立します。
   - 認証機関を保有するアカウントへのアクセスを追跡するための CloudTrail ログを有効にします。CloudTrail でログファイルの整合性検証を設定して、ログデータの信頼性を検証することを検討してください。
   - プライベート CA が発行または取り消した証明書を一覧表示する監査レポートを定期的に生成します。これらのレポートは S3 バケットにエクスポートできます。
   - プライベート CA をデプロイするときは、証明書失効リスト (CRL) を保存する S3 バケットも確立する必要があります。ワークロードの要件に基づいてこの S3 バケットを設定するガイダンスについては、Planning a certificate revocation list (CRL) を参照してください。

リソース

関連する Well-Architected のベストプラクティス

- SEC02-BP02 一時的な認証情報を使用する (p. 22)
- SEC08-BP01 安全なキー管理を実装する (p. 85)
- SEC09-BP04 ネットワーク通信を認証する: (p. 97)

関連するドキュメント:

- How to host and manage an entire private certificate infrastructure in AWS
- How to secure an enterprise scale ACM Private CA hierarchy for automotive and manufacturing
- Private CA best practices
- How to use AWS RAM to share your ACM Private CA cross-account

関連動画:

- Activating AWS Certificate Manager Private CA (ワークショップ)

関連する例:

- Private CA workshop
- IOT Device Management Workshop (デバイスプロビジョニングを含む)

関連ツール:

- Plugin to Kubernetes cert-manager to use AWS Private CA

SEC09-BP02 伝送中に暗号化を適用する

組織的、法的、コンプライアンス要件を満たすための組織のポリシー、法的義務と標準に基づいて、定義された暗号化要件を適用します。機密データを仮想プライベートクラウド (VPC) の外部に送信する場合は、暗号化されたプロトコルのみを使用します。暗号化を行うと、データが信頼できないネットワークを伝送中も、データの機密性を保持できます。

期待される成果: すべてのデータは、安全な TLS プロトコルと暗号スイートを使用して伝送中に暗号化する必要があります。データへの不正なアクセスを軽減するためには、リソースとインターネット間のネットワークトラフィックを暗号化する必要があります。内部 AWS 環境内にのみあるネットワークトラフィックは、可能な場合に TLS を使って暗号化する必要があります。AWS 内部ネットワークはデフォルトで暗号化され、VPC 内のネットワークトラフィックは、トラフィック (Amazon EC2 インスタンス、Amazon ECS コンテナなど) を生成しているリソースに権限のない人がアクセスしない限り、なりすましや盗聴を行うことはできませんIPsec 仮想プライベートネットワーク (VPN) を使ってネットワーク間のトラフィックを保護することを検討してください。

一般的なアンチパターン:

- 廃止されたバージョンの SSL、TLS、および暗号スイートコンポーネント (SSL v3.0、1024-bit RSA キー、および RC4 暗号) を使用する。
- パブリック向けリソースとの間で暗号化されていない (HTTP) トラフィックを許可する。
- X.509 証明書をモニタリングし、期限が切れる前に交換しない。
- TLS に自己署名 X.509 証明書を使用する。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

AWS のサービスには、通信に TLS を使用し、AWS API との通信の際に伝送中データの暗号化を利用できる、HTTPS エンドポイントが用意されています。HTTP など安全でないプロトコルは、セキュリティグループを使用して VPC で監査およびブロックできます。HTTP リクエストは、Amazon CloudFront または Application Load Balancer でHTTPS に自動的にリダイレクトすることもできます。コンピューティングリソースを完全に制御して、サービス全体に伝送中データの暗号化を実装できます。また、外部ネットワークまたは AWS Direct Connect からお使いの VPC に VPN で接続して、トラフィックの暗号化を促進できます。クライアントが AWS API に電話かける際に、最低でも TLS 1.2 を使用していることを確認してください。AWS は、2023 年 6 月に TLS 1.0 と 1.1 の使用を廃止予定です。特別な要件がある場合は、AWS Marketplace でサードパーティーのソリューションを入手できます。

実装手順

- 伝送中に暗号化を適用する: 暗号化の要件は、最新の標準とベストプラクティスに基づき、安全なプロトコルのみを許可する必要があります。たとえば、Application Load Balancer または Amazon EC2 インスタンスに対してのみ HTTPS プロトコルを許可するよう、セキュリティグループを設定します。
- エッジサービスで安全なプロトコルを設定する: HTTPS を Amazon CloudFront と設定して、自分のセキュリティ体制やユースケースに適した セキュリティプロファイルを使用します。
- 外部接続に VPN を使用する: ポイントツーポイント接続やネットワーク間接続を IPsec VPN で保護し、データのプライバシーと整合性の両方を提供することを検討してください。
- ロードバランサーで安全なプロトコルを設定する: リスナーに接続し、クライアントがサポートするなかで最強の暗号スイートを提供するセキュリティポリシーを選択します。Application Load Balancer に HTTPS リスナーを作成します。
- Amazon Redshift で安全なプロトコルを設定する: クラスターで Secure Socket Layer (SSL) または Transport Layer Security (TLS) 接続が必要となるよう設定します。
- 安全なプロトコルを設定する: AWS サービスのドキュメントをレビューして、転送時の暗号化機能を決定します。
- Amazon S3 バケットにアップロードする際、安全なアクセスを設定する: Amazon S3 バケットポリシーコントロールを使用して、データに対して安全なアクセスを適用します。
- AWS Certificate Managerの使用を検討する: ACM では、AWS サービスで使用するためのパブリック TLS 証明書をプロビジョニング、管理、およびデプロイできます。
- プライベート PKI ニーズに対してAWS Private Certificate Authority の使用を検討する: AWS Private CA では、プライベート認証局 (CA) 階層を作成し、暗号化された TLS チャネルの作成に使用できるエンドエンティティ X.509 証明書を発行することができます。

リソース

関連するドキュメント:

- AWS ドキュメント
- Using HTTPS with CloudFront (CloudFront で HTTPS を使う)
- Connect your VPC to remote networks using AWS Virtual Private Network (AWS Virtual Private Network を使用して VPC をリモートネットワークに接続する)
- Create an HTTPS listener for your Application Load Balancer (アプリケーションロードバランサーの HTTPS リスナーを作成する)
- チュートリアル: Amazon Linux 2 で SSL/TLS を設定する
- Using SSL/TLS to encrypt a connection to a DB instance (SSL/TLS を使用した DB インスタンスへの接続の暗号化)

- Configuring security options for connections (接続のセキュリティオプションを設定する)

SEC09-BP03 意図しないデータアクセスの検出を自動化する

Amazon GuardDuty などのツールを使用して、疑わしい活動や定義された境界外にデータを移動させようとする試みを自動的に検出します。例えば、GuardDuty は Amazon Simple Storage Service (Amazon S3) 読み取りアクティビティを検出できますが、それには Exfiltration:S3/AnomalousBehavior 調査結果を使用します.GuardDuty に加えて、ネットワークトラフィック情報をキャプチャする Amazon VPC フローログを Amazon EventBridge とともに使用して、異常な接続 (成功と拒否の両方) の検出をトリガーできます。Amazon S3 Access Analyzer は Amazon S3 バケット内で誰がどのデータにアクセス可能かを評価するのに役立ちます。

このベストプラクティスを活用しない場合のリスクレベル: ミディアム

実装のガイダンス

- 意図しないデータアクセスの検出を自動化する: ツールまたは検出メカニズムを使用し、定義された境界の外側にデータを移動する試みを自動的に検出します。例えば、認識できないホストにデータをコピーしているデータベースシステムを検出します。
  - VPC フローログ
- Amazon Macie を検討する: Amazon Macie は、機械学習とパターンマッチングを使用して AWS の機密データを検出および保護する、フルマネージドのデータセキュリティおよびデータプライバシーサービスです。
  - Amazon Macie

リソース

関連ドキュメント:

- VPC フローログ
- Amazon Macie

SEC09-BP04 ネットワーク通信を認証する:

Transport Layer Security (TLS) や IPsec など、認証をサポートするプロトコルを使用して、通信の ID を検証します。

サービス間、アプリケーション間、またはユーザーへの通信には常に、安全で認証済みのネットワークプロトコルを使用するようにワークロードを設計してください。認証と承認をサポートするネットワークプロトコルを使用すれば、ネットワークフローの制御を強化し、不正アクセスによる影響を軽減できます。

期待される成果: サービス間のデータプレーンとコントロールプレーンのトラフィックフローがワークロードで明確に定義されている。技術上可能な場合は必ず、認証および暗号化されたネットワークプロトコルをトラフィックフローが使用する。

一般的なアンチパターン:

- ワークロード内のトラフィックフローが暗号化されていない、または認証されていない。
- 複数のユーザーやエンティティで認証情報を再利用している。
- アクセス制御のメカニズムとしてネットワーク統制にばかり依存している。

- 業界標準の認証メカニズムに頼る代わりに、カスタムの認証メカニズムを作成する。
- VPC 内のサービスコンポーネントや他のリソース間のトラフィックフローが必要以上に許可されている。

このベストプラクティスを活用するメリット:

- 不正アクセスによる影響が及ぶ範囲をワークロードの一部に制限します。
- アシュアランスのレベルを上げ、認証済みのエンティティだけがアクションを実行するように徹底します。
- 導入予定のデータ転送インターフェイスを明確に定義し、実際に導入して、サービスの分離を強化します。
- リクエストのアトリビューションと、明確に定義された通信インターフェイスにより、モニタリング、ログ記録、インシデント対応を強化します。
- ネットワーク統制に認証と承認の統制を組み合わせて、ワークロードの多層防御を実現します。

このベストプラクティスが確立されていない場合のリスクレベル: 低

実装のガイダンス

ワークロードのネットワークトラフィックのパターンは、次の 2 つのカテゴリに分類できます。

- East-West トラフィックは、ワークロードを構成するサービス間のトラフィックフローを表します。
- North-South トラフィックは、ワークロードとコンシューマー間のトラフィックフローを表します。

一般的には North-South トラフィックを暗号化し、認証済みプロトコルを用いて East-West トラフィックを保護する例はあまり見られません。最近のセキュリティ対策では、ネットワークの設計だけで、2 つのエンティティ間に信頼関係があるとは想定しないというのが通例となっています。2 つのサービスが共通のネットワーク境界の中にある場合でも、サービス間の通信を暗号化、認証、承認することがベストプラクティスです。

一例として、AWS サービス API は AWS Signature Version 4 (Sigv4) 署名プロトコルを使用して、リクエストの発信元のネットワークに関係なく、呼び出し元を認証します。この認証を通じて AWS API はアクションの要求元の ID を確認することができ、その ID を承認決定のポリシーと組み合わせて、アクションを許可するかどうかを判断できます。

Amazon VPC Lattice や Amazon API Gateway などのサービスでは、同じ SigV4 署名プロトコルを使用して、独自のワークロードの East-West トラフィックに認証と承認を追加できます。AWS 環境の外のリソースが、SigV4 ベースの認証と承認を必要とするサービスと通信する必要がある場合は、その非 AWS リソースで AWS Identity and Access Management (IAM) Roles Anywhere を使用して、一時的な AWS 認証情報を取得できます。この認証情報を使用して、アクセス権の承認に SigV4 を使用するサービスへのリクエストに署名できます。

East-West トラフィックを認証するメカニズムとしては、TLS 相互認証 (mTLS) も一般的です。モノのインターネット (IoT)、ビジネス間 (B2B) アプリケーション、マイクロサービスの多くは、mTLS を採用しています。TLS 通信のクライアント側とサーバー側の両方が X.509 証明書を使用して、双方のアイデンティティを認証し合います。これらの証明書は AWS Private Certificate Authority (AWS Private CA) で発行できます。Amazon API Gateway や AWS App Mesh などのサービスを使用して、ワークロード間またはワークロード内の通信で mTLS 認証を行うことができます。mTLS は TLS 通信の両側に認証情報を提供しますが、承認のメカニズムは提供しません。

最後に、OAuth 2.0 と OpenID Connect (OIDC) の 2 つのプロトコルは、ユーザーがサービスへのアクセスを制御する際に一般的に使用されていますが、最近ではサービス間トラフィックでもよく利用されています。API Gateway の JSON ウェブトークン (JWT) オーソライザーを使用すると、OIDC または OAuth 2.0 の ID プロバイダーが発行した JWT を使用して、API ルートへのアクセスをワークロードで制限できます。OAuth2 のスコープを基本的な承認決定のソースとして使用できますが、依然として承認チェック

をアプリケーション層に実装する必要があります。OAuth2 スコープ単体で複雑な承認ニーズに対応することはできません。

実装手順

- ワークロードのネットワークフローを定義および文書化する: 多層防御戦略を実装するためには、まず、ワークロードのトラフィックフローを定義します。
  - ワークロードを構成するさまざまなサービス間でデータがどのように転送されるかを明確に定義したデータフロー図を作成します。これらのフローを認証済みのネットワークチャネルに実際に流していく前に、まずこの図を用意します。
  - 開発段階とテスト段階でワークロードを計測して、ランタイム時のワークロードの動作がデータフロー図に正確に反映されていることを確認してください。
  - データフロー図は、脅威モデリングを行う (「SEC01-BP07 脅威モデルを使用して脅威を特定し、緩和策の優先順位を付ける」を参照) ときにも役立ちます。
- ネットワーク統制を確立する: AWS の機能を使用して、データフローに応じたネットワーク統制を確立することを検討してください。ネットワーク境界は、それだけでは十分なセキュリティ統制にはなりませんが、ワークロードを保護する多層防御戦略の 1 層にはなります。
  - セキュリティグループを使用して、リソース間のデータフローを確立、定義、制限します。
  - AWS サービスとサードパーティーの AWS PrivateLink 対応サービスの両方との通信に、AWS PrivateLink を使用することを検討してください。AWS PrivateLink インターフェイスエンドポイントを介して送信されるデータは、AWS ネットワークバックボーン内にとどまり、公開インターネットを経由しません。
- ワークロードのサービス全体に認証と承認を実装する: ワークロードのトラフィックフローを認証および暗号化するために最適な一連の AWS サービスを選択してください。
  - Amazon VPC Lattice でサービス間通信のセキュリティを確保することを検討してください。VPC Lattice では、SigV4 認証を認証ポリシーと組み合わせて使用して、サービス間のアクセスを制御できます。
  - mTLS を使用するサービス間通信では、API Gateway または App Mesh を検討してください。AWS Private CA を使用して、mTLS で使用する証明書を発行可能なプライベート CA 階層を確立できます。
  - OAuth 2.0 または OIDC を使用するサービスと統合する場合は、API Gateway で JWT オーソライザーを使用することを検討してください。
  - ワークロードと IoT デバイス間の通信については、AWS IoT Core を検討してください。IoT Core は、ネットワークトラフィックの暗号化と認証のオプションをいくつか提供します。
- 不正アクセスを監視する: 意図しない通信チャネル、保護されたリソースにアクセスしようとする非承認のプリンシパル、その他の不適切なアクセスパターンを継続的に監視します。
  - サービスへのアクセスの管理に VPC Lattice を使用する場合は、VPC Lattice アクセスログを有効にし、監視することを検討してください。これらのアクセスログには、リクエスト元のエンティティに関する情報、ソースとターゲットの VPC などのネットワーク情報、リクエストのメタデータが記録されています。
  - VPC フローログを有効にして、ネットワークフローのメタデータをキャプチャし、異常がないか定期的に確認することを検討してください。
  - セキュリティインシデントの計画、シミュレーション、対応に関する詳細なガイダンスについては、「AWS セキュリティインシデント対応ガイド」および「AWS Well-Architected フレームワーク」の「セキュリティの柱」の「インシデント対応」セクションを参照してください。

