AWS IoT Device Defender デベロッパーガイド

AWS IoT Device Defender

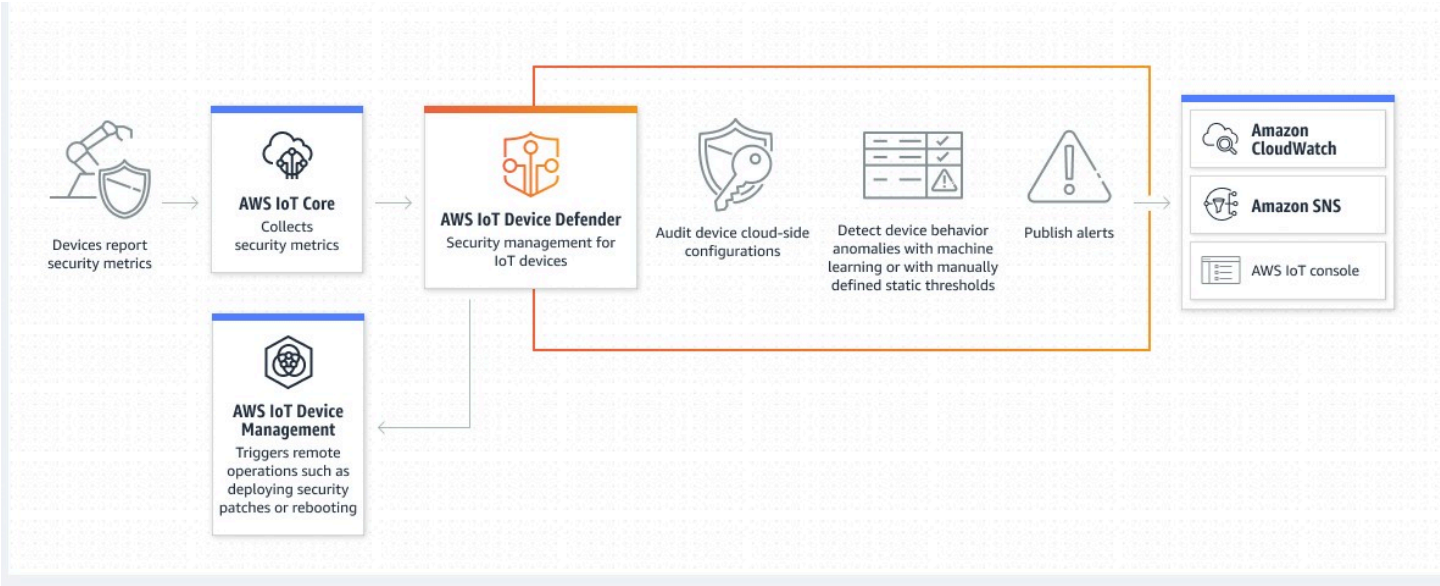
AWS IoT Device Defender: AWS IoT Device Defender デベロッパーガイド

Table of Contents

AWS IoT Device Defender とは

セキュリティおよびモニタリングサービスである AWS IoT Device Defender を使用して、デバイスの設定の監査、接続されたデバイスのモニタリング、およびセキュリティリスクの緩和ができます。このサービスで AWS IoT Device Defender を使用すると、AWS IoT デバイスのフリートで一貫性のあるセキュリティポリシーを適用し、デバイスが侵害された場合にはすばやく応答することができます。IoT フリートは、多様な機能を持ち、存続期間が長く、地理的に分散される多数のデバイスで構成されることがあります。このような特性によってフリートのセットアップが複雑になり、エラーを起こしやすくなります。デバイスの計算能力、メモリ、ストレージの機能には制約があるため、デバイス自体での暗号化や他の形式のセキュリティの使用が制限されます。

多く場合、デバイスは既知の脆弱性を持つソフトウェアを使用しています。これらの要因により、IoT フリートはハッカーの魅力的なターゲットとなり、デバイスフリートを継続的に保護することが困難になります。AWS IoT Device Defender は、セキュリティの問題とベストプラクティスからの逸脱を特定するツールを提供することで、これらの課題に対処します。AWS IoT Device Defender は、デバイスフリートを監査して、セキュリティのベストプラクティスに準拠していることを確認し、デバイスでの異常な動作を検出できます。次の図は、AWS IoT Device Defender の基本アーキテクチャと、それが AWS IoT Core、Amazon CloudWatch、Amazon SNS などのサービスにどのように関連しているかを示しています。



トピック

- AWS IoT Device Defender を初めてお使いになる方向けの情報
- AWS IoT Device Defender の働き

AWS IoT Device Defender を初めてお使いになる方向けの情報

AWS IoT Device Defender を初めて使用する方には、次のセクションを初めに読むことをお勧めします。

AWS IoT Device Defender の働き

AWS IoT Device Defender は、IoT デバイスのフリートを保護するのに役立つ完全マネージド型のセキュリティおよびモニタリングサービスです。AWS IoT Device Defender は、デバイスに関連付けられた IoT リソースを監査して、セキュリティのベストプラクティスに準拠していることを確認します。Audit チェックは、セキュリティリスクが検出された場合にアラートを送信し、問題を軽減するために関連情報を提供します。また、AWS IoT Device Defender は、クラウド側やデバイス側のセキュリティメトリクスを継続的にモニタリングし、予期しないデバイス動作を検出して、侵害された可能性のあるデバイスを特定します。オンデマンドで、またはスケジュールに基づいて監査チェックを起動して、IoT デバイスの設定を評価できます。

AWS IoT Device Defender は AWS IoT Core と連携して、デバイスインタラクションのコンテキストを組み込むことで、監査チェックの精度を向上させます。AWS IoT Device Defender は、接続されたデバイスから値の高いセキュリティメトリクスを収集して分析し、異常な動作を検出します。Rules Detect を使用すると、メトリクスデータはユーザー定義の動作に対して継続的に評価されます。ML Detect を使用すると、メトリクスデータは、異常を識別するために自動的に構築された機械学習 (ML) モデルによって継続的に評価されます。

スケジュールされた監査タスクの結果と検出されたデバイスアクティビティの異常は、AWS IoT コンソールと AWS IoT Device Defender API に発行されます。これらは Amazon CloudWatch からアクセスできます。さらに、セキュリティダッシュボードとの統合や自動修復ワークフローの開始のために結果を Amazon SNS トピックに送信するように AWS IoT Device Defender を設定できます。

AWS IoT Device Defender は、次のような幅広いユースケースをサポートしています。

- デバイスの保護: デバイス関連のリソースを AWS IoT セキュリティのベストプラクティスに照らして監査することで、デバイスの脆弱性を検出できます。AWS IoT Device Defender 監査は、デバイスに対するリスクを特定、発見し、セキュリティ対策が講じられていることを確認するのに役立ちます。
- 異常なデバイス動作の検出: 接続パターンの変化を特定し、不正なエンドポイントとのデバイス通信を明らかにし、インバウンドおよびアウトバウンドのデバイストラフィックパターンの変化を特定できます。
- リスク軽減のためのインサイトの取得: Audit の結果または Detect アラームで明らかになった問題を緩和するためのアクションを実行できます。
- デバイスセキュリティの保持と維持: Audit チェックと Detect チェックからのインサイトを使用して、潜在的なセキュリティ違反を診断して修正できます。
- デバイスセキュリティの強化: 正しく設定されていないデバイスを区別し、デバイスフリートの状態を調べ、予期しないデバイスの動作メトリクスを特定できます。

AWS IoT Device Defender の機能

AWS IoT Device Defender の主な機能のいくつかを以下に示します。

主な機能

監査	AWS IoT Device Defender は、「IAM ユーザーガイド」の AWS IoT におけるセキュリティのベストプラクティスに照らして、デバイス関連のリソースを監査します。AWS IoT Device Defender は、セキュリティのベストプラクティスに準拠していない設定 (過度に許容されているポリシーで 1 つのデバイスが他の多くの

	デバイスのデータを読み取って更新できるものなど) を報告します。
Rules Detect	AWS IoT Device Defender は、デバイスと AWS IoT Core から値の高いセキュリティメトリクスを継続的にモニタリングすることで、侵害を示す可能性のある異常なデバイス動作を検出します。これらのメトリクスの動作 (ルール) を設定することで、デバイスのグループの通常のデバイス動作を指定できます。AWS IoT Device Defender は、これらのメトリクスについて報告された各データポイントを、ユーザー定義の動作 (ルール) と照らしてモニタリングおよび評価し、異常が検出された場合に警告します。
ML 検出	AWS IoT Device Defender は、過去 14 日間における 6 つのクラウド側のメトリクスと 7 つのデバイス側のメトリクスのデバイスデータを使用して、機械学習 (ML) モデルでデバイス動作を自動的に設定します。次に、モデルを毎日 (モデルのトレーニングに十分なデータがある限り) 再トレーニングし、最初のモデルの構築から 14 日後までの最新のデバイス動作を更新します。AWS IoT Device Defender は、これらのメトリクスの異常なデータポイントを ML モデルでモニタリングおよび識別し、異常が検出された場合はアラームを作動させます。
アラート	AWS IoT Device Defender は、AWS IoT コンソール、Amazon CloudWatch、および Amazon SNS にアラームを発行します。

緩和策	AWS IoT Device Defender は、デバイスメタデータ、デバイス統計、デバイスの履歴アラートなど、デバイスに関するコンテキスト情報と履歴情報を提供することで、問題を調査するために使用できます。また、AWS IoT Device Defender 組み込み型の緩和アクションを使用して、モノのグループへのモノの追加、デフォルトのポリシーバージョンの置き換え、デバイス証明書の更新など、Audit アラームと Detect アラームの緩和ステップを実行することもできます。

AWS IoT Device Defender の開始方法

AWS IoT Device Defender の使用を開始するには、次のチュートリアルを参照してください。

- セットアップ
- ML Detect ガイド
- Audit ガイド
- AWS IoT Device Defender 監査結果をいつどのように表示するかをカスタマイズする

関連サービス

- AWS IoT Greengrass: AWS IoT Greengrass は、デバイスの動作を継続的にモニタリグするために、AWS IoT Device Defender との事前構築された統合を提供します。
- AWS IoT Device Management: AWS IoT Device Management フリートのインデックス作成を使用して、AWS IoT Device Defender Detect 違反をインデックス化、検索、および集計できます。

AWS IoT Device Defender へのアクセス

AWS IoT Device Defender コンソールまたは API を使用して AWS IoT Device Defender にアクセスできます。

AWS IoT Device Defender の料金

AWS IoT Device Defender では、実際に使用した分に対してのみお支払いいただきます。最低料金やサービス使用義務はありません。ただし、Audit 機能と Detect 機能については別途請求されます。Audit 料金は、1 デバイスあたり、1 月あたりの料金です。Audit を有効にすると、月あたりのアクティブなデバイスプリンシパルの数に基づいて課金されます。したがって、監査チェックを追加または削除しても、この機能を使用する際の毎月の請求書には影響しません。AWS 料金計算ツールを使用して、AWS IoT Device Defender およびアーキテクチャのコストを 1 回の見積りで計算できます。

- AWS 料金計算機

AWS IoT Device Defender の開始方法

次のチュートリアルを使用して、AWS IoT Device Defender を操作できます。

トピック

セットアップ

AWS IoT Device Defender を初めて使用する場合は、事前に以下のタスクをすべて実行してください。

トピック

AWS アカウントへのサインアップ

AWS アカウント がない場合は、以下のステップを実行して作成します。

AWS アカウントにサインアップするには

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、電話またはテキストメッセージを受け取り、電話キーパッドで検証コードを入力します。

   AWS アカウント にサインアップすると、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、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用してルートユーザーアクセスが必要なタスクを実行してください。

サインアップ処理が完了すると、AWS からユーザーに確認メールが送信されます。https://aws.amazon.com/ の [マイアカウント] 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。

管理アクセスを持つユーザーを作成する

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

AWS アカウントのルートユーザーをセキュリティで保護する

1. [ルートユーザー] を選択し、AWS アカウント のメールアドレスを入力して、アカウント所有者として AWS Management Console 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」で AWS アカウントのルートユーザーの仮想 MFA デバイスを有効にする方法 (コンソール) を確認してください。

管理アクセスを持つ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、ユーザーに管理アクセスを付与します。

   IAM アイデンティティセンターディレクトリ をアイデンティティソースとして使用するチュートリアルについては、「AWS IAM Identity Center ユーザーガイド」の「 Configure user access with the default IAM アイデンティティセンターディレクトリ」を参照してください。

管理アクセス権を持つ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 アイデンティティセンター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 User Guide」の「Signing in to the AWS access portal」を参照してください。

追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「権限設定を作成する」を参照してください。
2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「グループの結合」を参照してください。

これらのタスクでは、AWS アカウント とユーザーをそのアカウントの管理者権限で作成します。

監査ガイド

このチュートリアルでは、定期的な監査の設定、アラームの設定、監査結果の確認、および監査で検出された問題の緩和の方法について説明します。

トピック

前提条件

このチュートリアルを完了するには、以下が必要です。

- AWS アカウント。これを持っていない場合は、設定を参照してください。

監査チェックを有効化する

次の手順では、アカウントおよびデバイスの設定とポリシーを調べて、セキュリティ対策が実装されていることを確認します。このチュートリアルでは、すべての監査チェックを有効にすることとしていますが、任意のチェックを選択できます。

監査料金は、1 か月あたりのデバイス数 (AWS IoT に接続されたフリートデバイス) あたりの料金です。したがって、監査チェックを追加または削除しても、この機能を使用する際の毎月の請求書には影響しません。

1. AWS IoT コンソールを開きます。ナビゲーションペインで、[セキュリティ] を展開し、[イントロダクション] を選択します。
2. [AWS IoT セキュリティ監査を自動化] を選択します。監査チェックは自動的にオンになります。
3. [監査] を展開し、[設定] を選択して監査チェックを表示します。監査チェックの名前を選択すると、監査チェックの目的について確認できます。監査チェックの詳細については、[監査チェック」を参照してください。
4. (オプション) 使用したいロールが既にある場合は、[サービスのアクセス権限を管理] を選択し、一覧からロールを選択して、[更新] を選択します。

監査結果を表示する

次の手順では、監査結果を表示する方法を示します。このチュートリアルでは、監査チェックを有効化する チュートリアルで設定された監査チェックの監査結果を確認します。

監査結果を表示するには

1. AWS IoT コンソールを開きます。ナビゲーションペインで、[セキュリティ]、[監査] の順に展開し、[結果] を選択します。
2. 調査する監査スケジュールの [名前] を選択します。

3. [非準拠のチェック] の [軽減] で情報ボタンを選択すると、コンプライアンス非準拠の理由が表示されます。不適合チェックを適合にする方法のガイダンスについては、「監査チェック項目」を参照してください。

監査結果の緩和アクションの作成

以下の手順では、AWS IoT Device Defender Audit Mitigation Action を作成して、AWS IoT ログ記録を有効にします。各監査チェックには、マッピングされた緩和アクションがあります。これは、修正する監査チェック用に選択する [Action type] (アクションの種類) に影響します。詳細については、緩和アクションを参照してください。

AWS IoT コンソールを使用して緩和アクションを作成するには

1. AWS IoT コンソールを開きます。ナビゲーションペインで、[セキュリティ]、[検出] の順に展開し、[軽減アクション] を選択します。
2. [Mitigation actions] (緩和アクション) ページで、[Create] (作成) を選択します。
3. [新しい軽減アクションを作成する] ページの [アクション名] に、*EnableErrorLoggingAction* などの軽減アクションの一意の名前を入力します。
4. [アクションタイプ] で、[AWS IoT ログ記録を有効にする] を選択します。
5. [アクセス権限] で、[ロールの作成] を選択します。[ロール名] で、*[IoTMitigationActionErrorLoggingRole]* を使用します。続いて、[作成] を選択します。
6. [パラメータ] の [ログ記録のロール] で、[IoTMitigationActionErrorLoggingRole] を選択します。[ログレベル] で、[Error] を選択します。
7. [Create] (作成) を選択します。

監査所見に緩和措置を適用する

次の手順では、緩和アクションを監査結果に適用する方法を示します。

不適合の監査結果を緩和するには

1. AWS IoT コンソールを開きます。ナビゲーションペインで、[セキュリティ]、[監査] の順に展開し、[結果] を選択します。
2. 対応する監査結果を選択します。
3. 結果を確認します。

4. [Start mitigation actions] (緩和アクションの開始) を選択します。
5. [ログ記録が無効] で、以前に作成した軽減アクション `EnableErrorLoggingAction` を選択します。問題に対処するために、非準拠の検出結果ごとに適切なアクションを選択できます。
6. [理由コードを選択する] で、監査チェックで返された理由コードを選択します。
7. [タスクを開始] を選択します。軽減アクションが実行されるまで数分かかる場合があります。

緩和アクションが機能したことを確認するには

1. AWS IoT コンソールのナビゲーションペインで、[設定] を選択します。
2. [サービスログ] で、[ログレベル] が `Error (least verbosity)` であることを確認します。

AWS IoT Device Defender 監査 IAM ロールの作成 (オプション)

次の手順では、AWS IoT への AWS IoT Device Defender 読み取りアクセス権を提供する AWS IoT Device Defender 監査 IAM ロールを作成します。

AWS IoT Device Defender のサービスロールを作成するには (IAM コンソール)

1. AWS Management Console にサインインして、IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. IAM コンソールのナビゲーションペインで、[ロール]、[ロールを作成] を選択します。
3. AWS のサービス のロールタイプを選択します。
4. [その他の AWS のサービスのユースケース] で、[AWS IoT] を選択し、[IoT - Device Defender Audit] を選択します。
5. [Next] を選択します。
6. (オプション) アクセス許可の境界を設定します。このアドバンスド機能は、サービスロールで使用できますが、サービスにリンクされたロールではありません。

   [Permissions boundary] (アクセス許可の境界) セクションを展開し、[Use a permissions boundary to control the maximum role permissions] (アクセス許可の境界を使用して、ロールのアクセス許可の上限を設定する) を選択します。IAM には、あなたのアカウント内の AWS 管理ポリシーとカスタマー管理ポリシーのリストがあります。アクセス許可の境界に使用するポリシーを選択するか、[ポリシーを作成] を選択して新しいブラウザタブを開き、新しいポリシーをゼロから作成します。詳細については、『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。ポリシーを作成したら、そのタブを閉じて元のタブに戻り、アクセス許可の境界として使用するポリシーを選択します。

7. [Next] を選択します。
8. このロールの目的を識別しやすくするロール名を入力します。ロール名は AWS アカウント アカウント内で一意である必要があります。大文字と小文字は区別されません。例えば、**PRODROLE** と **prodrole** というロール名を両方作成することはできません。多くのエンティティによりロールが参照されるため、作成後にロール名を変更することはできません。
9. (オプション) [Description (説明)] には、新しいロールの説明を入力します。
10. [Step 1: Select trusted entities] (ステップ 1: 信頼済みエンティティの選択) または [Step 2: Select permissions] (ステップ 2: 権限の選択) のセクションで [Edit] (編集) を選択し、ロールのユースケースと権限を変更します。
11. (オプション) タグをキーバリューペアとしてアタッチして、メタデータをユーザーに追加します。IAM でのタグの使用に関する詳細については、「IAM ユーザーガイド」の「IAM リソースにタグを付ける」を参照してください。
12. ロール情報を確認し、ロールの作成 を選択します。

