aws

ユーザーガイド

Amazon Inspector

Amazon Inspector: ユーザーガイド

Table of Contents

Amazon Inspector とは

Amazon Inspector は、ソフトウェアの脆弱性や意図しないネットワークへの露出について AWS ワークロードを継続的にスキャンする脆弱性管理サービスです。Amazon Inspector は、実行中の Amazon EC2 インスタンス、Amazon Elastic Container Registry (Amazon ECR) のコンテナイメージ、および既知のソフトウェアの脆弱性や意図しないネットワークへの露出について AWS Lambda 関数を自動的に検出してスキャンします。

Amazon Inspector は、ソフトウェアの脆弱性やネットワーク設定の問題を発見すると、検出結果を作成します。検出結果は脆弱性を説明し、影響を受けるリソースを特定し、脆弱性の重要度を評価し、修正ガイダンスを提供します。Amazon Inspector コンソールを使用して検出結果を分析することも、他の AWS のサービスを使用して検出結果を表示し、処理することもできます。詳細については、「Amazon Inspector の検出結果について」を参照してください。

トピック

Amazon Inspector の特徴

複数の Amazon Inspector アカウントを一元管理

AWS 環境に複数のアカウントがある場合は、 AWS Organizations を使用して 1 つのアカウントで環境を一元管理できます。この方法を使用することで、Amazon Inspector の委任された管理者アカウントとしてアカウントを指定できます。

Amazon Inspector は、ワンクリックで組織全体にアクティブ化できます。さらに、今後新たなメンバーが組織に入るたびに、自動的にサービスをアクティブ化することもできます。Amazon Inspector の委任された管理者アカウントは、組織のメンバーの検出結果データと特定の設定を管理できます。これには、すべてのメンバーアカウントの集約された結果の詳細の表示、メンバーアカウントのスキャンのアクティブ化または非アクティブ化、 AWS 組織内のスキャンされたリソースの確認が含まれます。

環境を継続的にスキャンして、脆弱性やネットワークの露出がないかを確認します。

Amazon Inspector を使用すれば、評価スキャンを手動でスケジュールまたは設定する必要はありません。Amazon Inspector は、対象となるリソースを自動的に検出し、スキャンを開始しま

す。Amazon Inspector は、EC2 インスタンスへの新しいパッケージのインストール、パッチのインストール、リソースに影響を与える新しい共通脆弱性識別子 (CVE) が発行された場合など、新しい脆弱性をもたらす可能性のある変更に対応してリソースを自動的に再スキャンすることで、リソースのライフサイクル全体を通じて引き続き環境を評価します。従来のセキュリティスキャンソフトウェアとは異なり、Amazon Inspector はお客様のフリートのパフォーマンスへの影響を最小限に抑えます。

脆弱性またはオープンネットワークパスが特定されると、Amazon Inspector は調査可能な検出結果を生成します。検出結果には、脆弱性、影響を受けるリソース、および修復に関する推奨事項に関する包括的な詳細が含まれます。検出結果を適切に修復すると、Amazon Inspector は自動的に修正を検出し、検出結果を終了します。

Amazon Inspector のリスクスコアを使用して脆弱性を正確に評価

Amazon Inspector はスキャンを通じて環境に関する情報を収集し、その環境に合わせて特別に調整された重要度スコアを提供します。Amazon Inspector は、脆弱性の全国脆弱性データベース (NVD) の基本スコアを構成するセキュリティメトリクスを調べ、コンピューティング環境に応じて調整します。たとえば、脆弱性がネットワーク上で悪用可能であるが、インスタンスからインターネットへのオープンネットワークパスが利用できない場合、サービスは Amazon EC2 インスタンスの検出結果の Amazon Inspector スコアを下げる可能性があります。このスコアは CVSS 形式で、NVD が提供する共通脆弱性スコアリングシステム (CVSS) の基本スコアを修正したものです。

Amazon Inspector ダッシュボードを使用して影響の大きい検出結果を特定する

Amazon Inspector ダッシュボードには、環境全体から得られた検出結果の概要が表示されます。ダッシュボードから、検出結果の詳細にアクセスできます。ダッシュボードには、環境内のスキャン範囲、最も緊急の検出結果、および最も多くの検出結果が得られたリソースに関する効率的な情報が表示されます。Amazon Inspector ダッシュボードのリスクベースの修復パネルには、最も多くのインスタンスとイメージに影響する検出結果が表示されます。このパネルでは、環境に最も大きな影響を与える検出結果の特定、検出結果の詳細確認、および推奨される解決策の確認がより容易になります。

カスタマイズ可能なビューを使用して検出結果を管理

ダッシュボードに加えて、Amazon Inspector コンソールには検出結果ビューがあります。このページでは、環境に関する検出結果をリスト化し、個別の検出結果の詳細を提供します。検出結果は、カテゴリまたは脆弱性タイプ別にグループ化して表示できます。各ビューでは、フィルターを使用して結果をさらにカスタマイズできます。フィルターを使用して、不要な検出結果をビューから隠す非表示ルールを作成することもできます。

フィルターと抑制ルールを使用して、すべての検出結果またはカスタマイズされた結果の選択を表示する結果レポートを生成できます。レポートは CSV 形式または JSON 形式で生成できます。

他のサービスおよびシステムを用いた検出結果のモニタリングと処理

他の サービスやシステムとの統合をサポートするために、Amazon Inspector は検出結果を検出結果イベントとして Amazon に発行します EventBridge。 EventBridge は、検出結果を AWS Lambda 関数や Amazon Simple Notification Service (Amazon SNS) トピックなどのターゲットにルーティングできるサーバーレスイベントバスサービスです。を使用すると EventBridge、既存のセキュリティおよびコンプライアンスワークフローの一部として、結果をほぼリアルタイムでモニタリングおよび処理できます。

AWS Security Hub をアクティブ化すると、Amazon Inspector は検出結果を Security Hub にも公開します。Security Hub は、 AWS 環境全体のセキュリティ体制を包括的に把握し、セキュリティ業界標準とベストプラクティスに照らして環境をチェックするのに役立つサービスです。Security Hub を使用すると、 AWS内の組織のセキュリティ体制の広範な分析の一部として、検出結果をより簡単にモニタリングおよび処理できます。

Amazon Inspector へのアクセス

Amazon Inspector はほとんどの で使用できます AWS リージョン。Amazon Inspector が現在利用可能な リージョンの一覧については、Amazon Web Services 全般リファレンス の「Amazon Inspector のエンドポイントとクォータ」を参照してください。 AWS リージョンの詳細については、「Amazon Web Services General Reference」の「Managing AWS リージョン」を参照してください。各リージョンで、次のいずれかの方法で Amazon Inspector を使用できます。

「AWS マネジメントコンソール」

AWS Management Console は、 リソースの作成と管理 AWS に使用できるブラウザベースのインターフェイスです。そのコンソールの一部として、Amazon Inspector コンソールは Amazon Inspector アカウントとリソースへのアクセスを提供します。Amazon Inspector タスクは、Amazon Inspector コンソールから実行できます。

AWS コマンドラインツール

AWS コマンドラインツールを使用すると、システムのコマンドラインでコマンドを発行して Amazon Inspector タスクを実行できます。コマンドラインを使用すると、コンソールを使用するよりも高速で便利になります。コマンドラインツールは、タスクを実行するスクリプトを作成する場合にも便利です。

AWS には、 AWS Command Line Interface （ AWS CLI) と の 2 セットのコマンドラインツールが用意されています AWS Tools for PowerShell。のインストールと使用の詳細については AWS CLI、AWS コマンドラインインターフェイスユーザーガイドを参照してください。Tools for のインストールと使用の詳細については PowerShell、「 AWS Tools for PowerShell ユーザーガイド」を参照してください。

AWS SDK

AWS はSDKs を提供します。SDK は、Amazon Inspector および他の AWS のサービスへの便利なプログラムによるアクセスを提供します。SDK は、暗号署名によるリクエスト、エラーの管理、リクエストの自動再試行などのタスクも処理します。 AWS SDKsで構築するツール AWS」を参照してください。

Amazon Inspector REST API

Amazon Inspector REST API は、Amazon Inspector アカウントとリソースへの包括的なプログラムによるアクセスを提供します。この API を使用すると、HTTPS リクエストを Amazon Inspector に直接送信できます。ただし、 AWS コマンドラインツールや SDKsを使用するには、アプリケーションがリクエストに署名するハッシュの生成など、低レベルの詳細を処理する必要があります。

Amazon Inspector の開始方法

このチュートリアルでは、Amazon Inspector の実践的入門を示します。

ステップ 1 では、スタンドアロンアカウントの Amazon Inspector スキャンをアクティブ化するか、マルチアカウント環境で を使用する Amazon Inspector の委任管理者としてアクティブ化 AWS Organizations します。

ステップ 2 では、コンソールに表示される Amazon Inspector の検出結果を理解する方法について説明します。

> **Note**
>
> このチュートリアルでは、現在の でタスクを完了します AWS リージョン。他のリージョンで Amazon Inspector をセットアップするには、それぞれのリージョンで以下の手順を実行する必要があります。

トピック

開始する前に

Amazon Inspector は、Amazon EC2 インスタンス、Amazon ECR コンテナイメージ、および AWS Lambda 関数を継続的にスキャンして、ソフトウェアの脆弱性や意図しないネットワークへの露出を検出する脆弱性管理サービスです。

Amazon Inspector をアクティブ化する前に、次の点に注意してください。

- Amazon Inspector はリージョナルサービスであり、データはサービスを使用する AWS リージョン に保存されます。このチュートリアルで実行する設定手順は、Amazon Inspector でモニタリング AWS リージョン する各 で繰り返す必要があります。
- Amazon Inspector では、Amazon EC2 インスタンス、Amazon ECR コンテナイメージ、および AWS Lambda 関数スキャンを柔軟にアクティブ化できます。スキャンタイプは、Amazon

Inspector コンソールのアカウント管理ページから、または Amazon Inspector API を使用して管理できます。

- Amazon Inspector は、Amazon EC2 Systems Manager (SSM) エージェントがインストールされアクティブ化になっている場合にのみ、EC2 インスタンスの共通脆弱性識別子 (CVE) データを提供できます。このエージェントは多くの EC2 インスタンスにプリインストールされていますが、手動でアクティブ化する必要がある場合があります。SSM Agent のステータスにかかわらず、すべての EC2 インスタンスがスキャンされ、ネットワークの露出の問題がないかどうかが確認されます。Amazon EC2 のスキャンを設定する方法については、「Amazon EC2 インスタンスのスキャン」を参照してください。Amazon ECR と AWS Lambda 関数スキャンでは、エージェントを使用する必要はありません。
- の管理者権限を持つ IAM ユーザー ID は、Amazon Inspector を有効に AWS アカウント できます。データ保護の目的で、認証情報を保護し、 AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。そうすることで、各ユーザーには Amazon Inspector の管理に必要なアクセス許可のみが与えられます。Amazon Inspector を有効にするために必要なアクセス許可の詳細については、「AWS マネージドポリシー: AmazonInspector2FullAccess」を参照してください。
- いずれかのリージョンで Amazon Inspector をはじめてアクティブ化すると、`AWSServiceRoleForAmazonInspector2` というアカウントのグローバルなサービスリンクロールが作成されます。このロールには、Amazon Inspector が脆弱性検出結果を生成するために Amazon Inspector がソフトウェアパッケージの詳細を収集し、Amazon VPC 設定を分析することを可能にする許可と信頼ポリシーが含まれます。詳細については、「Amazon Inspector でのサービスにリンクされたロールの使用」を参照してください。サービスにリンクされたロールの詳細については、「サービスリンクロールの使用」を参照してください。

ステップ 1: Amazon Inspector をアクティブ化する

Amazon Inspector を使用する最初のステップは、お客様の AWS アカウントでそれをアクティブ化することです。Amazon Inspector のスキャンタイプをアクティブ化すると、Amazon Inspector はただちにすべての対象リソースの検出とスキャンを開始します。

一元化された管理者アカウントを使用して組織内の複数のアカウントの Amazon Inspector を管理する場合は、Amazon Inspector の委任された管理者を割り当てる必要があります。以下のいずれかのオプションを選択し、お使いの環境に合わせて Amazon Inspector をアクティブ化する方法を確認します。

Standalone account environment

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. [今すぐ始める] を選択します。
3. [Amazon Inspector をアクティブ化] を選択します。

スタンドアロンアカウントで Amazon Inspector をアクティブ化すると、すべてのスキャンタイプがデフォルトでアクティブ化されます。アクティブ化されたスキャンタイプは、Amazon Inspector コンソールのアカウント管理ページから、または Amazon Inspector API を使用して管理できます。Amazon Inspector がアクティブ化されると、対象となるすべてのリソースが自動的に検出され、スキャンが開始されます。以下のスキャンタイプ情報を確認して、どのリソースがデフォルトで対象となるかを確認してください。

Amazon EC2 スキャン

EC2 インスタンスに共通脆弱性識別子 (CVE) データを提供するために、Amazon Inspector では AWS Systems Manager (SSM) エージェントをインストールしてアクティブ化する必要があります。このエージェントは多くの EC2 インスタンスにプリインストールされていますが、手動によるアクティブ化が必要な場合があります。SSM Agent のステータスにかかわらず、すべての EC2 インスタンスがスキャンされ、ネットワークへの露出の問題がないかどうかが確認されます。Amazon EC2 のスキャンを設定する方法については、「Amazon Inspector による Amazon EC2 インスタンスのスキャン」を参照してください。

Amazon ECR スキャン

Amazon ECR スキャンをアクティブ化すると、Amazon Inspector は、Amazon ECR が提供するデフォルトの基本的なスキャンに設定されているプライベートレジストリ内のすべてのコンテナリポジトリを、継続的スキャンによる拡張スキャンに変換します。また、オプションでこの設定をオンプッシュ時のみスキャンしたり、包含ルールを使用して特定のリポジトリをスキャンしたりするように設定することもできます。過去 30 日以内にプッシュされたすべてのイメージは、ライフタイムスキャンをするよう予定されます。この Amazon ECR スキャン設定はいつでも変更できます。Amazon ECR のスキャンを設定する方法については、「Amazon Inspector による Amazon ECR コンテナイメージのスキャン」を参照してください。

AWS Lambda 関数スキャン

AWS Lambda 関数スキャンを有効にすると、Amazon Inspector はアカウント内の Lambda 関数を検出し、脆弱性のスキャンをすぐに開始します。Amazon Inspector は、新しい Lambda

関数とレイヤーをデプロイ時にスキャンし、それらが更新されたとき、または新しい共通脆弱性識別子 (CVE) が発行されたときに再スキャンします。Amazon Inspector には 2 つの異なるレベルの Lambda 関数スキャンが用意されています。デフォルトでは、Amazon Inspector を初めてアクティブ化すると、関数内のパッケージの依存関係をスキャンする Lambda 標準スキャンがアクティブ化されます。さらに、Lambda コードスキャンをアクティブ化し、関数内の開発者コードをスキャンしてコードの脆弱性がないか調べることができます。Lambda 関数のスキャンを設定する方法については、「Amazon Inspector による AWS Lambda 関数のスキャン」を参照してください。

Multi-account environment

Important

これらの手順を実行するには、管理するすべてのアカウントと同じ組織に属し、組織内で Amazon Inspector の管理者を委任するための　AWS Organizations 管理アカウントへのアクセス許可を持っている必要があります。管理者の委任には追加の許可が必要になる場合があります。詳細については、「委任された管理者の指定に必要な許可」を参照してください。

Note

複数のリージョンの複数のアカウントで Amazon Inspector をプログラム的に有効にするには、Amazon Inspector が開発したシェルスクリプトを使用できます。このスクリプトの使用の詳細については、 の Inspector2-enablement-with-cli を参照してください GitHub。

Amazon Inspector の管理者の委任

1. AWS Organizations 管理アカウントにログインします。
2. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
3. 委任管理者ペインで、組織の Amazon Inspector 委任管理者として AWS アカウント 指定する の 12 桁の ID を入力します。その後、[削除] をクリックします。次に、確認ウィンドウで [委任] をもう一度選択します。

> **Note**
>
> 管理者を委任すると、アカウントの Amazon Inspector がアクティブ化されます。

メンバーアカウントを組織に追加する

委任された管理者は、組織の管理アカウントに関連付けられているすべてのメンバーのスキャンをアクティブ化できます。このワークフローでは、すべてのメンバーアカウントのすべてのスキャンタイプをアクティブ化にします。ただし、メンバーは自分のアカウントで Amazon Inspector をアクティベート化したり、委任された管理者がサービスのスキャンを選択的にアクティブ化したりすることもできます。詳細については、「複数のアカウントの管理」を参照してください。

1. 委任された管理者のアカウントにログインします。
2. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
3. ナビゲーションペインで、[アカウント管理] を選択します。[アカウント] テーブルには、組織の管理アカウントに関連付けられているすべてのメンバーアカウントが表示されます。
4. アカウント管理ページから、トップバナーからすべてのアカウントのスキャンを有効にするを選択して、EC2 インスタンス、ECR コンテナイメージ、および組織内のすべてのアカウントの AWS Lambda 関数スキャンをアクティブ化できます。または、[アカウント] テーブルでメンバーとして追加するアカウントを選択することもできます。次に、[アクティブ化] メニューから[すべてのスキャン] を選択します。
5. （オプション）[新しいメンバーアカウントの Inspector を自動的にアクティブ化] の機能をオンにし、含めるスキャンタイプを選択して、組織に追加する新しいメンバーアカウントのスキャンをアクティブ化します。

Amazon Inspector は現在、EC2 インスタンス、ECR コンテナイメージ、および AWS Lambda 関数のスキャンを提供しています。Amazon Inspector をアクティブ化すると、対象となるすべてのリソースの検出とスキャンが自動的に開始されます。以下のスキャンタイプ情報を確認して、どのリソースがデフォルトで対象となるかを確認してください。

Amazon EC2 スキャン

EC2 インスタンスに CVE 脆弱性データを提供するために、Amazon Inspector では AWS Systems Manager (SSM) エージェントをインストールしてアクティブ化する必要があります。このエージェントは多くの EC2 インスタンスにプリインストールされていますが、手動によるアクティブ化が必要な場合があります。SSM Agent のステータスにかかわらず、すべての EC2 インスタンスがスキャンされ、ネットワークへの露出の問題がないかどうかが確認されます。Amazon EC2 のスキャンを設定する方法については、「Amazon Inspector による Amazon EC2 インスタンスのスキャン」を参照してください。

Amazon ECR スキャン

Amazon ECR スキャンをアクティブ化すると、Amazon Inspector は、Amazon ECR が提供するデフォルトの基本的なスキャンに設定されているプライベートレジストリ内のすべてのコンテナリポジトリを、継続的スキャンによる拡張スキャンに変換します。また、オプションでこの設定をオンプッシュ時のみスキャンしたり、包含ルールを使用して特定のリポジトリをスキャンしたりするように設定することもできます。過去 30 日以内にプッシュされたすべてのイメージは、ライフタイムスキャンをするよう予定されます。この Amazon ECR スキャン設定は、委任された管理者がいつでも変更できます。Amazon ECR のスキャンを設定する方法については、「Amazon Inspector による Amazon ECR コンテナイメージのスキャン」を参照してください。

AWS Lambda 関数スキャン

AWS Lambda 関数スキャンを有効にすると、Amazon Inspector はアカウント内の Lambda 関数を検出し、脆弱性のスキャンをすぐに開始します。Amazon Inspector は、新しい Lambda 関数とレイヤーをデプロイ時にスキャンし、それらが更新されたとき、または新しい共通脆弱性識別子 (CVE) が発行されたときに再スキャンします。Lambda 関数のスキャンを設定する方法については、「Amazon Inspector による AWS Lambda 関数のスキャン」を参照してください。

ステップ 2: Amazon Inspector の検出結果を表示する

Amazon Inspector コンソールで、または API を使用して、お使いの環境の検出結果を表示できます。すべての検出結果は Amazon EventBridge と AWS Security Hub （アクティブ化されている場合) にもプッシュされます。さらに、コンテナイメージの検出結果は Amazon ECR にプッシュされます。

Amazon Inspector コンソールには、検出結果の表示形式がいくつか用意されています。Amazon Inspector ダッシュボードには環境に対するリスクの概要が表示され、[検出結果] テーブルでは特定の検出結果の詳細を表示できます。

このステップでは、[検出結果] テーブルと検出結果ダッシュボードを使用して検出結果の詳細を調べます。Amazon Inspector ダッシュボードの詳細については、「ダッシュボードについて」を参照してください。

Amazon Inspector コンソールでご使用の環境の検出結果の詳細を表示するには:

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ナビゲーションペインの [ダッシュボード] を選択します。ダッシュボード内の任意のリンクを選択して、その項目の詳細が記載された Amazon Inspector コンソールのページに移動できます。
3. ナビゲーションペインで [検出結果] を選択します。
4. デフォルトでは、すべての検出結果タブが表示され、環境のすべての EC2 インスタンス、ECR コンテナイメージ、 AWS Lambda 関数の検出結果が表示されます。
5. 検出結果リストのタイトル列で検出結果名を選択すると、その検出結果の詳細ペインが開きます。すべての検出結果には [検出結果の詳細] タブがあります。[検出結果の詳細] タブは次の方法で操作できます。

   - 脆弱性の詳細については、[脆弱性の詳細] セクションのリンクをクリックして、この脆弱性に関するドキュメントを開いてください。
   - リソースをさらに調査するには、[影響を受けるリソース] セクションの「リソース ID」リンクをクリックして、影響を受けるリソースのサービスコンソールを開きます。

   パッケージ脆弱性タイプの結果には、インスペクタースコアと脆弱性インテリジェンスタブもあり、その結果の Amazon Inspector スコアがどのように計算されたかを説明し、検出結果に関連する Common Vulnerability and Exploits (CVE) に関する情報を提供します。検出結果タイプの詳細については、「Amazon Inspector での 検出結果タイプ」を参照してください。

Amazon Inspector ダッシュボードについて

Amazon Inspector ダッシュボードには、現在の AWS リージョンの AWS リソースの集約された統計のスナップショットが表示されます。これらの統計には、リソースカバレッジとアクティブな脆弱性に関する主要なメトリクスが含まれます。ダッシュボードには、Amazon Elastic Compute Cloud (Amazon EC2) インスタンス、Amazon Elastic Container Registry (Amazon ECR)、および最も重要な検出結果を含む AWS Lambda 関数など、アカウントの集計された検出結果データのグループも表示されます。詳細な分析を実行するには、ダッシュボード項目のサポートデータを表示することができます。

お客様のアカウントが組織の Amazon Inspector の委任された管理者アカウントである場合、ダッシュボードには、お客様のアカウントを含む組織内のすべてのアカウントのアカウント範囲、集約された統計、および検出結果データが表示されます。

ダッシュボードを表示する

ダッシュボードには、対象環境の概要と緊急の検出結果が表示されます。

ダッシュボードを表示するには

1. Amazon Inspector コンソール (https://console.aws.amazon.com/inspector/v2/home) を開きます。
2. ナビゲーションペインで、ダッシュボード を選択します。
3. 以下の方法でダッシュボードを操作できます。

    - ダッシュボードは 5 分ごとに自動的に更新されます。ただし、ページの上部右端にある更新アイコンを選択することで、データを手動で更新できます。
    - ダッシュボードで項目のサポートデータを表示するには、項目を選択します。
    - Amazon Inspector の委任された管理者として AWS 組織を通じて複数のアカウントを管理する場合、ダッシュボードにはメンバーアカウントの集計統計が表示されます。ダッシュボードをフィルタし、特定のアカウントのデータのみを表示するには、[アカウント] ボックスにアカウント ID を入力します。

ダッシュボードコンポーネントを理解しデータを解釈する

Amazon Inspector ダッシュボードの各セクションには、現在の AWS における AWS リージョンリソースの脆弱性を把握するのに役立つ、主要なメトリクスまたはアクティブな検出結果データについてのインサイドを提供します。

環境カバレッジ

環境カバレッジセクションには、Amazon Inspector によってスキャンされたリソースに関する統計が表示されます。このセクションでは、Amazon Inspector によってスキャンされた Amazon EC2 インスタンス、Amazon ECR イメージ、および AWS Lambda 関数の数と割合を確認できます。 Amazon Inspector Amazon Inspector の委任された管理者 AWS Organizations として で複数のアカウントを管理する場合、組織アカウントの合計数、Amazon Inspector がアクティブ化された数、およびその結果の組織カバレッジの割合も表示されます。このセクションを使用して、Amazon Inspector の対象とならないリソースを特定することもできます。これらのリソースには、悪用されて組織を危険にさらす可能性のある脆弱性が含まれている可能性があります。詳細については、「Amazon Inspector による AWS 環境のカバレッジを評価する」を参照してください。

カバレッジグループを選択すると、選択したグループのアカウント管理ページに移動します。アカウント管理ページには、Amazon Inspector の対象となっているアカウント、Amazon EC2 インスタンス、Amazon ECR リポジトリに関する詳細が表示されます。

次のカバレッジグループが使用できます。

- アカウント
- インスタンス
- コンテナリポジトリ
- コンテナイメージ
- Lambda

緊急の検出結果

「緊急の検出結果」セクションには、環境内の重大な脆弱性の数と、環境内のすべての検出結果の合計数が表示されます。このセクションには、リソースと評価タイプごとの数が表示されます。緊急の検出結果と Amazon Inspector が緊急度を判断する方法の詳細については、「Amazon Inspector の検出結果について」を参照してください。

[緊急の検出結果グループ] を選択すると、[すべての検出結果] ページに移動し、選択したグループに一致するすべての緊急の検出結果を表示するフィルタが自動的に適用されます。

以下の緊急の検出結果グループを使用できます。

- ECR コンテナーイメージの検出結果
- Amazon EC2 の検出結果
- ネットワーク到達可能性の検出結果
- AWS Lambda 関数の検出結果

リスクベースの修正

リスクベースの修復セクションには、環境内のほとんどのリソースに影響を与える緊急の脆弱性がある上位 5 つのソフトウェアパッケージが表示されます。これらのパッケージを修正することで、環境に重大なリスクの数を大幅に減らすことができます。ソフトウェアパッケージ名を選択すると、関連する脆弱性の詳細と影響を受けるリソースが表示されます。

最も緊急の検出結果があるアカウント

最も緊急の検出結果を持つアカウントセクションには、環境内の最も緊急の検出結果を持つ上位 5 つの AWS アカウントと、そのアカウントの検出結果の合計数が表示されます。このセクションは、Amazon Inspector が によるマルチアカウントスキャン用に設定されている場合にのみ、委任管理者アカウントから表示できます AWS Organizations。このビューは、委任された管理者が組織内でどのアカウントが最も危険にさらされているかを把握するのに役立ちます。

[アカウント ID] を選択すると、影響を受けるメンバーアカウントに関する詳細情報が表示されます。

最も緊急の検出結果がある Amazon ECR リポジトリ

[最も緊急の検出結果がある Elastic Container Registry (ECR) リポジトリ] セクションには、お客様の環境内で最も緊急のコンテナイメージの検出結果がある上位 5 つの Amazon ECR リポジトリが表示されます。ビューには、リポジトリ名、 AWS アカウント識別子、リポジトリ作成日、重大な脆弱性の数、および脆弱性の総数が表示されます。このビューは、どのリポジトリが最も危険にさらされているかを特定するのに役立ちます。

リポジトリ名を選択すると、影響を受けるリポジトリに関する詳細が表示されます。

最も緊急の検出結果があるコンテナイメージ

[最も緊急の検出結果があるコンテナイメージ] セクションには、環境内で最も緊急の検出結果をがある上位 5 つのコンテナイメージが表示されます。ビューには、イメージタグデータ、リポジトリ名、イメージダイジェスト、 AWS アカウント識別子、重大な脆弱性の数、および脆弱性の合計数が表示されます。このビューは、どのコンテナイメージを再構築して再起動する必要があるかをアプリケーション所有者が特定するのに役立ちます。

[コンテナイメージ] を選択すると、影響を受けるコンテナイメージに関する詳細が表示されます。

最も緊急の検出結果があるインスタンス

[最も緊急の検出結果があるインスタンス] セクションには、最も緊急の検出結果がある上位 5 つの Amazon EC2 インスタンスが表示されます。ビューには、インスタンス識別子、 AWS アカウント識別子、Amazon マシンイメージ (AMI) 識別子、緊急な脆弱性の数、および脆弱性の総数が表示されます。このビューは、インフラストラクチャの所有者がどのインスタンスにパッチを適用する必要があるかを特定するのに役立ちます。

[インスタンス ID] を選択すると、影響を受ける Amazon EC2 インスタンスの詳細が表示されます。

最も緊急の検出結果がある Amazon マシンイメージ (AMI)

[最も緊急の検出結果がある Amazon マシンイメージ (AMI)] セクションには、環境内で最も緊急の検出結果を含む上位 5 つの AMI が表示されます。ビューには、AMI 識別子、 AWS アカウント識別子、環境内で実行されている影響を受ける EC2 インスタンスの数、AMI の作成日、AMI のオペレーティングシステムプラットフォーム、緊急な脆弱性の数、および脆弱性の総数が表示されます。このビューは、インフラストラクチャーの所有者が再構築が必要な AMI を特定するのに役立ちます。

[影響を受けるインスタンス] を選択すると、影響を受ける AMI から起動したインスタンスの詳細情報が表示されます。

AWS Lambda 最も重要な検出結果を持つ 関数

最も緊急の検出結果が出たAWS Lambda 関数のセクションには、環境内の最も緊急な検出結果を含む上位 5 つの Lambda 関数が表示されます。ビューには、Lambda 関数名、 AWS アカウント識別子、ランタイム環境、重大な脆弱性の数、高い脆弱性の数、および脆弱性の総数が表示されます。このビューは、インフラストラクチャの所有者が修正が必要な Lambda 機能を特定するのに役立ちます。

関数名を選択すると、影響を受ける AWS Lambda 関数に関する詳細情報が表示されます。

Amazon Inspector の検出結果について

検出結果は、いずれかの AWS リソースに影響する脆弱性に関する詳細なレポートです。検出結果は、検出された脆弱性にちなんで命名され、重要度評価、影響を受けるリソースに関する情報、報告された脆弱性の修正方法の詳細を提供します。

Amazon Inspector は、Amazon EC2 インスタンス、Amazon ECR リポジトリのコンテナイメージ、または AWS Lambda 関数の脆弱性を検出するたびに、結果を生成します。Amazon Inspector は、コンピューティング環境を継続的にスキャンし、ユーザーが修正するまでアクティブな検出結果をすべて保存します。

検出結果を修正すると、検出結果は自動的に閉じられ、Amazon Inspector は 7 日後に検出結果を削除します。リソースを削除すると、Amazon Inspector は 30 日後にリソースに関連付けられた検出結果をすべて削除します。

Amazon Inspector を無効にすると、検出結果は 24 時間後に削除されます。がアカウント AWS を停止すると、検出結果は 90 日後に削除されます。

結果は次のいずれかの状態に分類されます。

[アクティブ]

Amazon Inspector は、修正されていない検出結果をアクティブ として識別します。

抑制

Amazon Inspector は、1 つ以上の抑制ルールの対象となる検出結果を Suppressed として識別します。抑制された検出結果は、抑制された検出結果リストで確認できます。詳細については、「抑制ルールによる Amazon Inspector で検出結果を抑制する」を参照してください。

[Closed] (クローズ)

脆弱性を修正すると、Amazon Inspector はこれを自動的に検出し、検出結果の状態を「クローズド」に変更します。クローズされた検出結果は 7 日後に削除されます。

トピック

- Amazon Inspector での 検出結果タイプ
- Amazon Inspector の検出結果の検索と表示

- Amazon Inspector の検出結果の詳細
- Amazon Inspector スコアと脆弱性インテリジェンス
- Amazon Inspector の検出結果の重要度レベル

Amazon Inspector での 検出結果タイプ

Amazon Inspector は、Amazon Elastic Compute Cloud (Amazon EC2) インスタンス、Amazon Elastic Container Registry (Amazon ECR) リポジトリのコンテナイメージ、および AWS Lambda 関数の結果を生成します。Amazon Inspector では、次のタイプの検出結果を生成できます。

パッケージ脆弱性

パッケージ脆弱性の検出結果は、共通脆弱性識別子 (CVE) にさらされる AWS 環境内のソフトウェアパッケージを識別します。 CVEs 攻撃者は、こうしたパッチが適用されていない脆弱性を利用し、データの機密性、完全性、可用性を侵害したり、他のシステムにアクセスしたりする可能性があります。CVE システムは、セキュリティの脆弱性や露出についての既知の情報を参照する方法です。詳細については、https://www.cve.org/ を参照してください。

Linux の CVE 検出は、ベンダーのセキュリティアドバイザリによってリリースから 24 時間以内に Amazon Inspector に追加されます。Windows 用の CVE 検出は、Microsoft がリリースしてから 48 時間以内に Amazon Inspector に追加されます。Amazon Inspector 脆弱性データベースの検索 を使用して CVE 検出がサポートされているかどうかを確認できます。

Amazon Inspector では、EC2 インスタンス、ECR コンテナイメージ、および Lambda 関数についてパッケージ脆弱性検出結果を生成できます。パッケージ脆弱性検出結果には、この検出結果タイプに固有の詳細が追加されており、インスペクタスコアと脆弱性インテリジェンスがこれに該当します。

コードの脆弱性

コード脆弱性の検出結果は、攻撃者が悪用する可能性のあるコード内の行を特定します。コードの脆弱性には、インジェクションの欠陥、データ漏洩、脆弱な暗号化、コード内の暗号化の欠落などがあります。

Amazon Inspector は、自動推論と機械学習を使用して Lambda 関数のアプリケーションコードを評価し、アプリケーションコードを分析して全体的なセキュリティコンプライアンスを確認します。Amazon との共同開発の内部ディテクターに基づいて、ポリシー違反と脆弱性を特定します

CodeGuru。可能な検出のリストについては、CodeGuru 「ディテクターライブラリ」を参照してください。

> **Important**
>
> Amazon Inspector のコードスキャンでは、コードスニペットをキャプチャして検出された脆弱性をハイライトします。これらのスニペットには、ハードコードされた認証情報やその他の機密情報がプレーンテキストで表示される場合があります。

Amazon Inspector では、Amazon Inspector Lambda コードスキャン をアクティブ化している場合、Lambda 関数のコード脆弱性検出結果を生成できます。

コードの脆弱性に関連して検出されたコードスニペットは、 CodeGuru サービスによって保存されます。デフォルトでは、 によって管理される AWS 所有キー CodeGuru を使用してコードを暗号化しますが、Amazon Inspector API による暗号化には独自のカスタマーマネージドキーを使用できます。詳細については、検出結果のコードの保管時の暗号化を参照してください。

ネットワーク到達可能性

ネットワーク到達可能性の検出結果は、環境内の Amazon EC2 インスタンスへのネットワークパスが開いていることを示しています。インターネットゲートウェイ (Application Load Balancer または Classic Load Balancer の背後にあるインスタンスを含む)、VPC ピアリング接続、または仮想ゲートウェイを介した VPN など、VPC エッジから TCP および UDP ポートに到達可能な場合、これらの結果が表示されます。これらの結果では、セキュリティグループ、アクセス制御リスト、インターネットゲートウェイなどの管理が不適切であったり、潜在的に悪意のあるアクセスを許している可能性があるなど、過度に寛容なネットワーク設定をハイライトします。

Amazon Inspector は、Amazon EC2 インスタンスのネットワーク到達可能性の検出結果のみを生成します。Amazon Inspector は 24 時間ごとにネットワークの到達可能性に関する検出結果のスキャンを実行します。

Amazon Inspector は、ネットワークパスをスキャンする際に以下の設定を評価します。

- Amazon EC2 インスタンス
- AWS Lambda 関数
- アプリケーション ロード バランサー
- Direct Connect

- 弾性ロードバランサ
- 弾性ネットワークインターフェース
- インターネットゲートウェイ
- ネットワークアクセスコントロールリスト
- ルートテーブル
- セキュリティグループ
- サブネット
- 仮想プライベートクラウド
- 仮想プライベートゲートウェイ
- VPC エンドポイント
- ゲートウェイ VPC エンドポイント
- VPC ピアリング接続
- VPN 接続

Amazon Inspector の検出結果の検索と表示

このセクションの手順では、Amazon Inspector コンソールと API を使用して Amazon Inspector で結果を検索して表示する方法について説明します。検出結果の詳細は、検出結果タイプ、脆弱性タイプ、影響を受けるリソースによって異なります。詳細については、「Amazon Inspector の検出結果の詳細」を参照してください。

Console

コンソールで検出結果を表示するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ナビゲーションペインから、検出結果 を選択します。すべての検出結果を表示できる検出結果画面が表示されます。検出結果 テーブルで、タイトル 列で検出結果の名前を選択して検出結果を選択できます。
3. (オプション) カテゴリ別にグループ化された結果を表示することもできます。ナビゲーションペインから検出結果 を選択し、次のいずれかのカテゴリを選択します。

    - 脆弱性別

- インスタンス別

> **Note**
>
> インスタンス別にグループ化された検出結果には、ネットワークの可用性に関する情報は含まれません。

- コンテナイメージ別
- コンテナリポジトリ別
- Lambda 関数別

API

API オペレーション ListFindings を実行します。リクエストでは、特定の検出結果を返すように `filterCriteria` を指定できます。

Amazon Inspector の検出結果の詳細

Amazon Inspector コンソールでは、各検出結果の詳細を見ることができます。検出結果の詳細は検出結果のタイプによって異なります。

検出結果の詳細を表示するには

1. Amazon Inspector コンソール (https://console.aws.amazon.com/inspector/v2/home) にサインインします。
2. 検出結果を表示するリージョンを選択します。
3. ナビゲーションペインで、[検出結果] を選択して検出結果リストを表示する
4. (オプション) フィルターバーを使用して特定の検出結果を選択します。詳細については、「Amazon Inspector の検出結果のフィルタリング」を参照してください。
5. 検出結果を選択して、その詳細パネルを表示します。

[検出結果の詳細] パネルには、検索結果の基本的な識別機能が含まれています。これには、検出結果のタイトル、特定された脆弱性の基本的な説明、修復の提案、および重要度スコアが含まれます。スコアリングについては、「Amazon Inspector の検出結果の重要度レベル」を参照してください。

検出結果の詳細は、検出結果の種類と影響を受けるリソースによって異なります。

すべての検出結果には、検出結果が識別された AWS アカウント ID 番号、重要度、検出結果タイプ、検出結果が作成された日付、およびそのリソースに関する詳細を含むリソースの影響を受けるセクションが含まれます。

検出結果のタイプは、その検出結果に対して利用可能な修復と脆弱性インテリジェンス情報を決定します。検出結果のタイプに応じて、さまざまな検出結果の詳細を使用できます。

パッケージの脆弱性

EC2 インスタンス、ECR コンテナイメージ、Lambda 関数に関しては、パッケージの脆弱性の検出結果を利用できます。詳細については、「パッケージ脆弱性」を参照してください。

パッケージの脆弱性の検出結果には Amazon Inspector スコアと脆弱性インテリジェンス も含まれます。

この検出結果タイプには以下の詳細があります。

- 修正可能 — 影響を受けるパッケージの新しいバージョンで脆弱性が修正されているかどうかを示します。次のいずれかの値があります。
  - YES、これは影響を受けるパッケージのすべてに修正されたバージョンがあることを意味します。
  - NO、これは影響を受けるパッケージに修正されたバージョンがないことを意味します。
  - PARTIAL、これは影響を受けるパッケージの 1 つ以上 (すべてではない) に修正されたバージョンがあることを意味します。
- 考えられる攻撃 — 脆弱性に既知の悪用があることを示します。
  - YES、これは環境内で発見された脆弱性には既知の悪用があるということです。Amazon Inspector では、環境における悪用の使用状況を可視化することはできません。
  - NO、これはこの脆弱性には既知の悪用がないことを意味します。
- 影響を受けるパッケージ — 検出結果で脆弱であると特定された各パッケージと、各パッケージの詳細を一覧表示します。
- ファイルパス — 検出結果に関連付けられた EBS ボリューム ID とパーティション番号です。このフィールドは、エージェントレススキャン を使用してスキャンされた EC2 インスタンスの検出結果に存在します。
- インストール済みバージョン/修正バージョン — 脆弱性が検出された、現在インストールされているパッケージのバージョン番号。インストールされているバージョン番号とスラッシュ (/) の後の値を比較します。2 番目の値は、検出された脆弱性を修正するパッケージのバージョン番号です。このバージョン番号は、共通脆弱性識別子 (CVE) または検出結果に関連するアド

バイザリによって提供されています。脆弱性が複数のバージョンで修正されている場合、このフィールドには修正を含む最新バージョンが表示されます。修正がない場合、この値は `None available` です。

> **Note**
>
> Amazon Inspector がこのフィールドを検出結果に含み始める前に検出結果が検出された場合には、このフィールドの値は空となります。ただし、修正できる場合もあります。

- パッケージマネージャー — このパッケージの設定に使用されるパッケージマネージャー。
- 修復 — 更新されたパッケージまたはプログラミングライブラリから修正が入手できる場合、このセクションには更新を行うために実行できるコマンドが含まれています。提供されたコマンドをコピーして、ご使用の環境で実行できます。

> **Note**
>
> 修復コマンドはベンダーのデータフィードから提供され、システム設定によって異なる場合があります。より具体的なガイダンスについては、検出結果の参考資料またはオペレーティングシステムのマニュアルを参照してください。

- 脆弱性の詳細 — National Vulnerability Database (NVD)、REDHAT、または別の OS ベンダーなど、検出結果で特定された CVE の Amazon Inspector 優先ソースへのリンクを提供します。さらに、検出結果の重要度スコアも表示されます。重要度スコアリングの詳細については、「Amazon Inspector の検出結果の重要度レベル」を参照してください。それぞれのスコアリングベクトルを含め、以下のスコアが含まれています。
  - EPSS スコア
  - Inspector スコア
  - Amazon CVE の CVSS 3.1
  - NVD の CVSS 3.1
  - NVD の CVSS 2.0 (該当する場合、過去の CVE について)
- 関連する脆弱性 — 検出結果に関連する他の脆弱性を指定します。通常、これらは同じパッケージバージョンに影響する他の CVE、または検出結果の CVE と同じグループ内のベンダーが判断した他の CVE です。

コードの脆弱性

コード脆弱性の検出結果は Lambda 関数でのみ確認できます。詳細については、「コードの脆弱性」を参照してください。この検出結果タイプには以下の詳細があります。

- 修正可能 — コード脆弱性の場合、この値は常に YES です。
- ディテクター名 – コードの脆弱性を検出するために使用されるディ CodeGuru テクターの名前。可能な検出のリストについては、CodeGuru 「ディテクターライブラリ」を参照してください。
- ディテクタータグ – CodeGuruディテクターに関連付けられたタグは、タグ CodeGuru を使用して検出を分類します。
- 関連する CWE — コードの脆弱性に関連する共通脆弱性タイプ (CWE) の ID。
- ファイルパス — コード脆弱性のファイルの場所。
- 脆弱性の場所 — Lambda コードスキャンコードの脆弱性の場合、このフィールドには Amazon Inspector が脆弱性を検出した正確なコード行が表示されます。
- 推奨される是正 — 検出結果を修正するためにコードを編集する方法を提示します。

ネットワーク到達可能性

ネットワーク到達可能性の検出結果は EC2 インスタンスでのみ確認できます。詳細については、「ネットワーク到達可能性」を参照してください。この検出結果タイプには以下の詳細があります。

- ポート範囲を開く — EC2 インスタンスにアクセスできるポート範囲。
- ネットワークパスを開く — EC2 インスタンスへのオープンアクセスパスを表示します。パス上の項目を選択すると、詳細が表示されます。
- 修復 — 開いているネットワークパスを閉鎖する方法を推奨します。

Amazon Inspector スコアと脆弱性インテリジェンス

Amazon Inspector コンソールでは、検出結果を選択すると、[Inspector スコアと脆弱性インテリジェンス] タブが表示されます。このタブには、パッケージの脆弱性検出結果のスコアリングの詳細と脆弱性インテリジェンスの詳細が表示されます。これらの詳細は パッケージ脆弱性 検出結果にのみ表示されます。

Amazon Inspector スコア

Amazon Inspector スコアは、Amazon Inspector が各 EC2 インスタンス検出結果ごとに作成される、状況に応じたスコアです。Amazon Inspector スコアは、CVSS v3.1 の基本スコア情報を、ネットワーク到達可能性の結果や悪用可能性データなど、スキャン中にコンピューティング環境から収集された情報と関連付けることによって決定されます。たとえば、脆弱性がネットワーク上で悪用可能であるのに、Amazon Inspector が脆弱なインスタンスへのオープンネットワークパスがインターネットから利用できないと判断した場合、検出結果の Amazon Inspector スコアは基本スコアよりも低くなる可能性があります。

検出結果の基本スコアは、ベンダーが提供する CVSS v3.1 の基本スコアです。RHEL、Debian、または Amazon ベンダーベースのスコアがサポートされています。その他のベンダー、またはベンダーがスコアを提供していない場合、Amazon Inspector は National Vulnerability Database (NVD) の基本スコアを使用します。Amazon Inspector は、共通脆弱性評価システムバージョン 3.1 の計算機を使用してスコアを計算します。個々の検出結果の基本スコアのソースは、[脆弱性の詳細] の下にある検出結果の詳細で、[脆弱性ソース] (または検出結果の JSON では packageVulnerabilityDetails.source) として確認できます。

> **Note**
>
> Amazon Inspector スコアは、Ubuntu を実行している Linux インスタンスでは使用できません。これは、Ubuntu が独自の脆弱性重要度を定義しており、関連する CVE の重要度とは異なる可能性があるためです。

Amazon Inspector スコアの詳細

検出結果の詳細ページを開くと、[Inspector スコアと脆弱性インテリジェンス] タブを選択できます。このパネルには、ベーススコアと [Inspector スコア] の差が表示されます。このセクションでは、Amazon Inspector がソフトウェアパッケージの Amazon Inspector スコアとベンダースコアの組み合わせに基づいて重要度評価を割り当てる方法について説明します。スコアが異なる場合は、このパネルに理由の説明が表示されます。

[CVSS スコアメトリクス] セクションには、CVSS ベーススコアメトリクスと [Inspector スコア] の比較表があります。比較されるメトリクスは、first.org が管理する CVSS 仕様書で定義されている基本メトリクスです。基本メトリクスの概要を以下に示します。

攻撃ベクトル

脆弱性が悪用される可能性がある状況。Amazon Inspector の検出結果では、[ネットワーク]、[隣接ネットワーク]、または [ローカル] を選択できます。

攻撃の複雑さ

これは、攻撃者が脆弱性を悪用したときに直面する難易度を示しています。スコアが低いということは、攻撃者が脆弱性を悪用するために必要な追加条件がほとんどないか全くないことを意味します。高スコアは、攻撃者がこの脆弱性を利用した攻撃を成功させるために多大な労力を費やす必要があることを意味します。

必要な権限

これは、攻撃者が脆弱性を悪用するのに必要な権限レベルを表します。

ユーザーインタラクション

このメトリクスは、この脆弱性を利用した攻撃を成功させるには、攻撃者以外の人間のユーザーが必要かどうかを示します。

スコープ

これは、ある脆弱なコンポーネントの脆弱性が、その脆弱なコンポーネントのセキュリティ範囲外のコンポーネントのリソースに影響を与えるかどうかを示します。この値が [未変更] の場合、影響を受けるリソースと影響を受けるリソースは同じです。この値が「変更済」の場合、脆弱なコンポーネントを悪用して、異なるセキュリティ機関が管理するリソースに影響を与える可能性があります。

機密性

脆弱性が悪用された場合に、リソース内のデータの機密性がどの程度影響を受けるかを測定します。その範囲は、機密性が失われない「なし」から、リソース内のすべての情報が漏洩したり、パスワードや暗号化キーなどの機密情報が漏洩したりする可能性のある「高」までさまざまです。

整合性

これは、脆弱性が悪用された場合に、影響を受けるリソース内のデータの完全性がどの程度影響を受けるかを測定します。攻撃者が影響を受けるリソース内のファイルを変更すると、完全性が危険にさらされます。スコアの範囲は、攻撃者がどの情報も変更することを許可しない「なし」から、脆弱性が悪用された場合、攻撃者がいずれかまたはすべてのファイルを変更することができる「高」まであります。

現在利用できるリージョン

これは、脆弱性が悪用された場合に、影響を受けるリソースの可用性に与える影響の度合いを測定します。スコアの範囲は、脆弱性が可用性にまったく影響を与えない場合の「なし」から、悪用された場合、攻撃者のリソースの利用を完全に拒否したり、サービスを利用できなくしたりすることができる「高」まであります。

脆弱性インテリジェンス

このセクションでは、Amazon が提供する CVE に関する情報のほか、Recorded Future や Cybersecurity and Infrastructure Security Agency (CISA) などの業界標準のセキュリティインテリジェンスソースについてもまとめています。

> **Note**
>
> CISA、Amazon、または Recorded Future のインテリジェンスは、すべての CVE で利用できるわけではありません。

脆弱性インテリジェンスの詳細は、コンソールまたは BatchGetFindingDetails API を使用して表示できます。以下の詳細が、コンソールで使用可能です。

ATT&CK

このセクションでは、CVE に関連する MITRE の戦術、技法、手順 (TTP) について説明します。関連する TTP が表示されます。該当する TTP が 3 つ以上ある場合は、リンクを選択すると詳細なリストが表示されます。戦術や技法を選択すると、MITRE のウェブサイトにその情報が表示されます。

CISA

このセクションでは、脆弱性に関連する日付について説明します。Cybersecurity and Infrastructure Security Agency (CISA) が、活発な悪用の証拠に基づいてその脆弱性を「Known Exploited Vulnerabilities Catalog」(悪用された既知の脆弱性カタログ) に追加した日付と、CISA がシステムにパッチを適用する期限。この情報は CISA から提供されています。

既知のマルウェア

このセクションでは、この脆弱性を悪用する既知のエクスプロイトキットとツールを一覧表示します。

証拠

このセクションでは、この脆弱性に関係する最も緊急のセキュリティイベントを要約します。緊急度が同じイベントが 3 つ以上ある場合は、最新のイベントの上位 3 つが表示されます。

最終レポート時刻

このセクションには、この脆弱性が一般に悪用されたことが判明した最終日が表示されます。

Amazon Inspector の検出結果の重要度レベル

Amazon Inspector が脆弱性の検出結果を生成すると、その検出結果に重要度が自動的に割り当てられます。検出結果の重要度は、検出結果の主要な特性を反映し、検出結果の評価と優先順位付けに役立ちます。検出結果の重要度は、影響を受けたリソースが組織に対して緊急性または重要性を意味する、または示すものではありません。

検出結果の重要度評価は、参考、低、中、高、緊急のいずれのレベルに該当する数値スコアによって決まります。

Amazon Inspector が重要度を判断する方法は、検出結果タイプによって異なります。Amazon Inspector が各検出結果タイプの重要度評価を決定する方法の詳細については、以下のセクションを参照してください。

ソフトウェアパッケージの脆弱性の重要度

Amazon Inspector は、ソフトウェアパッケージの脆弱性の重要度スコアの基礎として NVD/CVSS スコアを使用します。NVD/CVSS スコアは、NVD によって公開され、CVSS によって定義される脆弱性重要度スコアです。NVD/CVSS スコアは、攻撃の複雑さ、悪用コードの成熟度、必要な権限などのセキュリティメトリクスで構成されています。Amazon Inspector は、脆弱性の重要度を反映した 1 から 10 までの数値スコアを生成します。Amazon Inspector では、これを基本スコアとして分類しています。これは、脆弱性の重要度がその固有の特性に従って反映され、時間が経過しても変化しないためです。このスコアは、デプロイされたさまざまな環境における最悪の場合の妥当な影響も想定しています。CVSS v3 標準では、CVSS スコアを以下の重要度評価にマッピングしています。

スコア	Rating
0	情報
0.1 ~ 3.9	低

4.0 ~ 6.9	中程度
7.0 ~ 8.9	高い
9.0 ~ 10.0	[非常事態]

パッケージ脆弱性の検出結果は、重要度が「未選別」である場合もあります。つまり、ベンダーは検出された脆弱性の脆弱性スコアをまだ設定していないということです。この場合、検出結果の参照 URL を使用して脆弱性を調査し、それに応じて対応することをおすすめします。

パッケージ脆弱性の検出結果には、検出結果の詳細の一部として、以下のスコアと関連するスコアリングベクトルが含まれます。

- EPSS スコア
- Inspector スコア
- Amazon CVE の CVSS 3.1
- NVD の CVSS 3.1
- NVD の CVSS 2.0 (該当する場合)

コード脆弱性の重要度

コード脆弱性の検出結果の場合、Amazon Inspector は検出結果を生成した Amazon CodeGuru ディテクターによって定義された重要度レベルを使用します。各ディテクターには CVSS v3 スコアリングシステムを使用して重要度が割り当てられます。重要度 CodeGuru が使用する説明については、「 CodeGuru ガイド」の「重要度の定義」を参照してください。重要度別のディテクターのリストについては、以下のサポートされているプログラミング言語から選択してください。

- 重要度別 Python ディテクター
- 重要度別 Java ディテクター

ネットワーク到達可能性の重要度

Amazon Inspector は、公開されているサービス、ポート、プロトコル、およびオープンパスのタイプに基づいて、ネットワーク到達可能性の脆弱性の重要度を判断します。次の表では、これらの重要度評価を定義しています。オープンパス評価列の値は、仮想ゲートウェイ、ピア接続された VPCs、

および AWS Direct Connect ネットワークからのオープンパスを表します。公開されている他のすべてのサービス、ポート、プロトコルには「情報重要度」という評価があります。

サービス	TCP ポート	UDP ポート	インターネットパス評価	オープンパス評価
DHCP	67、68、546、547	67、68、546、547	中程度	情報
Elasticsearch	9300、9200	該当なし	中程度	情報
FTP	21	21	高	中程度
グローバルカタログ LDAP	3268	該当なし	中程度	情報
TLS経由のグローバルカタログLDAP	3269	該当なし	中程度	情報
HTTP	80	80	低	情報
HTTPS	443	443	低	情報
Kerberos	88、464、543、544、749、751	88、464、749、750、751、752	中程度	情報
LDAP	389	389	中程度	情報
TLS 経由の LDAP	636	該当なし	中程度	情報
MongoDB	27017、27018、27019、28017	該当なし	中程度	情報
MySQL	3306	該当なし	中程度	情報
NetBIOS	137、139	137、138	中程度	情報

NFS	111、2049、4045、1110	111、2049、4045、1110	中程度	情報
Oracle	1521、1630	該当なし	中程度	情報
PostgreSQL	5432	該当なし	中程度	情報
印刷サービス	515	該当なし	高	中程度
RDP	3389	3389	中程度	低
RPC	111、135、530	111、135、530	中程度	情報
SMB	445	445	中程度	情報
SSH	22	22	中程度	低
SQL Server	1433	1434	中程度	情報
Syslog	601	514	中程度	情報
Telnet	23	23	高	中程度
WINS	1512、42	1512、42	中程度	情報

Amazon Inspector での検出結果の管理

Amazon Inspector には、検出結果をソート、グループ化、管理するためのさまざまな方法が用意されています。これらの機能は、検出結果を環境に合わせて調整し、さまざまなビュー別に検出結果を集約し、特定の AWS 環境の脆弱性に焦点を当てるのに役立ちます。