リソース

関連するベストプラクティス:

- SEC03-BP07 パブリックおよびクロスアカウントアクセスの分析
- SEC02-BP02 一時的な認証情報を使用する

- SEC01-BP07 脅威モデルを使用して脅威を特定し、緩和策の優先順位を付ける

関連するドキュメント:

- Evaluating access control methods to secure Amazon API Gateway APIs
- REST API の相互 TLS 認証の設定
- How to secure API Gateway HTTP endpoints with JWT authorizer
- AWS IoT Core 認証情報プロバイダーを使用して、AWS サービスへの直接呼び出しを認証
- AWS セキュリティインシデント対応ガイド

関連動画:

- AWS re:invent 2022: Introducing VPC Lattice
- AWS re:invent 2020: Serverless API authentication for HTTP APIs on AWS

関連する例:

- Amazon VPC Lattice Workshop
- Zero-Trust Episode 1 – The Phantom Service Perimeter workshop

インシデント対応

成熟した予防的、発見的統制が実装されていても、組織はセキュリティインシデントの潜在的な影響に対応し、影響を緩和するメカニズムを実装する必要があります。準備することで、インシデントの際にチームが効果的に動作し、問題を切り分け、封じ込め、フォレンジックを実行し、運用を既知の正常な状態に復元する能力に強く影響します。セキュリティインシデントが起こる前にツールとアクセス権を整備し、ゲームデー (実践訓練) を通じてインシデント対応を定期的に実施しておけば、ビジネスの中断を最小限に抑えながら復旧することができます。

トピック

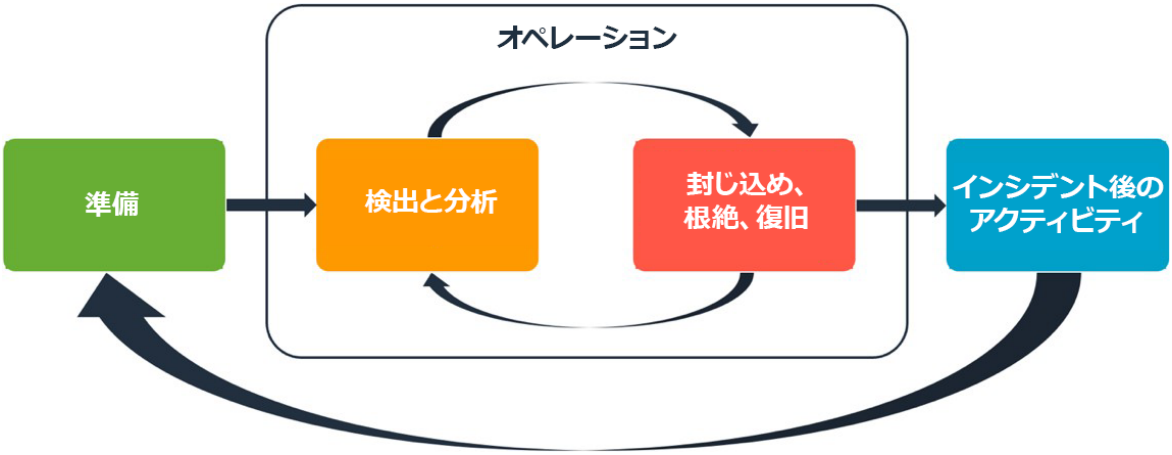
AWS におけるインシデント対応の諸側面

組織内のすべての AWS ユーザーは、セキュリティインシデント対応プロセスの基本を理解している必要があり、セキュリティ担当者はセキュリティ問題への対応方法を理解している必要があります。教育、トレーニング、経験は、クラウドインシデント対応プログラムを成功させるために不可欠であり、起こり得るセキュリティインシデントに対処する前に十分な余裕を持って実施するのが理想的です。クラウドでのインシデント対応プログラムの成功の基盤は、 準備、 オペレーション、 インシデント後のアクティビティです。

これらの各側面を理解するには、以下の説明を参考にしてください。

- 準備: 検出制御を有効にし、必要なツールやクラウドサービスへの適切なアクセスを検証することで、インシデント対応チームが AWS 内のインシデントを検出して対応できるように準備します。さらに、信頼性の高い一貫した応答を検証するために、手動と自動の両方で必要なプレイブックを準備します。
- オペレーション: NIST のインシデント対応フェーズ (検出、分析、封じ込め、根絶、復旧) に従って、セキュリティイベントと潜在的なインシデントに対処します。
- インシデント後のアクティビティ: セキュリティイベントとシミュレーションの結果を反復することで、対応の有効性を改善し、対応と調査から得られる価値を高め、リスクをさらに軽減します。インシデントから学び、改善活動に対する強いオーナーシップを持つ必要があります。

下図は、前述の NIST のインシデント対応ライフサイクルに沿った、これらの側面のフローを示しています。ここでの業務には、検出と分析に加えて、封じ込め、根絶、復旧が含まれています。



AWS におけるインシデント対応の諸側面

クラウドレスポンスの設計目標

ただし、NIST SP 800-61 Computer Security Incident Handling Guide などで定義されているインシデント対応の一般的なプロセスと メカニズムは、引き続き正しいものですが、クラウド環境におけるセキュリティインシデントへの対応に関連する、以下の具体的な設計目標を評価することをお勧めします。

- 対応目標の確立: ステークホルダー、法律顧問、組織のリーダーと協力してインシデント対応の目標を決定します。共通の目標には、問題の封じ込めと緩和、影響を受けたリソースの復旧、フォレンジック用のデータの保全、既知の安全な運用への復帰、そして最終的にはインシデントからの学習などがあります。
- クラウドを使用して応答する: イベントとデータが発生するクラウド内に応答パターンを実装します。
- 持っているものと必要なものを知る: ログ、リソース、スナップショット、その他の証拠は、対応専用の一元化されたクラウドアカウントにコピーして保存します。管理ポリシーを適用するタグ、メタデータ、メカニズムを使用します。使用しているサービスを把握し、それらのサービスを調査するための要件を特定する必要があります。環境を把握しやすくするために、タグ付けを使用することもできます。
- 再デプロイメカニズムを使用する: セキュリティの異常が設定ミスに起因する場合は、適切な設定でリソースを再デプロイして差異を取り除くだけで解決できる場合があります。セキュリティ侵害の可能性が見つかった場合は、根本原因に対する適切で検証済みの緩和策が再デプロイに含まれていることを確認します。
- 可能な場合は自動化する: 問題が発生したり、インシデントが繰り返されたりした場合は、一般的なイベントをプログラムで優先順位付けして対応するメカニズムを構築します。自動化が不十分で、特殊かつ複雑、または機密性の高いインシデントには、人手で対応します。
- スケーラブルなソリューションを選択する: 組織のアプローチのスケーラビリティがクラウドコンピューティングと適合しているように努めます。環境全体にスケールできる検出および対応のメカニズムを実装して、検出から対応までの時間を効果的に短縮します。
- プロセスを学び、改善する: プロセス、ツール、人員におけるギャップを積極的に特定し、それらを修正する計画を実施します。シミュレーションは、ギャップを見つけてプロセスを改善する安全な方法です。

これらの設計目標は、インシデント対応と脅威検知の両方を実施する能力について、アーキテクチャの実装を確認することを促すものです。クラウドの実装を計画するときは、インシデントへの対応を検討します。フォレンジックに基づいた対応方法論を使用するのが理想的です。これは、場合によっては、このような対応タスク用に複数の組織、アカウント、ツールを特別に設定することを意味します。これらのツールと機能は、デプロイパイプラインによってインシデント対応担当者が利用できるようにする必要があります。リスクを大きくする可能性があるため、静的な状態のままにしないでください。

準備

インシデントへの準備は、タイムリーかつ効果的なインシデント対応にとって重要です。準備は次の 3 つのドメインにわたって行われます。

- 人材: セキュリティインシデントに備えて人員を準備するには、インシデント対応に関連するステークホルダーを特定し、インシデント対応とクラウド技術に関するトレーニングを行う必要があります。
- プロセス: セキュリティインシデントに備えてプロセスを準備するには、アーキテクチャの文書化、徹底的なインシデント対応計画の策定、セキュリティイベントへの一貫した対応のためのプレイブックの作成が必要です。
- テクノロジー: セキュリティインシデントに備えてテクノロジーを準備するには、アクセスの設定、必要なログの集約と監視、効果的なアラートメカニズムの実装、対応と調査機能の開発が必要です。

これらの各分野は、効果的なインシデント対応にとって等しく重要です。3 つすべてが揃わなければ、インシデント対応プログラムは完全でも効果的でもありません。インシデントに備えるには、人員、プロセス、テクノロジーを緊密に連携して準備する必要があります。

ベストプラクティス

- SEC10-BP01 重要な人員と外部リソースを特定する: (p. 103)
- SEC10-BP02 インシデント管理計画を作成する (p. 104)
- SEC10-BP03 フォレンジック機能を備える: (p. 106)
- SEC10-BP04 セキュリティインシデント対応プレイブックを作成し、テストする (p. 108)
- SEC10-BP05 アクセスを事前プロビジョニングする (p. 110)
- SEC10-BP06 ツールを事前デプロイする (p. 112)
- SEC10-BP07 シミュレーション行う (p. 114)

SEC10-BP01 重要な人員と外部リソースを特定する:

組織がインシデントに対応するのに役立てるため、社内外の担当者、リソース、法的義務を特定します。

クラウド上でのインシデントレスポンスへのアプローチを他のチーム (顧問弁護士、経営陣、ビジネスステークホルダー、AWS サポートサービスなど) と連携して定義する場合、重要な人物、ステークホルダー、関連する担当者を特定する必要があります。依存性を減らし、応答時間を短縮するために、チームや専門のセキュリティチーム、応答者が利用するサービスについて教育を受け、実践的な演習をする機会を持つようにしてください。

対応能力を強化するために、外部の専門知識および異なる視点を備えた社外の AWS セキュリティパートナーを探しておくことをお勧めします。信頼できるセキュリティパートナーは、馴染みのない潜在的なリスクや脅威を特定するのに役立ちます。

このベストプラクティスを活用しない場合のリスクレベル: 高

実装のガイダンス

- 組織内の主要な人員を特定する: インシデント対応と復旧に必要な組織内の人員の連絡先リストを保持します。
- 外部パートナーを特定する: 必要に応じて、インシデント対応と復旧を支援できる外部パートナーと連携します。

リソース

関連するドキュメント:

- AWS インシデント対応ガイド

関連動画:

- AWS 環境のセキュリティインシデントの準備と対応

関連する例:

SEC10-BP02 インシデント管理計画を作成する

インシデント対応のために最初に作成する文書は、インシデント対応計画です。インシデント対応計画は、インシデント対応プログラムと戦略の基礎となるように設計されています。

このベストプラクティスを活用するメリット: インシデント対応のプロセスを熟考し、明確に定義することは、インシデント対応プログラムを成功させ、拡張性を持たせるための鍵となります。セキュリティイベントが発生した場合、明確な手順とワークフローがあれば、タイムリーに対応できます。既にインシデント対応プロセスがある場合もあります。現在の状態にかかわらず、インシデント対応プロセスを定期的に更新、反復、テストすることが重要です。

このベストプラクティスを活用しない場合のリスクレベル: 高

実装のガイダンス

インシデント管理計画は、セキュリティインシデントの潜在的な影響への対応、復旧、軽減に不可欠です。インシデント管理計画は、セキュリティインシデントをタイムリーに特定し、修復、対応するための体系的なプロセスです。

クラウドには、オンプレミス環境と同じオペレーション上のロールと要件があります。インシデント管理計画を作成する際は、ビジネス成果とコンプライアンス要件と最も合致する対応および復旧戦略を組み込むことが重要です。例えば、米国の FedRAMP 準拠のワークロードを AWS で運用している場合は、『NIST SP 800-61 Computer Security Handling Guide』を遵守することが役に立ちます。同様に、ヨーロッパの個人を特定できる情報 (PII) データを含むワークロードを運用している場合は、EU 一般データ保護規則 (GDPR) で義務付けられているようにデータレジデンシー関連の問題に対してどのように防御、対応するかといったシナリオを考慮します。

AWS で運用するワークロードについてインシデント管理計画を策定する際は、AWS 責任共有モデルから始め、インシデント対応に向けた多層防御アプローチを構築することを目指します。このモデルでは、AWS はクラウドのセキュリティを管理します。クラウド内のセキュリティについてはお客様の責任です。つまり、お客様はコントロールを保持するとともに、実装しようとするセキュリティコントロールに責任を持つということです。『AWS Security Incident Response Guide』(AWS セキュリティインシデント対応ガイド) には、クラウド中心のインシデント管理計画を策定するための重要なコンセプトと基本的なガイダンスが記載されています。

効果的なインシデント管理計画は、クラウド運用の目標に沿って継続的に繰り返し、最新の状態に保つ必要があります。インシデント管理計画を作成して進化させるにあたり、以下に記載の実装計画を使用することを検討してください。

実装手順

役割と責任の定義

セキュリティイベントに対処するためには、組織横断的な規律と行動力が必要です。組織内には、人事 (HR)、経営陣、法務部など、インシデント発生時に責任、説明責任、相談、情報提供の役割を持つ担当者が多くいるはずです。これらの役割と責任、および第三者が関与する必要があるかどうかを検討してください。多くの地域には、義務や禁止事項を規定する現地の法律があることに注意してください。セキュリティ対応計画のために責任、説明責任、相談、情報提供 (RACI) チャートを作成するのはマニュアル的に思われるかもしれませんが、作成することで、迅速かつ直接的なコミュニケーションを促進し、イベントのさまざまな段階のリーダーシップを明確に説明できます。