SNS 通知を有効にする (オプション)

次の手順では、Amazon SNS (SNS) 通知を有効にして、監査で不適合のリソースが特定されたときに警告します。このチュートリアルでは、監査チェックを有効化する チュートリアルで有効にした監査チェックの通知を設定します。

1. まだ設定していない場合は、AWS Management Console 経由で SNS へのアクセスを提供するポリシーをアタッチします。これを行うには、「IAM ユーザーガイド」の「IAM ユーザーグループへのポリシーのアタッチ」の手順に従って、AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction ポリシーを選択します。
2. AWS IoT コンソールを開きます。ナビゲーションペインで、[セキュリティ]、[監査] の順に展開し、[設定] を選択します。
3. [Device Defender の監査設定] ページの下部にある、[SNS アラートを有効にする] を選択します。
4. [有効] を選択します。
5. [トピック]、[新しいトピックを作成] の順に選択します。トピックに *IoTDDNotifications* という名前を付け、[作成] を選択します。[ロール] には、「AWS IoT Device Defender 監査 IAM ロールの作成 (オプション)」で作成したロールを選択します。

6. [Update] (更新) を選択します。
7. Ops プラットフォームで E メールやテキストを Amazon SNS 経由で受信したい場合は、「ユーザー通知に Amazon Simple Notification Service を使用する」を参照してください。

(オプション) ログ記録を有効化する

この手順では、AWS IoT が CloudWatch Logs に情報を記録できるようにする方法について説明します。これにより、監査結果を表示できます。ログ記録を有効にすると、料金が発生する場合があります。

ログ記録を有効にするには

1. AWS IoT コンソールを開きます。ナビゲーションペインで [設定] を選択します。
2. [ログ] で、[ログの管理] を選択します。
3. [新しいロール] で、[ロールの作成] をクリックします。ロールに *AWSIoTLoggingRole* という名前を付け、[作成] を選択します。ポリシーは自動的にアタッチされます。
4. [ログレベル] で、[デバッグ (最も冗長)] を選択します。
5. [Update] (更新) を選択します。

ML Detect ガイド

この開始方法のガイドでは、機械学習 (ML) を使用して、デバイスからの履歴メトリクスデータに基づいて想定される動作のモデルを作成する ML Detect Security Profile を作成します。ML Detect による ML モデルの作成中に、進捗状況を監視できます。ML モデルの構築後、アラームを継続的に表示および調査し、特定された問題を緩和できます。

ML Detect ならびに API コマンドおよび CLI コマンドの詳細については、「ML 検出」を参照してください。

この章には、以下のセクションが含まれています。

- 前提条件
- コンソールで ML Detect を使用する方法
- CLI で ML Detect を使用する方法

前提条件

- AWS アカウント。これを持っていない場合は、設定を参照してください。

コンソールで ML Detect を使用する方法

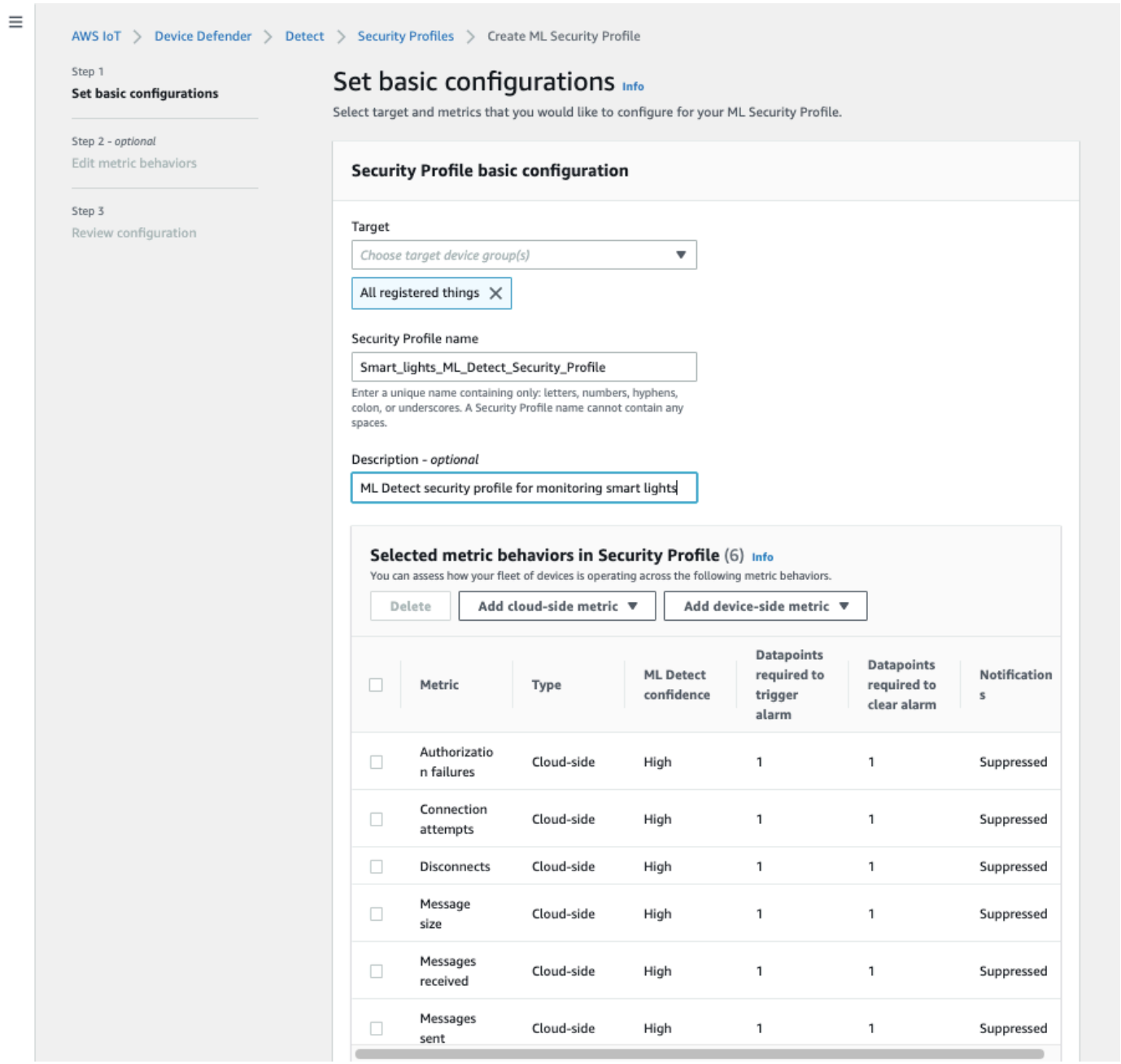
チュートリアル

- ML Detect を有効化する
- ML モデルのステータスを監視する
- ML Detect アラームを確認する
- ML アラームを微調整する
- アラームの確認状態をマークする
- 特定されたデバイスの問題を緩和する

ML Detect を有効化する

次の手順では、コンソールで ML Detect を設定する方法について説明します。

1. まず、モデルの継続的なトレーニングと更新のために、デバイスが ML Detect の最小要件で定義されているところに従って、必要となる最小データポイントを作成することを確認します。データ収集を進めるには、セキュリティプロファイルがターゲット (モノまたはモノのグループである場合があります) に確実にアタッチされているようにします。
2. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開します。[Detect] (検出)、[Security profiles] (セキュリティプロファイル、[Create security profile] (セキュリティプロファイルの作成)、[Create ML anomaly Detect profile] (ML 異常検出プロファイルの作成) の順に選択します。
3. [Set basic configurations] (基本的な設定を編集) ページで、以下を実行します。
   - [Target] (ターゲット) で、ターゲットデバイスグループを選択します。
   - [Security profile name] (セキュリティプロファイル名) で、セキュリティプロファイルの名前を入力します。
   - (オプション) [Description] (説明) で、ML プロファイルの簡単な説明を記入できます。
   - [Selected metric behaviors in Security Profile] (セキュリティプロファイルでの選択されたメトリクスの動作) で、監視するメトリクスを選択します。

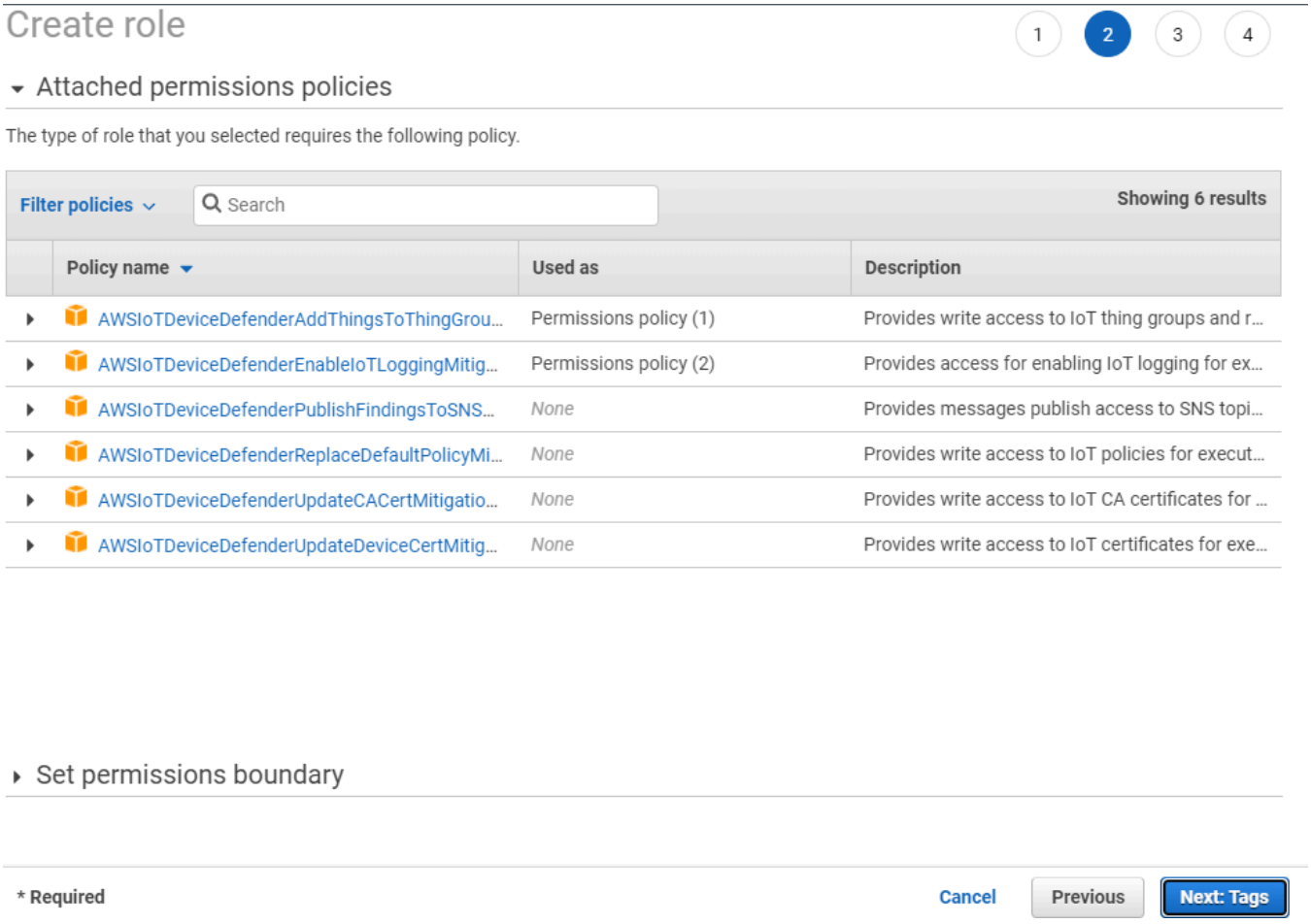


終了したら、[Next (次へ)] を選択します。

4. リポジトリの [Set SNS (optional)] (SNS の設定 (オプション)) ページで、デバイスがプロファイルの動作に違反した場合のアラーム通知の SNS トピックを指定します。選択した SNS トピックで発行するために使用する IAM ロールを選択します。

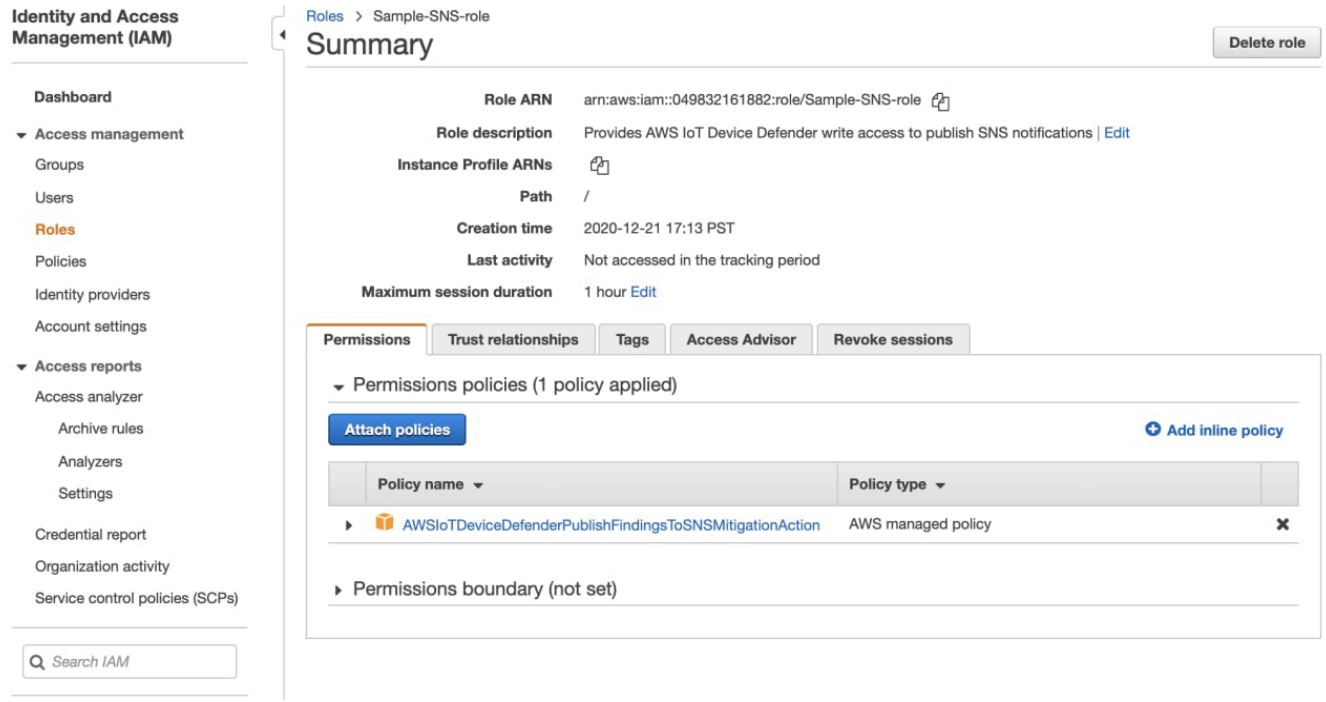
   SNS ロールをまだ持っていない場合は、次の手順に従って、必要となる適切なアクセス許可と信頼関係を持つロールを作成します。

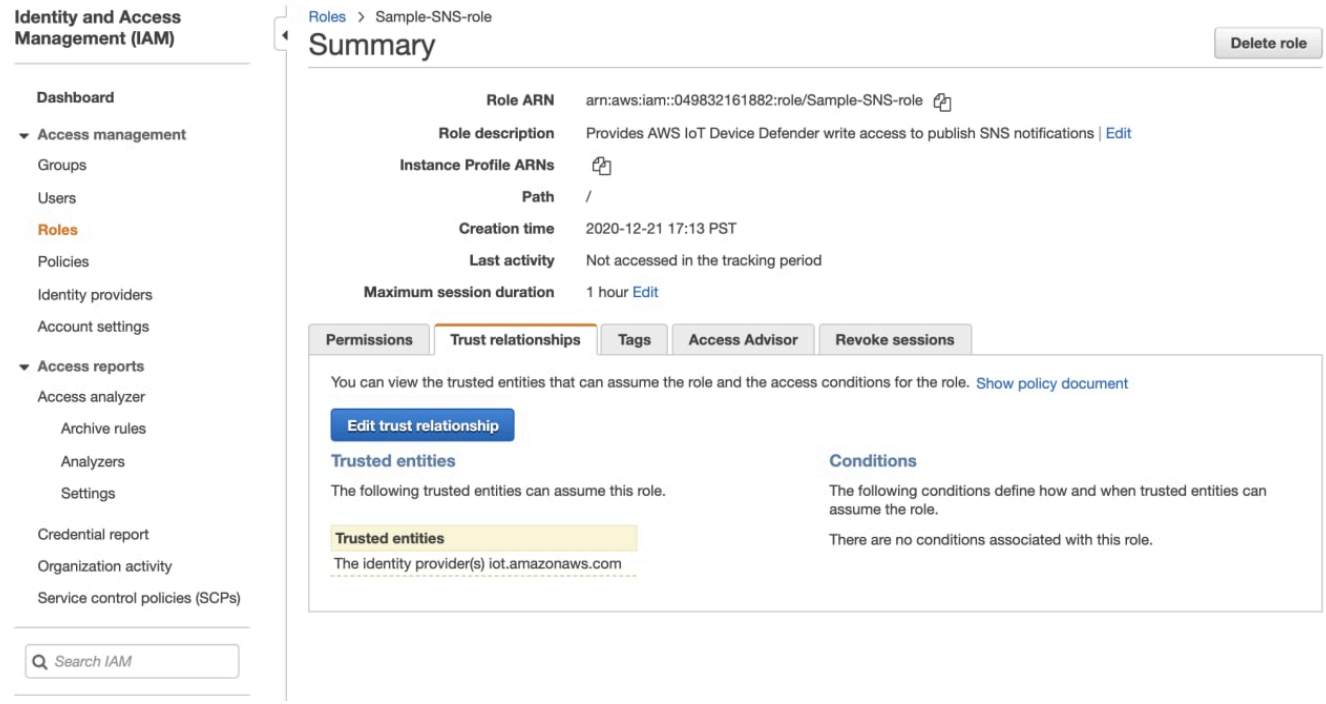
- [IAM console] (IAM コンソール) に入ります。ナビゲーションペインで [ロール] を選択した後、[ロールの作成] を選択します。
- [信頼されたエンティティのタイプを選択] で、[AWS のサービス] を選択します。その後、[Choose a use case] (ユースケースの選択) で [IoT] を選択し、[Select your use case] (ユースケースの選択) で [IoT - Device Defender Mitigation Actions] を選択します。完了したら、[Next: Permissions] (次へ: アクセス許可) を選択します。
- [Attached permissions policies] (アタッチされたアクセス許可ポリシー) で、[AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction] が選択されていることを確認してから、[Next: Tags] (次へ: タグ) を選択します。