検出結果は、その状態 (アクティブ、抑制、終了) に応じてさまざまなビューに表示されます。デフォルトでは、各ビューにはアクティブな結果のみが表示されます。アクティブな検出結果とは、Amazon Inspector によって検出された、脆弱性または潜在的な脅威を示す、潜在的なセキュリティ問題を表します。抑制された検出結果とは、抑制ルールを使用して除外した、アクティブな検出結果のことです。Amazon Inspector は、検出結果が修復されたことを検出すると、自動的に結果のステータスを終了ステータスに設定します。検出結果を手動で終了させることはありません。

AWS 環境全体のセキュリティ状態を包括的に把握できる AWS Security Hubサービスである で結果を表示することもできます。詳細については、「Amazon Inspector と の統合 AWS Security Hub」を参照してください。コンテナイメージの検出結果は Amazon ECR コンソールでも利用でき、AWS Command Line Interface （ AWS CLI) または API を使用してすべてのリソースの検出結果を表示できます。

トピック

- Amazon Inspector 結果の確認
- Amazon Inspector の検出結果のフィルタリング
- 抑制ルールによる Amazon Inspector で検出結果を抑制する
- Amazon Inspector からの調査結果レポートのエクスポート
- Amazon を使用した Amazon Inspector の検出結果へのカスタムレスポンスの作成 EventBridge

Amazon Inspector 結果の確認

Amazon Inspector コンソールでは、関連するグループ分けに基づいて検出結果がタブ付きビューに表示されます。各ビューには、特定の脆弱性の分析、最も脆弱なリソースの特定、および環境における脆弱性の全体的な影響の測定に役立つ情報が含まれています。検出結果 ナビゲーションサイドパネルのオプションを選択すると、別の検出結果ビューに移動できます。また、各ビューにフィルターを作成して、特定のタイプの検出結果に焦点を当てることもできます。フィルタを使用する方法については、「Amazon Inspector の検出結果のフィルタリング」を参照してください。

検出結果は次のパラメータでグループ化できます。

- 脆弱性別 — 環境内で検出された最も緊急な脆弱性を一覧表示します。このビューから脆弱性のタイトルを選択すると、詳細ペインが開き、追加情報が表示されます。
- アカウント別 — アカウント、各アカウントの Amazon Inspector のスキャンカバレッジ率、各アカウントの重要度が「緊急」と「高」の合計件数が表示されます。このグループ化は委任された管理者のみが利用できます。
- インスタンス別 — 環境内で最も脆弱な Amazon EC2 インスタンスを一覧表示します。
- コンテナイメージ別 — 環境内で最も脆弱な Amazon ECR コンテナイメージを一覧表示します。
- コンテナリポジトリ別 — 最も脆弱なリポジトリを表示します。
- Lambda 関数別 — 最も脆弱な Lambda 関数を表示します。
- すべての結果 — ご使用の環境に関するすべての検出結果のリストを表示します。これは [検出結果] ページに移動したときのデフォルトビューです。このビューでは、アクティブな結果、抑制された結果、終了した結果でフィルタリングできます。

フィルターに基づいて抑制ルールを作成し、ビューから検出結果を除外できます。詳細については、「抑制ルールによる Amazon Inspector で検出結果を抑制する」を参照してください。

Amazon Inspector の検出結果のフィルタリング

検出結果フィルターを使用すると、指定した条件に一致する結果のみを表示できます。フィルタ条件に一致しない検出結果はビューから除外されます。Amazon Inspector コンソールを使用して検出結果フィルターを作成できます。これらのフィルターを使用して既存および今後の検出結果を自動的に抑制するには、「抑制ルールによる Amazon Inspector で検出結果を抑制する」を参照してください。

Amazon Inspector コンソールでフィルターを作成

各検出結果ビューでは、フィルター機能を使用して特定の特性を持つ検出結果を検索できます。別のタブ付きビューに移動すると、フィルターは削除されます。

フィルタは、フィルタ属性とフィルタ値からなるフィルタ基準で構成されます。フィルター条件に一致しない検出結果は検出結果リストから除外されます。例えば、管理者アカウントに関連付けられているすべての結果を表示するには、 AWS アカウント ID 属性を選択し、12 桁の AWS アカウント ID の値とペアリングします。

すべての検出結果に適用されるフィルター条件もあれば、特定のリソースタイプや検出結果タイプのみに適用されるフィルター条件もあります。

検出結果ビューにフィルターを適用するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ナビゲーションペインで 調査結果 を選択します。デフォルトビューでは、「アクティブ」ステータスのすべての検出結果が表示されます。
3. 検索結果を条件でフィルタリングするには、[フィルタを追加] バーを選択すると、そのビューに適用可能なすべてのフィルタ条件のリストが表示されます。さまざまなビューで、さまざまなフィルター条件を使用できます。
4. フィルターに使う条件をリストから選択します。
5. 条件入力ペインから、その条件を定義するために必要なフィルター値を入力します。
6. [適用] を選択して、そのフィルター条件を現在の結果に適用します。フィルター入力バーをもう一度選択すると、他のフィルター条件を引き続き追加できます。
7. (オプション) 抑制された検出結果または終了した結果を表示するには、フィルターバーの [アクティブ] を選択し、次に [抑制] または [終了] を選択します。[すべて表示] を選択すると、アクティブな結果、抑制された結果、終了した結果が同じビューに表示されます。

抑制ルールによる Amazon Inspector で検出結果を抑制する

抑制ルールを使用して、条件に一致する検出結果を除外します。例えば、脆弱性スコアが低いすべての検出結果を抑制するルールを作成して、最も重要度の高い検出結果のみに集中できます。

> **Note**
>
> 抑制ルールは、検出結果のリストをフィルタリングするためにのみ使用され、検出結果に影響を与えたり、Amazon Inspector が検出結果を生成したりすることはありません。

Amazon Inspector が抑制ルールに一致する検出結果を生成する場合、検出結果は抑制された に設定されます。抑制ルールに一致する検出結果は、デフォルトではリストに表示されません。

Amazon Inspector は、抑制された検出結果が修復されるまで保存します。Amazon Inspector は修復された検出結果を検出します。Amazon Inspector は、修正された検出結果を検出すると、検出結果を「クローズ」に設定し、7 日間保存します。

抑制された検出結果は、イベント EventBridge として AWS Security Hub および Amazon に発行されます。 EventBridge ルールを使用して検出結果のステータスを変更することで、Security Hub で不

要な検出結果を自動的に抑制できます。詳細については、「 で自動抑制ルールを作成する方法 AWS Security Hub」を参照してください。

検出結果を閉じて修正する抑制ルールを作成することはできません。サプレッションルールを作成して、リストに表示される結果をフィルタリングすることしかできません。抑制された検出結果は、Amazon Inspector コンソールでいつでも表示できます。

> **Note**
>
> 組織のメンバーアカウントは、抑制ルールを作成または管理できません。

抑制ルールを作成する

抑制ルールを作成して、デフォルトで表示される検出結果のリストを絞り込むことができます。CreateFilter API を使用して を の値SUPRESSとして指定することで、抑制ルールをプログラムで作成できますaction。

> **Note**
>
> 抑制ルールを作成および管理できるのは、スタンドアロンアカウントと Amazon Inspector の委任管理者のみです。組織のメンバーには、ナビゲーションペインに抑制ルールのオプションが表示されません。

抑制ルールを作成するには (コンソール)

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ナビゲーションペインで、[抑制ルール] を選択します。次に、[Create rule (ルールを作成)] を選択します。
3. 各条件について、以下を実行します。
   - フィルターバーを選択すると、抑制ルールに追加できるフィルター条件のリストが表示されます。
   - 抑制ルールのフィルター条件を選択します。
4. 条件を追加し終えたら、ルールの名前と、必要に応じて説明を入力します。

5. [Save rule] (ルールを保存) を選択します。Amazon Inspector は、新しい抑制ルールを直ちに適用し、条件に一致する結果はすべて非表示にします。

抑制された検出結果を表示する

デフォルトでは、Amazon Inspector は抑制された検出結果を Amazon Inspector コンソールに表示しません。ただし、特定のルールによって抑制された結果は表示できます。

抑制された検出結果を表示するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ナビゲーションペインで、[抑制ルール] を選択します。
3. 抑制ルールリストで、ルールのタイトルを選択します。

抑制ルールを変更する

抑制ルールはいつでも変更ができます。

抑制ルールを変更するには

1. Amazon Inspector コンソール (https://console.aws.amazon.com/inspector/v2/home) にサインインします。
2. ナビゲーションペインで、[抑制ルール] を選択します。
3. 変更する抑制ルールのタイトルを選択します。
4. 必要な変更を加え、[保存] を選択してルールを更新します。

抑制ルールを削除する

抑制ルールは削除できます。抑制ルールを削除すると、Amazon Inspector は、ルールの基準を満たし、他のルールによって抑制されていない、新規および既存の結果の抑制を停止します。

抑制ルールを削除したすると、ルールの基準を満たした検出結果の新規および現在の発生は、ステータスが [アクティブ] になります。これは、それらが Amazon Inspector コンソールにデフォルトで表示されることを意味します。さらに、Amazon Inspector はこれらの結果を AWS Security Hub と Amazon にイベント EventBridge として発行します。

抑制ルールを削除するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ナビゲーションペインで、[抑制ルール] を選択します。
3. 削除する抑制ルールのタイトルの横にあるチェックボックスをオンにします。
4. [削除] を選択し、その選択を確定してルールを完全に削除します。

Amazon Inspector からの調査結果レポートのエクスポート

Amazon EventBridge および に結果を送信するだけでなく AWS Security Hub、オプションで結果レポートとして Amazon Simple Storage Service (Amazon S3) バケットに結果をエクスポートすることもできます。調査結果レポートは、レポートに含めることを選択した検出結果の詳細を含む CSV または JSON ファイルです。特定の時点における検出結果の詳細なスナップショットを提供します。ファイルには、検出結果ごとに、影響を受けるリソースの Amazon リソースネーム (ARN)、検出結果が作成された日時、関連する共通脆弱性識別子 (CVE) ID、検出結果の重要度、ステータス、および Amazon Inspector と CVSS スコアなどの詳細が含まれています。

調査結果レポートを設定するときは、まずどの検出結果をレポートに含めるかを指定します。デフォルトでは、Amazon Inspector には、ステータスが「アクティブ」になっている現在の AWS リージョン におけるすべての検出結果データが含まれます。お客様が組織の委任された Amazon Inspector 管理者である場合、これには、組織内のすべてのメンバーアカウントの検出結果データが含まれます。

オプションで、データをフィルタリングしてレポートをカスタマイズできます。フィルターを使用して、特定の特性を持つ検出結果のデータを含めたり除外したりできます。たとえば、特定の時間範囲に作成されたすべての「緊急」の検出結果、特定のリソースに関するすべての「アクティブ」な検出結果、または特定のタイプのすべての「緊急」の検出結果などです。ユーザーが組織の Amazon Inspector 管理者である場合は、フィルターを使用して、組織 AWS アカウント 内の特定の の検出結果を含むレポートを作成できます。たとえば、ステータスがアクティブで、修正が利用可能なアカウントのすべての緊急の検出結果などです。その後、修復するためにそのレポートをアカウント所有者と共有することができます。

> **Note**
>
> CreateFindingsReport API を使用して検出結果レポートをエクスポートすると、デフォルトではアクティブな検出結果のみが表示されます。[抑制] された検出結果または[終了] した検

出結果を表示するには、findingStatus フィルタ条件の値として SUPPRESSED または CLOSED を指定する必要があります。

検出結果レポートをエクスポートすると、Amazon Inspector は指定した AWS Key Management Service（ AWS KMS) キーでデータを暗号化し、指定した S3 バケットにレポートを追加します。暗号化キーは、現在の にあるカスタマー管理の AWS Key Management Service（ AWS KMS) 対称暗号化キーである必要がありますAWS リージョン。さらに、キーポリシーで Amazon Inspector がキーの使用を許可する必要があります。S3 バケットも現在のリージョンにある必要があり、バケットのポリシーで Amazon Inspector がバケットにオブジェクトを追加することを許可する必要があります。

Amazon Inspector がレポートの暗号化と保存を完了したら、指定した S3 バケットからレポートをダウンロードするか、別の場所に移動できます。あるいは、レポートを同じ S3 バケットに保存し、そのバケットを後でエクスポートする調査結果レポートのリポジトリとして使用することもできます。

このトピックでは、 を使用して検出結果レポートを AWS Management Console エクスポートするプロセスについて説明します。このプロセスでは、必要なアクセス許可を確認し、必要なリソースを設定し、レポートを設定してエクスポートします。

> **Note**
>
> 調査結果レポートは一度に 1 つしかエクスポートできません。エクスポートが現在進行中の場合は、エクスポートが完了するまで待ってから、別のレポートをエクスポートします。

タスク

- ステップ 1: アクセス許可を確認する
- ステップ 2: S3 バケットを設定する
- ステップ 3: AWS KMS keyを設定する
- ステップ 4: 調査結果レポートを設定しエクスポートする
- エクスポートエラーのトラブルシューティング

調査結果レポートを初めてエクスポートした後、ステップ 1～3 は省略できます。これは主に、同じ S3 バケットを使用するかどうか、および後続のレポート AWS KMS key に使用するかどうかによって異なります。

ステップ 1～3 の後にプログラムでレポートをエクスポートする場合は、Amazon Inspector API の CreateFindingsReportオペレーションを使用します。

ステップ 1: アクセス許可を確認する

Amazon Inspector から調査結果レポートをエクスポートする前に、調査結果レポートのエクスポートと、レポートの暗号化と保存のためのリソースの設定の両方に必要なアクセス許可があることを確認してください。アクセス許可を確認するには、 AWS Identity and Access Management (IAM) を使用して、IAM ID にアタッチされている IAM ポリシーを確認します。次にこれらのポリシー内の情報を、調査結果レポートをエクスポートするために実行を許可されなければならない以下のアクションのリストと比較します。

Amazon Inspector

Amazon Inspector の場合、次のアクションの実行が許可されていることを確認します。

- `inspector2:ListFindings`
- `inspector2:CreateFindingsReport`

これらのアクションにより、アカウントの検出結果データを取得し、そのデータを調査結果レポートにエクスポートできます。

サイズの大きいレポートをプログラムでエクスポートする予定の場合は、レポートのステータスを確認する、`inspector2:CancelFindingsReport`、進行中のエクスポートをキャンセルする操作が許可されていることも確認してください。`inspector2:GetFindingsReportStatus`

AWS KMS

で AWS KMS、次のアクションを実行できることを確認します。

- `kms:GetKeyPolicy`
- `kms:PutKeyPolicy`

これらのアクションにより、Amazon Inspector にレポートの暗号化に使用させたい AWS KMS key のキーポリシーを取得して更新できます。

Amazon Inspector コンソールを使用してレポートをエクスポートするには、次のアクションの実行も許可されていることを確認します AWS KMS 。

- `kms:DescribeKey`
- `kms:ListAliases`

これらのアクションにより、アカウントの AWS KMS keys に関する情報を取得して表示することが許可されます。その後、これらのキーのいずれかを選択してレポートを暗号化できます。

レポートを暗号化するために新しい KMS キーを作成することを計画している場合、`kms:CreateKey` アクションの実行も許可される必要があります。

Amazon S3

Amazon S3 の場合、次のアクションの実行が許可されていることを確認します。

- `s3:CreateBucket`
- `s3:DeleteObject`
- `s3:PutBucketAcl`
- `s3:PutBucketPolicy`
- `s3:PutBucketPublicAccessBlock`
- `s3:PutObject`
- `s3:PutObjectAcl`

これらのアクションにより、Amazon Inspector でレポートを保存する S3 バケットを作成して設定できます。また、バケットにオブジェクトを追加したり、バケットからオブジェクトを削除したりすることもできます。

Amazon Inspector コンソールを使用してレポートをエクスポートする予定がある場合は、`s3:ListAllMyBuckets` および `s3:GetBucketLocation` アクションの実行が許可されていることも確認してください。これらのアクションにより、アカウントの S3 バケットに関する情報を取得して表示することができます。その後、レポートを保存するバケットを 1 つ選択できます。

必要なアクションの 1 つ以上を実行できない場合は、次のステップに進む前に、 AWS 管理者にサポートを依頼してください。

ステップ 2: S3 バケットを設定する

アクセス許可を確認したら、調査結果レポートを保存する S3 バケットを設定します。自分のアカウントの既存のバケットでも、別の が所有するアクセスが許可されている既存のバケットでも AWS アカウント かまいません。レポートを新しいバケットに保存する場合は、先に進む前にバケットを作成してください。

S3 バケットは、エクスポートする検出結果データ AWS リージョン と同じ にある必要があります。例えば、Amazon Inspector を米国東部 (バージニア北部) リージョンで使用していて、そのリージョンの検出結果データをエクスポートする場合、バケットも米国東部 (バージニア北部) リージョンにある必要があります。

さらに、バケットのポリシーでは、Amazon Inspector がバケットにオブジェクトを追加することを許可する必要があります。このトピックでは、バケットポリシーを更新する方法について説明し、ポリシーに追加するステートメントの例も示します。バケットポリシーの追加と更新の詳細については、「Amazon Simple Storage Service ユーザーガイド」の「バケットポリシーの使用」を参照してください。

別のアカウントが所有する S3 バケットにレポートを保存する場合は、バケットの所有者と連携してバケットのポリシーを更新します。また、バケットの URI も取得します。レポートをエクスポートするときに、この URI を入力する必要があります。

バケットポリシーを更新するには

1. https://console.aws.amazon.com/s3 で Amazon S3 コンソールを開きます。
2. ナビゲーションペインで、バケットを選択します。
3. 調査結果レポートを保存する S3 バケットを選択します。
4. アクセス許可 タブを選択します。
5. バケットポリシー セクションで、編集 を選択します。
6. 次のステータス例をクリップボードにコピーします。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "allow-inspector",
   "Effect": "Allow",
   "Principal": {
    "Service": "inspector2.amazonaws.com"
   },
   "Action": [
    "s3:PutObject",
    "s3:PutObjectAcl",
    "s3:AbortMultipartUpload"
   ],
   "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
   "Condition": {
```

```
    "StringEquals": {
     "aws:SourceAccount": "111122223333"
    },
    "ArnLike": {
     "aws:SourceArn": "arn:aws:inspector2:Region:111122223333:report/*"
    }
   }
  }
 ]
}
```

7. Amazon S3 コンソールのバケットポリシーエディタで、前のステートメントをポリシーに貼り付けてポリシーに追加します。

   ステートメントをポリシーに追加するときに、構文が有効であることを確認します。バケットポリシーは JSON 形式を使用します。これは、ステートメントをポリシーに追加する場所に応じて、ステートメントの前後にカンマを追加する必要があることを意味します。ステートメントを最後のステートメントとして追加する場合は、前のステートメントの右中括弧の後にカンマを追加します。最初のステートメントとして追加するか、既存の 2 つのステートメントの間に追加する場合は、右中括弧の後にカンマを追加します。

8. 環境に合った正しい値でステートメントを更新します。

   - *DOC-EXAMPLE-BUCKET* は、バケットの名前です。
   - *111122223333* は、お客様の AWS アカウントのアカウント ID です。
   - *#####*は AWS リージョン 、Amazon Inspector を使用していて、Amazon Inspector がバケットにレポートを追加できるようにする です。たとえば、米国東部 (バージニア北部) リージョンの場合は `us-east-1` です。

   > **Note**
   >
   > 手動で有効にした で Amazon Inspector を使用している場合は AWS リージョン、`Service`フィールドの値に適切なリージョンコードも追加します。このフィールドは Amazon Inspector サービスプリンシパルを指定します。
   > たとえば、リージョンコード `me-south-1` が設定されている中東 (バーレーン) リージョンで Amazon Inspector を使用している場合は、ステートメントで `inspector2.amazonaws.com` を `inspector2.me-south-1.amazonaws.com` に置き換えます。

これらのステートメント例は、2 つの IAM グローバル条件キーを使用する条件を定義していることにご注意してください。

- aws:SourceAccount – この条件により、Amazon Inspector はアカウントのバケットにのみレポートを追加できます。これにより、Amazon Inspector が他のアカウントのバケットにレポートを追加できなくなります。具体的には、条件は、aws:SourceArn 条件で指定されたリソースおよびアクションに対して、バケットを使用できるアカウントを指定します。

  バケット内の追加アカウントのレポートを保存するには、追加のアカウントごとにアカウント ID をこの条件に追加します。例:

  ```
  "aws:SourceAccount": [111122223333,444455556666,123456789012]
  ```

- aws:SourceArn – この条件は、バケットに追加されるオブジェクトのソースに基づいてバケットへのアクセスを制限します。これにより、他の AWS のサービス がバケットにオブジェクトを追加できなくなります。また、Amazon Inspector がお客様のアカウントで他のアクションを実行中にオブジェクトをバケットに追加するのを防ぐこともできます 具体的には、Amazon Inspectorは、オブジェクトが調査結果レポートであり、かつ、それらのレポートがアカウントによって作成され、かつ、条件で指定されたリージョンにある場合にのみ、オブジェクトをバケットに追加することができます。

  Amazon Inspector が追加のアカウントで指定したアクションを実行することを許可するには、追加のアカウントごとに Amazon リソースネーム (ARN) をこの条件に追加します。例:

  ```
  "aws:SourceArn": [
      "arn:aws:inspector2:Region:111122223333:report/*",
      "arn:aws:inspector2:Region:444455556666:report/*",
      "arn:aws:inspector2:Region:123456789012:report/*"
  ]
  ```

  aws:SourceAccount と aws:SourceArn の条件によって指定されたアカウントは、一致する必要があります。

どちらの条件も、Amazon Inspector が Amazon S3 とのトランザクション中に [混乱した代理] として使用されるのを防ぐのに役立ちます。お勧めしませんが、バケットポリシーからこれらの条件を削除できます。

9. バケットポリシーの更新が完了したら、変更を保存する を選択します。

ステップ 3: AWS KMS keyを設定する

アクセス許可を確認して S3 バケットを設定したら、Amazon Inspector で調査結果レポートを暗号化するためにどの AWS KMS key を使用するかを決定します。キーは、カスタマーマネージドキーで、対称暗号化 KMS キーである必要があります。さらに、キーは、レポートを保存するように設定した S3 バケット AWS リージョン と同じ にある必要があります。

キーは、ご自分のアカウントの既存の KMS キーでも、別のアカウントが所有する既存の KMS キーでもかまいません。新しい KMS キーを使用する場合は、先に進む前にキーを作成します。別のアカウントが所有する既存のキーを使用したい場合は、そのキーの Amazon リソースネーム (ARN) を取得します。Amazon Inspector からレポートをエクスポートするときに、この ARN を入力する必要があります。KMS キー設定の作成と確認については、「AWS Key Management Service デベロッパーガイド」の「キーの管理」を参照してください。

使用する KMS キーを決定した後、Amazon Inspector にそのキーを使用するアクセス許可を付与します。そうしないと、Amazon Inspector がレポートを暗号化しエクスポートすることができません。Amazon Inspector にキーを使用するアクセス許可を付与するには、キーのキーポリシーを更新します。キーポリシーと KMS キーへのアクセス管理の詳細については、「AWS Key Management Service デベロッパーガイド」の「AWS KMSのキーポリシー」を参照してください。

キーポリシーを更新するには

> **Note**
>
> 以下の手順は、Amazon Inspector が既存のキーを使用できるように更新するためのものです。既存のキーがまだない場合は、それを作成するためのガイダンスとして https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html を参照してください。

1. https://console.aws.amazon.com/kms で AWS KMS コンソールを開きます。
2. を変更するには AWS リージョン、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. レポートの暗号化に使用する KMS キーを選択します。キーは対称暗号化 (SYMMETRIC_DEFAULT) キーである必要があります。
5. アクセスポリシー タブで 編集 を選択します。[編集] ボタンのあるキーポリシーが表示されない場合は、まず [ポリシービューへの切り替え] を選択する必要があります。
6. 次のステートメント例をクリップボードにコピーします。

```
{
    "Sid": "Allow Amazon Inspector to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": "inspector2.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:inspector2:Region:111122223333:report/*"
        }
    }
}
```

7. AWS KMS コンソールのキーポリシーエディタで、前述のステートメントをキーポリシーに貼り付けて、ポリシーに追加します。

   ステートメントをポリシーに追加するときに、構文が有効であることを確認します。キーポリシーは JSON 形式を使用します。これは、ステートメントをポリシーに追加する場所に応じて、ステートメントの前後にカンマを追加する必要があることを意味します。ステートメントを最後のステートメントとして追加する場合は、前のステートメントの右中括弧の後にカンマを追加します。最初のステートメントとして追加するか、既存の 2 つのステートメントの間に追加する場合は、右中括弧の後にカンマを追加します。

8. 環境に合った正しい値でステートメントを更新します。

   - *111122223333* は、お客様の AWS アカウントのアカウント ID です。
   - *#####*は、Amazon Inspector AWS リージョン が キーを使用してレポートを暗号化できるようにする です。たとえば、米国東部 (バージニア北部) リージョンの場合は `us-east-1` です。

> **Note**
>
> 手動で有効にした で Amazon Inspector を使用している場合は AWS リージョン、`Service`フィールドの値に適切なリージョンコードも追加します。たとえば、リージョンコードが設定されている中東 (バーレーン) リージョンで Amazon Inspector を使用している場合は、`inspector2.amazonaws.com` を `inspector2.me-south-1.amazonaws.com` に置き換えます。

前のステップのバケットポリシーのサンプルステートメントと同様に、この例の `Condition` フィールドでは 2 つの IAM グローバル条件キーを使用しています。

- aws:SourceAccount – この条件により、Amazon Inspector はアカウントに対してのみ指定されたアクションを実行できます。具体的には、`aws:SourceArn` 条件で指定されたリソースおよびアクションに対して、指定されたアクションを実行できるアカウントを決定します。

  Amazon Inspector が追加のアカウントに対して指定されたアクションを実行することを許可するには、追加の各アカウントのアカウント ID をこの条件に追加します。例:

  ```
  "aws:SourceAccount": [111122223333,444455556666,123456789012]
  ```

- aws:SourceArn – この条件は、他の AWS のサービス が指定されたアクションを実行できないようにします。また、Amazon Inspector がお客様のアカウントで他のアクションを実行中にキーを使用するのを防ぐこともできます。つまり、Amazon Inspector は、オブジェクトが調査結果レポートであり、それらのレポートがアカウントによって作成され、条件で指定されたリージョンにある場合にのみ、S3 オブジェクトをキーで暗号化することができます。

  Amazon Inspector が追加のアカウントで指定したアクションを実行することを許可するには、追加のアカウントごとに ARN をこの条件に追加します。例:

  ```
  "aws:SourceArn": [
      "arn:aws:inspector2:us-east-1:111122223333:report/*",
      "arn:aws:inspector2:us-east-1:444455556666:report/*",
      "arn:aws:inspector2:us-east-1:123456789012:report/*"
  ]
  ```

`aws:SourceAccount` と `aws:SourceArn` の条件によって指定されたアカウントは、一致する必要があります。

これらの条件は、 とのトランザクション中に Amazon Inspector が混乱した代理として使用されるのを防ぐのに役立ちます AWS KMS。お勧めしませんが、ステートメントからこれらの条件を削除できます。

9. キーポリシーの更新が完了したら、[変更を保存する] を選択します。

ステップ 4: 調査結果レポートを設定しエクスポートする

アクセス許可を確認し、調査結果レポートを暗号化して保存するリソースを設定したら、レポートを設定してエクスポートします。

調査結果レポートの設定とエクスポートをするには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ナビゲーションペインで [検出結果] の [すべての検出結果] を選択します。
3. (オプション) 検出結果 テーブルの上にあるフィルターバーを使用して、レポートに含める検出結果を指定するフィルター条件を追加します。条件を追加すると、Amazon Inspector は条件に一致する検出結果のみを含むようにテーブルを更新します。このテーブルには、レポートに含まれるデータのプレビューが表示されます。

> **Note**
>
> フィルター条件を追加することをお勧めします。これを行わない場合、レポートには、ステータス AWS リージョン がアクティブ である現在の のすべての検出結果のデータが含まれます。お客様が組織の Amazon Inspector 管理者である場合、これには、組織内のすべてのメンバーアカウントの検出結果データが含まれます。
> レポートにすべてまたは多数の検出結果データが含まれている場合、レポートの生成とエクスポートには時間がかかり、エクスポートは一度に 1 つのレポートしか生成できません。

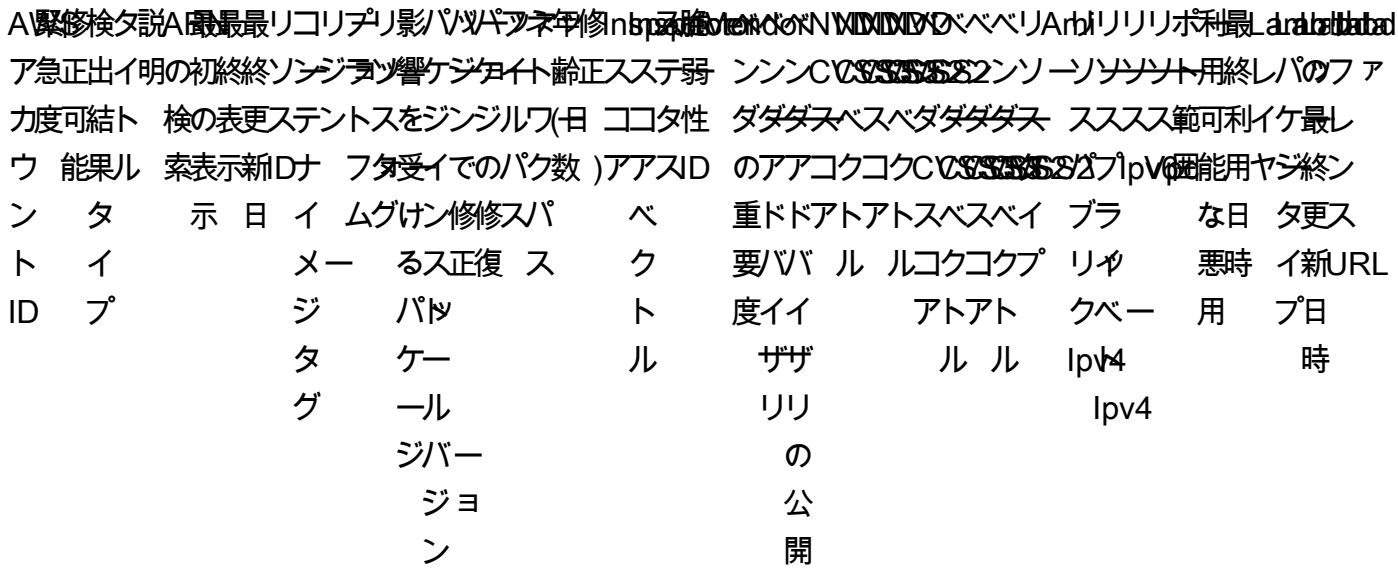
4. [検出結果をエクスポート] を選択します。
5. [エクスポート設定] セクションの [エクスポートファイルタイプ] で、レポートのファイル形式を指定します。

- データを含む JavaScript Object Notation (.json) ファイルを作成するには、JSON を選択します。

  [JSON] オプションを選択した場合、レポートには各検出結果のすべてのフィールドが含まれます。使用可能な JSON フィールドのリストについては、Amazon Inspector API リファレンスの「検出結果データタイプ」を参照してください。

- データを含むカンマ区切り値 (CSV) ファイルを作成するには、[CSV] を選択します。

  CSV オプションを選択した場合、レポートには各検出結果のフィールドのサブセット、つまり検出結果の主要な属性を報告する約 45 のフィールドのみが含まれます。フィールドには、[検出結果のタイプ]、[タイトル]、[重要度]、[ステータス]、[説明]、[初回確認日]、[最終確認日]、[使用可能な修正]、[AWS アカウント ID]、[リソース ID]、[リソースタグ]、および [対策] が含まれます。これらのフィールドに加え、各検出結果のスコアリングの詳細と参照 URL をキャプチャするフィールドもあります。以下は、検出結果レポートの CSV ヘッダーのサンプルです。



6. [エクスポートの場所] の [S3 URI] で、レポートを保存する S3 バケットを指定します。

   - アカウントが所有するバケットにレポートを保存するには、[S3 の参照] を選択します。Amazon Inspector は、アカウントの S3 バケットのテーブルを表示します。使用したいバケットを選択し、[選択] を選択します。

> **Tip**
>
> レポートの Amazon S3 パスプレフィックスも指定するには、[S3 URI] ボックスの値にスラッシュ (/) とプレフィックスを追加します。その後、Amazon Inspector はレポートをバケットに追加するときにプレフィックスを含め、Amazon S3 はプレフィックスで指定されたパスを生成します。
> 例えば、 AWS アカウント ID をプレフィックスとして使用し、アカウント ID が 111122223333 の場合、S3 URI ボックスの値**/111122223333**に を追加します。
> [プレフィックス] は、バケット内のディレクトリパスと類似しています。これにより、類似ファイルをファイルシステム上のフォルダにまとめて保存する場合と同様に、バケット内の類似オブジェクトをまとめてグループ化できます。詳細については、「Amazon Simple Storage Service ユーザーガイド」の「フォルダを使用して Amazon S3 コンソールのオブジェクトを整理する」を参照してください。

- 別のアカウントが所有するバケットにレポートを保存するには、そのバケットの URI を入力します。たとえば **s3://DOC-EXAMPLE_BUCKET**、DOC-EXAMPLE_BUCKET はバケットの名前です。バケット所有者は、この情報をバケットのプロパティで確認できます。

7. KMS キー AWS KMS key で、レポートの暗号化に使用する を指定します。

   - ご自分のアカウントのキーを使用するには、リストからキーを選択します。リストには、アカウントのカスタマーマネージド、対称暗号化 KMS キーが表示されます。
   - 別のアカウントによって所有されているキーを使用するには、キーの Amazon リソースネーム (ARN) を入力します。キーの所有者は、キーのプロパティでこの情報をユーザーに代わって確認できます。詳細については、「AWS Key Management Service デベロッパーガイド」の「キー ID と ARN の検索」を参照してください。

8. [エクスポート] をクリックします。

Amazon Inspector は調査結果レポートを生成し、指定した KMS キーで暗号化して、指定した S3 バケットに追加します。レポートに含めるように選択した検出結果の数によっては、このプロセスに数分または数時間かかる場合があります。エクスポートが完了すると、Amazon Inspector は調査結果レポートが正常にエクスポートされたことを示すメッセージを表示します。オプションで、メッセージ内の [レポートを表示] を選択し、Amazon S3 のレポートに移動します。

一度に 1 つのレポートしかエクスポートできないことに注意してください。エクスポートが進行中の場合は、エクスポートが完了するまで待ってから別のレポートをエクスポートしてください。

エクスポートエラーのトラブルシューティング

調査結果レポートをエクスポートしようとしたときにエラーが発生した場合、Amazon Inspector では、エラーを説明するメッセージが表示されます。このトピックに記載されている情報を参考にして、考えられるエラーの原因と解決策を特定してください。

例えば、S3 バケットが現在の にあり AWS リージョン 、バケットのポリシーで Amazon Inspector がバケットにオブジェクトを追加できることを確認します。また、 AWS KMS key が現在のリージョンで有効になっていることを確認し、キーポリシーが Amazon Inspector にキーの使用を許可していることを確認します。

エラーに対処したら、もう一度レポートのエクスポートを試します。

「Cannot have multiple reports」エラー

レポートを作成しようとしても、Amazon Inspector が既にレポートを生成している場合、「Reason: Cannot have multiple reports in-progress」というエラーが表示されます。このエラーは、Amazon Inspector がアカウントに対して一度に 1 つのレポートしか生成できないために発生します。

エラーを解決するには、他のレポートが終了するのを待つか、キャンセルしてから新しいレポートをリクエストしてください。

GetFindingsReportStatus オペレーションを使用してレポートのステータスを確認できます。このオペレーションは、現在生成されているレポートのレポート ID を返します。

必要に応じて、 `GetFindingsReportStatus`オペレーションで指定されたレポート ID を使用して、 CancelFindingsReportオペレーションを使用して現在進行中のエクスポートをキャンセルできます。

Amazon を使用した Amazon Inspector の検出結果へのカスタムレスポンスの作成 EventBridge

Amazon Inspector は、新しく生成された検出結果、新しく集約された検出結果、および検出結果の状態の変化に関する Amazon EventBridge のイベントを作成します。`updatedAt`、`lastObservedAt` フィールドの変更以外は、新しいイベントを発行します。つまり、リソースの再起動やリソースに関連付けられているタグの変更などのアクションを実行すると、検出結果に関する新しいイベントが生成されます。ただし、`id` フィールド内の検出結果 ID は変わりません。イベントは、ベストエフォートベースで出力されます。

> **Note**
>
> アカウントが Amazon Inspector の委任管理者である場合、 は、イベントの発生元のメンバーアカウントに加えて、イベントをアカウントに EventBridge 発行します。

Amazon Inspector で EventBridge イベントを使用すると、タスクを自動化してAmazon Inspectorの検出結果によって明らかになったセキュリティ問題に対処できます。

Amazon Inspector は、同じリージョンのデフォルトのイベントバスにイベントを送信します。つまり、Amazon Inspector を実行している各リージョンのイベントルールを設定して、そのリージョンのイベントを確認する必要があります。

EventBridge イベントに基づいて Amazon Inspector の検出結果に関する通知を受信するには、Amazon Inspector の EventBridge ルールとターゲットを作成する必要があります。このルールにより EventBridge 、 は、Amazon Inspector が生成する検出結果の通知を、ルールで指定されたターゲットに送信できます。詳細については、「Amazon EventBridge ユーザーガイド」の「Amazon ルール EventBridge 」を参照してください。

イベントスキーマ

以下は、EC2 の検出結果イベントの Amazon Inspector イベントフォーマットの例です。他の検出結果タイプやイベントタイプのスキーマの例については、「EventBridge スキーマ」を参照してください。


```
{
    "version": "0",
    "id": "66a7a279-5f92-971c-6d3e-c92da0950992",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-19T22:46:15Z",
    "region": "us-east-1",
    "resources": ["i-0c2a343f1948d5205"],
    "detail": {
        "awsAccountId": "111122223333",
        "description": "\n It was discovered that the sound subsystem in the Linux
 kernel contained a\n race condition in some situations. A local attacker could use
 this to cause\n a denial of service (system crash).",
        "exploitAvailable": "YES",
```

```
        "exploitabilityDetails": {
            "lastKnownExploitAt": "Oct 24, 2022, 11:08:59 PM"
        },
        "findingArn": "arn:aws:inspector2:us-east-1:111122223333:finding/FINDING_ID",
        "firstObservedAt": "Jan 19, 2023, 10:46:15 PM",
        "fixAvailable": "YES",
        "lastObservedAt": "Jan 19, 2023, 10:46:15 PM",
        "packageVulnerabilityDetails": {
            "cvss": [{
                "baseScore": 4.7,
                "scoringVector": "CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:N/A:H",
                "source": "NVD",
                "version": "3.1"
            }],
            "referenceUrls": ["https://lore.kernel.org/all/
CAFcO6XN7JDM4xSXGhtusQfS2mSBcx50VJKwQpCq=WeLt57aaZA@mail.gmail.com/", "https://
ubuntu.com/security/notices/USN-5792-1", "https://ubuntu.com/security/notices/
USN-5791-2", "https://ubuntu.com/security/notices/USN-5791-1", "https://ubuntu.com/
security/notices/USN-5793-2", "https://git.kernel.org/pub/scm/linux/kernel/git/
torvalds/linux.git/commit/?id=8423f0b6d513b259fdab9c9bf4aaa6188d054c2d", "https://
ubuntu.com/security/notices/USN-5793-1", "https://ubuntu.com/security/notices/
USN-5792-2", "https://ubuntu.com/security/notices/USN-5791-3", "https://ubuntu.com/
security/notices/USN-5793-4", "https://ubuntu.com/security/notices/USN-5793-3",
 "https://git.kernel.org/linus/8423f0b6d513b259fdab9c9bf4aaa6188d054c2d(6.0-rc5)",
 "https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-3303"],
            "relatedVulnerabilities": [],
            "source": "UBUNTU_CVE",
            "sourceUrl": "https://people.canonical.com/~ubuntu-security/cve/2022/
CVE-2022-3303.html",
            "vendorCreatedAt": "Sep 27, 2022, 11:15:00 PM",
            "vendorSeverity": "medium",
            "vulnerabilityId": "CVE-2022-3303",
            "vulnerablePackages": [{
                "arch": "X86_64",
                "epoch": 0,
                "fixedInVersion": "0:5.15.0.1027.31~20.04.16",
                "name": "linux-image-aws",
                "packageManager": "OS",
                "remediation": "apt update && apt install --only-upgrade linux-image-
aws",
                "version": "5.15.0.1026.30~20.04.16"
            }]
        },
        "remediation": {
```

```
            "recommendation": {
                "text": "None Provided"
            }
        },
        "resources": [{
            "details": {
                "awsEc2Instance": {
                    "iamInstanceProfileArn": "arn:aws:iam::111122223333:instance-
profile/AmazonSSMRoleForInstancesQuickSetup",
                    "imageId": "ami-0b7ff1a8d69f1bb35",
                    "ipV4Addresses": ["172.31.85.212", "44.203.45.27"],
                    "ipV6Addresses": [],
                    "launchedAt": "Jan 19, 2023, 7:53:14 PM",
                    "platform": "UBUNTU_20_04",
                    "subnetId": "subnet-8213f2a3",
                    "type": "t2.micro",
                    "vpcId": "vpc-ab6650d1"
                }
            },
            "id": "i-0c2a343f1948d5205",
            "partition": "aws",
            "region": "us-east-1",
            "type": "AWS_EC2_INSTANCE"
        }],
        "severity": "MEDIUM",
        "status": "ACTIVE",
        "title": "CVE-2022-3303 - linux-image-aws",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Jan 19, 2023, 10:46:15 PM"
    }
}
```


Amazon Inspector の検出結果を通知する EventBridge ルールの作成

Amazon Inspector の検出結果の可視性を高めるために、 を使用してメッセージングハブ EventBridge に送信される自動検出結果アラートを設定できます。このトピックでは、E メール、Slack、または Amazon Chime に、CRITICAL および HIGH の重要度の検出結果に対するアラートを送信する方法を説明します。Amazon Simple Notification Service トピックをセットアップし、そのトピックを EventBridge イベントルールに接続する方法について説明します。

Step 1. Amazon SNS トピックおよびエンドポイントの設定

自動アラートを設定するには、まず Amazon Simple Notification Service でトピックを設定し、エンドポイントを追加する必要があります。詳細については、「SNS ガイド」を参照してください。

この手順では、Amazon Inspector の検出結果データを送信したい場所を設定します。SNS トピックは、 EventBridge イベントルールの作成中または作成後にイベントルールに追加できます。

Email setup

SNS トピックの作成

1. https://console.aws.amazon.com/sns/v3/home で Amazon SNS コンソール にサインインします。
2. ナビゲーションペインから [トピック] 、[トピックの作成] を選択します。
3. [トピックの作成] セクションで [標準] を選択します。次に、**`Inspector_to_Email`** のようなトピック名を入力します。その他の詳細はオプションです。
4. [Create Topic] (トピックの作成) を選択します。新しいトピックの詳細が表示された新しいパネルが開きます。
5. [サブスクリプション] セクションで、[サブスクリプションの作成] を選択します。
6. a. [Protocol] (プロトコル) メニューから [Email] (E メール) を選択します。
   b. [エンドポイント] フィールドに、通知を受信する E メールアドレスを入力します。

   > **Note**
   >
   > サブスクリプションを作成後、E メールクライアントを通じてサブスクリプションを確認する必要があります。

   c. [サブスクリプションを作成] を選択します。
7. 受信トレイでサブスクリプションのメッセージを検索し、[サブスクリプションを確認] を選択します。

Slack setup

SNS トピックの作成

1. https://console.aws.amazon.com/sns/v3/home で Amazon SNS コンソール にサインインします。

2. ナビゲーションペインから [トピック] 、[トピックの作成] を選択します。
3. [トピックの作成] セクションで [標準] を選択します。次に、**Inspector_to_Slack** のようなトピック名を入力します。その他の詳細はオプションです。[トピックを作成] を選択してエンドポイントの作成を完了します。

AWS Chatbot クライアントの設定

1. で AWS Chatbot コンソールに移動しますhttps://console.aws.amazon.com/chatbot/。
2. [設定されたクライアント] ペインから [新しいクライアントを設定] を選択します。
3. [Slack] を選択して確認し、[設定] を選択します。

> **Note**
>
> Slack を選択するときは、[許可] を選択してチャネルにアクセスするために、 AWS Chatbot のためのアクセス許可を確認する必要があります。

4. [新しいチャネルを設定] を選択し、設定の詳細ペインを開きます。
   a. チャンネルの名前を入力します。
   b. [Slack チャネル] で、使用したいチャネルを選択します。
   c. Slack で、チャネル名を右クリックして [リンクのコピー] を選択して、コピーのリンクを選択することでプライベートチャンネルのチャネル ID をコピーします。
   d. AWS Chatbot ウィンドウで AWS Management Console、Slack からコピーしたチャネル ID をプライベートチャネル ID フィールドに貼り付けます。
   e. [アクセス許可] で、まだロールを持っていない場合は、テンプレートを使用して IAM ロールを作成することを選択します。
   f. [ポリシー] テンプレートで、[通知の許可] を選択します。これは の IAM ポリシーテンプレートです AWS Chatbot。このポリシーは、 CloudWatch アラーム、イベント、ログ、および Amazon SNS トピックに必要な読み取りおよび一覧表示のアクセス許可を提供します。
   g. チャネルガードレールポリシー で、AmazonInspector2ReadOnlyAccess を選択します。
   h. 以前に SNS トピックを作成したリージョンを選択し、Slack チャネルに通知を送信するために作成した Amazon SNS トピックを選択します。
5. [Configure] (設定) を選択します。

Amazon Chime setup

SNS トピックの作成

1. https://console.aws.amazon.com/sns/v3/home で Amazon SNS コンソール にサインインします。
2. ナビゲーションペインから [トピック]、[トピックの作成] を選択します。
3. [トピックの作成] セクションで [標準] を選択します。次に、**Inspector_to_Chime** のようなトピック名を入力します。その他の詳細はオプションです。[トピックを作成] を選択して完了します。

AWS Chatbot クライアントの設定

1. で AWS Chatbot コンソールに移動しますhttps://console.aws.amazon.com/chatbot/。
2. [Configured clients] (設定されたクライアント) パネルから [Configure new client] (新しいクライアントを設定) を選択します。
3. [Chime]、[設定] を選択して確認します。
4. [Configuration details] (設定の詳細) ペインから、チャンネルの名前を入力します。
5. Amazon Chime で目的のチャットルームを開きます。
   a. 右上の歯車アイコンを選択してから、[Manage webhooks and bots] (ウェブフックとボットの管理) を選択します。
   b. [Copy URL] (URL をコピー)を選択し、Webhook URL をクリップボードにコピーします。
6. AWS Chatbot ウィンドウで AWS Management Console、コピーした URL を Webhook URL フィールドに貼り付けます。
7. [アクセス許可] で、まだロールを持っていない場合は、テンプレートを使用して IAM ロールを作成することを選択します。
8. [ポリシー] テンプレートで、[通知の許可] を選択します。これは の IAM ポリシーテンプレートです AWS Chatbot。 CloudWatch アラーム、イベント、ログ、および Amazon SNS トピックに必要な読み取りおよび一覧表示のアクセス許可を提供します。
9. 以前に SNS トピックを作成したリージョンを選択し、Amazon Chime ルームに通知を送信するために作成した Amazon SNS トピックを選択します。
10. [Configure] (設定) を選択します。

Step 2. Amazon Inspector の検出結果の EventBridge ルールを作成する

1. https://console.aws.amazon.com/events/ で Amazon EventBridge コンソールを開きます。
2. ナビゲーションペインから [ルール] を選択し、[ルールの作成] を選択します。
3. ルールの名前と必要に応じて説明を入力します。
4. [イベントパターンを持つルール] を選択してから、[次へ] を選択します。
5. [イベントパターン] ペインで [カスタムパターン (JSON エディタ)] を選択します。
6. 以下の JSON をエディタに貼り付けます。

```
{
  "source": ["aws.inspector2"],
  "detail-type": ["Inspector2 Finding"],
  "detail": {
    "severity": ["HIGH", "CRITICAL"],
    "status": ["ACTIVE"]
  }
}
```

> **Note**
>
> このパターンは、Amazon Inspector によって検出されたアクティブな CRITICAL または HIGH 重要度の検出結果に関する通知を送信します。

   イベントパターンの入力が終了したら、[次へ] を選択します。
7. [ターゲットを選択] ページで、[AWS のサービス] を選択します。次に、[ターゲットタイプの選択] で [SNS トピック] を選択します。
8. [トピック] で、ステップ 1 で作成した SNS トピックの名前を選択します。次いで、[次へ] を選択します。
9. 必要に応じてオプションのタグを追加し、[次へ] を選択します。
10. ルールを確認し、[ルールの作成] を選択します。

EventBridge for Amazon Inspector マルチアカウント環境

Amazon Inspector の委任された管理者の場合、 EventBridge メンバーアカウントからの該当する検出結果に基づいてルールがアカウントに表示されます。前のセクションで説明したように、管理者ア

カウント EventBridge で を使用して検出結果通知を設定すると、複数のアカウントに関する通知が届きます。つまり、ご自身のアカウントで生成された結果とイベントに加え、メンバーアカウントによって生成された結果とイベントが通知されます。

結果の JSON 詳細から `accountId` を使用して、Amazon Inspector の検出結果の元となったメンバーアカウントを特定することができます。

Amazon Inspector による SBOM のエクスポート

Amazon Inspector コンソールまたは API を使用して、リソースのソフトウェア部品表 (SBOM) を生成できます。SBOM は、コードベースに含まれるすべてのオープンソースソフトウェアコンポーネントとサードパーティソフトウェアコンポーネントのネストされたインベントリです。Amazon Inspector は、環境内の個々のリソースに SBOM を提供します。Amazon Inspector からエクスポートされた SBOM は、最も一般的に使用されているパッケージや組織全体の関連する脆弱性など、ソフトウェアサプライに関する情報を可視化するのに役立ちます。

Amazon Inspector によってアクティブにモニタリングされている、サポートされているすべてのリソースの SBOM をエクスポートできます。リソースのステータスは Amazon Inspector による AWS 環境のカバレッジを評価する で確認できます。

> **Note**
>
> Amazon Inspector では Windows EC2 インスタンス用の SBOM のエクスポートはサポートしていません。

Amazon Inspector 形式

Amazon Inspector では、CycloneDX 1.4 および SPDX 2.3 互換フォーマットでの SBOM のエクスポートをサポートしています。Amazon Inspector では、選択した Amazon S3 バケットに SBOM を `JSON` ファイルとしてエクスポートします。

> **Note**
>
> Amazon Inspector からの SPDX 形式のエクスポートは SPDX 2.3 を使用するシステムと互換性がありますが、クリエイティブコモンズゼロ (CC0) フィールドは含まれていません。これは、このフィールドを含めると、ユーザーがマテリアルを再配布したり編集したりできるようになるためです。

Amazon Inspector の CycloneDX 1.4 SBOM 形式の例

```
                    {
  "bomFormat": "CycloneDX",
```

```
  "specVersion": "1.4",
  "version": 1,
  "metadata": {
    "timestamp": "2023-06-02T01:17:46Z",
    "component": null,
    "properties": [
      {
        "name": "imageId",
        "value":
 "sha256:c8ee97f7052776ef223080741f61fcdf6a3a9107810ea9649f904aa4269fdac6"
      },
      {
        "name": "architecture",
        "value": "arm64"
      },
      {
        "name": "accountId",
        "value": "111122223333"
      },
      {
        "name": "resourceType",
        "value": "AWS_ECR_CONTAINER_IMAGE"
      }
    ]
  },
  "components": [
    {
      "type": "library",
      "name": "pip",
      "purl": "pkg:pypi/pip@22.0.4?path=usr/local/lib/python3.8/site-packages/
pip-22.0.4.dist-info/METADATA",
      "bom-ref": "98dc550d1e9a0b24161daaa0d535c699"
    },
    {
      "type": "application",
      "name": "libss2",
      "purl": "pkg:dpkg/libss2@1.44.5-1+deb10u3?
arch=ARM64&epoch=0&upstream=libss2-1.44.5-1+deb10u3.src.dpkg",
      "bom-ref": "2f4d199d4ef9e2ae639b4f8d04a813a2"
    },
    {
      "type": "application",
      "name": "liblz4-1",
```

```
      "purl": "pkg:dpkg/liblz4-1@1.8.3-1+deb10u1?
arch=ARM64&epoch=0&upstream=liblz4-1-1.8.3-1+deb10u1.src.dpkg",
      "bom-ref": "9a6be8907ead891b070e60f5a7b7aa9a"
    },
    {
      "type": "application",
      "name": "mawk",
      "purl": "pkg:dpkg/mawk@1.3.3-17+b3?
arch=ARM64&epoch=0&upstream=mawk-1.3.3-17+b3.src.dpkg",
      "bom-ref": "c2015852a729f97fde924e62a16f78a5"
    },
    {
      "type": "application",
      "name": "libgmp10",
      "purl": "pkg:dpkg/libgmp10@6.1.2+dfsg-4+deb10u1?
arch=ARM64&epoch=2&upstream=libgmp10-6.1.2+dfsg-4+deb10u1.src.dpkg",
      "bom-ref": "52907290f5beef00dff8da77901b1085"
    },
    {
      "type": "application",
      "name": "ncurses-bin",
      "purl": "pkg:dpkg/ncurses-bin@6.1+20181013-2+deb10u3?
arch=ARM64&epoch=0&upstream=ncurses-bin-6.1+20181013-2+deb10u3.src.dpkg",
      "bom-ref": "cd20cfb9ebeeadba3809764376f43bce"
    }
  ],
  "vulnerabilities": [
    {
      "id": "CVE-2022-40897",
      "affects": [
        {
          "ref": "a74a4862cc654a2520ec56da0c81cdb3"
        },
        {
          "ref": "0119eb286405d780dc437e7dbf2f9d9d"
        }
      ]
    }
  ]
}
```

Amazon Inspector の SPDX 2.3 SBOM 形式の例


```
{
 "name": "409870544328/EC2/i-022fba820db137c64/ami-074ea14c08effb2d8",
 "spdxVersion": "SPDX-2.3",
 "creationInfo": {
  "created": "2023-06-02T21:19:22Z",
  "creators": [
   "Organization: 409870544328",
   "Tool: Amazon Inspector SBOM Generator"
  ]
 },
 "documentNamespace": "EC2://i-022fba820db137c64/AMAZON_LINUX_2/null/x86_64",
 "comment": "",
 "packages": [{
   "name": "elfutils-libelf",
   "versionInfo": "0.176-2.amzn2",
   "downloadLocation": "NOASSERTION",
   "sourceInfo": "/var/lib/rpm/Packages",
   "filesAnalyzed": false,
   "externalRefs": [{
    "referenceCategory": "PACKAGE-MANAGER",
    "referenceType": "purl",
    "referenceLocator": "pkg:rpm/elfutils-libelf@0.176-2.amzn2?
arch=X86_64&epoch=0&upstream=elfutils-libelf-0.176-2.amzn2.src.rpm"
   }],
   "SPDXID": "SPDXRef-Package-rpm-elfutils-libelf-ddf56a513c0e76ab2ae3246d9a91c463"
  },
  {
   "name": "libcurl",
   "versionInfo": "7.79.1-1.amzn2.0.1",
   "downloadLocation": "NOASSERTION",
   "sourceInfo": "/var/lib/rpm/Packages",
   "filesAnalyzed": false,
   "externalRefs": [{
     "referenceCategory": "PACKAGE-MANAGER",
     "referenceType": "purl",
     "referenceLocator": "pkg:rpm/libcurl@7.79.1-1.amzn2.0.1?
arch=X86_64&epoch=0&upstream=libcurl-7.79.1-1.amzn2.0.1.src.rpm"
    },
    {
     "referenceCategory": "SECURITY",
```