インシデントが発生した場合、影響の測定に役立つ情報や背景を提供できる対象分野のエキスパート (SME) である、影響を受けるアプリケーションやリソースの所有者と開発者を巻き込むことが重要です。インシデント対応について開発者やアプリケーション所有者の専門知識に頼る際は、事前にやり取りを行い、関係を構築してください。アプリケーション所有者や SME (クラウド管理者やエンジニアなど) は、不慣れまたは複雑な環境、対応者がアクセスできない状況下で対応することが必要な場合もあります。

最後に、信頼できるパートナーは、さらなる専門知識や価値のある調査を提供できるため、調査や対応に関与する可能性があります。自分のチームにこれらのスキルがない場合は、外部の人材に支援を依頼するということも検討できます。

AWS 対応チームとサポートを理解する

- AWS Support
  - AWS Support には、AWS ソリューションの成功と運用上の健全性をサポートするツールや専門知識を利用できるさまざまなプランが用意されています。AWS 環境の計画、導入、最適化に役立つテクニカルサポートや、より多くのリソースが必要な場合は、AWS ユースケースに最適なサポートプランを選択できます。
  - お客様 の AWS リソースに影響を与える問題についてのサポートを受けるには、AWS Management Console (サインインが必要) のサポートセンターが中心的な窓口になります。AWS Support へのアクセスは AWS Identity and Access Management によって制御されます。AWS Support 機能へのアクセスの詳細については、 Getting started with AWS Supportを参照してください。
- AWS カスタマーインシデント対応チーム (CIRT)
  - AWS カスタマーインシデント対応チーム (CIRT) は、24 時間 365 日の体制で専門のグローバル AWS チームであり、 AWS 責任共有モデルのカスタマーサイドでのアクティブなセキュリティイベント中にカスタマーをサポートします。
  - AWS CIRT がお客様をサポートすると、AWS で発生しているセキュリティイベントの優先順位付けと復旧を支援します。AWS サービスログを使用して根本原因の分析を支援し、復旧のための推奨事項を提示します。また、将来のセキュリティイベントを回避するのに役立つセキュリティに関する推奨事項やベストプラクティスを提供することもできます。
  - AWS のお客様は、AWS CIRT と AWS Support ケースで連携できます。
- DDoS 対応のサポート
  - AWS オファー AWS Shieldは、AWS で実行中のウェブアプリケーションを保護するマネージドで分散型のサービス拒否 (DDoS) 保護サービスを提供します。Shield は、常時検出と自動インライン緩和により、アプリケーションのダウンタイムとレイテンシーを最小限に抑えることができるため、DDoS 保護のメリットを得るために AWS Support と連携する必要はありません。Shield のティアには、AWS Shield Standard および AWS Shield Advanced の 2 つのティアがあります。これら 2 つのティアの違いについては、 Shield 機能のドキュメントを参照してください。
- AWS Managed Services (AMS)
  - AWS Managed Services (AMS) は AWS インフラストラクチャ管理を継続的に提供するため、お客様はアプリケーションに集中できます。インフラストラクチャを維持するためのベストプラクティスを実行することで、AMS によって運用上のオーバーヘッドとリスクの軽減を支援します。AMS は、変更リクエスト、モニタリング、パッチ管理、セキュリティ、バックアップサービスなどの一般的なアクティビティを自動化し、インフラストラクチャをプロビジョニング、実行、サポートするためにライフサイクル全体にわたるサービスを利用できます。
  - AMS は、一連のセキュリティ検出コントロールの展開に責任を持ち、24 時間 365 日、第一線でアラートに対応します。アラートが発生すると、AMS は標準的な自動プレイブックと手動プレイブックに従って、一貫した対応が行われていることを確認します。これらのプレイブックは、オンボーディング中に AMS のお客様と共有され、AMS との対応策を練り、調整することができます。

インシデント対応計画の策定

インシデント対応計画は、インシデント対応プログラムと戦略の基礎となるように設計されています。インシデント対応計画は正式な文書にする必要があります。インシデント対応計画には通常、次のセクションが含まれます。

- インシデント対応チームの概要: インシデント対応チームの目標と機能の概要を説明します。
- ロールと責任: インシデントに対応する利害関係者を一覧で表示し、インシデントが発生時のそれぞれの役割を詳しく説明します。
- コミュニケーション計画: 連絡先情報とインシデント発生時の連絡方法を詳しく説明します。
- バックアップの通信方法: インシデント時の通信のバックアップとして帯域外通信を行うことがベストプラクティスです。安全な帯域外通信チャネルを提供するアプリケーションの例は AWS Wickr です。
- インシデント対応の段階と取るべき措置: インシデント対応のフェーズ (検出、分析、根絶、封じ込め、復旧など) を列挙し、それらのフェーズで取るべき大まかなアクションも含めます。
- インシデントの重要度と優先順位の定義: インシデントの重大度を分類する方法、インシデントの優先順位付け方法、および重要度の定義がエスカレーション手順にどのように影響するかを詳しく説明します。

これらのセクションは、さまざまな規模や業界の企業で共通していますが、各組織のインシデント対応計画は異なります。組織に最適なインシデント対応計画を立てる必要があります。

リソース

関連するベストプラクティス:

- SEC04 (セキュリティイベントは、どのように検出して調査するのですか?)

関連するドキュメント:

- AWS Security Incident Response Guide (AWS セキュリティインシデント対応ガイド)
- NIST: Computer Security Incident Handling Guide

SEC10-BP03 フォレンジック機能を備える:

セキュリティインシデントが発生する前に、セキュリティイベントの調査を支援するフォレンジック機能の整備を検討します。

このベストプラクティスが確立されていない場合のリスクレベル: 中

AWS には、従来のオンプレミスフォレンジックの概念が適用されます。AWS クラウド でフォレンジック機能の構築を開始するための重要な情報については、「Forensic investigation environment strategies in the AWS クラウド」を参照してください。

フォレンジックのための環境と AWS アカウント構造が整ったら、次の 4 つのフェーズにわたってフォレンジックに適した方法論を効果的に実行するために必要なテクノロジーを定義します。

- 収集: AWS CloudTrail、AWS Config、VPC フローログ、ホストレベルのログなどの関連 AWS ログを収集します。可能であれば、影響を受けた AWS リソースのスナップショット、バックアップ、メモリダンプを収集します。
- 調査: 関連する情報を抽出して評価することにより、収集されたデータを検証します。
- 分析: 収集したデータを分析してインシデントを解明し、そこから結論を導き出します。
- レポート: 分析フェーズから得られた情報を報告します。

実装手順

フォレンジック環境を準備する

AWS Organizations では、AWS リソースの拡大とスケールに合わせて、AWS 環境を一元的に管理および運用できます。AWS 組織を利用することで、AWS アカウントを統合して 1 つのユニットとして管理でき

るようになります。組織単位 (OU) を使用してアカウントをグループ化し、1 つのユニットとして管理できます。

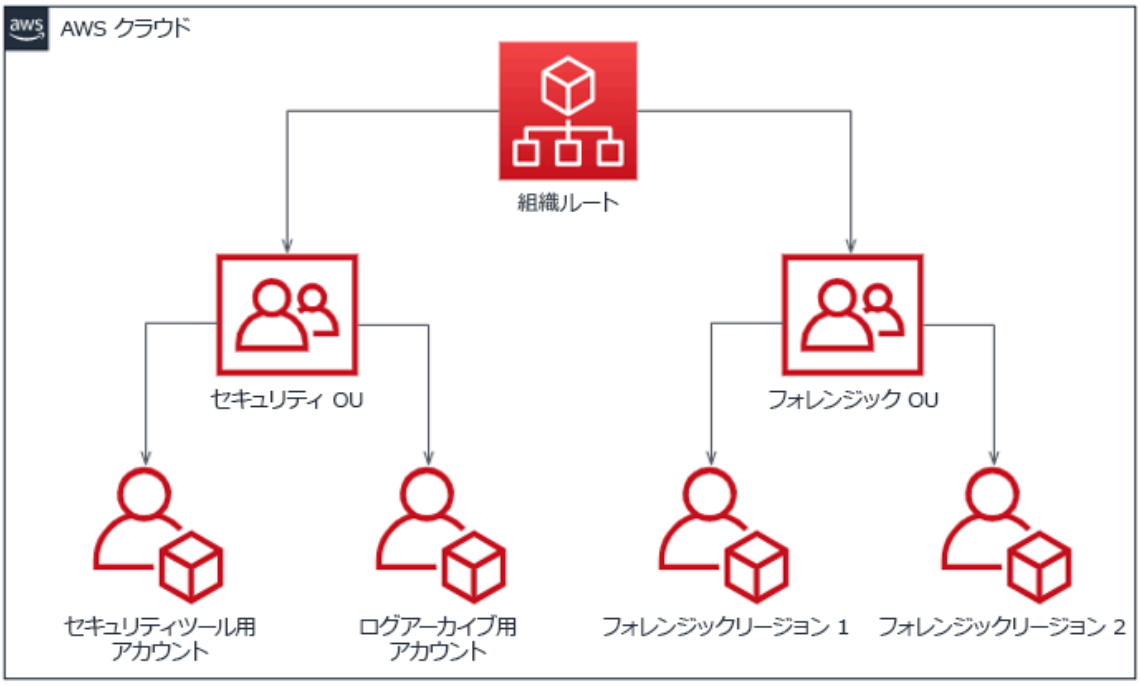
インシデント対応には、セキュリティ OU とフォレンジック OU を含むインシデント対応の機能をサポートする AWS アカウント構造があると便利です。セキュリティ OU 内には、次のアカウントが必要です。

- ログアーカイブ: 限られたアクセス許可を持つログアーカイブ用の AWS アカウントにログを集約します。
- セキュリティツール: セキュリティサービスをセキュリティツール用の AWS アカウントに一元化します。このアカウントは、セキュリティサービスの委任管理者として機能します。

フォレンジック OU 内では、お客様のビジネスモデルと運用モデルに最適なフォレンジックアカウントに応じて、フォレンジック用に 1 つのアカウントを実装するか、事業を展開するリージョンごとにアカウントを実装できます。リージョンごとにフォレンジックアカウントを作成すると、そのリージョン外での AWS リソースの作成をブロックし、リソースが意図しないリージョンにコピーされるリスクを低減できます。例えば、米国東部 (バージニア北部) (us-east-1) と米国西部 (オレゴン) (us-west-2) のみで事業を行う場合、フォレンジック OU には 2 つのアカウントを作成することになります。1 つは us-east-1 用で、もう 1 つは us-west-2 用です。

複数のリージョンのフォレンジック AWS アカウントを作成できます。そのアカウントに AWS リソースをコピーする場合は、データ主権に関する要件に準拠しているか注意する必要があります。新しいアカウントのプロビジョニングには時間がかかるため、インシデントのかなり前にフォレンジックアカウントを作成して実装し、対応担当者が効果的に対応できるように準備しておくことが重要です。

次の図は、リージョンごとのフォレンジックアカウントを持つフォレンジック OU を含むアカウント構造の例を示しています。



インシデント対応のためのリージョンごとのアカウント構造

バックアップとスナップショットをキャプチャする

主要なシステムとデータベースのバックアップをセットアップすることは、セキュリティインシデントからの回復とフォレンジックのために重要です。バックアップを作成しておけば、システムを以前の安全

な状態に復元できます。AWS では、さまざまなリソースのスナップショットを作成できます。スナップショットでは、こうしたリソースのポイントインタイムバックアップを作成できます。バックアップや復旧をサポートできる AWS のサービスは数多くあります。バックアップと復旧のサービスとアプローチの詳細については、バックアップと復旧についての規範ガイダンスとブログ記事「Use backups to recover from security incidents」を参照してください。

特にランサムウェアのような状況では、バックアップをしっかりと保護することが重要です。バックアップの保護に関するガイダンスについては、Top 10 security best practices for securing backups in AWS」を参照してください。バックアップの保護に加えて、バックアップと復元のプロセスを定期的にテストして、導入しているテクノロジーとプロセスが想定どおりに機能することを確認する必要があります。

フォレンジックを自動化する

セキュリティイベント中、インシデント対応チームは、イベント前後の期間の証拠を、正確性を維持しながら迅速に収集して分析できなければなりません (特定のイベントやリソースに関連するログのキャプチャ、Amazon EC2 インスタンスのメモリダンプの収集など)。インシデント対応チームにとって、関連する証拠を手作業で収集することは困難であり、時間もかかります。多数のインスタンスやアカウントが対象となる場合は特にそうです。さらに、手作業による収集では人為的ミスが起こりやすくなります。このような理由から、フォレンジックの自動化を可能な限り開発し、実装する必要があります。

AWS には、フォレンジック用の自動化リソースが多数用意されており、これらのリソースは以下のリソースセクションに一覧表示されています。これらのリソースは、AWS が開発し、お客様が実装したフォレンジックパターンの例です。手始めに参考にするリファレンスアーキテクチャとしては有効かもしれませんが、環境、要件、ツール、フォレンジックプロセスに基に変更するか、新しいフォレンジック自動化パターンを作成することを検討してください。

リソース

関連するドキュメント:

- AWS Security Incident Response Guide - Develop Forensics Capabilities
- AWS Security Incident Response Guide - Forensics Resources
- Forensic investigation environment strategies in the AWS クラウド
- How to automate forensic disk collection in AWS
- AWS Prescriptive Guidance - Automate incident response and forensics

関連動画:

- インシデント対応とフォレンジックの自動化

関連する例:

- Automated Incident Response and Forensics Framework
- Automated Forensics Orchestrator for Amazon EC2

SEC10-BP04 セキュリティインシデント対応プレイブックを作成し、テストする

インシデント対応プロセスを準備する上で重要なのは、プレイブックを作成することです。インシデント対応プレイブックには、セキュリティイベントが発生したときに従うべき一連の規範的なガイダンスと手順が記載されています。明確な体制と手順があると、対応が簡単になり、人為的ミスの可能性が低くなります。