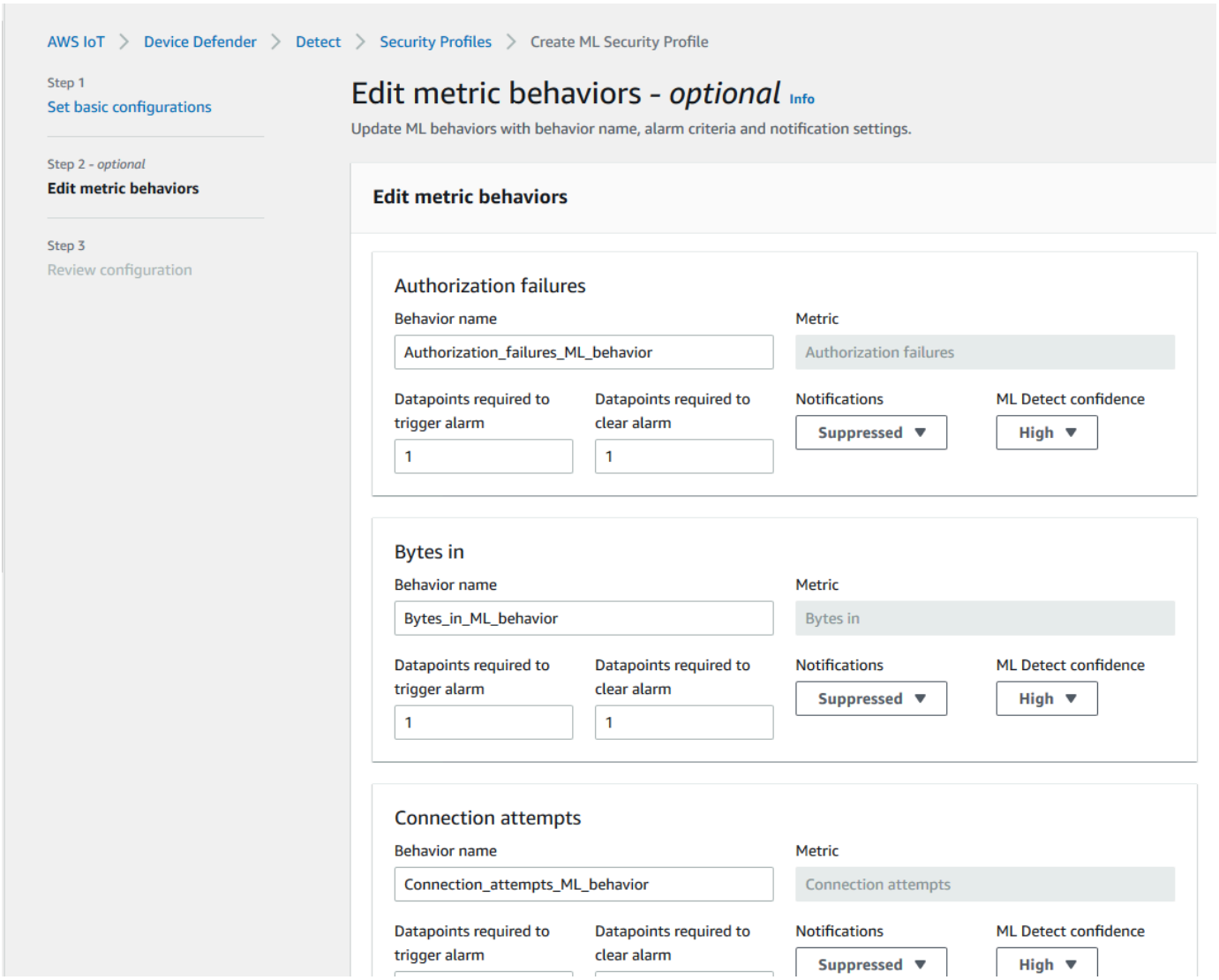
- [Add tags (optional)] (タグの追加 (オプション)) では、ロールに関連付けるタグを追加できます。終了したら、[Next: Review] を選択します。
- [Review] (レビュー) で、ロールに名前を付け、AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction が [Permissions] (アクセス許可) の下に表示され、AWS service: iot.amazonaws.com が [Trust relationships] (信頼関係)

の下に表示されていることを確認します。終了したら、[Create role] (ロールの作成) を選択します。

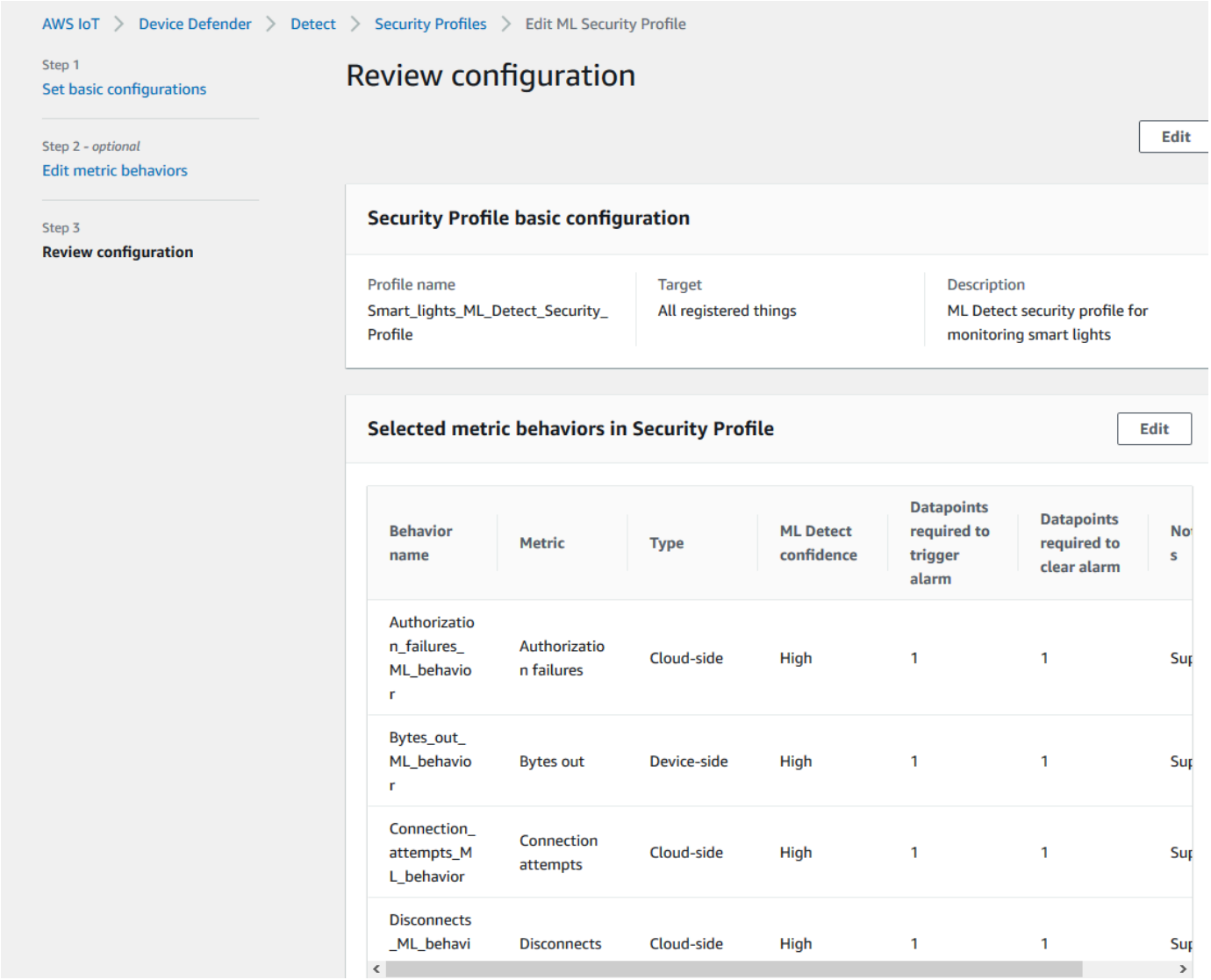




5. [Edit Metric behavior] (メトリクスの動作の編集) ページでは、ML の動作の設定をカスタマイズできます。



6. 終了したら、[Next (次へ)] を選択します。
7. [Review configuration] (設定の確認) ページで、機械学習で監視する動作を確認し、[Next] (次へ) を選択します。



8. セキュリティプロファイルの作成後は、[Security Profiles] (セキュリティプロファイル) ページにリダイレクトされます。ここでは、新しく作成されたセキュリティプロファイルが表示されます。

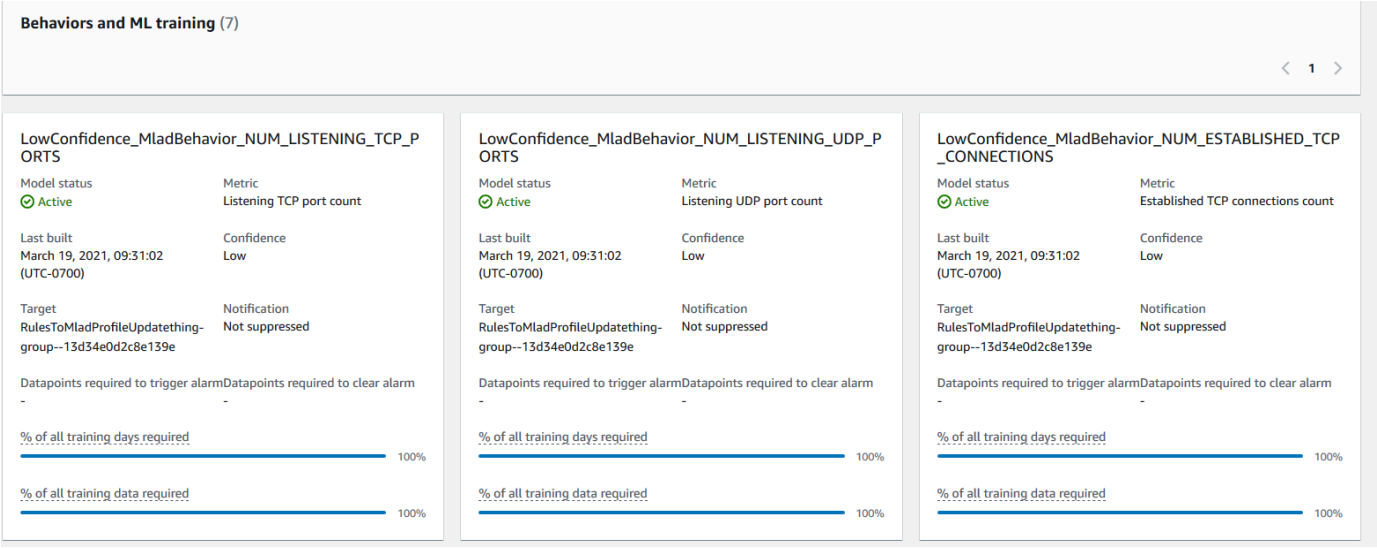
> **Note**
>
> ML モデルの初期トレーニングと作成が完了するまでに 14 日間かかります。デバイスに異常なアクティビティがある場合は、完了後にアラームが表示されることが想定されます。

ML モデルのステータスを監視する

ML モデルが初期トレーニング期間内にある間は、次の手順を実行することでいつでも進捗状況を監視できます。

1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Security profiles] (セキュリティプロファイル) の順に選択します。
2. [Security Profiles] (セキュリティプロファイル) ページで、確認するセキュリティプロファイルを選択します。その後、[Behaviors and ML training] (動作と ML トレーニング) を選択します。
3. [Behaviors and ML training] (動作と ML トレーニング) ページで、ML モデルのトレーニングの進捗状況を確認します。

   モデルのステータスが [Active] (アクティブ) になると、使用状況に基づいて Detect の決定を開始し、毎日プロファイルを更新します。

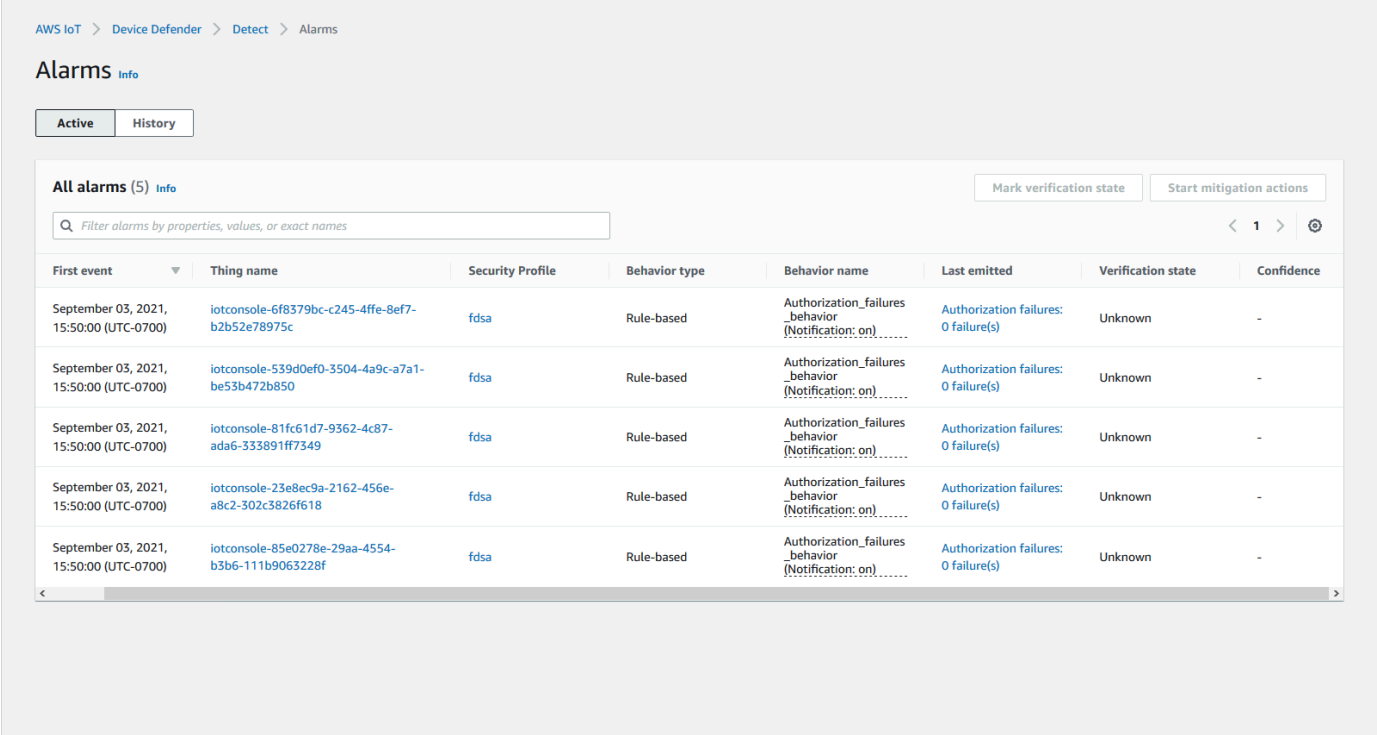


> **Note**
>
> モデルが想定したとおりに進捗しない場合は、デバイスが 最小要件 を満たしていることを確認してください。

ML Detect アラームを確認する

ML モデルが構築され、データ推論の準備が整ったら、モデルによって識別されるアラームを定期的に表示および調査できます。

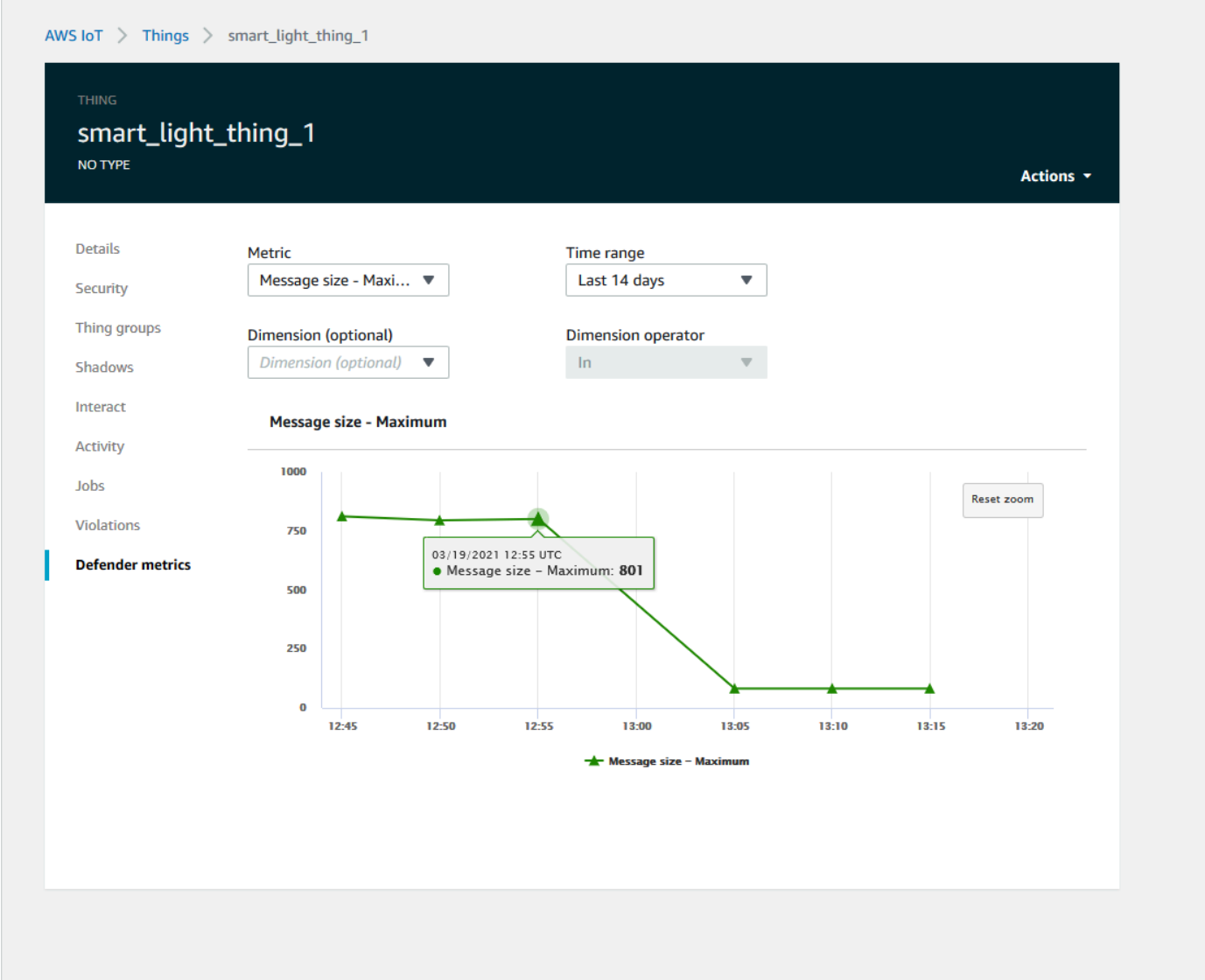
1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Alarms] (アラーム) の順に選択します。