```
      "referenceType": "vulnerability",
      "referenceLocator": "CVE-2022-32205"
     }
    ],
    "SPDXID": "SPDXRef-Package-rpm-libcurl-710fb33829bc5106559bcd380cddb7d5"
   },
   {
    "name": "hunspell-en-US",
    "versionInfo": "0.20121024-6.amzn2.0.1",
    "downloadLocation": "NOASSERTION",
    "sourceInfo": "/var/lib/rpm/Packages",
    "filesAnalyzed": false,
    "externalRefs": [{
     "referenceCategory": "PACKAGE-MANAGER",
     "referenceType": "purl",
     "referenceLocator": "pkg:rpm/hunspell-en-US@0.20121024-6.amzn2.0.1?
arch=NOARCH&epoch=0&upstream=hunspell-en-US-0.20121024-6.amzn2.0.1.src.rpm"
    }],
    "SPDXID": "SPDXRef-Package-rpm-hunspell-en-US-de19ae0883973d6cea5e7e079d544fe5"
   },
   {
    "name": "grub2-tools-minimal",
    "versionInfo": "2.06-2.amzn2.0.6",
    "downloadLocation": "NOASSERTION",
    "sourceInfo": "/var/lib/rpm/Packages",
    "filesAnalyzed": false,
    "externalRefs": [{
      "referenceCategory": "PACKAGE-MANAGER",
      "referenceType": "purl",
      "referenceLocator": "pkg:rpm/grub2-tools-minimal@2.06-2.amzn2.0.6?
arch=X86_64&epoch=1&upstream=grub2-tools-minimal-2.06-2.amzn2.0.6.src.rpm"
     },
     {
      "referenceCategory": "SECURITY",
      "referenceType": "vulnerability",
      "referenceLocator": "CVE-2021-3981"
     }
    ],
    "SPDXID": "SPDXRef-Package-rpm-grub2-tools-minimal-c56b7ea76e5a28ab8f232ef6d7564636"
   },
   {
    "name": "unixODBC-devel",
    "versionInfo": "2.3.1-14.amzn2",
    "downloadLocation": "NOASSERTION",
```

```
   "sourceInfo": "/var/lib/rpm/Packages",
   "filesAnalyzed": false,
   "externalRefs": [{
    "referenceCategory": "PACKAGE-MANAGER",
    "referenceType": "purl",
    "referenceLocator": "pkg:rpm/unixODBC-devel@2.3.1-14.amzn2?
arch=X86_64&epoch=0&upstream=unixODBC-devel-2.3.1-14.amzn2.src.rpm"
   }],
   "SPDXID": "SPDXRef-Package-rpm-unixODBC-devel-1bb35add92978df021a13fc9f81237d2"
  }
 ],
 "relationships": [{
   "spdxElementId": "SPDXRef-DOCUMENT",
   "relatedSpdxElement": "SPDXRef-Package-rpm-elfutils-libelf-
ddf56a513c0e76ab2ae3246d9a91c463",
   "relationshipType": "DESCRIBES"
  },
  {
   "spdxElementId": "SPDXRef-DOCUMENT",
   "relatedSpdxElement": "SPDXRef-Package-rpm-yajl-8476ce2db98b28cfab2b4484f84f1903",
   "relationshipType": "DESCRIBES"
  },
  {
   "spdxElementId": "SPDXRef-DOCUMENT",
   "relatedSpdxElement": "SPDXRef-Package-rpm-unixODBC-
devel-1bb35add92978df021a13fc9f81237d2",
   "relationshipType": "DESCRIBES"
  }
 ],
 "SPDXID": "SPDXRef-DOCUMENT"
}
```


SBOM 用フィルター

SBOM をエクスポートする際、フィルターを追加してリソースの特定のサブセットに関するレポートを作成できます。フィルターを指定しない場合、サポート対象のすべてのアクティブリソースの SBOM がエクスポートされます。また、委任された管理者の場合は、メンバー全員のリソースも含まれます。以下のフィルタが利用可能です。

- AccountID — このフィルターは、特定の AccountID に関連付けられている任意のリソースの SBOM をエクスポートするために使用できます。

- EC2 インスタンスタグ — このフィルターは、特定のタグを持つ EC2 インスタンスの SBOM をエクスポートするために使用できます。
- 関数名 — このフィルターは、特定の Lambda 関数の SBOM をエクスポートするために使用できます。
- イメージタグ — このフィルターは、特定のタグが付いたコンテナイメージの SBOM をエクスポートするために使用できます。
- Lambda 関数タグ — このフィルターを使用して、特定のタグを持つ Lambda 関数の SBOM をエクスポートできます。
- リソースタイプ — このフィルターは、リソースタイプ EC2/ECR/Lambda をフィルタリングするために使用できます。
- リソース ID — このフィルターは、特定のリソースの SBOM をエクスポートするために使用できます。
- リポジトリ名 — このフィルターを使用して、特定のリポジトリ内のコンテナイメージの SBOM を生成できます。

SBOM の設定とエクスポート

SBOMsエクスポートするには、まず Amazon S3 バケットと Amazon Inspector が使用できる AWS KMS キーを設定する必要があります。フィルタを使用して、リソースの特定のサブセットの SBOM をエクスポートできます。 AWS Organization 内の複数のアカウントの SBOMs をエクスポートするには、Amazon Inspector の委任された管理者としてサインインしているときに、以下の手順に従います。

前提条件

- Amazon Inspector によってアクティブにモニタリングされているサポート対象リソース。
- Amazon Inspector にオブジェクトの追加を許可するポリシーが設定された Amazon S3 バケット。ポリシーの設定については、「Configure export permissions」を参照してください。
- Amazon Inspector が を使用してレポートを暗号化できるようにするポリシーで設定された AWS KMS キー。ポリシーの設定については、「エクスポート用の AWS KMS キーを設定する」を参照してください。

> **Note**
>
> 以前に Amazon S3 バケットと検出結果のエクスポート用の AWS KMS キーを設定したことがある場合は、SBOM エクスポートに同じバケットとキーを使用できます。

任意のアクセス方法を選択して、SBOM をエクスポートします。

Console

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、SBOM をエクスポートするリソースがあるリージョンを選択します。
3. ナビゲーションペインの [SBOM をエクスポート] を選択します。
4. (オプション) [SBOM のエクスポート] ページで、[フィルターを追加] メニューを使用して、レポートを作成するリソースのサブセットを選択します。フィルターが指定されていない場合、Amazon Inspector はすべてのアクティブなリソースのレポートをエクスポートします。委任された管理者の場合は、これには組織内のすべてのアクティブなリソースが含まれます。
5. [エクスポート設定] で、SBOM に必要な形式を選択します。
6. Amazon S3 URI を入力するか、[Amazon S3 を参照] を選択して SBOM を保存する Amazon S3 の場所を選択します。
7. Amazon Inspector がレポートの暗号化に使用するように設定したAWS KMS キーを入力します。

API

- リソースSBOMs をプログラムでエクスポートするには、Amazon Inspector API の CreateSbomExportオペレーションを使用します。

  リクエストでは、`reportFormat` パラメータを使用して SBOM 出力形式を指定し、`CYCLONEDX_1_4` または `SPDX_2_3` を選択します。`s3Destination` パラメータは必須で、Amazon Inspector による書き込みを許可するポリシーで設定された S3 バケットを指定する必要があります。オプションで、`resourceFilterCriteria` パラメーターを使用して、レポートのスコープを特定のリソースに制限します。

AWS CLI

- を使用してリソースSBOMs をエクスポートするには、次のコマンド AWS Command Line Interface を実行します。

  ```
  aws inspector2 create-sbom-export --report-format
  FORMAT --s3-destination bucketName=DOC-EXAMPLE-
  BUCKET1,keyPrefix=PREFIX,kmsKeyArn=arn:aws:kms:Region:111122223333:key/123
  ```

  リクエスト内の *FORMAT* を任意の形式 (CYCLONEDX_1_4 または SPDX_2_3) に置き換えます。次に、S3 送信先の *user input placeholders* を、エクスポート先の S3 バケットの名前、S3 での出力に使用するプレフィックス、およびレポートの暗号化に使用している KMS キーの ARN に置き換えます。

Amazon Inspector 脆弱性データベースの検索

Amazon Inspector 脆弱性データベースで脆弱性と露出 (CVEsを検索できます。Amazon Inspector は、脆弱性データベースからの情報を使用して、CVE ID に関連する詳細を生成します。これらの詳細には、CVE の詳細ページでアクセスできます。

このトピックでは、CVE ID を使用して Amazon Inspector の vulernability データベースを検索し、CVE の詳細ページをインターペットする方法について説明します。検出結果の詳細については、「」を参照してくださいAmazon Inspector の検出結果の詳細。

> **Note**
>
> Amazon Inspector は、データベース内の他のソフトウェアの脆弱性を追跡して検出結果を生成します。ただし、Amazon Inspector は、CVEs「検出プラットフォーム」セクションにリストされているプラットフォームを持つ CVE のみをサポートします。現在、CVE 検索は をサポートしていませんMicrosoft Windows。

脆弱性データベースの検索

このセクションでは、コンソールおよび Amazon Inspector API で脆弱性データベースを検索する方法について説明します。

> **Note**
>
> 脆弱性データベースを検索する AWS リージョン 前に、現在の で Amazon Inspector をアクティブ化する必要があります。

Console

1. Amazon Inspector コンソール (https://console.aws.amazon.com/inspector/) にサインインします。
2. ナビゲーションペインから、脆弱性データベース検索 を選択します。
3. 検索バーに CVE ID を入力し、検索 を選択します。

API

Amazon Inspector SearchVulnerabilities API を実行し、 として 1 つの CVE ID を `filterCriteria`の形式で指定します`CVE-<year>-<ID>`。

CVE の詳細について

このセクションでは、CVE の詳細ページをインターペットする方法について説明します。

CVE の詳細

CVE の詳細セクションには、次の情報が含まれます。

- CVE の説明と ID
- CVE 重要度
- 共通脆弱性スコアリングシステム (CVSS) およびエクスプロイト予測スコアリングシステム (EPSS) スコア
- 検出プラットフォーム

> **Note**
>
> このフィールドが空の場合、Amazon Inspector は CVE ID の検出をサポートしていません。

- 共通脆弱性列挙 (CWE)
- ベンダーが作成および更新した日付

脆弱性インテリジェンス

脆弱性インテリジェンスセクションには、エクスプロイトターゲットや最後に公開された既知のエクスプロイト日などの脅威インテリジェンスデータが表示されます。

また、Cybersecurity and Infrastructure Security Agency (CISA) のデータも提供します。これには、修復アクション、CVE が既知の脆弱性カタログに追加された日付、CISA が連邦政府機関が CVE を修復することを期待する日付が含まれます。

リファレンス

リファレンスセクションには、CVE に関する詳細情報のリソースへのリンクがあります。

Amazon Inspector EventBridge イベントの Amazon イベントスキーマ Amazon Inspector

モニタリングやイベント管理システムなどの他のアプリケーション、サービス、システムとの統合をサポートするために、Amazon Inspector は検出結果をイベント EventBridge として Amazon に自動的に発行します。 EventBridge は、アプリケーションやその他の から AWS Lambda 関数、Amazon Simple Notification Service トピック、Amazon Kinesis Data Streams ストリームなどのターゲット AWS のサービス にリアルタイムデータのストリームを配信するサーバーレスイベントバスサービスです。 EventBridge および EventBridge イベントの詳細については、「Amazon ユーザーガイド EventBridge 」を参照してください。

Amazon Inspector は、検出結果、リソースカバレッジの変更、個々のリソースの初回スキャンに関するイベントを公開します。各イベントは、イベントの EventBridge スキーマに準拠する JSON オブジェクトです AWS 。データは EventBridge イベントとして構成されているため、他のアプリケーション、サービス、ツールを使用して、検出結果やサポートされている Amazon Inspector イベントをより簡単にモニタリング、処理、対応できます。

トピック

Amazon Inspector の Amazon EventBridge ベーススキーマ

Amazon Inspector の EventBridge イベントの基本スキーマの例を次に示します。イベントの詳細は、イベントのタイプによって異なります。

```
{
    "version": "0",
    "id": "Event ID",
    "detail-type": "Inspector2 *event type*",
    "source": "aws.inspector2",
    "account": "AWS ##### ID (string)",
    "time": "event timestamp (string)",
    "region": "AWS ##### (string)",
```

```
    "resources": [
        *IDs or ARNs of the resources involved in the event*
    ],
    "detail": {
        *Details of an Amazon Inspector event type*
    }
}
```

Amazon Inspector 検出結果イベントスキーマの例

Amazon Inspector の検出結果の EventBridge イベントのスキーマの例を次に示します。検出結果イベントは、Amazon Inspector がリソースの 1 つでソフトウェアの脆弱性またはネットワークの問題を特定したときに作成されます。このタイプのイベントに対して通知を作成するガイドについては、「Amazon を使用した Amazon Inspector の検出結果へのカスタムレスポンスの作成 EventBridge」を参照してください。

以下のフィールドは検出結果イベントを識別します。

- `detail-type` フィールドは `Inspector2 Finding` に設定されます。
- `detail` オブジェクトは検出結果を記述します。

オプションを選択すると、さまざまなリソースのイベントスキーマの検出結果と検出結果タイプが表示されます。

Amazon EC2 package vulnerability finding

```
{
    "version": "0",
    "id": "66a7a279-5f92-971c-6d3e-c92da0950992",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-19T22:46:15Z",
    "region": "us-east-1",
    "resources": ["i-0c2a343f1948d5205"],
    "detail": {
        "awsAccountId": "111122223333",
        "description": "\n It was discovered that the sound subsystem in the Linux
 kernel contained a\n race condition in some situations. A local attacker could use
 this to cause\n a denial of service (system crash).",
```

```
        "exploitAvailable": "YES",
        "exploitabilityDetails": {
            "lastKnownExploitAt": "Oct 24, 2022, 11:08:59 PM"
        },
        "findingArn": "arn:aws:inspector2:us-east-1:111122223333:finding/
FINDING_ID",
        "firstObservedAt": "Jan 19, 2023, 10:46:15 PM",
        "fixAvailable": "YES",
        "lastObservedAt": "Jan 19, 2023, 10:46:15 PM",
        "packageVulnerabilityDetails": {
            "cvss": [{
                "baseScore": 4.7,
                "scoringVector": "CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:N/A:H",
                "source": "NVD",
                "version": "3.1"
            }],
            "referenceUrls": ["https://lore.kernel.org/all/
CAFcO6XN7JDM4xSXGhtusQfS2mSBcx50VJKwQpCq=WeLt57aaZA@mail.gmail.com/", "https://
ubuntu.com/security/notices/USN-5792-1", "https://ubuntu.com/security/notices/
USN-5791-2", "https://ubuntu.com/security/notices/USN-5791-1", "https://ubuntu.com/
security/notices/USN-5793-2", "https://git.kernel.org/pub/scm/linux/kernel/git/
torvalds/linux.git/commit/?id=8423f0b6d513b259fdab9c9bf4aaa6188d054c2d", "https://
ubuntu.com/security/notices/USN-5793-1", "https://ubuntu.com/security/notices/
USN-5792-2", "https://ubuntu.com/security/notices/USN-5791-3", "https://ubuntu.com/
security/notices/USN-5793-4", "https://ubuntu.com/security/notices/USN-5793-3",
 "https://git.kernel.org/linus/8423f0b6d513b259fdab9c9bf4aaa6188d054c2d(6.0-rc5)",
 "https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-3303"],
            "relatedVulnerabilities": [],
            "source": "UBUNTU_CVE",
            "sourceUrl": "https://people.canonical.com/~ubuntu-security/cve/2022/
CVE-2022-3303.html",
            "vendorCreatedAt": "Sep 27, 2022, 11:15:00 PM",
            "vendorSeverity": "medium",
            "vulnerabilityId": "CVE-2022-3303",
            "vulnerablePackages": [{
                "arch": "X86_64",
                "epoch": 0,
                "fixedInVersion": "0:5.15.0.1027.31~20.04.16",
                "name": "linux-image-aws",
                "packageManager": "OS",
                "remediation": "apt update && apt install --only-upgrade linux-
image-aws",
                "version": "5.15.0.1026.30~20.04.16"
            }]
```

```
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
            }
        },
        "resources": [{
            "details": {
                "awsEc2Instance": {
                    "iamInstanceProfileArn": "arn:aws:iam::111122223333:instance-
profile/AmazonSSMRoleForInstancesQuickSetup",
                    "imageId": "ami-0b7ff1a8d69f1bb35",
                    "ipV4Addresses": ["172.31.85.212", "44.203.45.27"],
                    "ipV6Addresses": [],
                    "launchedAt": "Jan 19, 2023, 7:53:14 PM",
                    "platform": "UBUNTU_20_04",
                    "subnetId": "subnet-8213f2a3",
                    "type": "t2.micro",
                    "vpcId": "vpc-ab6650d1"
                }
            },
            "id": "i-0c2a343f1948d5205",
            "partition": "aws",
            "region": "us-east-1",
            "type": "AWS_EC2_INSTANCE"
        }],
        "severity": "MEDIUM",
        "status": "ACTIVE",
        "title": "CVE-2022-3303 - linux-image-aws",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Jan 19, 2023, 10:46:15 PM"
    }
}
```

Amazon EC2 network reachability finding

```
{
    "version": "0",
    "id": "d0384f63-1621-1b75-d014-a5e45628ef3e",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
```

```
    "account": "111122223333",
    "time": "2023-01-20T09:17:57Z",
    "region": "us-east-1",
    "resources": ["i-0a96278c2206a8e4b"],
    "detail": {
        "awsAccountId": "111122223333",
        "description": "On the instance i-0a96278c2206a8e4b, the port range
 22-22 is reachable from the InternetGateway igw-72069c09 from an attached ENI
 eni-0976efe678170408f.",
        "findingArn": "arn:aws:inspector2:us-east-1:111122223333:finding/
FINDING_ID",
        "firstObservedAt": "Jan 20, 2023, 9:17:57 AM",
        "lastObservedAt": "Jan 20, 2023, 9:17:57 AM",
        "networkReachabilityDetails": {
            "networkPath": {
                "steps": [{
                    "componentId": "igw-72069c09",
                    "componentType": "AWS::EC2::InternetGateway"
                }, {
                    "componentId": "acl-91d74eec",
                    "componentType": "AWS::EC2::NetworkAcl"
                }, {
                    "componentId": "sg-0aaed0af450bd0165",
                    "componentType": "AWS::EC2::SecurityGroup"
                }, {
                    "componentId": "eni-0976efe678170408f",
                    "componentType": "AWS::EC2::NetworkInterface"
                }, {
                    "componentId": "i-0a96278c2206a8e4b",
                    "componentType": "AWS::EC2::Instance"
                }]
            },
            "openPortRange": {
                "begin": 22,
                "end": 22
            },
            "protocol": "TCP"
        },
        "remediation": {
            "recommendation": {
                "text": "You can restrict access to your instance by modifying the
 Security Groups or ACLs in the network path."
            }
        },
```

```
        "resources": [{
            "details": {
                "awsEc2Instance": {
                    "iamInstanceProfileArn": "arn:aws:iam::111122223333:instance-
profile/AmazonSSMRoleForInstancesQuickSetup",
                    "imageId": "ami-0b5eea76982371e91",
                    "ipV4Addresses": ["3.89.90.19", "172.31.93.57"],
                    "ipV6Addresses": [],
                    "keyName": "example-inspector-test",
                    "launchedAt": "Jan 19, 2023, 7:25:02 PM",
                    "platform": "AMAZON_LINUX_2",
                    "subnetId": "subnet-8213f2a3",
                    "type": "t2.micro",
                    "vpcId": "vpc-ab6650d1"
                }
            },
            "id": "i-0a96278c2206a8e4b",
            "partition": "aws",
            "region": "us-east-1",
            "type": "AWS_EC2_INSTANCE"
        }],
        "severity": "MEDIUM",
        "status": "ACTIVE",
        "title": "Port 22 is reachable from an Internet Gateway",
        "type": "NETWORK_REACHABILITY",
        "updatedAt": "Jan 20, 2023, 9:17:57 AM"
    }
}
```


Amazon ECR package vulnerability finding


```
{
    "version": "0",
    "id": "5b52952e-26df-3a51-6d14-4dbe737e58ec",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-19T21:59:00Z",
    "region": "us-east-1",
    "resources": [
```

```
        "arn:aws:ecr:us-east-1:111122223333:repository/inspector2/
sha256:98f0304b3a3b7c12ce641177a99d1f3be56f532473a528fda38d53d519cafb13"
    ],
    "detail": {
        "awsAccountId": "111122223333",
        "description": "libcurl would reuse a previously created connection even
 when a TLS or SSHrelated option had been changed that should have prohibited
 reuse.libcurl keeps previously used connections in a connection pool for
 subsequenttransfers to reuse if one of them matches the setup. However, several TLS
 andSSH settings were left out from the configuration match checks, making themmatch
 too easily.",
        "exploitAvailable": "NO",
        "findingArn": "arn:aws:inspector2:us-east-1:111122223333:finding/
FINDING_ID",
        "firstObservedAt": "Jan 19, 2023, 9:59:00 PM",
        "fixAvailable": "YES",
        "inspectorScore": 7.5,
        "inspectorScoreDetails": {
            "adjustedCvss": {
                "adjustments": [],
                "cvssSource": "NVD",
                "score": 7.5,
                "scoreSource": "NVD",
                "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N",
                "version": "3.1"
            }
        },
        "lastObservedAt": "Jan 19, 2023, 9:59:00 PM",
        "packageVulnerabilityDetails": {
            "cvss": [
                {
                    "baseScore": 5,
                    "scoringVector": "AV:N/AC:L/Au:N/C:N/I:P/A:N",
                    "source": "NVD",
                    "version": "2.0"
                },
                {
                    "baseScore": 7.5,
                    "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N",
                    "source": "NVD",
                    "version": "3.1"
                }
            ],
            "referenceUrls": [
```

```
                "https://hackerone.com/reports/1555796",
                "https://security.gentoo.org/glsa/202212-01",
                "https://lists.debian.org/debian-lts-announce/2022/08/
msg00017.html",
                "https://www.debian.org/security/2022/dsa-5197"
            ],
            "relatedVulnerabilities": [],
            "source": "NVD",
            "sourceUrl": "https://nvd.nist.gov/vuln/detail/CVE-2022-27782",
            "vendorCreatedAt": "Jun 2, 2022, 2:15:00 PM",
            "vendorSeverity": "HIGH",
            "vendorUpdatedAt": "Jan 5, 2023, 5:51:00 PM",
            "vulnerabilityId": "CVE-2022-27782",
            "vulnerablePackages": [
                {
                    "arch": "X86_64",
                    "epoch": 0,
                    "fixedInVersion": "0:7.61.1-22.el8_6.3",
                    "name": "libcurl",
                    "packageManager": "OS",
                    "release": "22.el8",
                    "remediation": "yum update libcurl",
                    "sourceLayerHash":
 "sha256:38a980f2cc8accf69c23deae6743d42a87eb34a54f02396f3fcfd7c2d06e2c5b",
                    "version": "7.61.1"
                },
                {
                    "arch": "X86_64",
                    "epoch": 0,
                    "fixedInVersion": "0:7.61.1-22.el8_6.3",
                    "name": "curl",
                    "packageManager": "OS",
                    "release": "22.el8",
                    "remediation": "yum update curl",
                    "sourceLayerHash":
 "sha256:38a980f2cc8accf69c23deae6743d42a87eb34a54f02396f3fcfd7c2d06e2c5b",
                    "version": "7.61.1"
                }
            ]
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
            }
```

```
        },
        "resources": [
            {
                "details": {
                    "awsEcrContainerImage": {
                        "architecture": "amd64",
                        "imageHash":
 "sha256:98f0304b3a3b7c12ce641177a99d1f3be56f532473a528fda38d53d519cafb13",
                        "imageTags": [
                            "o3"
                        ],
                        "platform": "ORACLE_LINUX_8",
                        "pushedAt": "Jan 19, 2023, 7:38:39 PM",
                        "registry": "111122223333",
                        "repositoryName": "inspector2"
                    }
                },
                "id": "arn:aws:ecr:us-east-1:111122223333:repository/inspector2/
sha256:98f0304b3a3b7c12ce641177a99d1f3be56f532473a528fda38d53d519cafb13",
                "partition": "aws",
                "region": "us-east-1",
                "type": "AWS_ECR_CONTAINER_IMAGE"
            }
        ],
        "severity": "HIGH",
        "status": "ACTIVE",
        "title": "CVE-2022-27782 - libcurl, curl",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Jan 19, 2023, 9:59:00 PM"
    }
}
```

Lambda package vulnerability finding

```
{
    "version": "0",
    "id": "040bb590-3a12-353f-ecb1-05e54b0fbea7",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-19T19:20:25Z",
```

```
    "region": "us-east-1",
    "resources": [
        "arn:aws:lambda:us-east-1:111122223333:function:ExampleFunction:$LATEST"
    ],
    "detail": {
        "awsAccountId": "111122223333",
        "description": "Those using Woodstox to parse XML data may be vulnerable to
 Denial of Service attacks (DOS) if DTD support is enabled. If the parser is running
 on user supplied input, an attacker may supply content that causes the parser to
 crash by stackoverflow. This effect may support a denial of service attack.",
        "exploitAvailable": "NO",
        "findingArn": "arn:aws:inspector2:us-east-1:111122223333:finding/
FINDING_ID",
        "firstObservedAt": "Jan 19, 2023, 7:20:25 PM",
        "fixAvailable": "YES",
        "inspectorScore": 7.5,
        "inspectorScoreDetails": {
            "adjustedCvss": {
                "cvssSource": "NVD",
                "score": 7.5,
                "scoreSource": "NVD",
                "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H",
                "version": "3.1"
            }
        },
        "lastObservedAt": "Jan 19, 2023, 7:20:25 PM",
        "packageVulnerabilityDetails": {
            "cvss": [
                {
                    "baseScore": 7.5,
                    "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H",
                    "source": "NVD",
                    "version": "3.1"
                }
            ],
            "referenceUrls": [
                "https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=47434"
            ],
            "relatedVulnerabilities": [],
            "source": "NVD",
            "sourceUrl": "https://nvd.nist.gov/vuln/detail/CVE-2022-40152",
            "vendorCreatedAt": "Sep 16, 2022, 10:15:00 AM",
            "vendorSeverity": "HIGH",
            "vendorUpdatedAt": "Nov 25, 2022, 11:15:00 AM",
```

```
            "vulnerabilityId": "CVE-2022-40152",
            "vulnerablePackages": [
                {
                    "epoch": 0,
                    "filePath": "lib/woodstox-core-6.2.7.jar",
                    "fixedInVersion": "6.4.0",
                    "name": "com.fasterxml.woodstox:woodstox-core",
                    "packageManager": "JAR",
                    "remediation": "Update woodstox-core to 6.4.0",
                    "version": "6.2.7"
                }
            ]
        },
        "remediation": {
            "recommendation": {
                "text": "None Provided"
            }
        },
        "resources": [
            {
                "details": {
                    "awsLambdaFunction": {
                        "architectures": [
                            "X86_64"
                        ],
                        "codeSha256": "+EwrOrht2um4fdVCD73gj
+O7HJIAUvUxi8AD0eKHSkc=",
                        "executionRoleArn": "arn:aws:iam::111122223333:role/
ExampleFunction-ExecutionRole",
                        "functionName": "Example-function",
                        "lastModifiedAt": "Nov 7, 2022, 8:29:27 PM",
                        "packageType": "ZIP",
                        "runtime": "JAVA_11",
                        "version": "$LATEST"
                    }
                },
                "id": "arn:aws:lambda:us-
east-1:111122223333:function:ExampleFunction:$LATEST",
                "partition": "aws",
                "region": "us-east-1",
                "tags": {
                    "TargetAlias": "DeploymentStack",
                    "SoftwareType": "Infrastructure"
                },
```

```
                "type": "AWS_LAMBDA_FUNCTION"
            }
        ],
        "severity": "HIGH",
        "status": "ACTIVE",
        "title": "CVE-2022-40152 - com.fasterxml.woodstox:woodstox-core",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Jan 19, 2023, 7:20:25 PM"
    }
}
```

Lambda code vulnerability finding

```
{
   "version":"0",
   "id":"9df01cb1-df24-bc46-5650-085a4087e7aa",
   "detail-type":"Inspector2 Finding",
   "source":"aws.inspector2",
   "account":"111122223333",
   "time":"2023-12-07T22:14:45Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:lambda:us-east-1:111122223333:function:code-finding:$LATEST"
   ],
   "detail":{
      "awsAccountId":"111122223333",
      "codeVulnerabilityDetails":{
         "detectorId":"python/lambda-override-reserved@v1.0",
         "detectorName":"Override of reserved variable names in a Lambda function",
         "detectorTags":[
            "availability",
            "aws-python-sdk",
            "aws-lambda",
            "data-integrity",
            "maintainability",
            "security",
            "security-context",
            "python"
         ],
         "filePath":{
            "endLine":6,
```

```
            "fileName":"lambda_function.py",
            "filePath":"lambda_function.py",
            "startLine":6
         },
         "ruleId":"Rule-434311"
      },
      "description":"Overriding environment variables that are reserved by AWS
 Lambda might lead to unexpected behavior or failure of the Lambda function.",
      "findingArn":"arn:aws:inspector2:us-east-1:111122223333:finding/FINDING_ID",
      "firstObservedAt":"Aug 8, 2023, 7:33:58 PM",
      "lastObservedAt":"Dec 7, 2023, 10:14:45 PM",
      "remediation":{
         "recommendation":{
            "text":"Your code attempts to override an environment variable that is
 reserved by the Lambda runtime environment. This can lead to unexpected behavior
 and might break the execution of your Lambda function.\n\n[Learn more](https://
docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html#configuration-
envvars-runtime)"
         }
      },
      "resources":[
         {
            "details":{
               "awsLambdaFunction":{
                  "architectures":[
                     "X86_64"
                  ],
                  "codeSha256":"2mtfH+CgubesG6NYpb2zEqBja5WN6FfbH4AAYDuF8RE=",
                  "executionRoleArn":"arn:aws:iam::193043430472:role/service-role/
code-finding-role-7jgg3wan",
                  "functionName":"code-finding",
                  "lastModifiedAt":"Dec 7, 2023, 10:12:48 PM",
                  "packageType":"ZIP",
                  "runtime":"PYTHON_3_7",
                  "version":"$LATEST"
               }
            },
            "id":"arn:aws:lambda:us-east-1:193043430472:function:code-finding:
$LATEST",
            "partition":"aws",
            "region":"us-east-1",
            "type":"AWS_LAMBDA_FUNCTION"
         }
      ],
```

```
      "severity":"HIGH",
      "status":"ACTIVE",
      "title":"Overriding environment variables that are reserved by AWS Lambda
 might lead to unexpected behavior.",
      "type":"CODE_VULNERABILITY",
      "updatedAt":"Dec 7, 2023, 10:14:45 PM"
   }
}
```

> **Note**
>
> 詳細値は、単一の検出結果の JSON の詳細をオブジェクトとして返します。配列内の複数の検出結果をサポートする検出結果レスポンスの構文全体は返されません。

Amazon Inspector の初回スキャン完了イベントスキーマの例

以下は、初期スキャンを完了するための Amazon Inspector EventBridge イベントのイベントスキーマの例です。このイベントは、Amazon Inspector がリソースの 1 つの初回スキャンを完了したときに作成されます。

以下のフィールドは初回スキャン完了イベントを識別します。

- `detail-type` フィールドは `Inspector2 Scan` に設定されます。
- この `detail` オブジェクトには、`CRITICAL`、`HIGH`、および `MEDIUM` など、該当する重要度カテゴリの検出結果の数を詳細に示す `finding-severity-counts` オブジェクトが含まれています。

オプションから選択すると、リソースタイプごとに異なる初回スキャンイベントスキーマが表示されます。

Amazon EC2 instance initial scan

```
{
    "version": "0",
    "id": "28a46762-6ac8-6cc4-4f55-bc9ab99af928",
    "detail-type": "Inspector2 Scan",
```

```
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-20T22:52:35Z",
    "region": "us-east-1",
    "resources": [
        "i-087d63509b8c97098"
    ],
    "detail": {
        "scan-status": "INITIAL_SCAN_COMPLETE",
        "finding-severity-counts": {
            "CRITICAL": 0,
            "HIGH": 0,
            "MEDIUM": 0,
            "TOTAL": 0
        },
        "instance-id": "i-087d63509b8c97098",
        "version": "1.0"
    }
}
```

Amazon ECR image initial scan

```
{
    "version": "0",
    "id": "fdaa751a-984c-a709-44f9-9a9da9cd3606",
    "detail-type": "Inspector2 Scan",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-20T23:15:18Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ecr:us-east-1:111122223333:repository/inspector2"
    ],
    "detail": {
        "scan-status": "INITIAL_SCAN_COMPLETE",
        "repository-name": "arn:aws:ecr:us-east-1:111122223333:repository/
inspector2",
        "finding-severity-counts": {
            "CRITICAL": 0,
            "HIGH": 0,
            "MEDIUM": 0,
```

```
            "TOTAL": 0
        },
        "image-digest":
 "sha256:965fbcae990b0467ed5657caceaec165018ef44a4d2d46c7cdea80a9dff0d1ea",
        "image-tags": [
            "ubuntu22"
        ],
        "version": "1.0"
    }
}
```

Lambda function initial scan

```
{
  "version": "0",
  "id": "4f290a7c-361b-c442-03c8-a629f6f20d6c",
  "detail-type": "Inspector2 Scan",
  "source": "aws.inspector2",
  "account": "111122223333",
  "time": "2023-02-23T18:06:03Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:lambda:us-west-2:111122223333:function:lambda-example:$LATEST"
  ],
  "detail": {
    "scan-status": "INITIAL_SCAN_COMPLETE",
    "finding-severity-counts": {
      "CRITICAL": 0,
      "HIGH": 0,
      "MEDIUM": 0,
      "TOTAL": 0
    },
    "version": "1.0"
  }
}
```

Amazon Inspector カバレッジイベントスキーマの例

以下は、Amazon Inspector EventBridge イベントのイベントスキーマのカバレッジの例です。このイベントは、リソースの Amazon Inspector スキャンカバレッジが変更されたときに作成されます。以下のフィールドはカバレッジイベントを識別します。

- `detail-type` フィールドは `Inspector2 Coverage` に設定されます。
- この `detail` オブジェクトには、リソースの新しいスキャンステータスを示す `scanStatus` オブジェクトが含まれています。

```
{
    "version": "0",
    "id": "000adda5-0fbf-913e-bc0e-10f0376412aa",
    "detail-type": "Inspector2 Coverage",
    "source": "aws.inspector2",
    "account": "111122223333",
    "time": "2023-01-20T22:51:39Z",
    "region": "us-east-1",
    "resources": [
        "i-087d63509b8c97098"
    ],
    "detail": {
        "scanStatus": {
            "reason": "UNMANAGED_EC2_INSTANCE",
            "statusCodeValue": "INACTIVE"
        },
        "scanType": "PACKAGE",
        "eventTimestamp": "2023-01-20T22:51:35.665501Z",
        "version": "1.0"
    }
}
```

Amazon Inspector スキャンを CI/CD パイプラインに統合する

Amazon Inspector のコンテナイメージスキャンを CI/CD パイプラインに直接統合して、ソフトウェアの脆弱性をスキャンし、ビルドの最後にレポートを提供できます。Amazon Inspector が生成する脆弱性レポートにより、デプロイ前にリスクを調査して修正することができます。

Amazon Inspector を CI/CD に統合した場合、Amazon Inspector SBOM Generator と Amazon Inspector スキャン API の組み合わせを利用してコンテナイメージの脆弱性レポートが作成されます。Amazon Inspector SBOM Generator が、提供されたコンテナイメージからソフトウェア部品表 (SBOM) を作成し、Amazon Inspector スキャン API がその SBOM をスキャンして、検出された脆弱性の詳細を含むレポートを作成します。

Amazon Inspector を CI/CD に統合するには、個々の CI/CD ソリューション向けに意図的に作成され、それぞれのマーケットプレイスで入手可能な Amazon Inspector プラグインを利用できます。あるいは、自らスキャン統合をカスタムして作成することもできます。

トピック

プラグインによる統合

Amazon Inspector には、サポートされている CI/CD ソリューション用のプラグインが用意されています。これらのプラグインをそれぞれのマーケットプレイスからインストールし、それらを使用して Amazon Inspector スキャンをパイプラインのビルドステップとして追加できます。プラグインのビルドステップでは、指定したイメージに対して Amazon Inspector SBOM Generator を実行し、生成された SBOM に対して Amazon Inspector スキャン API を実行します。

プラグインによる Amazon Inspector の CI/CD への統合がどのように機能するか、その概要を以下に示します。

1. Amazon Inspector スキャン API へのアクセス AWS アカウント を許可するように を設定します。手順については、「Amazon Inspector CI/CD 統合を使用するように AWS アカウントを設定する」を参照してください。
2. マーケットプレイスから Amazon Inspector プラグインをインストールします。
3. Amazon Inspector SBOM Generator バイナリをインストールして設定します。手順については、「Amazon Inspector SBOM Generator」を参照してください。
4. Amazon Inspector スキャンを CI/CD パイプラインのビルドステップとして追加し、スキャンを設定します。
5. ビルドを実行すると、プラグインはコンテナイメージを入力として受け取り、そのイメージに対して Amazon Inspector SBOM Generator を実行して CycloneDX と互換がある SBOM を生成します。
6. そこから、プラグインは生成された SBOM を Amazon Inspector スキャン API のエンドポイントに送信します。このエンドポイントは、各 SBOM コンポーネントの脆弱性を評価します。
7. Amazon Inspector スキャン API のレスポンスは、CSV、SBOM、JSON、および HTML 形式の脆弱性レポートに変換されます。レポートには、Amazon Inspector が検出したあらゆる脆弱性に関する詳細が含まれています。

サポートされている CI/CD ソリューション

Amazon Inspector は現在、以下の CI/CD ソリューションをサポートしています。プラグインを使用して CI/CD の統合をセットアップする詳細な手順については、CI/CD ソリューションに適したプラグインを以下から選択してください。

- Jenkins プラグイン
- TeamCity プラグイン

カスタム統合

Amazon Inspector から CI/CD ソリューション用のプラグインが提供されていない場合は、Amazon Inspector SBOM Generator と Amazon Inspector スキャン API を組み合わせて、独自にカスタムした CI/CD 統合を作成できます。カスタム統合では、Amazon Inspector SBOM Generator で利用可能なオプションを使用してスキャンを微調整することもできます。

カスタムによる Amazon Inspector の CI/CD への統合がどのように機能するか、その概要を以下に示します。

1. Amazon Inspector スキャン API へのアクセス AWS アカウント を許可するように を設定します。手順については、「Amazon Inspector CI/CD 統合を使用するように AWS アカウントを設定する」を参照してください。
2. Amazon Inspector SBOM Generator バイナリをインストールして設定します。手順については、「Amazon Inspector SBOM Generator」を参照してください。
3. Amazon Inspector SBOM Generator を使用し、コンテナイメージに対して CycloneDX と互換性のある SBOM を生成します。
4. 生成された SBOM に Amazon Inspector スキャン API を使用して、脆弱性レポートを作成します。

カスタム統合を設定する手順については、「Amazon Inspector スキャンを使用してカスタム CI/CD パイプライン統合を独自に作成する」を参照してください。

Amazon Inspector CI/CD 統合を使用するように AWS アカウントを設定する

Amazon Inspector CI/CD 統合 AWS アカウント を使用するには、 にサインアップする必要があります。には、Amazon Inspector スキャン API へのパイプラインアクセスを許可する IAM ロール AWS アカウント が必要です。

以下のトピックのタスクを完了して、 にサインアップし AWS アカウント、管理者ユーザーを作成し、CI/CD 統合用の IAM ロールを設定します。

> **Note**
>
> に既にサインアップしている場合は AWS アカウント、「」にスキップできますCI/CD への統合のための IAM ロールを設定する。

トピック

- にサインアップする AWS アカウント
- 管理ユーザーの作成

- CI/CD への統合のための IAM ロールを設定する

にサインアップする AWS アカウント

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. https://portal.aws.amazon.com/billing/signup を開きます。
2. オンラインの手順に従います。

   サインアップ手順の一環として、通話呼び出しを受け取り、電話キーパッドで検証コードを入力するように求められます。

   にサインアップすると AWS アカウント、 AWS アカウントのルートユーザーが作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティのベストプラクティスとして、管理ユーザーに管理アクセスを割り当て、ルートユーザーアクセスが必要なタスクを実行する場合にのみ、ルートユーザーを使用してください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。https://aws.amazon.com/ の [アカウント] をクリックして、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理することができます。

管理ユーザーの作成

にサインアップしたら AWS アカウント、 を保護し AWS アカウントのルートユーザー、 を有効にして AWS IAM Identity Center、日常的なタスクにルートユーザーを使用しないように管理ユーザーを作成します。

のセキュリティ保護 AWS アカウントのルートユーザー

1. ルートユーザーを選択し、 AWS アカウント E メールアドレスを入力して、アカウント所有者AWS Management Consoleとして にサインインします。次のページでパスワードを入力します。

   ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの「ルートユーザーとしてサインインする」を参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

   手順については、「IAM ユーザーガイド」の AWS アカウント 「ルートユーザーの仮想 MFA デバイスを有効にする (コンソール)」を参照してください。

管理ユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「AWS IAM Identity Centerの有効化」を参照してください。

2. IAM アイデンティティセンターで、管理ユーザーに管理アクセス権を付与します。

   を ID ソース IAM アイデンティティセンターディレクトリ として使用する方法のチュートリアルについては、「 ユーザーガイド」の「デフォルト でユーザーアクセス IAM アイデンティティセンターディレクトリを設定するAWS IAM Identity Center 」を参照してください。

管理ユーザーとしてサインインする

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

  IAM Identity Center ユーザーを使用してサインインする方法については、「 AWS サインイン ユーザーガイド」の AWS 「 アクセスポータルにサインインする」を参照してください。

CI/CD への統合のための IAM ロールを設定する

Amazon Inspector のスキャンを CI/CD パイプラインに統合するには、ソフトウェア部品表 (SBOM) をスキャンする Amazon Inspector スキャン API へのアクセスを許可する IAM ポリシーを作成する必要があります。次に、そのポリシーを IAM ロールにアタッチし、アカウントが Amazon Inspector スキャン API を実行できるようにします。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/iam/ で IAM コンソールを開きます。
2. IAM コンソールのナビゲーションペインで、[ポリシー]、[ポリシーを作成] の順に選択します。
3. [ポリシーエディタ] で [JSON] を選択し、以下のステートメントを貼り付けます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "inspector-scan:ScanSbom",
            "Resource": "*"
        }
    ]
}
```

4. [次へ] を選択します。
5. ポリシーに名前を付けて (例えば InspectorCICDscan-policy)、必要に応じて説明を入力してから、[ポリシーの作成] を選択します。このポリシーは、次の手順で作成するロールにアタッチされます。
6. IAM コンソールのナビゲーションペインで、[ロール]、[新しいロールの作成] の順に選択します。
7. [信頼されたエンティティを選択] で [カスタム信頼ポリシー] を選択し、以下のポリシーを入力します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::{ACCOUNT_ID}:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {}
        }
    ]
}
```

8. [次へ] をクリックします。
9. [許可を追加] で、前に作成したたポリシーを検索して選択し、[次へ] を選択します。

10. ロールに名前を付けて (例えば `InspectorCICDscan-role`)、必要に応じて説明を追加して、`Create Role` を選択します。

Amazon Inspector SBOM Generator

Amazon Inspector SBOM Generator (Sbomgen) は、コンテナイメージのソフトウェア部品表 (SBOM) を作成するバイナリツールです。SBOM は、システムにインストールされているソフトウェアのインベントリを収集したものです。

Sbomgen は、インストールされているパッケージに関する情報が含まれているとわかっているファイルをスキャンすることで機能します。これらのファイルのいずれかが見つかると、ツールはパッケージ名、バージョン、その他のメタデータを抽出します。その後、このパッケージメタデータは CycloneDX SBOM に変換されます。

Sbomgen は、CycloneDX SBOM をファイルまたは STDOUT として提供するためのスタンドアロンツールとして使用できます。また、デプロイパイプラインの一部としてコンテナイメージを自動的にスキャンする Amazon Inspector CI/CD 統合の一部としても使用されます。詳細については、「Amazon Inspector スキャンを CI/CD パイプラインに統合する」を参照してください。

サポートされているパッケージとイメージ形式

現時点では、Sbomgen は以下のパッケージタイプのインベントリを収集できます。

- Alpine APK
- Debian / Ubuntu DPKG
- Red Hat RPM
- `go.mod` および `go mod cache` を介した Go パッケージ
- `pom.properties` を介した Java パッケージ
- `node_modules` 内の `package.json` ファイルを介した Node.js パッケージ
- NuGet ファイル経由の C# パッケージ (`.deps.json`、`csproj`、`Packages.config`、packages.lock.json)
- `installed.json` および `composer.lock` を介した PHP
- `requirements.txt`、`Pipfile.lock`、`poetry.lock`、`egg/wheel` ファイルを介した Python パッケージ
- `Gemfile.lock`、`.gemspec`、およびグローバルにインストールされた gem を介した Ruby パッケージ

- `Cargo.lock` および `Cargo.toml` を介した Rust パッケージ

Sbomgen は、次のイメージのコンテナイメージマニフェスト形式をサポートします。

- OCI イメージマニフェスト
- Docker Image Manifest Version 2、Schema 2
- Docker Image Manifest Version 2、Schema 1
- Docker Image Manifest Version 1

> **Important**
>
> コンテナイメージのサイズが 5 GB を超える場合、レイヤーの数が 60 を超える場合、またはインストールされているパッケージが 2,000 を超える場合、Sbomgen はコンテナイメージをスキャンできません。

Amazon Inspector SBOM Generator のインストール (Sbomgen)

Sbomgen は、Linux オペレーティングシステムでのみ利用できます。コンテナイメージの分析に使用する場合は、Docker、 Podman、containerd などのコンテナサービスがインストールされている必要があります。

最高のパフォーマンスを得るには、以下の最低限のハードウェア要件を満たすシステムからバイナリを実行することをお勧めします。

- 4x Core CPU
- 8 GB RAM

Sbomgen をインストールするには

1. 使用しているアーキテクチャの正しい URL から Sbomgen zip ファイルをダウンロードします。

   Linux AMD64:

   https://amazon-inspector-sbomgen.s3.amazonaws.com/latest/linux/amd64/inspector-sbomgen.zip

Linux ARM64:

https://amazon-inspector-sbomgen.s3.amazonaws.com/latest/linux/arm64/inspector-sbomgen.zip

2. 次のコマンドを使用して、ダウンロードした zip ファイルを解凍します。

```
unzip inspector-sbomgen.zip
```

3. アーカイブに以下のファイルがあるかチェックします。

- `inspector-sbomgen` — SBOM を生成するために実行するバイナリです。
- `README.txt` — Sbomgen の使用に関するドキュメントです。
- `LICENSE.txt` — このファイルには、Sbomgen のソフトウェアライセンスが含まれています。
- `licenses` — このフォルダには、Sbomgen が使用するサードパーティパッケージのライセンス情報が含まれています。
- `checksums.txt` — このファイルは Sbomgen バイナリのハッシュを提供します。
- `sbom.json` — Sbomgen バイナリの CycloneDX SBOM です。

4. (オプション) 以下のコマンドを使用して、バイナリの信頼性と整合性を検証します。

```
sha256sum < inspector-sbomgen
```

- 結果を `checksums.txt` ファイルの内容と比較します。

5. 以下のコマンドを使用して、バイナリに実行権限を付与します。

```
chmod +x inspector-sbomgen
```

6. 以下のコマンドを実行して、Sbomgen が正常にインストールされたことを確認します。

```
./inspector-sbomgen --version
```

次のような出力が表示されます。

```
Version: 1.X.X
```

Sbomgen を使用する

Sbomgen を使用して、コンテナイメージの SBOM を生成できます。

SBOM 生成の結果は、特定のファイルを除外したり、ツールがスキャンするパッケージを定義したりするなどのオプションによってカスタマイズすることもできます。これらのユースケースなどでは、以下のコマンドを実行します。

```
./inspector-sbomgen list-examples
```

コンテナイメージの SBOM を生成し、結果をファイルに出力するには

この例では、*image:tag* をイメージの ID に置き換え、*output_path.json* を出力の保存先のパスに置き換えます。

```
./inspector-sbomgen container --image image:tag -o output_path.json
```

Sbomgen を使用したプライベートレジストリへの認証

プライベートレジストリの認証情報を指定することで、プライベートレジストリでホストされているコンテナから SBOM を生成できます。認証情報は、キャッシュされた認証情報、インタラクティブ方式、または Sbomgen の実行前に環境変数として認証情報を指定する非インタラクティブ方式など、さまざまな方法で提供することができます。

キャッシュされた認証情報を使用した認証 (推奨)

1. Sbomgen は、エージェントで利用可能な場合、キャッシュされた認証情報を使用しようとします。この方法では、まずコンテナレジストリに対して認証を行います。例えば、Docker を使用している場合は、以下の Docker `login` コマンドを使用してレジストリに対して認証を行うことができます。

   ```
   docker login
   ```
2. その後、プライベートレジストリへの認証に成功すると、そのレジストリ内のコンテナイメージで Sbomgen を使用できます。以下の例を使用するには、*image:tag* を、スキャンするイメージの名前に置き換えてください。

   ```
   ./inspector-sbomgen container --image image:tag
   ```

インタラクティブ方式による認証

- この方法では、ユーザー名をパラメータとして指定すると、Sbomgen から必要に応じて安全なパスワードの入力を求められます。以下の例を使用するには、*image:tag* をスキャンするイメージの名前に置き換え、*your_username* をその画像にアクセスできるユーザー名に置き換えます。

```
./inspector-sbomgen container --image image:tag --username
your_username
```

非インタラクティブ方式による認証

- この方法を使用するには、現在のユーザーだけが読める .txt ファイルにパスワードまたはレジストリトークンを保存する必要があります。テキストファイルには、パスワードまたはトークンのみが 1 行にまとめられている必要があります。以下の例を使用するには、*your_username* をユーザー名に置き換え、*password.txt* をパスワードまたはトークンを含むファイルに置き換え、*image:tag* をスキャンするイメージの名前に置き換えます。

```
INSPECTOR_SBOMGEN_USERNAME=your_username\
INSPECTOR_SBOMGEN_PASSWORD=`cat password.txt` \
./inspector-sbomgen container --image image:tag
```

Sbomgen からの出力例

Sbomgen を使用してインベントリが作成されたコンテナイメージの SBOM の例を以下に示します。

コンテナイメージの SBOM


```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.5",
  "serialNumber": "urn:uuid:828875ef-8c32-4777-b688-0af96f3cf619",
  "version": 1,
  "metadata": {
    "timestamp": "2023-11-17T21:36:38Z",
    "tools": [
      {
        "vendor": "Amazon Web Services, Inc. (AWS)",
        "name": "Amazon Inspector SBOM Generator",
        "version": "1.0.0",
        "hashes": [
          {
            "alg": "SHA-256",
            "content":
 "10ab669cfc99774786301a745165b5957c92ed9562d19972fbf344d4393b5eb1"
```

```
          }
        ]
      }
    ],
    "component": {
      "bom-ref": "comp-1",
      "type": "container",
      "name": "fedora:latest",
      "properties": [
        {
          "name": "amazon:inspector:sbom_generator:image_id",
          "value":
"sha256:c81c8ae4dda7dedc0711daefe4076d33a88a69a28c398688090c1141eff17e50"
        },
        {
          "name": "amazon:inspector:sbom_generator:layer_diff_id",
          "value":
"sha256:eddd0d48c295dc168d0710f70364581bd84b1dda6bb386c4a4de0b61de2f2119"
        }
      ]
    }
  },
  "components": [
    {
      "bom-ref": "comp-2",
      "type": "library",
      "name": "dnf",
      "version": "4.18.0",
      "purl": "pkg:pypi/dnf@4.18.0",
      "properties": [
        {
          "name": "amazon:inspector:sbom_generator:source_file_scanner",
          "value": "python-pkg"
        },
        {
          "name": "amazon:inspector:sbom_generator:source_package_collector",
          "value": "python-pkg"
        },
        {
          "name": "amazon:inspector:sbom_generator:source_path",
          "value": "/usr/lib/python3.12/site-packages/dnf-4.18.0.dist-info/METADATA"
        },
        {
          "name": "amazon:inspector:sbom_generator:is_duplicate_package",
```

```
          "value": "true"
        },
        {
          "name": "amazon:inspector:sbom_generator:duplicate_purl",
          "value": "pkg:rpm/fedora/python3-dnf@4.18.0-2.fc39?
arch=noarch&distro=39&epoch=0"
        }
      ]
    },
    {
      "bom-ref": "comp-3",
      "type": "library",
      "name": "libcomps",
      "version": "0.1.20",
      "purl": "pkg:pypi/libcomps@0.1.20",
      "properties": [
        {
          "name": "amazon:inspector:sbom_generator:source_file_scanner",
          "value": "python-pkg"
        },
        {
          "name": "amazon:inspector:sbom_generator:source_package_collector",
          "value": "python-pkg"
        },
        {
          "name": "amazon:inspector:sbom_generator:source_path",
          "value": "/usr/lib64/python3.12/site-packages/libcomps-0.1.20-py3.12.egg-
info/PKG-INFO"
        },
        {
          "name": "amazon:inspector:sbom_generator:is_duplicate_package",
          "value": "true"
        },
        {
          "name": "amazon:inspector:sbom_generator:duplicate_purl",
          "value": "pkg:rpm/fedora/python3-libcomps@0.1.20-1.fc39?
arch=x86_64&distro=39&epoch=0"
        }
      ]
    }
  ]
}
```

Amazon Inspector スキャンを使用してカスタム CI/CD パイプライン統合を独自に作成する

CI/CD マーケットプレイスで入手できる場合は、Amazon Inspector CI/CD プラグインを使用することをお勧めします。利用可能なプラグインのリストについては、「サポートされている CI/CD ソリューション」を参照してください。

Amazon Inspector から CI/CD ソリューション用のプラグインが提供されていない場合は、Amazon Inspector SBOM Generator と Amazon Inspector スキャン API を組み合わせて、独自にカスタムした CI/CD 統合を作成できます。カスタムによる統合では、Amazon Inspector SBOM Generator で利用可能なオプションを使用してスキャンを微調整することもできます。

独自のカスタム統合をセットアップするには

1. Amazon Inspector スキャン API へのアクセス AWS アカウント を許可するように を設定します。手順については、「Amazon Inspector CI/CD 統合を使用するように AWS アカウントを設定する」を参照してください。
2. Amazon Inspector SBOM Generator バイナリをインストールして設定します。手順については、「Amazon Inspector SBOM Generator のインストール (Sbomgen)」を参照してください。
3. SBOM Generator を使用して、スキャンするコンテナイメージの SBOM ファイルを作成します。以下の例を使用するには、*image:id* をスキャンするイメージの名前と置き換え、*sbom_path.json* を SBOM 出力を保存する場所に置き換えます。

   ```
   ./inspector-sbomgen container –image image:id -o sbom_path.json
   ```

4. inspector-scan API を呼び出して、生成された SBOM をスキャンし、脆弱性レポートを提供します。以下の例を使用するには、*sbom_path.json* を CycloneDX と互換性のある有効な SBOM ファイルへのファイルパスに置き換えます。次に、*ENDPOINT* を現在認証 AWS リージョン されている の API エンドポイントに置き換え、*REGION* を対応するリージョンに置き換えます。リージョンとエンドポイントのリストについては、「Amazon Inspector スキャン API のエンドポイント」を参照してください。

   ```
   aws inspector-scan scan-sbom --sbom file://sbom_path.json --endpoint
   "ENDPOINT" --region REGION
   ```

API 出力形式

Amazon Inspector スキャン API は、脆弱性レポートを CycloneDX 1.5 形式で出力することも、Amazon Inspector 検出結果 JSON 形式で出力することもできます。デフォルトは `--output-format` フラグを使用して変更できます。

CycloneDX 1.5 形式の出力例