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

プレイブックは、次のようなインシデントシナリオ向けに作成する必要があります。

- 予想されるインシデント: プレイブックは、予測されるインシデントに合わせて作成する必要があります。これには、サービス拒否 (DoS)、ランサムウェア、認証情報の漏えいなどの脅威が含まれます。
- 既知のセキュリティ上の検出結果またはアラート: プレイブックは、既知のセキュリティ上の検出結果とアラート (GuardDuty の検出結果など) に基づいて作成する必要があります。GuardDuty の検出結果を受け取っても、どうすればよいかわからないといったことがあるかもしれません。 そこで、GuardDuty の検出結果を誤って処理したり無視したりすることがないように、GuardDuty で検出される可能性のある問題ごとにプレイブックを作成しておきます。修正に関する詳細とガイダンスについては、GuardDuty ドキュメントを参照してください。なお、GuardDuty はデフォルトでは有効になっておらず、コストがかかりますので注意してください。GuardDuty の詳細については、「Appendix A: Cloud capability definitions - Visibility and alerting」を参照してください。

プレイブックには、起こりうるセキュリティインシデントを適切に調査して対応するために、セキュリティアナリストが実行すべき技術的な手順を記載する必要があります。

実装手順

プレイブックに記載すべき項目には次のようなものがあります。

- プレイブックの概要: このプレイブックがどのようなリスクやインシデントシナリオに対応しているか。このプレイブックの目的は何か。
- 前提条件: このインシデントシナリオには、どのようなログ、検出メカニズム、自動ツールが必要か。どのような通知が想定されるか。
- コミュニケーションとエスカレーションに関する情報: 誰が関与しているか、連絡先は把握できているか。 各利害関係者の責任は何か。
- 対応ステップ: インシデント対応の各フェーズで、どのような戦術的措置を講じるべきか。 アナリストはどのようなクエリを実行すべきか。 望ましい結果を得るためにどのようなコードを実行すべきか。
  - 検知: インシデントはどのように検出されるか。
  - 分析: 影響範囲はどのように特定されるか。
  - 封じ込め: 影響範囲を限定するために、インシデントをどのように隔離するか。
  - 根絶: どのようにして脅威を環境から取り除くか。
  - 復旧: 影響を受けたシステムやリソースをどのようにして本番環境に戻すか。
- 想定される結果: クエリとコードが実行された後、プレイブックで想定される結果はどのようなものか。

リソース

関連する Well-Architected のベストプラクティス

- SEC10-BP02 - インシデント管理計画を作成する

関連するドキュメント:

- Framework for Incident Response Playbooks
- Develop your own Incident Response Playbooks
- Incident Response Playbook Samples
- Building an AWS incident response runbook using Jupyter playbooks and CloudTrail Lake

SEC10-BP05 アクセスを事前プロビジョニングする

インシデント対応者が AWS に事前プロビジョニングされた正しいアクセス権を持っていることを検証しておき、調査から復旧までに必要な時間を短縮します。

一般的なアンチパターン:

- ルートアカウントをインシデント対応に使用する
- 既存のユーザーアカウントに変更を加える
- ジャストインタイムの権限昇格を提供する際に IAM アクセス許可を直接操作する

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

AWS は、可能であれば長期的な認証情報への依存を削減または排除し、一時的な認証情報と ジャストインタイム の権限昇格メカニズムを優先することを推奨します。長期的な認証情報は、セキュリティリスクにさらされやすく、オペレーションのオーバーヘッドを増大させます。ほとんどの管理タスクと、インシデント対応タスクについては、管理アクセスの一時的な昇格と併せて ID フェデレーション を実装することを お勧めします。このモデルでは、ユーザーはより高いレベルの権限 (インシデント対応ロールなど) への昇格をリクエストします。ユーザーに昇格の資格がある場合、リクエストは承認者に送信されます。リクエストが承認された場合、ユーザーは、一時的な AWS 認証情報 のセットを受け取り、これを使用してタスクを完了できます。これらの認証情報の期限が切れたら、ユーザーは新たな昇格リクエストを送信する必要があります。

インシデント対応の大半のケースでは、一時的な権限昇格を使用することをお勧めします。そのための適切な方法は、 AWS Security Token Service および セッションポリシー を使用してアクセスのスコープを定義することです。

ID フェデレーションを使用できないケースがあります。例えば次のケースです。

- 侵害を受けた ID プロバイダー (IdP) に関連する停止状態
- 設定ミスや人的エラーに起因する、フェデレーションアクセス管理システムの障害
- 分散型サービス拒否 (DDoS) イベントやシステムがレンダリング不可となるなどの悪意あるアクティビティ

上記のケースでは、緊急 break glass アクセス設定により、インシデントの調査とタイムリーな修復を許可する必要があります。AWS は、 適切なアクセス許可を持つ IAM ユーザー を使用することをお勧めします。IAM ユーザーがタスクを実行し AWS のリソースにアクセスするための適切な許可を付与します。ルート認証情報は、 ルートユーザーアクセスが必要なタスクのみに使用します。インシデント対応者が AWS と他の関連システムへの適切なレベルのアクセス権を持っていることを検証するには、専用のユーザーアカウントへの事前プロビジョニングをお勧めします。このユーザーアカウントには特権アクセスが必要で、アカウントは厳格に制御、監視されなければなりません。このアカウントは、必要なタスクの実行で要求される最小特権で構成しなければなりません。アクセス権のレベルは、インシデント管理計画の一環として作成されたプレイブックに基づいている必要があります。

ベストプラクティスとして、特定の目的のための専用のユーザーとロールを使用します。IAM ポリシーの追加によりユーザーまたはロールアクセスを一時的に昇格させると、インシデント対応中にユーザーがどのアクセス権を持っていたかが明確でなくなり、昇格された権限が取り消されないリスクが生じます。

できるだけ多くの依存関係を削除し、できるだけ多くの障害シナリオでアクセスが可能になることを検証することが重要です。そのためには、インシデント対応ユーザーが、専用のセキュリティアカウントで AWS Identity and Access Management ユーザーとして作成されており、既存のフェデレーションまたはシングルサインオン (SSO) ソリューションにより管理されていないことを検証するためのプレイブックを作成します。個々のインシデント対応者は、自分の名前が付いたアカウントを持つ必要があります。アカウント設定では、 強力なパスワードポリシー および多要素認証 (MFA) を適用する必要があります。

インシデント対応プレイブックで AWS Management Console へのアクセスのみが要求されている場合、そのユーザーのアクセスキーが設定されてはならず、アクセスキー作成を明示的に禁止する必要があります。これは IAM ポリシーまたはサービスコントロールポリシー (SCP) で設定できます。詳細は、『AWS Security Best Practices for AWS Organizations SCPs』(AWS Organizations SCP のための AWS セキュリティベストプラクティス) に記載されています。ユーザーは、他のアカウントのインシデント対応ロールを引き受ける以外の権限を持つべきではありません。

インシデント対応中、調査、修復、または復旧アクティビティをサポートするためのアクセス権を社内または社外の他の個人に付与する必要が生じる可能性があります。この場合、前述のプレイブックメカニズムを使用します。また、インシデント完結後直ちに追加のアクセス権を取り消すためのプロセスが必要です。

インシデント対応ロールの使用が適切に監視および監査されていることを検証するには、この目的のために作成された IAM ユーザーアカウントが個人間で共有されないようにすること、および特定のタスクで必要な場合を除き、AWS アカウント ルートユーザーが使用されないようにすることが 不可欠です。ルートユーザーが必要な場合 (例えば、特定のアカウントへの IAM アクセスが利用できない場合) は、用意されたプレイブックに従って別個のプロセスを使用し、ルートユーザーのパスワードと MFA トークンの使用の可否を検証します。

インシデント対応ロールのための IAM ポリシーを設定するには、 IAM Access Analyzer を使用し AWS CloudTrail ログに基づいてポリシーを生成することを検討します。そのためには、非本番アカウントのインシデント対応ロールに管理者アクセス権を付与し、プレイブックを一通り実行します。完了したら、実行されたアクションのみを許可するポリシーを作成できます。このポリシーは、すべてのアカウントのすべてのインシデント対応ロールに適用できます。各プレイブックについて個別の IAM ポリシーを作成すると、管理と監査が容易になるでしょう。プレイブックの例には、ランサムウェア、データ侵害、本番環境へのアクセス不可、その他のシナリオについての対応計画が含まれています。

インシデント対応ユーザーアカウントを使用して、 別の AWS アカウント アカウントのインシデント対応専用の IAM ロールを引き受けます。これらのロールは、セキュリティアカウントのユーザーのみが引き受け可能なように設定する必要があります。信頼関係では、呼び出しプリンシパルが MFA を使用して認証されたことを要求する必要があります。ロールは、スコープが厳密に定義された IAM ポリシーを使用してアクセスを制御する必要があります。これらのロールに対するすべての `AssumeRole` リクエストが CloudTrail ログに記録され、アラートが送信されるようにします。また、これらのロールを使用して実行されたアクションがログに記録されるようにします。

IAM ユーザーアカウントと IAM ロールの両方を CloudTrail ログで見つけやすくするために、これらに明快な名前を付けることを強くお勧めします。例えば、IAM アカウントに *`<USER_ID>`*`break-glass` 、IAM ロールに `BREAK-GLASS-ROLE` という名前を付けます。

CloudTrail を使用して、AWS アカウントの API アクティビティをログに記録します。また、 インシデント対応ロールの使用状況に関するアラートを設定する必要があります。ルートキーを使用する際のアラートの設定に関するブログ記事を参照してください。インストラクションに変更を加えて、 Amazon CloudWatch メトリクスフィルターを `AssumeRole` イベント (インシデント対応 IAM ロールに関連する) に対して設定できます。

```
{ $.eventName = "AssumeRole" && $.requestParameters.roleArn =
 "<INCIDENT_RESPONSE_ROLE_ARN>" && $.userIdentity.invokedBy NOT EXISTS && $.eventType !=
 "AwsServiceEvent" }
```

インシデント対応ロールは高いレベルのアクセス権を持っている可能性があるため、これらのアラートは幅広いグループに送信され、速やかに対応が取られることが重要です。

インシデント対応中、対応者は、IAM によって直接保護されていないシステムへのアクセスが必要となる可能性があります。これには Amazon Elastic Compute Cloud インスタンス、Amazon Relational Database Service データベース、Software-as-a-Service (SaaS) プラットフォームが含まれます。SSH や RDP などのネイティブプロトコルではなく、AWS Systems Manager Session Manager を使用して Amazon EC2 インスタンスへの管理アクセスを行うことを強くお勧めします。このアクセスは、IAM を使用して制御できます。それにより安全が確保され、監査が行われます。また、 AWS Systems Manager Run Command ドキュメントを使用してプレイブックの一部を自動化することも可能です。それにより、ユーザーのエラー

を減らし、復旧にかかる時間を短縮できます。データベースとサードパーティーツールへのアクセスでは、アクセス認証情報を AWS Secrets Manager に保管し、インシデント対応者ロールにアクセス権を付与することをお勧めします。

最後に、インシデント対応 IAM ユーザーアカウントの管理は、 Joiners、Movers、および Leavers プロセス に追加し、定期的にテストして、意図されたアクセスのみが許可されていることを検証する必要があります。

リソース

関連するドキュメント:

- Managing temporary elevated access to your AWS environment (AWS 環境へのアクセスの一時的な昇格の管理)
- AWS Security Incident Response Guide (AWS セキュリティインシデント対応ガイド)
- AWS Elastic Disaster Recovery
- AWS Systems Manager Incident Manager
- IAM ユーザー用のアカウントパスワードポリシーの設定
- AWS での多要素認証 (MFA) の使用
- Configuring Cross-Account Access with MFA (MFA を使用したクロスアカウントアクセスの設定)
- Using IAM Access Analyzer to generate IAM policies (IAM Access Analyzer を使用した IAM ポリシーの設定)
- Best Practices for AWS Organizations Service Control Policies in a Multi-Account Environment (マルチアカウント環境の AWS Organizations サービスコントロールポリシーのためのベストプラクティス)
- How to Receive Notifications When Your AWS Account's Root Access Keys Are Used (AWS アカウントのルートアクセスキーを使用した場合の通知の受信方法)
- Create fine-grained session permissions using IAM managed policies(AWS マネージドポリシーを使用して、きめ細かいセッション許可を作成する)

関連動画:

- AWS のインシデント対応とフォレンジックの自動化
- ランブック、インシデントレポート、インシデント対応の DIY ガイド
- AWS 環境のセキュリティインシデントの準備と対応

関連サンプル:

- ラボ: AWS Account Setup and Root User (AWS アカウントのセットアップとルートユーザー)
- ラボ: Incident Response with AWS Console and CLI (AWS コンソールと CLI を使用したインシデント対応)

SEC10-BP06 ツールを事前デプロイする

復旧までの調査時間を短縮できるように、セキュリティ担当者は適切なツールを事前にデプロイしておきます。

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

セキュリティ対応と運用機能を自動化するために、AWS の包括的な API とツールセットを使用できます。ID 管理、ネットワークセキュリティ、データ保護、モニタリング機能を完全に自動化し、すでに導入

されている一般的なソフトウェア開発方法を使用して提供できます。セキュリティオートメーションを構築すれば、担当者がセキュリティ上の位置づけを監視し、イベントに手動で応答する代わりに、システムが監視、レビューを行い応答を開始できます。

インシデント対応チームが同じ方法でアラートに対応し続けると、アラート疲れになるリスクがあります。時間の経過とともに、チームはアラートに対する感度が鈍くなり、通常の状況の処理で間違いを犯したり、異常なアラートを見逃したりする可能性があります。自動化を利用すれば、繰り返し発生する通常のアラートを処理する機能を使用してアラート疲れを回避し、機密性の高いインシデントや独自のインシデントの処理を人間に任せることができます。Amazon GuardDuty, AWS CloudTrail Insights、および Amazon CloudWatch Anomaly Detection などの異常の検出システムを統合することで、よくあるしきい値ベースのアラートの負担を減らすことができます。

プロセス内のステップをプログラムで自動化すれば、手動プロセスを改善できます。イベントに対する修復パターンを定義したら、そのパターンを実行可能なロジックに分解して、そのロジックを実行するコードを記述できます。その後、対応者は、そのコードを実行して問題を修正します。時間の経過とともに、より多くのステップを自動化し、最終的には一般的なインシデントのクラス全体を自動的に処理できるようになります。

セキュリティ調査中、インシデントの全容とタイムラインを記録して理解するために、関連ログを確認できる必要があります。ログはまた、関心のある特定のアクションが発生したことを示すアラート生成にも必須です。クエリと取得のメカニズムとアラートを選択、有効化、保存、セットアップし、アラート発行を設定することが非常に重要となります。さらに、ログデータを検索するツールとして、Amazon Detective が有効です。