2. [History] (履歴) タブに移動すると、アラーム状態ではなくなったデバイスの詳細を表示することもできます。

詳細情報を取得するには、[Manage] (管理) で [Things] (モノ) を選択し、詳細を表示するモノを選択して、[Defender metrics] (Defender メトリクス) に移動します。[Active] (アクティブ) タブ

から、[Defender metrics graph] (Defender メトリクスグラフ) にアクセスして、アラーム状態にあるものすべてに対して調査を実行できます。この場合、グラフにはメッセージサイズのスパイクが表示され、アラームが開始されます。その後、アラームがクリアされたことを確認できます。

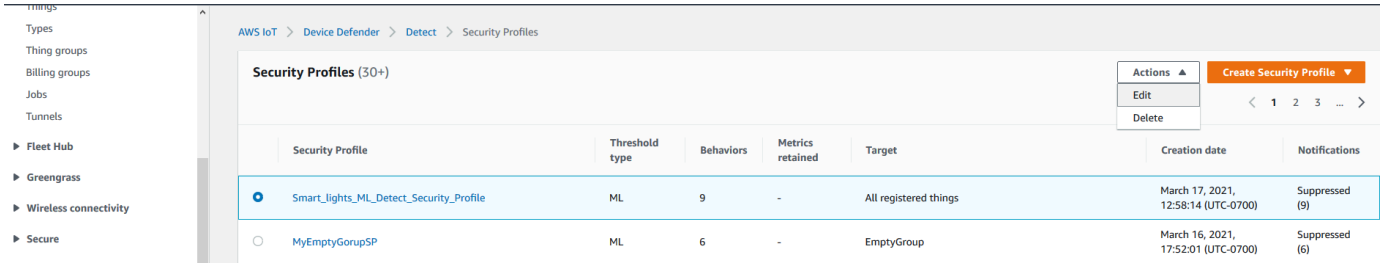


ML アラームを微調整する

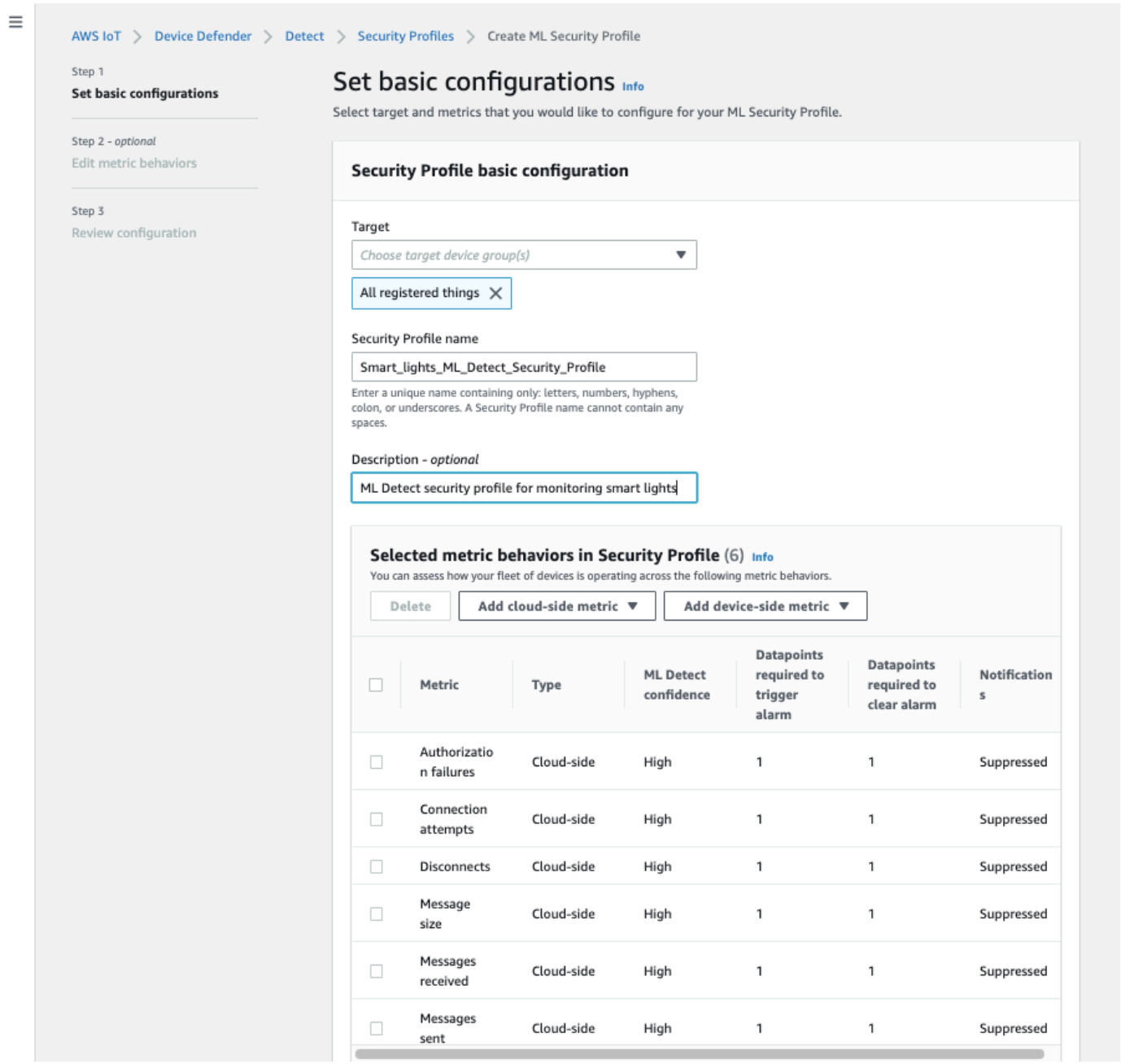
ML モデルが構築され、データ評価の準備が整ったら、セキュリティプロファイルの ML 動作設定を更新して設定を変更できます。次の手順は、AWS CLI でセキュリティプロファイルの ML 動作設定を更新する方法を示しています。

1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Security profiles] (セキュリティプロファイル) の順に選択します。

2. [Security Profiles] (セキュリティプロファイル) ページで、確認するセキュリティプロファイルの横にあるチェックボックスをオンにします。その後、[Actions] (アクション)、[Edit] (編集) の順に選択します。

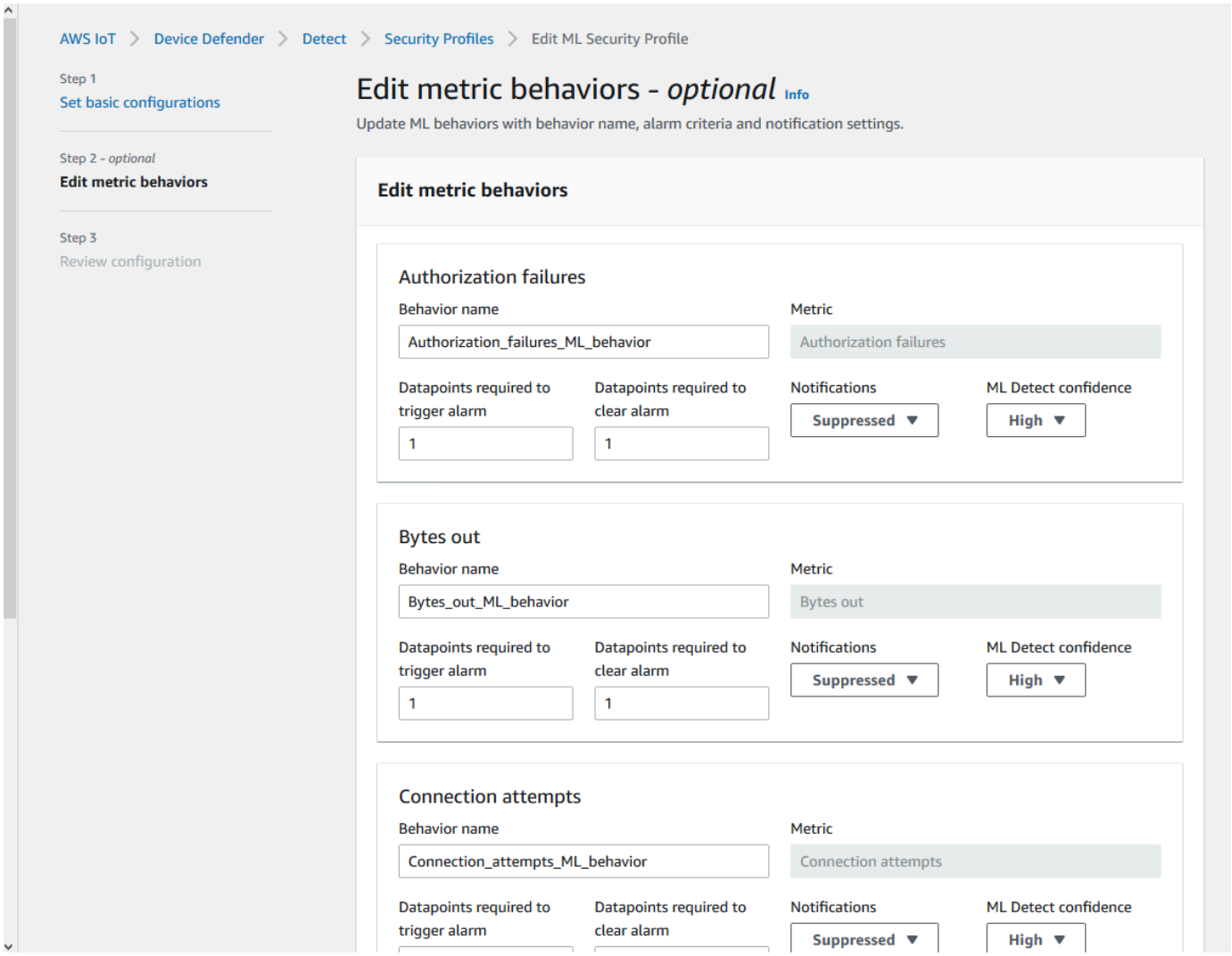


3. [Set basic configurations] (基本構成の設定) では、セキュリティプロファイルのターゲットのモノのグループを調整したり、監視するメトリクスを変更したりできます。



4. 次のいずれかを更新するには、[Edit metric behaviors] (メトリクスの動作の編集) に移動します。

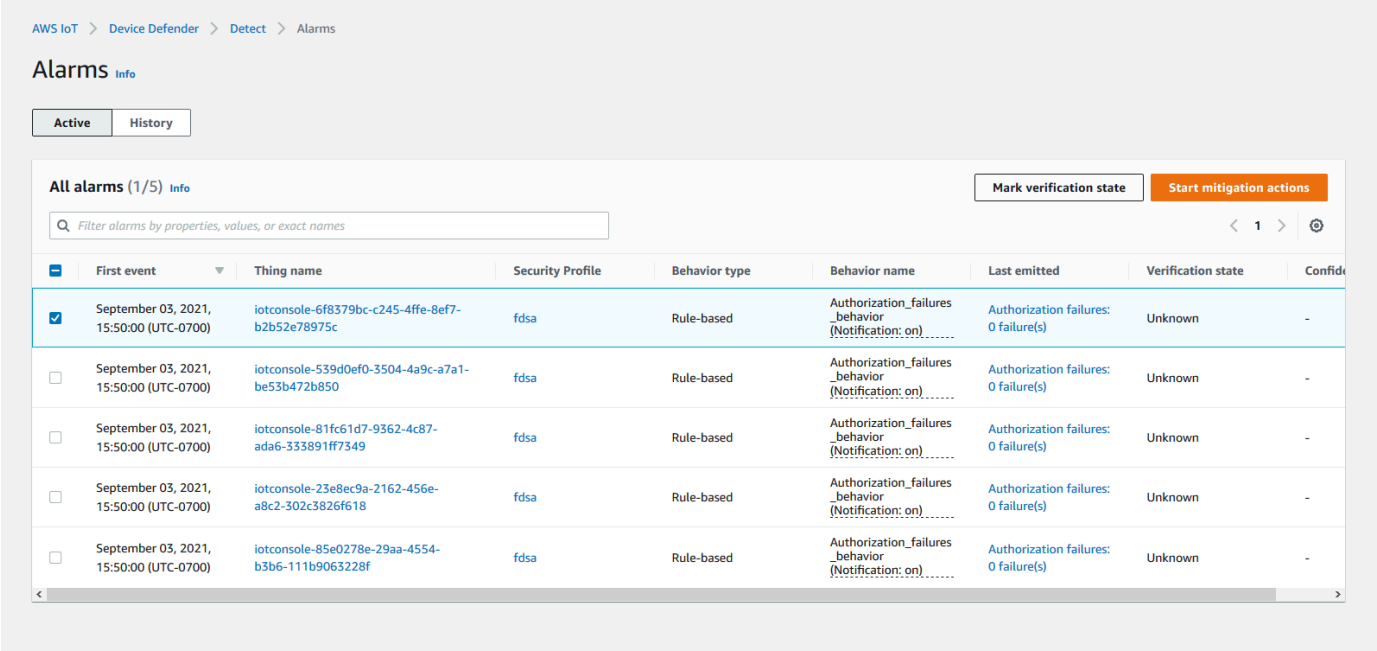
   - アラームを開始するために必要な ML モデルのデータポイント
   - アラームをクリアするために必要な ML モデルのデータポイント
   - ML Detect 信頼レベル
   - ML Detect 通知 (例えば、抑制されていません,抑制された)



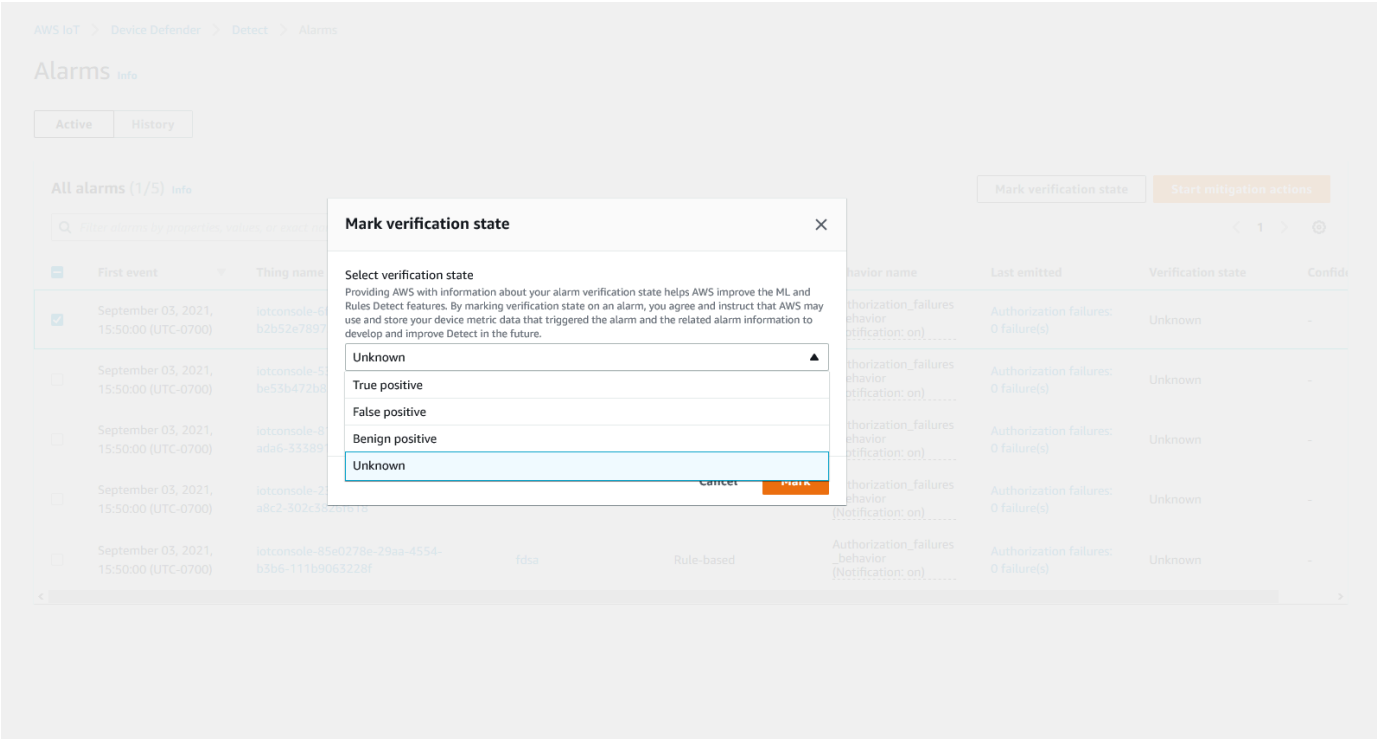
アラームの確認状態をマークする

検証状態を設定し、その検証状態の説明を入力して、アラームをマークします。これは、自分と自身のチームが応答する必要のないアラームを特定するのに役立ちます。

1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Alarms] (アラーム) の順に選択します。検証状態をマークするアラームを選択します。



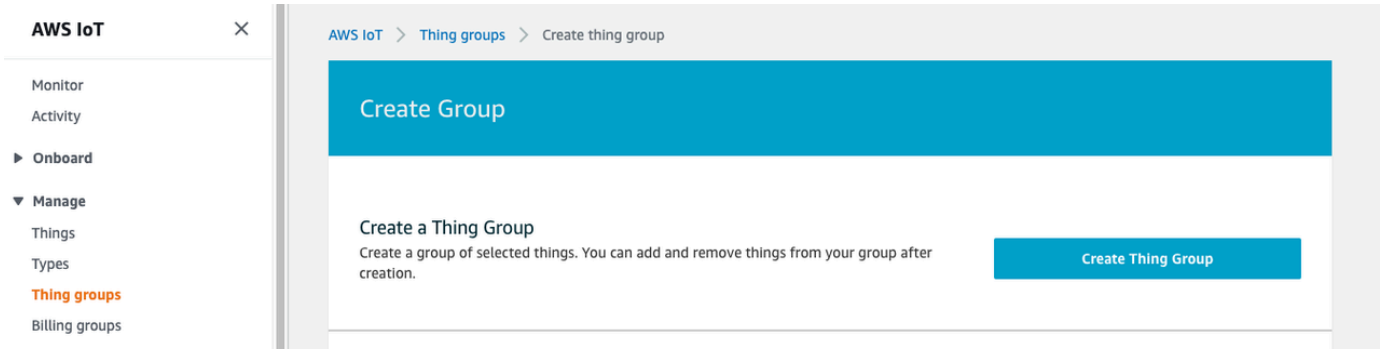
2. 検証状態をマークするを選びます。検証状態モーダルが開きます。
3. 適切な検証状態を選択し、検証の説明（オプション）を入力し、Markを選びます。このアクションは、選択したアラームに検証状態と説明を割り当てます。