```
{
  "status": "SBOM parsed successfully, 1 vulnerabilities found",
  "sbom": {
    "bomFormat": "CycloneDX",
    "specVersion": "1.5",
    "serialNumber": "urn:uuid:0077b45b-ff1e-4dbb-8950-ded11d8242b1",
    "metadata": {
      "properties": [
        {
          "name": "amazon:inspector:sbom_scanner:critical_vulnerabilities",
          "value": "1"
        },
        {
          "name": "amazon:inspector:sbom_scanner:high_vulnerabilities",
          "value": "0"
        },
        {
          "name": "amazon:inspector:sbom_scanner:medium_vulnerabilities",
          "value": "0"
        },
        {
          "name": "amazon:inspector:sbom_scanner:low_vulnerabilities",
          "value": "0"
        }
      ],
      "tools": [
        {
          "name": "CycloneDX SBOM API",
          "vendor": "Amazon Inspector",
          "version": "empty:083c9b00:083c9b00:083c9b00"
        }
      ],
      "timestamp": "2023-06-28T14:15:53.760Z"
    },
    "components": [
```

```
      {
        "bom-ref": "comp-1",
        "type": "library",
        "name": "log4j-core",
        "purl": "pkg:maven/org.apache.logging.log4j/log4j-core@2.12.1",
        "properties": [
          {
            "name": "amazon:inspector:sbom_scanner:path",
            "value": "/home/dev/foo.jar"
          }
        ]
      }
    ],
    "vulnerabilities": [
      {
        "bom-ref": "vuln-1",
        "id": "CVE-2021-44228",
        "source": {
          "name": "NVD",
          "url": "https://nvd.nist.gov/vuln/detail/CVE-2021-44228"
        },
        "references": [
          {
            "id": "SNYK-JAVA-ORGAPACHELOGGINGLOG4J-2314720",
            "source": {
              "name": "SNYK",
              "url": "https://security.snyk.io/vuln/SNYK-JAVA-
ORGAPACHELOGGINGLOG4J-2314720"
            }
          },
          {
            "id": "GHSA-jfh8-c2jp-5v3q",
            "source": {
              "name": "GITHUB",
              "url": "https://github.com/advisories/GHSA-jfh8-c2jp-5v3q"
            }
          }
        ],
        "ratings": [
          {
            "source": {
              "name": "NVD",
              "url": "https://www.first.org/cvss/v3-1/"
            },
```

```
            "score": 10.0,
            "severity": "critical",
            "method": "CVSSv31",
            "vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H"
          },
          {
            "source": {
              "name": "NVD",
              "url": "https://www.first.org/cvss/v2/"
            },
            "score": 9.3,
            "severity": "critical",
            "method": "CVSSv2",
            "vector": "AC:M/Au:N/C:C/I:C/A:C"
          },
          {
            "source": {
              "name": "EPSS",
              "url": "https://www.first.org/epss/"
            },
            "score": 0.97565,
            "severity": "none",
            "method": "other",
            "vector": "model:v2023.03.01,date:2023-06-27T00:00:00+0000"
          },
          {
            "source": {
              "name": "SNYK",
              "url": "https://security.snyk.io/vuln/SNYK-JAVA-
ORGAPACHELOGGINGLOG4J-2314720"
            },
            "score": 10.0,
            "severity": "critical",
            "method": "CVSSv31",
            "vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H/E:H"
          },
          {
            "source": {
              "name": "GITHUB",
              "url": "https://github.com/advisories/GHSA-jfh8-c2jp-5v3q"
            },
            "score": 10.0,
            "severity": "critical",
            "method": "CVSSv31",
```

```
            "vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H"
          }
        ],
        "cwes": [
          400,
          20,
          502
        ],
        "description": "Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security
 releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages,
 and parameters do not protect against attacker controlled LDAP and other JNDI related
 endpoints. An attacker who can control log messages or log message parameters can
 execute arbitrary code loaded from LDAP servers when message lookup substitution is
 enabled. From log4j 2.15.0, this behavior has been disabled by default. From version
 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely
 removed. Note that this vulnerability is specific to log4j-core and does not affect
 log4net, log4cxx, or other Apache Logging Services projects.",
        "advisories": [
          {
            "url": "https://www.intel.com/content/www/us/en/security-center/advisory/
intel-sa-00646.html"
          },
          {
            "url": "https://support.apple.com/kb/HT213189"
          },
          {
            "url": "https://msrc-blog.microsoft.com/2021/12/11/microsofts-response-to-
cve-2021-44228-apache-log4j2/"
          },
          {
            "url": "https://logging.apache.org/log4j/2.x/security.html"
          },
          {
            "url": "https://www.debian.org/security/2021/dsa-5020"
          },
          {
            "url": "https://cert-portal.siemens.com/productcert/pdf/ssa-479842.pdf"
          },
          {
            "url": "https://www.oracle.com/security-alerts/alert-cve-2021-44228.html"
          },
          {
            "url": "https://www.oracle.com/security-alerts/cpujan2022.html"
          },
```

```
          {
            "url": "https://cert-portal.siemens.com/productcert/pdf/ssa-714170.pdf"
          },
          {
            "url": "https://lists.fedoraproject.org/archives/list/package-
announce@lists.fedoraproject.org/message/M5CSVUNV4HWZZXGOKNSK6L7RPM7BOKIB/"
          },
          {
            "url": "https://cert-portal.siemens.com/productcert/pdf/ssa-397453.pdf"
          },
          {
            "url": "https://cert-portal.siemens.com/productcert/pdf/ssa-661247.pdf"
          },
          {
            "url": "https://lists.fedoraproject.org/archives/list/package-
announce@lists.fedoraproject.org/message/VU57UJDCFIASIO35GC55JMKSRXJMCDFM/"
          },
          {
            "url": "https://www.oracle.com/security-alerts/cpuapr2022.html"
          },
          {
            "url": "https://twitter.com/kurtseifried/status/1469345530182455296"
          },
          {
            "url": "https://tools.cisco.com/security/center/content/
CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd"
          },
          {
            "url": "https://lists.debian.org/debian-lts-announce/2021/12/msg00007.html"
          },
          {
            "url": "https://www.kb.cert.org/vuls/id/930724"
          }
        ],
        "created": "2021-12-10T10:15:00Z",
        "updated": "2023-04-03T20:15:00Z",
        "affects": [
          {
            "ref": "comp-1"
          }
        ],
        "properties": [
          {
            "name": "amazon:inspector:sbom_scanner:exploit_available",
```

```
            "value": "true"
          },
          {
            "name": "amazon:inspector:sbom_scanner:exploit_last_seen_in_public",
            "value": "2023-03-06T00:00:00Z"
          },
          {
            "name": "amazon:inspector:sbom_scanner:cisa_kev_date_added",
            "value": "2021-12-10T00:00:00Z"
          },
          {
            "name": "amazon:inspector:sbom_scanner:cisa_kev_date_due",
            "value": "2021-12-24T00:00:00Z"
          },
          {
            "name": "amazon:inspector:sbom_scanner:fixed_version:comp-1",
            "value": "2.15.0"
          }
        ]
      }
    ]
  }
}
```

Inspector 形式の出力例

```
                        {
  "status": "SBOM parsed successfully, 1 vulnerability found",
  "inspector": {
    "messages": [
      {
        "name": "foo",
        "purl": "pkg:maven/foo@1.0.0", // Will not exist in output if missing in sbom
        "info": "Component skipped: no rules found."
      }
    ],
    "vulnerability_count": {
      "critical": 1,
      "high": 0,
      "medium": 0,
      "low": 0
    },
```

```
    "vulnerabilities": [
      {
        "id": "CVE-2021-44228",
        "severity": "critical",
        "source": "https://nvd.nist.gov/vuln/detail/CVE-2021-44228",
        "related": [
          "SNYK-JAVA-ORGAPACHELOGGINGLOG4J-2314720",
          "GHSA-jfh8-c2jp-5v3q"
        ],
        "description": "Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security
 releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages,
 and parameters do not protect against attacker controlled LDAP and other JNDI related
 endpoints. An attacker who can control log messages or log message parameters can
 execute arbitrary code loaded from LDAP servers when message lookup substitution is
 enabled. From log4j 2.15.0, this behavior has been disabled by default. From version
 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely
 removed. Note that this vulnerability is specific to log4j-core and does not affect
 log4net, log4cxx, or other Apache Logging Services projects.",
        "references": [
          "https://www.intel.com/content/www/us/en/security-center/advisory/intel-
sa-00646.html",
          "https://support.apple.com/kb/HT213189",
          "https://msrc-blog.microsoft.com/2021/12/11/microsofts-response-to-
cve-2021-44228-apache-log4j2/",
          "https://logging.apache.org/log4j/2.x/security.html",
          "https://www.debian.org/security/2021/dsa-5020",
          "https://cert-portal.siemens.com/productcert/pdf/ssa-479842.pdf",
          "https://www.oracle.com/security-alerts/alert-cve-2021-44228.html",
          "https://www.oracle.com/security-alerts/cpujan2022.html",
          "https://cert-portal.siemens.com/productcert/pdf/ssa-714170.pdf",
          "https://lists.fedoraproject.org/archives/list/package-
announce@lists.fedoraproject.org/message/M5CSVUNV4HWZZXGOKNSK6L7RPM7BOKIB/",
          "https://cert-portal.siemens.com/productcert/pdf/ssa-397453.pdf",
          "https://cert-portal.siemens.com/productcert/pdf/ssa-661247.pdf",
          "https://lists.fedoraproject.org/archives/list/package-
announce@lists.fedoraproject.org/message/VU57UJDCFIASIO35GC55JMKSRXJMCDFM/",
          "https://www.oracle.com/security-alerts/cpuapr2022.html",
          "https://twitter.com/kurtseifried/status/1469345530182455296",
          "https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-
sa-apache-log4j-qRuKNEbd",
          "https://lists.debian.org/debian-lts-announce/2021/12/msg00007.html",
          "https://www.kb.cert.org/vuls/id/930724"
        ],
        "created": "2021-12-10T10:15:00Z",
```

```
        "updated": "2023-04-03T20:15:00Z",
        "properties": {
          "cisa_kev_date_added": "2021-12-10T00:00:00Z",
          "cisa_kev_date_due": "2021-12-24T00:00:00Z",
          "cwes": [
            400,
            20,
            502
          ],
          "cvss": [
            {
              "source": "NVD",
              "severity": "critical",
              "cvss3_base_score": 10.0,
              "cvss3_base_vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H",
              "cvss2_base_score": 9.3,
              "cvss2_base_vector": "AC:M/Au:N/C:C/I:C/A:C"
            },
            {
              "source": "SNYK",
              "severity": "critical",
              "cvss3_base_score": 10.0,
              "cvss3_base_vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H/E:H"
            },
            {
              "source": "GITHUB",
              "severity": "critical",
              "cvss3_base_score": 10.0,
              "cvss3_base_vector": "AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H"
            }
          ],
          "epss": 0.97565,
          "exploit_available": true,
          "exploit_last_seen_in_public": "2023-03-06T00:00:00Z"
        },
        "affects": [
          {
            "installed_version": "pkg:maven/org.apache.logging.log4j/log4j-
core@2.12.1",
            "fixed_version": "2.15.0",
            "path": "/home/dev/foo.jar"
          }
        ]
      }
```

```
    ]
  }
}
```

Amazon Inspector Jenkins プラグインを使用する

このJenkinsプラグインは、Amazon Inspector SBOM Generator バイナリと Amazon Inspector スキャン API を活用してビルドの最後に詳細なレポートを生成するため、デプロイ前にリスクを調査して修正できます。

Amazon Inspector は、CVE に基づいてコンテナイメージをスキャンしてオペレーティングシステムとプログラミング言語パッケージの脆弱性を検出する脆弱性管理サービスです。 CVEs

Amazon Inspector Jenkinsプラグインを使用して、Amazon Inspector 脆弱性スキャンをJenkinsパイプラインに追加できます。

> **Note**
>
> Amazon Inspector の脆弱性スキャンは、検出された脆弱性の数と重大度に基づいて、パイプラインの実行に合格または失敗するように設定できます。

最新バージョンのJenkinsプラグインは、https://plugins.jenkins.io/amazon-inspector-image-scanner/ でJenkinsマーケットプレイスで確認できます。

次の手順では、Amazon Inspector Jenkins プラグインを設定する方法について説明します。

> **Important**
>
> 次の手順を完了する前に、プラグインを実行するために Jenkins をバージョン 2.387.3 以降にアップグレードする必要があります。

Step 1. のセットアップ AWS アカウント

Amazon Inspector スキャン API へのアクセスを許可する IAM ロール AWS アカウント を使用してを設定します。手順については、「Amazon Inspector CI/CD 統合を使用するように AWS アカウントを設定する」を参照してください。

Step 2. Amazon Inspector Jenkins プラグインをインストールする

次の手順では、Jenkinsダッシュボードから Amazon Inspector Jenkins プラグインをインストールする方法について説明します。

1. Jenkins ダッシュボードから Jenkins の管理 を選択し、プラグインの管理 を選択します。
2. 使用可能 を選択します。
3. 使用可能タブで Amazon Amazon Inspector スキャン を検索し、プラグインをインストールします。

(オプション) ステップ 3. Docker 認証情報を に追加する Jenkins

> **Note**
>
> Docker イメージがプライベートリポジトリにある場合にのみ、Docker 認証情報を追加します。それ以外の場合は、この手順をスキップしてください。

次の手順では、JenkinsダッシュボードJenkinsから に Docker 認証情報を追加する方法について説明します。

1. Jenkins ダッシュボードから、Jenkins の管理、認証情報、システム を選択します。
2. グローバル認証情報 を選択し、認証情報 を追加します。
3. Kind で、パスワード のユーザー名を選択します。
4. スコープ で、グローバル (Jenkins、ノード、項目、すべての子項目など) を選択します。
5. 詳細を入力し、OK を選択します。

(オプション) ステップ 4。 AWS 認証情報を追加する

> **Note**
>
> IAM ユーザーに基づいて認証する場合にのみ、 AWS 認証情報を追加します。それ以外の場合は、この手順をスキップしてください。

次の手順では、Jenkinsダッシュボードから AWS 認証情報を追加する方法について説明します。

1. Jenkins ダッシュボードから、Jenkins の管理、認証情報、システム を選択します。
2. グローバル認証情報 を選択し、認証情報 を追加します。
3. Kind で、AWS 認証情報 を選択します。
4. アクセスキー ID やシークレットアクセスキー などの詳細を入力し、OK を選択します。

Step 5. Jenkins スクリプトに CSS サポートを追加する

次の手順では、Jenkiknsスクリプトに CSS サポートを追加する方法について説明します。

1. Jenkins を再起動します。
2. ダッシュボードから、「Jenkins の管理」、「ノード」、「組み込みノード」、「スクリプトコンソール」を選択します。
3. テキストボックスに行 を追加 し`System.setProperty("hudson.model.DirectoryBrowserSupport.CSP", "")`、「 の実行」を選択します。

ステップ 6。Amazon Inspector スキャンをビルドに追加する

Amazon Inspector スキャンをビルドに追加するには、プロジェクトにビルドステップを追加するか、Jenkins 宣言型パイプラインを使用します。

プロジェクトにビルドステップを追加してビルドに Amazon Inspector スキャンする

1. 設定ページで、ビルドステップ まで下にスクロールし、ビルドステップ を追加 を選択します。次に、Amazon Inspector スキャン を選択します。
2. inspector-sbomgen のインストール方法は、自動 と 手動 の 2 つから選択します。

   a. （オプション 1) 自動 を選択して、最新バージョンの inspector-sbomgen をダウンロードします。この方法を選択した場合は、プラグインを実行するシステムに一致する CPU アーキテクチャを必ず選択してください。

   b. （オプション 2) スキャン用に Amazon Inspector SBOM Generator バイナリを設定する場合は、手動 を選択します。この方法を選択した場合は、以前にダウンロードしたバージョンの inspector-sbomgen へのフルパスを必ず指定してください。

   詳細については、「Amazon Inspector SBOM Generator」の「Amazon Inspector SBOM Generator (Sbomgen) のインストール」を参照してください。

3. Amazon Inspector スキャンのビルドステップの設定を完了するには、以下を実行します。

   a. [Image Id] を入力します。イメージはローカル、リモート、アーカイブされたもののいずれでもかまいません。イメージ名は Docker の命名規則に従う必要があります。エクスポートされたイメージを分析する場合は、予想される tar ファイルへのパスを指定します。イメージ ID のパスの例については、以下を参照してください。

      i. ローカルコンテナまたはリモートコンテナの場合: `NAME[:TAG|@DIGEST]`

      ii. tar ファイルの場合: `/path/to/image.tar`

   b. [AWS リージョン] を選択して、スキャンリクエストを送信します。

   c. (オプション) [Docker credentials] で Docker ユーザー名を選択します。これは、コンテナイメージがプライベートリポジトリにある場合にのみ行ってください。

   d. （オプション) 以下のサポートされている AWS 認証方法を指定できます。

      i. （オプション) IAM ロール には、ロール ARN (arn:aws:iam::role/*RoleName*) *AccountNumber*を指定します。

      ii. （オプション) AWS 認証情報 で、IAM ユーザーに基づいて認証する ID を選択します。

      iii. （オプション) AWS プロファイル名 には、プロファイル名を使用して認証するプロファイルの名前を指定します。

   e. (オプション) 重大度ごとに [Vulnerability thresholds] を指定します。スキャン中に指定した数を超えると、イメージのビルドは失敗します。値がすべて `0` の場合、脆弱性が見つかったかどうかに関係なくビルドは成功します。

4. [保存] を選択します。

宣言型パイプラインを使用して Amazon Inspector Jenkins スキャンをビルドに追加する

Jenkins 宣言型パイプラインを使用して、Amazon Inspector スキャンをビルドに自動または手動で追加できます。

SBOMGen 宣言型パイプラインを自動的にダウンロードするには

- Amazon Inspector スキャンをビルドに追加するには、次の構文例を使用します。Amazon Inspector SBOM Generator ダウンロードの任意の OS アーキテクチャに基づいて、*SBOMGEN_SOURCE* を linuxAmd64 または linuxArm64 に置き換えます。プライベートリ

ポジトリを使用している場合は、*IMAGE_PATH* をイメージへのパス (*alpine:latest* など) に、*IAM_ROLE* をステップ 1 で設定した IAM ロールの ARN に、*ID* をDocker認証情報 ID に置き換えます。オプションで脆弱性のしきい値を有効にし、重大度ごとに値を指定できます。

```
pipeline {
   agent any
   stages {
       stage('amazon-inspector-image-scanner') {
           steps {
               script {
               step([
               $class:
 'com.amazon.inspector.jenkins.amazoninspectorbuildstep.AmazonInspectorBuilder',
               sbomgenSource: 'SBOMGEN_SOURCE', // this can be linuxAmd64 or linuxArm64
               archivePath: 'IMAGE_PATH',
               awsRegion: 'REGION',
               iamRole: 'IAM ROLE',
               credentialId: 'Id', // provide empty string if image not in private
 repositories
               awsCredentialId: ''AWS ID;',
               awsProfileName: 'Profile Name',
               isThresholdEnabled: false,
               countCritical: 0,
               countHigh: 0,
               countLow: 10,
               countMedium: 5,
              ])
           }
         }
       }
   }
 }
```

SBOMGen 宣言型パイプラインを手動でダウンロードするには

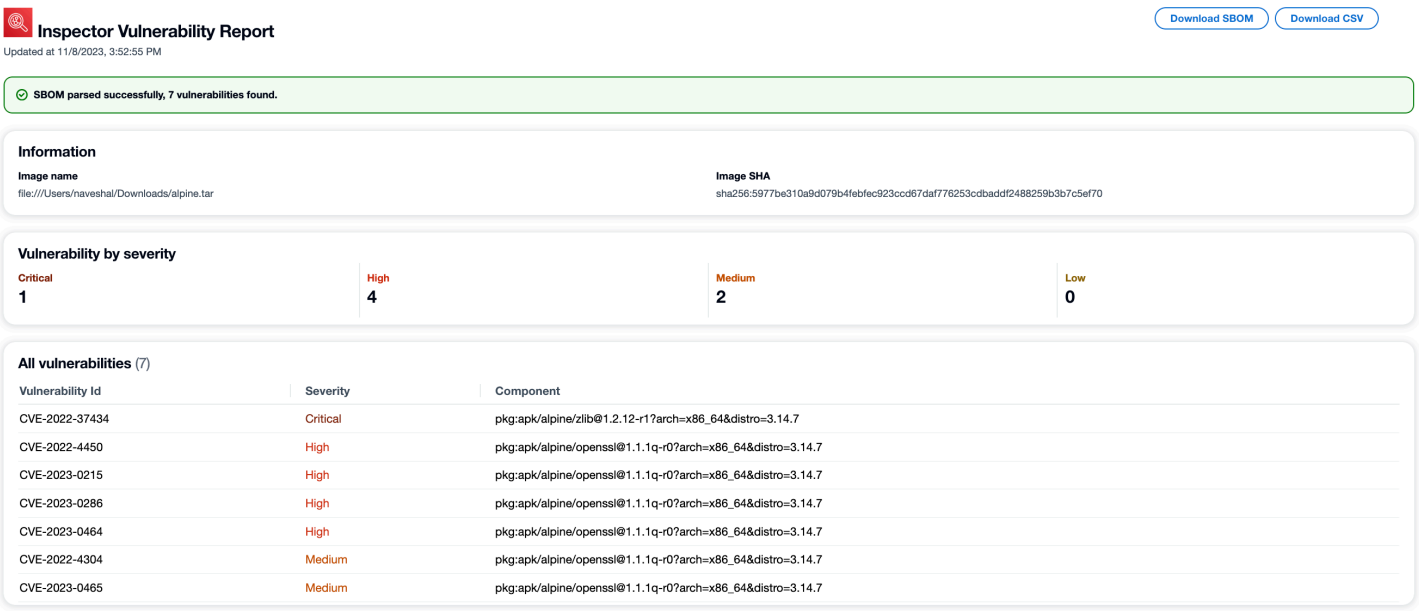
- Amazon Inspector スキャンをビルドに追加するには、次の構文例を使用します。*SBOMGEN_PATH* をステップ 3 でインストールした Amazon Inspector SBOM Generator へのパスに、*IMAGE_PATH* をイメージへのパス (*alpine:latest* など) に、*IAM_ROLE* をステップ 1 で設定した IAM ロールの ARN に、*ID* をプライベートリポジトリを使用している場合は Docker認証情報 ID に置き換えます。オプションで脆弱性のしきい値を有効にし、重大度ごとに値を指定できます。

> **Note**
>
> Jenkins ディレクトリSbomgenに配置し、プラグインの Jenkins ディレクトリへのパス (*/opt/folder/arm64/inspector-sbomgen ##*) を指定します。

```
pipeline {
    agent any
    stages {
        stage('amazon-inspector-image-scanner') {
            steps {
                script {
                step([
                $class:
 'com.amazon.inspector.jenkins.amazoninspectorbuildstep.AmazonInspectorBuilder',
                sbomgenPath: 'SBOMGEN_PATH',
                archivePath: 'IMAGE_PATH',
                awsRegion: 'REGION',
                iamRole: 'IAM ROLE',
                awsCredentialId: ''AWS ID;',
                credentialId: 'Id;', // provide empty string if image not in private
 repositories
                awsProfileName: 'Profile Name',
                isThresholdEnabled: false,
                countCritical: 0,
                countHigh: 0,
                countLow: 10,
                countMedium: 5,
               ])
            }
          }
       }
    }
 }
```

ステップ 7。Amazon Inspector 脆弱性レポートを表示する

1. プロジェクトの新しいビルドを完了します。
2. ビルドが完了したら、結果から出力形式を選択します。HTML を選択した場合は、レポートの JSON SBOM または CSV バージョンをダウンロードするオプションがあります。HTML レポートの例を次に示します。



トラブルシューティング

に Amazon Inspector スキャンプラグインを使用する場合に発生する可能性がある一般的なエラーを次に示しますJenkins。

認証情報または sts 例外のロードに失敗しました

エラー:

```
InstanceProfileCredentialsProvider(): Failed to load credentials or sts exception.
```

リソース

AWS アカウントの `aws_access_key_id`と `aws_secret_access_key` を取得します。`~/.aws/credentials` の `aws_access_key_id` と`aws_secret_access_key` をセットアップします。

Inspector-sbomgen パスエラー

エラー:

```
Exception:com.amazon.inspector.jenkins.amazoninspectorbuildstep.exception.Sbomge
There was an issue running inspector-sbomgen, is /opt/inspector/inspector-sbomgen the correct path?
```

解決策:

問題を解決するには、次の手順を実行します。

1. 正しい OS アーキテクチャ Inspector-sbomgen を Jenkins ディレクトリに配置します。詳細については、Amazon Inspector SBOM Generator」を参照してください。
2. 次のコマンドを使用して、バイナリに実行可能なアクセス許可を付与します: `chmod +x inspector-sbomgen`。
3. など、プラグインに正しいJenkinsマシンパスを指定します`/opt/folder/arm64/inspector-sbomgen`。
4. 設定を保存し、Jenkinsジョブを実行します。

Amazon Inspector TeamCity プラグインを使用する

Amazon Inspector TeamCity プラグインを使用すると、Amazon Inspector の脆弱性スキャンを TeamCity パイプラインに追加することができます。このプラグインは、Amazon Inspector SBOM Generator バイナリと Amazon Inspector スキャン API を活用して、ビルドの最後に詳細なレポートを作成します。これにより、デプロイ前にリスクを調査して修正できます。また、検出された脆弱性の数と重大度に基づいて、パイプラインの実行に成功または失敗するようにスキャンを設定することもできます。

Amazon Inspector は、CVE に基づいてオペレーティングシステムとプログラミング言語パッケージの両方の脆弱性についてコンテナイメージをスキャン AWS する、 が提供する脆弱性管理サービスです。 CVEs Amazon Inspector と CI/CD の統合の詳細については、「Amazon Inspector スキャンを CI/CD パイプラインに統合する」を参照してください。

Amazon Inspector プラグインがサポートするパッケージとコンテナイメージ形式のリストについては、「サポートされているパッケージとイメージ形式」を参照してください。

最新バージョンのプラグインは、https://plugins.jetbrains.com/plugin/23236-amazon-inspector-scanner でマーケットTeamCityプレイスで確認できます。または、このドキュメントの各セクションの手順に従って Amazon Inspector TeamCity プラグインを設定します。

1. をセットアップします AWS アカウント。

   - Amazon Inspector スキャン API へのアクセスを許可する IAM ロール AWS アカウント を使用して を設定します。手順については、「Amazon Inspector CI/CD 統合を使用するように AWS アカウントを設定する」を参照してください。
2. Amazon Inspector TeamCity プラグインをインストールする。

a. ダッシュボードから、[Administration]、[Plugins] の順に移動します。

b. [Amazon Inspector Scans] を検索します。

c. プラグインをインストールします。

3. Amazon Inspector SBOM Generator をインストールする。

- Amazon Inspector SBOM Generator バイナリを Teamcity サーバーディレクトリにインストールします。手順については、「Amazon Inspector SBOM Generator のインストール (Sbomgen)」を参照してください。

4. Amazon Inspector スキャンのビルドステップをプロジェクトに追加する。

a. 設定ページで、「ビルドステップ」までスクロールし、「ビルドステップを追加」を選択し、Amazon Inspector スキャン」を選択します。

b. 以下の詳細を入力して、Amazon Inspector スキャンのビルドステップを設定します。

- ステップ名 を追加します。
- Amazon Inspector SBOM Generator のインストール方法として、自動 と手動 の 2 つから選択します。
  - システムおよび CPU アーキテクチャに基づいて、Amazon Inspector SBOM Generator の最新バージョンを自動ダウンロードします。
  - 手動では、以前にダウンロードしたバージョンの Amazon Inspector SBOM Generator への完全なパスを指定する必要があります。

  詳細については、Amazon Inspector SBOM Generator での Amazon Inspector SBOM Generator (Sbomgen) のインストール」を参照してください。 Amazon Inspector
- [Image Id] を入力します。イメージはローカル、リモート、アーカイブされたもののいずれでもかまいません。イメージ名は Docker の命名規則に従う必要があります。エクスポートされたイメージを分析する場合は、予想される tar ファイルへのパスを指定します。イメージ ID のパスの例については、以下を参照してください。
  - ローカルコンテナまたはリモートコンテナの場合: `NAME[:TAG|@DIGEST]`
  - tar ファイルの場合: `/path/to/image.tar`
- [IAM ロール] には、手順 1 で設定したロールの ARN を入力します。
- [AWS リージョン] を選択して、スキャンリクエストを送信します。

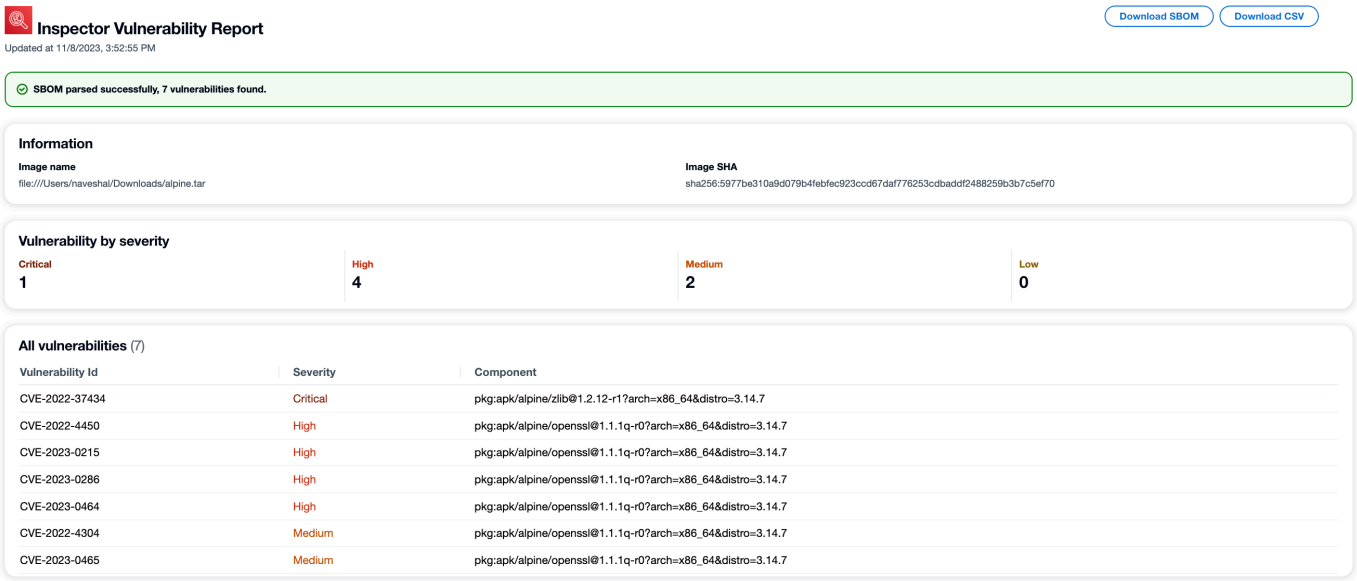
- (オプション) [Docker Authentication] では、[Docker Username] と [Docker Password] を入力します。これは、コンテナイメージがプライベートリポジトリにある場合にのみ行ってください。
- (オプション) AWS 認証 には、 AWS アクセスキー ID と AWS シークレットキーを入力します。これは、 AWS 認証情報に基づいて認証する場合にのみ実行します。
- (オプション) 重大度ごとに [Vulnerability thresholds] を指定します。スキャン中に指定した数を超えると、イメージのビルドは失敗します。値がすべて `0` の場合、見つかった脆弱性の数に関係なくビルドは成功します。

c. [Save] を選択します。

5. Amazon Inspector の脆弱性レポートを確認する。

a. プロジェクトの新しいビルドを完了します。

b. ビルドが完了したら、結果から出力形式を選択します。HTML を選択した場合、JSON SBOM または CSV バージョンのレポートをダウンロードするオプションがあります。HTML レポートの例を以下に示します。



Amazon Inspector CycloneDX 名前空間

Amazon Inspector には、Amazon Inspector SBOM Generator と Amazon Inspector スキャン API によって生成される SBOM に使用するための、CycloneDX 名前空間とプロパティ名が予約されています。このページには、Amazon Inspector ツールを使用して作成された CycloneDX SBOM のコン

ポーネントに追加できるすべてのカスタムキー/値プロパティが記載されています。CycloneDX プロパティ分類の詳細については、公式ドキュメントを参照してください。

amazon:inspector:sbom_scanner 名前空間の分類

amazon:inspector:sbom_scanner 名前空間は Amazon Inspector スキャン API によって使用されます。以下のプロパティがあります。

プロパティ	説明
amazon:inspector:sbom_scanner:critical_vulnerabilities	SBOM で見つかった重大度が重大な脆弱性の総数。
amazon:inspector:sbom_scanner:high_vulnerabilities	SBOM で見つかった重大度が高い脆弱性の総数。
amazon:inspector:sbom_scanner:medium_vulnerabilities	SBOM で見つかった重大度が中程度の脆弱性の総数。
amazon:inspector:sbom_scanner:low_vulnerabilities	SBOM で見つかった重大度が低い脆弱性の総数。
amazon:inspector:sbom_scanner:info	特定のコンポーネントのスキャンコンテキストを提供します。例:「コンポーネントがスキャンされました: 脆弱性は見つかりませんでした」
amazon:inspector:sbom_scanner:warning	特定のコンポーネントがスキャンされなかった理由のコンテキストを提供します。例:「コンポーネントがスキップされました: purl が提供されていません」
amazon:inspector:sbom_scanner:fixed_version: *component_bom_ref*	特定の脆弱性に対して、指定されたコンポーネントの修正バージョンを提供します。
amazon:inspector:sbom_scanner:exploit_available	特定の脆弱性に対して、エクスプロイトが利用可能かどうかを示します。

プロパティ	説明
amazon:inspector:sbom_scanner:exploit_last_seen_in_public	特定の脆弱性に対するエクスプロイトが最後に公開された時期を示します。
amazon:inspector:sbom_scanner:cisa_kev_date_added	脆弱性が CISA の「Known Exploited Vulnerabilities Catalog」に追加された時期を示します。
amazon:inspector:sbom_scanner:cisa_kev_date_due	CISA の「Known Exploited Vulnerabilities Catalog」に従って、脆弱性の修正期限がいつであるかを示します。
amazon:inspector:sbom_scanner:path	対象パッケージ情報を生成したファイルへのパス。

amazon:inspector:sbom_generator 名前空間の分類

amazon:inspector:sbom_generator 名前空間は Amazon Inspector SBOM Generator によって使用されます。以下のプロパティがあります。

プロパティ	説明
amazon:inspector:sbom_generator:os_hostname	インベントリ対象のシステムのホスト名。
amazon:inspector:sbom_generator:kernel_name	インベントリ対象のシステムのカーネル名。
amazon:inspector:sbom_generator:kernel_version	インベントリ対象のシステムのカーネルバージョン。
amazon:inspector:sbom_generator:cpu_architecture	インベントリ対象のシステムの CPU アーキテクチャ (x86_64 など)。
amazon:inspector:sbom_generator:image_id	コンテナイメージの設定ファイルのハッシュ (イメージ ID とも呼ばれます)。

プロパティ	説明
amazon:inspector:sbom_generator:layer_diff_id	非圧縮コンテナイメージレイヤーのハッシュ。
amazon:inspector:sbom_generator:source_file_scanner	パッケージ情報を含むファイルを検出したスキャナー。例: /var/lib/dpkg/status
amazon:inspector:sbom_generator:source_package_collector	特定のファイルからパッケージ名とバージョンを抽出したコレクター。
amazon:inspector:sbom_generator:source_path	対象パッケージ情報が抽出されたファイルへのパス。
amazon:inspector:sbom_generator:is_duplicate_package	対象パッケージが複数のファイルスキャナーによって検出されたことを示します。
amazon:inspector:sbom_generator:go_toolchain	Go 実行ファイルの生成に使用された Go コンパイラーまたはツールチェーンのバージョンを示します。
amazon:inspector:sbom_generator:expires_before	SSL 証明書が有効になる前の日付。
amazon:inspector:sbom_generator:expires_after	SSL 証明書が無効になる日付。
amazon:inspector:sbom_generator:is_expired	SSL 証明書の有効期限が切れているかどうかを示すブール値。

Amazon Inspector による自動リソーススキャン

Amazon Inspector の Amazon EC2 向けエージェントレススキャンはプレビューリリース中です。Amazon EC2 エージェントレススキャン機能の使用には、「AWS のサービス条件」の第 2 項 (「ベータ版とプレビュー」) が適用されます。

Amazon Inspector は、独自の専用スキャンエンジンを使用しています。このエンジンは、ワークロードの侵害、リソースの悪用、データへの不正アクセスを引き起こす可能性のあるソフトウェアの脆弱性やオープンネットワークパスがないかリソースをモニタリングします。Amazon Inspector が脆弱性を検出すると、検出結果が作成されます。検出結果には、脆弱性の修復に役立つ検出に関連する詳細が含まれます。検出結果は Amazon インスペクターコンソールと Amazon Inspector API を使用して確認できます。詳細については、「Amazon Inspector での検出結果の管理」を参照してください。

アクティブ化すると、Amazon Inspector は対象となるすべてのリソースを自動的に検出し、それらのリソースの継続的なスキャンを開始します。Amazon Inspector では、ソフトウェアの脆弱性や意図しないネットワークの露出がないかをスキャンします。Amazon Inspector は、新しいアプリケーションやパッチのインストールなどのイベントに応じたスキャンも実行します。

初めて Amazon Inspector をアクティブ化すると、アカウントはすべてのスキャンタイプに自動的に登録されます。以下のトピックでは、Amazon Inspector が提供するスキャンタイプに関する具体的な詳細について説明します。Amazon Inspector は、脆弱性の影響を受けるリソースタイプに基づいてスキャンタイプを分類します。以下のトピックでは、Amazon Inspector がスキャンするリソース、それらのリソースの新しいスキャンを開始する条件、および各リソースタイプのスキャンを設定する方法について説明します。

トピック

Amazon Inspector を初めてアクティブ化すると、アカウントは Amazon Amazon EC2 スキャン、Amazon ECR スキャン、Lambda 標準スキャンの各スキャンタイプに自動的に登録されます。Lambda コードスキャンは Lambda 関数スキャンのオプションレイヤーで、いつでもアクティブ化できます。

Amazon Inspector のスキャンタイプの概要

Amazon Inspector には、 AWS 環境内の特定のリソースタイプに焦点を当てたさまざまなスキャンタイプが用意されています。

Amazon EC2 スキャン

Amazon EC2 スキャンをアクティブ化すると、Amazon Inspector は Amazon EC2 インスタンスをスキャンして、オペレーティングシステムパッケージとプログラミング言語パッケージの脆弱性、およびネットワーク到達可能性を確認します。Amazon Inspector は EC2 インスタンスをスキャンして、共通脆弱性識別子 (CVE) やネットワークへの露出の問題がないか調べます。Amazon Inspector は、インスタンスにインストールされている SSM エージェントを使用するか、インスタンスの Amazon EBS スナップショットを使用してスキャンを実行します。Amazon EC2 のスキャンの詳細については、「Amazon Inspector による Amazon EC2 インスタンスのスキャン」を参照してください。

Amazon ECR スキャン

Amazon ECR スキャンを有効にすると、Amazon Inspector はプライベートレジストリ内のすべてのベーシックスキャンコンテナリポジトリを継続的スキャンによる拡張スキャンに変換します。また、オプションでこの設定をオンプッシュ時のみスキャンしたり、包含ルールを使用して特定のリポジトリをスキャンしたりするように設定することもできます。過去 30 日以内にプッシュされたイメージ、または過去 90 日以内にプルされたイメージはすべて、最初にスキャンされます。Amazon Inspector はデフォルトで 90 日間イメージをモニタリングし続けます。この設定はいつでも変更できます。Amazon ECR のスキャンの詳細については、「Amazon Inspector による Amazon ECR コンテナイメージのスキャン」を参照してください。

Lambda 標準スキャン

Lambda 標準スキャンをアクティブ化すると、Amazon Inspector はアカウント内の Lambda 関数を検出し、すぐに脆弱性のスキャンを開始します。Amazon Inspector は、新しい Lambda 関数とレイヤーをデプロイ時にスキャンし、それらが更新されたとき、または新しい共通脆弱性識別子 (CVE) が発行されたときに再スキャンします。Lambda 関数のスキャンの詳細については、「Amazon Inspector による AWS Lambda 関数のスキャン」を参照してください。

Lambda 標準スキャン + Lambda コードスキャン

このオプションでは、Lambda 標準スキャンと Lambda コードスキャンを組み合わせることができます。Lambda コードスキャンをアクティブ化すると、Amazon Inspector はアカウント内の Lambda 関数とレイヤーを検出し、アプリケーションパッケージの依存関係にあるコードの脆弱性をスキャンします。Lambda コードスキャンは、Lambda 関数内のカスタムアプリケーションコードをスキャンして、コードの脆弱性がないか調べます。これら 2 つのスキャンタイプは同時にアクティブ化する必要があります。詳細については、Amazon Inspector Lambda コードスキャンを参照してください。

スキャンタイプをアクティブ化する

新しい Amazon Inspector スキャンタイプはいつでもアクティブ化できます。スキャンタイプをアクティブ化すると、Amazon Inspector はそのスキャンタイプの対象となるリソースのスキャンを直ちに開始します。使用可能なスキャンタイプの概要については、「Amazon Inspector のスキャンタイプの概要」を参照してください。各スキャンタイプのを初めてアクティブ化した場合の動作について説明します。

- Amazon EC2 スキャン — アカウントの Amazon Inspector Amazon EC2 スキャンをアクティブ化すると、Amazon Inspector はアカウント内のすべての対象インスタンスをスキャンして、パッケージの脆弱性とネットワーク到達可能性の問題がないか調べます。Amazon Inspector SSM プラグインは、すべての SSM マネージドWindowsホストにインストールされます。詳細については、「Windows インスタンスのスキャン」を参照してください。さらに、Amazon Inspector はアカウントに次の SSM 関連付けを作成します。
  - `InspectorDistributor-do-not-delete`
  - `InspectorInventoryCollection-do-not-delete`
  - `InspectorLinuxDistributor-do-not-delete`
  - `InvokeInspectorLinuxSsmPlugin-do-not-delete`
  - `InvokeInspectorSsmPlugin-do-not-delete`.
- Amazon ECR スキャン — アカウントの Amazon ECR コンテナイメージスキャンをアクティブ化すると、そのアカウントのプライベートリポジトリの Amazon ECR スキャンタイプ Amazon ECR による基本的なスキャンから Amazon Inspector による拡張スキャンに変わります。その後、過去 30 日以内にプッシュされた、または過去 90 日以内にプルされた、対象となるすべての Amazon ECR コンテナイメージが、パッケージの脆弱性についてスキャンされます。さらに、Amazon ECR の再スキャン期間は、イメージのプッシュ日とプル日で 90 日に設定されます。

- Lambda 標準スキャン — アカウントで Lambda 標準スキャンをアクティブ化すると、過去 90 日間に呼び出されたか更新されたアカウント内のすべての Lambda 関数がスキャンされ、パッケージの脆弱性が検出されます。さらに、 CloudTrailサービスにリンクされたチャネルがアカウントに作成されます。
- Lambda 標準スキャン + Lambda コードスキャン — これらの Lambda 関数スキャンタイプは同時にアクティブ化されます。アカウントで Lambda コードスキャンをアクティブ化すると、過去 90 日間に呼び出されたか更新されたアカウント内のすべての Lambda 関数がスキャンされ、コードの脆弱性が検出されます。

スキャンのアクティブ化

AWS 組織の Amazon Inspector の委任管理者である場合は、 で Amazon Inspector Amazon Inspectorによって開発されたシェルスクリプトを使用して、複数のリージョンの複数のアカウントのさまざまな Amazon Inspector スキャンタイプを自動的に有効にできます GitHub。 enablement-with-cliそれ以外の場合、コンソールからマルチアカウント環境でこの手順を完了するには、Amazon Inspector の委任された管理者としてサインインして次のステップを実行します。

Console

スキャンをアクティブ化するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、新しいスキャンタイプをアクティブ化するリージョンを選択します。
3. ナビゲーションペインで、[アカウント管理] を選択します。
4. [アカウント管理] ページで、スキャンタイプをアクティブ化するアカウントを選択します。
5. [アクティブ化] を選択し、アクティブ化するスキャンタイプを選択します。
6. （推奨) スキャンタイプをアクティブ化する各 AWS リージョン で、これらのステップを繰り返します。

API

有効化 APL オペレーションを実行します。リクエストには、スキャンをアクティブ化するアカウント ID、冪等性トークン、およびそのタイプのスキャンをアクティブ化するための、EC2、ECR、LAMBDA、または `resourceTypes` の `LAMBDA_CODE` を 1 つ以上指定します。

Amazon Inspector による Amazon EC2 インスタンスのスキャン

Amazon Inspector の Amazon EC2 向けエージェントレススキャンはプレビューリリース中です。Amazon EC2 エージェントレススキャン機能の使用には、「AWS のサービス条件」の第 2 項 (「ベータ版とプレビュー」) が適用されます。

Amazon Inspector EC2 スキャンは、EC2 インスタンスからメタデータを抽出し、このメタデータをセキュリティアドバイザリから収集されたルールと比較して、検出結果を生成します。Amazon Inspector はインスタンスをスキャンして、パッケージの脆弱性とネットワークの到達性の問題がないか調べます。これらの問題について生成される結果のタイプについては、「Amazon Inspector での 検出結果タイプ」を参照してください。

Amazon Inspector は、24 時間ごとにネットワーク到達可能性のスキャンを実行しますが、パッケージの脆弱性スキャンに関してはインスタンスに関連付けられたスキャン方式に応じてさまざまな頻度で実行されます。

スキャン方式

パッケージの脆弱性スキャンは、エージェントベースまたはエージェントレスのスキャン方式を使用して実行できます。これらのスキャン方式により、Amazon Inspector がパッケージの脆弱性スキャンのために EC2 インスタンスからソフトウェアインベントリを収集する方法と時期が決まります。エージェントベース方式では SSM エージェントに依存してソフトウェアインベントリを収集しますが、エージェントレス方式ではエージェントの代わりに Amazon EBS スナップショットを使用します。

Amazon Inspector で使用されるスキャン方式は、アカウントの [スキャンモード] 設定によって異なります。詳細については、「スキャンモードの管理」を参照してください。

Amazon EC2 スキャンをアクティブ化するには、「スキャンタイプをアクティブ化する」を参照してください。

エージェントベースのスキャン

エージェントベースのスキャンは、対象となるすべてのインスタンスで SSM エージェントを使用して継続的に実行されます。エージェントベースのスキャンの場合、Amazon Inspector は SSM 関連付けと、これらの関連付けを通じてインストールされたプラグインを使用して、インスタンスからソフトウェアインベントリを収集します。Amazon Inspector のエージェントベースのスキャンでは、

オペレーティングシステムパッケージに対するパッケージの脆弱性スキャンに加えて、Linux ベースのインスタンス内のアプリケーションプログラミング言語パッケージに対するパッケージの脆弱性も Amazon EC2 Linux インスタンス向け Amazon Inspector 詳細検査 によって検出できます。

以下のプロセスでは、Amazon Inspector が SSM を使用してインベントリを収集し、エージェントベースのスキャンを実行する方法について説明します。

1. Amazon Inspector は、インスタンスからインベントリを収集するために、アカウントに SSM 関連付けを作成します。一部のインスタンスタイプ (Windows および Linux) では、これらの関連付けによって、インベントリが収集するために個々のインスタンスにプラグインがインストールされます。
2. Amazon Inspector は SSM を使用してインスタンスからパケージインベントリを抽出します。
3. Amazon Inspector は抽出されたインベントリを評価し、検出された脆弱性について検出結果を生成します。

対象インスタンス

Amazon Inspector は、インスタンスが以下の条件を満たす場合、エージェントベース方式でスキャンします。

- サポートされている OS がインスタンスに備わっている。サポートされている OS のリストについては、「the section called “Amazon EC2 のスキャンでサポートされているオペレーティングシステム”」の「エージェントベースのスキャンのサポート」コラムを参照してください。
- インスタンスが Amazon Inspector の EC2 除外タグによってスキャンから除外されていない。
- SSM マネージドインスタンスである。エージェントを確認および設定する手順については、「SSM Agent の設定」を参照してください。

エージェントベーススキャンの動作

エージェントベースのスキャン方式を使用する場合、Amazon Inspector は、以下の状況において EC2 インスタンスの脆弱性スキャンを新たに開始します。

- 新しい EC2 インスタンスを起動するとき。
- 既存の EC2 インスタンス (Linux および Mac) に新しいソフトウェアをインストールするとき。
- Amazon Inspector がデータベースに新しい共通脆弱性識別子 (CVE) 項目を追加し、その CVE が EC2 インスタンス (Linux と Mac) に関連する場合。

Amazon Inspector は、初回スキャンが完了すると EC2 インスタンスの [最終スキャン日] フィールドを更新します。この後、Amazon Inspector が SSM インベントリを評価したとき (デフォルトでは 30 分ごと)、またはインスタンスに影響を与える CVE が Amazon Inspector データベースに新たに追加されたためにインスタンスが再スキャンされたときに、[最終スキャン日] フィールドが更新されます。

EC2 インスタンスの脆弱性の最終スキャン日は、[アカウント管理] ページの [インスタンス] タブから、または ListCoverage コマンドを使用して確認できます。

SSM Agent の設定

Amazon Inspector がエージェントベースのスキャン方式を使用して Amazon EC2 インスタンスのソフトウェア脆弱性を検出するには、そのインスタンスが Amazon EC2 Systems Manager (SSM) のマネージドインスタンスである必要があります。SSM マネージドインスタンスには SSM Agent がインストールされ実行されており、SSM にはインスタンスを管理するアクセス許可があります。すでに SSM を使用してインスタンスを管理している場合、エージェントベースのスキャンのために必要な他の手順はありません。

SSM Agent は、一部の Amazon マシンイメージ (AMI) から作成された EC2 インスタンスにデフォルトでインストールされます。詳細については、「AWS Systems Manager ユーザーガイド」の「SSM Agent について」を参照してください。ただし、SSM Agent がインストールされている場合でも、それを手動でアクティブ化し、SSM にインスタンスを管理するためのアクセス許可の付与が必要の場合があります。

以下の手順では、IAM インスタンスプロファイルを使用して Amazon EC2 インスタンスをマネージドインスタンスとして設定する方法について説明します。この手順では、「AWS Systems Manager ユーザーガイド」にあるより詳細な情報のリンクも提供しています。

AmazonSSMManagedInstanceCore は、インスタンスプロファイルをアタッチするときに使用する推奨ポリシーです。このポリシーには、Amazon Inspector EC2 スキャンに必要なすべてのアクセス許可が含まれています。

> **Note**
>
> SSM デフォルトのホスト管理設定を使用すると、IAM インスタンスプロファイルを使用しなくても、すべての EC2 インスタンスの SSM 管理を自動化できます。詳細については、「デフォルトのホスト管理設定」を参照してください。

Amazon EC2 インスタンス用に SSM を設定する方法

1. オペレーティングシステムベンダーがまだ SSM Agent をインストールしていない場合は、SSM Agent をインストールします。詳細については、「SSM Agent の使用」を参照してください。
2. を使用して AWS CLI 、SSM エージェントが実行中であることを確認します。詳細については、「SSM Agent ステータスの確認とエージェントの起動」を参照してください。
3. SSM にインスタンスを管理するアクセス許可を付与します。IAM インスタンスプロファイルを作成してインスタンスにアタッチすることで、アクセス許可を付与できます。AmazonSSMManagedInstanceCore ポリシーを使用することをお勧めします。このポリシーには、Amazon Inspector がスキャンに必要な SSM ディストリビューター、SSM インベントリ、および SSM ステートマネージャーのアクセス許可があるためです。これらの許可を持つインスタンスプロファイルを作成してインスタンスにアタッチする手順については、「Configure instance permissions for Systems Manager」を参照してください。
4. (オプション) SSM Agent の自動アップデートをアクティブ化にします。詳細については、「SSM Agent への更新の自動化」を参照してください。
5. (オプション) Amazon Virtual Private Cloud (Amazon VPC) エンドポイントを使用するように Systems Manager を設定します。詳細については、「Create Amazon VPC endpoints」を参照してください。

> **Important**
>
> Amazon Inspector では、ソフトウェアアプリケーションインベントリを収集するために、お客様のアカウントに Systems Manager ステートマネージャーの関連付けが必要です。まだ存在しない場合は、Amazon Inspector によって `InspectorInventoryCollection-do-not-delete` という名前の関連付けが自動的に作成されます。
> Amazon Inspector ではリソースデータの同期も必要で、まだ存在しない場合は自動的に `InspectorResourceDataSync-do-not-delete` という名前で作成されます。詳細については、「AWS Systems Manager ユーザーガイド」の「インベントリのリソースデータの同期の設定」を参照してください。各アカウントは、リージョンごとに設定された数のリソースデータを同期することができます。詳細については、「SSM エンドポイントのリソースデータ同期の最大数 (リージョン AWS アカウント ごと) とクォータ」を参照してください。 https://docs.aws.amazon.com/general/latest/gr/ssm.html#limits_ssmこの最大数に達した場合は、リソースデータ同期を削除する必要があります。「Managing resource data syncs」を参照してください。

スキャン用に作成された SSM リソース

Amazon Inspector では、Amazon EC2 スキャンを実行するためにアカウントに多数の SSM リソースが必要です。Amazon Inspector EC2 スキャンを初めてアクティブ化すると、以下のリソースが作成されます。

> **Note**
>
> アカウントで Amazon Inspector Amazon EC2 スキャンが有効になっている間にこれらの SSM リソースのいずれかが削除された場合、Amazon Inspector は次のスキャン間隔でリソースの再作成を試みます。

`InspectorInventoryCollection-do-not-delete`

これは Amazon Inspector が Amazon EC2 インスタンスからソフトウェアアプリケーションインベントリを収集するために使用するSystems Manager ステートマネージャー (SSM) 関連付けです。`InstanceIds*` からインベントリを収集するための SSM 関連付けがすでにアカウントに存在する場合、Amazon Inspector は独自のものを作成する代わりにその関連付けを使用します。

`InspectorResourceDataSync-do-not-delete`

これは Amazon Inspector が、Amazon EC2 インスタンスから収集したインベントリデータを所有する Amazon S3 バケットに送信するために使用するリソースデータ同期です。詳細については、「AWS Systems Manager ユーザーガイド」の「インベントリのリソースデータの同期の設定」を参照してください。

`InspectorDistributor-do-not-delete`

これは Amazon Inspector が Windows インスタンスのスキャンに使用する SSM 関連付けです。この関連付けにより、Windows インスタンスに Amazon Inspector SSM プラグインがインストールされます。プラグインファイルが誤って削除された場合、この関連付けは次の関連付け間隔でプラグインを再インストールします。

`InvokeInspectorSsmPlugin-do-not-delete`

これは Amazon Inspector が Windows インスタンスのスキャンに使用する SSM 関連付けです。この関連付けにより、Amazon Inspector はプラグインを使用してスキャンを開始できます。また、これを使用して Windows インスタンスのスキャンのカスタム間隔を設定することもできます。詳細については、「Windows インスタンススキャンのカスタムスケジュールの設定」を参照してください。

`InspectorLinuxDistributor-do-not-delete`

これは、Amazon Inspector が Amazon EC2 Linux の詳細検査に使用する SSM 関連付けです。Amazon EC2 この関連付けにより、Amazon Inspector SSM プラグインが Linux インスタンスにインストールされます。

`InvokeInspectorLinuxSsmPlugin-do-not-delete`

これは、Amazon Inspector が Amazon EC2 Linux の詳細検査に使用する SSM 関連付けです。Amazon EC2 この関連付けにより、Amazon Inspector はプラグインを使用してスキャンを開始できます。

> **Note**
>
> Amazon Inspector Amazon EC2 スキャンまたは詳細検査を無効にすると、対応する Linux ホストからすべての SSM リソースが自動的にアンインストールされます。

エージェントレススキャン

アカウントがハイブリッドスキャンモード (エージェントベースのスキャンとエージェントレススキャンの両方を含む) の場合、Amazon Inspector は対象となるインスタンスに対してエージェントレススキャン方式を使用します。エージェントレススキャンの場合、Amazon Inspector は EBS スナップショットを使用してインスタンスからソフトウェアインベントリを収集します。エージェントレス方式を使用してスキャンされたインスタンスは、オペレーティングシステムパッケージとアプリケーションプログラミング言語パッケージの両方の脆弱性をスキャンされます。

> **Note**
>
> Linux インスタンスのアプリケーションプログラミング言語パッケージの脆弱性をスキャンする場合、エージェントレス方式では使用可能なすべてのパスがスキャンされます。一方、エージェントベーススキャンでは、デフォルトパスと、Amazon EC2 Linux インスタンス向け Amazon Inspector 詳細検査 の一部として指定した追加パスのみがスキャンされます。これにより、同じインスタンスでも、エージェントベース方式を使用してスキャンされたか、エージェントレス方式を使用してスキャンされたかによって、異なる検出結果が得られる可能性があります。

以下のプロセスでは、Amazon Inspector が EBS スナップショットを使用してインベントリを収集し、エージェントレススキャンを実行する方法について説明します。

1. Amazon Inspector は、インスタンスにアタッチされたすべてのボリュームの EBS スナップショットを作成します。Amazon Inspector が使用している間、スナップショットはアカウントに保存され、タグキーとして `InspectorScan`、タグ値として一意のスキャン ID でタグ付けされます。
2. Amazon Inspector は、EBS ダイレクト API を使用してスナップショットからデータを取得し、脆弱性がないか評価します。検出された脆弱性に対して検出結果が生成されます。
3. Amazon Inspector は、アカウント内に作成した EBS スナップショットを削除します。

対象インスタンス

Amazon Inspector は、インスタンスが以下の条件を満たす場合、エージェントレス方式でスキャンします。

- サポートされている OS がインスタンスに備わっている。サポートされている OS のリストについては、「the section called “Amazon EC2 のスキャンでサポートされているオペレーティングシステム”」の「エージェントベースのスキャンのサポート」コラムを参照してください。
- インスタンスが Amazon Inspector の EC2 除外タグによってスキャンから除外されていない。
- インスタンスのステータスは、`Unmanaged EC2 instance`、`Stale inventory`、または `No inventory`です。
- インスタンスが EBS-backed で、以下のいずれかのファイルシステム形式である。
  - `ext3`
  - `ext4`
  - `xfs`

エージェントレススキャンの動作

アカウントが [ハイブリッドスキャン] 用に設定されている場合、Amazon Inspector は 24 時間ごとに対象インスタンスに対してエージェントレススキャンを実行します。Amazon Inspector は、新たに対象となるインスタンスを 1 時間ごとに検出してスキャンします。これには、SSM エージェントのない新しいインスタンス、またはステータスが `SSM_UNMANAGED` に変わった既存のインスタンスが含まれます。

Amazon Inspector は、エージェントレススキャンの後にインスタンスから抽出されたスナップショットをスキャンするたびに、Amazon EC2 インスタンスの [最終スキャン日] フィールドを更新します。

EC2 インスタンスの脆弱性の最終スキャン日は、[アカウント管理] ページの [インスタンス] タブから、または ListCoverage コマンドを使用して確認できます。

スキャンモードの管理

アカウントで EC2 スキャンを実行するときに Amazon Inspector がどのスキャン方式を使用するかは、EC2 スキャンモードで決まります。アカウントのスキャンモードは、[全般設定] の [EC2 スキャン設定] ページで確認できます。スタンドアロンアカウントまたは Amazon Inspector の委任管理者は、スキャンモードを変更できます。Amazon Inspector の委任管理者としてスキャンモードを設定すると、そのスキャンモードが組織内のすべてのメンバーアカウントに設定されます。Amazon Inspector には以下のスキャンモードがあります。

エージェントベースのスキャン — このスキャンモードでは、Amazon Inspector はパッケージの脆弱性をスキャンする際にエージェントベースのスキャン方式のみを使用します。このスキャンモードは、アカウント内の SSM マネージドインスタンスのみをスキャンしますが、新しい CVE やインスタンスへの変更に応じて継続的にスキャンできるという利点があります。エージェントベースのスキャンでは、対象となるインスタンスに対して Amazon Inspector の詳細検査を行うこともできます。これは、新しくアクティブ化されたアカウントではデフォルトのスキャンモードです。

ハイブリッドスキャン — このスキャンモードでは、Amazon Inspector はエージェントベースの方式とエージェントレス方式の両方を組み合わせてパッケージの脆弱性をスキャンします。SSM エージェントがインストールおよび設定されている適格な EC2 インスタンスでは、Amazon Inspector はエージェントベースの方式を使用します。SSM マネージドではない対象インスタンスの場合、Amazon Inspector は対象の EBS-backed インスタンスに対してエージェントレス方式を使用します。

スキャンモードを変更するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、EC2 スキャンモードを変更するリージョンを選択します。
3. サイドナビゲーションパネルの [全般設定] から、[EC2 スキャン設定] を選択します。
4. [スキャンモード] で [編集] を選択します。

5. スキャンモードを選択し、[変更を保存] を選択します。

Amazon Inspector スキャンからのインスタンスの除外

特定のインスタンスにタグを付けて、Amazon Inspector のスキャンからインスタンスを除外できます。インスタンスをスキャンから除外すると、実行不可能なアラートを防ぐことができます。除外されたインスタンスには課金されません。

EC2 インスタンスをスキャンから除外するには、そのインスタンスに次のキーをタグ付けします。

- `InspectorEc2Exclusion`

値はオプションです。

タグの追加の詳細については、「Amazon EC2 リソースのタグ付け」を参照してください。

さらに、暗号化された EBS ボリュームをエージェントレススキャンから除外するには、そのボリュームの暗号化に使用される AWS KMS キーに `InspectorEc2Exclusion` タグを付けます。詳細については、「キーのタグ付け」を参照してください。

サポートされるオペレーティングシステム

Amazon Inspector は、サポートされている Mac、Windows、Linux の EC2 インスタンスをスキャンして、オペレーティングシステムパッケージの脆弱性を検出します。Linux インスタンスの場合、Amazon Inspector では Amazon EC2 Linux インスタンス向け Amazon Inspector 詳細検査 を使用してアプリケーションプログラミング言語パッケージの結果を生成できます。Mac および Windows インスタンスの場合、オペレーティングシステムパッケージのみがスキャンされます。

SSM エージェントなしでスキャンできるオペレーティングシステムなど、サポートされているオペレーティングシステムについては、「Amazon EC2 のスキャンでサポートされているオペレーティングシステム」を参照してください。

Amazon EC2 Linux インスタンス向け Amazon Inspector 詳細検査

Amazon Inspector はAmazon EC2 スキャンカバレッジを拡張して詳細検査を含めます。詳細検査により、Amazon Inspector は Linux ベースの Amazon EC2 インスタンス内のアプリケーションプログラミング言語パッケージのパッケージ脆弱性を検出します。

Amazon Inspector は、プログラミング言語パッケージライブラリのデフォルトパスをスキャンします。デフォルトパスに加えてカスタムパスを設定することもできます。詳細については、「Amazon Inspector の詳細検査のカスタムパス」を参照してください。

Amazon Inspector は、Amazon Inspector SSM プラグインで収集されたデータを使用して詳細検査スキャンを実行します。プラグインを管理し、Linux の詳細な検査を実行するために、Amazon Inspector はアカウントに次の SSM 関連付けを自動的に作成`InvokeInspectorLinuxSsmPlugin-do-not-delete`します。これは、Amazon Inspector が詳細検査をアクティブ化したときに発生します。

Amazon Inspector は、6 時間ごとに詳細検査のためにインスタンスから更新されたアプリケーションインベントリを収集します。

Amazon Inspector が詳細検査でサポートするプログラミング言語のリストについては、「サポートされているプログラミング言語: Amazon EC2 の詳細検査」を参照してください。

> **Note**
>
> Windows または Mac インスタンスでは Deep inspection はサポートされていません。

詳細検査のアクティブ化または非アクティブ化

> **Note**
>
> Deep inspection は、2023 年 4 月 17 日以降、Amazon Inspector をアクティベートしたアカウントでは、Amazon EC2 スキャンの一環として自動的にアクティブ化されます。

アカウントに対して詳細検査がアクティブ化になっているかどうかは、Amazon Inspector コンソールの [アカウント管理] ページの [Amazon EC2 スキャン] 列で確認できます。詳細検査がアクティブになっていない場合、この列には「アクティブ化済み (詳細検査が非アクティブ化されました)」と表示されます。アクティベーションのステータスをプログラムで確認するには、GetEc2DeepInspectionConfiguration を使用します。複数のアカウントの場合は、BatchGetMemberEc2DeepInspectionStatus API を使用してください。

2023 年 4 月 17 日より前に Amazon Inspector をアクティベートした場合は、コンソールバナーまたは UpdateEc2DeepInspectionConfiguration API を使用して詳細検査を

アクティベートできます。組織において Amazon Inspector の委任管理者である場合は、BatchUpdateMemberEc2DeepInspectionStatus API を使用して、自分とメンバーアカウントに対して詳細検査をアクティブ化できます。

UpdateEc2DeepInspectionConfiguration API を使用して詳細検査を非アクティブ化できます。組織のメンバーアカウントは詳細検査を非アクティブ化することはできません。その代わりに、委任管理者が BatchUpdateMemberEc2DeepInspectionStatus API を使用してメンバーアカウントを非アクティブ化する必要があります。

Linux 向け Amazon Inspector SSM プラグインについて

Amazon Inspector は Amazon Inspector SSM プラグインを使用して Linux インスタンスの詳細検査を実行します。Amazon Inspector SSM プラグインは、Linux インスタンスの次のディレクトリに自動的にインストールされます:`/opt/aws/inspector/bin` 実行可能ファイルの名前は `inspectorssmplugin` です。

> **Note**
>
> Amazon Inspector は、Systems Manager Distributor を使用して Amazon EC2 インスタンスにプラグインをデプロイします。Systems Manager Distributor は、「Systems Manager ガイド」の「サポートされているパッケージのプラットフォームとアーキテクチャ」に記載されているオペレーティングシステムをサポートします。Amazon Inspector が詳細検査スキャンを実行するには、Amazon EC2 インスタンスのオペレーティングシステムが Systems Manager Distributor と Amazon Inspector によってサポートされている必要があります。

Amazon Inspector は、Amazon Inspector SSM プラグインによって詳細検査向けに収集されたデータを管理するために、以下のファイルディレクトリを作成します。

- `/opt/aws/inspector/var/input`
- `/opt/aws/inspector/var/output`
  - このディレクトリの `packages.txt` は、詳細検査によって検出されたパッケージへのフルパスを保存します。Amazon Inspector がインスタンスで同じパッケージを複数回検出した場合、このファイルにはそのパッケージが見つかった各場所が一覧表示されます。

Amazon Inspector は、プラグインのログを `/var/log/amazon/inspector` ディレクトリに保存します。

Amazon Inspector SSM プラグインのアンインストール

`inspectorssmplugin` ファイルが誤って削除された場合、`InspectorLinuxDistributor-do-not-delete` SSM 関連付けは次のスキャン間隔でプラグインの再インストールを試みます。

Amazon EC2 スキャンを無効にすると、プラグインはすべての Linux ホストから自動的にアンインストールされます。

Amazon Inspector の詳細検査のカスタムパス

Linux Amazon EC2 インスタンスの詳細な検査を実行するときに検索するように Amazon Inspector のカスタムパスを設定できます。 Amazon EC2 カスタムパスを追加すると、Amazon Inspector はそのディレクトリとその中のすべてのサブディレクトリにあるパッケージをスキャンします。

すべてのアカウントは、個々のアカウントに最大 5 つのカスタムパスを定義できます。組織の委任された管理者の場合は、組織全体に適用される 5 つの追加パスを定義できます。これにより、組織内のアカウントごとにスキャンされるカスタムパスは合計で最大 10 個になります。

Amazon Inspector は、すべてのアカウントでスキャンされる以下のデフォルトパスに加えて、すべてのカスタムパスをスキャンします。

- `/usr/lib`
- `/usr/lib64`
- `/usr/local/lib`
- `/usr/local/lib64`

> **Note**
>
> カスタムパスはローカルパスである必要があります。Amazon Inspector は、ネットワークファイルシステム (NFS) マウントや Amazon S3 ファイルシステムマウントなどのマッピングされたネットワークパスをスキャンしません。

カスタムパスのフォーマット

カスタムパスの形式の例を次に示します。`/home/usr1/project01`

カスタムパスは 256 文字未満にする必要があります。

1 つのインスタンスにつき 5,000 個のパッケージ制限があり、パッケージインベントリ収集の最大時間制限は 15 分です。これらの制限を避けるため、カスタムパスを選択することをおすすめします。

コンソールでカスタムパスを設定してください。

Console

Amazon Inspector の委任された管理者としてサインインし、以下の手順に従って組織のカスタムパスを追加します。

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、Lambda 標準スキャンをアクティブ化するリージョンを選択します。
3. サイドナビゲーションパネルの [全般設定] から、[EC2 スキャン設定] を選択します。
4. [独自のアカウントのカスタムパス] で [編集] を選択し、個人アカウントのパスを追加します。委任された管理者の場合は、[組織のカスタムパス] パネルで [編集] を選択して、組織内のすべてのアカウントにカスタムパスを追加できます。
5. テキストボックスにカスタムパスを入力します。
6. [保存] を選択してカスタムパスを保存します。Amazon Inspector は、次の詳細検査にこれらのパスを含めます。

API

UpdateEc2DeepInspectionConfiguration コマンドを実行します。`packagePaths` には、スキャンするパスの配列を指定します。

サポートされているプログラミング言語

Linux インスタンスの場合、Amazon Inspector の詳細検査により、オペレーティングシステムパッケージの脆弱性に加えて、アプリケーションプログラミング言語パッケージの検出結果も生成できます。Mac および Windows インスタンスの場合、オペレーティングシステムパッケージのみがスキャンされます。

サポートされているプログラミング言語については、「Amazon Inspector の詳細検査でサポートされているプログラミング言語」を参照してください。

Amazon Inspector による Windows EC2 インスタンスのスキャン

> **Note**
>
> 2022 年 8 月 31 日、Amazon Inspector は Amazon EC2 スキャンの対象範囲を拡大し、Windows を実行している EC2 インスタンスを対象としました。

Amazon Inspector は、サポートされているすべて Windows インスタンスを自動的に検出し、追加のアクションなしで継続的スキャンにそれらを含めます。サポート対象のインスタンスの詳細については、「Amazon EC2 のスキャンでサポートされているオペレーティングシステム」を参照してください。

Linux ベースのインスタンスのスキャンとは異なり、Amazon Inspector は定期的に Windows スキャンを実行します。Windows インスタンスは初回検出時にスキャンされ、その後 6 時間ごとにスキャンされます。ただし、デフォルトの 6 時間のスキャン間隔は調整可能です。詳細については、「Windows インスタンススキャンのカスタムスケジュールの設定」を参照してください。Amazon Inspector が Windows インスタンスをスキャンする方法について概要は次のとおりです。

1. Amazon EC2 スキャンがアクティブ化になると、Amazon Inspector は Windows リソースに対して新しい SSM 関連付け `InspectorDistributor-do-not-delete`、`InspectorInventoryCollection-do-not-delete` および `InvokeInspectorSsmPlugin-do-not-delete` を作成します。
2. `InspectorDistributor-do-not-delete` SSM 関連付けは`AWS-ConfigureAWSPackage`、SSM ドキュメントと `AmazonInspector2-InspectorSsmPlugin` SSM Distributor パッケージを使用して、Windowsインスタンスに Amazon Inspector SSM プラグインをインストールします。詳細については、「の Amazon Inspector SSM プラグインについて Windows」を参照してください。
3. `InvokeInspectorSsmPlugin-do-not-delete` SSM 関連付けは、Amazon Inspector SSM プラグインを定期的に実行してインスタンスデータを収集し、Amazon Inspector の検出結果を生成します。デフォルトでは、この間隔は 6 時間ごとです。ただし、SSM を使用して関連付けに cron 式または rate 式を設定することで、これをカスタマイズできます。詳細については、「AWS Systems Manager ユーザーガイド」の「リファレンス: Systems Manager の Cron 式および rate 式」を参照してください。

> **Note**
>
> Amazon Inspector は、更新されたオープン脆弱性評価言語 (OVAL) 定義ファイルを S3 バケット `inspector2-oval-prod-REGION` にステージングします。この S3 バケットにはスキャンに使用された OVAL 定義が含まれており、変更されるべきではありません。この設定が変更されると、Amazon Inspector はリリース時に新しい CVE をスキャンできなくなります。

Windows インスタンの Amazon Inspector スキャン要件

Windows インスタンスをスキャンするには、Amazon Inspector ではインスタンスが次の基準を満たす必要があります。

- インスタンスは SSM マネージドインスタンスです。インスタンスをスキャン用に設定する手順については、「SSM Agent の設定」を参照してください。
- インスタンスオペレーティングシステムは、サポートされている Windows オペレーティングシステムの 1 つです。サポートされているオペレーティングシステムの完全なリストについては、「Amazon EC2 のスキャンでサポートされているオペレーティングシステム」を参照してください。
- インスタンスには Amazon Inspector SSM プラグインがインストールされています。Amazon Inspector は、検出時にマネージドインスタンス用の Amazon Inspector SSM プラグインを自動的にインストールします。プラグインの詳細については、次のトピックを参照してください。

> **Note**
>
> ホストがインターネットにアクセスできない Amazon VPC で動作している場合、Windows スキャンでは、ホストがリージョンの Amazon S3 エンドポイントにアクセスできる必要があります。Amazon S3 Amazon VPC エンドポイントの設定方法については、「Amazon Virtual Private Cloud User Guide」の「Create a gateway endpoint」を参照してください。Amazon VPC エンドポイントポリシーが外部 S3 バケットへのアクセスを制限している場合は、インスタンスの評価に使用される OVAL 定義 AWS リージョン を保存する で Amazon Inspector が管理するバケットへのアクセスを特に許可する必要があります。このバケットは次の形式になります。`inspector2-oval-prod-REGION`

の Amazon Inspector SSM プラグインについて Windows

Amazon Inspector がWindowsインスタンスをスキャンするには、Amazon Inspector SSM プラグインが必要です。Amazon Inspector SSM プラグインは のWindowsインスタンスに自動的にインストールされ`C:\Program Files\Amazon\Inspector`、実行可能バイナリファイルの名前は ですInspectorSsmPlugin.exe。

Amazon Inspector SSM プラグインが収集するデータを保存するために、次のファイルの場所が作成されます。

- `C:\ProgramData\Amazon\Inspector\Input`
- `C:\ProgramData\Amazon\Inspector\Output`
- `C:\ProgramData\Amazon\Inspector\Logs`

> **Note**
>
> デフォルトでは、Amazon Inspector SSM プラグインは通常の優先度未満で実行されます。

Amazon Inspector SSM プラグインのアンインストール

`InspectorSsmPlugin.exe` ファイルが誤って削除された場合、`InspectorDistributor-do-not-delete` SSM 関連付けは次の Windows スキャン間隔でプラグインを再インストールします。Amazon Inspector SSM プラグインをアンインストールする場合は、`AmazonInspector2-ConfigureInspectorSsmPlugin`ドキュメントで Uninstall アクションを使用できます。

さらに、Amazon Inspector SSM プラグインはすべてのWindowsホストから自動的にアンインストールされます。 Amazon EC2

> **Note**
>
> Amazon Inspector を非アクティブ化する前に SSM エージェントをアンインストールした場合、Amazon Inspector SSM プラグインはWindowsホストに残りますが、Amazon Inspector SSM プラグインにデータを送信しなくなります。詳細については、「Amazon Inspector の非アクティブ化」を参照してください。

Windows インススキャンのカスタムスケジュールの設定

SSM を使用して Windows 関連付けの cron 式または rate 式を設定すること
で、InvokeInspectorSsmPlugin-do-not-delete Amazon EC2 インススキャンの間隔
をカスタマイズできます。詳細については、「AWS Systems Manager ユーザーガイド」の「リファレンス: Systems Manager の Cron 式および rate 式」を参照するか、次の手順を使用してください。

次のコード例から選択し、rate 式または cron 式を使用して Windows インスタンスのスキャン間隔をデフォルトの 6 時間から 12 時間に変更します。

次の例では、 という名前の関連付けAssociationIdに を使用する必要がありますInvokeInspectorSsmPlugin-do-not-delete。次のコマンドAssociationIdを実行して、AWS CLI を取得できます。