AWS では、200 を超えるクラウドサービスと数千の機能を提供しています。インシデント対応戦略をサポートし、簡素化できるサービスを確認することをお勧めします。

ログ記録に加えて、タグ付け戦略を策定して実装する必要があります。タグ付けを行うことで、AWS リソースの目的についての背景情報を付け加えることができます。タグ付けは自動化にも使用できます。

実装手順

分析とアラート発行のためのログを選択して設定する

インシデント対応のログ記録の設定については、次のドキュメントを参照してください。

- Logging strategies for security incident response
- SEC04-BP01 サービスとアプリケーションのログ記録を設定する (p. 56)

検出と対応をサポートするセキュリティサービスを有効にする

AWS は検出、予防、対応のネイティブ機能備えているほか、カスタムセキュリティソリューションの構築に使用できるサービスも提供しています。セキュリティインシデント対応にに最も関連性の高いサービスのリストについては、「Cloud capability definitions」を参照してください。

タグ付け戦略を策定し、実装する

AWS リソースを取り巻くビジネスユースケースや関わりのある内部関係者についての背景情報の入手は難しい場合があります。これを達成する方法の 1 つとして、タグを使用して、ユーザー定義のキーと値で構成されるメタデータを AWS リソースに割り当てる方法があります。タグを作成して、目的、所有者、環境、処理されるデータの種類など、任意の基準でリソースを分類できます。

一貫したタグ付け戦略があると、AWS リソースに関する背景情報をすばやく特定、識別できるため、応答時間を短縮し、組織の背景情報の把握に費やす時間を最小限に抑えることができます。タグは、対応の自動化を開始するためのメカニズムとしても機能します。タグ付けの対象に関する詳細については、「Tagging your AWS resources」を参照してください。まず、組織全体に導入するタグを定義する必要があります。その後、このタグ付け戦略を導入し、適用します。導入と適用の詳細については、「Implement AWS resource tagging strategy using AWS Tag Policies and Service Control Policies (SCPs)」を参照してください。

リソース

関連する Well-Architected のベストプラクティス

- SEC04-BP01 サービスとアプリケーションのログ記録を設定する (p. 56)
- SEC04-BP02 ログ、結果、メトリクスを一元的に分析する (p. 59)

関連するドキュメント:

- Logging strategies for security incident response
- Incident response cloud capability definitions

関連する例:

- Threat Detection and Response with Amazon GuardDuty and Amazon Detective
- Security Hub Workshop
- Vulnerability Management with Amazon Inspector

SEC10-BP07 シミュレーション行う

組織が成長し進化するにつれて、脅威の状況も変化するため、インシデント対応能力を継続的に見直すことが重要になります。この評価を行う方法の 1 つとして、シミュレーション (ゲームデーとも呼ばれる) の実施があります。シミュレーションでは、脅威アクターの戦術、手法、手順 (TTP) を模倣するように設計された現実のセキュリティイベントシナリオを使用します。これにより、組織は実際に発生する可能性のある模擬サイバーイベントに対応することで、インシデント対応能力を訓練し、評価できます。

このベストプラクティスを確立するメリット: シミュレーションにはさまざまな利点があります。

- サイバー脅威への準備状況を検証し、インシデント対応者の信頼度を高めます。
- ツールとワークフローの精度と効率性をテストします。
- インシデント対応計画に沿うように、コミュニケーションとエスカレーションの方法を改良します。
- あまり一般的でないベクトルに対応する機会を提供します。

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

シミュレーションには主に 3 つのタイプがあります。

- 机上演習: 机上でのシミュレーションは、インシデントに対応するさまざまな利害関係者が参加して役割や責任を実践し、確立されたコミュニケーションツールやプレイブックを活用するディスカッションベースのセッションです。演習は、通常はバーチャル会場、実際の施設、またはそれらの組み合わせが可能で、丸 1 日かけて進行します。ディスカッションベースのため、机上演習ではプロセス、人材、コラボレーションに焦点を当てます。テクノロジーは議論に不可欠ですが、インシデント対応ツールやスクリプトを実際に使用することは、一般的に机上演習には含まれません。
- パープルチーム演習: パープルチーム演習は、インシデント対応者 (ブルーチーム) と模擬の脅威アクター (レッドチーム) のコラボレーションレベルを高めるものです。ブルーチームはセキュリティオペレーションセンター (SOC) のメンバーで構成されますが、実際のサイバーイベントに関与する他の利害関係者が参加することもあります。レッドチームは、攻撃的なセキュリティのトレーニングを受けたペネトレーションテストチームまたは主な利害関係者で構成されています。レッドチームは、シナリオが正確で実現可能なものになるように、演習のファシリテーターと協力して作業します。パープルチーム

演習では、インシデント対応の取り組みを支援する検出メカニズム、ツール、標準運用手順 (SOP) に重点が置かれます。

- レッドチーム演習: レッドチーム演習では、攻撃側 (レッドチーム) は、あらかじめ決められた範囲から、ある目的または一連の目的を達成するためのシミュレーションを行います。防御側 (ブルーチーム) は、演習の範囲と期間について必ずしも知識を持っているとは限らないため、実際のインシデントにどのように対応するか、より現実的に評価できます。レッドチーム演習では侵入テストになる可能性があります。そのため慎重に行い、コントロールを実施して、演習によって環境に実害を与えないことを確認してください。

定期的にサイバーシミュレーションを実施することを検討してください。演習は、タイプごとにそれぞれのメリットを参加者と組織全体にもたらします。それほど複雑ではないタイプのシミュレーション (机上演習など) から始めて、より複雑なシミュレーションタイプ (レッドチーム演習) に進むこともできます。セキュリティの成熟度、リソース、目標とする成果に基づいてシミュレーションタイプを選択する必要があります。お客様によっては、複雑さやコスト面から、レッドチーム演習を選択しない場合があります。

実装手順

選択したシミュレーションタイプにかかわらず、シミュレーションは通常、以下のような実施手順に従います。

1. 演習の中核要素の定義: シミュレーションのシナリオとの目的を定義します。いずれも、リーダーの承認が必要です。
2. 主な利害関係者の特定: 少なくとも、演習には演習のファシリテーターと参加者が必要です。シナリオによっては、追加で法務、コミュニケーション、経営幹部などの利害関係者が関与する場合があります。
3. シナリオの構築とテスト: 特定の要素が実現不可能な場合は、シナリオの構築中に再定義が必要なこともあります。このステージのアウトプットとして、シナリオの最終版が完成することが期待されます。
4. シミュレーションの進行: シミュレーションのタイプによって、使用する進行内容 (紙ベースのシナリオと技術的に高度なシミュレーションシナリオの比較) が決まります。ファシリテーターは、演習進行の戦略を目的に合わせて調整し、最大の効果が得られるように、できるだけすべての参加者に演習に参加してもらう必要があります。
5. アフターアクションレビュー (AAR) の作成: うまくいった部分、改善の余地がある部分、潜在的なギャップを特定します。AAR では、シミュレーションの有効性だけでなく、シミュレートされたイベントに対するチームの反応も測定して、今後のシミュレーションの進捗を経時的に追跡できるようにする必要があります。

リソース

関連するドキュメント:

- AWS Incident Response Guide

関連動画:

- AWS GameDay - Security Edition

オペレーション

インシデント対応の実施では、運用が中核となります。ここで、セキュリティインシデントへの対応と修復が行われます。運用には次の 5 つのフェーズが含まれます。 検知、分析、 封じ込め、根絶、 復旧。これらのフェーズと目標の説明は、次の表にあります。

フェーズ	目標
検知	潜在的なセキュリティイベントを特定します。
分析	セキュリティイベントがインシデントかどうかを判断し、インシデントの範囲を評価します。
封じ込め	セキュリティイベントの範囲を最小限に抑え、制限します。
根絶	セキュリティイベントに関連する不正なリソースやアーティファクトを削除します。セキュリティインシデントの原因となった緩和策を実装します。
復旧	システムを既知の安全な状態に復元し、これらのシステムを監視して脅威が再発しないことを確認します。

これらのフェーズは、効果的かつ堅牢な方法で対応するために、セキュリティインシデントに対応して運用する際の指針となるはずです。実際に実行するアクションは、インシデントによって異なります。例えば、ランサムウェアが関係するインシデントは、パブリック Amazon S3 バケットに関連するインシデントとは異なる対応手順を踏む必要があります。さらに、これらのフェーズは必ずしも連続して発生するわけではありません。封じ込めおよび根絶後は、分析に戻って対策が効果的だったかどうかを把握する必要があるかもしれません。

人員、プロセス、テクノロジーを綿密に準備することが、業務を効果的に行う鍵となります。したがって、準備のセクションの ベストプラクティスに従って、 (p. 103) アクティブなセキュリティイベントに効果的に対応できるようにしてください。

詳細 については AWS セキュリティインシデント対応ガイドの運用のセクションを参照してください。

インシデント後のアクティビティ

脅威の状況は絶えず変化しているため、環境を効果的に保護するためには、組織の能力も同様に動的なものにすることが重要です。継続的な改善の鍵は、インシデントとシミュレーションの結果を反復することで、想定されるセキュリティインシデントを効果的に検出、対応、調査する能力を向上させることです。これにより、想定される脆弱性や、対応までの時間を短縮し、安全な運用に復帰することができます。以下のメカニズムは、組織がどのような状況でも効果的に対応するための最新の能力と知識を十分に備えていることを確認するのに役立ちます。

ベストプラクティス

- SEC10-BP08 インシデントから学ぶためのフレームワークを確立する (p. 116)

SEC10-BP08 インシデントから学ぶためのフレームワークを確立する

教訓フレームワークと根本原因分析プロセスを導入することは、インシデント対応能力の向上だけでなく、インシデントの再発防止にも役立ちます。各インシデントから学ぶことで、同じ失敗、露出、設定ミスの繰り返しを防ぐことができ、セキュリティ体制が強化されるだけでなく、予防できたはずの状況に無駄にする時間を最小限に抑えることができます。

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

次の要点を大局的に確立し、達成する教訓フレームワークを導入することが重要です。

- 事後検証会を実施するタイミング
- 事後検証会を通して行うこと
- 事後検証会の実施方法
- そのプロセスに関わる人物、また関わり方
- 改善の余地がある領域の特定方法
- 改善事項を効果的に追跡、実装する方法

フレームワークは、個人に焦点を当てたり非難したりするのではなく、ツールやプロセスの改善に焦点を当てるべきです。

実装手順

上記の大局的な成果とは別に、プロセスから最大の価値 (実行可能な改善につながる情報) を引き出すためには、適切な質問を行うことが重要です。教訓についての議論を進めるうえで役立つ質問には次のようなものがあります。

- どのようなインシデントでしたか。
- インシデントが最初に特定されたのはいつでしたか。
- どのようにして特定されましたか。
- どのシステムからアクティビティについてのアラートが発行されましたか。
- どのようなシステム、サービス、データが関与しましたか。
- 具体的に何が起きましたか。
- 何がうまくいきましたか。
- 何がうまくいきませんでしたか。
- インシデントに対応できなかった、またはスケールに失敗したのはどのプロセスまたは手順ですか。
- 次の領域で改善できることは何でしょうか。
  - 人材
    - 連絡する必要があった担当者に実際に連絡がつきましたか。また、連絡先リストの情報は最新のものでしたか。
    - インシデントに効果的に対応して調査するために必要なトレーニングや能力を欠いていましたか。
    - 適切なリソースは用意されていましたか。
  - プロセス
    - 対応はプロセスと手順に従って進められましたか。
    - この (タイプの) インシデントについて、プロセスと手順が文書化され、利用可能になっていましたか。
    - 必要なプロセスや手順が欠けていましたか。
    - 対応担当者は、問題に対応するために必要な情報にタイムリーにアクセスできましたか。
  - 技術
    - 既存のアラートシステムは、アクティビティを効果的に特定してアラートを出しましたか。
    - どうすれば検出までの時間を 50% 短縮できたでしょうか。
    - この (タイプの) インシデントに備えて、既存のアラートを改善する、または新しいアラートを作成する必要がありますか。
    - 既存のツールでインシデントを効果的に調査 (検索/分析) できましたか。
    - この (タイプの) インシデントをより早く特定するにはどうすればよいでしょうか。
    - この (タイプの) インシデントの再発を防ぐにはどうすればよいでしょうか。

- 改善計画の所有者は誰ですか。また、その実施状況をどのように検証しますか。
- 追加のモニタリング、予防的統制やプロセスを導入し、テストするまでのスケジュールはどのようになっていますか。

このリストはすべてを網羅しているわけではありませんが、インシデントから最も効果的に学び、セキュリティ体制の継続的な改善に向けて、組織とビジネスのニーズを見極め、その分析方法を特定するための出発点として活用いただくことを目的としています。最も重要なのは、事後検証会を標準的なインシデント対応プロセスと文書化の一部として取り入れ、想定されるものとして関係者全員にも定着させることです。

リソース

関連するドキュメント:

- AWS Security Incident Response Guide - Establish a framework for learning from incidents
- NCSC CAF guidance - Lessons learned

アプリケーションのセキュリティ

アプリケーションのセキュリティ (AppSec) は、開発するワークロードのセキュリティ特性の設計、構築、テストの各方法の全体的なプロセスを説明するものです。適切なトレーニングを受けた人材を組織に配置した上で、ビルドのセキュリティ特性を理解してインフラストラクチャをリリースし、自動化を使用してセキュリティの問題を識別する必要があります。

ソフトウェア開発ライフサイクル (SDLC) とリリース後プロセスの一部としてアプリケーションセキュリティテストを採用することで、本稼働環境でのアプリケーションのセキュリティ問題の識別、修正、防止のための構造的なメカニズムを確立することができます。

ワークロードを設計、構築、デプロイ、運用する際、アプリケーション開発手法にはセキュリティコントロールを含める必要があります。その中で、継続的な欠陥削減のプロセスと技術的負債の低減を調整します。例えば、脅威モデリングを設計フェーズで使用すると、設計上の欠陥を早期に発見することができ、後になって問題を軽減するのと比較して、欠陥の修正をより簡単に、より安価に行うことができます。

一般的に、SDLC の早期に欠陥を解決することで、コストと複雑性を抑えられます。最も簡単な問題解決の方法は、そもそも問題を抱えないことです。したがって、最初に脅威モデルを作成することで、設計フェーズから適切な成果にフォーカスすることができます。AppSec プログラムが成熟するにつれて、自動化を使ってテストの数を増やしたり、ビルダーへのフィードバックの忠実性を高めたり、セキュリティレビューに必要な時間を短縮したりすることができます。これらすべてのアクションは、構築するソフトウェアの品質を改善し、本稼働への機能の実装までの時間を短縮します。