特定されたデバイスの問題を緩和する

1. (オプション) 検査結果の緩和アクションを設定する前に、違反しているデバイスを移動する検査グループを設定しましょう。既存のグループを使用することもできます。
2. [Manage] (管理)、[Thing groups] (モノのグループ)、[Create Thing Group] (モノのグループの作成) の順に移動します。モノのグループに名前を付けます。このチュートリアルでは、モノのグループに `Quarantine_group` という名前を付けます。[Thing group] (モノのグループ)、[Security] (セキュリティ) では、モノのグループに次のポリシーを適用します。

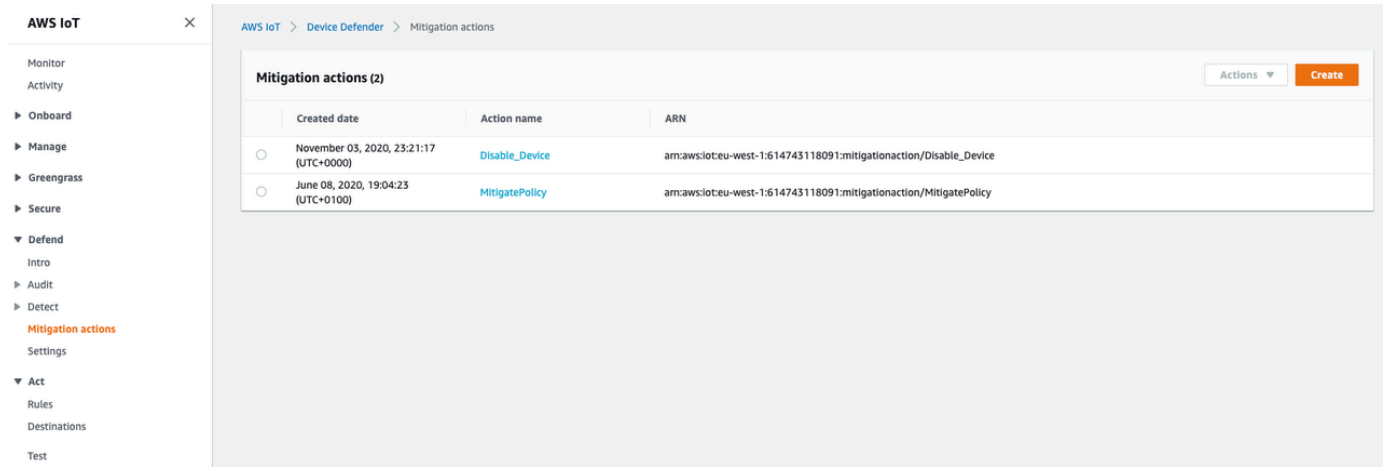
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "iot:*",
      "Resource": "*",
    }
  ]
}
```



終了したら、[Create thing group] (モノのグループの作成) を選択します。

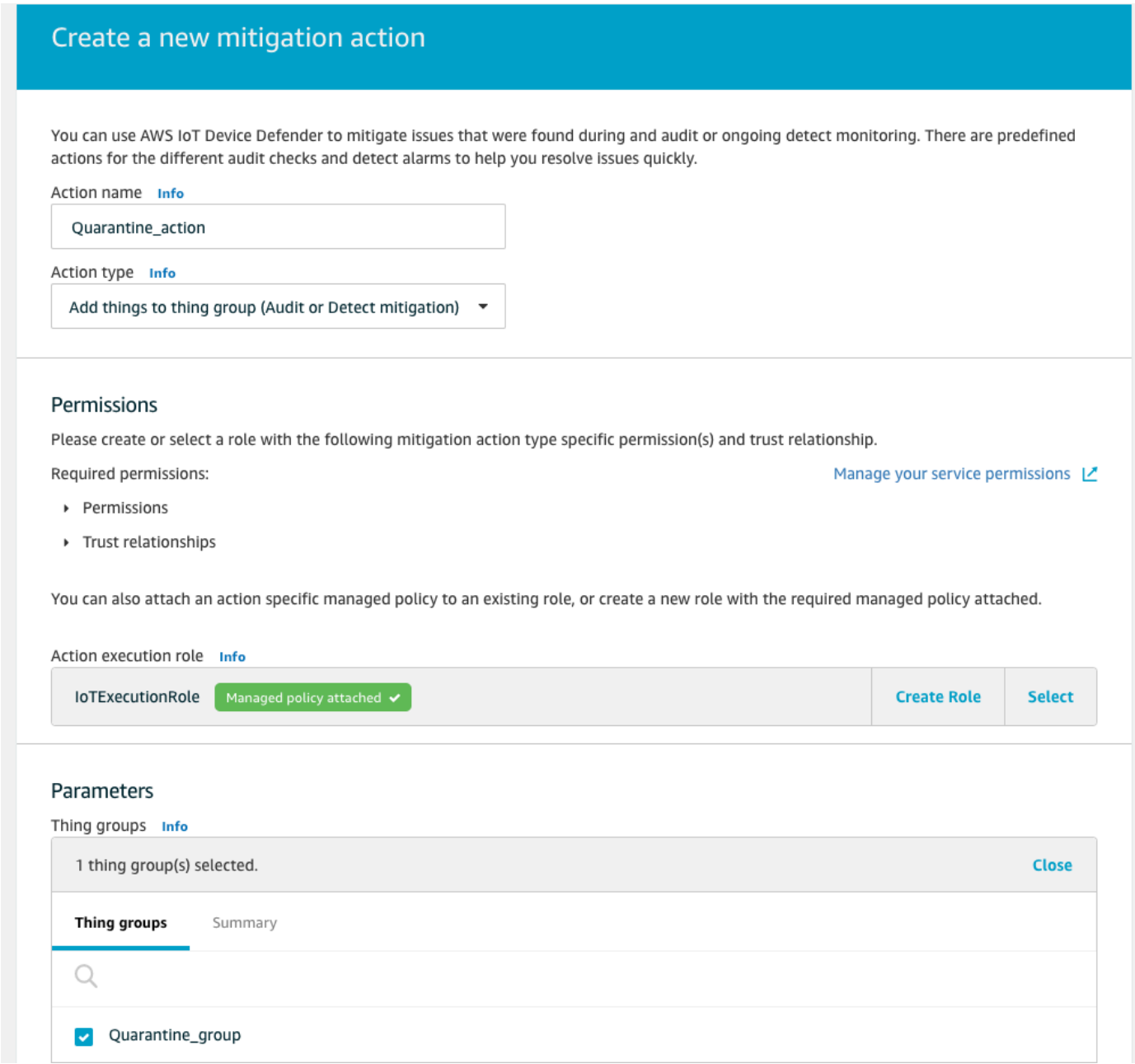
3. モノのグループを作成したので、アラーム状態のデバイスを `Quarantine_group` に移動する緩和アクションを作成しましょう。

   [Defend] (防御) の [Mitigation actions] (緩和アクション) で、[Create] (作成) を選択します。



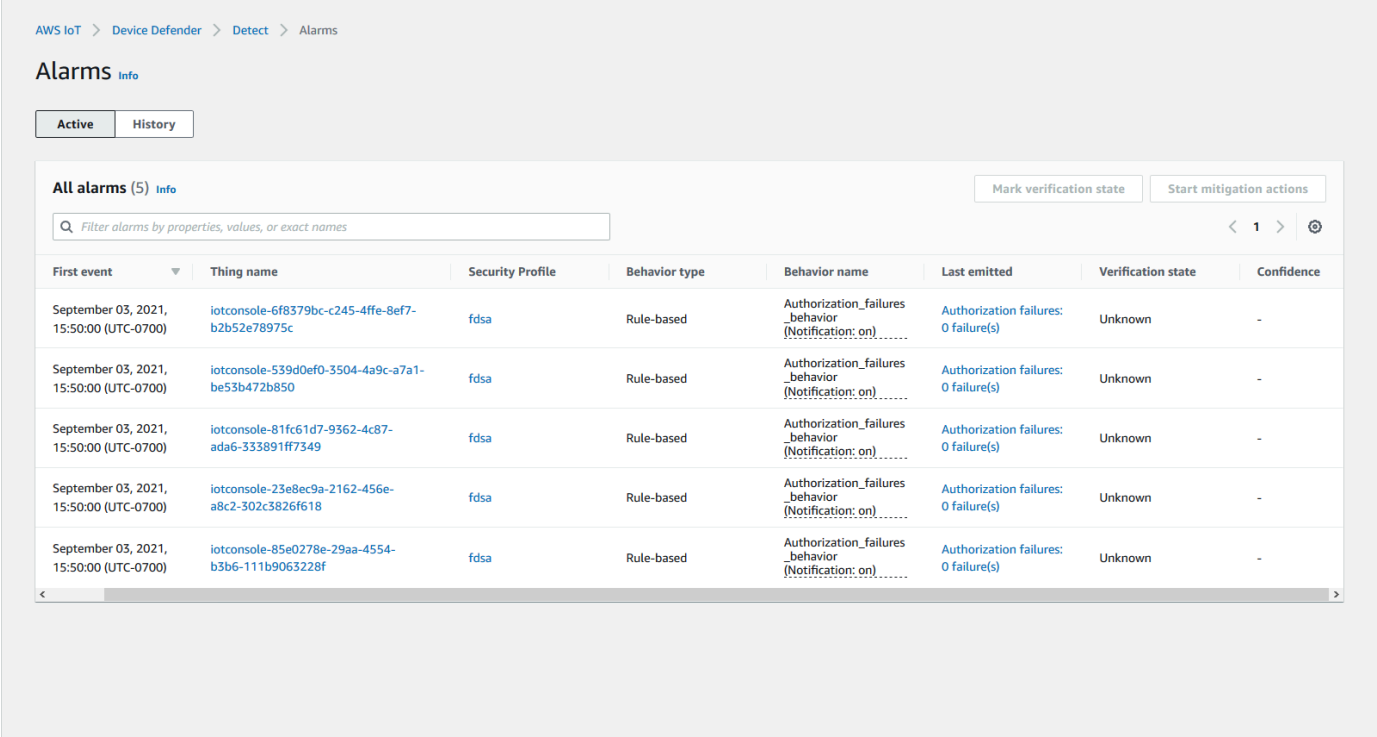
4. [Create a new mitigation action] (新しい緩和アクションの作成) ページで、以下の情報を入力します。

   - [Action name] (アクション名): 緩和アクションに名前 (例: **`Quarantine_action`**) を付けます。
   - [Action type] (アクションの種類): アクションの種類を選択します。[Add things to thing group (Audit or Detect mitigation)] (モノのグループにモノを追加 (緩和機能の監査または検出)を選択します。
   - [Action execution role] (アクション実行ロール): ロールを作成するか、事前に作成している場合は既存のロールを選択します。
   - Parameters (パラメータ): モノのグループを選択します。以前に作成した `Quarantine_group` を使用できます。



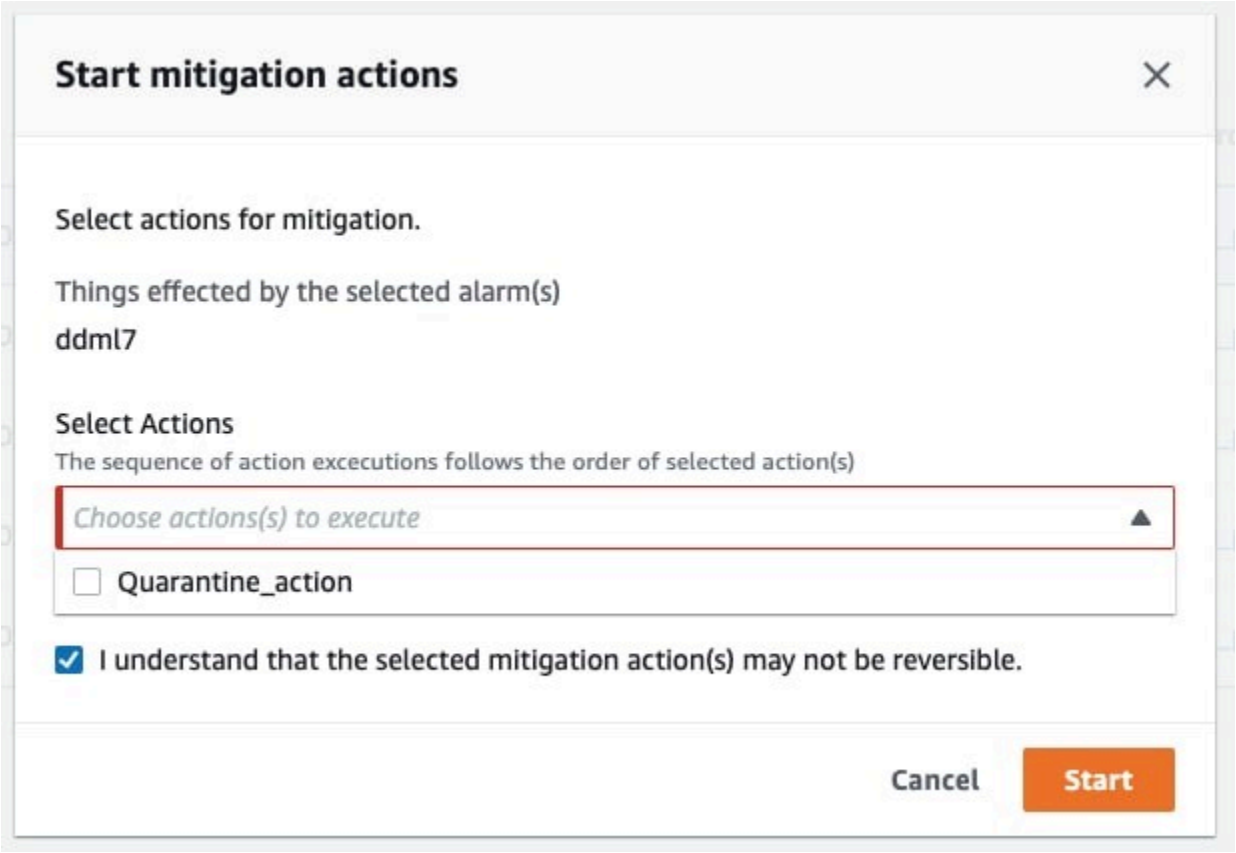
完了したら、[Save] を選択します。これで、アラーム状態のデバイスを検査対象のモノのグループに移動する緩和アクションと、調査中にデバイスを隔離する緩和アクションが追加されました。

5. [Defender]、[Detect] (検出)、[Alarms] (アラーム) の順に移動します。[Active] (アクティブ) の下で、アラーム状態になっているデバイスを確認できます。

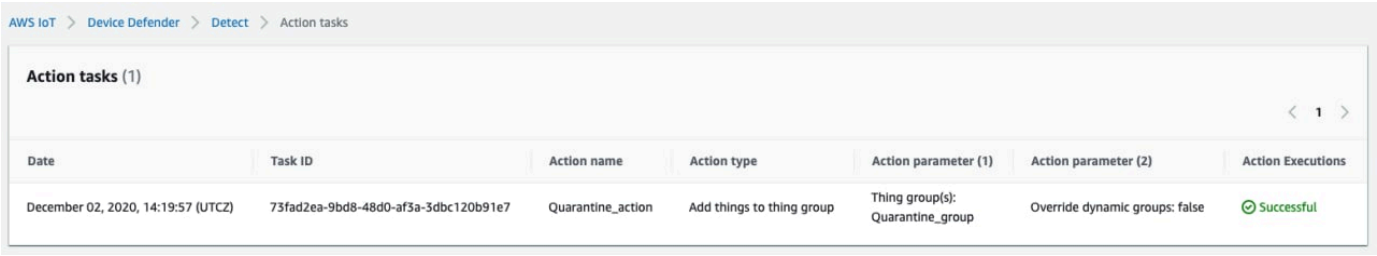


検査対象のグループに移動するデバイスを選択し、[Start Mitigation Actions] (緩和アクションの開始) を選択します。

6. [Start Mitigation Actions] (緩和アクションの開始) の [Start Actions] (アクションの開始) で、先ほど作成した緩和アクションを選択します。例えば、[**`Quarantine_action`**] を選択してから、[Start] (開始) を選択します。[Action Tasks] (アクションタスク) ページが開きます。



7. これでデバイスが **Quarantine_group** に隔離され、アラームを引き起こした問題の根本原因を調査できます。調査が完了したら、デバイスをモノのグループから移動したり、さらにアクションを実行したりできます。



CLI で ML Detect を使用する方法

CLI を使用して ML Detect を設定する方法を次に示します。

チュートリアル

- ML Detect を有効化する
- ML モデルのステータスを監視する

- ML Detect アラームを確認する
- ML アラームを微調整する
- アラームの確認状態をマークする
- 特定されたデバイスの問題を緩和する

ML Detect を有効化する

以下の手順では、AWS CLI で ML Detect を有効にする方法を示します。

1. モデルの継続的なトレーニングと更新のために、ML Detect の最小要件で定義されているところに従って必要となる最小データポイントをデバイスが作成することを確認します。データ収集を進めるには、セキュリティプロファイルにアタッチされたモノのグループにモノが確実に存在するようにしてください。
2. `create-security-profile` コマンドを使用して、ML Detect セキュリティプロファイルを作成します。次の例では、*`security-profile-for-smart-lights`* という名前のセキュリティプロファイルを作成します。このプロファイルは、送信されたメッセージの数、承認の失敗の数、接続の試行の数、および切断の数をチェックします。この例では、メトリクスが ML Detect モデルを使用することを確立するために `mlDetectionConfig` を使用します。

```
aws iot create-security-profile \
    --security-profile-name security-profile-for-smart-lights \
    --behaviors \
     '[{
    "name": "num-messages-sent-ml-behavior",
    "metric": "aws:num-messages-sent",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  },
  {
    "name": "num-authorization-failures-ml-behavior",
    "metric": "aws:num-authorization-failures",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
```

```
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  },
  {
    "name": "num-connection-attempts-ml-behavior",
    "metric": "aws:num-connection-attempts",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  },
  {
    "name": "num-disconnects-ml-behavior",
    "metric": "aws:num-disconnects",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  }]'
```

出力:

```
{
    "securityProfileName": "security-profile-for-smart-lights",
    "securityProfileArn": "arn:aws:iot:eu-
west-1:123456789012:securityprofile/security-profile-for-smart-lights"
  }
```

3. 次に、セキュリティプロファイルを 1 つまたは複数のモノのグループに関連付けます。attach-security-profile コマンドを使用して、モノのグループをセキュリティプロファイルにアタッチします。次の例では、*ML_Detect_beta_static_group* という名前のモ

ノのグループを *security-profile-for-smart-lights* セキュリティプロファイルに関連付けます。

```
aws iot attach-security-profile \
--security-profile-name security-profile-for-smart-lights \
--security-profile-target-arn arn:aws:iot:eu-
west-1:123456789012:thinggroup/ML_Detect_beta_static_group
```

出力:

なし。

4. 完全なセキュリティプロファイルを作成したら、ML モデルがトレーニングを開始します。ML モデルの初期トレーニングと構築が完了するまでに 14 日間かかります。14 日が経過すると、デバイスで異常なアクティビティが発生した場合、アラームが表示されます。

ML モデルのステータスを監視する

以下の手順では、ML モデルの進行中のトレーニングを監視する方法を示しています。

- get-behavior-model-training-summaries コマンドを使用して、ML モデルの進捗状況を表示します。次の例では、*security-profile-for-smart-lights* セキュリティプロファイルの ML モデルトレーニングの進捗状況の概要を取得します。modelStatus は、モデルがトレーニングを完了したか、特定の動作のビルドがまだ保留中であるかを示します。