```
$ aws ssm list-associations --association-filter-list
 "key=AssociationName,value=InvokeInspectorSsmPlugin-do-not-delete" --region us-east-1
```

> **Note**
>
> AssociationId はリージョナルであるため、まず各 の一意の ID を取得する必要がありますAWS リージョン。その後、コマンドを実行して、Windows インスタンスのカスタムスキャンスケジュールを設定したい各リージョンのスキャン頻度を変更できます。

Example rate expression

```
$ aws ssm update-association \
--association-id "YourAssociationId" \
--association-name "InvokeInspectorSsmPlugin-do-not-delete" \
--schedule-expression "rate(12 hours)"
```

Example cron expression

```
$ aws ssm update-association \
--association-id "YourAssociationId" \
--association-name "InvokeInspectorSsmPlugin-do-not-delete" \
--schedule-expression "cron(0 0/12 * * ? *)"
```

Amazon Inspector による Amazon ECR コンテナイメージのスキャン

Amazon Inspector は、Amazon ECR に保存されているコンテナイメージをスキャンしてソフトウェアの脆弱性がないかを調べ、パッケージ 脆弱性の検出結果を生成します。これらの問題について生成される結果のタイプについては、「Amazon Inspector での 検出結果タイプ」を参照してください。

Amazon ECR の Amazon Inspector スキャンをアクティブ化すると、Amazon Inspector をプライベートレジストリの優先スキャンサービスとして設定します。これにより、Amazon ECR が無料で提供するデフォルトの基本的なスキャンが、Amazon Inspector を通じて提供および請求される拡張スキャンに置き換わります。

Amazon Inspector が提供する拡張スキャンでは、オペレーティングシステムパッケージとプログラミング言語パッケージの両方をレジストリレベルで脆弱性スキャンできるという利点があります。拡張スキャンを使用して検出された検出結果は、Amazon ECR コンソールで、イメージのレイヤーごとにイメージレベルで確認できます。さらに、 や AWS Security Hub Amazon など、基本的なスキャン検出結果には利用できない他の のサービスで、これらの検出結果を確認して操作できます EventBridge。スキャンによって検出された検出結果は、https://console.aws.amazon.com/inspector/v2/home の Amazon Inspector コンソールで表示できます。検出結果の操作の詳細については、「Amazon Inspector での検出結果の管理」を参照してください。

Amazon ECR スキャンをアクティブ化する手順については、「スキャンタイプをアクティブ化する」を参照してください。

Amazon ECR スキャンのスキャン動作

ECR スキャンを初めてアクティブ化し、リポジトリが連続スキャン用に設定されている場合、Amazon Inspector は 30 日以内にプッシュした、または過去 90 日以内にプルしたすべての対象イメージを検出します。次に、Amazon Inspector は検出されたイメージをスキャンし、スキャンステータスを に設定しますactive。Amazon Inspector は、過去 90 日間 (デフォルトで) または設定した ECR 再スキャン期間内にプッシュまたはプルされたイメージを引き続きモニタリングします。詳細については、「ECR 再スキャン期間の設定」を参照してください。

継続的スキャンの場合、Amazon Inspector は次のような状況でコンテナイメージの新しい脆弱性スキャンを開始します。

- 新しいコンテナイメージがプッシュされる場合。

- Amazon Inspector がデータベースに新しい共通脆弱性識別子 (CVE) 項目を追加し、その CVE がそのコンテナイメージに関連する場合 (継続的スキャンのみ)。

プッシュスキャン時にリポジトリを設定すると、イメージはプッシュ時にのみスキャンされます。

コンテナイメージの脆弱性の最終確認日は、[アカウント管理] ページの [コンテナイメージ] タブから、または ListCoverage API を使用して確認できます。Amazon Inspector は、以下のイベントに応じて Amazon ECR イメージの [最終スキャン日] フィールドを更新します。

- Amazon Inspector がコンテナイメージの初回スキャンを完了した場合。
- Amazon Inspector がコンテナイメージを再スキャンしたとき。これは、そのコンテナイメージに影響を与える新しい共通脆弱性識別子 (CVE) 項目が Amazon Inspector データベースに追加されたためです。

サポートされているオペレーティングシステムとメディアタイプ

サポートされるオペレーティングシステムの詳細については、「Amazon ECR スキャンでサポートされているオペレーティングシステム」を参照してください。

Amazon ECR リポジトリの Amazon Inspector スキャンは、サポートされている以下のメディアタイプを対象としています。

- `"application/vnd.docker.distribution.manifest.v1+json"`
- `"application/vnd.docker.distribution.manifest.v1+prettyjws"`
- `"application/vnd.oci.image.manifest.v1+json"`
- `"application/vnd.docker.distribution.manifest.v2+json"`

> **Note**
>
> スクラッチイメージと `DockerV2ListMediaType` イメージはサポートされていません。

Amazon ECR リポジトリの拡張スキャンの設定

Amazon ECR コンテナイメージの Amazon Inspector スキャンをアクティブ化すると、プライベートレジストリのスキャン設定が変更されます。レジストリのスキャンタイプが、[基本的なスキャン]

から Amazon Inspector が提供する [拡張スキャン] に変更されました。詳細については、「Amazon ECR ユーザーガイド」の「イメージスキャン」を参照してください。

ECR のリポジトリレベルで拡張スキャンの設定を管理できます。リポジトリの継続的スキャンまたはオンプッシュスキャンを選択できます。継続的スキャンには、オンプッシュスキャンと自動再スキャンが含まれます。オンプッシュスキャンは、最初にイメージをプッシュしたときにのみスキャンされます。どちらのオプションでも、包含フィルターを使用してスキャン範囲を絞り込むことができます。デフォルトでは、拡張スキャンを初めてアクティブ化すると、設定は [すべてのリポジトリを継続的にスキャン] に設定されます。

拡張スキャンの設定を行うには

1. Amazon ECR コンソール (https://console.aws.amazon.com/ecr/) を開きます。
2. ページの右上隅にある AWS リージョン セレクターで、スキャンするリポジトリがあるリージョンを選択します。
3. ナビゲーションペインで、[プライベートレジストリ]、[スキャン] の順に選択します。
4. [スキャンタイプ] で [拡張スキャン] が選択されていることを確認します。選択されていない場合は、[拡張スキャン] を選択します。

   デフォルトでは、[すべてのリポジトリを継続的にスキャン] オプションが選択されています。これにより、すべてのリポジトリの Amazon Inspector スキャン範囲が完全に有効になります。
5. [すべてのリポジトリを継続的にスキャン] を選択解除して、どのリポジトリを継続的にスキャンするか、またはプッシュ時にスキャンするかをフィルタリングします。

拡張スキャンの設定の詳細については、「Amazon ECR ユーザーガイド」の「拡張スキャンの使用」を参照してください。

ECR 再スキャン期間の設定

ECR 再スキャン期間設定は、Amazon Inspector がリポジトリ内のコンテナイメージを継続的にモニタリングする期間を決定します。イメージのプッシュ日とイメージのプル日の再スキャン期間を設定できます。組織に追加された新しいアカウントを含む、新しいアカウントのデフォルトのスキャン期間は 90 日間です。

イメージのプッシュ日の期間

イメージのプッシュ日の期間によって、Amazon Inspector が最新のプル日以降にリポジトリにプッシュされた後にイメージを継続的にモニタリングする期間が決まります。再スキャン期間として以下のオプションを使用できます。

- 14 日間
- 30 日間
- 60 日
- 90 日 (デフォルト)
- 180 日間
- 有効期間

イメージのプル日の期間

イメージのプル日期間はAmazon Inspector が最新のプル日以降にイメージを継続的にモニタリングする期間を決定します。再スキャン期間として以下のオプションを使用できます。

- 14 日間
- 30 日間
- 60 日
- 90 日 (デフォルト)
- 180 日間

Amazon Inspector は、設定されたプッシュおよびプル日内にプッシュまたはプルされている限り、イメージのモニタリングと再スキャンを続行します。設定されたプッシュおよびプル日内にイメージがプッシュまたはプルされていない場合、Amazon Inspector はイメージのモニタリングを停止します。

> **Note**
>
> Amazon Inspector がイメージのモニタリングを停止すると、イメージスキャンステータスコードは に設定`inactive`され、理由コードは に設定されます`expired`。次に、関連するすべての画像検出結果をクローズするようにスケジュールします。

環境に最適な再スキャン期間を設定します。例えば、イメージを頻繁に構築する場合は、短いスキャン時間を選択します。同様に、イメージを長期間使用する場合は、より長いスキャン期間を選択します。

委任された管理者アカウントから再スキャン期間を設定すると、Amazon Inspector は組織内のすべてのメンバーアカウントに 設定を適用します。

ECR 再スキャン期間を設定するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ナビゲーションペインから、全般設定 を選択し、ECR スキャン設定 を選択します。
3. ECR スキャン設定 で、ECR 再スキャン期間 で、設定するイメージプッシュ日の期間とイメージプル日の期間を選択します。
4. [保存] を選択します。新しい設定はすぐに適用されます。

> **Note**
>
> プッシュ日の期間を長くすると、Amazon Inspector は継続的スキャン用に設定されたリポジトリ内のすべてのアクティブスキャン済みイメージに変更を適用します。ただし、非アクティブなイメージは、新しい期間内にプッシュした場合でも非アクティブのままになります。

Amazon Inspector による AWS Lambda 関数のスキャン

Amazon Inspector による AWS Lambda 関数のサポートにより、Lambda 関数とレイヤーのセキュリティ脆弱性を継続的に自動評価できます。Amazon Inspector では、2 種類の Lambda スキャンを提供しています。これらのスキャンタイプは、異なるタイプの脆弱性を検出します。

Amazon Inspector Lambda 標準スキャン

これはデフォルトの Lambda スキャンタイプです。Lambda 標準スキャンは、Lambda 関数とそのレイヤー内のアプリケーションの依存関係をスキャンして、パッケージの脆弱性がないか調べます。詳細については、「Lambda 標準スキャン」を参照してください。

Amazon Inspector Lambda コードスキャン

このスキャンタイプは、関数やレイヤー内のカスタムアプリケーションコードをスキャンして、コードの脆弱性がないか調べます。Lambda 標準スキャンをアクティブ化することも、Lambda コードスキャンと同時に Lambda 標準スキャンをアクティブ化することもできます。詳細については、「Amazon Inspector Lambda コードスキャン」を参照してください。

Lambda スキャンをアクティブ化すると、Amazon Inspector はアカウントに以下の AWS CloudTrail サービスにリンクされたチャネルを作成します。

- `cloudtrail:CreateServiceLinkedChannel`
- `cloudtrail:DeleteServiceLinkedChannel`

Amazon Inspector はこれらのチャネルを管理し、それらを使用してスキャンの CloudTrail イベントをモニタリングします。サービスにリンクされたチャネルの詳細については、「 AWS CLI を使用した のサービスにリンク CloudTrail されたチャネルの表示」を参照してください。

> **Note**
>
> Amazon Inspector によって作成されたサービスにリンクされたチャネルでは、証 CloudTrail 跡があるかのようにアカウントの CloudTrail イベントを表示できますが、アカウントのイベントを管理する CloudTrail ために独自の を作成することをお勧めします。

Lambda 関数スキャンのアクティブ化の手順については、「スキャンタイプをアクティブ化する」を参照してください。

Lambda 関数スキャンのスキャン動作

Amazon Inspector はアクティベーション時に、アカウント内で過去 90 日間に呼び出された、または更新されたすべての Lambda 関数をスキャンします。Amazon Inspector は、次のような状況で Lambda 関数の脆弱性スキャンを開始します。

- Amazon Inspector が既存の Lambda 関数を検出した場合。
- 新しい Lambda 関数が Lambda サービスにデプロイされた場合。
- 既存の Lambda 関数またはそのレイヤーのアプリケーションコードまたは依存関係に更新がデプロイされた場合。

- Amazon Inspector がデータベースに新しい共通脆弱性識別子 (CVE) 項目を追加し、その CVE が関数に関連している場合。

Amazon Inspector は、各 Lambda 関数が削除されるかスキャンから除外されるまで、そのライフタイム期間を通じてモニタリングします。

Lambda 関数の脆弱性の最終確認日は、[アカウント管理] ページの [Lambda 関数] タブから、または ListCoverage API を使用して確認できます。Amazon Inspector は、以下のイベントに応じて Lambda 関数の [最終スキャン日] フィールドを更新します。

- Amazon Inspector が Lambda 関数の初回スキャンを完了した時。
- Lambda 関数の更新時。
- Amazon Inspector が Lambda 関数を再スキャンしたとき。これは、その関数に影響する新しい CVE 項目が Amazon Inspector データベースに追加されたためです。

サポートされているランタイムと対象となる関数

Amazon Inspector は、Lambda 標準スキャンと Lambda コードスキャンのさまざまなランタイムをサポートしています。各スキャンタイプでサポートされているランタイムのリストについては、「サポートされているランタイム: Amazon Inspector Lambda 標準スキャン」と「サポートされているランタイム: Amazon Inspector Lambda コードスキャン」を参照してください。

Lambda 関数が Amazon Inspector スキャンの対象となるには、ランタイムがサポートされていることに加えて、以下の基準を満たす必要があります。

- この関数は過去 90 日間に呼び出されたか、または更新されています。
- この関数は `$LATEST` にマークされています。
- この関数はタグによるスキャンから除外されていません。

> **Note**
>
> 過去 90 日間に呼び出されたり変更されたりしていない Lambda 関数は、自動的にスキャンから除外されます。Amazon Inspector は、再度呼び出されたり、Lambda 関数コードに変更が加えられたりする場合に、自動的に除外された関数のスキャンを再開します。

Amazon Inspector Lambda 標準スキャン

Amazon Inspector Lambda 標準スキャンでは、Lambda 関数のコードとレイヤーに追加したアプリケーションパッケージの依存関係内にあるソフトウェアの脆弱性を特定します。たとえば、Lambda 関数が既知の脆弱性があるバージョンの `python-jwt` パッケージを使用している場合、Lambda 標準スキャンはその関数の検出結果を生成します。

Amazon Inspector が Lambda 関数のアプリケーションパッケージの依存関係に脆弱性を検出すると、Amazon Inspector は詳細なパッケージ 脆弱性タイプの検出結果を生成します。

スキャンタイプをアクティブ化する手順については、「スキャンタイプをアクティブ化する」を参照してください。

> **Note**
>
> Lambda 標準スキャンでは、Lambda ランタイム環境にデフォルトでインストールされている AWS SDK 依存関係はスキャンされません。Amazon Inspector は、関数コードとともにアップロードされた依存関係、またはレイヤーから継承された依存関係のみをスキャンします。

> **Note**
>
> Amazon Inspector Lambda 標準スキャンを非アクティブ化すると、Amazon Inspector Lambda コードスキャンも非アクティブ化になります。

Lambda 標準スキャンから関数の除外

特定の関数にタグを付けて、Amazon Inspector Lambda 標準スキャンから除外することができます。スキャンから関数を除外すると、実行不可能なアラートを防ぐことができます。

Lambda 関数を Lambda 標準スキャンから除外するには、関数に次のキーと値のペアをタグ付けします。

- キー: `InspectorExclusion`
- 値: `LambdaStandardScanning`

Lambda 標準スキャンから関数を除外するには

1. Lambda コンソール (https://console.aws.amazon.com/lambda/) を開きます。
2. [関数] を選択します。
3. 関数テーブルから、Amazon Inspector Lambda 標準スキャンから除外する関数の名前を選択します。
4. [設定] を選択し、メニューから [タグ] を選択します。
5. [タグを管理]、[新しいタグを追加] の順に選択します。
6. [キー] フィールドに「`InspectorExclusion`」と入力し、[値] フィールドに「`LambdaStandardScanning`」と入力します。
7. [保存] を選択してタグを追加し、Amazon Inspector Lambda 標準スキャンから関数を除外します。

Lambda へのタグの追加について詳しくは、「Lambda 関数でのタグの使用」を参照してください。

Amazon Inspector Lambda コードスキャン

> **Important**
>
> コードスキャンは、Lambda 関数からコードスニペットをキャプチャして、検出された脆弱性をハイライトします。これらのスニペットには、ハードコードされた認証情報やその他の機密情報がプレーンテキストで表示される場合があります。

Amazon Inspector Lambda コードスキャンは、 AWS セキュリティのベストプラクティスに基づいて、Lambda 関数内のカスタムアプリケーションコードをスキャンして、コードの脆弱性を検出します。Lambda コードスキャンでは、コード内のインジェクションの欠陥、データ漏洩、脆弱な暗号化、または暗号化の欠落を検出できます。利用可能なリージョンについては、「リージョン固有機能の可用性」を参照してください。

Lambda 標準スキャンは、関数で使用されるアプリケーションパッケージの依存関係を評価して、共通脆弱性識別子 (CVE) がないかを評価する機能です。Lambda コードスキャンを Lambda 標準スキャンと同時にアクティブ化できます。

Amazon Inspector は、自動推論と機械学習を使用して Lambda 関数のアプリケーションコードを評価し、アプリケーションコードを分析して全体的なセキュリティコンプライアンスを確認します。Amazon との共同開発の内部ディテクターに基づいて、ポリシー違反と脆弱性を特定します

CodeGuru。可能な検出のリストについては、CodeGuru「ディテクターライブラリ」を参照してください。

Amazon Inspector が Lambda 関数のアプリケーションコードに脆弱性を検出すると、Amazon Inspector は詳細なコード脆弱性タイプの検出結果を生成します。この検出結果タイプには、コード内の問題の正確な場所、問題を示すコードスニペット、および推奨される修復方法が含まれます。推奨される修復には、脆弱な plug-and-play コード行を置き換えるために使用できるコードブロックが含まれています。これらのコード修正案は、その検出結果に対する一般的なコード修正ガイダンスに加えて提供されます。

> **Important**
>
> コード修正案は、自動推論と生成 AI サービスを利用しているため、意図したとおりに機能しない場合があります。採用するコード修正案に対する責任はユーザーにあります。採用する前に、必ず修正案を確認してください。コードが意図したとおりに実行されるように、コードの修正案の編集が必要となる場合があります。「責任ある AI ポリシー」を参照してください。

コードの脆弱性検出結果におけるコードの暗号化

Lambda コードスキャンを使用したコード脆弱性の検出結果に関連して検出されたコードスニペットは、 CodeGuru サービスによって保存されます。デフォルトでは、 によって管理される AWS 所有キー CodeGuru を使用してコードを暗号化しますが、Amazon Inspector API による暗号化には独自のカスタマーマネージドキーを使用できます。詳細については、「検出結果のコードの保管時の暗号化」を参照してください。

Lambda コードスキャンは、Lambda 標準スキャンと同時にアクティブ化できます。スキャンタイプをアクティブ化する手順については、「スキャンタイプをアクティブ化する」を参照してください。

Lambda コードスキャンから関数の除外

特定の関数にタグを付けて、Amazon Inspector Lambda コードスキャンから除外することができます。スキャンから関数を除外すると、実行不可能なアラートを防ぐことができます。

Amazon Inspector から Lambda 関数を除外するには、Lambda コードスキャンは関数に次のキーと値のペアをタグ付けします。

- キー: `InspectorCodeExclusion`

- 値: `LambdaCodeScanning`

Lambda コードスキャンから関数を除外するには

1. Lambda コンソール (https://console.aws.amazon.com/lambda/) にサインインします。
2. [関数] を選択します。
3. 関数テーブルから、Amazon Inspector Lambda コードスキャンから除外する関数の名前を選択します。
4. [設定] を選択し、メニューから [タグ] を選択します。
5. [タグを管理]、[新しいタグを追加] の順に選択します。
6. [キー] フィールドに「`InspectorCodeExclusion`」と入力し、[値] フィールドに「`LambdaCodeScanning`」と入力します。
7. [保存] を選択してタグを追加し、Amazon Inspector Lambda コードスキャンから関数を除外します。

Lambda へのタグの追加について詳しくは、「Lambda 関数でのタグの使用」を参照してください。

スキャンタイプの非アクティブ化

新しい Amazon Inspector スキャンタイプはいつでも非アクティブ化できます。スキャンタイプを非アクティブ化すると、そのスキャンタイプによって生成された既存の検出結果にはアクセスできなくなります。スキャンタイプを再度アクティブ化すると、対象となるリソースがスキャンされ、Amazon Inspector が新しい検出結果を生成します。検出結果データを記録しておくため、非アクティブ化する前に検出結果をエクスポートできます。詳細については、「Amazon Inspector からの調査結果レポートのエクスポート」を参照してください。

スキャンタイプを非アクティブ化すると、非アクティブ化されるスキャンタイプに応じて、その AWS アカウントで特定の変更が発生する可能性があります。これらのスキャンタイプを非アクティブ化すると、次のような変更が生じます。

- Amazon EC2 スキャン — アカウントの Amazon EC2 スキャンを非アクティブ化すると、Amazon Inspector が使用する次の SSM 関連付けが削除されます。
  - `InspectorDistributor-do-not-delete`
  - `InspectorInventoryCollection-do-not-delete`
  - `InspectorLinuxDistributor-do-not-delete`

  - `InvokeInspectorLinuxSsmPlugin-do-not-delete`
  - `InvokeInspectorSsmPlugin-do-not-delete`。 さらに、この関連付けを通じてインストールされた Amazon Inspector SSM プラグインは、すべてのWindowsホストから削除されます。詳細については、「Windows インスタンスのスキャン」を参照してください。
- Amazon ECR スキャン — アカウントの Amazon ECR コンテナイメージスキャンを非アクティブ化すると、そのアカウントの Amazon ECR スキャンタイプが Amazon Inspector による拡張スキャンから Amazon ECR による基本的なスキャンに変わります。
- Lambda 標準スキャン — アカウントの Lambda 標準スキャンを非アクティブ化すると、コードスキャンもアクティブになっていれば Lambda コードスキャンは非アクティブ化になります。さらに、スキャンが有効になったときに作成された CloudTrail サービスリンクチャネルも削除されます。

スキャンの非アクティブ化

アカウントのすべてのスキャンタイプを非アクティブ化すると、 AWS リージョン内のそのアカウントの Amazon Inspector も非アクティブ化されます 詳細については、「Amazon Inspector の非アクティブ化」を参照してください。

マルチアカウント環境でこの手順を完了するには、Amazon Inspector の委任された管理者としてサインインした状態で以下の手順を行います。

Console

スキャンを非アクティブ化するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、スキャンを非アクティブ化するリージョンを選択します。
3. ナビゲーションペインで、[アカウント管理] を選択します。
4. [アカウント] タブを選択すると、アカウントのスキャンステータスが表示されます。
5. スキャンを非アクティブ化するアカウントのチェックボックスを選択します。
6. [アクション] を選択し、[非アクティブ化] オプションから、非アクティブ化するスキャンタイプを選択します。
7. （推奨) スキャンタイプを非アクティブ化する各 AWS リージョン で、これらのステップを繰り返します。

API

無効化 APL オペレーションを実行します。リクエストには、スキャンを非アクティブ化するアカウント ID を指定し、resourceTypes についてはスキャンを非アクティブ化するために、EC2、ECR、LAMBDA、または LAMBDA_CODE の 1 つ以上を指定します。

EC2 インスタンスのインターネットセキュリティセンター (CIS) スキャン

アカウントの Amazon Inspector EC2 スキャンを有効にすると、Amazon Inspector が CIS スキャンを実行またはスケジュールできるようになります。Amazon Inspector CIS スキャンは、Amazon EC2 インスタンスのオペレーティングシステムをベンチマークして、Center for Internet Security によって確立されたベストプラクティスの推奨事項に従って設定されているかどうかを確認します。CIS Security Benchmarks プログラムは、システムを安全に設定するための業界標準の設定ベースラインとベストプラクティスを提供します。詳細については、「CIS Benchmarks とは」を参照してください。

Amazon Inspector は、スキャン設定で定義したインスタンスタグとスキャンスケジュールに基づいて、ターゲット Amazon EC2 インスタンスで CIS スキャンを実行します。ターゲットインスタンスごとに、Amazon Inspector はインスタンスに対して一連のチェックを実行します。各チェックでは、システム設定が特定の CIS Benchmark レコメンデーションを満たしているかどうかを評価します。すべてのチェックには CIS チェック ID とタイトルがあり、そのプラットフォームの CIS Benchmark レコメンデーションに直接関連しています。スキャンが完了すると、結果を表示し、そのシステムでインスタンスが合格、失敗、スキップされたチェックを確認できます。

Amazon Inspector CIS スキャンの EC2 インスタンス要件

インスタンスで CIS スキャンを実行するには、Amazon Inspector では、インスタンスが次の基準を満たしている必要があります。

- インスタンスオペレーティングシステムは、CIS スキャンでサポートされているオペレーティングシステムの 1 つです。サポートされているオペレーティングシステムの完全なリストについては、「サポートされているオペレーティングシステム: CIS スキャン」を参照してください。
- インスタンスは Amazon EC2 Systems Manager (SSM) マネージドインスタンスです。詳細については、「SSM Agent の使用」を参照してください。
- インスタンスには Amazon Inspector SSM プラグインがインストールされています。Amazon Inspector は、SSM マネージドインスタンス用にこのプラグインを自動的にインストールします。
- インスタンスには、SSM がインスタンスを管理するためのアクセス許可を付与するインスタンスプロファイルと、そのインスタンスの CIS スキャンを実行するための Amazon Inspector があります。これらのアクセス許可を付与するには、AmazonInspector2、AmazonSSMFullAccess、および AmazonInspector2ManagedCispolicy つのポリシーを IAM ロールにアタッチし、そのロールをインスタンスプロファイルとしてインスタンスにアタッチします。

AmazonSSMManagedInstanceCore インスタンスプロファイルを作成してアタッチする手順については、「Amazon EC2 ユーザーガイド」の「IAM ロールの操作」を参照してください。Amazon EC2

> **Note**
>
> インスタンスで CIS スキャンを実行する場合、Amazon Inspector のディープインスペクションを有効にすることはもはや必須ではありません。詳細検査を無効にすると、Amazon Inspector は引き続き SSM エージェントをインストールしますが、プラグインが呼び出されて詳細検査を実行できなくなります。つまり、アカウントに次の関連付けが存在することになります: `InspectorLinuxDistributor-do-not-delete`。

CIS スキャンの実行

CIS スキャンは、オンデマンドで 1 回実行することも、スケジュールされた定期的なスキャンとして実行することもできます。スキャンを実行するには、まずスキャン設定を作成します。

スキャン設定を作成するときは、インスタンスをターゲットにするタグのキーと値のペアを指定します。ユーザーが組織の Amazon Inspector の委任管理者である場合、スキャン設定で複数のアカウントを指定でき、Amazon Inspector は各アカウントの指定されたタグを持つインスタンスを検索します。スキャンの CIS ベンチマークレベルを選択します。CIS は、ベンチマークごとにレベル 1 とレベル 2 のプロファイルをサポートしています。これは、異なる環境が必要とするさまざまなレベルのセキュリティのベースラインを提供するように設計されています。

- レベル 1 — は、どのシステムでも設定できる基本的なセキュリティ設定を推奨します。これらの設定を実装すると、サービスの中断はほとんどまたはまったく発生しません。これらのレコメンデーションの目的は、システムへのエントリポイントの数を減らし、全体的なサイバーセキュリティリスクを減らすことです。
- レベル 2 — では、セキュリティの高い環境では、より高度なセキュリティ設定を推奨しています。これらの設定を実装するには、ビジネスへの影響のリスクを最小限に抑えるための計画と調整が必要です。これらのレコメンデーションの目的は、規制コンプライアンスの達成を支援することです。

レベル 2 はレベル 1 を拡張します。レベル 2 を選択すると、Amazon Inspector はレベル 1 とレベル 2 に推奨されるすべての設定をチェックします。

スキャンのパラメータを定義したら、設定の完了後に実行する 1 回限りのスキャンとして実行するか、定期的なスキャンとして実行するかを選択できます。繰り返しスキャンは、毎日、毎週、または毎月、任意のタイミングで実行できます。

> **Tip**
>
> スキャンの実行中にシステムに影響を与える可能性が最も低い日時を選択することをお勧めします。

CIS スキャン設定を作成するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、CIS スキャンを実行する AWS リージョン を選択します。
3. ナビゲーションパネルのオンデマンドスキャン で、CIS スキャン を選択します。
4. 新しいスキャンの作成 を選択します。
   a. スキャン設定名 を入力します。
   b. ターゲットリソースには、スキャンするインスタンスのタグのキーと対応する値を入力します。スキャンに含めるタグは合計 25 個指定でき、キーごとに最大 5 つの異なる値を指定できます。
   c. CIS ベンチマークレベル を選択します。基本的なセキュリティ設定にはレベル 1、高度なセキュリティ設定にはレベル 2 を選択できます。
5. ターゲットアカウント で、スキャンに含めるアカウントを指定します。スタンドアロンアカウントまたは組織のメンバーは、自己を選択して、自分のアカウントのスキャン設定を作成できます。Amazon Inspector の委任管理者は、すべてのアカウントを選択して組織内のすべてのアカウントをターゲットにするか、アカウントを指定を選択してターゲットにするメンバーアカウントのサブセットを指定できます。委任管理者は、アカウント ID SELFの代わりに を入力して、自分のアカウントのスキャン設定を作成できます。詳細については、AWS 組織内の Amazon Inspector CIS スキャンを管理する際の考慮事項を参照してください。
6. スキャンのスケジュールを選択します。スキャン設定の作成が完了するとすぐに実行される 1 回限りのスキャン、または削除されるまでスケジュールされた時間に実行される の繰り返しスキャンのいずれかを選択します。
7. 作成 を選択して、スキャン設定の作成を完了します。

CIS スキャン設定の表示と編集

以前にスケジュールしたスキャンはいつでも表示または編集できます。

CIS スキャン設定を表示または編集するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、CIS スキャン設定を作成した を選択します AWS リージョン 。
3. ナビゲーションパネルのオンデマンドスキャン で、CIS スキャン を選択します。
4. スケジュールされたスキャン設定を表示するには、スケジュールされた を選択します。
5. スキャン設定名列から項目を選択して、そのスキャン設定の詳細を開きます。
6. （オプション) 編集 を選択して、このスキャンのパラメータを変更します。

CIS スキャンの結果の表示

Amazon Inspector は、スキャン設定が実行されるたびにスキャンジョブを作成し、一意のスキャン ID でスキャンの結果を収集します。

スキャン結果は、スキャン完了後 90 日間使用できます。スキャンの結果は、チェックまたはターゲットリソース別に集計して表示できます。

チェックによって集計されたスキャン結果

スキャンの結果は、スキャン中に実行された個々のチェックごとにグループ化されます。チェックごとに、合格、不合格、スキップされたリソースの数に関するレポートが表示されます。

リソース別に集計されたスキャン結果

スキャンの結果は、スキャン設定がターゲットとする各リソース別にグループ化されます。リソースごとに、そのリソースに対して合格、不合格、またはスキップされたリソースをチェックするレポートを取得します。

スキャン結果を表示するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。

2. ページの右上隅にある AWS リージョン セレクターを使用して、スキャン結果を表示する AWS リージョン を選択します。
3. ナビゲーションパネルのオンデマンドスキャン で、CIS スキャン を選択します。
4. スキャン ID 列から、結果を表示するスキャンの ID を選択します。
5. スキャン結果の表示方法を選択します。
   - チェックタブを選択すると、チェック別に集計されたスキャン結果が表示されます。
     - リストされたチェックでは、リソースステータス列で合格、スキップ、または不合格の数値を選択して、そのステータスとそのチェックでフィルタリングされたリソースのビューを開きます。
   - スキャンされたリソース タブを選択すると、リソースごとに集計されたスキャン結果が表示されます。
     - リソースを選択して、リソースが合格、失敗、スキップしたチェックを一覧表示する詳細パネルを開きます。
6. (オプション) いずれかのビューのフィルターバーを使用して結果を絞り込みます。

CIS スキャンの結果は、コンソールまたは API を使用してダウンロードできます。

スキャン結果をダウンロードするには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、スキャン結果を表示する AWS リージョン を選択します。
3. ナビゲーションパネルのオンデマンドスキャン で、CIS スキャン を選択します。
4. スキャン ID 列から、結果を表示するスキャンの ID を選択します。
5. [ダウンロード] を選択します。委任管理者の場合は、特定のメンバーアカウントの結果をダウンロードすることを選択できます。

AWS 組織内の Amazon Inspector CIS スキャンを管理する際の考慮事項

組織内で CIS スキャンを実行すると、メンバーアカウントと Amazon Inspector の委任管理者は CIS スキャン設定とスキャン結果をさまざまな方法で操作します。

委任管理者がすべてのアカウントの CIS スキャン設定を作成するか、組織がそのスキャン設定を所有しているメンバーアカウント IDs のリストを作成する場合。現在の委任管理者がどのアカウントでも、別のアカウントがスキャン設定を作成している場合でも、組織が所有するスキャン設定を管理できます。組織が所有する CIS スキャン設定には、次のパターンに従って、組織 ID を所有者として一覧表示する ARN があります: `arn:aws:inspector2:Region:111122223333:owner/OrganizationId/cis-configuration/scanId`。アカウント ID は Organizations 管理アカウントの ID になります。

> **Important**
>
> 組織が所有する CIS スキャン設定にタグを追加することはできません。

委任管理者がスキャン設定を作成し、ターゲットアカウントSELFとして を指定すると、そのアカウントがそのスキャン設定を所有します。組織を離れても、そのスキャン設定を管理できます。

> **Note**
>
> 委任管理者は、 をターゲットとするスキャン設定のターゲットを変更することはできませんSELF。

メンバーアカウント、スタンドアロンアカウント、または委任された管理者が をターゲットSELFとして作成したスキャン設定は、それらを作成したアカウントによって所有されます。これらの CIS スキャン設定には、パターン に従って、そのアカウントを所有者として一覧表示する ARN がありますarn:aws:inspector2:*Region*:*111122223333*:owner/*111122223333*/cis-configuration/*scanId*。アカウント ID は、スキャンを作成したアカウントです。

組織のメンバーアカウントは、自分のアカウントに対してスキャン設定を作成できます。委任管理者は、メンバーによって作成されたスキャン設定を表示できますが、編集または削除することはできません。メンバーアカウントが組織を離れると、委任された管理者は、そのアカウントによって作成されたスキャン設定を表示できなくなります。

委任管理者は、メンバーによってスケジュールされたアカウントを含め、組織内の任意のアカウントのスキャン結果を表示できます。メンバーアカウントは、委任された管理者によってスケジュールされたリソースを含む、アカウント内のリソースの CIS スキャンの結果を表示できます。

Amazon Inspector CIS スキャンに使用される Amazon Inspector 所有の Amazon S3 バケット Amazon Inspector

Amazon Inspector は、CIS スキャンに必要な更新されたオープン脆弱性評価言語 (OVAL) 定義ファイルをステージングします。次の表は、サポートされている ごとに CIS スキャンが使用する OVAL 定義を持つ Amazon Inspector が所有するすべての Amazon S3 バケットを示しています AWS リージョン。 Amazon S3 必要に応じて、バケットを VPCsで許可リストに登録する必要があります。

> **Note**
>
> 以下の Amazon Inspector 所有の Amazon S3 バケットの詳細は変更されません。ただし、リストは、新しくサポートされた新しい を反映するように更新される場合があります AWS リージョン。これらのバケットを他の Amazon S3 オペレーションや独自の Amazon S3 バケットで使用することはできません。

CIS バケット	AWS リージョン
cis-datasets-prod-arn-5908f6f	欧州 (ストックホルム)
cis-datasets-prod-bah-8f88801	中東 (バーレーン)
cis-datasets-prod-bjs-0f40506	中国 (北京)
cis-datasets-prod-bom-435a167	アジアパシフィック (ムンバイ)
cis-datasets-prod-cdg-f3a9c58	欧州 (パリ)
cis-datasets-prod-cgk-09eb12f	アジアパシフィック (ジャカルタ)
cis-datasets-prod-cmh-63030b9	米国東部 (オハイオ)
cis-datasets-prod-cpt-02c5c6f	アフリカ (ケープタウン)
cis-datasets-prod-dub-984936f	欧州 (アイルランド)
cis-datasets-prod-fra-6eb96eb	欧州 (フランクフルト)
cis-datasets-prod-gru-de69f99	南米 (サンパウロ)

CIS バケット	AWS リージョン
cis-datasets-prod-hkg-8e30800	アジアパシフィック (香港)
cis-datasets-prod-iad-8438411	米国東部 (バージニア北部)
cis-datasets-prod-icn-f4eff1c	アジアパシフィック (ソウル)
cis-datasets-prod-kix-5743b21	アジアパシフィック (大阪)
cis-datasets-prod-lhr-8b1fbd0	欧州 (ロンドン)
cis-datasets-prod-mxp-7b1bbce	欧州 (ミラノ)
cis-datasets-prod-nrt-464f684	アジアパシフィック (東京)
cis-datasets-prod-osu-5bead6f	AWS GovCloud （米国東部）
cis-datasets-prod-pdt-adadf9c	AWS GovCloud （米国西部）
cis-datasets-prod-pdx-acfb052	米国西部 (オレゴン)
cis-datasets-prod-sfo-1515ba8	米国西部 (北カリフォルニア)
cis-datasets-prod-sin-309725b	アジアパシフィック (シンガポール)
cis-datasets-prod-syd-f349107	アジアパシフィック (シドニー)
cis-datasets-prod-yul-5e0c95e	カナダ (中部)
cis-datasets-prod-zhy-5a8eacb	中国 (寧夏)
cis-datasets-prod-zrh-67e0e3d	欧州 (チューリッヒ)

Amazon Inspector による AWS 環境のカバレッジを評価する

AWS 環境の Amazon Inspector カバレッジの評価と解釈に役立つように、Amazon Inspector コンソールのアカウント管理ページには、アカウントとリソースの Amazon Inspector スキャンのステータスに関する統計と詳細が表示されます。このページでは、リソースの集約された統計やその他のデータを確認することができます。また、個々のリソースの Amazon Inspector の対象範囲を詳細に分析し、ドリルダウンして特定のリソースの検出結果を確認することもできます。ご自身が組織の委任された Amazon Inspector 管理者である場合、データには、組織内のすべてのアカウントの統計情報と詳細が含まれます。

AWS 環境の Amazon Inspector カバレッジを評価するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ナビゲーションペインで、[アカウント管理] を選択します。
3. [アカウント管理] ページで、5 つの異なるカバレッジビューのいずれかのタブを選択します。
    - アカウント、アカウントレベルのカバレッジ。
    - インスタンス、Amazon Elastic Compute Cloud (Amazon EC2)インスタンスのカバレッジ。
    - レポジトリ、Amazon Elastic Container Registry (Amazon ECR)リポジトリのカバレッジ。
    - イメージ、Amazon ECR コンテナイメージのカバレッジ。
    - Lambda、Lambda 関数のカバレッジ。

このセクションのトピックでは、個々のリソースのスキャンステータスなど、各タブに表示される情報について説明します。

トピック

アカウントレベルのカバレッジを評価する

アカウントが組織の一部ではない場合、または組織の委任された Amazon Inspector 管理者アカウントではない場合、[アカウント] タブには、アカウントに関する情報と、アカウントのリソーススキャンのステータスが表示されます。このタブでは、アカウントのすべてまたは特定のタイプのリソースのみのスキャンをアクティブ化または非アクティブ化にできます。詳細については、「Amazon Inspector による自動リソーススキャン」を参照してください。

アカウントが組織の委任された Amazon Inspector 管理者アカウントの場合、[アカウント] タブには組織内のアカウントの自動アクティベーション設定が表示され、組織内のすべてのアカウントが一覧表示されます。アカウントごとに、そのアカウントで Amazon Inspector がアクティブ化になっているかどうかを表示し、その場合は、そのアカウントでアクティブ化になっているリソーススキャンタイプがリストに表示されます。委任された管理者は、このタブを使用して組織の自動アクティベーション設定を変更できます。また、個々のメンバーアカウントの特定のタイプのリソーススキャンをアクティブ化または非アクティブ化することもできます。詳細については、「メンバーアカウントの Amazon Inspector スキャンをアクティブ化する」を参照してください。

Amazon EC2 インスタンスのカバレッジを評価する

[インスタンス] タブには、 AWS 環境内の Amazon EC2 インスタンスが表示されます。リストは次のタブにグループ分けされています。

- すべて — 環境内のすべてのインスタンスを表示します。ステータス列には、インスタンスの現在のスキャンステータスが表示されます。
- スキャン — Amazon Inspector が環境でアクティブにモニタリングおよびスキャンしているすべてのインスタンスを表示します。
- スキャンしない — Amazon Inspector が環境でモニタリングおよびスキャンしていないすべてのインスタンスを表示します。理由列には、Amazon Inspector がインスタンスをモニタリングおよびスキャンしていない理由が表示されます。

  EC2 インスタンスが [スキャンしない] タブに表示される理由はいくつかあります。Amazon Inspector は AWS Systems Manager (SSM) と SSM エージェントを使用して、EC2 インスタンスの脆弱性を自動的にモニタリングおよびスキャンします。インスタンスで SSM Agent が実行されていない場合、Systems Manager をサポートする AWS Identity and Access Management (IAM) ロールがない場合、またはサポートされているオペレーティングシステムやアーキテクチャを実行していない場合、Amazon Inspector はインスタンスをモニタリングおよびスキャンできません。詳細については、「Amazon EC2 インスタンスのスキャン」を参照してください。

各タブで、 アカウント列はインスタンスを所有 AWS アカウント する を指定します。

[EC2 インスタンスタグ] — この列には、インスタンスに関連付けられているタグが表示され、インスタンスがタグによるスキャンから除外されているかどうかを判断できます。

[オペレーティングシステム] — この列には、オペレーティングシステムの種類 (WINDOWS、MAC、LINUX、または UNKNOWN) が表示されます。

[使用中の監視] — この列には、このインスタンスで Amazon Inspector がエージェントベースのスキャン方式を使用しているか、エージェントレスのスキャン方式を使用しているかが表示されます。

[最終スキャン日] - この列には、Amazon Inspector がそのリソースの脆弱性を最後にチェックした日時が表示されます。Amazon Inspector がスキャンを実行する頻度は、インスタンスのスキャンに使用されているスキャン方式によって異なります。

EC2 インスタンスに関するその他の詳細を確認するには、EC2 インスタンス列のリンクを選択します。次に、Amazon Inspector はインスタンスに関する詳細と、そのインスタンスに関する現在の検出結果を表示します。特定の検出結果の詳細を確認するには、[タイトル] 列のリンクを選択します。その詳細については、「Amazon Inspector の検出結果の詳細」を参照してください。

Amazon EC2 インスタンスのステータス値のスキャン

Amazon Elastic Compute Cloud (Amazon EC2) インスタンスの場合、[ステータス] の値は以下のようになります：

- アクティブにモニタリング中 — Amazon Inspector ではインスタンスを継続的にモニタリングおよびスキャンしています。
- EC2 インスタンスが停止 — インスタンスが停止状態になったため、Amazon Inspector はインスタンスのスキャンを一時停止しました。既存の検出結果はすべて、インスタンスが終了するまで保持されます。インスタンスが再起動されると、Amazon Inspector はインスタンスのスキャンを自動的に再開します。
- 内部エラー — Amazon Inspector がインスタンスをスキャンしようとしたときに内部エラーが発生しました。Amazon Inspector は自動的にエラーに対処し、できるだけ早くスキャンを再開します。
- インベントリなし — Amazon Inspector は、インスタンスをスキャンするソフトウェアアプリケーションインベントリを見つけることができませんでした。インスタンスの Amazon Inspector の関連付けが削除されているか、実行に失敗した可能性があります。

この問題を修正するには、 AWS Systems Manager を使用して、`InspectorInventoryCollection-do-not-delete`関連付けが存在し、関連付けのステータスが成功していることを確認します。さらに、 AWS Systems Manager Fleet Manager を使用してインスタンスのソフトウェアアプリケーションインベントリを確認します。

- 保留中の無効化 — Amazon Inspector はインスタンスのスキャンを停止しました。インスタンスは無効になっており、クリーンアップタスクの完了を待っています。
- 保留中の初期スキャン — Amazon Inspector は初期スキャンのためにインスタンスをキューに入れました。
- リソースが終了しました - インスタンスが終了しました。Amazon Inspector は現在、インスタンスの既存の検出結果とカバレッジデータをクリーンアップしています。
- 古いインベントリ — Amazon Inspector は、過去 7 日以内にインスタンスについてキャプチャされた更新されたソフトウェアアプリケーションインベントリを収集できませんでした。

  この問題を修正 AWS Systems Manager するには、 を使用して、インスタンスに必要な Amazon Inspector の関連付けが存在し、実行されていることを確認します。さらに、 AWS Systems Manager Fleet Manager を使用してインスタンスのソフトウェアアプリケーションインベントリを確認します。
- アンマネージド型 EC2 インスタンス — Amazon Inspector はインスタンスをモニタリングまたはスキャンしていません。インスタンスは AWS Systems Managerによって管理されていません。

  この問題を修正するには、 AWS Systems Manager オートメーションAWSSupport-TroubleshootManagedInstance runbookが提供する を使用できます。インスタンスを管理する AWS Systems Manager ように を設定すると、Amazon Inspector は自動的にインスタンスの継続的なモニタリングとスキャンを開始します。
- サポートされていない OS — Amazon Inspector はインスタンスをモニタリングまたはスキャンしていません。インスタンスは、Amazon Inspector がサポートしていないオペレーティングシステムまたはアーキテクチャを使用しています。Amazon Inspector がサポートしているオペレーティングシステムのリストについては、「Amazon EC2 のスキャンでサポートされているオペレーティングシステム」を参照してください。
- アクティブにモニタリングして、部分的なエラーが見つかりました — このステータスは、EC2 スキャンはアクティブですが、Amazon EC2 Linux インスタンス向け Amazon Inspector 詳細検査 に関連するエラーがあることを意味します。考えられる詳細検査エラーは次のとおりです。
  - 詳細検査パッケージの収集制限を超えました – インスタンスが Amazon Inspector 詳細検査の 5000 パッケージ制限を超えました。このインスタンスの詳細な検査を再開するには、アカウントに関連付けられたカスタムパスの調整を試みます。

- Deep inspection daily ssm inventory limit exceeded – SSM エージェントが Amazon Inspector にインベントリを送信できませんでした。これは、このインスタンスで 1 日あたりインスタンスごとに収集されるインベントリデータの SSM クォータにすでに達しているためです。詳細については、「Amazon EC2 Systems Manager エンドポイントとクォータ」を参照してください。
- 詳細検査の収集時間の制限を超えました – パッケージ収集時間が最大しきい値の 15 分を超えたため、Amazon Inspector はパッケージインベントリの抽出に失敗しました。
- 詳細検査にインベントリがありません — Amazon Inspector SSM プラグインは、このインスタンスのパッケージのインベントリをまだ収集できていません。これは通常、保留中のスキャンの結果ですが、6 時間経ってもこの状態が続く場合は、Amazon EC2 Systems Manager を使用して、必要な Amazon Inspector 関連付けインスタンスで存在し、実行されていることを確認してください。

EC2 インスタンスのスキャン設定の詳細については、「Amazon EC2 インスタンスのスキャン」を参照してください。

Amazon ECR リポジトリのカバレッジを評価する

[リポジトリ] タブには、 AWS 環境内の Amazon ECR リポジトリが表示されます。リストは以下のタブでグループにまとめられています。

- すべて — 環境内のすべてのリポジトリを表示します。[ステータス] 列には、レポジトリの現在のスキャンステータスが表示されます。
- アクティブ化 — Amazon Inspector が環境でモニタリングおよびスキャンするように設定されているすべてのリポジトリを表示します。[ステータス] 列には、レポジトリの現在のスキャンステータスが表示されます。
- アクティブ化されていません — Amazon Inspector が環境でモニタリングおよびスキャンしていないすべてのリポジトリを表示します。[理由] 列には、Amazon Inspector がインスタンスをモニタリングおよびスキャンしていない理由が表示されます。

各タブで、アカウント列はリポジトリを所有 AWS アカウント する を指定します。

リポジトリに関するその他の詳細を確認するには、リポジトリの名前を選択します。次に、Amazon Inspector はリポジトリ内のコンテナイメージのリストと各イメージの詳細を表示します。詳細には、イメージタグ、イメージダイジェスト、スキャンステータスが含まれます。また、イメージの緊急の検出結果の数など、主要な検出結果の統計情報も含まれます。検出結果の裏付けとなるデータを掘り下げて確認するには、イメージの [イメージ] タグを選択します。

Amazon ECR リポジトリのステータス値のスキャン

Amazon Elastic Container Registry (Amazon ECR) リポジトリの場合、可能なステータス値は次のとおりです。

- 有効 (継続的) – リポジトリの場合、Amazon Inspector はこのリポジトリ内のイメージを継続的にモニタリングしています。リポジトリの拡張スキャン設定は継続的スキャンに設定されています。Amazon Inspector は、プッシュされたときに新しいイメージを最初にスキャンし、そのイメージに関連する新しい CVE が公開されたときにイメージを再スキャンします。Amazon Inspector は、設定した ECR スキャン期間中、このリポジトリ内のイメージを引き続きモニタリングします。
- 有効 (プッシュ時) – Amazon Inspector は、新しいイメージがプッシュされると、リポジトリ内の個々のコンテナイメージを自動的にスキャンします。拡張スキャンはリポジトリに対して有効になり、プッシュ時にスキャンするように設定されます。
- アクセス拒否 — Amazon Inspector は、リポジトリまたはリポジトリ内のコンテナイメージへのアクセスを許可されていません。

  この問題を修正するには、リポジトリの AWS Identity and Access Management (IAM) ポリシーで Amazon Inspector がリポジトリにアクセスすることを許可していることを確認します。
- 非アクティブ化 (手動) — Amazon Inspector はリポジトリ内のコンテナイメージをモニタリングまたはスキャンしていません。リポジトリの Amazon ECR スキャン設定は、基本的な手動スキャンに設定されています。

  Amazon Inspector を使用してリポジトリ内のイメージのスキャンを開始するには、リポジトリのスキャン設定を拡張スキャンに変更し、イメージを継続的にスキャンするか、新しいイメージがプッシュされたときにのみスキャンするかを選択します。
- 有効 (プッシュ時) – Amazon Inspector は、新しいイメージがプッシュされると、リポジトリ内の個々のコンテナイメージを自動的にスキャンします。リポジトリの拡張スキャン設定は、プッシュ時にスキャンするように設定されています。
- 内部エラー — Amazon Inspector がリポジトリをスキャンしようとしたときに内部エラーが発生しました。Amazon Inspector は自動的にエラーに対処し、できるだけ早くスキャンを再開します。

リポジトリ のスキャン設定の構成の詳細については、「」を参照してくださいAmazon ECR コンテナイメージのスキャン。

Amazon ECR コンテナイメージのカバレッジを評価する

[イメージ] タブには、環境内の Amazon ECR コンテナイメージが表示されます。 AWS リストは次のタブでグループにまとめられています。

- すべて — 環境内のすべてのコンテナイメージを表示します。[ステータス] 列には、イメージの現在のスキャンステータスが表示されます。
- スキャン — Amazon Inspector が環境でアクティブにモニタリングおよびスキャンしているすべてのコンテナイメージを表示します。[ステータス] 列には、イメージの現在のスキャンステータスが表示されます。
- スキャンしない — Amazon Inspector が環境でモニタリングおよびスキャンしていないすべてのコンテナイメージを表示します。[理由] 列には、Amazon Inspector がイメージをモニタリングおよびスキャンしていない理由が表示されます。

  コンテナイメージが [アクティブ化されていません] タブに表示される理由はいくつかあります。Amazon Inspector のスキャンがアクティブ化されていないリポジトリにイメージが保存されているか、Amazon ECR フィルタリングルールによってそのリポジトリがスキャンされない場合があります。または、ECR 再スキャン期間 に設定された日数内にイメージがプッシュまたはプルされていない。詳細については、「ECR 再スキャン期間の設定」を参照してください。

各タブの [リポジトリ名] 列には、コンテナイメージを格納するリポジトリの名前を指定します。Account 列は、リポジトリを所有 AWS アカウント する を指定します。[最終スキャン日] 列には、Amazon Inspector がそのリソースの脆弱性を最後にチェックした日時が表示されます。これには、メタデータの検出結果が更新されたとき、リソースのアプリケーションインベントリが更新されたとき、または新しい CVE に応じて再スキャンが行われたときのチェックが含まれます。詳細については、「Amazon ECR スキャンのスキャン動作」を参照してください。

コンテナイメージに関するその他の詳細を確認するには、[ECR コンテナイメージ] 列のリンクを選択します。次に、Amazon Inspector はイメージに関する詳細と、そのイメージに関する現在の検出結果を表示します。特定の検出結果の詳細を確認するには、[タイトル] 列のリンクを選択します。その詳細については、「Amazon Inspector の検出結果の詳細」を参照してください。

Amazon ECR コンテナイメージのステータス値のスキャン

Amazon Elastic Container Registry コンテナイメージの場合、可能なステータス値は次のとおりです。

- アクティブモニタリング (継続的) — Amazon Inspector は継続的にモニタリングされており、関連する新しい CVE が公開されるたびにイメージと新しいスキャンが実行されます。イメージの Amazon ECR 再スキャン期間は、イメージがプッシュまたはプルされるたびに更新されます。イメージを保存するリポジトリでは拡張スキャンが有効になっており、リポジトリの拡張スキャン設定は継続的スキャンに設定されています。
- 有効 (プッシュ時) — Amazon Inspector は、新しいイメージがプッシュされるたびに自動的にイメージをスキャンします。イメージを保存するリポジトリでは拡張スキャンがアクティブ化になり、リポジトリの拡張スキャン設定はプッシュ時にスキャンするように設定されます。
- 内部エラー — Amazon Inspector がコンテナイメージをスキャンしようとしたときに内部エラーが発生しました。Amazon Inspector は自動的にエラーに対処し、できるだけ早くスキャンを再開します。
- 保留中の初期スキャン — Amazon Inspector は初期スキャンのためにイメージをキューに入れました。
- スキャン資格の有効期限が切れました (継続的) — Amazon Inspector はイメージのスキャンを中断しました。リポジトリ内のイメージを自動再スキャンするように指定した期間内に、イメージが更新されていません。イメージをプッシュまたはプルしてスキャンを再開できます。
- スキャン資格の有効期限が切れました (プッシュ中) — Amazon Inspector はイメージのスキャンを中断しました。リポジトリ内のイメージを自動再スキャンするように指定した期間内に、イメージが更新されていません。イメージをプッシュしてスキャンを再開できます。
- スキャン頻度 (手動) — Amazon Inspector は Amazon ECR コンテナイメージをスキャンしません。イメージを保存するリポジトリの Amazon ECR スキャン設定は、基本の手動スキャンに設定されています。Amazon Inspector で自動的にイメージのスキャンを開始するには、リポジトリの設定を拡張スキャンに変更し、イメージを継続的にスキャンするか、新しいイメージがプッシュされたときにのみスキャンするかを選択します。
- サポートされていない OS – Amazon Inspector はイメージをモニタリングまたはスキャンしていません。イメージは、Amazon Inspector がサポートしていないオペレーティングシステムに基づいているか、Amazon Inspector がサポートしていないメディアタイプを使用しています。

  Amazon Inspector がサポートしているオペレーティングシステムのリストについては、「Amazon ECR スキャンでサポートされているオペレーティングシステム」を参照してください。Amazon Inspector がサポートするメディアタイプのリストについては、「サポートされているメディアタイプ」を参照してください。

リポジトリとイメージのスキャン設定の詳細については、「Amazon ECR コンテナイメージのスキャン」を参照してください。

AWS Lambda 関数のカバレッジの評価

Lambda タブには AWS 、環境内の Lambda 関数が表示されます。このページには 2 つのテーブルがあります。1 つは Lambda 標準スキャンの関数カバレッジの詳細を示し、もう 1 つは Lambda コードスキャンの関数カバレッジの詳細を示しています。以下のタブに基づいて関数をグループ化できます。

- すべて — 環境内のすべての Lambda 関数を表示します。[ステータス] 列には、Lambda 関数の現在のスキャンステータスが表示されます。
- スキャン — Amazon Inspector がスキャンするように設定されている Lambda 関数を表示します。[ステータス] 列には、各 Lambda 関数の現在のスキャンステータスが表示されます。
- スキャンしない — Amazon Inspector がスキャンするように設定されていない Lambda 関数を表示します。[理由] 列には、Amazon Inspector が関数をモニタリングおよびスキャンしていない理由が表示されます。

  Lambda 関数が [スキャンしない] タブに表示される理由はいくつかあります。Lambda 関数が Amazon Inspector に追加されていないアカウントに属しているか、フィルタリングルールによりこの関数がスキャンされない可能性があります。詳細については、「AWS Lambda 関数のスキャン」を参照してください。

各タブの [関数名] 列には、Lambda 関数の名前を指定します。Account 列は、関数を所有 AWS アカウント する を指定します。[ランタイム] には、関数のランタイムを指定します。[ステータス] 列には、各 Lambda 関数の現在のスキャンステータスが表示されます。[リソース] タグは、関数に適用されたタグを表示します。[最終スキャン日] 列には、Amazon Inspector がそのリソースの脆弱性を最後にチェックした日時が表示されます。これには、メタデータの検出結果が更新されたとき、リソースのアプリケーションインベントリが更新されたとき、または新しい CVE に応じて再スキャンが行われたときのチェックが含まれます。詳細については、「Lambda 関数スキャンのスキャン動作」を参照してください。

AWS Lambda 関数のステータス値のスキャン

Lambda 関数の場合、指定できるステータス値は次のとおりです。

- アクティブにモニタリング中 — Amazon Inspector は Lambda 関数を継続的にモニタリングおよびスキャンしています。継続的スキャンには、新しい関数がリポジトリにプッシュされたときの初回スキャンと、関数が更新されたときや新しい共通脆弱性識別子 (CVE) がリリースされたときの関数の自動再スキャンが含まれます。

- タグで除外済み — この関数はタグによるスキャンから除外されているため、Amazon Inspector はスキャンしていません。
- スキャンの適格性が失効しました — Amazon Inspector は、前回呼び出されたり更新されたりしてから 90 日以上が経過しているため、この関数をモニタリングしていません。
- 内部エラー — Amazon Inspector が関数をスキャンしようとしたときに内部エラーが発生しました。Amazon Inspector は自動的にエラーに対処し、できるだけ早くスキャンを再開します。
- 保留中の初期スキャン — Amazon Inspector は初期スキャンの関数をキューに入れました。
- サポートなし — Lambda 関数のランタイムはサポートされていません。

Organizations を使用した Amazon Inspector での複数のアカウントの管理

Amazon Inspector を使用して、AWS Organizations を通じて関連付けられている複数のアカウントを管理できます。複数の Amazon Inspector アカウントを管理するために、Organizations 管理アカウントは、組織内のアカウントを Amazon Inspector の委任管理者アカウントとして指定します。委任された管理者は、組織の Amazon Inspector を管理し、組織に代わってタスクを実行するための特別なアクセス許可が付与されます。これらのタスクには、メンバーアカウントのスキャンのアクティブ化または非アクティブ化、組織全体の集約された検出結果データの表示、抑制ルールの作成と管理が含まれます。

> **Note**
>
> 複数の の複数のアカウントに対して Amazon Inspector をプログラムで有効にするには AWS リージョン、Amazon Inspector によって開発されたシェルスクリプトを使用できます。このスクリプトの使用の詳細については、 GitHub ウェブサイトの「 Inspector2-enablement-with-cli」を参照してください。

トピック

- Amazon Inspector の管理者とメンバーアカウントの関係について理解する
- Amazon Inspector 用の委任された管理者の指定

Amazon Inspector の管理者とメンバーアカウントの関係について理解する

複数アカウント環境で Amazon Inspector を使用すると、Amazon Inspector の委任された管理者アカウントは特定のメタデータにアクセスできます。このメタデータには、Amazon EC2 と Amazon ECR の設定データと、メンバーアカウントのセキュリティ検出結果が含まれます。管理者アカウントは、メンバーアカウントに適用される検出結果抑制ルールを作成することもできます。詳細については、「抑制ルールによる Amazon Inspector で検出結果を抑制する」を参照してください。

委任管理者のアクション

通常、委任管理者が自分のアカウントに設定を適用すると、それらの設定は組織内の他のすべてのアカウントに適用されます。委任管理者は、自分のアカウントや関連するメンバーの情報を表示および取得することもできます。Amazon Inspector の委任管理者アカウントは、以下のアクションを実行できます。

- Amazon Inspector のアクティブ化や非アクティブ化など、関連するアカウントの Amazon Inspector のステータスを表示および管理できます。
- 組織内のすべてのメンバーアカウントのスキャンタイプをアクティブ化または非アクティブ化します。
- 組織全体の集約された検出結果データと、組織内のすべてのメンバーアカウントの検出結果の詳細が表示されます。
- 組織内のすべてのアカウントの検出結果に適用される抑制ルールを作成および管理します。
- 組織のすべてのメンバーに対して Amazon ECR 拡張スキャンをアクティブ化します。
- 組織全体のリソースカバレッジを表示します。
- 組織内のすべてのメンバーアカウントの ECR コンテナイメージを自動的に再スキャンする期間を定義します。委任された管理者のスキャン期間設定は、メンバーアカウントが以前に設定した設定よりも優先されます。組織内のすべてのアカウントは、委任された管理者の Amazon ECR 自動再スキャン期間を共有します。個々のアカウントに対して異なる再スキャン期間を設定することはできません。
- 組織内のすべてのアカウントで使用される Amazon EC2 の Amazon Inspector 詳細検査用のカスタムパスを 5 つ指定します。 Amazon EC2 これは、委任された管理者個々のアカウントに設定できる 5 つのカスタムパスに追加されます。詳細検査カスタムパスの設定の詳細については、「」を参照してくださいAmazon Inspector の詳細検査のカスタムパス。
- メンバーアカウントの Amazon Inspector 詳細検査を有効または無効にします。
- 組織内のあらゆるメンバーアカウントの SBOM をエクスポートします。
- 組織内のすべてのメンバーアカウントの Amazon EC2 スキャンモードを設定します。詳細については、「スキャンモードの管理」を参照してください。
- メンバーアカウントによって作成されたスキャン設定を除き、組織内のすべてのアカウントの CIS スキャン設定を作成および管理します。

> **Note**
>
> メンバーアカウントが組織を離れると、委任された管理者は、そのアカウントによってスケジュールされたスキャン設定を表示できなくなります。

- 組織内のすべてのアカウントの CIS スキャン結果を表示します。

メンバーアカウントのアクション

メンバーアカウントは Amazon Inspector で自分のアカウントに関する情報を表示および取得できますが、アカウントの設定は委任された管理者によって管理されます。組織内のメンバーアカウントは Amazon Inspector で以下のアクションを実行できます。

- メンバー自身のアカウントで Amazon Inspector をアクティベートします。
- メンバー自身のアカウントのリソースカバレッジを表示します。
- メンバー自身のアカウントの検出結果の詳細を表示します。
- メンバー自身のアカウントの ECR コンテナイメージ自動再スキャン期間設定を表示します。
- 個々のアカウントに使用される EC2 の Amazon Inspector 詳細検査のカスタムパスを 5 つ指定します。これらのパスは、委任管理者が組織に指定したカスタムパスに加えてスキャンされます。詳細検査パスの設定の詳細については、「」を参照してくださいAmazon Inspector の詳細検査のカスタムパス。
- Amazon Inspector の詳細検査用に委任管理者が設定したカスタムパスを表示します。
- アカウントに関連付けられているあらゆるリソースの SBOM をエクスポートします。
- アカウントのスキャンモードを表示します。
- アカウントの CIS スキャン設定を作成および管理します。
- 委任された管理者によってスケジュールされたリソースを含め、アカウント内のリソースの CIS スキャンの結果を表示します。

> **Note**
>
> アクティベーション後、Amazon Inspector は委任された管理者アカウントでのみ非アクティブ化できます。

Amazon Inspector 用の委任された管理者の指定

委任された管理者のための重要な考慮事項

Amazon Inspector の委任された管理者が行う操作方法を定義する、次の要素に注意してください。

委任された管理者は、最大 5,000 のメンバーを管理することができます。

Amazon Inspector の委任された管理者には、それぞれ 5,000 件のメンバーアカウントが割り当てられています。ただし、組織内に 5,000 を超えるアカウントが存在する可能性があります。メンバーアカウントが 5,000 を超えると、Amazon CloudWatch Personal Health Dashboard を通じて通知が送信され、委任された管理者アカウントに E メールが送信されます。

委任された管理者はリージョンレベルです。

とは異なり AWS Organizations、Amazon Inspector はリージョナルサービスです。つまり、委任された管理者を指定し、メンバーアカウントを追加し、Amazon Inspector を使用する各 でスキャンタイプ AWS リージョン をアクティブ化する必要があります。

組織は、委任された管理者を 1 名だけ持つことができます。

Amazon Inspector の委任された管理者は、1 つの組織につき 1 名のみです。あるリージョンでアカウントを委任管理者として指定した場合、そのアカウントは他のすべてのリージョンで委任管理者である必要があります。

委任された管理者を変更しても、メンバーアカウントの Amazon Inspector は非アクティブ化になりません。

委任された管理者を削除しても、Amazon Inspector はそれらのアカウントで非アクティブ化されず、スキャン設定も影響を受けません。

Organization では、すべての機能がアクティブ化されている AWS 必要があります。

これは のデフォルト設定です AWS Organizations。アクティブ化されていない場合は、「組織内のすべての機能のアクティブ化」を参照してください。

委任された管理者の指定に必要な許可

Amazon Inspector をアクティブ化し、Amazon Inspector に委任された管理者を指定するアクセス許可が必要です。

IAM ポリシーの最後に次のステートメントを追加することで、これらの許可を付与します。