これらの実装ガイドラインは、組織と文化、パイプラインのセキュリティ、パイプラインでのセキュリティ、および依存関係管理の 4 つの領域にフォーカスしています。各領域は、実装可能な一連の原則と、ワークロードの設計、開発、構築、デプロイ、運用のエンドツーエンドビューを提供します。

AWS には、アプリケーションのセキュリティプログラムに対処する際に使用できる多くのアプローチがあります。これらのアプローチの一部はテクノロジーに依存し、他のアプローチはアプリケーションのセキュリティプログラムにおける人や組織にフォーカスします。

ベストプラクティス

- SEC11-BP01 アプリケーションのセキュリティに関するトレーニングを実施する (p. 119)
- SEC11-BP02 開発およびリリースライフサイクル全体を通じてテストを自動化する (p. 121)
- SEC11-BP03 定期的にペネトレーションテストを実施する (p. 123)
- SEC11-BP04 手動のコードレビュー (p. 125)
- SEC11-BP05 パッケージと依存関係のサービスを一元化する (p. 127)
- SEC11-BP06 ソフトウェアをプログラムでデプロイする (p. 128)
- SEC11-BP07 パイプラインのセキュリティ特性を定期的に評価する (p. 130)
- SEC11-BP08 ワークロードチームにセキュリティのオーナーシップを根付かせるプログラムを構築する (p. 131)

SEC11-BP01 アプリケーションのセキュリティに関するトレーニングを実施する

組織のビルダーに対して、アプリケーションのセキュアな開発と運用のための一般的な手法に関するトレーニングを実施します。セキュリティを重視した開発手法を導入することは、セキュリティのレビューステージでしか検知されない問題が発生する可能性を減らすうえで役立ちます。

期待される成果: セキュリティを考慮したソフトウェアの設計と構築脅威モデルを起点とするセキュアな開発プラクティスについて組織のビルダーをトレーニングすることで、開発されるソフトウェアの全体的

な品質とセキュリティを向上させることができます。このアプローチによって、セキュリティレビューステージ後に必要な再作業を削減でき、ソフトウェアや機能をリリースするまでの時間を短縮できます。

このベストプラクティスでは、セキュアな開発は、開発されるソフトウェア、およびソフトウェア開発ライフサイクル (SDLC) をサポートするツールまたはシステムを指します。

一般的なアンチパターン:

- セキュリティレビューを待ち、その後、システムのセキュリティ特性を考慮する。
- セキュリティに関するすべての意思決定をセキュリティチームに委ねる。
- SDLC での意思決定方法に関するコミュニケーションの欠如が、全体的なセキュリティの期待や組織のポリシーに影響を与える。
- セキュリティレビュープロセスが遅延する。

このベストプラクティスを活用するメリット:

- 開発サイクルの早い段階で、組織のセキュリティ要件に関するより良い理解を得る。
- 潜在的なセキュリティの問題をすばやく識別および修正し、機能リリースまでの時間を短縮する。
- ソフトウェアとシステムの品質の向上。

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

組織のビルダーに対してトレーニングを実施します。セキュリティに関するトレーニングについては、脅威モデリングのコースから始めることを推奨します。理想的には、ビルダーは、ワークロードに関する情報に自分たちでアクセスできることが望まれます。このアクセスによって、ビルダーは他のチームに尋ねることなく、開発するシステムのセキュリティ特性に関する十分な情報に基づいた意思決定を行えます。レビューにおけるセキュリティチームの関与プロセスは、明確に定義され、容易に実行できる必要があります。レビュープロセスの各ステップは、セキュリティトレーニングに含める必要があります。既存の実装パターンやテンプレートを利用できる場合、それらは容易に見つけることができ、全体的なセキュリティ要件にリンクされている必要があります。AWS CloudFormation、AWS Cloud Development Kit (AWS CDK) Constructs、Service Catalog、または他のテンプレートツールの使用を検討し、カスタム構成に必要な時間を短縮します。

実装手順

- 良い基盤を築くため、ビルダーのトレーニングを脅威モデリングのコースから始め、セキュリティをどのように考慮すべきかについてのトレーニングを実施します。
- AWS トレーニング と認定、業種、または AWS パートナートレーニングへのアクセスを提供します。
- セキュリティチーム、ワークロードチーム、および他のステークホルダー間の責任分担を明確にするために、組織のセキュリティレビュープロセスについてのトレーニングを実施します。
- 利用可能な場合、コードの例やテンプレートを含め、セキュリティ要件を満たすためのセルフサービス型のガイダンスを提供します。
- セキュリティレビュープロセスとトレーニングについて、ビルダーチームからフィードバックを定期的に取得し、得られたフィードバックをもとに改善を行います。
- ゲームデーやバグバッシュキャンペーンを活用して、問題数の低減やビルダーのスキル向上に役立てます。

リソース

関連するベストプラクティス:

- SEC11-BP08 ワークロードチームにセキュリティのオーナーシップを根付かせるプログラムを構築する (p. 131)

関連するドキュメント:

- AWS トレーニング と認定
- How to think about cloud security governance (クラウドのセキュリティガバナンスをどのように考えるか)
- How to approach threat modeling (脅威モデリングにアプローチする方法)
- Accelerating training – The AWS Skills Guild (トレーニングの加速化 – AWS Skills Guild)

関連動画:

- Proactive security: Considerations and approaches (プロアクティブなセキュリティ: 考慮事項とアプローチ)

関連する例:

- Workshop on threat modeling (脅威モデリングについてのワークショップ)
- Industry awareness for developers (開発者向けの業界認識)

関連サービス:

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK) (AWS CDK) Constructs
- Service Catalog
- AWS BugBust

SEC11-BP02 開発およびリリースライフサイクル全体を通じてテストを自動化する

開発およびリリースライフサイクル全体を通じて、セキュリティ特性のテストを自動化します。自動化により、リリース前にソフトウェアの潜在的な問題を一貫して繰り返し確認することが容易になります。これにより、提供されるソフトウェアにおけるセキュリティ問題のリスクが減ります。

期待される成果: 自動化テストの目標は、開発の初期段階に、また多くの場合開発ライフサイクルをとおして、潜在的な問題を検知するプログラムを使用した手段を提供することです。リグレッションテストを自動化すると、すでにテスト済みのソフトウェアに変更を加えた後も、そのソフトウェアが期待どおりに動作することを確認するための機能テストおよび非機能テストを実行できます。機能していない認証、または不足している認証など、一般的な設定ミスをチェックするためのセキュリティユニットテストを定義すると、開発プロセスの初期にこれらの問題を識別し修正できます。

テストの自動化では、アプリケーションの要件と期待される機能性に基づいた、アプリケーション検証の目的別テストケースを使用します。自動化テストの結果は、生成されたテスト結果とそれに対応する期待される結果の比較に基づき、全体的なテストライフサイクルを促進します。リグレッションテストやユニットテストスイートなどのテスト手法は、自動化に最も適しています。セキュリティ特性のテストを自動化することで、ビルダーはセキュリティレビューを待つことなく、自動化されたフィードバックを得ることができます。静的または動的なコード分析の形式の自動化テストは、コード品質を改善し、開発ライフサイクルの初期での潜在的なソフトウェアの問題の検知に役立ちます。

一般的なアンチパターン:

- テストケースおよび自動化テストの結果のコミュニケーションの欠如。
- リリース直前のみでの自動化テストの実施。
- 頻繁に変更される要件に関する自動化テストケース。
- セキュリティテストの結果への対処方法に関するガイダンスの欠如。

このベストプラクティスを活用するメリット:

- システムのセキュリティ特性を評価するチームへの依存の低減。
- 複数のワークストリームにわたる一貫した検出結果による一貫性の向上。
- 本稼働ソフトウェアでのセキュリティ問題の低減。
- ソフトウェアの問題を早期に検知することにより、検知から修正までの時間を短縮。
- システム的な動作、または複数のワークストリームにわたって繰り返される動作の可視性の向上により、組織全体での改善を促進。

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

ソフトウェアを構築する際は、アプリケーションのビジネスロジックに基づいた機能性要件と、アプリケーションの信頼性、パフォーマンス、セキュリティにフォーカスした非機能性要件の両方をテストする、ソフトウェアテストのさまざまなメカニズムを採用します。

静的アプリケーションセキュリティテスト (SAST) は、ソースコードの異常なセキュリティパターンを分析し、脆弱性のあるコードを検知します。SAST はさまざまな既知のセキュリティ問題のテストにおいて、ドキュメント (要件仕様、設計文書、設計仕様) やアプリケーションのソースコードなどの、静的なインプットに依存します。静的コードアナライザーは、大規模なコードの迅速な分析に役立ちます。NIST の品質グループは、バイトコードスキャナーやバイナリコードスキャナーのオープンソースツールを含むソースコードセキュリティアナライザーの比較結果を提供しています。

潜在的な予期していない動作を識別するため、実行中のアプリケーションでテストを実施する、動的分析セキュリティテスト (DAST) によって静的テストを補完します。動的テストは、静的分析では検知できない潜在的な問題の検知に使用できます。コードリポジトリ、ビルド、パイプラインステージでテストを実施することで、コード内にあるさまざまなタイプの潜在的な問題をチェックすることができます。Amazon CodeWhisperer は、ビルダーの IDE 内で、セキュリティスキャンを含むコードのレコメンデーションを提供します。Amazon CodeGuru Reviewer は、アプリケーション開発中の重大な問題、セキュリティの問題、検知が難しいバグを識別し、コード品質の改善のためのレコメンデーションを提供します。

開発者のためのセキュリティワークショップでは、SAST と DAST のテスト手法を含むリリースパイプライン自動化のための AWS CodeBuild、AWS CodeCommit、AWS CodePipeline などの AWS 開発者ツールを使用します。

SDLC を進める中で、セキュリティチームと一緒に定期的なアプリケーションレビューを含む反復プロセスを確立します。リリース準備レビューの一部として、これらのセキュリティレビューで得たフィードバックに対処し検証します。これらのレビューにより、アプリケーションの堅固なセキュリティを確立でき、潜在的な問題に対処するための実行可能なフィードバックをビルダーに提供できます。

実装手順

- セキュリティテストを含む、一貫した IDE、コードレビュー、CI/CD ツールを実装します。
- 単に修正が必要な問題をビルダーに伝えるのではなく、SDLC のどこでパイプラインをブロックするのが適切かを考慮します。
- 開発者のためのセキュリティワークショップは、リリースパイプラインでの静的および動的テストの統合の例を提供します。

- 開発者の IDE と統合された Amazon CodeWhisperer、コミットのコードスキャン用の Amazon CodeGuru Reviewer などの自動化ツールを使用したテストまたはコード分析の実施は、適切なタイミングでビルダーにフィードバックを提供するのに役立ちます。
- AWS Lambda を使用した構築では、Amazon Inspector を使用して機能内のアプリケーションコードをスキャンできます。
- AWS CI/CD ワークショップは、AWS での CI/CD パイプライン構築の出発点を提供します。
- 自動化テストを CI/CD パイプラインに含める際は、ソフトウェアの問題の検知と修正を追跡するチケットシステムを使用します。
- 検出結果を生成するセキュリティテストでは、修正のガイダンスをリンクすることで、ビルダーのコード品質の改善を支援します。
- 自動化ツールからの結果を定期的に分析し、次の自動化、ビルダートレーニング、啓発活動の優先順位付けに役立てます。

リソース

関連するドキュメント:

- 継続的デリバリーと継続的なデプロイ

- AWS DevOps コンピテンシーパートナー
- アプリケーションセキュリティの AWS セキュリティコンピテンシーパートナー
- Choosing a Well-Architected CI/CD approach (Well-Architected CI/CD アプローチの選択)
- Monitoring CodeCommit events in Amazon EventBridge and Amazon CloudWatch Events (Amazon EventBridge と Amazon CloudWatch Events での CodeCommit イベントの監視)
- Secrets detection in Amazon CodeGuru Review (Amazon CodeGuru Review でのシークレット検知)
- Accelerate deployments on AWS with effective governance (効果的なガバナンスによる AWS でのデプロイの加速)
- AWS での安全なハンズオフデプロイメントの自動化

関連動画:

- Hands-off: Automating continuous delivery pipelines at Amazon (ハンズオフ: Amazon での継続的デリバリーパイプラインの自動化)
- Automating cross-account CI/CD pipelines (クロスアカウント CI/CD パイプラインの自動化)

関連する例:

- Industry awareness for developers (開発者向けの業界認識)
- AWS CodePipeline Governance (AWS CodePipeline ガバナンス) (GitHub)
- Security for Developers workshop (開発者のためのセキュリティワークショップ)
- AWS CI/CD Workshop (AWS CI/CD ワークショップ)

SEC11-BP03 定期的にペネトレーションテストを実施する

定期的にソフトウェアのペネトレーションテストを実施します。このメカニズムは、自動化されたテストや手動のコードレビューでは検知できない、ソフトウェアの潜在的な問題を識別するうえで役立ちます。

また、発見的コントロールの効率について把握するうえでも有効です。ペネトレーションテストでは、保護する必要があるデータを公開する、または予期したよりも広範なアクセス許可を付与するなど、ソフトウェアを予期しない方法で実行できるかどうかの確認を試みます。

期待される成果: ペネトレーションテストは、アプリケーションのセキュリティ特性の検知、修正、検証に使用されます。ソフトウェア開発ライフサイクル (SDLC) の一部として、定期的かつ計画的にペネトレーションテストを実施します。ペネトレーションテストでの検出結果は、ソフトウェアのリリース前に対処する必要があります。ペネトレーションテストでの検出結果を分析し、自動化によって検知できる問題があるかどうかを識別します。アクティブなフィードバックメカニズムを含む、定期的で反復可能なペネトレーションテストを実施することで、ビルダーへのガイダンスの提供やソフトウェア品質の向上に役立ちます。

一般的なアンチパターン:

- 既知のセキュリティの問題、または広く発生しているセキュリティの問題に対してのみペネトレーションテストを実施する。
- 依存するサードパーティツールやライブラリを除いてアプリケーションのペネトレーションテストを実施する。
- 実装されたビジネスロジックを評価せずに、パッケージセキュリティの問題のみについてペネトレーションテストを実施する。

このベストプラクティスを活用するメリット:

- リリース前のソフトウェアのセキュリティ特性についての信頼性の向上。
- 好ましいアプリケーションパターンを識別する機会を創出することによる、ソフトウェアのさらなる品質の向上。
- 開発サイクルの早期に、ソフトウェアのセキュリティ特性を改善するための自動化や追加のトレーニングを識別するフィードバックループの確立。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

ペネトレーションテストは、計画されたセキュリティ侵入シナリオを実行して、セキュリティコントロールを検知、修正、検証する、構造化されたセキュリティテストです。ペネトレーションテストは、アプリケーションの現在の設計とその依存関係に基づいてデータを収集する調査から始まります。セキュリティに特化した厳選されたテストシナリオの一覧が作成され実施されます。これらのテストの主な目的は、環境への意図しないアクセスやデータへの不正アクセスに悪用される可能性のある、アプリケーションのセキュリティの問題を発見することです。新しい機能をリリースしたり、機能や技術的な実装の大きな変更をアプリケーションに加えたりする際は、必ずペネトレーションテストを実施する必要があります。

ペネトレーションテストを実施するのに最適な開発ライフサイクルのステージを特定します。このテストは、システムの機能がリリースに十分に近く、さらに問題の修正に十分な時間を確保できる時期に行う必要があります。

実装手順

- コンテキストを維持するために、脅威モデルに基づいて、ペネトレーションテストのスコープを決定する構造化されたプロセスを確立します。
- ペネトレーションテストの実施に最適な開発ライフサイクルの時期を特定します。これは、アプリケーションで大きな変更が予定されておらず、問題の修正に十分な時間を確保できる時期である必要があります。
- ペネトレーションテストから期待される検出結果、および問題の修正に関する情報の取得について、ビルダーへのトレーニングを実施します。

- 一般的、または反復的なテストを自動化するためのツールを使用して、ペネトレーションテストプロセスの時間を短縮します。
- ペネトレーションテストでの検出結果を分析してシステム的なセキュリティの問題を識別し、このデータを使用して追加の自動化テストと継続的なビルダー教育に役立てます。

リソース

関連するベストプラクティス:

- SEC11-BP01 アプリケーションのセキュリティに関するトレーニングを実施する (p. 119)
- SEC11-BP02 開発およびリリースライフサイクル全体を通じてテストを自動化する (p. 121)

関連するドキュメント:

- AWS ペネトレーションテストでは、AWS でのペネトレーションテストの詳細なガイダンスを提供しています。
- Accelerate deployments on AWS with effective governance (効果的なガバナンスによる AWS でのデプロイの加速)
- AWS セキュリティコンピテンシーパートナー
- Modernize your penetration testing architecture on AWS Fargate (AWS Fargate でのペネトレーションテストアーキテクチャのモダナイゼーション)
- AWS Fault injection Simulator

関連する例:

- Automate API testing with AWS CodePipeline (AWS CodePipeline での API テストの自動化) (GitHub)
- Automated security helper (セキュリティヘルパーの自動化) (GitHub)

SEC11-BP04 手動のコードレビュー

作成するソフトウェアについて、手動のコードレビューを実施します。このプロセスは、コードを記述した人物が、コードの品質を確認する唯一のユーザーでないことを検証するうえで役立ちます。

期待される成果: 開発に手動のコードレビューステップを含めることで、開発中のソフトウェアの品質を改善したり、経験の浅いチームメンバーのスキルアップを支援したり、自動化を使用できる領域を識別したりすることができます。手動のコードレビューは、自動化ツールやテストによってサポートすることができます。

一般的なアンチパターン:

- デプロイ前にコードレビューを実施していない。
- コードの作成とレビューを同じ担当者が行っている。
- コードレビューの支援と調整に自動化を使用していない。
- コードレビューの前に、アプリケーションセキュリティについてビルダーをトレーニングしていない。

このベストプラクティスを活用するメリット:

- コード品質の向上。
- 共通のアプローチの再利用によるコード開発の一貫性の向上。

- ペネトレーションテストや後工程において検知される問題数の低減。
- チーム内での知識の移転の改善。

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

全体的なコード管理フローの一部として、レビューステップを実装します。詳細は、分岐、プルリクエスト、マージで使用するアプローチによって異なります。それらのアプローチでは AWS CodeCommit、または GitHub、GitLab、Bitbucket のようなサードパーティソリューションを使用する場合があります。使用する手法にかかわらず、プロセスを本稼働環境にデプロイする前に、プロセスのレビューが必要であることを認識することが重要です。Amazon CodeGuru Reviewer などのツールを使用すると、コードレビュープロセスを簡単に調整することができます。

実装手順

- コード管理フローの一部として手動のレビューステップを実装し、次に進む前にこのレビューを実施します。
- コードレビューの管理と支援のために Amazon CodeGuru Reviewer の使用を検討します。
- コードを次のステージに進める前にコードレビューの完了を必須とする承認フローを実装します。
- 手動のコードレビューで発見された問題を自動的に検知するプロセスがないか確認します。
- コード開発プラクティスに沿って手動のコードレビューを統合します。

リソース

関連するベストプラクティス:

- SEC11-BP02 開発およびリリースライフサイクル全体を通じてテストを自動化する (p. 121)

関連するドキュメント:

- Working with pull requests in AWS CodeCommit repositories (AWS CodeCommit でのプルリクエストの使用)
- Working with approval rule templates in AWS CodeCommit (AWS CodeCommit での承認ルールテンプレートの使用)
- About pull requests in GitHub (GitHub でのプルリクエストについて)
- Automate code reviews with Amazon CodeGuru Reviewer (Amazon CodeGuru Reviewer を使用したコードレビューの自動化)
- Automating detection of security vulnerabilities and bugs in CI/CD pipelines using Amazon CodeGuru Reviewer CLI (Amazon CodeGuru Reviewer CLI を使用した CI/CD パイプラインでのセキュリティ脆弱性とバグの検知の自動化)

関連動画:

- Continuous improvement of code quality with Amazon CodeGuru (Amazon CodeGuru を使用したコード品質の継続的な改善)

関連する例:

- Security for Developers workshop (開発者のためのセキュリティワークショップ)

SEC11-BP05 パッケージと依存関係のサービスを一元化する

ビルダーチームに対して、ソフトウェアパッケージとその他の依存関係を取得するための一元化されたサービスを提供します。これにより、記述するソフトウェアに含まれる前に、パッケージの検証が可能になります。また、これは組織で使用中のソフトウェアを分析するためのデータソースとなります。

期待される成果: ソフトウェアは、作成されたコードと、一連の他のソフトウェアパッケージによって構成されます。このため、JSON パーサーや暗号化ライブラリなどの繰り返し使用される機能を容易に実装することができます。これらのパッケージのソースと依存関係を論理的に一元化することで、パッケージが使用される前に、そのパッケージのプロパティを検証するためのメカニズムをセキュリティチームに提供することができます。またこのアプローチによって、既存のパッケージの変更による予期しない問題のリスクや、インターネット上の任意のパッケージをビルダーチームが使用することによるリスクを低減できます。このアプローチを手動および自動化されたテストフローと組み合わせて使用することで、開発中のソフトウェアの品質についての信頼性を高めることができます。

一般的なアンチパターン:

- インターネット上の任意のリポジトリからパッケージを取得する。
- ビルダーに提供する前に、新しいパッケージをテストしていない。

このベストプラクティスを活用するメリット:

- 構築中のソフトウェアで使用されるパッケージのより良い理解。
- 誰が、どのようなパッケージを使用しているかを把握することで、パッケージの更新が必要な場合に、ワークロードチームに通知することができる。
- 問題のあるパッケージがソフトウェアで使用されるリスクの低減。

このベストプラクティスが確立されていない場合のリスクレベル: 中

実装のガイダンス

ビルダーが簡単に使用できるように、パッケージと依存関係の一元化されたサービスを提供します。一元化されたサービスは、実装されるモノリシックシステムとしてではなく、論理的に一元化します。このアプローチを使用することで、ビルダーのニーズに合ったサービスを提供できます。更新が必要な場合や新しい要件が発生した際に新しいパッケージをリポジトリに追加する効率的な方法を実装します。AWS CodeArtifact または類似する AWS パートナーソリューションなどの AWS サービスは、このような機能を提供します。

実装手順:

- ソフトウェアが開発されているすべての環境で利用可能な、論理的に一元化されたリポジトリサービスを実装します。
- AWS アカウント のベンディングプロセスの一部として、リポジトリへのアクセスを含めます。
- パッケージをリポジトリに公開する前に、パッケージの自動化テストを構築します。
- 最も大きな変更が発生する頻繁に使用されるパッケージ、言語、チームに関するメトリクスを維持します。
- 新しいパッケージのリクエストとフィードバックの提供を行うための自動化されたメカニズムをビルダーチームに提供します。
- リポジトリ内のパッケージを定期的にスキャンして、新しく発見された問題の潜在的な影響を識別します。

リソース

関連するベストプラクティス:

- SEC11-BP02 開発およびリリースライフサイクル全体を通じてテストを自動化する (p. 121)

関連するドキュメント:

- Accelerate deployments on AWS with effective governance (効果的なガバナンスによる AWS でのデプロイの加速)
- Tighten your package security with CodeArtifact Package Origin Control toolkit (CodeArtifact Package Origin Control ツールキットを使用してパッケージのセキュリティを高める)
- Detecting security issues in logging with Amazon CodeGuru Reviewer (Amazon CodeGuru Reviewer を使用してセキュリティの問題を口グで検知する)
- Supply chain Levels for Software Artifacts (SLSA) (ソフトウェアアーティファクトのサプライチェーンレベル)

関連動画:

- Proactive security: Considerations and approaches (プロアクティブなセキュリティ: 考慮事項とアプローチ)
- The AWS Philosophy of Security (re:Invent 2017) (AWS のセキュリティ哲学、re:Invent 2017)
- When security, safety, and urgency all matter: Handling Log4Shell (セキュリティ、安全性、緊急性のすべてが問題になるとき: Log4Shell への対応)

関連する例:

- Multi Region Package Publishing Pipeline (マルチリージョンパッケージ公開パイプライン) (GitHub)
- Publishing Node.js Modules on AWS CodeArtifact using AWS CodePipeline (AWS CodePipeline を使用した Node.js モジュールの AWS CodeArtifact への公開) (GitHub)
- AWS CDK Java CodeArtifact Pipeline Sample (AWS CDK Java CodeArtifact パイプラインの例) (GitHub)
- Distribute private .NET NuGet packages with AWS CodeArtifact (AWS CodeArtifact を使用したプライベート .NET NuGet パッケージの配布) (GitHub)

SEC11-BP06 ソフトウェアをプログラムでデプロイする

可能な場合は、ソフトウェアのデプロイをプログラムで行います。この手法により、デプロイに失敗したり、人的エラーにより予期しない問題が発生したりする可能性を低減できます。

期待される成果: AWS クラウド でセキュアに構築するためには、原則としてデータへの人的関与を排除します。この原則には、ソフトウェアのデプロイ方法も含まれます。

ソフトウェアのデプロイにおいて人的な依存を排除することで、テスト済みのソフトウェアがデプロイされ、デプロイが常に一貫して実行されるという信頼性を大幅に高めることができます。異なる環境で動作させるために、ソフトウェアを変更する必要はありません。12 要素のアプリケーション開発の原則、具体的には構成の外部化の原則を用いることで、変更を必要とすることなく、複数の環境に同じコードをデプロイすることができます。暗号化を用いたソフトウェアパッケージの署名は、環境間で変更がないことを証明するための便利な手法です。このアプローチによって、変更プロセスでのリスクを低減し、ソフトウェアリリースの一貫性を向上させることができます。

一般的なアンチパターン:

- 本稼働環境にソフトウェアを手動でデプロイする。
- 環境に合わせて手動でソフトウェアに変更を加える。

このベストプラクティスを活用するメリット:

- ソフトウェアリリースプロセスの信頼性の向上。
- 変更の失敗がビジネスの機能性に与えるリスクの低減。
- 変更リスクの低減によるリリース頻度の向上。
- デプロイ中に予期しないイベントが発生した場合の自動ロールバック機能。
- テスト済みのソフトウェアとデプロイされたソフトウェアが同じであることを暗号化によって証明する能力。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

持続的な人的アクセスを環境から排除するために AWS アカウント 構造を構築し、CI/CD ツールを使用してデプロイを実施します。環境固有のデータを AWS Systems Manager Parameter Store などの外部ソースから取得するようにアプリケーションを設計します。テスト後にパッケージに署名し、デプロイ中にこれらの署名を検証します。アプリケーションコードをプッシュするように CI/CD パイプラインを設定し、Canary を使用してデプロイの成功を確認します。AWS CloudFormation または AWS CDK などのツールを使用してインフラストラクチャを定義し、AWS CodeBuild と AWS CodePipeline を使用して CI/CD のオペレーションを実行します。

実装手順

- 明確に定義された CI/CD パイプラインを構築して、デプロイプロセスを合理化します。
- AWS CodeBuild と AWS Code Pipeline を使用して、パイプラインへのセキュリティテストの統合を容易にする CI/CD 機能を提供します。
- ホワイトペーパー「Organizing Your AWS Environment Using Multiple Accounts (複数のアカウントを使用した AWS 環境の組織化)」で説明している環境の分離に関するガイダンスに従います。
- 本稼働ワークロードが実行されている環境への持続的な人的アクセスがないことを確認します。
- 構成データの外部化をサポートするようアプリケーションを設計します。
- ブルー/グリーンデプロイモデルを使用したデプロイを検討します。
- Canary を実装してソフトウェアのデプロイの成功を検証します。
- AWS Signer または AWS Key Management Service (AWS KMS) などの暗号化ツールを使用して、デプロイするソフトウェアパッケージの署名と検証を行います。

リソース

関連するベストプラクティス:

- SEC11-BP02 開発およびリリースライフサイクル全体を通じてテストを自動化する (p. 121)

関連するドキュメント:

- AWS CI/CD Workshop (AWS CI/CD ワークショップ)
- Accelerate deployments on AWS with effective governance (効果的なガバナンスによる AWS でのデプロイの加速)

- 安全なハンズオフデプロイメントの自動化
- Code signing using AWS Certificate Manager Private CA and AWS Key Management Service asymmetric keys (AWS Certificate Manager Private CA および AWS Key Management Service 非対称キーを使用したコードの署名)
- Code Signing, a Trust and Integrity Control for AWS Lambda (コード署名、AWS Lambda の信頼性および整合性のコントロール)

関連動画:

- Hands-off: Automating continuous delivery pipelines at Amazon (ハンズオフ: Amazon での継続的デリバリーパイプラインの自動化)

関連する例:

- Blue/Green deployments with AWS Fargate (AWS Fargate を使用したブルー/グリーンデプロイ)

SEC11-BP07 パイプラインのセキュリティ特性を定期的に評価する

アクセス許可の分離に特に注意しながら、Well-Architected セキュリティの柱の原則をパイプラインに適用します。パイプラインインフラストラクチャのセキュリティ特性を定期的に評価します。パイプラインのセキュリティを効率的に管理することにより、パイプラインを通過するソフトウェアのセキュリティを確保することができます。

期待される成果: ソフトウェアの構築とデプロイに使用するパイプラインは、環境内の他のワークロードと同じ推奨プラクティスに従う必要があります。パイプラインに実装されるテストは、テストを使用するビルダーによって編集できないようにする必要があります。パイプラインへのアクセス許可は、実施するデプロイに必要なものだけに制限し、誤った環境へのデプロイを防ぐための安全対策を実装する必要があります。パイプラインは長期的な認証情報に依存せず、構築環境の整合性を検証できるようにステータスを送信する必要があります。

一般的なアンチパターン:

- ビルダーが回避可能なセキュリティテスト。
- デプロイパイプラインへの広範すぎるアクセス許可。
- 入力を検証するように設定されていないパイプライン。
- CI/CD インフラストラクチャに関連付けられているアクセス許可を定期的にレビューしていない。
- 長期的な認証情報、またはハードコード化された認証情報の使用。

このベストプラクティスを活用するメリット:

- パイプラインをとおして構築およびデプロイされるソフトウェアの整合性についての信頼性の向上。
- 不審なアクティビティが存在するデプロイを停止する能力。

このベストプラクティスが確立されていない場合のリスクレベル: 高

実装のガイダンス

IAM ロールをサポートするマネージド CI/CD サービスから始めることで、認証情報の流出リスクを低減することができます。セキュリティの柱の原則を CI/CD パイプラインインフラストラクチャに適用すると、

セキュリティの改善が可能な領域を判断するのに役立ちます。AWS デプロイパイプラインリファレンスアーキテクチャに沿うことは、CI/CD 環境の構築の良い起点となります。パイプラインの実装を定期的にレビューし、予期しない動作のログを分析すると、ソフトウェアのデプロイで使用されているパイプラインの使用パターンの理解に役立ちます。

実装手順

- AWS デプロイパイプラインリファレンスアーキテクチャを起点とします。
- AWSIAM Access Analyzer を使用して、プログラム的にパイプラインの最小特権の IAM ポリシーを生成することを検討します。
- 予期しないアクティビティや異常なアクティビティを検知するために、監視と警告をパイプラインに実装します。AWS のマネージドサービスである Amazon EventBridge を使用すると、AWS Lambda や Amazon Simple Notification Service (Amazon SNS) などのターゲットにデータをルーティングすることができます。

リソース

関連するドキュメント:

- AWS Deployment Pipelines Reference Architecture (AWS デプロイパイプラインリファレンスアーキテクチャ)
- Monitoring AWS CodePipeline (AWS CodePipeline のモニタリング)
- Security best practices for AWS CodePipeline (AWS CodePipeline のセキュリティのベストプラクティス)

関連する例:

- DevOps monitoring dashboard (DevOps モニタリングダッシュボード) (GitHub)

SEC11-BP08 ワークロードチームにセキュリティのオーナーシップを根付かせるプログラムを構築する

ビルダーチームが、作成するソフトウェアに関するセキュリティの決定を行えるようにするプログラムやメカニズムを構築します。セキュリティチームは、引き続きレビュー中にこれらの決定を検証する必要がありますが、ビルダーチームにセキュリティのオーナーシップを根付かせることで、より迅速でセキュアなワークロードの構築が可能になります。また、このメカニズムは構築するシステムの運用に良い影響を与える、オーナーシップのカルチャーを育みます。

期待される成果: セキュリティのオーナーシップと意思決定をビルダーチームに根付かせるには、セキュリティについての考え方についてビルダーにトレーニングを実施したり、ビルダーチームにセキュリティスタッフを配置または連携させてトレーニングを補強したりします。いずれのアプローチも適切で、チームは開発サイクルの初期にセキュリティに関する質の高い意思決定を行えるようになります。このオーナーシップモデルは、アプリケーションセキュリティのトレーニングを前提にしています。特定のワークロードの脅威モデルから始めると、適切なコンテキストでの設計思考にフォーカスするのに役立ちます。セキュリティにフォーカスしたビルダーコミュニティの構築、またはセキュリティエンジニアグループとビルダーチームを連携させることの別の利点として、ソフトウェアの作成についてより深い理解を得られることが挙げられます。この理解は自動化に関する次の改善領域の特定に役立ちます。

一般的なアンチパターン:

- セキュリティ設計に関するすべての意思決定をセキュリティチームに委ねる。
- 開発プロセスの十分に早い段階でセキュリティ要件を策定していない。
- プログラムの運用に関して、ビルダーとセキュリティスタッフからのフィードバックを入手していない。

このベストプラクティスを活用するメリット:

- セキュリティレビューを完了するまでの時間の短縮。
- セキュリティレビューのステージでようやく検知されるセキュリティ問題の削減。
- 作成されるソフトウェアの全体的な品質の向上。
- システム的な問題、または価値の高い改善領域を識別し、理解する機会の創出。
- セキュリティレビューでの結果に基づく再作業量の削減。
- セキュリティ機能に関する認識の改善。

このベストプラクティスが確立されていない場合のリスクレベル: 低

実装のガイダンス

SEC11-BP01 アプリケーションのセキュリティに関するトレーニングを実施する (p. 119) のガイダンスから始めます。その後、組織に最も適切と思われるプログラムの運用モデルを識別します。主な 2 つのパターンは、ビルダーのトレーニングの実施、またはビルダーチームへのセキュリティスタッフの配置です。初期アプローチの決定後、組織に適したモデルであるかどうかを検証するために、単一または小規模のワークロードチームに対してテストを実施します。プログラムの実行と成功には、組織のビルダーおよびセキュリティ部門のリーダーシップによるサポートが役立ちます。このプログラムを構築する際は、プログラムの価値を計測できるメトリクスを選択することが重要です。AWS がこの課題にどのようにアプローチしたかを学ぶことは有益です。ベストプラクティスは、主に組織の変化と文化にフォーカスしています。ビルダーとセキュリティのコミュニティ間のコラボレーションをサポートするツールを使用します。

実装手順

- アプリケーションのセキュリティに関するビルダーのトレーニングから始めます。
- ビルダーの教育のためのコミュニティとオンボーディングプログラムを作成します。
- プログラムの名称を決定します。よく使用される名称は、ガーディアン、チャンピオン、アドボケイトなどです。
- ビルダートレーニング、セキュリティエンジニアの配置、セキュリティスタッフとの連携、という三択から、使用するモデルを選択します。
- セキュリティ部門、ビルダー部門、および他の潜在的な関連部門からプロジェクトスポンサーを特定します。
- プログラムへの参加人数、レビューに要した時間、ビルダーやセキュリティスタッフからのフィードバックを計測するメトリクスを追跡します。これらのメトリクスを使用して、改善を行います。

リソース

関連するベストプラクティス:

- SEC11-BP01 アプリケーションのセキュリティに関するトレーニングを実施する (p. 119)
- SEC11-BP02 開発およびリリースライフサイクル全体を通じてテストを自動化する (p. 121)

関連するドキュメント:

- How to approach threat modeling (脅威モデリングにアプローチする方法)
- How to think about cloud security governance (クラウドのセキュリティガバナンスをどのように考えるか)

関連動画:

- Proactive security: Considerations and approaches (プロアクティブなセキュリティ: 考慮事項とアプローチ)

まとめ

セキュリティは、継続的な取り組みです。発生したインシデントは、アーキテクチャのセキュリティを向上させるための機会として扱う必要があります。強力な ID コントロール、セキュリティイベントへの対応の自動化、複数レベルでのインフラストラクチャの保護、暗号化による適切に分類されたデータの管理により、あらゆる組織に実装する必要がある多層防御が可能になります。このホワイトペーパーで説明したプログラム関数と AWS の機能やサービスがあれば、このような取り組みもより簡単に実現できます。

AWS は、ビジネス価値を実現しながら、情報、システム、アセットを保護するアーキテクチャの構築と運用を支援します。

寄稿者

本ドキュメントは、次の人物および組織が寄稿しました。

- Sarita Dharankar、セキュリティピラーリード、Well-Architected、Amazon Web Services
- Adam Cerini、シニアソリューションアーキテクト、Amazon Web Services
- Bill Shinn、シニアプリンシパル、CISO オフィス、Amazon Web Services
- Brigid Johnson、シニアソフトウェア開発マネージャー、AWS Identity、Amazon Web Services
- Byron Pogson、シニアソリューションアーキテクト、Amazon Web Services
- Charlie Hammell、プリンシパルエンタープライズアーキテクト、Amazon Web Services
- Darran Boyd、プリンシパルセキュリティソリューションアーキテクト、金融サービス、Amazon Web Services
- Dave Walker、プリンシパルスペシャリストソリューションアーキテクト、セキュリティとコンプライアンス、Amazon Web Services
- John Formento、シニアソリューションアーキテクト、Amazon Web Services
- Paul Hawkins、プリンシパル、CISO オフィス、Amazon Web Services
- Sam Elmalak、シニアテクノロジーリーダー、Amazon Web Services
- Pat Gaw、プリンシパルセキュリティコンサルタント、Amazon Web Services
- Daniel Begimher、シニアコンサルタント、セキュリティ、Amazon Web Services
- Danny Cortegaca、シニアセキュリティソリューションアーキテクト、Amazon Web Services
- Ana Malhotra、セキュリティソリューションアーキテクト、Amazon Web Services
- Debashis Das、プリンシパル、CISO オフィス、Amazon Web Services
- Reef Dsouza、プリンシパルソリューションアーキテクト、Amazon Web Services
- Brad Burnett、アイデンティティ担当セキュリティソリューションアーキテクト、Amazon Web Services
- Anna McAbee、脅威検出およびインシデント対応担当シニアセキュリティソリューションアーキテクト、Amazon Web Services
- Jason Garman、プリンシパルセキュリティソリューションアーキテクト、Amazon Web Services

その他の資料

追加情報については、次の資料を参照してください。

- AWS Well-Architected Framework のホワイトペーパー
- AWS アーキテクチャセンター

改訂履歴

このホワイトペーパーの更新に関する通知を受け取るには、RSS フィードをサブスクライブしてください。

変更	説明	日付
ベストプラクティスガイダンスの更新 (p. 137)	ベストプラクティスを更新し、以下の領域に関するガイダンスを追加。「ワークロードを安全に運用する」、「転送中のデータの保護」。	December 6, 2023
ベストプラクティスガイダンスの更新 (p. 137)	「インシデント対応」のガイダンスとベストプラクティスを大幅に更新。 「準備」の複数のベストプラクティスを更新。「インシデント対応」に 2 つの新しい領域、「オペレーション」、「インシデント後のアクティビティ」を追加。新しいベストプラクティス「SEC10-BP08 インシデントから学ぶためのフレームワークを確立する」を追加。	October 3, 2023
ベストプラクティスガイダンスの更新 (p. 137)	ベストプラクティスを更新し、以下の領域に関する新しいガイダンスを追加。「準備」、「シミュレーション」。	July 13, 2023
新しいフレームワークの更新 (p. 137)	規範ガイダンスを使用してベストプラクティスを更新、および新しいベストプラクティスを追加。アプリケーションのセキュリティ (AppSec) の新しいベストプラクティス領域を追加。	April 10, 2023
ホワイトペーパーの更新 (p. 137)	新しい実装ガイダンスを使用してベストプラクティスを更新。	December 15, 2022
ホワイトペーパーの更新 (p. 137)	ベストプラクティスに加筆し、改善計画を追加。	October 20, 2022
マイナーな更新 (p. 137)	最新のベストプラクティスを反映して IAM 情報を更新。	June 28, 2022
マイナーな更新 (p. 137)	AWS PrivateLink 情報を追加し、壊れたリンクを修正。	May 19, 2022
マイナーな更新 (p. 92)	AWS PrivateLink を追加。	May 6, 2022
マイナーな更新 (p. 137)	インクルーシブでない表現を削除。	April 22, 2022

マイナーな更新 (p. 137)	VPC Network Access Analyzer に関する情報を追加。	February 2, 2022
マイナーな更新 (p. 1)	イントロダクションに持続可能性の柱を追加。	December 2, 2021
マイナーな更新 (p. 137)	壊れたリンクを修正。	May 27, 2021
マイナーな更新 (p. 137)	全体を通した編集の変更。	May 17, 2021
メジャーアップデート (p. 137)	ガバナンスに関するセクションを追加、さまざまなセクションに詳細を追加、全体を通して新機能やサービスを追加。	May 7, 2021
マイナーな更新 (p. 137)	リンクを更新。	March 10, 2021
マイナーな更新 (p. 137)	壊れたリンクを修正。	July 15, 2020
新しいフレームワークの更新 (p. 137)	アカウント、ID、アクセス権限の管理に関するガイダンスを更新。	July 8, 2020
新しいフレームワークの更新 (p. 137)	すべての分野、新しいベストプラクティス、サービス、機能のアドバイスを拡張する更新。	April 30, 2020
ホワイトペーパーの更新 (p. 137)	新しい AWS のサービスと機能を反映するための更新とリファレンスの更新。	July 1, 2018
ホワイトペーパーの更新 (p. 137)	システムセキュリティの設定とメンテナンスのセクションを更新し、新しい AWS のサービスと機能を反映。	May 1, 2017
初版発行 (p. 137)	信頼性の柱 - AWS Well-Architected フレームワークを公開。	November 1, 2016

注意

お客様は、この文書に記載されている情報を独自に評価する責任を負うものとします。本書は、(a) 情報提供のみを目的とし、(b) AWS の現行製品と慣行について説明しており、これらは予告なしに変更されることがあり、(c) AWS およびその関連会社、サプライヤーまたはライセンサーからの契約上の義務や保証をもたらすものではありません。AWS の製品やサービスは、明示または暗示を問わず、一切の保証、表明、条件なしに「現状のまま」提供されます。お客様に対する AWS の責任は、AWS 契約により規定されます。本書は、AWS とお客様の間で締結されるいかなる契約の一部でもなく、その内容を修正するものでもありません。