```
aws iot get-behavior-model-training-summaries \
   --security-profile-name security-profile-for-smart-lights
```

出力:

```
{
    "summaries": [
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Messages_sent_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 29.408,
            "lastModelRefreshDate": "2020-12-07T14:35:19.237000-08:00"
        },
```

```
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Messages_received_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Authorization_failures_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 35.464,
            "lastModelRefreshDate": "2020-12-07T14:29:44.396000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Message_size_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 29.332,
            "lastModelRefreshDate": "2020-12-07T14:30:44.113000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Connection_attempts_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 32.891999999999996,
            "lastModelRefreshDate": "2020-12-07T14:29:43.121000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Disconnects_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 35.46,
            "lastModelRefreshDate": "2020-12-07T14:29:55.556000-08:00"
        }
    ]
}
```

> **Note**
>
> モデルが想定したとおりに進捗しない場合は、デバイスが 最小要件 を満たしていることを確認してください。

ML Detect アラームを確認する

ML モデルが構築され、データ評価の準備が整ったら、モデルによって推測されるアラームを定期的に表示できます。以下の手順では、AWS CLI でアラームを表示する方法を示します。

- すべてのアクティブなアラームを表示するには、`list-active-violations` コマンドを使用します。

```
aws iot list-active-violations \
--max-results 2
```

出力:

```
{
    "activeViolations": []
}
```

または、`list-violation-events` コマンドを使用して、特定の期間中に検出されたすべての違反を表示することもできます。次の例では、2020 年 9 月 22 日 5:42:13 GMT から 2020 年 10 月 26 日 5:42:13 GMT までの違反イベントをリストします。

```
aws iot list-violation-events \
    --start-time 1599500533 \
    --end-time 1600796533 \
    --max-results 2
```

出力:

```
{
    "violationEvents": [
        {
            "violationId": "1448be98c09c3d4ab7cb9b6f3ece65d6",
            "thingName": "lightbulb-1",
```

```
            "securityProfileName": "security-profile-for-smart-lights",
            "behavior": {
                "name": "LowConfidence_MladBehavior_MessagesSent",
                "metric": "aws:num-messages-sent",
                "criteria": {
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1,
                    "mlDetectionConfig": {
                        "confidenceLevel": "HIGH"
                    }
                },
                "suppressAlerts": true
            },
            "violationEventType": "alarm-invalidated",
            "violationEventTime": 1600780245.29
        },
        {
            "violationId": "df4537569ef23efb1c029a433ae84b52",
            "thingName": "lightbulb-2",
            "securityProfileName": "security-profile-for-smart-lights",
            "behavior": {
                "name": "LowConfidence_MladBehavior_MessagesSent",
                "metric": "aws:num-messages-sent",
                "criteria": {
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1,
                    "mlDetectionConfig": {
                        "confidenceLevel": "HIGH"
                    }
                },
                "suppressAlerts": true
            },
            "violationEventType": "alarm-invalidated",
            "violationEventTime": 1600780245.281
        }
    ],
    "nextToken":
 "Amo6XIUrsOohsojuIG6TuwSR3X9iUvH2OCksBZg6bed2j21VSnD1uP1pflxKX1+a3cvBRSosIB0xFv40kM6RYBknZ
vxabMe/ZW31Ps/WiZHlr9Wg7R7eEGli59IJ/U0iBQ1McP/ht0E2XA2TTIvYeMmKQQPsRj/
eoV9j7P/wveu7skNGepU/mvpV0O2Ap7hnV5U+Prx/9+iJA/341va
+pQww7jpUeHmJN9Hw4MqW0ysw0Ry3w38hOQWEpz2xwFWAxAARxeIxCxt5c37RK/lRZBlhYqoB
+w2PZ74730h8pICGY4gktJxkwHyyRabpSM/G/f5DFrD9O5v8idkTZzBxW2jrbzSUIdafPtsZHL/
yAMKr3HAKtaABz2nTsOBNre7X2d/jIjjarhon0Dh9l+8I9Y5Ey
+DIFBcqFTvhibKAafQt3gs6CUiqHdWiCenfJyb8whmDE2qxvdxGElGmRb
```

```
+k6kuN5jrZxxw95gzfYDgRHv11iEn8h1qZLD0czkIFBpMppHj9cetHPvM
+qffXGAzKi8tL6eQuCdMLXmVE3jbqcJcjk9ItnaYJi5zKDz9FVbrz9qZZPtZJFHp"
}
```

ML アラームを微調整する

ML モデルが構築され、データ評価の準備が整ったら、セキュリティプロファイルの ML 動作設定を更新して設定を変更できます。次の手順は、AWS CLI でセキュリティプロファイルの ML 動作設定を更新する方法を示しています。

- セキュリティプロファイルの ML 動作設定を変更するには、update-security-profile コマンドを使用します。次の例では、いくつかの動作の confidenceLevel を変更して、*security-profile-for-smart-lights* セキュリティプロファイルの動作を更新し、すべての動作の通知を抑制解除します。

```
aws iot update-security-profile \
    --security-profile-name security-profile-for-smart-lights \
    --behaviors \
     '[{
      "name": "num-messages-sent-ml-behavior",
      "metric": "aws:num-messages-sent",
      "criteria": {
          "mlDetectionConfig": {
              "confidenceLevel" : "HIGH"
          }
      },
      "suppressAlerts": false
  },
  {
      "name": "num-authorization-failures-ml-behavior",
      "metric": "aws:num-authorization-failures",
      "criteria": {
          "mlDetectionConfig": {
              "confidenceLevel" : "HIGH"
          }
      },
      "suppressAlerts": false
  },
  {
      "name": "num-connection-attempts-ml-behavior",
      "metric": "aws:num-connection-attempts",
```

```
      "criteria": {
          "mlDetectionConfig": {
              "confidenceLevel" : "HIGH"
          }
      },
      "suppressAlerts": false
  },
  {
      "name": "num-disconnects-ml-behavior",
      "metric": "aws:num-disconnects",
      "criteria": {
          "mlDetectionConfig": {
              "confidenceLevel" : "LOW"
          }
      },
      "suppressAlerts": false

  }]'
```

出力:

```
 {
    "securityProfileName": "security-profile-for-smart-lights",
    "securityProfileArn": "arn:aws:iot:eu-
west-1:123456789012:securityprofile/security-profile-for-smart-lights",
    "behaviors": [
        {
            "name": "num-messages-sent-ml-behavior",
            "metric": "aws:num-messages-sent",
            "criteria": {
                "mlDetectionConfig": {
                    "confidenceLevel": "HIGH"
                }
            }
        },
        {
            "name": "num-authorization-failures-ml-behavior",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "mlDetectionConfig": {
                    "confidenceLevel": "HIGH"
                }
            }
```

```
        },
        {
            "name": "num-connection-attempts-ml-behavior",
            "metric": "aws:num-connection-attempts",
            "criteria": {
                "mlDetectionConfig": {
                    "confidenceLevel": "HIGH"
                }
            },
            "suppressAlerts": false
        },
        {
            "name": "num-disconnects-ml-behavior",
            "metric": "aws:num-disconnects",
            "criteria": {
                "mlDetectionConfig": {
                    "confidenceLevel": "LOW"
                }
            },
            "suppressAlerts": true
        }
    ],
    "version": 2,
    "creationDate": 1600799559.249,
    "lastModifiedDate": 1600800516.856
}
```

アラームの確認状態をマークする

アラームに検証ステータスをマークして、アラームの分類や異常の調査に役立てることができます。

- アラームに確認状態とその状態の説明をマークします。例えば、アラームの検証状態を False positive に設定するために、次のコマンドを使用します。

```
aws iot put-verification-state-on-violation --violation-id 12345  --verification-
state FALSE_POSITIVE --verification-state-description "This is dummy description"
 --endpoint https://us-east-1.iot.amazonaws.com --region us-east-1
```

出力:

なし。

特定されたデバイスの問題を緩和する

1. create-thing-group コマンドを使用して、緩和アクション用のモノのグループを作成します。次の例では、ThingGroupForDetectMitigationAction というモノのグループを作成します。

```
aws iot create-thing-group –thing-group-name ThingGroupForDetectMitigationAction
```

出力:

```
{
 "thingGroupName": "ThingGroupForDetectMitigationAction",
 "thingGroupArn": "arn:aws:iot:us-
east-1:123456789012:thinggroup/ThingGroupForDetectMitigationAction",
 "thingGroupId": "4139cd61-10fa-4c40-b867-0fc6209dca4d"
}
```

2. 次に、create-mitigation-action コマンドを使用して、緩和アクションを作成します。次の例では、緩和アクションを適用するために使用される IAM ロールの ARN を使用して、detect_mitigation_action という緩和アクションを作成します。また、アクションのタイプとそのアクションのパラメータを定義します。この場合、緩和アクションにより、ThingGroupForDetectMitigationAction という以前に作成したモノのグループにモノが移動します。

```
aws iot create-mitigation-action --action-name detect_mitigation_action \
--role-arn arn:aws:iam::123456789012:role/MitigationActionValidRole \
--action-params \
'{
     "addThingsToThingGroupParams": {
         "thingGroupNames": ["ThingGroupForDetectMitigationAction"],
         "overrideDynamicGroups": false
     }
 }'
```

出力:

```
{
 "actionArn": "arn:aws:iot:us-
east-1:123456789012:mitigationaction/detect_mitigation_action",
 "actionId": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3"
}
```

3. start-detect-mitigation-actions-task コマンドを使用して、緩和アクションタスクを開始します。task-id、target、および actions は必須パラメータです。

```
aws iot start-detect-mitigation-actions-task \
    --task-id taskIdForMitigationAction \
    --target '{ "violationIds" : [ "violationId-1", "violationId-2" ] }' \
    --actions "detect_mitigation_action" \
    --include-only-active-violations \
    --include-suppressed-alerts
```

出力:

```
{
    "taskId": "taskIdForMitigationAction"
}
```

4. (オプション) タスクに含まれる緩和アクションの実行を表示するには、list-detect-mitigation-actions-executions コマンドを使用します。

```
aws iot list-detect-mitigation-actions-executions \
    --task-id taskIdForMitigationAction \
    --max-items 5 \
    --page-size 4
```

出力:

```
{
    "actionsExecutions": [
        {
            "taskId": "e56ee95e - f4e7 - 459 c - b60a - 2701784290 af",
            "violationId": "214_fe0d92d21ee8112a6cf1724049d80",
            "actionName": "underTest_MAThingGroup71232127",
            "thingName": "cancelDetectMitigationActionsTaskd143821b",
            "executionStartDate": "Thu Jan 07 18: 35: 21 UTC 2021",
            "executionEndDate": "Thu Jan 07 18: 35: 21 UTC 2021",
            "status": "SUCCESSFUL",
        }
    ]
}
```

5. (オプション) describe-detect-mitigation-actions-task コマンドを使用すると、緩和アクションタスクに関する情報を取得できます。

```
aws iot describe-detect-mitigation-actions-task \
    --task-id taskIdForMitigationAction
```

出力:

```
{
    "taskSummary": {
        "taskId": "taskIdForMitigationAction",
        "taskStatus": "SUCCESSFUL",
        "taskStartTime": 1609988361.224,
        "taskEndTime": 1609988362.281,
        "target": {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "num-messages-sent-ml-behavior"
        },
        "violationEventOccurrenceRange": {
            "startTime": 1609986633.0,
            "endTime": 1609987833.0
        },
        "onlyActiveViolationsIncluded": true,
        "suppressedAlertsIncluded": true,
        "actionsDefinition": [
            {
                "name": "detect_mitigation_action",
                "id": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3",
                "roleArn":
 "arn:aws:iam::123456789012:role/MitigatioActionValidRole",
                "actionParams": {
                    "addThingsToThingGroupParams": {
                        "thingGroupNames": [
                            "ThingGroupForDetectMitigationAction"
                        ],
                        "overrideDynamicGroups": false
                    }
                }
            }
        ],
        "taskStatistics": {
            "actionsExecuted": 0,
```

```
            "actionsSkipped": 0,
            "actionsFailed": 0
        }
    }
}
```

6. (オプション) 緩和アクションタスクのリストを取得するには、[list-detect-mitigation-actions-tasks](#) コマンドを使用します。

```
aws iot list-detect-mitigation-actions-tasks \
    --start-time 1609985315 \
    --end-time 1609988915 \
    --max-items 5 \
    --page-size 4
```

出力:

```
{
    "tasks": [
        {
            "taskId": "taskIdForMitigationAction",
            "taskStatus": "SUCCESSFUL",
            "taskStartTime": 1609988361.224,
            "taskEndTime": 1609988362.281,
            "target": {
                "securityProfileName": "security-profile-for-smart-lights",
                "behaviorName": "num-messages-sent-ml-behavior"
            },
            "violationEventOccurrenceRange": {
                "startTime": 1609986633.0,
                "endTime": 1609987833.0
            },
            "onlyActiveViolationsIncluded": true,
            "suppressedAlertsIncluded": true,
            "actionsDefinition": [
                {
                    "name": "detect_mitigation_action",
                    "id": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3",
                    "roleArn": "arn:aws:iam::123456789012:role/
MitigatioActionValidRole",
                    "actionParams": {
                        "addThingsToThingGroupParams": {
                            "thingGroupNames": [
```

```
                                "ThingGroupForDetectMitigationAction"
                            ],
                            "overrideDynamicGroups": false
                        }
                    }
                }
            ],
            "taskStatistics": {
                "actionsExecuted": 0,
                "actionsSkipped": 0,
                "actionsFailed": 0
            }
        }
    ]
}
```

7. (オプション) 緩和アクションタスクをキャンセルするには、[cancel-detect-mitigation-actions-task](#) コマンドを使用します。