```
{
    "Sid": "PermissionsForInspectorAdmin",
    "Effect": "Allow",
    "Action": [
        "inspector2:EnableDelegatedAdminAccount",
        "organizations:EnableAWSServiceAccess",
        "organizations:RegisterDelegatedAdministrator",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:DescribeOrganizationalUnit",
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization"
    ],
    "Resource": "*"
}
```

AWS 組織の委任管理者の指定

次の手順では、 AWS 組織の委任管理者を指定する方法を示します。この指定が完了すると、組織管理アカウントと選択した委任された管理者アカウントの両方で Amazon Inspector がアクティブ化になります。

> **Note**
>
> 組織の管理者アカウントのみが、委任された管理者を指定できます。

Amazon Inspector を初めてアクティブ化すると、`AWSServiceRoleForAmazonInspector`アカウントのサービスにリンクされたロール (SLR) が作成されます。Amazon Inspector がサービスリンクロールを使用する方法の詳細については、「Amazon Inspector でのサービスにリンクされたロールの使用」を参照してください。一般的なサービスにリンクされたロールの詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの使用」を参照してください。

Amazon Inspector 用の委任された管理者の指定

Console

コンソールに委任された管理者を指定する

1. AWS Organizations 管理アカウント AWS Management Console を使用して にサインインします。
2. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開き、右上の AWS リージョン セレクターを使用して、管理者を指定するリージョンを指定します。
3. 委任管理者ペインで、組織の Amazon Inspector 委任管理者として AWS アカウント 指定する の 12 桁のアカウント ID を入力します。次に、管理の委任 を選択します。
4. (推奨) 各 AWS リージョン毎に前のステップを繰り返します。

API

API で委任された管理者を指定する

- Organizations 管理アカウントの AWS アカウント の認証情報を使用して EnableDelegatedAdminAccount API オペレーションを実行します。次の CLI コマンドを実行して、これを行う AWS Command Line Interface こともできますaws inspector2 enable-delegated-admin-account --delegated-admin-account-id 11111111111。

> **Note**
>
> Amazon Inspector の委任された管理者にするアカウントのアカウント ID を必ず指定してください。

委任された管理者を指定した後は、委任された管理者アカウントを変更または削除するためにのみ、管理アカウントを使用する必要があります AWS Organizations 。

メンバーアカウントの Amazon Inspector スキャンをアクティブ化する

組織の委任された管理者として、 AWS Organizations 管理アカウントに関連付けられているすべてのメンバーに対して Amazon EC2 スキャン、Amazon ECR スキャン、またはその両方をアクティブ化できます。メンバーアカウントのスキャンをアクティブ化すると、そのアカウントは委任された管理者に関連付けられ、Amazon Inspector が自動的にアクティブになり、選択したタイプのスキャン

がすぐに開始されます。スキャンできるリソースとスキャンの設定方法については、「」を参照してくださいAmazon Inspector による自動リソーススキャン。

Amazon Inspector には、メンバーアカウントのスキャンを管理およびアクティブ化するためのオプションがいくつか用意されています。これには、メンバーアカウントが Amazon Inspector をアクティブ化できるようにするオプションが含まれます。次のいずれかのオプションを使用して、メンバーアカウントがスキャンを開始できます。

すべてのメンバーアカウントのスキャンを自動的にアクティブ化するには

1. 委任管理者アカウントにサインインします。
2. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。次に、 AWS リージョン 右上の セレクターを使用して、すべてのメンバーアカウントのスキャンをアクティブ化するリージョンを指定します。
3. ナビゲーションペインで [設定] の [アカウント管理] を選択します。アカウントテーブルには、AWS Organizations 管理アカウントに関連付けられているすべてのメンバーアカウントが表示されます。
4. テーブルの上部にあるチェックボックスをオンにし、そのページのすべてのアカウントを選択します。次に [アクティブ化] を選択し、メニューから希望するスキャンタイプオプションを選択します。

   > **Note**
   >
   > ページに現在表示されているアカウントのみが選択されます。アカウントのページが複数ある場合は、各ページでこのプロセスを繰り返す必要があります。ページに表示されるアカウント数を変更するには、歯車アイコンを選択します。

5. 新しいメンバーアカウントの Inspector を自動的にアクティブ化設定をオンにし、スキャンタイプを選択して、組織に追加された新しいメンバーをアクティブ化します。
6. (推奨) メンバーアカウントをスキャンするリージョンごとに、これらのステップを繰り返します。

[新しいメンバーアカウントの Inspector を自動的にアクティブ化] 設定により、組織の今後のメンバー全員に対して Amazon Inspector がアクティブ化されます。これにより、Amazon Inspector の委任された管理者は、組織内で作成された、または組織に追加された新しいメンバーを管理できます。メンバーアカウントの数が 5,000 のクォータに達すると、この設定は自動的にオフになります。ア

カウントが削除され、メンバーの総数が 5,000 未満になると、設定は自動的に再度アクティブになります。

メンバーアカウントを選択的にアクティブ化するには

1. 委任管理者アカウントにサインインします。
2. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開き、右上の セレクターを使用して AWS リージョン 、特定のメンバーアカウントのスキャンをアクティブ化するリージョンを指定します。
3. ナビゲーションペインで [設定] の [アカウント管理] を選択します。アカウントテーブルには、AWS Organizations 管理アカウントに関連付けられているすべてのメンバーアカウントが表示されます。
4. [アカウント管理] ページで、スキャンをアクティブ化するメンバーアカウントのチェックボックスをそれぞれオンにします。
5. [アクティブ化] を選択します。
6. [アクティブ化] メニューから、選択したアカウントに対してアクティブ化するスキャンタイプを選択します。次のスキャンオプションから選択できます。
   - すべてのスキャン — すべてのスキャンタイプをアクティブ化します。
   - EC2 スキャン — Amazon EC2 インスタンスのスキャンをアクティブ化します。
   - ECR コンテナスキャン — ECR コンテナイメージのスキャンをアクティブ化します。
   - AWS Lambda 標準スキャン — Lambda 関数のスキャンをアクティブ化します。
7. （推奨) 特定のメンバーのスキャンをアクティブ化するリージョンごとに、これらのステップを繰り返します。

AWS Organizations 管理アカウントが Amazon Inspector の管理者を委任している場合は、自分のアカウントをメンバーとしてアクティブ化し、自分のアカウントのスキャンの詳細を表示できます。

メンバーアカウントとしてスキャンをアクティブ化するには

1. アカウントにログインします。
2. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開き、右上の セレクターを使用して AWS リージョン 、スキャンをアクティブ化するリージョンを指定します。
3. ナビゲーションペインで [設定] の [アカウント管理] を選択します。

4. [アカウント管理] ページで、アカウントのチェックボックスをオンにします。
5. [アクティブ化] メニューから、アクティブ化するスキャンタイプを選択します。次のsキャンオプションから選択できます。

   - すべてのスキャン — すべてのスキャンタイプをアクティブ化します。
   - EC2 スキャン — Amazon EC2 インスタンスのスキャンをアクティブ化します。
   - ECR コンテナスキャン — ECR コンテナイメージのスキャンをアクティブ化します。
   - AWS Lambda 標準スキャン — Lambda 関数のスキャンをアクティブ化します。

6. （推奨) スキャンをアクティブ化するリージョンごとに、これらのステップを繰り返します。

Amazon Inspector でのメンバーアカウントの関連付けを解除する

以下の手順は、メンバーアカウントの関連付けを解除する方法を示しています。関連付けが解除されたメンバーアカウントは、スタンドアロンの Amazon Inspector アカウントとして AWS Organizations 組織内に残ります。Amazon Inspector の委任された管理者には、これらのアカウントの Amazon Inspector をアクティブ化および管理するためのアクセス許可がなくなりました。関連付けが解除されたアカウントは、後でメンバーとして再度追加できます。

> **Note**
>
> アカウントの関連付けを解除しても、そのアカウントの Amazon Inspector スキャンは非アクティブ化されません。

Console

コンソールを使用して、メンバーアカウントの関連付けを解除するには

1. 委任された管理者のアカウントにログイン
2. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開き、右上の AWS リージョン セレクターを使用して、1 つ以上のメンバーアカウントの関連付けを解除するリージョンを指定します。
3. ナビゲーションペインで [設定] の [アカウント管理] を選択します。
4. [アカウント管理] ページで、関連付けを解除する各アカウントのチェックボックスをオンにします。
5. [アクション] メニューから、[アカウントの関連付けを解除する] を選択します。

6. (推奨) アカウントの関連付けを解除するリージョンごとに、これらのステップを繰り返します。

API

API を使用してメンバーアカウントの関連付けを解除するには

API オペレーション DisassociateMember を実行します。リクエストで、関連付けを解除するアカウント IDs を指定します。

Amazon Inspector 委任された管理者の削除

新しい Amazon Inspector の委任された管理者を割り当てる必要がある場合は、既存の委任された管理者 AWS Organizations を管理アカウントとして削除できます。

委任された管理者を削除しても、そのアカウントまたは組織メンバーアカウントで Amazon Inspector は非アクティブ化されません。組織内のアカウントはスタンドアロンアカウントに変換され、委任された管理者によって管理される前のスキャン設定を保持します。

委任された管理者を削除するには

1. AWS Organizations 管理アカウント AWS Management Console を使用して にログインします。
2. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開き、右上の セレクターを使用して AWS リージョン 、委任された管理者を削除するリージョンを指定します。
3. ナビゲーションペインで [設定] の [アカウント管理] を選択します。
4. [委任された管理者] セクションで [削除] を選択し、アクションを確認します。
5. この委任された管理者を登録した各リージョンで、これらの手順を繰り返します。

新しい Amazon Inspector 委任管理者を追加するときは、組織メンバーを新しい管理者アカウントに手動で関連付ける必要があります。組織メンバーを新しい管理者アカウントに関連付けるには、次の手順を実行します。

メンバーを新しい委任された管理者と関連付けるには

1. 委任管理者アカウント AWS Management Console を使用して にログインします。

2. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開き、右上の セレクターを使用して AWS リージョン 、メンバーを新しい委任管理者に関連付けるリージョンを指定します。
3. ナビゲーションペインで [設定] の [アカウント管理] を選択します。
4. 上部のチェックボックスを使用して、組織内のリストされているすべてのアカウントを選択します。
5. [アクション] メニューから、[メンバーを追加] を選択します。
6. メンバーを新しい委任された管理者に関連付けるリージョンごとに、これらのステップを繰り返します。

Amazon Inspector での使用状況とコストのモニタリング

Amazon Inspector コンソールと API オペレーションを使用して、ご使用の環境で Amazon Inspector を使用する場合の月額料金を予測できます。マルチアカウント環境の Amazon Inspector 管理者であれば、環境全体の総コストと、各メンバーアカウントのコストメトリクスを表示できます。

コンソール使用状況を使用する

Amazon Inspector の使用状況と予測コストをコンソールから評価できます。

使用統計にアクセスするには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、コストをモニタリングするリージョンを選択します。
3. ナビゲーションペインで 使用状況を選択します。

[アカウント別] タブには、[アカウント使用量] に表示されている 30 日間の期間に基づく予測総コストが表示されます。[予測コスト] 列の表で値を選択すると、そのアカウントのスキャンタイプ別の使用量の内訳が表示されます。この詳細ウィンドウでは、そのアカウントで無料トライアルがアクティブになっているスキャンタイプも確認できます。

あなたが組織の委任された管理者である場合は、組織内の各アカウントの表に 1 行が表示されます。組織内のアカウントの関連付けが解除されると、コンソールにはそのアカウントの予測コストが - として表示されます。

[スキャンタイプ別] タブには、現在の 30 日間の実際の使用量の内訳がスキャンのタイプごとに表示されます。この情報は、[アカウント別] タブの予測コストの計算に使用されます。

組織の委任された管理者の場合は、組織内の各アカウントの使用状況を確認できます。

このタブでは、以下のいずれかのペインを展開して使用状況統計を表示できます。

Amazon EC2 スキャン

Amazon Inspector 使用状況コンソールは、エージェントベースのスキャンとエージェントレススキャンの次のメトリクスを追跡します。

- インスタンス (Avg) — Amazon Inspector は、カバレッジ時間を使用して EC2 インスタンススキャンの平均リソース数を計算します。平均は、合計カバレッジ時間を 720 時間 (30 日間の時間数) で割ったものです。
- カバレッジ時間 — Amazon EC2 スキャンの場合、Amazon Inspector がアカウント内の各 EC2 インスタンスに対してアクティブカバレッジを提供した過去 30 日間の合計時間数です。EC2 インスタンスの場合、カバレッジ時間とは、Amazon Inspector がインスタンスを発見してから、インスタンスが終了または停止されるまで、またはタグによってスキャンから除外されるまでの時間です (停止したインスタンスを再起動するか、除外タグを削除すると、Amazon Inspector カバレッジを再開し、そのインスタンスのカバレッジ時間は引き続き加算されます)。

CIS インスタンススキャン — アカウント内のインスタンスに対して実行された CIS スキャンの合計数。

Amazon ECR スキャン

初回スキャン — 過去 30 日間にアカウント内のイメージを初めてスキャンした合計回数。

再スキャン — 過去 30 日以内にアカウント内のイメージを再スキャンした合計回数。再スキャンとは、Amazon Inspector が以前にスキャンした ECR イメージに対して実行されるスキャンです。ECR リポジトリを継続的スキャン用に設定している場合、Amazon Inspector がデータベースに新しい共通脆弱性識別子 (CVE) を追加すると、再スキャンが自動的に行われます。

Lambda スキャン

Amazon Inspector 使用状況コンソールは、Lambda 標準スキャンと Lambda コードスキャンの次のメトリクスを追跡します。

- Lambda 関数の数 (平均) — Amazon Inspector はカバレッジ時間を使用して、Lambda 関数スキャンの平均関数数を計算します。平均は、カバレッジ時間の合計を 720 時間 (30 日間の時間数) で割ったものです。
- カバレッジ時間 — Lambda 関数スキャンの場合、Amazon Inspector がアカウント内の各 Lambda 関数に対してアクティブカバレッジを提供した過去 30 日間の合計時間数です。 AWS Lambda 関数については、Amazon Inspector が関数を検出した時点から、その関数が削除されるか、スキャンから除外されるまでの期間で、対応時間が計算されます。除外された関数が再び含まれた場合でも、その関数のカバレッジ時間は引き続き加算されます。

Amazon Inspector での使用コストの計算方法について

Amazon Inspector が提供するコストは実際のコストではなく見積もりであるため、 AWS Billing コンソールのコストとは異なる場合があります。

Amazon Inspector が「料金表」ページでコストを計算する方法について、次の点に注意してください。

- 使用コストには現在のリージョンのみが反映されます。スキャンタイプごとの料金は AWS リージョンによって異なります。リージョンごとの正確な料金を確認するには、Amazon Inspector の料金」を参照してください。
- 予測される使用量はすべて、最も近い米ドルに四捨五入されています。
- 割引は、予測コストには含まれません。
- 予測コストは、スキャンタイプごとの 30 日間の使用期間の合計料金です。アカウントの使用日数が 30 日未満の場合、Amazon Inspector は、現在対象となっているリソースが 30 日間の残りの期間も引き続き適用されるものとして、30 日後のコストを予測します。
- スキャンタイプごとのコストは、以下に基づいて計算されます。
  - EC2 スキャン: コストは、過去 30 日間に Amazon Inspector の対象となった EC2 インスタンスの平均数を反映しています。
  - ECR コンテナスキャン: コストは、過去 30 日間の初回イメージスキャンとイメージ再スキャンの合計が反映されます。
  - Lambda 標準スキャン: コストは、過去 30 日間に Amazon Inspector の対象となった Lambda 関数の平均数を反映しています。
  - Lambda コードスキャン: コストは、過去 30 日間に Amazon Inspector の対象となった Lambda 関数の平均数を反映しています。

Amazon Inspector の無料トライアルについて

Amazon Inspector のスキャンタイプをアクティブ化すると、そのスキャンタイプの 15 日間の無料トライアルに自動的に登録されます。各スキャンの種類には、EC2 スキャン、ECR スキャン、Lambda 標準スキャン、Lambda コードスキャンなど、独立した無料トライアルがあります。

> **Note**
>
> 無料トライアルは CIS スキャンには適用されません。

無料トライアル中にスキャンタイプを無効にすると、そのスキャンタイプの無料トライアルは一時停止されます。そのサービスを再開すると、無料トライアルが再開され、その無料トライアルの残りの日数が利用できるようになります。

Amazon Inspector のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、 AWS とユーザーの間で共有される責任です。責任共有モデルではこれを、クラウドのセキュリティ、およびクラウド内でのセキュリティと説明しています:

- クラウドのセキュリティ — AWS は、 で AWS サービスを実行するインフラストラクチャを保護する責任を担います AWS クラウド。また、 は、お客様が安全に使用できるサービス AWS も提供します。コンプライアンスAWS プログラムコンプライアンスプログラム の一環として、サードパーティーの監査者は定期的にセキュリティの有効性をテストおよび検証。Amazon Inspector に適用されるコンプライアンスプログラムの詳細については、「コンプライアンスプログラムAWS による対象範囲内のサービスコンプライアンスプログラム」を参照してください。
- クラウドのセキュリティ — お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、Amazon Inspector の使用時に責任共有モデルがどのように適用されるかを理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンス上の目的を達成するように Amazon Inspector を設定する方法について説明します。また、Amazon Inspector リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

トピック

Amazon Inspector におけるデータ保護

責任 AWS 共有モデル、Amazon Inspector でのデータ保護に適用されます。このモデルで説明されているように、 AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、使用する AWS のサービス のセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された記事「AWS 責任共有モデルおよび GDPR」を参照してください。

データ保護の目的で、 認証情報を保護し AWS アカウント 、 AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 は必須であり TLS 1.3 がお勧めです。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。
- AWS 暗号化ソリューションと、 内のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API AWS を介して にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキストフィールドに配置しないことを強くお勧めします。これは、コンソール、API、または SDK を使用して Amazon Inspector AWS CLIまたは他の AWS のサービス を使用する場合も同様です。 AWS SDKs 名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。外部サーバーへの URL を提供する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないように強くお勧めします。

トピック

- 保管中の暗号化

- 転送中の暗号化

保管中の暗号化

Amazon Inspector は、デフォルトで AWS 暗号化ソリューションを使用して保管中のデータを安全に保存します。Amazon Inspector は、 AWS Systems Manager を使用して収集されたリソースインベントリ、Amazon ECR イメージから解析されたリソースインベントリ、生成されたセキュリティ検出結果などのデータを、 AWS Key Management Service () の AWS 所有の暗号化キーを使用して暗号化しますAWS KMS。 AWS 所有キーを表示、管理、使用したり、その使用を監査したりすることはできません。ただし、データを暗号化するキーを保護するために何らかの操作を行ったり、プログラムを変更したりする必要はありません。詳細については、「AWS 所有キー」を参照してください。

Amazon Inspector を無効にすると、収集したインベントリやセキュリティ検出結果など、Amazon Inspector が保存または管理するすべてのリソースが永久に削除されます。

検出結果のコードの保管時の暗号化

Amazon Inspector Lambda コードスキャンの場合、Amazon Inspector は と提携 CodeGuru してコードの脆弱性をスキャンします。脆弱性が検出されると、脆弱性を含むコードのスニペットが CodeGuru 抽出され、Amazon Inspector がアクセスをリクエストするまでそのコードが保存されます。デフォルトでは、 AWS 所有キー CodeGuru を使用して抽出されたコードを暗号化しますが、暗号化に独自のカスタマーマネージド AWS KMS キーを使用するように Amazon Inspector を設定できます。

次のワークフローでは、Amazon Inspector が、設定したキーを使用してコードを暗号化する方法を説明しています。

1. Amazon Inspector UpdateEncryptionKey API を使用して Amazon Inspector に AWS KMS キーを指定します。
2. Amazon Inspector は、 AWS KMS キーに関する情報を に転送します CodeGuru。 は、将来の使用のために情報 CodeGuru を保存します。
3. CodeGuru は、Amazon Inspector で設定したキー AWS KMS の から許可をリクエストします。
4. CodeGuru は、キーから暗号化されたデータ AWS KMS キーを作成し、保存します。このデータキーは、 によって保存されているコードデータを暗号化するために使用されます CodeGuru。
5. Amazon Inspector がコードスキャンからデータをリクエストするたびに、 はグラント CodeGuru を使用して暗号化されたデータキーを復号します。その後、 はそのキーを使用してデータを復号し、取得できるようにします。

Lambda コードスキャンを無効にすると、グラントが CodeGuru 廃止され、関連するデータキーが削除されます。

カスタマーマネージドキーによるコード暗号化のアクセス許可

暗号化を使用するには、 AWS KMS アクションへのアクセスを許可するポリシーと、Amazon Inspector にそれらのアクションを条件キーで使用するための CodeGuru アクセス許可を付与するステートメントが必要です。

アカウントの暗号化キーを設定、更新、またはリセットする場合は、AWS マネージドポリシー: AmazonInspector2FullAccess などの Amazon Inspector 管理者ポリシーを使用する必要があります。また、暗号化対象として選択したキーに関する検出結果またはデータからコードス二ペットを取得する必要がある読み取り専用ユーザーには、次のアクセス許可を付与する必要があります。

KMS の場合、ポリシーは以下のアクションを実行できるようにする必要があります。

- `kms:CreateGrant`
- `kms:Decrypt`
- `kms:DescribeKey`
- `kms:GenerateDataKeyWithoutPlainText`
- `kms:Encrypt`
- `kms:RetireGrant`

ポリシーに正しい AWS KMS アクセス許可があることを確認したら、Amazon Inspector と が暗号化にキーを使用できるようにするステートメント CodeGuru をアタッチする必要があります。以下のポリシーステートメントをアタッチします。

> **Note**
>
> Region を、Amazon Inspector Lambda コードスキャンが有効になっている AWS リージョンに置き換えます。

```
{
            "Sid": "allow CodeGuru Security to request a grant for a AWS KMS key",
        "Effect": "Allow",
        "Action": "kms:CreateGrant",
```

```
        "Resource": "*",
        "Condition": {
         "ForAllValues:StringEquals": {
          "kms:GrantOperations": [
           "GenerateDataKey",
           "GenerateDataKeyWithoutPlaintext",
           "Encrypt",
           "Decrypt",
           "RetireGrant",
           "DescribeKey"
          ]
         },
         "StringEquals": {
          "kms:ViaService": [
           "codeguru-security.Region.amazonaws.com"
          ]
         }
        }
       },
       {
        "Sid": "allow Amazon Inspector and CodeGuru Security to use your AWS KMS key",
        "Effect": "Allow",
        "Action": [
         "kms:Encrypt",
         "kms:Decrypt",
         "kms:RetireGrant",
         "kms:DescribeKey",
         "kms:GenerateDataKeyWithoutPlaintext"
        ],
        "Resource": "*",
        "Condition": {
         "StringEquals": {
          "kms:ViaService": [
           "inspector2.Region.amazonaws.com",
           "codeguru-security.Region.amazonaws.com"
          ]
         }
        }
       }
```

> **Note**
>
> ステートメントをポリシーに追加するときに、構文が有効であることを確認します。ポリシーは JSON 形式を使用します。これは、ステートメントをポリシーに追加する場所に応じて、ステートメントの前後にカンマを追加する必要があることを意味します。ステートメントを最後のステートメントとして追加する場合は、前のステートメントの右中括弧の後にカンマを追加します。最初のステートメントとして追加するか、既存の 2 つのステートメントの間に追加する場合は、右中括弧の後にカンマを追加します。

カスタマーマネージドキーによる暗号化の設定

カスタマーマネージドキーを使用してアカウントの暗号化を設定するには、カスタマーマネージドキーによるコード暗号化のアクセス許可 で説明されているアクセス許可を持つ Amazon Inspector 管理者である必要があります。さらに、検出結果と同じ AWS リージョンに AWS KMS キー 、またはマルチリージョンキー が必要になります。アカウント内の既存の対称キーを使用するか、 AWS マネジメントコンソール、または AWS KMS APIs を使用して対称カスタマーマネージドキーを作成できます。詳細については、 ユーザーガイドの「対称暗号化 AWS KMS キーの作成 AWS KMS 」を参照してください。

Amazon Inspector APL を使用して暗号化を設定する

Amazon Inspector 管理者としてサインインしているときに Amazon Inspector API の Amazon InspectorUpdateEncryptionKeyオペレーションを暗号化するためのキーを設定するには。API リクエストで、 `kmsKeyId`フィールドを使用して、使用する AWS KMS キーの ARN を指定します。`scanType` に `CODE` を、`resourceType` に `AWS_LAMBDA_FUNCTION` を入力します。

UpdateEncryptionKey API を使用して、Amazon Inspector が暗号化に使用している AWS KMS キーを確認できます。

> **Note**
>
> カスタマーマネージドキーを設定していない`GetEncryptionKey`ときに を使用しようとすると、オペレーションは`ResourceNotFoundException`エラーを返します。これは、AWS 所有キーが暗号化に使用されていることを意味します。

または キーを削除したり、Amazon Inspector へのアクセスを拒否するポリシーを変更 CodeGuru したりすると、コードの脆弱性の検出結果にアクセスできなくなり、Lambda コードのスキャンがアカウントで失敗します。

`ResetEncryptionKey` を使用して、Amazon Inspector の検出結果の一部として抽出されたコードを暗号化するために、 AWS 所有キーの使用を再開できます。

転送中の暗号化

AWS は、 AWS 内部システムと他の AWS サービス間で転送されるすべてのデータを暗号化します。

インベントリ収集の場合、Systems Manager は、評価のために Transport Layer Security (TLS) で保護されたチャネル AWS を介して に返送する顧客所有の EC2 インスタンスからテレメトリデータを収集します。SSM が転送中のデータを暗号化する方法については、「Systems Manager でのデータ保護」を参照してください。

同様に、Security Hub に送信される Amazon ECR および AWS Lambda 関数スキャンの検出結果は、TLS で保護されたチャネルを使用して暗号化されます。

Amazon Inspector のための Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰を認証 (サインイン) し、誰に Amazon Inspector リソースの使用を承認 (許可を付与) するかを制御します。IAM は、追加料金なしで AWS のサービス 使用できる です。

トピック

- 対象者
- アイデンティティを使用した認証
- ポリシーを使用したアクセスの管理
- Amazon Inspector と IAM の連携
- Amazon Inspector アイデンティティベースのポリシーの例
- AWS Amazon Inspector の マネージドポリシー
- Amazon Inspector でのサービスにリンクされたロールの使用

- Amazon Inspector アイデンティティとアクセスのトラブルシューティング

対象者

AWS Identity and Access Management (IAM) の使用方法は、Amazon Inspector で行う作業によって異なります。

サービスユーザー – ジョブを実行するために Amazon Inspector サービスを使用する場合は、管理者から必要な認証情報と許可が与えられます。さらに多くの Amazon Inspector 機能を使用して作業を行う場合は、追加の許可が必要になることがあります。アクセスの管理方法を理解しておくと、管理者に適切な許可をリクエストするうえで役立ちます。Amazon Inspector の機能 にアクセスできない場合は、「Amazon Inspector アイデンティティとアクセスのトラブルシューティング」を参照してください。

サービス管理者 – 社内の Amazon Inspector リソースを担当している場合は、Amazon Inspector に対する完全なアクセス権があると思われます。サービスのユーザーがどの Amazon Inspector の機能やリソースにアクセスするかを決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。会社で Amazon Inspector と IAM を併用する方法の詳細については、「Amazon Inspector と IAM の連携」を参照してください。

IAM 管理者 – IAM 管理者は、Amazon Inspector へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる Amazon Inspector アイデンティティベースのポリシーの例を表示するには、「Amazon Inspector アイデンティティベースのポリシーの例」を参照してください。

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして AWS アカウントのルートユーザー、または IAM ロールを引き受けて認証（ にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。 AWS IAM Identity Center（ IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーティッド ID の例です。フェデレーティッドアイデンティティとしてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、 AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「 ユーザーガイド」の「 へのサインイン AWS アカウント方法AWS サインイン 」を参照してください。

AWS プログラムで にアクセスする場合、 は Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストに暗号で署名します。 AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用してリクエストを自分で署名する方法の詳細については、IAM ユーザーガイドの API AWS リクエストの署名を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、 AWS では、多要素認証 (MFA) を使用してアカウントのセキュリティを向上させることをお勧めします。詳細については、『AWS IAM Identity Center ユーザーガイド』の「Multi-factor authentication」(多要素認証) および『IAM ユーザーガイド』の「AWSにおける多要素認証 (MFA) の使用」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての およびリソースへの AWS のサービス 完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、『IAM ユーザーガイド』の「ルートユーザー認証情報が必要なタスク」を参照してください。

フェデレーティッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、一時的な認証情報を使用して にアクセスするための ID プロバイダーとのフェデレーションの使用を要求 AWS のサービス します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、Identity Center ディレクトリのユーザー、または ID ソースを通じて提供された認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーティッド ID が にアクセスすると AWS アカウント、ロールが引き受けられ、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、 AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成することも、独自の ID ソース内のユーザーとグループのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用することもできます。IAM Identity Center の詳細については、『AWS IAM Identity Center ユーザーガイド』の「What is IAM Identity Center?」(IAM Identity Center とは) を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、IAM ユーザーガイドの「長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする」を参照してください。

IAM グループは、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdmins という名前のグループを設定して、そのグループに IAM リソースを管理する権限を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が提供されます。詳細については、『IAM ユーザーガイド』の「IAM ユーザー (ロールではなく) の作成が適している場合」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。ロール を切り替える AWS Management Console ことで、 で IAM ロールを一時的に引き受けることができます。ロールを引き受けるには、 または AWS API AWS CLI オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「IAM ロールの使用」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます:

- フェデレーションユーザーアクセス – フェデレーティッドアイデンティティに権限を割り当てるには、ロールを作成してそのロールの権限を定義します。フェデレーティッドアイデンティティが認証されると、そのアイデンティティはロールに関連付けられ、ロールで定義されている権限が付与されます。フェデレーションの詳細については、『IAM ユーザーガイド』の「サードパーティーアイデンティティプロバイダー向けロールの作成」を参照してください。IAM アイデンティティセンターを使用する場合、権限セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。権限セットの詳細については、『AWS IAM Identity Center ユーザーガイド』の「権限セット」を参照してください。
- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、(ロールをプロキシとして使用する代わりに) ポリシーをリソースに直接アタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、『IAM ユーザーガイド』の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。
- クロスサービスアクセス — 一部の は、他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの権限、サービスロール、またはサービスにリンクされたロールを使用してこれを行う場合があります。
  - 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して でアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストリクエストリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
  - サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細

については、IAM ユーザーガイドの「 AWS のサービスにアクセス許可を委任するロールの作成」を参照してください。

- サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールの権限を表示できますが、編集することはできません。

- Amazon EC2 で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。 AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、『IAM ユーザーガイド』の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して権限を付与する」を参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、『IAM ユーザーガイド』の「(IAM ユーザーではなく) IAM ロールをいつ作成したら良いのか?」を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、 AWS ID またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義する のオブジェクトです。 は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) AWS がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの権限を定義します。例えば、`iam:GetRole` アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 AWS Management Console、、 AWS CLIまたは AWS API からロール情報を取得できます。

アイデンティティベースのポリシー

アイデンティティベースポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 権限ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、IAM ユーザーガイドの「IAM ポリシーの作成」を参照してください。

アイデンティティベースポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、 内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、 AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、『IAM ユーザーガイド』の「マネージドポリシーとインラインポリシーの比較」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための権限を持つかをコントロールします。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、 AWS WAF、および Amazon VPC は、ACLs。ACL の詳細については、『Amazon Simple Storage Service デベロッパーガイド』の「アクセスコントロールリスト (ACL) の概要」を参照してください。

その他のポリシータイプ

AWS は、一般的ではない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- アクセス許可の境界 - 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によって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。許可の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティの許可の境界」を参照してください。
- サービスコントロールポリシー (SCPs) – SCPs は、 の組織または組織単位 (OU) に対する最大アクセス許可を指定する JSON ポリシーです AWS Organizations。 AWS Organizations は、 AWS アカウント ビジネスが所有する複数の をグループ化して一元管理するサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、『AWS Organizations ユーザーガイド』の「SCP の仕組み」を参照してください。
- セッションポリシー - 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の一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「IAM ユーザーガイド」の「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、IAM ユーザーガイドの「ポリシー評価ロジック」を参照してください。

Amazon Inspector と IAM の連携

IAM を使用して Amazon Inspector へのアクセスを管理する前に、Amazon Inspectorで使用できる IAM 機能について理解しておく必要があります。

Amazon Inspector で使用できる IAM の機能

IAM 機能	Amazon Inspector のサポート
アイデンティティベースのポリシー	Yes
リソースベースのポリシー	No
ポリシーアクション	Yes
ポリシーリソース	はい
ポリシー条件キー (サービス固有)	はい
ACL	No
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	Yes
プリンシパル権限	Yes
サービスロール	いいえ
サービスリンクロール	はい

Amazon Inspector およびその他の がほとんどの IAM 機能と AWS のサービス 連携する方法の概要を把握するには、「IAM ユーザーガイド」のAWS のサービス 「IAM と連携する 」を参照してください。

Amazon Inspector アイデンティティベースのポリシー

アイデンティティベースポリシーをサポートする	Yes

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 権限ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、『IAM ユーザーガイド』の「IAM ポリシーの作成」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素については、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

Amazon Inspector アイデンティティベースのポリシーの例

Amazon Inspector アイデンティティベースのポリシーの例は、「Amazon Inspector アイデンティティベースのポリシーの例」でご確認ください。

Amazon Inspector内のリソースベースのポリシー

リソースベースのポリシーのサポート	No

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、アカウント全体、または別のアカウントの IAM エンティティをリソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる にある場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリ

ンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーを追加する必要はありません。詳細については、『IAM ユーザーガイド』の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

Amazon Inspector のポリシーアクション

ポリシーアクションに対するサポート	はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

JSON ポリシーの`Action`要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない権限のみのアクションなど、いくつかの例外があります。また、ポリシーに複数アクションが必要なオペレーションもあります。これらの追加アクションは、 依存アクション と呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

Amazon Inspector アクションのリストを確認するには、「サービス認証リファレンス」の「Amazon Inspectorで定義されるアクション」を参照してください。

Amazon Inspector のポリシーアクションは、アクションの前にプレフィックスを使用します。

```
inspector2
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
      "inspector2:action1",
      "inspector2:action2"
         ]
```

Amazon Inspector アイデンティティベースのポリシーの例は、「Amazon Inspector アイデンティティベースのポリシーの例」を参照してください。

Amazon Inspector のポリシーリソース

ポリシーリソースに対するサポート	はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースにどのような条件でアクションを実行できるかということです。

Resource JSON ポリシー要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、Resource または NotResource要素を含める必要があります。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの権限と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

Amazon Inspectorリソースのタイプとその ARN のリストを確認するには、「サービス認可リファレンス」の「Amazon Inspector で定義されるリソース」を参照してください。どのアクションで各リソースの ARN を指定できるかについては、「Amazon Inspector で定義されるアクション」を参照してください。

Amazon Inspector アイデンティティベースのポリシーの例は、「Amazon Inspector アイデンティティベースのポリシーの例」を参照してください。

Amazon Inspector のポリシー条件キー

サービス固有のポリシー条件キーのサポート	はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの 条件演算子 を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素を指定するか、1 つの Condition 要素に複数のキーを指定すると、 AWS は AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、『IAM ユーザーガイド』の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「 グローバル条件コンテキストキー」を参照してください。

Amazon Inspectorでの条件キーの一覧については、「サービス認証リファレンス」の「Amazon Inspector の条件キー」を参照してください。どのアクションやリソースで条件キーを使用できるかについては、「Amazon Inspector で定義されるアクション」を参照してください。

Amazon Inspector アイデンティティベースのポリシーの例は、「Amazon Inspector アイデンティティベースのポリシーの例」でご確認ください。

Amazon Inspector の ACL

ACL のサポート	No

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための権限を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon Inspector と ABAC

ABAC (ポリシー内のタグ) のサポート	部分的

属性ベースのアクセスコントロール (ABAC) は、属性に基づいて権限を定義する認可戦略です。では AWS、これらの属性はタグ と呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初

の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合に操作を許可するように ABAC ポリシーを設計します。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値ははいです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、『IAM ユーザーガイド』の「ABAC とは?」を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、「IAM ユーザーガイド」の「属性に基づくアクセスコントロール (ABAC) を使用する」を参照してください。

Amazon Inspector での一時的な認証情報の使用

一時的な認証情報のサポート	はい

一部の は、一時的な認証情報を使用してサインインすると機能 AWS のサービス しません。一時的な認証情報 AWS のサービス を使用する などの詳細については、IAM ユーザーガイドのAWS のサービス「IAM と連携する 」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ロールへの切り替え (コンソール)」を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用して . AWS recommends にアクセスできます AWS。これは、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することを推奨しています。詳細については、「IAM の一時的セキュリティ認証情報」を参照してください。

Amazon Inspector のクロスサービスプリンシパル許可

フォワードアクセスセッション (FAS) をサポート	はい

IAM ユーザーまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストリクエストリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。

Amazon Inspector のサービスロール

サービスロールのサポート	いいえ

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの「 AWS のサービスにアクセス許可を委任するロールの作成」を参照してください。

> **Warning**
>
> サービスロールの許可を変更すると、Amazon Inspectorの機能が破損する可能性があります。Amazon Inspector が指示する場合以外は、サービスロールを編集しないでください。

Amazon Inspector でのサービスにリンクされたロール

サービスリンクロールのサポート	はい

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービ

スにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスリンクロールの権限を表示できますが、編集することはできません。

サービスリンクロールの作成または管理の詳細については、「IAM と提携するAWS のサービス」を参照してください。表の中から、[Service-linked role (サービスリンクロール)] 列に Yes と記載されたサービスを見つけます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[Yes] リンクを選択します。

Amazon Inspector アイデンティティベースのポリシーの例

デフォルトでは、ユーザーおよびロールにはAmazon Inspectorリソースを作成または変更する許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または AWS API を使用してタスクを実行することはできません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

これらサンプルの JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、『IAM ユーザーガイド』の「IAM ポリシーの作成」を参照してください。

Amazon Inspector が定義するアクションとリソースタイプ (リソースタイプごとの ARN の形式を含む) の詳細については、「サービス認証リファレンス」の「Amazon Inspector のアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーのベストプラクティス
- Amazon Inspector コンソールの使用
- 自分の権限の表示をユーザーに許可する
- すべての Amazon Inspector リソースへの読み取り専用アクセスを許可する
- すべての Amazon Inspector リソースへのフルアクセスを許可する

ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウント内で誰かが Amazon Inspectorリソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可を付与するには、多くの一般的なユースケースにアクセス許可を付与する AWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義して、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイドの「AWS マネージドポリシー」または「AWS ジョブ機能の管理ポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで権限を設定するときは、タスクの実行に必要な権限のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権権限とも呼ばれています。IAM を使用して権限を適用する方法の詳細については、『IAM ユーザーガイド』の「IAM でのポリシーと権限」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、 などの特定の を介してサービスアクションが使用される場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます AWS CloudFormation。詳細については、IAM ユーザーガイドの [IAM JSON policy elements: Condition] (IAM JSON ポリシー要素：条件) を参照してください。
- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、「IAM ユーザーガイド」の「IAM Access Analyzer ポリシーの検証」を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、「IAM ユーザーガイド」の「MFA 保護 API アクセスの設定」を参照してください。

IAM でのベストプラクティスの詳細については、『IAM ユーザーガイド』の「IAM でのセキュリティのベストプラクティス」を参照してください。

Amazon Inspector コンソールの使用

Amazon Inspector コンソールにアクセスするには、許可の最小限のセットが必要です。アクセス許可により、 AWS アカウントの Amazon Inspector リソースの詳細をリストおよび表示できます。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリ

シーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

ユーザーとロールが引き続き Amazon Inspector コンソールを使用できるようにするには、エンティティに Amazon Inspector *ConsoleAccess*または *ReadOnly* AWS 管理ポリシーもアタッチします。詳細については、「IAM ユーザーガイド」の「ユーザーへのアクセス許可の追加」を参照してください。

自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
```

```
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

すべての Amazon Inspector リソースへの読み取り専用アクセスを許可する

この例では、すべての Amazon Inspector リソースへの読み取り専用アクセスを許可するポリシーを示します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "inspector2:Describe*",
                "inspector2:Get*",
                "inspector2:BatchGet*",
                "inspector2:List*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "organizations:ListDelegatedAdministrators",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:DescribeOrganizationalUnit",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

すべての Amazon Inspector リソースへのフルアクセスを許可する

この例では、すべての Amazon Inspector リソースへのフルアクセスを許可するポリシーを示します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "inspector2:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "inspector2.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "organizations:EnableAWSServiceAccess",
                "organizations:RegisterDelegatedAdministrator",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:DescribeOrganizationalUnit",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS Amazon Inspector の マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、多くの一般的なユースケースにアクセス許可を付与するように設計されているため、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できます。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケース別にカスタマーマネージドポリシーを定義して、マネージドポリシーを絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS 管理ポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) が更新されます。 は、新しい AWS のサービス が起動されるか、既存のサービスで新しい API AWS オペレーションが使用可能になると、 AWS 管理ポリシーを更新する可能性が最も高くなります。

詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS マネージドポリシー: AmazonInspector2FullAccess

`AmazonInspector2FullAccess` ポリシーは IAM ID にアタッチできます。

このポリシーは、Amazon Inspector へのフルアクセスを許可する管理許可を付与します。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `inspector2` – Amazon Inspector 機能へのフルアクセスを許可します。
- `iam` — Amazon Inspector がサービスリンクロール `AmazonInspector2AgentlessServiceRole` を作成できるようにします。これは、Amazon Inspector が、Amazon EC2 インスタンスや Amazon ECR リポジトリ、コンテナイメージに関する情報の取得、VPC ネットワークの分析、組織に関連するアカウントの記述などの処理を実行するために必要です。詳細については、「Amazon Inspector でのサービスにリンクされたロールの使用」を参照してください。

- `organizations` — 管理者による AWS Organizationsの組織への Amazon Inspector の使用を許可します。で Amazon Inspector の信頼されたアクセスをアクティブ化すると AWS Organizations、委任された管理者アカウントのメンバーは、組織全体の設定を管理し、結果を表示できます。
- `codeguru-security` — 管理者が Amazon Inspector を使用して情報コードス二ペットを取得し、 CodeGuru Security に保存されているコードの暗号化設定を変更できるようにします。詳細については、「検出結果のコードの保管時の暗号化」を参照してください。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "inspector2:*",
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "codeguru-security:BatchGetFindings",
    "codeguru-security:GetAccountConfiguration",
    "codeguru-security:UpdateAccountConfiguration"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": "iam:CreateServiceLinkedRole",
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "iam:AWSServiceName": "inspector2.amazonaws.com"
    }
   }
  },
  {
   "Effect": "Allow",
   "Action": [
    "organizations:EnableAWSServiceAccess",
    "organizations:RegisterDelegatedAdministrator",
    "organizations:ListDelegatedAdministrators",
    "organizations:ListAWSServiceAccessForOrganization",
```

```
    "organizations:DescribeOrganizationalUnit",
    "organizations:DescribeAccount",
    "organizations:DescribeOrganization"
   ],
   "Resource": "*"
  }
 ]
}
```

AWS マネージドポリシー: AmazonInspector2ReadOnlyAccess

`AmazonInspector2ReadOnlyAccess` ポリシーは IAM ID にアタッチできます。

このポリシーは、Amazon Inspector への読み取り専用アクセスを可能にする許可を付与します。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `inspector2` – Amazon Inspector 機能への読み取り専用アクセスを許可します。
- `organizations` – 内の組織の Amazon Inspector カバレッジの詳細を表示 AWS Organizations できるようにします。
- `codeguru-security` — CodeGuru セキュリティからコードスニペットを取得できるようにします。また、 CodeGuru Security に保存されているコードの暗号化設定を表示することもできます。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "organizations:ListDelegatedAdministrators",
    "organizations:ListAWSServiceAccessForOrganization",
    "organizations:DescribeOrganizationalUnit",
    "organizations:DescribeAccount",
    "organizations:DescribeOrganization",
```

```
    "inspector2:BatchGet*",
    "inspector2:List*",
    "inspector2:Describe*",
    "inspector2:Get*",
    "inspector2:Search*",
    "codeguru-security:BatchGetFindings",
    "codeguru-security:GetAccountConfiguration"
   ],
   "Resource": "*"
  }
 ]
}
```

AWS マネージドポリシー: AmazonInspector2ManagedCisPolicy

IAM エンティティに AmazonInspector2ManagedCisPolicy ポリシーをアタッチできます。このポリシーは、インスタンスの CIS スキャンを実行するためのアクセス許可を Amazon EC2 インスタンスに付与するロールにアタッチする必要があります。IAM ロールを使用して、EC2 インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。 AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、「IAM ユーザーガイド」の「IAM ロールを使用して、Amazon EC2 インスタンスで実行されるアプリケーションにアクセス許可を付与する」を参照してください。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- inspector2 — CIS スキャンの実行に使用されるアクションへのアクセスを許可します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
            "inspector2:StartCisSession",
            "inspector2:StopCisSession",
            "inspector2:SendCisSessionTelemetry",
```

```
            "inspector2:SendCisSessionHealth"
            ],
            "Resource": "*",
        }
    ]
}
```

AWS マネージドポリシー: AmazonInspector2ServiceRolePolicy

IAM エンティティに AmazonInspector2ServiceRolePolicy ポリシーをアタッチすることはできません。このポリシーは、Amazon Inspector がユーザーに代わってアクションを実行することを許可するサービスリンクロールにアタッチされます。詳細については、「Amazon Inspector でのサービスにリンクされたロールの使用」を参照してください。

AWS 管理ポリシー: AmazonInspector2AgentlessServiceRolePolicy

IAM エンティティに AmazonInspector2AgentlessServiceRolePolicy ポリシーをアタッチすることはできません。このポリシーは、Amazon Inspector がユーザーに代わってアクションを実行することを許可するサービスリンクロールにアタッチされます。詳細については、「Amazon Inspector でのサービスにリンクされたロールの使用」を参照してください。

Amazon Inspector による AWS マネージドポリシーの更新

Amazon Inspector の AWS マネージドポリシーの更新に関する詳細を、このサービスがこれらの変更の追跡を開始した以降の分について表示します。このページへの変更に関する自動アラートを受け取るには、Amazon Inspector のドキュメント履歴ページで RSS フィードにサブスクライブしてください。

変更	説明	日付
AmazonInspector2ManagedCisPolicy – 新しいポリシー	Amazon Inspector は、インスタンスプロファイルの一部として使用して、インスタンスで CIS スキャンを許可できる新しい マネージドポリシーを追加しました。	2024 年 1 月 23 日

変更	説明	日付
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector は、Amazon Inspector がターゲットインスタンスで CIS スキャンを開始できるようにする新しいアクセス許可を追加しました。	2024 年 1 月 23 日
AmazonInspector2AgentlessServiceRolePolicy – 新しいポリシー	Amazon Inspector には、EC2 インスタンスのエージェントレススキャンを可能にする、サービスリンクロールの新しいポリシーが追加されました。	2023 年 11 月 27 日
AmazonInspector2ReadOnlyAccess – 既存のポリシーの更新	Amazon Inspector には、読み取り専用ユーザーがパッケージの脆弱性検出結果の脆弱性インテリジェンスの詳細を取得できる新しいアクセス許可が追加されました。	2023 年 9 月 22 日
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector には、Elastic Load Balancing ターゲットグループの一部である Amazon EC2 インスタンスのネットワーク設定をスキャンできるようにする新しいアクセス許可が追加されました。	2023 年 8 月 31 日

変更	説明	日付
AmazonInspector2ReadOnlyAccess – 既存のポリシーの更新	Amazon Inspector には、読み取り専用ユーザーがリソースのソフトウェア部品表 (SBOM) をエクスポートできる新しいアクセス許可が追加されました。	2023 年 6 月 29 日
AmazonInspector2ReadOnlyAccess – 既存のポリシーの更新	Amazon Inspector には、読み取り専用ユーザーが自分のアカウントの Lambda コードスキャン検出結果の暗号化設定の詳細を取得できるようにする新しいアクセス許可が追加されました。	2023 年 6 月 13 日
AmazonInspector2FullAccess – 既存のポリシーの更新	Amazon Inspector には、Lambda コードスキャンの検出結果に含まれるコードを暗号化するようにカスタマーマネージドキー KMS キーをユーザーが設定できる新しいアクセス許可が追加されました。	2023 年 6 月 13 日
AmazonInspector2ReadOnlyAccess – 既存のポリシーの更新	Amazon Inspector には、読み取り専用ユーザーが自分のアカウントの Lambda コードスキャンのステータスと検出結果の詳細を取得できる新しいアクセス許可が追加されました。	2023 年 5 月 2 日

変更	説明	日付
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector は、Lambda スキャンをアクティブ化するときに Amazon Inspector がアカウントに AWS CloudTrail サービスにリンクされたチャネルを作成できるようにする新しいアクセス許可を追加しました。これにより、Amazon Inspector はアカウント内の CloudTrail イベントをモニタリングできます。	2023 年 4 月 30 日
AmazonInspector2FullAccess – 既存のポリシーの更新	Amazon Inspector には、ユーザーが Lambda コードスキャンから得られたコード脆弱性の検出結果を取得できる新しいアクセス許可が追加されました。	2023 年 4 月 21 日
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector に、Amazon Inspector が Amazon EC2 の詳細検査用に定義したカスタムパスに関する情報を Amazon EC2 Systems Manager に送信できるようにする新しいアクセス許可が追加されました。Amazon EC2	2023 年 4 月 17 日

変更	説明	日付
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector は、Lambda スキャンをアクティブ化するときに Amazon Inspector がアカウントに AWS CloudTrail サービスにリンクされたチャネルを作成できるようにする新しいアクセス許可を追加しました。これにより、Amazon Inspector はアカウント内の CloudTrail イベントをモニタリングできます。	2023 年 4 月 30 日
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector は、Amazon Inspector が AWS Lambda 関数内のデベロッパーコードのスキャンをリクエストし、Amazon CodeGuru Security からスキャンデータを受信できるようにする新しいアクセス許可を追加しました。さらに、Amazon Inspector には、IAM ポリシーを確認するためのアクセス許可が追加されました。Amazon Inspector はこの情報を使用して Lambda 関数のコード脆弱性をスキャンします。	2023 年 2 月 28 日

変更	説明	日付
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector は、 AWS Lambda 関数が最後に呼び出された日時 CloudWatch に関する情報の取得を Amazon Inspector に許可する新しいステートメントを追加しました。Amazon Inspector はこの情報を使用して、過去 90 日間にアクティブだった環境内の Lambda 関数にスキャンの対象を絞ります。	2023 年 2 月 20 日
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector は、Amazon Inspector が各 AWS Lambda 関数に関連付けられている各レイヤーバージョンを含む関数に関する情報を取得できるようにする新しいステートメントを追加しました。Amazon Inspector はこの情報を使用して Lambda 関数のセキュリティ脆弱性をスキャンします。	2022 年 11 月 28 日

変更	説明	日付
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector には、Amazon Inspector が SSM 関連付け実行を記述できるようにする新しいアクションが追加されました。さらに、Amazon Inspector では、`AmazonInspector2` 所有の SSM ドキュメントとの SSM 関連付けを作成、更新、削除、および開始できるように、リソーススコープが追加されました。	2022 年 8 月 31 日
AmazonInspector2ServiceRolePolicy 既存のポリシーの更新	Amazon Inspector は、Amazon Inspector が他の AWS パーティションでソフトウェアインベントリを収集できるように、ポリシーのリソーススコープを更新しました。	2022 年 8 月 12 日
AmazonInspector2ServiceRolePolicy – 既存のポリシーの更新	Amazon Inspector はアクションのリソーススコープを再構築し、Amazon Inspector が SSM 関連付けを作成、削除、および更新できるようにしました。	2022 年 8 月 10 日
AmazonInspector2ReadOnlyAccess – 新しいポリシー	Amazon Inspector には、機能への読み取り専用アクセスを許可する新しいポリシーが追加されました。	2022 年 1 月 21 日

変更	説明	日付
AmazonInspector2FullAccess – 新しいポリシー	Amazon Inspector には、機能へのフルアクセスを許可する新しいポリシーが追加されました。	2021 年 11 月 29 日
AmazonInspector2ServiceRolePolicy – 新しいポリシー	Amazon Inspector には、Amazon Inspector がお客様に代わって他のサービスでアクションを実行できるようになりました。	2021 年 11 月 29 日
Amazon Inspector が変更の追跡を開始しました。	Amazon Inspector が AWS マネージドポリシーの変更の追跡を開始しました。	2021 年 11 月 29 日

Amazon Inspector でのサービスにリンクされたロールの使用

Amazon Inspector は、 という名前の AWS Identity and Access Management (IAM) サービスにリンクされたロールを使用しますAWSServiceRoleForAmazonInspector2。このサービスリンクロールは、Amazon Inspector に直接リンクされた IAM ロールです。これは Amazon Inspector によって事前定義されており、Amazon Inspector がユーザーに代わって他の を呼び出すために必要なすべてのアクセス許可が含まれています。 AWS のサービス

必要な許可を手動で追加する必要がないため、サービスリンクロールは Amazon Inspector のセットアップを容易にします。Amazon Inspector はサービスリンクロールのアクセス許可を定義し、他に定義されていない限り、Amazon Inspector のみがそのロールを引き受けることができます。定義したアクセス許可には、信頼ポリシーと許可ポリシーが含まれます。この許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスにリンクされたロールの作成、編集、削除を IAM エンティティ (グループまたはロールなど) に許可するには、アクセス許可を設定する必要があります。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールのアクセス許可」を参照してください。サービスリンクロールを削除するには、その関連リソースを削除します。これにより、リソースへの意図しないアクセスによる許可の削除が防止され、 Amazon Inspector リソースは保護されます。

サービスリンクロールをサポートする他のサービスについては、「IAM と連動するAWS のサービス」を参照し、[Service-linked roles] (サービスリンクロール) の列内で [Yes] (はい) と表記されたサービスを確認してください。サービスにリンクされたロールに関するドキュメントをサービスで確認するには、[はい] リンクを選択します。

Amazon Inspector のサービスにリンクされたロールの許可

Amazon Inspector では、`AWSServiceRoleForAmazonInspector2` という名前のサービスにリンクされたロールを使用します。このサービスにリンクされたロールは、ロールを引き受ける上で `inspector2.amazonaws.com` サービスを信頼します。

`AmazonInspector2ServiceRolePolicy` という名前のロールのアクセス許可ポリシーにより、Amazon Inspector は次のようなタスクを実行することが許可されます。

- Amazon Elastic Compute Cloud (Amazon EC2) のアクションを使用して、インスタンスとネットワークパスに関する情報を取得します。
- AWS Systems Manager アクションを使用して Amazon EC2 インスタンスからインベントリを取得し、カスタムパスからサードパーティーパッケージに関する情報を取得します。
- アクションを使用して、ターゲットインスタンスの CIS スキャンを AWS Systems Manager `SendCommand`呼び出します。
- Amazon Elastic Container Registry アクションを使用して、コンテナイメージに関する情報を取得します。
- AWS Lambda アクションを使用して、Lambda 関数に関する情報を取得します。
- AWS Organizations アクションを使用して、関連付けられたアカウントを記述します。
- CloudWatch アクションを使用して、Lambda 関数が最後に呼び出された時刻に関する情報を取得します。
- 選択した IAM アクションを使用して、Lambda コードにセキュリティ脆弱性を生じさせる可能性のある IAM ポリシーに関する情報を取得します。
- CodeGuru セキュリティアクションを使用して、Lambda 関数でコードのスキャンを実行します。Amazon Inspector は、次の CodeGuru セキュリティアクションを使用します。
  - codeguru-security: CreateScan – CodeGuru セキュリティスキャンを作成するアクセス許可を付与します。
  - codeguru-security: GetScan – CodeGuru セキュリティスキャンメタデータを取得するアクセス許可を付与します。
  - codeguru-security: ListFindings – CodeGuru Security によって生成された結果を取得するアクセス許可を付与します。

- codeguru-security: DeleteScansByCategory – Amazon Inspector によって開始されたスキャンを削除するアクセス許可を CodeGuru セキュリティに付与します。 Amazon Inspector
- codeguru-security: BatchGetFindings – CodeGuru Security によって生成された特定の結果のバッチを取得するアクセス許可を付与します。

- 選択した Elastic Load Balancing アクションを使用して、Elastic Load Balancing ターゲットグループの一部である EC2 インスタンスのネットワークスキャンを実行します。

ロールは、次のアクセス許可ポリシーを使用して設定されます。

```
    {
"Version": "2012-10-17",
"Statement": [
 {
  "Sid": "TirosPolicy",
  "Effect": "Allow",
  "Action": [
   "directconnect:DescribeConnections",
   "directconnect:DescribeDirectConnectGatewayAssociations",
   "directconnect:DescribeDirectConnectGatewayAttachments",
   "directconnect:DescribeDirectConnectGateways",
   "directconnect:DescribeVirtualGateways",
   "directconnect:DescribeVirtualInterfaces",
   "ec2:DescribeAvailabilityZones",
   "ec2:DescribeCustomerGateways",
   "ec2:DescribeInstances",
   "ec2:DescribeInternetGateways",
   "ec2:DescribeManagedPrefixLists",
   "ec2:DescribeNatGateways",
   "ec2:DescribeNetworkAcls",
   "ec2:DescribeNetworkInterfaces",
   "ec2:DescribePrefixLists",
   "ec2:DescribeRegions",
   "ec2:DescribeRouteTables",
   "ec2:DescribeSecurityGroups",
   "ec2:DescribeSubnets",
   "ec2:DescribeTransitGatewayAttachments",
   "ec2:DescribeTransitGatewayConnects",
   "ec2:DescribeTransitGatewayPeeringAttachments",
   "ec2:DescribeTransitGatewayRouteTables",
   "ec2:DescribeTransitGatewayVpcAttachments",
```

```
    "ec2:DescribeTransitGateways",
    "ec2:DescribeVpcEndpointServiceConfigurations",
    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcPeeringConnections",
    "ec2:DescribeVpcs",
    "ec2:DescribeVpnConnections",
    "ec2:DescribeVpnGateways",
    "ec2:GetManagedPrefixListEntries",
    "ec2:GetTransitGatewayRouteTablePropagations",
    "ec2:SearchTransitGatewayRoutes",
    "elasticloadbalancing:DescribeListeners",
    "elasticloadbalancing:DescribeLoadBalancerAttributes",
    "elasticloadbalancing:DescribeLoadBalancers",
    "elasticloadbalancing:DescribeRules",
    "elasticloadbalancing:DescribeTags",
    "elasticloadbalancing:DescribeTargetGroups",
    "elasticloadbalancing:DescribeTargetGroupAttributes",
    "elasticloadbalancing:DescribeTargetHealth",
    "network-firewall:DescribeFirewall",
    "network-firewall:DescribeFirewallPolicy",
    "network-firewall:DescribeResourcePolicy",
    "network-firewall:DescribeRuleGroup",
    "network-firewall:ListFirewallPolicies",
    "network-firewall:ListFirewalls",
    "network-firewall:ListRuleGroups",
    "tiros:CreateQuery",
    "tiros:GetQueryAnswer"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Sid": "PackageVulnerabilityScanning",
   "Effect": "Allow",
   "Action": [
    "ecr:BatchGetImage",
    "ecr:BatchGetRepositoryScanningConfiguration",
    "ecr:DescribeImages",
    "ecr:DescribeRegistry",
    "ecr:DescribeRepositories",
    "ecr:GetAuthorizationToken",
    "ecr:GetDownloadUrlForLayer",
    "ecr:GetRegistryScanningConfiguration",
```

```
    "ecr:ListImages",
    "ecr:PutRegistryScanningConfiguration",
    "organizations:DescribeAccount",
    "organizations:DescribeOrganization",
    "organizations:ListAccounts",
    "ssm:DescribeAssociation",
    "ssm:DescribeAssociationExecutions",
    "ssm:DescribeInstanceInformation",
    "ssm:ListAssociations",
    "ssm:ListResourceDataSync"
   ],
   "Resource": "*"
  },
  {
   "Sid": "LambdaPackageVulnerabilityScanning",
   "Effect": "Allow",
   "Action": [
    "lambda:ListFunctions",
    "lambda:GetFunction",
    "lambda:GetLayerVersion",
    "cloudwatch:GetMetricData"
   ],
   "Resource": "*"
  },
  {
   "Sid": "GatherInventory",
   "Effect": "Allow",
   "Action": [
    "ssm:CreateAssociation",
    "ssm:StartAssociationsOnce",
    "ssm:DeleteAssociation",
    "ssm:UpdateAssociation"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:instance/*",
    "arn:aws:ssm:*:*:document/AmazonInspector2-*",
    "arn:aws:ssm:*:*:document/AWS-GatherSoftwareInventory",
    "arn:aws:ssm:*:*:managed-instance/*",
    "arn:aws:ssm:*:*:association/*"
   ]
  },
  {
   "Sid": "DataSyncCleanup",
   "Effect": "Allow",
```

```
    "Action": [
     "ssm:CreateResourceDataSync",
     "ssm:DeleteResourceDataSync"
    ],
    "Resource": [
     "arn:aws:ssm:*:*:resource-data-sync/InspectorResourceDataSync-do-not-delete"
    ]
   },
   {
    "Sid": "ManagedRules",
    "Effect": "Allow",
    "Action": [
     "events:PutRule",
     "events:DeleteRule",
     "events:DescribeRule",
     "events:ListTargetsByRule",
     "events:PutTargets",
     "events:RemoveTargets"
    ],
    "Resource": [
     "arn:aws:events:*:*:rule/DO-NOT-DELETE-AmazonInspector*ManagedRule"
    ]
   },
   {
    "Sid": "LambdaCodeVulnerabilityScanning",
    "Effect": "Allow",
    "Action": [
     "codeguru-security:CreateScan",
     "codeguru-security:GetAccountConfiguration",
     "codeguru-security:GetFindings",
     "codeguru-security:GetScan",
     "codeguru-security:ListFindings",
     "codeguru-security:BatchGetFindings",
     "codeguru-security:DeleteScansByCategory"
    ],
    "Resource": [
     "*"
    ]
   },
   {
    "Sid": "CodeGuruCodeVulnerabilityScanning",
    "Effect": "Allow",
    "Action": [
     "iam:GetRole",
```

```
    "iam:GetRolePolicy",
    "iam:GetPolicy",
    "iam:GetPolicyVersion",
    "iam:ListAttachedRolePolicies",
    "iam:ListPolicies",
    "iam:ListPolicyVersions",
    "iam:ListRolePolicies",
    "lambda:ListVersionsByFunction"
   ],
   "Resource": [
    "*"
   ],
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "codeguru-security.amazonaws.com"
     ]
    }
   }
  },
  {
   "Sid": "Ec2DeepInspection",
   "Effect": "Allow",
   "Action": [
    "ssm:PutParameter",
    "ssm:GetParameters",
    "ssm:DeleteParameter"
   ],
   "Resource": [
    "arn:aws:ssm:*:*:parameter/inspector-aws/service/inspector-linux-application-paths"
   ],
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  },
  {
   "Sid": "AllowManagementOfServiceLinkedChannel",
   "Effect": "Allow",
   "Action": [
    "cloudtrail:CreateServiceLinkedChannel",
    "cloudtrail:DeleteServiceLinkedChannel"
   ],
```

```
    "Resource": [
     "arn:aws:cloudtrail:*:*:channel/aws-service-channel/inspector2/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   },
   {
    "Sid": "AllowListServiceLinkedChannels",
    "Effect": "Allow",
    "Action": [
     "cloudtrail:ListServiceLinkedChannels"
    ],
    "Resource": [
     "*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   },
   {
    "Sid": "AllowToRunInvokeCisSpecificDocuments",
    "Effect": "Allow",
    "Action": [
     "ssm:SendCommand",
     "ssm:GetCommandInvocation"
    ],
    "Resource": [
     "arn:aws:ssm:*:*:document/AmazonInspector2-InvokeInspectorSsmPluginCIS"
    ]
   },
   {
    "Sid": "AllowToRunCisCommandsToSpecificResources",
    "Effect": "Allow",
    "Action": [
     "ssm:SendCommand"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:instance/*"
    ],
```

```
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     }
    }
   },
   {
    "Sid": "AllowToPutCloudwatchMetricData",
    "Effect": "Allow",
    "Action": [
     "cloudwatch:PutMetricData"
    ],
    "Resource": [
     "*"
    ],
    "Condition": {
     "StringEquals": {
      "cloudwatch:namespace": "AWS/Inspector2"
     }
    }
   }
  ]
}
```

Amazon Inspector のサービスリンクロールの作成

サービスリンクロールを手動で作成する必要はありません。、 AWS Management Console、 AWS CLIまたは AWS API で Amazon Inspector をアクティブ化すると、Amazon Inspector によってサービスにリンクされたロールが作成されます。

Amazon Inspector のサービスリンクロールの編集

Amazon Inspector では、AWSServiceRoleForAmazonInspector2 のサービスにリンクされたロールを編集することはできません。サービスにリンクされたロールが作成されると、多くのエンティティによってロールが参照される可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの編集」を参照してください。

Amazon Inspector のサービスリンクロールの削除

Amazon Inspector の使用が不要になった場合は、AWSServiceRoleForAmazonInspector2 サービスにリンクされたロールを削除することをお勧めします。ロールを削除する前に、アクティブ化さ

れている各 で Amazon Inspector AWS リージョン を非アクティブ化する必要があります。Amazon Inspector を非アクティブ化しても、ロールは削除されません。したがって、Amazon Inspector を再度アクティブ化すると、既存のロールを使用することができます。そうすることで、使用していないエンティティがアクティブにモニタリングまたはメンテナンスされるのを防ぐことができます。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

サービスにリンクされたこのロールを削除したが、再作成する必要がある場合は、同じプロセスで、アカウントにロールを再作成することができます。Amazon Inspector をアクティブ化すると、Amazon Inspector がサービスリンクロールを再作成します。

> **Note**
>
> リソースを削除しようとしているときに Amazon Inspector サービスがロールを使用している場合は、削除が失敗する可能性があります。失敗した場合は、数分待ってから操作を再試行してください。

IAM コンソール、、または AWS API を使用して AWS CLI、`AWSServiceRoleForAmazonInspector2`サービスにリンクされたロールを削除できます。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの削除」を参照してください。

Amazon Inspector のエージェントレススキャンに対するサービスリンクロールのアクセス許可

Amazon Inspector のエージェントレススキャンでは、`AWSServiceRoleForAmazonInspector2Agentless` という名前のサービスリンクロールを使用します。このサービスリンクロールにより、Amazon Inspector はアカウントに Amazon EBS ボリュームのスナップショットを作成し、そのスナップショットからデータにアクセスできるようになります。このサービスリンクロールは、ロールを引き受ける上で `agentless.inspector2.amazonaws.com` サービスを信頼します。

> **Important**
>
> このサービスリンクロールのステートメントにより、Amazon Inspector は、`InspectorEc2Exclusion` タグを使用してスキャンから除外した EC2 インスタンスに対してエージェントレススキャンを実行できなくなります。さらに、このステートメントは、暗号化に使用される KMS キーに `InspectorEc2Exclusion` タグが付いている場合

に、Amazon Inspector がボリュームの暗号化されたデータにアクセスできないようにします。詳細については、「Amazon Inspector スキャンからのインスタンスの除外」を参照してください。

AmazonInspector2AgentlessServiceRolePolicy という名前のロールのアクセス許可ポリシーにより、Amazon Inspector は次のようなタスクを実行することが許可されます。

- Amazon Elastic Compute Cloud (Amazon EC2) アクションを使用して、EC2 インスタンス、ボリューム、スナップショットに関する情報を取得します。
  - Amazon EC2 のタグ付けアクションを使用して、InspectorScan タグキーでスキャンのスナップショットにタグを付けます。
  - Amazon EC2 のスナップショットアクションを使用してスナップショットを作成し、InspectorScan タグキーでタグ付けしてから、InspectorScan タグキーでタグ付けされた Amazon EBS ボリュームのスナップショットを削除します。
- Amazon EBS アクションを使用して、InspectorScan タグキーでタグ付けされたスナップショットから情報を取得します。
- Select 復 AWS KMS 号アクションを使用して、 AWS KMS カスタマーマネージドキーで暗号化されたスナップショットを復号します。Amazon Inspector は、スナップショットの暗号化に使用された KMS キーに InspectorEc2Exclusion タグが付いている場合、スナップショットを復号しません。

ロールは、次のアクセス許可ポリシーを使用して設定されます。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "InstanceIdentification",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeVolumes",
    "ec2:DescribeSnapshots"
   ],
   "Resource": "*"
  },
```

```
{
 "Sid": "GetSnapshotData",
 "Effect": "Allow",
 "Action": [
  "ebs:ListSnapshotBlocks",
  "ebs:GetSnapshotBlock"
 ],
 "Resource": "arn:aws:ec2:*:*:snapshot/*",
 "Condition": {
  "StringLike": {
   "aws:ResourceTag/InspectorScan": "*"
  }
 }
},
{
 "Sid": "CreateSnapshotsAnyInstanceOrVolume",
 "Effect": "Allow",
 "Action": "ec2:CreateSnapshots",
 "Resource": [
  "arn:aws:ec2:*:*:instance/*",
  "arn:aws:ec2:*:*:volume/*"
 ]
},
{
 "Sid": "DenyCreateSnapshotsOnExcludedInstances",
 "Effect": "Deny",
 "Action": "ec2:CreateSnapshots",
 "Resource": "arn:aws:ec2:*:*:instance/*",
 "Condition": {
  "StringEquals": {
   "ec2:ResourceTag/InspectorEc2Exclusion": "true"
  }
 }
},
{
 "Sid": "CreateSnapshotsOnAnySnapshotOnlyWithTag",
 "Effect": "Allow",
 "Action": "ec2:CreateSnapshots",
 "Resource": "arn:aws:ec2:*:*:snapshot/*",
 "Condition": {
  "Null": {
   "aws:TagKeys": "false"
  },
  "ForAllValues:StringEquals": {
```

```
      "aws:TagKeys": "InspectorScan"
     }
    }
   },
   {
    "Sid": "CreateOnlyInspectorScanTagOnlyUsingCreateSnapshots",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition": {
     "StringLike": {
      "ec2:CreateAction": "CreateSnapshots"
     },
     "Null": {
      "aws:TagKeys": "false"
     },
     "ForAllValues:StringEquals": {
      "aws:TagKeys": "InspectorScan"
     }
    }
   },
   {
    "Sid": "DeleteOnlySnapshotsTaggedForScanning",
    "Effect": "Allow",
    "Action": "ec2:DeleteSnapshot",
    "Resource": "arn:aws:ec2:*:*:snapshot/*",
    "Condition": {
     "StringLike": {
      "ec2:ResourceTag/InspectorScan": "*"
     }
    }
   },
   {
    "Sid": "DenyKmsDecryptForExcludedKeys",
    "Effect": "Deny",
    "Action": "kms:Decrypt",
    "Resource": "arn:aws:kms:*:*:key/*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/InspectorEc2Exclusion": "true"
     }
    }
   },
   {
```

```
    "Sid": "DecryptSnapshotBlocksVolContext",
    "Effect": "Allow",
    "Action": "kms:Decrypt",
    "Resource": "arn:aws:kms:*:*:key/*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     },
     "StringLike": {
      "kms:ViaService": "ec2.*.amazonaws.com",
      "kms:EncryptionContext:aws:ebs:id": "vol-*"
     }
    }
   },
   {
    "Sid": "DecryptSnapshotBlocksSnapContext",
    "Effect": "Allow",
    "Action": "kms:Decrypt",
    "Resource": "arn:aws:kms:*:*:key/*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     },
     "StringLike": {
      "kms:ViaService": "ec2.*.amazonaws.com",
      "kms:EncryptionContext:aws:ebs:id": "snap-*"
     }
    }
   },
   {
    "Sid": "DescribeKeysForEbsOperations",
    "Effect": "Allow",
    "Action": "kms:DescribeKey",
    "Resource": "arn:aws:kms:*:*:key/*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceAccount": "${aws:PrincipalAccount}"
     },
     "StringLike": {
      "kms:ViaService": "ec2.*.amazonaws.com"
     }
    }
   },
   {
```

```
    "Sid": "ListKeyResourceTags",
    "Effect": "Allow",
    "Action": "kms:ListResourceTags",
    "Resource": "arn:aws:kms:*:*:key/*"
   }
  ]
}
```

エージェントレススキャンのサービスリンクロールの作成

サービスリンクロールを手動で作成する必要はありません。 AWS Management Console、、 AWS CLIまたは AWS API で Amazon Inspector をアクティブ化すると、Amazon Inspector によってサービスにリンクされたロールが作成されます。

エージェントレススキャンのサービスリンクロールの編集

Amazon Inspector では、AWSServiceRoleForAmazonInspector2Agentless のサービスにリンクされたロールを編集することはできません。サービスにリンクされたロールが作成されると、多くのエンティティによってロールが参照される可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの編集」を参照してください。

エージェントレススキャンのサービスリンクロールの削除

サービスリンクロールが必要な機能またはサービスが不要になった場合には、そのロールを削除することをお勧めします。これにより、使用していないエンティティがアクティブにモニタリングされたり、メンテナンスされたりすることがなくなります。

> **Important**
>
> AWSServiceRoleForAmazonInspector2Agentless ロールを削除するには、エージェントレススキャンが可能なすべてのリージョンで、スキャンモードをエージェントベースに設定する必要があります。詳細については、「TBD スキャンモードの設定のリンク」を参照してください。

サービスにリンクされたロールを IAM で手動削除するには

IAM コンソール、、または AWS API を使用して AWS CLI、AWSServiceRoleForAmazonInspector2Agentless サービスにリンクされたロールを削除します。詳

細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの削除」を参照してください。

Amazon Inspector アイデンティティとアクセスのトラブルシューティング

以下の情報を使用して、Amazon Inspector と IAM の使用時に発生する可能性がある一般的な問題の診断と修正に役立てます。

トピック

- Amazon Inspector でアクションを実行する認可がない
- iam を実行する権限がありません。PassRole
- 自分の 以外のユーザーに Amazon Inspector リソース AWS アカウント へのアクセスを許可したい

Amazon Inspector でアクションを実行する認可がない

あるアクションを実行するアクセス許可がないというエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要なinspector2:*GetWidget* アクセス許可を持っていない場合に発生するものです。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 inspector2:GetWidget on resource: my-example-widget
```

この場合、inspector2:*GetWidget* アクションを使用して *my-example-widget*リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

iam を実行する権限がありません。PassRole

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して Amazon Inspector にロールを渡せるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

以下のエラーの例は、marymajor という名前の IAM ユーザーがコンソールを使用して Amazon Inspector でアクションを実行しようする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。Mary には、ロールをサービスに渡す権限がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

自分の 以外のユーザーに Amazon Inspector リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- Amazon Inspector がこれらの機能をサポートしているかどうかを確認するには、「Amazon Inspector と IAM の連携」を参照してください。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、IAM ユーザーガイドの「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」を参照してください。
- リソースへのアクセスをサードパーティー に提供する方法については AWS アカウント、IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する へのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、『IAM ユーザーガイド』の「外部で認証されたユーザー (ID フェデレーション) へのアクセス権限」を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いの詳細については、「IAM ユーザーガイド」の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

Amazon Inspector のモニタリング

モニタリングは、Amazon Inspector およびその他の AWS ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。 は、Amazon Inspector をモニタリングし、問題が発生した場合は報告し、必要に応じて自動アクションを実行するためのモニタリングツール AWS を提供します。

- Amazon EventBridge は、アプリケーションをさまざまなソースのデータに簡単に接続できるサーバーレスイベントバスサービスです。 は、独自のアプリケーション、S oftware-as-aサービス (SaaS) アプリケーション、および AWS のサービスからリアルタイムデータのストリームを EventBridge 配信し、そのデータを Lambda などのターゲットにルーティングします。これにより、サービスで発生したイベントをモニタリングし、イベント駆動型アーキテクチャを構築できます。詳細については、「Amazon ユーザーガイド EventBridge 」を参照してください。
- AWS CloudTrail は、 AWS アカウントによって行われた、またはそのアカウントに代わって実行された API コールと関連イベントをキャプチャします。 CloudTrail 次に、指定した Amazon S3 バケットにログファイルを配信します。を呼び出したユーザーとアカウント AWS、呼び出し元の IP アドレス、呼び出しが発生した日時を特定できます。詳細については、『AWS CloudTrail ユーザーガイド』を参照してください。

AWS CloudTrailでの Amazon Inspector API コールのログ記録

Amazon Inspector は AWS CloudTrail、IAM ユーザーまたはロール、または によって実行されたアクションを記録するサービスである と統合されています。Amazon Inspector では AWS のサービス、Amazon Inspector のすべての API Amazon Inspector コールがイベントとして CloudTrail キャプチャされます。キャプチャされた呼び出しには、Amazon Inspector コンソールからの呼び出しと、Amazon Inspector API オペレーションへの呼び出しが含まれます。証跡を作成する場合は、Amazon Inspector の CloudTrail イベントなど、Amazon S3 バケットへのイベントの継続的な配信を有効にすることができます。 Amazon S3 証跡を設定しない場合でも、 コンソールの CloudTrail イベント履歴 で最新のイベントを表示できます。によって収集された情報を使用して CloudTrail、以下を判断できます。

- Amazon Inspector に対して行われたリクエスト
- リクエストが行われた IP アドレス。
- 誰がリクエストを行ったか。
- リクエストが行われた時。

の詳細については CloudTrail、「 AWS CloudTrail ユーザーガイド」を参照してください。

の Amazon Inspector 情報 CloudTrail

CloudTrail アカウントを作成する AWS アカウント と、 で が有効になります。Amazon Inspector でアクティビティが発生すると、そのアクティビティは CloudTrail イベント履歴 の他の AWS のサービス イベントとともにイベントに記録されます。 で最近のイベントを表示、検索、ダウンロードできます AWS アカウント。詳細については、「イベント履歴 を使用した CloudTrail イベントの表示」を参照してください。

Amazon Inspector のイベントなど AWS アカウント、 のイベントの継続的な記録については、証跡を作成します。証跡により CloudTrail 、 はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョンに適用されます。証跡は、 AWS パーティションのすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、他の を設定 AWS のサービス して、 CloudTrail ログで収集されたイベントデータをさらに分析し、それに基づく対応を行うことができます。詳細については、次のトピックを参照してください。