```
aws iot cancel-detect-mitigation-actions-task \
    --task-id taskIdForMitigationAction
```

出力:

なし。

AWS IoT Device Defender 監査結果をいつどのように表示するかをカスタマイズする

AWS IoT Device Defender 監査では、AWS IoT デバイスとリソースがベストプラクティスに適合していることを確認するために、定期的にセキュリティチェックを行います。各チェックで、監査結果は適合または不適合に分類され、不適合の場合はコンソールの警告アイコンが表示されます。既知の問題のノイズの繰り返しを緩和するために、監査所見の抑制機能を使用すると、これらの不適合の通知を一時的に停止することができます。

特定のリソースまたはアカウントの監査チェックを、あらかじめ決められた期間だけ抑制できます。抑制された監査チェックの結果は、適合カテゴリと不適合カテゴリとは別に、抑制された所見として分類されます。この新しいカテゴリは、不適合の結果の場合のようにアラームをトリガーしません。

これにより、既知のメンテナンス期間中または更新の完了がスケジュールされるまで、不適合の通知に煩わされる機会を減らせます。

開始方法

次のセクションでは、監査結果の抑制を使用して、コンソールおよび CLI での `Device certificate expiring` チェックを抑制する方法について詳しく説明します。いずれかのデモンストレーションに従う場合は、Device Defender が検出できるように、まずは失効する証明書を 2 つ作成する必要があります。

証明書を作成するには、以下を使用します。

- 「AWS IoT Core デベロッパーガイド」の「CA 証明書の作成と登録」
- CA 証明書を使用してクライアント証明書を作成する ステップ 3 で、`days` パラメータを **1** に設定します。

CLI を使用して証明書を作成している場合は、次のコマンドを入力します。

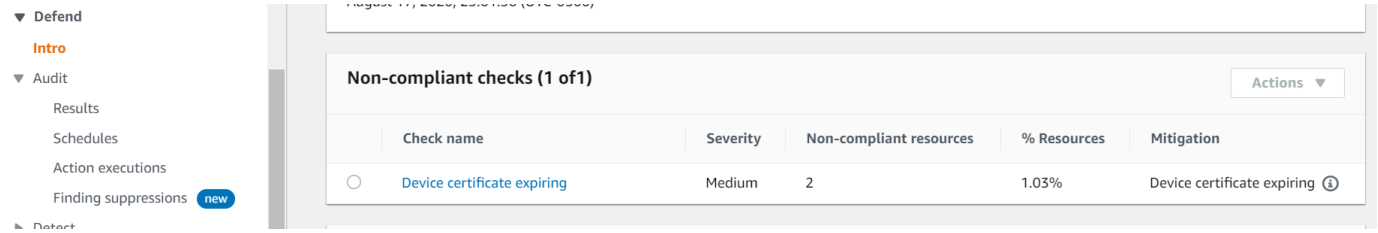
```
openssl x509 -req \
    -in device_cert_csr_filename \
    -CA root_ca_pem_filename \
    -CAkey root_ca_key_filename \
    -CAcreateserial \
    -out device_cert_pem_filename \
    -days 1 -sha256
```

コンソールで監査所見をカスタマイズする

次のチュートリアルでは、不適合の監査チェックをトリガーする、2 つの失効したデバイス証明書を持つアカウントを使用します。このシナリオでは、デベロッパーが問題に対処する新しい機能をテストしているため、警告を無効にします。各証明書用に監査所見の抑制を作成し、次週以降、監査結果が不適合にならないようにします。

1. まず、オンデマンド監査を実行して、失効したデバイス証明書のチェックが不適合であることを示します。

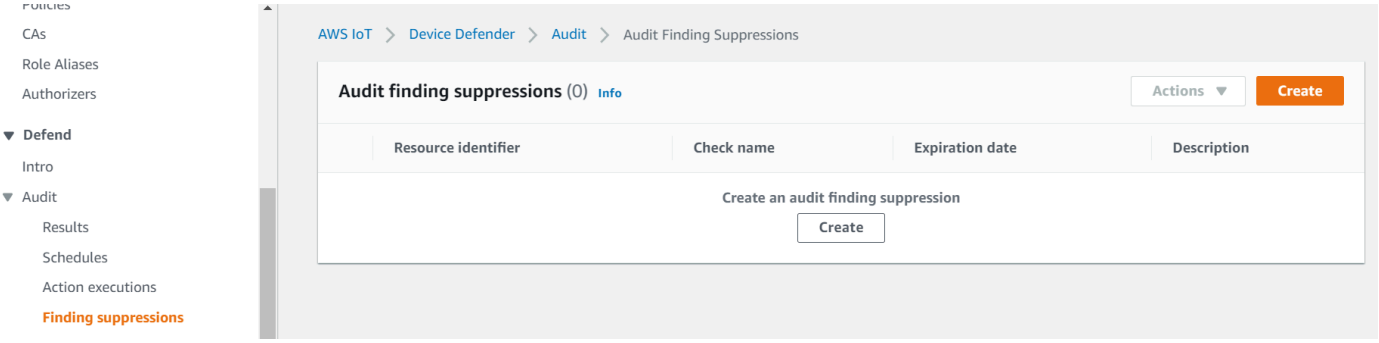
   AWS IoT コンソールで、左側のサイドバーから [Defend] (防御) を選択し、[Audit] (監査)、[Results] (結果) の順に選択します。[Audit Results] (監査結果) ページで、[Create] (作成) を選択します。[新しい監査を作成する] ウィンドウが開きます。[Create] (作成) を選択します。



オンデマンドの監査結果から、[Device certificate expiring] (デバイス証明書の有効期限が切れます) が 2 つのリソースで不適合であることがわかります。

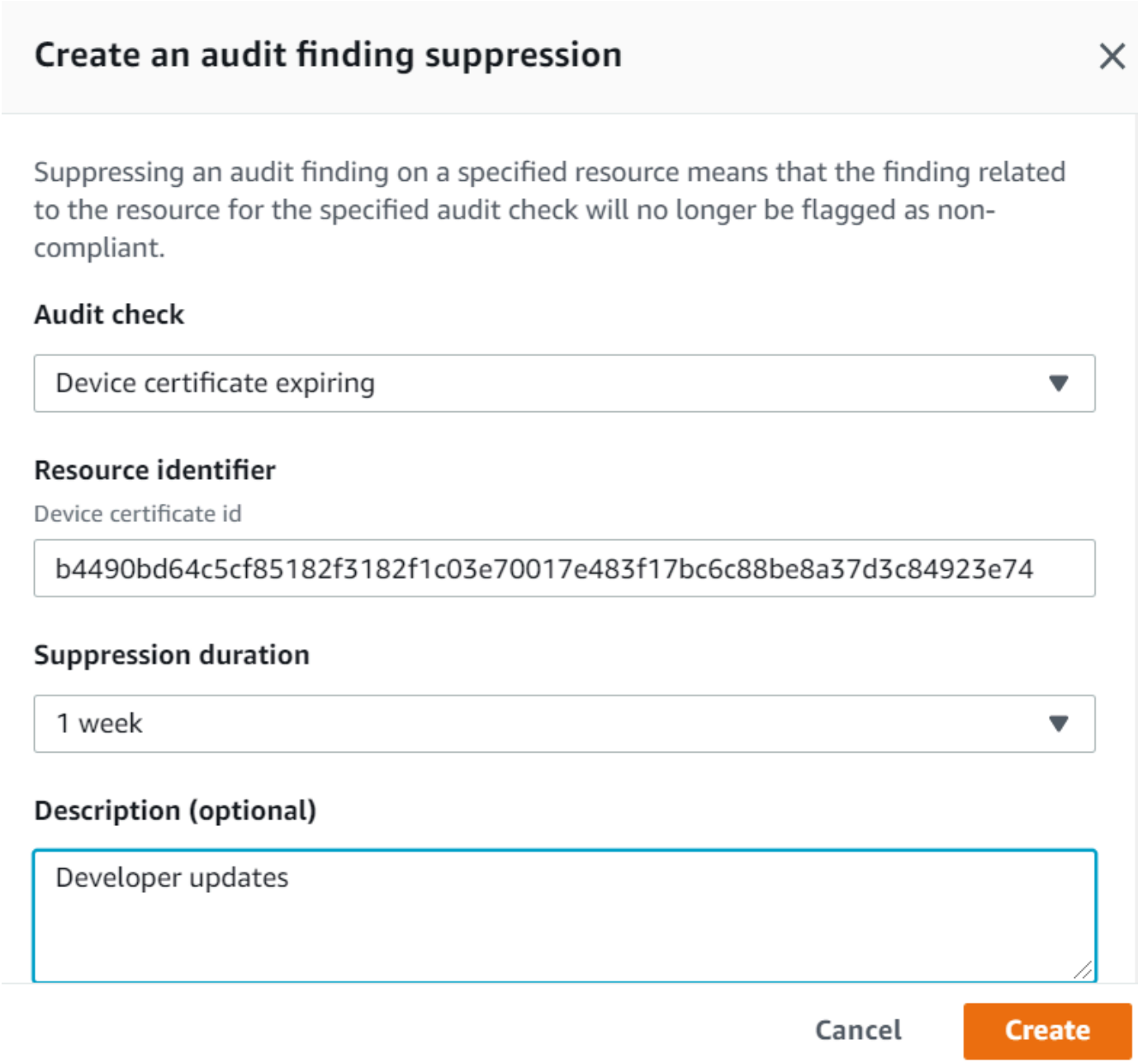
2. ここでは、デベロッパーが警告内容を修正する新機能をテストしているため、[Device certificate expiring] (デバイス証明書の有効期限が切れます) の不適合チェック警告を無効にします。

   [防御] の下の左側のサイドバーから、[監査] を選択し、[検索結果の抑制] を選択します。[監査結果の抑制] ページで、[作成] を選択します。



3. [監査結果の抑制を作成する] ウィンドウで、次のように入力する必要があります。

   - Audit check (監査チェック): 抑制したい監査チェックである、`Device certificate expiring` を選択します。
   - リソース識別: 監査結果を抑制する証明書の 1 つのデバイス証明書 ID を入力します。
   - Suppression duration (抑制期間): `Device certificate expiring`の監査チェックを抑制したい期間である、`1 week` を選択します。
   - 説明 (オプション): この監査結果を抑制する理由を説明するメモを追加します。

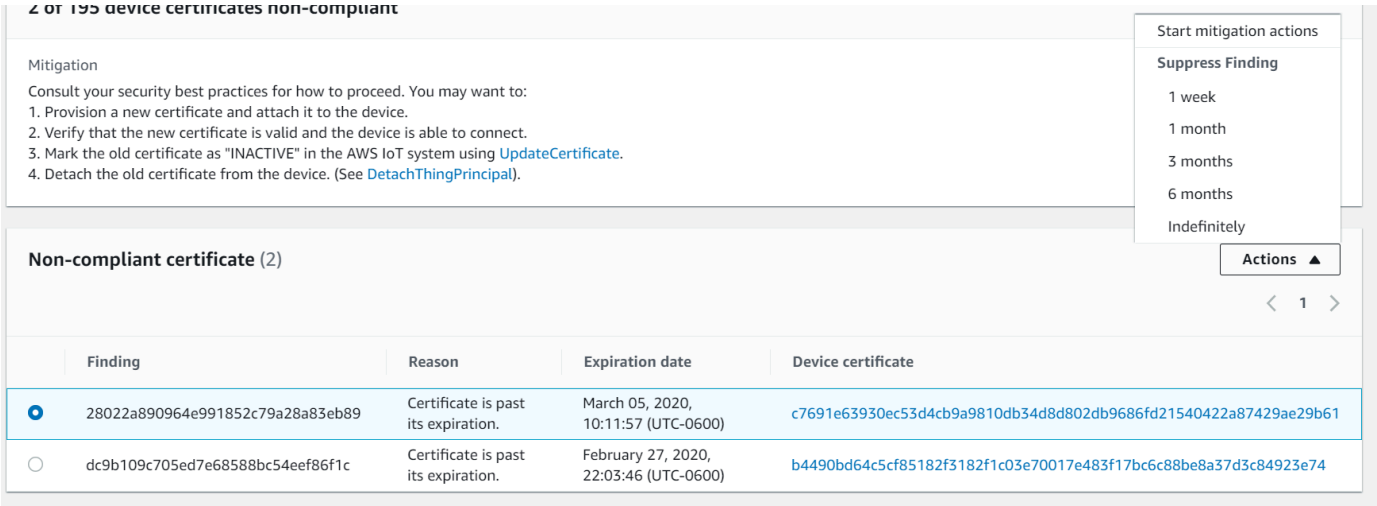


フィールドに入力したら、[作成] を選択します。監査所見の抑制が作成された後に、成功バナーが表示されます。

4. 証明書の 1 つに対して監査所見を非表示にしました。次に、2 番目の証明書の監査所見を抑制する必要があります。ステップ 3 で使用したのと同じ抑制方法を使用することもできますが、デモンストレーションのために、別の方法を使用します。

[防御] の下の左側のサイドバーから、[監査] を選択し、[結果] を選択します。[監査結果] のページで、不適合のリソースの監査を選択します。その後、[非準拠のチェック] でリソースを選択します。この例では、[Device certificate expiring] (デバイス証明書の有効期限が切れます) を選択します。

5. [デバイス証明書の有効期限が切れます] のページ内の [非準拠のポリシー] で、抑制が必要な検出結果の横にあるオプションボタンを選択します。次に、[アクション] ドロップダウンメニューを選択し、検出結果を抑制したい期間を選択します。ここでは、他の証明書で行ったのと同じように 1 week を選択します。[抑制を確認] ウィンドウで、[抑制を有効化] を選択します。



監査所見の抑制が作成された後に、成功バナーが表示されます。現在、両方の監査所見が 1 週間抑制されており、デベロッパーは警告に対処するためのソリューションに取り組んでいます。

CLI で監査所見をカスタマイズする

次のチュートリアルでは、不適合の監査チェックをトリガーする、1 つの失効したデバイス証明書を持つアカウントを使用します。このシナリオでは、デベロッパーが問題に対処する新しい機能をテストしているため、警告を無効にします。証明書用に監査所見の抑制を作成し、次週以降、監査結果が不適合にならないようにします。

次の CLI コマンドを使用します。

- create-audit-suppression
- describe-audit-suppression
- update-audit-suppression
- delete-audit-suppression

- list-audit-suppressions

1. 次のコマンドを使用して、監査を有効にします。

```
aws iot update-account-audit-configuration \
   --audit-check-configurations "{\"DEVICE_CERTIFICATE_EXPIRING_CHECK\":{\"enabled
\":true}}"
```

出力:

なし。

2. 次のコマンドを使用して、DEVICE_CERTIFICATE_EXPIRING_CHECK 監査チェックを対象とするオンデマンド監査を実行します。

```
aws iot start-on-demand-audit-task \
    --target-check-names DEVICE_CERTIFICATE_EXPIRING_CHECK
```

出力:

```
{
    "taskId": "787ed873b69cb4d6cdbae6ddd06996c5"
}
```

3. describe-account-audit-configuration コマンドを使用して、監査設定を記述します。DEVICE_CERTIFICATE_EXPIRING_CHECK の監査チェックをオンにしたことを確認します。

```
aws iot describe-account-audit-configuration
```

出力:

```
{
    "roleArn": "arn:aws:iam::<accountid>:role/service-role/project",
    "auditNotificationTargetConfigurations": {
        "SNS": {
            "targetArn": "arn:aws:sns:us-east-1:<accountid>:project_sns",
            "roleArn": "arn:aws:iam::<accountid>:role/service-role/project",
            "enabled": true
```

```
        }
    },
    "auditCheckConfigurations": {
        "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": false
        },
        "CA_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": false
        },
        "CA_CERTIFICATE_KEY_QUALITY_CHECK": {
            "enabled": false
        },
        "CONFLICTING_CLIENT_IDS_CHECK": {
            "enabled": false
        },
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": true
        },
        "DEVICE_CERTIFICATE_KEY_QUALITY_CHECK": {
            "enabled": false
        },
        "DEVICE_CERTIFICATE_SHARED_CHECK": {
            "enabled": false
        },
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
            "enabled": true
        },
        "IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK": {
            "enabled": false
        },
        "IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK": {
            "enabled": false
        },
        "LOGGING_DISABLED_CHECK": {
            "enabled": false
        },
        "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": false
        },
        "REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": false
        },
        "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": false
```

```
        }
    }
}
```

DEVICE_CERTIFICATE_EXPIRING_CHECK には true の値になっているはずです。

4. list-audit-task コマンドを使用して、完了した監査タスクを識別します。

```
aws iot list-audit-tasks \
    --task-status "COMPLETED" \
    --start-time 2020-07-31 \
    --end-time 2020-08-01
```

出力:

```
{
    "tasks": [
        {
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        }
    ]
}
```

ステップ 1 で実行した監査の taskId は、COMPLETED の taskStatus を持っている必要があります。

5. describe-audit-task コマンドを使用して、前のステップの taskId 出力を使用して完了した監査の詳細を取得します。このコマンドは、監査の詳細を一覧で表示します。

```
aws iot describe-audit-task \
    --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

出力:

```
{
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK",
    "taskStartTime": 1596168096.157,
    "taskStatistics": {
```

```
        "totalChecks": 1,
        "inProgressChecks": 0,
        "waitingForDataCollectionChecks": 0,
        "compliantChecks": 0,
        "nonCompliantChecks": 1,
        "failedChecks": 0,
        "canceledChecks": 0
    },
    "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
    "auditDetails": {
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "checkRunStatus": "COMPLETED_NON_COMPLIANT",
            "checkCompliant": false,
            "totalResourcesCount": 195,
            "nonCompliantResourcesCount": 2
        }
    }
}
```

6. list-audit-findings コマンドを使用して不適合の証明書 ID を検索し、このリソースの監査アラートを一時停止できるようにします。

```
aws iot list-audit-findings \
    --start-time 2020-07-31 \
    --end-time 2020-08-01
```

出力:

```
{
    "findings": [
        {
            "findingId": "296ccd39f806bf9d8f8de20d0ceb33a1",
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "taskStartTime": 1596168096.157,
            "findingTime": 1596168096.651,
            "severity": "MEDIUM",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId": "b4490<shortened>"
                },
                "additionalInfo": {
```

```
                "EXPIRATION_TIME": "1582862626000"
                }
            },
            "reasonForNonCompliance": "Certificate is past its expiration.",
            "reasonForNonComplianceCode": "CERTIFICATE_PAST_EXPIRATION",
            "isSuppressed": false
        },
        {
            "findingId": "37ecb79b7afb53deb328ec78e647631c",
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "taskStartTime": 1596168096.157,
            "findingTime": 1596168096.651,
            "severity": "MEDIUM",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId": "c7691<shortened>"
                },
                "additionalInfo": {
                "EXPIRATION_TIME": "1583424717000"
                }
            },
            "reasonForNonCompliance": "Certificate is past its expiration.",
            "reasonForNonComplianceCode": "CERTIFICATE_PAST_EXPIRATION",
            "isSuppressed": false
        }
    ]
}
```

7. create-audit-suppression コマンドを使用して、*2020 # 8 # 20 #*まで ID *c7691e<shortened>* を持つデバイス証明書の DEVICE_CERTIFICATE_EXPIRING_CHECK 監査チェックの通知を抑制します。

```
aws iot create-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>" \
    --no-suppress-indefinitely \
    --expiration-date 2020-08-20
```

8. list-audit-suppression コマンドを使用して、監査抑制設定を確認し、抑制の詳細を取得します。

```
aws iot list-audit-suppressions
```

出力:

```
{
    "suppressions": [
        {
        "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "resourceIdentifier": {
                "deviceCertificateId": "c7691e<shortened>"
            },
        "expirationDate": 1597881600.0,
        "suppressIndefinitely": false
        }
    ]
}
```

9. update-audit-suppression コマンドを使用して、監査結果の抑制を更新できます。以下の例では、expiration-date を08/21/20 に更新します。

```
aws iot update-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId=c7691e<shortened> \
    --no-suppress-indefinitely \
    --expiration-date 2020-08-21
```

10. delete-audit-suppression コマンドを使用して、監査結果の抑制を削除できます。

```
aws iot delete-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>"
```

削除を確認するには、list-audit-suppressions コマンドを使用します。

```
aws iot list-audit-suppressions
```

出力:

```
{
 "suppressions": []
```

```
}
```

このチュートリアルでは、コンソールと CLI で `Device certificate expiring` チェックを抑制する方法を示しました。監査所見の抑制の詳細については、監査の所見の抑制 を参照してください。

監査

AWS IoT Device Defender の監査では、アカウントやデバイスに関連する設定とポリシーを調べて、セキュリティ対策が実装されていることを確認します。監査は、セキュリティのベストプラクティスまたはアクセスポリシーからの逸脱が検出に役立ちます (例: 複数のデバイスによる同じ ID の使用、1 つのデバイスによる他の多くのデバイス用のデータの読み取りと更新を許可する、権限が過剰なポリシー)。監査は必要に応じて実行するか (オンデマンド監査) 、定期的に実行するようスケジュールできます (スケジュールによる監査)。

AWS IoT Device Defender の監査では、一般的な IoT セキュリティのベストプラクティスとデバイスの脆弱性について、事前に定義された一連のチェックが実行されます。事前に定義されたチェックの例として、複数のデバイスのデータを読み取りまたは更新するアクセス許可を付与するポリシー、ID を共有するデバイス (X.509 証明書)、失効または取り消されたがまだアクティブである証明書があります。

問題の重要度

問題の重要度は、特定された不適合の各インスタンスに関連する懸念のレベルと、修正までの推奨時間を示します。

非常事態

この重要度の不適合の監査チェックでは、緊急の注意が必要な問題が特定されます。重要な問題により、多くの場合、高度な知識がほとんどなく、インサイダーの知識や特別な認証情報を持たない悪意のある人物がアセットに簡単にアクセスしたり制御することができます。

高

この重要度の不適合の監査チェックでは、重要な問題に対処した後、緊急の調査と修復計画が必要になります。重要な問題と同様に、深刻度が高い問題では、悪意のある人物がアセットにアクセスしたり、アセットを管理することがよくあります。しかし、深刻度の高い問題は、多くの場合、悪用が困難です。特別なツール、インサイダーの知識、または特定のセットアップが必要な場合があります。

ミディアム

この重要度の不適合の監査チェックでは、継続的なセキュリティおよびメンテナンス体制の一環として注意が必要な問題を提示します。深刻度が中程度の問題により、セキュリティ制御の誤動作による予期しない停止など、運用に悪影響を及ぼす可能性があります。これらの問題は、悪い

アクターがアセットへのアクセスや制御を制限したり、悪意のあるアクションの一部を容易にする可能性もあります。

低

この重要度の不適合の監査チェックでは、セキュリティのベストプラクティスが見落とされたか、またはバイパスされたことを示します。これら自体は即時に対応の必要なセキュリティ上の影響を引き起こすことはありませんが、これらの失効は悪意のある人物によって悪用される可能性があります。深刻度が中程度の問題と同様に、深刻度が低い問題では、継続的なセキュリティおよびメンテナンス体制の一環として注意が必要です。

次のステップ

実行できる監査チェックのタイプについては、「監査チェック項目」を参照してください。監査に適用されるサービスクォータの詳細については、「Service Quotas」を参照してください。

監査チェック項目

> **Note**
>
> チェックを有効にすると、データ収集が即座に開始されます。収集するデータがアカウントに大量にある場合、チェックを有効にしてからチェックの結果が生成されるまでに時間がかかることがあります。

以下の監査チェック項目がサポートされています。

- アクティブなデバイス証明書の確認のための中間 CA が取り消されました
- 取り消された CA 証明書がアクティブのままです
- デバイス証明書が共有されました
- デバイス証明書のキー品質
- CA 証明書のキー品質
- 認証されていない Cognito ロールの権限が過剰です
- 認証された Cognito ロールの権限が過剰です
- AWS IoT ポリシーの権限が過剰です