- 「証跡作成の概要」
- CloudTrail がサポートするサービスと統合
- CloudTrail の Amazon SNS 通知の設定
- 複数のアカウントからの CloudTrail ログファイルの受信
- 複数のリージョンからの CloudTrail ログファイルの受信

すべての Amazon Inspector アクションは によってログに記録されます CloudTrail。Amazon Inspector で実行できるすべてのアクションは、「Amazon Inspector API リファレンス」に記載されています。例えば、`CreateFindingsReport`、、および `UpdateOrganizationConfiguration`アクションを呼び出すと`ListCoverage`、 CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます:

- リクエストが、ルートユーザーまたは IAM ユーザーのどちらの認証情報を使用して送信されたかどうか。
- リクエストが、ロールとフェデレーティッドユーザーのどちらかの一時的なセキュリティ認証情報を使用して送信されたかどうか。

- リクエストが、別の AWS のサービスによって送信されたかどうか。

詳細については、CloudTrail userIdentity 要素」を参照してください。

Amazon Inspector ログファイルエントリの理解

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できるようにする設定です。 CloudTrail ログファイルには 1 つ以上のログエントリが含まれます。イベント は、任意の送信元からの単一の要求を表します。イベントには、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。 CloudTrail ログファイルはパブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

の Amazon Inspector スキャン情報 CloudTrail

Amazon Inspector スキャンは と統合されています CloudTrail。Amazon Inspector スキャン API オペレーションはすべて管理イベントとしてログ記録されます。Amazon Inspector が に記録する Amazon Inspector スキャン API Amazon Inspector オペレーションのリストについては CloudTrail、Amazon Inspector API リファレンス」の「Amazon Inspector スキャン」を参照してください。 Amazon Inspector

次の例は、 `ScanSbom`アクションを示す CloudTrail ログエントリを示しています。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA123456789EXAMPLE:akua_mansa",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/akua_mansa",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123456789EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-10-17T15:22:59Z",
```

```
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-10-17T16:02:34Z",
    "eventSource": "gamma-inspector-scan.amazonaws.com",
    "eventName": "ScanSbom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "aws-sdk-java/2.20.162 Mac_OS_X/13.5.2 OpenJDK_64-
Bit_Server_VM/17.0.8+7-LTS Java/17.0.8 vendor/Amazon.com_Inc. io/sync http/
UrlConnection cfg/retry-mode/legacy",
    "requestParameters": {
        "sbom": {
            "specVersion": "1.5",
            "metadata": {
                "component": {
                    "name": "debian",
                    "type": "operating-system",
                    "version": "9"
                }
            },
            "components": [
                {
                    "name": "packageOne",
                    "purl": "pkg:deb/debian/packageOne@1.0.0?arch=x86_64&distro=9",
                    "type": "application"
                }
            ],
            "bomFormat": "CycloneDX"
        }
    },
    "responseElements": null,
    "requestID": "f041a27f-f33e-4f70-b09b-5fbc5927282a",
    "eventID": "abc8d1e4-d214-4f07-bc56-8a31be6e36fe",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Amazon Inspector のコンプライアンス検証

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、コンプライアンスプログラムAWS のサービス による対象範囲内のコンプライアンスプログラムを参照し、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、AWS 「コンプライアンスプログラム」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「 でのレポートのダウンロード AWS Artifact」の」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。 は、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- セキュリティとコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。
- アマゾン ウェブ サービスにおける HIPAA セキュリティとコンプライアンスのためのアーキテクチャ – このホワイトペーパーでは、企業が AWS を使用して HIPAA 対象アプリケーションを作成する方法について説明します。

  > **Note**
  >
  > すべて AWS のサービス HIPAA の対象となるわけではありません。詳細については、「HIPAA 対応サービスのリファレンス」を参照してください。

- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- AWS カスタマーコンプライアンスガイド – コンプライアンスの観点から責任共有モデルを理解します。このガイドでは、ガイダンスを保護し AWS のサービス 、複数のフレームワーク (米国国立標準技術研究所 (NIST)、Payment Card Industry Security Standards Council (PCI)、国際標準化機構 (ISO) を含む) のセキュリティコントロールにマッピングするためのベストプラクティスをまとめています。
- 「 デベロッパーガイド」の「 ルールによるリソースの評価」 – この AWS Config サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。 AWS Config

- AWS Security Hub – これにより AWS のサービス 、 内のセキュリティ状態を包括的に把握できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、「Security Hub のコントロールリファレンス」を参照してください。
- AWS Audit Manager – これにより AWS のサービス 、 AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

Amazon Inspector の耐障害性

AWS グローバルインフラストラクチャは、 AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。 は、低レイテンシー、高スループット、高冗長ネットワークで接続された、物理的に分離および分離された複数のアベイラビリティーゾーン AWS リージョン を提供します。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性が高く、フォールトトレラントで、スケーラブルです。

Amazon Inspector のインフラストラクチャセキュリティ

マネージドサービスである Amazon Inspector は AWS グローバルネットワークセキュリティで保護されています。 AWS セキュリティサービスと がインフラストラクチャ AWS を保護する方法については、AWS 「 クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「 Security Pillar AWS Well-Architected Framework」の「Infrastructure Protection」を参照してください。

が AWS 公開した API コールを使用して、ネットワーク経由で Amazon Inspector にアクセスします。クライアントは以下をサポートする必要があります:

- Transport Layer Security (TLS)。TLS 1.2 は必須で TLS 1.3 がお勧めです。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

また、リクエストには、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token Service (AWS

STS) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

Amazon Inspector でのインシデントへの対応

AWSでは、セキュリティが最優先事項です。 AWS クラウド責任共有モデル の一環として、 は、セキュリティを最も重視する組織の要件を満たすデータセンター、ネットワーク、およびソフトウェアアーキテクチャ AWS を管理します。 AWS は、 AWS Config サービス自体に関するインシデント対応を担当します。また、 AWS お客様はクラウドでセキュリティを維持する責任を共有します。つまり、ユーザーは、アクセスできる AWS ツールや機能から実装するセキュリティを制御し、責任共有モデルのユーザー側でインシデント対応に責任を負います。

クラウド上で稼働するアプリケーションの目標を満たすセキュリティベースラインを確立することで、対応可能な逸脱を検出できます。セキュリティインシデント対応は複雑なトピックになる可能性があるため、インシデント対応 (IR) とその選択が企業目標に与える影響をよりよく理解できるように、セキュリティインシデント対応AWS ガイド、AWS セキュリティのベストプラクティスホワイトペーパー、および AWS クラウド導入フレームワーク (CAF) のセキュリティの観点からホワイトペーパーを読むことをお勧めします。

Amazon Inspector との統合

Amazon Inspector は他の AWS サービスと統合されます。これらのサービスは Amazon Inspector からデータを取り込み、新しい方法で検出結果を表示できるようにします。次の統合オプションを参照して、そのサービスが Amazon Inspector でどのように動作するように設定されているかについての詳細を確認してください。

Amazon Inspector と Amazon ECR の統合

Amazon Elastic Container Registry (Amazon ECR) は、フルマネージドされた Docker コンテナレジストリで、コンテナイメージの保存、共有、デプロイを容易に行うことができます。Amazon ECR プライベートレジストリは、可用性の高いスケーラブルなアーキテクチャでコンテナイメージをホストします。Amazon Inspector を使用して、Amazon ECR リポジトリにあるコンテナイメージをスキャンし、脆弱なオペレーティングシステムパッケージやプログラミング言語パッケージを確認することができます。

Amazon Inspector での Amazon ECR の使用に関する詳細は、「Amazon Elastic Container Registry (Amazon ECR) と、Amazon Inspector の統合」を参照してください。

Amazon Inspector と の統合 AWS Security Hub

AWS Security Hub は、 AWS アカウント、サービス、およびその他のサポートされている製品からセキュリティデータを収集し、業界標準とベストプラクティスに従って環境のセキュリティ状態を評価します。セキュリティ体制の評価に加えて、Security Hub は、統合されたすべての AWS サービスおよび AWS パートナーネットワーク製品にわたる検出結果のための一元的な場所を作成します。Amazon Inspector で Security Hub をアクティブ化すると、Security Hub は Amazon Inspector の検出結果データを自動的に取り込むことができます。

Amazon Inspector で Security Hub を使用する方法については、「Amazon Inspector と の統合 AWS Security Hub」を参照してください。

Amazon Elastic Container Registry (Amazon ECR) と、Amazon Inspector の統合

Amazon ECR は AWSで Docker と OCI のイメージとアーティファクトをサポートするフルマネージド型のコンテナレジストリです。Amazon ECR を使用している場合は、レジストリの拡張スキャンをアクティブ化して、Amazon Inspector がコンテナイメージを自動的に検出し、脆弱なオペレー

ティングシステムパッケージやプログラミング言語パッケージをスキャンできるようにすることができます。

この統合により、コンテナイメージに関する Amazon Inspector の検出結果を Amazon ECR コンソール内で表示できるようになります。さらに、Amazon ECR コンソールでは、インクルージョンフィルタを作成してスキャンの頻度を管理し、スキャンの範囲を絞り込むことができます。

統合をアクティブ化する

統合をアクティブ化するには、Amazon Inspector コンソールまたは API を使用して Amazon Inspector のスキャンをアクティブ化するか、Amazon ECR コンソールまたは API を使用して Amazon Inspector による拡張スキャンを使用するようにリポジトリを設定します。

Amazon Inspector を使用して統合をアクティブ化する方法の詳細については、「Amazon Inspector による自動リソーススキャン」を参照してください。

Amazon ECR での拡張スキャンのアクティブ化と設定については、「Amazon ECR ユーザーガイド」の「拡張スキャン」を参照してください。

マルチアカウント環境との統合を使用する

マルチアカウント環境のメンバーであれば、Amazon ECR から拡張スキャンをアクティブ化にできます。ただし、一度アクティブ化すると、Amazon Inspector の委任された管理者だけが非アクティブ化できます。非アクティブ化すると、「基本的なスキャン」に戻ります。詳細については、「Amazon Inspector の非アクティブ化」を参照してください。

Amazon Inspector と の統合 AWS Security Hub

Security Hub は、 のセキュリティ状態を包括的に把握 AWS し、セキュリティ業界標準とベストプラクティスに照らして環境をチェックするのに役立ちます。Security Hub は、 AWS アカウント、サービス、その他のサポートされている製品全体からセキュリティデータを収集します。提供される情報を使用して、セキュリティの傾向を分析し、最も優先度の高いセキュリティ問題を特定できます。

Amazon Inspector と Security Hub との統合により、検出結果を Amazon Inspector から Security Hub へ送信できます。Security Hub では、このような検出結果をセキュリティ体制の分析に含めることができます。

では AWS Security Hub、セキュリティの問題は検出結果として追跡されます。一部の検出結果は、他の AWS サービスまたはサードパーティー製品によって検出された問題によって発生しま

す。Security Hub には、セキュリティの問題を検出し、検出結果を生成するために使用する一連のルールもあります。Security Hub には、これらすべてのソースからの結果を管理するためのツールが用意されています。検出結果の一覧を表示およびフィルタリングして、検出結果の詳細を表示できます。Security Hub での結果の詳細については、「AWS Security Hub ユーザーガイド」 の「検出結果の表示」を参照してください。検出結果の調査状況を追跡することもできます 「AWS Security Hub ユーザーガイド」の「Taking action on findings」(検出結果に対するアクションの実行) を参照してください。

Security Hub のすべての検出結果は、 AWS Security Finding Format (ASFF) と呼ばれる標準の JSON 形式を使用します。ASFF には、問題のソース、影響を受けるリソース、および検出結果の現在のステータスに関する詳細が含まれます。 AWS ユーザーガイド の「AWS Security Hub Security Finding 形式 (ASFF)」を参照してください 。

Security Hub は、Amazon Inspector で検出結果が処理され、Amazon Inspector で終了された時点でそれらの検出結果をアーカイブします。

AWS Security HubでAmazon Inspector の検出結果の表示

Amazon Inspector Classic と新しい Amazon Inspector の検出結果は、Security Hub の同じパネルで確認できます。ただし、フィルターバーに `"aws/inspector/ProductVersion": "2"` を追加することで、新しい Amazon Inspector の検出結果をフィルタリングできます。このフィルターを追加すると、Security Hub ダッシュボードの Amazon Inspector Classic の結果は除外されます。

Amazon Inspector の検出結果例

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:inspector2:us-east-1:123456789012:finding/FINDING_ID",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/inspector",
  "ProductName": "Inspector",
  "CompanyName": "Amazon",
  "Region": "us-east-1",
  "GeneratorId": "AWSInspector",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks/Vulnerabilities/CVE"
  ],
  "FirstObservedAt": "2023-01-31T20:25:38Z",
  "LastObservedAt": "2023-05-04T18:18:43Z",
  "CreatedAt": "2023-01-31T20:25:38Z",
  "UpdatedAt": "2023-05-04T18:18:43Z",
```

```
  "Severity": {
    "Label": "HIGH",
    "Normalized": 70
  },
  "Title": "CVE-2022-34918 - kernel",
  "Description": "An issue was discovered in the Linux kernel through 5.18.9. A type
 confusion bug in nft_set_elem_init (leading to a buffer overflow) could be used by a
 local attacker to escalate privileges, a different vulnerability than CVE-2022-32250.
 (The attacker can obtain root access, but must start with an unprivileged user
 namespace to obtain CAP_NET_ADMIN access.) This can be fixed in nft_setelem_parse_data
 in net/netfilter/nf_tables_api.c.",
  "Remediation": {
    "Recommendation": {
      "Text": "Remediation is available. Please refer to the Fixed version in the
 vulnerability details section above. For detailed remediation guidance for each of the
 affected packages, refer to the vulnerabilities section of the detailed finding JSON."
    }
  },
  "ProductFields": {
    "aws/inspector/FindingStatus": "ACTIVE",
    "aws/inspector/inspectorScore": "7.8",
    "aws/inspector/resources/1/resourceDetails/awsEc2InstanceDetails/platform":
 "AMAZON_LINUX_2",
    "aws/inspector/ProductVersion": "2",
    "aws/inspector/instanceId": "i-0f1ed287081bdf0fb",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/inspector/
arn:aws:inspector2:us-east-1:123456789012:finding/FINDING_ID",
    "aws/securityhub/ProductName": "Inspector",
    "aws/securityhub/CompanyName": "Amazon"
  },
  "Resources": [
    {
      "Type": "AwsEc2Instance",
      "Id": "arn:aws:ec2:us-east-1:123456789012:i-0f1ed287081bdf0fb",
      "Partition": "aws",
      "Region": "us-east-1",
      "Tags": {
        "Patch Group": "SSM",
        "Name": "High-SEv-Test"
      },
      "Details": {
        "AwsEc2Instance": {
          "Type": "t2.micro",
          "ImageId": "ami-0cff7528ff583bf9a",
```

```
          "IpV4Addresses": [
            "52.87.229.97",
            "172.31.57.162"
          ],
          "KeyName": "ACloudGuru",
          "IamInstanceProfileArn": "arn:aws:iam::123456789012:instance-profile/
AmazonSSMRoleForInstancesQuickSetup",
          "VpcId": "vpc-a0c2d7c7",
          "SubnetId": "subnet-9c934cb1",
          "LaunchedAt": "2022-07-26T21:49:46Z"
        }
      }
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "Vulnerabilities": [
    {
      "Id": "CVE-2022-34918",
      "VulnerablePackages": [
        {
          "Name": "kernel",
          "Version": "5.10.118",
          "Epoch": "0",
          "Release": "111.515.amzn2",
          "Architecture": "X86_64",
          "PackageManager": "OS",
          "FixedInVersion": "0:5.10.130-118.517.amzn2",
          "Remediation": "yum update kernel"
        }
      ],
      "Cvss": [
        {
          "Version": "2.0",
          "BaseScore": 7.2,
          "BaseVector": "AV:L/AC:L/Au:N/C:C/I:C/A:C",
          "Source": "NVD"
        },
        {
          "Version": "3.1",
          "BaseScore": 7.8,
```

```
          "BaseVector": "CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H",
          "Source": "NVD"
        },
        {
          "Version": "3.1",
          "BaseScore": 7.8,
          "BaseVector": "CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H",
          "Source": "NVD",
          "Adjustments": []
        }
      ],
      "Vendor": {
        "Name": "NVD",
        "Url": "https://nvd.nist.gov/vuln/detail/CVE-2022-34918",
        "VendorSeverity": "HIGH",
        "VendorCreatedAt": "2022-07-04T21:15:00Z",
        "VendorUpdatedAt": "2022-10-26T17:05:00Z"
      },
      "ReferenceUrls": [
        "https://git.kernel.org/pub/scm/linux/kernel/git/netdev/net.git/commit/?
id=7e6bc1f6cabcd30aba0b11219d8e01b952eacbb6",
        "https://lore.kernel.org/netfilter-devel/cd9428b6-7ffb-dd22-d949-
d86f4869f452@randorisec.fr/T/",
        "https://www.debian.org/security/2022/dsa-5191"
      ],
      "FixAvailable": "YES"
    }
  ],
  "FindingProviderFields": {
    "Severity": {
      "Label": "HIGH"
    },
    "Types": [
      "Software and Configuration Checks/Vulnerabilities/CVE"
    ]
  },
  "ProcessedAt": "2023-05-05T20:28:38.822Z"
}
```

統合のアクティブ化と構成

Amazon Inspector と の統合を使用するには AWS Security Hub、Security Hub をアクティブ化する必要があります。Security Hub をアクティブ化する方法については、「AWS Security Hub ユーザーガイド」の「Security Hub の設定」を参照してください。

Amazon Inspector と Security Hub の両方をアクティブ化すると、統合が自動的にアクティブ化になり、Amazon Inspector は検出結果を Security Hub へ送信し始めます。Amazon Inspectorは、AWS Security Finding Format (ASFF) を使用して生成したすべての検出結果を Security Hub に送信します。

への検出結果の発行の停止 AWS Security Hub

検出結果の送信を停止する方法

Security Hub への結果の送信を停止するには、Security Hub コンソールまたは API を使用できます。

「統合からの検出結果のフローの非アクティブ化とアクティブ化 (コンソール)」またはAWS Security Hub ユーザーガイドの「統合からの検出結果のフローの非アクティブ化 (Security Hub API、 AWS CLI)」を参照してください 。

Amazon Inspector でサポートされているオペレーティングシステムとプログラミング言語

Amazon Inspector は、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスにインストールされているソフトウェアアプリケーション、Amazon Elastic Container Registry (Amazon ECR) リポジトリに保存されているコンテナイメージ、および AWS Lambda 関数をスキャンできます。ECR コンテナイメージの場合、Amazon Inspector はオペレーティングシステムとプログラミング言語パッケージの両方の脆弱性をスキャンできます。Lambda 関数の場合、Amazon Inspector はコードの脆弱性をスキャンできます。Amazon Inspector がリソースをスキャンする際、専用のスキャンエンジンを使用し、50 を超えるデータフィードをソースとして、共通脆弱性識別子 (CVE) の検出結果を生成します。ソースには、ベンダーセキュリティアドバイザリ、NVD、MITRE、オープンソースフィード、内部リサーチ、ライセンスデータフィードなどが含まれます。

Amazon Inspector がリソースをスキャンするには、リソースがサポートされているオペレーティングシステムを実行しているか、サポートされているプログラミング言語を使用している必要があります。このセクションのトピックでは、Amazon Inspector が現在さまざまなリソースとスキャンタイプでサポートしているオペレーティングシステム、ランタイム、プログラミング言語を示します。また、Amazon Inspector が以前サポートしていたオペレーティングシステムも一覧表示されていますが、その後ベンダーによって廃止されています。ベンダーがオペレーティングシステムのサポートを終了した後は、Amazon Inspector では、そのオペレーティングシステムに対して限定的なサポートしか提供できません。

トピック

サポートされているオペレーティングシステム: Amazon EC2スキャン

次の表に、Amazon Inspector が Amazon EC2 インスタンスのスキャンで現在サポートしているオペレーティングシステムを示します。 Amazon EC2 また、各ベンダーのセキュリティアドバイザリのソースと、そのオペレーティングシステムをエージェントベースまたはエージェントレスのスキャン方法を使用してスキャンできるかどうかも一覧表示されます。スキャン方式の詳細については、「エージェントベースのスキャン」と「エージェントレススキャン」を参照してください。

> **Note**
>
> Linux オペレーティングシステムの検出は、デフォルトのパッケージマネージャーリポジトリでのみサポートされており、サードパーティーアプリケーション、拡張サポートリポジトリ (BYOS RHEL、PAYG RHEL、RHEL for SAP など)、Red Hat Application Streams などのオプションのリポジトリは含まれません。

オペレーティングシステム	Version	ベンダーのセキュリティアドバイザリ	エージェントレススキャンのサポート	エージェントベースのスキャンのサポート
AlmaLinux	8	ALSA	はい	はい
AlmaLinux	9	ALSA	はい	はい
Amazon Linux (AL2)	AL2	ALAS	はい	はい
Amazon Linux 2023 (AL2023)	AL2023	ALAS	はい	はい
Bottlerocket	1.7.0 以降	GHSA, CVE	いいえ	はい
CentOS Linux (CentOS)	7	CESA	はい	はい
Debian サーバー (Buster)	10	DSA	はい	はい

オペレーティングシステム	Version	ベンダーのセキュリティアドバイザリ	エージェントレススキャンのサポート	エージェントベースのスキャンのサポート
Debian サーバー (Bullseye)	11	DSA	はい	はい
Debian サーバー (Bookworm)	12	DSA	はい	はい
Fedora	38	CVE	はい	はい
Fedora	39	CVE	はい	はい
openSUSE	15.5	CVE	はい	はい
Oracle Linux (Oracle)	7	ELSA	はい	はい
Oracle Linux (Oracle)	8	ELSA	はい	はい
Oracle Linux (Oracle)	9	ELSA	はい	はい
Red Hat Enterprise Linux (RHEL)	7	RHSA	はい	はい
Red Hat Enterprise Linux (RHEL)	8	RHSA	はい	はい
Red Hat Enterprise Linux (RHEL)	9	RHSA	はい	はい
Rocky Linux	8	RLSA	はい	はい
Rocky Linux	9	RLSA	はい	はい

オペレーティングシステム	Version	ベンダーのセキュリティアドバイザリ	エージェントレススキャンのサポート	エージェントベースのスキャンのサポート
SUSE Linux Enterprise Server (SLES)	12.4	SUSE CVE	はい	はい
SUSE Linux Enterprise Server (SLES)	12.5	SUSE CVE	はい	はい
SUSE Linux Enterprise Server (SLES)	15.3	SUSE CVE	はい	はい
SUSE Linux Enterprise Server (SLES)	15.4	SUSE CVE	はい	はい
SUSE Linux Enterprise Server (SLES)	15.5	SUSE CVE	はい	はい
Ubuntu (Trusty)	14.04 (ESM)	USN、Ubuntu Pro	はい	はい
Ubuntu (Xenial)	16.04 (ESM)	USN、Ubuntu Pro	はい	はい
Ubuntu (Bionic)	18.04 (ESM)	USN、Ubuntu Pro	はい	はい
Ubuntu (Focal)	20.04 (LTS)	USN	はい	はい
Ubuntu (Jammy)	22.04 (LTS)	USN	はい	はい
Ubuntu (Mantic Minotaur)	23.10	USN	はい	はい

オペレーティングシステム	Version	ベンダーのセキュリティアドバイザリ	エージェントレススキャンのサポート	エージェントベースのスキャンのサポート
Windows Server	2016	MSKB	いいえ	はい
Windows Server	2019	MSKB	いいえ	はい
Windows Server	2022	MSKB	いいえ	はい
macOS (Mojave)	10.14	APPLE-SA	いいえ	はい
macOS (カタリナ)	10.15	APPLE-SA	いいえ	はい
macOS (ビッグサー)	11	APPLE-SA	いいえ	はい
macOS (モントレー)	12	APPLE-SA	いいえ	はい
macOS (ベンチュラ)	13	APPLE-SA	いいえ	はい

サポートされているプログラミング言語: Amazon EC2 の詳細検査

Amazon Inspector は現在、Amazon EC2 Linux インスタンスをスキャンしてサードパーティーソフトウェアパッケージの脆弱性を確認するときに、次のプログラミング言語をサポートしています。

- Java
- JavaScript
- Python

Amazon Inspector は、Systems Manager Distributor を使用して、Amazon EC2 インスタンスで詳細検査に使用されるプラグインをデプロイします。Systems Manager Distributor は、「Systems Manager ガイド」の「サポートされているパッケージのプラットフォームとアーキテクチャ」に記載されているオペレーティングシステムをサポートします。Amazon Inspector が詳細検査スキャ

ンを実行するには、Amazon EC2 インスタンスのオペレーティングシステムが Systems Manager Distributor と Amazon Inspector によってサポートされている必要があります。

> **Note**
>
> Deep inspection は Bottlerocket オペレーティングシステムではサポートされていません。

サポートされているオペレーティングシステム: CIS スキャン

次の表に、Amazon Inspector が CIS スキャンで現在サポートしているオペレーティングシステムを示します。この表には、そのオペレーティングシステムのスキャンを実行するために使用される CIS ベンチマークバージョンも含まれています。

オペレーティングシステム	Version	CIS ベンチマークバージョン
Amazon Linux 2	AL2	2.0.0
Amazon Linux 2023	AL2023	1.0.0
Windows Server	2019	2.0.0
Windows Server	2022	2.0.0

サポートされているオペレーティングシステム: Amazon Inspector による Amazon ECR スキャン

Amazon Inspector は現在、Amazon ECR リポジトリ内のコンテナイメージをスキャンするときに、次のオペレーティングシステムのスキャンをサポートしています。この表には、各オペレーティングシステムのベンダーセキュリティアドバイザリのソースも示されています。

オペレーティングシステム	Version	ベンダーのセキュリティアドバイザリ
Alpine Linux (Alpine)	3.16	Alpine SecDB

オペレーティングシステム	Version	ベンダーのセキュリティアドバイザリ
Alpine Linux (Alpine)	3.17	Alpine SecDB
Alpine Linux (Alpine)	3.18	Alpine SecDB
Alpine Linux (Alpine)	3.19	Alpine SecDB
AlmaLinux	8	ALSA
AlmaLinux	9	ALSA
Amazon Linux (AL2)	AL2	ALAS
Amazon Linux 2023 (AL2023)	AL2023	ALAS
CentOS Linux (CentOS)	7	CESA
Debian Server (Buster)	10	DSA
Debian Server (Bullseye)	11	DSA
Debian Server (Bookworm)	12	DSA
Fedora	38	CVE
Fedora	39	CVE
OpenSUSE	15.5	CVE
Oracle Linux (Oracle)	7	ELSA
Oracle Linux (Oracle)	8	ELSA
Oracle Linux (Oracle)	9	ELSA
Photon OS	3	PHSA
Photon OS	4	PHSA
Photon OS	5	PHSA

オペレーティングシステム	Version	ベンダーのセキュリティアドバイザリ
Red Hat Enterprise Linux (RHEL)	7	RHSA
Red Hat Enterprise Linux (RHEL)	8	RHSA
Red Hat Enterprise Linux (RHEL)	9	RHSA
Rocky Linux	8	RLSA
Rocky Linux	9	RLSA
SUSE Linux Enterprise Server (SLES)	12.4	SUSE CVE
SUSE Linux Enterprise Server (SLES)	12.5	SUSE CVE
SUSE Linux Enterprise Server (SLES)	15.3	SUSE CVE
SUSE Linux Enterprise Server (SLES)	15.4	SUSE CVE
SUSE Linux Enterprise Server (SLES)	15.5	SUSE CVE
Ubuntu (Trusty)	14.04 (ESM)	USN, Ubuntu Pro
Ubuntu (Xenial)	16.04 (ESM)	USN, Ubuntu Pro
Ubuntu (Bionic)	18.04 (ESM)	USN, Ubuntu Pro
Ubuntu (Focal)	20.04 (LTS)	USN
Ubuntu (Jammy)	22.04 (LTS)	USN

オペレーティングシステム	Version	ベンダーのセキュリティアドバイザリ
Ubuntu (Mantic Minotaur)	23.10	USN

サポートされているプログラミング言語: Amazon ECR スキャン

Amazon Inspector は現在、Amazon ECR リポジトリでコンテナイメージをスキャンするときに、次のプログラミング言語をサポートしています。

- C#
- Go
- Java
- JavaScript
- PHP
- Python
- Ruby
- Rust

サポートされているランタイム: Amazon Inspector Lambda 標準スキャン

Amazon Inspector Lambda 標準スキャンは、現在、Lambda 関数をスキャンしてサードパーティーのソフトウェアパッケージの脆弱性を確認するときに、次のプログラミング言語をサポートしています。

- Java
  - java8
  - java8.al2
  - java11
  - java17
- Node.js
  - nodejs12.x

  - nodejs14.x
  - nodejs16.x
  - nodejs18.x
  - nodejs20.x
- Python
  - python3.7
  - python3.8
  - python3.9
  - python3.10
  - python3.11
- Go
  - go1.x
- Ruby
  - ruby2.7
  - ruby3.2
- .NET
  - .NET 6

サポートされているランタイム: Amazon Inspector Lambda コードスキャン

Amazon Inspector Lambda コードスキャンでは、Lambda 関数でコードの脆弱性をスキャンするときに、現在次のプログラミング言語がサポートされています。

- Java
  - java8
  - java8.al2
  - java11
  - java17
- Node.js
  - nodejs12.x

- nodejs14.x
- nodejs16.x
- nodejs18.x
- nodejs20.x
- Python
  - python3.7
  - python3.8
  - python3.9
  - python3.10
  - python3.11
- Ruby
  - ruby2.7
  - ruby3.2

終了オペレーティングシステム

次の表に記載されているオペレーティングシステムに対する標準ベンダーサポートは、ベンダーによって終了されました。表の「終了」列には、ベンダーがオペレーティングシステムの標準サポートを終了した日が示されています。

Amazon Inspector は以前、これらのオペレーティングシステムを完全にサポートしており、それらを実行している Amazon EC2 インスタンスと Amazon ECR コンテナイメージを引き続きスキャンします。しかし、ベンダーのポリシーに従い、オペレーティングシステムはパッチで更新されなくなり、多くの場合、新しいセキュリティアドバイザリもリリースされなくなります。さらに、影響を受けるオペレーティングシステムが標準サポートが終了すると、既存のセキュリティアドバイザリと検出情報をフィードから削除するベンダーもあります。結果として、Amazon Inspector は既知の CVE に関する検出結果の生成を停止する場合があります。Amazon Inspector が終了したオペレーティングシステムについて生成した検出結果については、情報提供のみを目的としてご利用ください。

セキュリティのベストプラクティスとして、また Amazon Inspector の適用範囲を継続するためにも、現在サポートされているバージョンのオペレーティングシステムに移行することをお勧めします。

終了したオペレーティングシステム: Amazon EC2 スキャン

オペレーティングシステム	Version	終了
Amazon Linux (AL1)	2012	2021 年 12 月 31 日
CentOS Linux (CentOS)	8	2021 年 12 月 31 日
Debian Server (Stretch)	9	2022 年 6 月 30 日
Fedora	35	2022 年 12 月 13 日
Fedora	36	2023 年 5 月 16 日
Fedora	37	2023 年 12 月 5 日
OpenSUSE	15.3	2022 年 12 月 1 日
openSUSE	15.4	2023 年 12 月 7 日
OpenSUSE Leap (SUSE Leap)	15.2	2021 年 12 月 1 日
Oracle Linux (Oracle)	6	2021 年 3 月 1 日
SUSE Linux Enterprise Server (SLES)	12	2019 年 7 月 1 日
SUSE Linux Enterprise Server (SLES)	12.1	2020 年 5 月 31 日
SUSE Linux Enterprise Server (SLES)	12.2	2021 年 3 月 31 日
SUSE Linux Enterprise Server (SLES)	12.3	2022 年 6 月 30 日
SUSE Linux Enterprise Server (SLES)	15	2019 年 12 月 31 日
SUSE Linux Enterprise Server (SLES)	15.1	2021 年 1 月 31 日

オペレーティングシステム	Version	終了
SUSE Linux Enterprise Server (SLES)	15.2	2021 年 12 月 31 日
Ubuntu (Groovy)	20.10	2021 年 7 月 22 日
Ubuntu (Hirsute)	21.04	2022 年 1 月 20 日
Ubuntu (Impish)	21.10	2022 年 7 月 31 日
Ubuntu (Kinetic)	22.10	July 20, 2023
Ubuntu (Lunar Lobster)	23.04	January 25, 2024
Windows Server	2012	2023 年 10 月 10 日
Windows Server	2012 R2	2023 年 10 月 10 日

終了したオペレーティングシステム: Amazon ECR スキャン

オペレーティングシステム	Version	終了
Alpine Linux (Alpine)	3.12	2022 年 5 月 1 日
Alpine Linux (Alpine)	3.13	2022 年 11 月 1 日
Alpine Linux (Alpine)	3.14	May 1, 2023
Alpine Linux (Alpine)	3.15	November 1, 2023
Amazon Linux (AL1)	2012	2021 年 12 月 31 日
CentOS Linux (CentOS)	8	2021 年 12 月 31 日
Debian Server (Stretch)	9	2022 年 6 月 30 日
Fedora	35	2022 年 12 月 13 日
Fedora	36	2023 年 5 月 16 日

オペレーティングシステム	Version	終了
OpenSUSE	15.3	2022 年 12 月 1 日
OpenSUSE	15.4	December 7, 2023
OpenSUSE Leap (SUSE Leap)	15.2	2021 年 12 月 1 日
Oracle Linux (Oracle)	6	2021 年 3 月 1 日
SUSE Linux Enterprise Server (SLES)	12	2019 年 7 月 1 日
SUSE Linux Enterprise Server (SLES)	12.1	2020 年 5 月 31 日
SUSE Linux Enterprise Server (SLES)	12.2	2021 年 3 月 31 日
SUSE Linux Enterprise Server (SLES)	12.3	2022 年 6 月 30 日
SUSE Linux Enterprise Server (SLES)	15	2019 年 12 月 31 日
SUSE Linux Enterprise Server (SLES)	15.1	2021 年 1 月 31 日
SUSE Linux Enterprise Server (SLES)	15.2	2021 年 12 月 31 日
Ubuntu (Groovy)	20.10	2021 年 7 月 22 日
Ubuntu (Hirsute)	21.04	2022 年 1 月 20 日
Ubuntu (Impish)	21.10	2022 年 7 月 31 日
Ubuntu (Kinetic)	22.10	July 20, 2023
Ubuntu (Lunar Lobster)	23.04	January 25, 2024

Amazon Inspector の非アクティブ化

Amazon Inspector コンソールまたは API を使用して AWS リージョン 、 で Amazon Inspector を非アクティブ化できます。Amazon Inspector を非アクティブ化するには、このトピックの最後にある指示に従ってください。のすべての Amazon Inspector スキャンを非アクティブ化すると AWS アカウント、このアカウントでは Amazon Inspector が自動的に非アクティブ化されます。さまざまなリソースのスキャンタイプを非アクティブ化する方法については、「Amazon Inspector による自動リソーススキャン」を参照してください。

アカウントの Amazon Inspector が非アクティブ化されると、そのリージョンのそのアカウントのすべてのスキャンタイプが非アクティブ化されます。さらに、そのリージョンのアカウントの Amazon Inspector スキャン設定、抑制ルール、フィルター、および検出結果がすべて削除されます。

Amazon Inspector が非アクティブ化されている間は、Amazon Inspector の使用に対して課金されません。Amazon Inspector を非アクティブ化した後、後で再度アクティブ化することができます。

> **Note**
>
> Amazon Inspector を非アクティブ化する前に、検出結果をエクスポートすることをお勧めします。詳細については、「Amazon Inspector からの調査結果レポートのエクスポート」を参照してください。

Amazon Inspector Amazon EC2 スキャンを無効にすると、Amazon Inspector が使用する次の SSM 関連付けが削除されます。

- `InspectorDistributor-do-not-delete`
- `InspectorInventoryCollection-do-not-delete`
- `InvokeInspectorSsmPlugin-do-not-delete`。 さらに、この関連付けを介してインストールされた Amazon Inspector SSM プラグインは、すべてのWindowsホストから削除されます。詳細については、「Windows インスタンスのスキャン」を参照してください。

前提条件

アカウントタイプによっては、Amazon Inspector を無効にする前に次のように追加の手順が必要になる場合があります。

- スタンドアロンの Amazon Inspector アカウントをお持ちの場合は、いつでも非アクティブ化できます。
- Amazon Inspector マルチアカウント環境のメンバーアカウントの場合、独自のサービスを非アクティブ化することはできません。サービスを無効にするには、お客様の組織の委任された管理者に連絡する必要があります。
- 委任された管理者の場合は、Amazon Inspector を非アクティブ化する前に、すべてのメンバーアカウントの関連付けを解除する必要があります。詳細については、「Amazon Inspector でのメンバーアカウントの関連付けを解除する」を参照してください。

> **Note**
>
> アカウントの関連付けを解除しても、そのアカウントの Amazon Inspector は無効になりません。代わりに、関連付けを解除したメンバーアカウントはスタンドアロンアカウントになります。

> **Note**
>
> Amazon Inspector を委任された管理者として非アクティブ化すると、組織の自動アクティブ化機能は無効になります。

Amazon Inspector を非アクティブ化する

Console

Amazon Inspector を非アクティブ化するには

1. https://console.aws.amazon.com/inspector/v2/home で Amazon Inspector コンソールを開きます。
2. ページの右上隅にある AWS リージョン セレクターを使用して、Amazon Inspector を非アクティブ化するリージョンを選択します。
3. ナビゲーションペインで [全般設定] を選択します。
4. [Inspector を非アクティブ化] を選択します。
5. 確認を求められたら、テキストボックスに「deactivate」と入力し、[Inspector を非アクティブ化] を選択します。

6. (推奨) Amazon Inspector を非アクティブ化するリージョンごとに、これらの手順を繰り返します。

API

無効化 APL オペレーションを実行します。リクエストには、非アクティブ化するアカウント ID を指定し、すべてのスキャンを非アクティブ化する `resourceTypes` の `EC2, ECR, LAMBDA` を指定すると、アカウントが非アクティブ化されます。

Amazon Inspector のクォータ

AWS アカウントには、リージョンごとに Amazon Inspector の次のクォータがあります。

リソース	デフォルト	コメント
抑制ルール	500	リージョンごとの AWS アカウントあたりの保存済み抑制ルールの最大数。 クォータの引き上げはリクエストできません。
Amazon EC2 ネットワークの検出結果	10,000	AWS アカウントあたりの Amazon EC2 ネットワーク検出結果の最大数。 クォータの引き上げはリクエストできません。
メンバーアカウント	10000	Amazon Inspector の委任管理者アカウントに関連付けられたメンバーアカウントの最大数。この制限は に基づいています。「 のクォータ AWS Organizations AWS Organizations」を参照してください。
CIS スキャン設定	500	CIS スキャン設定の最大数。

リソース	デフォルト	コメント
		クォータの引き上げはリクエストできません。

Amazon Inspector Classic に関連するクォータのリストについては、AWS 全般のリファレンス の「Amazon Inspector service quotas」を参照してください。

組織に関連するクォータのリストについては、AWS 全般のリファレンス の「Organizations service quotas」を参照してください。

リージョンとエンドポイント

Amazon Inspector の Amazon EC2 向けエージェントレススキャンはプレビューリリース中です。Amazon EC2 エージェントレススキャン機能の使用には、「AWS のサービス条件」の第 2 項 (「ベータ版とプレビュー」) が適用されます。

AWS リージョン Amazon Inspector が利用可能な を表示するには、「」のAmazon Inspector エンドポイント」を参照してくださいAmazon Web Services 全般のリファレンス。

Amazon Inspector スキャン API のエンドポイント

次の表は、Amazon Inspector スキャン API を呼び出すときに使用できるリージョンエンドポイントを示しています。API を使用する場合は、エンドポイントと、現在認証されているリージョンに対応する AWS リージョンを指定する必要があります。

Amazon Inspector スキャンのエンドポイントの命名規則は `inspector-scan.region.amazonaws.com` です。例えば、`us-west-2` で認証されている場合は、エンドポイント `inspector-scan.us-west-2.amazonaws.com` を使用して `inspector-scan` API を呼び出します。

リージョン名	リージョン	エンドポイント	プロトコル
米国東部 (オハイオ)	us-east-2	inspector-scan.us-east-2.amazonaws.com inspector-scan-fips.us-east-2.amazonaws.com	HTTPS
米国東部 (バージニア北部)	us-east-1	inspector-scan.us-east-1.amazonaws.com	HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
		inspector-scan-fips.us-east-1.amazonaws.com	
米国西部 (北カリフォルニア)	us-west-1	inspector-scan.us-west-1.amazonaws.com inspector-scan-fips.us-west-1.amazonaws.com	HTTPS
米国西部 (オレゴン)	us-west-2	inspector-scan.us-west-2.amazonaws.com inspector-scan-fips.us-west-2.amazonaws.com	HTTPS
アフリカ (ケープタウン)	af-south-1	inspector-scan.af-south-1.amazonaws.com	HTTPS
アジアパシフィック (香港)	ap-east-1	inspector-scan.ap-east-1.amazonaws.com	HTTPS
アジアパシフィック (ジャカルタ)	ap-southeast-3	inspector-scan.ap-southeast-3.amazonaws.com	HTTPS
アジアパシフィック (ムンバイ)	ap-south-1	inspector-scan.ap-south-1.amazonaws.com	HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
アジアパシフィック (大阪)	ap-northeast-3	inspector-scan.ap-northeast-3.amazonaws.com	HTTPS
アジアパシフィック (ソウル)	ap-northeast-2	inspector-scan.ap-northeast-2.amazonaws.com	HTTPS
アジアパシフィック (シンガポール)	ap-southeast-1	inspector-scan.ap-southeast-1.amazonaws.com	HTTPS
アジアパシフィック (シドニー)	ap-southeast-2	inspector-scan.ap-southeast-2.amazonaws.com	HTTPS
アジアパシフィック (東京)	ap-northeast-1	inspector-scan.ap-northeast-1.amazonaws.com	HTTPS
カナダ (中部)	ca-central-1	inspector-scan.ca-central-1.amazonaws.com	HTTPS
欧州 (フランクフルト)	eu-central-1	inspector-scan.eu-central-1.amazonaws.com	HTTPS
欧州 (アイルランド)	eu-west-1	inspector-scan.eu-west-1.amazonaws.com	HTTPS
欧州 (ロンドン)	eu-west-2	inspector-scan.eu-west-2.amazonaws.com	HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
ヨーロッパ (ミラノ)	eu-south-1	inspector-scan.eu-south-1.amazonaws.com	HTTPS
欧州 (パリ)	eu-west-3	inspector-scan.eu-west-3.amazonaws.com	HTTPS
欧州 (ストックホルム)	eu-north-1	inspector-scan.eu-north-1.amazonaws.com	HTTPS
欧州 (チューリッヒ)	eu-central-2	inspector-scan.eu-central-2.amazonaws.com	HTTPS
中東 (バーレーン)	me-south-1	inspector-scan.me-south-1.amazonaws.com	HTTPS
南米 (サンパウロ)	sa-east-1	inspector-scan.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud （米国東部）	us-gov-east-1	inspector-scan.us-gov-east-1.amazonaws.com inspector-scan-fips.us-gov-east-1.amazonaws.com	HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
AWS GovCloud （米国西部）	us-gov-west-1	inspector-scan.us-gov-west-1.amazonaws.com inspector-scan-fips.us-gov-west-1.amazonaws.com	HTTPS

リージョン固有機能の可用性

このセクションでは、 AWS リージョンで利用可能な Amazon Inspector の機能について説明します。

Amazon EC2 リージョンのエージェントレス EC2 スキャン

次の表は、Amazon EC2 のエージェントレススキャン AWS リージョン が現在利用可能な を示しています。

リージョン名	リージョンコード
米国東部 (バージニア北部)	us-east-1
米国西部 (オレゴン)	us-west-2
欧州 (アイルランド)	eu-west-1

Lambda コードスキャンリージョン

次の表は、Lambda AWS リージョン コードスキャンが現在利用可能な を示しています。

リージョン名	リージョンコード
米国東部 (バージニア北部)	us-east-1
米国西部 (オレゴン)	us-west-2

リージョン名	リージョンコード
米国東部 (オハイオ)	us-east-2
アジアパシフィック (シドニー)	ap-southeast-2
アジアパシフィック (東京)	ap-northeast-1
欧州 (フランクフルト)	eu-central-1
欧州 (アイルランド)	eu-west-1
欧州 (ロンドン)	eu-west-2
欧州 (ストックホルム)	eu-north-1
アジアパシフィック (シンガポール)	ap-southeast-1

AWS GovCloud (US) リージョン

最新情報については、「AWS GovCloud (US) ユーザーガイド」の「Amazon Inspector」を参照してください。

「Amazon Inspector ユーザーガイド」のドキュメント履歴

次のテーブルに、Amazon Inspector の前回のリリース以後に行われたドキュメントの重要な変更を示します。このドキュメントの更新に関する通知を受け取るには、RSS フィードにサブスクライブできます。

変更	説明	日付
更新された機能	Amazon Inspector は、クローズされた検出結果の保持期間を 30 日から 7 日に更新します。詳細については、Amazon Inspector の検出結果について」を参照してください。	2024 年 2 月 12 日
更新された機能	Amazon Inspector では AmazonInspector2ServiceRolePolicy ポリシーに新しいステートメントが追加されました。新しいステートメントでは、Amazon Inspector がインスタンスの CIS スキャンを開始できます。	2024 年 1 月 23 日
新しいポリシー	Amazon Inspector に新しいポリシー が追加されました。これは、インスタンスプロファイルの AmazonInspector2ManagedCisPolicyの一部として使用して、インスタンスで CIS スキャンを許可できます。	2024 年 1 月 23 日
新機能	Amazon Inspector は、コンテナイメージをプルするときに ECR 再スキャン期間を更新するようになりました。プッ	2024 年 1 月 23 日

	シュ日またはプル日に基づいて再スキャン期間を変更するには、「ECR 再スキャン期間の設定」を参照してください。	
新機能	Amazon Inspector はEC2 インスタンスで Center for Internet Security (CIS) スキャンを実行できるようになりました。詳細については、Amazon Inspector CIS スキャン」を参照してください。	2024 年 1 月 23 日
新機能	Amazon Inspector で CI/CD パイプライン内のコンテナイメージをスキャンできるようになりました。詳細については、「Amazon Inspector と CI/CD の統合」を参照してください。	2023 年 11 月 30 日
新しいポリシー	Amazon Inspector には、Amazon Inspector が エージェントレススキャンのために EC2 インスタンスから Amazon EBS スナップショットをスキャンすることを許可する新しいポリシーが追加されました。ポリシーの詳細については、「エージェントレススキャン」を参照してください。	2023 年 11 月 27 日

新機能	Amazon Inspector では、エージェントレススキャンによって、サポート対象 Linux Amazon EC2 インスタンスの SSM エージェントなしでのスキャンがサポートされるようになりました。詳細については、「エージェントレススキャン」を参照してください。	2023 年 11 月 27 日
新しくサポートされたリソース	Amazon Inspector は MacOS Amazon EC2 インスタンスのスキャンをサポートするようになりました。サポートされている MacOS バージョンについては、「Supported operating systems: Amazon EC2 scanning」を参照してください。	2023 年 10 月 5 日
新しいリージョン	Amazon Inspector は現在、アジアパシフィック (ジャカルタ)、アフリカ (ケープタウン)、アジアパシフィック (大阪)、および欧州 (チューリッヒ) で利用可能です。	2023 年 9 月 29 日
新機能	除外タグを使用して Amazon Inspector のスキャンから EC2 インスタンスを除外できるようになりました。	2023 年 9 月 14 日

新機能	Amazon Inspector には、Elastic Load Balancing ターゲットグループの一部である Amazon EC2 インスタンスのネットワーク設定をスキャンできるようにする新しいアクセス許可が追加されました。	2023 年 8 月 31 日
新機能	Amazon Inspector では、パッケージ脆弱性の検出結果に関する脆弱性インテリジェンスの詳細を提供できるようになりました。	2023 年 7 月 31 日
更新された機能	Amazon Inspector には、読み取り専用ユーザーがリソースのソフトウェア部品表 (SBOM) をエクスポートできる新しいアクセス許可が追加されました。	2023 年 6 月 29 日
新機能	Amazon Inspector によってスキャンされているリソースの SBOM をエクスポートできるようになりました。	2023 年 6 月 13 日
新機能	Lambda コードスキャンの一般提供が開始されました。Lambda コードスキャンの検出結果で特定されたコードを暗号化できる新機能が追加されました。さらに、Lambda コードスキャンにより、コードの修復と書き換えが提案されるようになりました。	2023 年 6 月 13 日

更新された機能	Amazon Inspector では AmazonInspector2ReadOnlyAccess ポリシーに新しいステートメントが追加されました。新しいステートメントにより、読み取り専用ユーザーは、自分のアカウントの Lambda コードスキャンのステータスと検出結果の詳細を取得できます。	2023 年 5 月 2 日
新機能	Amazon Inspector に脆弱性データベース検索が追加されました。これにより、Amazon Inspector が特定の CVE を対象としているかどうかを確認できます。	2023 年 5 月 1 日
更新された機能	Amazon Inspector は、Lambda スキャンをアクティブ化するときに Amazon Inspector がアカウントに AWS CloudTrail サービスにリンクされたチャネルを作成できるようにする新しいアクセス許可を AmazonInspector2ServiceRolePolicy ポリシーに追加しました。これにより、Amazon Inspector はアカウント内の CloudTrail イベントをモニタリングできます。	2023 年 4 月 30 日

更新された機能	Amazon Inspector では AmazonInspector2FullAccess ポリシーに新しいステートメントが追加されました。新しいステートメントにより、ユーザーは Lambda コードスキャンのコード脆弱性検出結果を取得できます。	2023 年 4 月 17 日
更新された機能	Amazon Inspector では AmazonInspector2ServiceRolePolicy ポリシーに新しいステートメントが追加されました。新しいステートメントでは、Amazon Inspector は、Amazon EC2 の詳細検査用に定義したカスタムパスに関する情報を Amazon EC2 Systems Manager に送信できます。 Amazon EC2	2023 年 4 月 17 日
新機能	Amazon Inspector は、Amazon Inspector のディープインスペクションの形式で Linux EC2 インスタンスのサポートを追加します。これにより、アプリケーションプログラミング言語パッケージのパッケージの脆弱性がインスタンスでスキャンされます。 Amazon Inspector	2023 年 4 月 17 日

更新された機能	Amazon Inspector では AmazonInspector2ServiceRolePolicy ポリシーに新しいステートメントが追加されました。新しいステートメントにより、Amazon Inspector は AWS Lambda 関数内のデベロッパーコードのスキャンをリクエストし、Amazon CodeGuru Security からスキャンデータを受信できます。さらに、Amazon Inspector には IAM ポリシーを確認するためのアクセス許可が追加されました。Amazon Inspector はこの情報を使用して Lambda 関数のコード脆弱性をスキャンします。	2023 年 2 月 28 日
新機能	Amazon Inspector では、Lambda コードスキャンという形で Lambda 関数のサポートが追加され、Lambda 関数の開発者コードをスキャンしてセキュリティの脆弱性を調べます。	2023 年 2 月 28 日

更新された機能	Amazon Inspector では AmazonInspector2ServiceRolePolicy ポリシーに新しいステートメントが追加されました。新しいステートメントにより、Amazon Inspector は関数 AWS Lambda が最後に呼び出された日時 CloudWatch に関する情報を取得できます。はこの情報を使用して、過去 90 日間にアクティブだった環境内の Lambda 関数にスキャンを絞り込みます。	2023 年 2 月 20 日
更新された機能	Amazon Inspector では AmazonInspector2ServiceRolePolicy ポリシーに新しいステートメントが追加されました。新しいステートメントにより、Amazon Inspector では AWS Lambda 関数に関する情報を取得できます。Amazon Inspector はこの情報を使用して Lambda 関数をスキャンし、セキュリティの脆弱性がないか調べます。	2022 年 11 月 28 日
新機能	Amazon Inspector は、AWS Lambda 関数 のスキャンのサポートを追加します。	2022 年 11 月 28 日

更新された内容	Amazon Inspector から Amazon Simple Storage Service (Amazon S3) バケットに調査結果レポートをエクスポートする手順、ポリシー例、およびヒントを追加しました。	2022 年 10 月 14 日
新しいコンテンツ	Amazon Inspector コンソールを使用して AWS 環境の Amazon Inspector カバレッジを評価する方法に関する情報を追加しました。 Amazon Inspector この情報には、環境内の個々のリソースのステータス値についての説明が含まれます。	2022 年 10 月 7 日
新機能	Amazon Inspector では、パッケージの脆弱性を修復する方法について、さらなる詳細を提供するようになりました。検出結果の詳細に新しいフィールドが追加されました。新しいフィールドには、パッケージの更新によって修正が可能かどうかについてのコンテキストが表示されます。修正が入手可能な場合は、検出結果の [推奨される修復] セクションに、修正を行うために実行できるコマンドが表示されます。	2022 年 9 月 2 日

更新された機能	Amazon Inspector には AmazonInspector2ServiceRolePolicy ポリシーに新しいアクションが追加されました。新しいアクションにより、Amazon Inspector は SSM 関連付けの実行を記述できます。Amazon Inspector では、`AmazonInspector2` 所有の SSM ドキュメントとの SSM 関連付けを作成、更新、削除、開始できるように、リソーススコープも追加されました。	2022 年 8 月 31 日
新機能	Amazon Inspector は Windows インスタンスのスキャンをサポートするようになりました。Amazon Inspector では、サポートされている Windows オペレーティングシステムを実行している SSM マネージドインスタンスをスキャンできるようになりました。Windows ホストのスキャンは、Amazon Inspector によって自動的に作成された新しい SSM 関連付けを通じてインストールおよび呼び出される Amazon Inspector SSM プラグインによって実行されます。	2022 年 8 月 31 日

更新された機能	Amazon Inspector は、AmazonInspector2ServiceRolePolicyポリシーのリソーススコープを更新して、Amazon Inspector が他の AWS パーティションでソフトウェアインベントリを収集できるようにしました。	2022 年 8 月 12 日
更新された機能	AmazonInspector2ServiceRolePolicy ポリシーでは、Amazon Inspector がアクションのリソーススコープを再構築し、Amazon Inspector が SSM 関連付けを作成、削除、および更新できるようにしました。	2022 年 8 月 10 日

新機能	Amazon Inspector では ECR 自動再スキャン期間設定の変更がサポートされるようになりました。Amazon ECR 自動再スキャン期間設定で、Amazon Inspector がリポジトリにプッシュされたイメージを継続的にモニタリングする期間を決定します。イメージがスキャン期間より古い場合、Amazon Inspector はイメージをスキャンせず、そのイメージに関する既存の検出結果をすべて終了します。すべての新規アカウントでは、ECR 自動再スキャン期間が自動的にライフタイム期間に設定されます。以前に作成されたアカウントの ECR 自動再スキャン期間は 30 日でしたが、スキャンの期間を 30 日、180 日、またはライフタイム、から選択できるようになりました。	2022 年 6 月 25 日
新しい機能	Amazon Inspector は、Amazon Inspector AmazonInspector2ReadOnlyAccess機能への読み取り専用アクセスを許可する新しい AWS 管理ポリシーを追加しました。 Amazon Inspector	2022 年 1 月 21 日

一般提供	これは、「Amazon Inspector ユーザーガイド」の初回一般リリースです。	2021 年 11 月 29 日

AWS 用語集

最新の AWS 用語については、「 AWS の用語集 リファレンス」のAWS 「 用語集」を参照してください。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。