- AWS IoT ポリシーが誤って構成されている可能性がある
- ロールエイリアスの権限が過剰です
- ロールエイリアスが未使用サービスへのアクセスを許可します
- CA 証明書の有効期限が切れます
- MQTT クライアント ID の競合
- デバイス証明書の有効期限が切れます
- デバイス証明書の経過時間チェック
- 取り消されたデバイス証明書がアクティブのままです
- ログ記録が無効です

アクティブなデバイス証明書の確認のための中間 CA が取り消されました

このチェックを使用して、中間 CA を取り消してもまだアクティブな関連デバイス証明書をすべて特定します。

このチェックは、CLI および API で `INTERMEDIATE_CA_REVOKED_FOR_ACTIVE_DEVICE_CERTIFICATES_CHECK` として表示されます。

重要度: 非常事態

詳細

このチェックにより不適合が見つかった場合、次の理由コードが返されます。

- INTERMEDIATE_CA_REVOKED_BY_ISSUER

重要な理由

アクティブなデバイス証明書チェックのために取り消された中間 CA は、中間発行 CA が CA チェーンで取り消された AWS IoT Core にアクティブなデバイス証明書があるかどうかを判断することで、デバイスの ID と信頼性を評価します。

取り消し済み中間 CA を使用して、CA チェーン内の他の CA またはデバイス証明書に署名することはできません。中間 CA が取り消された後に、この CA 証明書を使用して署名された証明書を持つ新しく追加されたデバイスは、セキュリティ上の脅威をもたらします。

修正方法

CA 証明書を取り消した後のデバイス証明書の登録アクティビティを確認します。セキュリティのベストプラクティスに従って状況を軽減します。以下を行うことができます。

1. 影響を受けるデバイス用に、別の CA によって署名された新しい証明書をプロビジョニングします。
2. 新しい証明書が有効で、デバイスでそれらの証明書を使用して接続できることを確認します。
3. UpdateCertificate を使用して、AWS IoT で古い証明書を REVOKED としてマークします。緩和アクションを使用して、以下を行うこともできます。
   - 監査結果に `UPDATE_DEVICE_CERTIFICATE` 緩和アクションを適用して、この変更を行います。
   - `ADD_THINGS_TO_THING_GROUP` 緩和アクションを適用して、アクションを実行できるグループにデバイスを追加します。
   - Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、`PUBLISH_FINDINGS_TO_SNS` 緩和アクションを適用します。
   - 中間 CA 証明書を取り消した後のデバイス証明書の登録アクティビティを確認し、その期間に発行された可能性のあるデバイス証明書の取り消しを検討します。この CA 証明書によって署名されたデバイス証明書を一覧表示するには ListRelatedResourcesForAuditFinding を使用できます。デバイス証明書を取り消すには UpdateCertificate を使用できます。
   - 古い証明書をデバイスからデタッチします。(「DetachThingPrincipal」を参照してください)。

   詳細については、「緩和アクション」を参照してください。

取り消された CA 証明書がアクティブのままです

CA 証明書が取り消されましたが、AWS IoT では有効のままです。

このチェックは、CLI および API で `REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK` として表示されます。

重要度: 非常事態

詳細

CA 証明書は、発行機関によって維持されている証明書失効リストで失効済みとマークされていますが、AWS IoT ではまだ「ACTIVE」または「PENDING_TRANSFER」とマークされたままです。

このチェックにより不適合の CA 証明書が見つかった場合、次の理由コードが返されます。

- CERTIFICATE_REVOKED_BY_ISSUER

重要な理由

取り消し済み CA 証明書は、デバイス証明書への署名に使用できなくなります。侵害されたため、取り消された可能性があります。この CA 証明書で署名された証明書を使用して新しく追加されたデバイスはセキュリティ上の脅威になる場合があります。

修正方法

1. UpdateCACertificate を使用して、AWS IoT で CA 証明書を INACTIVE としてマークします。緩和アクションを使用して、以下を行うこともできます。
   - 監査結果に UPDATE_CA_CERTIFICATE 緩和アクションを適用して、この変更を行います。
   - Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

   詳細については、「緩和アクション」を参照してください。
2. CA 証明書を取り消した後のデバイス証明書の登録アクティビティを確認し、その期間に発行された可能性のあるデバイス証明書の取り消しを検討します。この CA 証明書によって署名されたデバイス証明書を一覧表示するには ListCertificatesByCA を使用できます。デバイス証明書を取り消すには UpdateCertificate を使用できます。

デバイス証明書が共有されました

複数の同時接続が、同じ X.509 証明書を使用して AWS IoT サービスに対して認証されます。

このチェックは、CLI および API で DEVICE_CERTIFICATE_SHARED_CHECK として表示されます。

重要度: 非常事態

詳細

オンデマンド監査の一部として実行された場合、このチェック項目は、監査の開始 31 日前からチェックが実行される 2 時間前までの間に接続するためにデバイスにより使用された証明書と clientID を調べます。スケジュールされた監査では、このチェック項目は、前回監査が実行された

時間の 2 時間前からこの監査インスタンスが開始された時間の 2 時間前までのデータを調べます。チェック時にこの条件を緩和するステップを実行した場合、問題が残っているかどうかを判断するために同時接続がいつ行われたかに注目してください。

このチェックにより不適合の証明書が見つかった場合、次の理由コードが返されます。

- CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES

さらに、このチェック項目によって返される結果には、共有証明書の ID、接続するためにこの証明書を使用したクライアントの ID、接続/切断時間が含まれます。最新の結果が最初に一覧表示されます。

重要な理由

AWS IoT で認証するには、各デバイスに一意の証明書が必要です。複数のデバイスで同じ証明書を使用している場合は、デバイスが危険にさらされていることを示している可能性があります。そのデバイスの ID が複製され、システムが危険にさらされている可能性があります。

修正方法

デバイス証明書が侵害されていないことを確認します。侵害されている場合は、セキュリティのベストプラクティスに従って状況を軽減します。

複数のデバイスで同じ証明書を使用している場合は、次の操作を行います。

1. 新しい一意の証明書をプロビジョニングし、各デバイスにアタッチします。
2. 新しい証明書が有効で、デバイスでそれらの証明書を使用して接続できることを確認します。
3. UpdateCertificate を使用して、AWS IoT で古い証明書を REVOKED としてマークします。緩和アクションを使用して、次の操作を行うこともできます。
   - 監査結果に UPDATE_DEVICE_CERTIFICATE 緩和アクションを適用して、この変更を行います。
   - ADD_THINGS_TO_THING_GROUP 緩和アクションを適用して、アクションを実行できるグループにデバイスを追加します。
   - Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

   詳細については、「緩和アクション」を参照してください。
4. 古い証明書を各デバイスからデタッチします。

デバイス証明書のキー品質

AWS IoT 顧客は、AWS IoT メッセージブローカーへの認証に X.509 証明書を使用した TLS 相互認証を利用することがよくあります。これらの証明書と認証局の証明書は、使用する前に AWS IoT アカウントに登録する必要があります。AWS IoT は、登録時にこれらの証明書に対して基本的な健全性チェックを実行します。これらのチェックには、次のものがあります。

- これらは有効な形式であることが必要です。
- これらは、登録された認証局によって署名されている必要があります。
- これらは有効期間内である必要があります (つまり、有効期限が切れていないことが必要です)。
- 暗号化キーのサイズは、必要な最小サイズを満たしている必要があります (RSA キーの場合は、2048 ビット以上である必要があります)。

この監査チェックでは、暗号化キーの品質に関する次の追加テストが提供されます。

- CVE-2008-0166 – Debian ベースのオペレーティングシステムで、0.9.8g-9 より前のバージョンまでの OpenSSL 0.9.8c-1 を使用してキーが生成されたかどうかをチェックします。これらのバージョンの OpenSSL では、予測可能な数値を生成する乱数ジェネレータが使用されているため、リモートの攻撃者が暗号化キーに対するブルートフォース推測攻撃を簡単に行うことができます。
- CVE-2017-15361 – キーが、インフィニオン社のトラステッドプラットフォームモジュール (TPM) ファームウェアでインフィニオン社の RSA ライブラリ 1.02.013 によって生成されたかどうかをチェックします。例えば、0000000000000422 – 4.34 より前のバージョン、000000000000062b – 6.43 より前のバージョン、および 0000000000008521 – 133.33 より前のバージョンなどです。このライブラリは RSA キーの生成を不正に処理するため、攻撃者がターゲットを絞った攻撃によって暗号保護メカニズムを突破することを容易にします。影響を受けるテクノロジーの例としては、TPM 1.2 を使用した BitLocker、YubiKey 4 （4.3.5 以前）の PGP キー生成、および Chrome OS のキャッシュユーザーデータの暗号化機能などがあります。

AWS IoT Device Defender は、これらのテストに失敗した場合、証明書を非準拠として報告します。

このチェックは、CLI および API で `DEVICE_CERTIFICATE_KEY_QUALITY_CHECK` として表示されます。

重要度: 非常事態

詳細

このチェック項目は、「ACTIVE」または「PENDING_TRANSFER」になっているデバイス証明書に適用されます。

このチェックにより不適合の証明書が見つかった場合、次の理由コードが返されます。

- CERTIFICATE_KEY_VULNERABILITY_CVE-2017-15361
- CERTIFICATE_KEY_VULNERABILITY_CVE-2008-0166

重要な理由

デバイスが脆弱な証明書を使用する場合、攻撃者はそのデバイスをより簡単に侵害する可能性があります。

修正方法

デバイス証明書を更新して、既知の脆弱性の証明書を置き換えます。

複数のデバイスで同じ証明書を使用している場合は、次の操作を行います。

1. 新しい一意の証明書をプロビジョニングし、各デバイスにアタッチします。
2. 新しい証明書が有効で、デバイスでそれらの証明書を使用して接続できることを確認します。
3. UpdateCertificate を使用して、AWS IoT で古い証明書を REVOKED としてマークします。緩和アクションを使用して、以下を行うこともできます。
   - 監査結果に UPDATE_DEVICE_CERTIFICATE 緩和アクションを適用して、この変更を行います。
   - ADD_THINGS_TO_THING_GROUP 緩和アクションを適用して、アクションを実行できるグループにデバイスを追加します。
   - Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

   詳細については、「緩和アクション」を参照してください。
4. 古い証明書を各デバイスからデタッチします。

CA 証明書のキー品質

AWS IoT 顧客は、AWS IoT メッセージブローカーへの認証に X.509 証明書を使用した TLS 相互認証を利用することがよくあります。これらの証明書と認証局の証明書は、使用する前に AWS IoT アカウントに登録する必要があります。AWS IoT は、これらの証明書の登録時に、次のような基本的な健全性チェックを実行します。

- この証明書は有効な形式です。
- この証明書の有効期間内です (つまり、有効期限が切れていません)。
- 暗号化キーのサイズは、必要な最小サイズを満たしています (RSA キーの場合は、2048 ビット以上である必要があります)。

この監査チェックでは、暗号化キーの品質に関する次の追加テストが提供されます。

- CVE-2008-0166 – Debian ベースのオペレーティングシステムで、0.9.8g-9 より前のバージョンまでの OpenSSL 0.9.8c-1 を使用してキーが生成されたかどうかをチェックします。これらのバージョンの OpenSSL では、予測可能な数値を生成する乱数ジェネレータが使用されているため、リモートの攻撃者が暗号化キーに対するブルートフォース推測攻撃を簡単に行うことができます。
- CVE-2017-15361 – キーが、インフィニオン社のトラステッドプラットフォームモジュール (TPM) ファームウェアでインフィニオン社の RSA ライブラリ 1.02.013 によって生成されたかどうかをチェックします。例えば、0000000000000422 – 4.34 より前のバージョン、000000000000062b – 6.43 より前のバージョン、および 0000000000008521 – 133.33 より前のバージョンなどです。このライブラリは RSA キーの生成を不正に処理するため、攻撃者がターゲットを絞った攻撃によって暗号保護メカニズムを突破することを容易にします。影響を受けるテクノロジーの例としては、TPM 1.2 を使用した BitLocker、YubiKey 4 (4.3.5 以前) の PGP キー生成、および Chrome OS のキャッシュユーザーデータの暗号化機能などがあります。

AWS IoT Device Defender は、これらのテストに失敗した場合、証明書を非準拠として報告します。

このチェックは、CLI および API で `CA_CERTIFICATE_KEY_QUALITY_CHECK` として表示されます。

重要度: 非常事態

詳細

このチェック項目は、「ACTIVE」または「PENDING_TRANSFER」になっている CA 証明書に適用されます。

このチェックにより不適合の証明書が見つかった場合、次の理由コードが返されます。

- CERTIFICATE_KEY_VULNERABILITY_CVE-2017-15361
- CERTIFICATE_KEY_VULNERABILITY_CVE-2008-0166

重要な理由

この CA 証明書を使用して署名され、新しく追加されたデバイスはセキュリティ上の脅威になる場合があります。

修正方法

1. UpdateCACertificate を使用して、AWS IoT で CA 証明書を INACTIVE としてマークします。緩和アクションを使用して、以下を行うこともできます。
   - 監査結果に `UPDATE_CA_CERTIFICATE` 緩和アクションを適用して、この変更を行います。
   - Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、`PUBLISH_FINDINGS_TO_SNS` 緩和アクションを適用します。

   詳細については、「緩和アクション」を参照してください。
2. CA 証明書を取り消した後のデバイス証明書の登録アクティビティを確認し、その期間に発行された可能性のあるデバイス証明書の取り消しを検討します。(この CA 証明書によって署名されたデバイス証明書を一覧表示するには ListCertificatesByCA を使用します。デバイス証明書を取り消すには UpdateCertificate を使用します)。

認証されていない Cognito ロールの権限が過剰です

認証されていない Amazon Cognito ID プールロールにアタッチされたポリシーは、以下のどの AWS IoT アクションでも実行できるアクセス許可が付与されるため、許可範囲が広過ぎるとみなされます。

- モノを管理または変更する。
- モノの管理データを読み取る。
- モノ以外に関連するデータやリソースを管理する。

または、さまざまなデバイスで以下の AWS IoT アクションを実行するアクセス許可が付与されるためです。

- MQTT を使用して予約済みトピック (シャドウまたはジョブ実行データを含む) に接続/発行/サブスクライブします。
- API コマンドを使用してシャドウまたはジョブ実行データを読み取るか変更する。

一般に、未認証の Amazon Cognito ID プールロールを使用して接続するデバイスには、モノ固有の MQTT トピックを発行/サブスクライブするか、API コマンドを使用してシャドウまたはジョブ実行データに関連するモノ固有のデータを読み取る/変更する限定されたアクセス許可のみ付与してください。

このチェックは、CLI および API で `UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK` として表示されます。

重要度: 非常事態

詳細

このチェック項目の場合、AWS IoT Device Defender は、監査の実行前 31 日間に AWS IoT メッセージブローカーに接続するために使用されたすべての Amazon Cognito ID プールを監査します。接続先の認証済みまたは未認証 Amazon Cognito ID のすべての Amazon Cognito ID プールが監査に含められます。

このチェックにより不適合の未認証 Amazon Cognito ID プールロールが見つかった場合、次の理由コードが返されます。

- ALLOWS_ACCESS_TO_IOT_ADMIN_ACTIONS
- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

重要な理由

認証されていない ID がユーザーにより認証されることはないため、認証された Amazon Cognito ID よりもはるかに大きいリスクを引き起こします。認証されていない ID が侵害された場合、管理アクションを使用して、アカウント設定の変更、リソースの削除、機密データへのアクセスを行う可能性があります。または、デバイス設定に広範にアクセスできるため、アカウント内のすべてのデバイスのシャドウとジョブにアクセスまたは変更する可能性があります。ゲストユーザーは、そのアクセス許可を使用して、フリート全体を侵害したり、メッセージで DDoS 攻撃を開始したりする可能性があります。

修正方法

認証されていない Amazon Cognito ID プールロールにアタッチされたポリシーには、デバイスがそのジョブを実行するのに必要なアクセス許可のみ付与してください。次のステップを推奨します。

1. 新しい適合ロールを作成します。
2. Amazon Cognito ID プールを作成し、適合ロールをアタッチします。
3. ID が新しいプールを使用して AWS IoT にアクセスできることを確認します。
4. 検証が完了したら、不適合としてフラグが設定された Amazon Cognito ID プールに適合ロールをアタッチします。

緩和アクションを使用して、以下を行うこともできます。

- Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、`PUBLISH_FINDINGS_TO_SNS` 緩和アクションを適用します。

詳細については、「緩和アクション」を参照してください。

モノを管理または変更する

次の AWS IoT API アクションは、モノを管理または変更するために使用されます。認証されていない Amazon Cognito ID プールを介して接続するデバイスには、これらのアクションを実行するアクセス許可を付与しないでください。

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`
- `DeleteThing`
- `DetachThingPrincipal`
- `ListThings`
- `ListThingsInThingGroup`
- `RegisterThing`
- `RemoveThingFromThingGroup`
- `UpdateThing`
- `UpdateThingGroupsForThing`

これらのアクションを実行するアクセス許可を付与するロールは、単一のリソースだけが持っていても、不適合とみなされます。

モノの管理データを読み取る

モノデータの読み取りや変更には、以下の AWS IoT API アクションが使用されます。認証されていない Amazon Cognito ID プールを介して接続するデバイスには、これらのアクションを実行するアクセス許可を与えないでください。

- `DescribeThing`
- `ListJobExecutionsForThing`
- `ListThingGroupsForThing`
- `ListThingPrincipals`

Example

- 不適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing"
      ]
    }
  ]
}
```

これにより、デバイスは 1 つのモノのみに対するアクセス許可が付与されていても、指定されたアクションを実行できるようになります。

モノ以外を管理する

未認証の Amazon Cognito ID プールを介して接続するデバイスには、これらのセクションで説明した AWS IoT API アクション以外のアクションを実行するアクセス許可を付与しないでください。未認証の Amazon Cognito ID プールを介して接続するアプリケーションを使用してアカウントを管理するには、デバイスで使用されていない別個の ID プールを作成します。

MQTT トピックにサブスクライブ/発行する

MQTT メッセージは、AWS IoT メッセージブローカー経由で送信され、シャドウステータスやジョブ実行ステータスへのアクセスや変更などのアクションを実行するためにデバイスにより使用されます。MQTT メッセージに接続、発行、またはサブスクライブするアクセス許可をデバイスに付与するポリシーは、以下のようにこれらのアクションを特定のリソースに制限します。

接続

- 不適合:

  ```
  arn:aws:iot:region:account-id:client/*
  ```

  ワイルドカード * を使用すると、どのデバイスでも AWS IoT に接続できます。

  ```
  arn:aws:iot:region:account-id:client/${iot:ClientId}
  ```

  条件キーで iot:Connection.Thing.IsAttached が true に設定されていない限り、これは前の例のワイルドカード * と同等の設定です。

- 適合:

  ```
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [ "iot:Connect" ],
        "Resource": [
          "arn:aws:iot:region:account-id:client/${iot:Connection.Thing.ThingName}"
        ],
        "Condition": {
          "Bool": { "iot:Connection.Thing.IsAttached": "true" }
        }
  ```

```
    }
  ]
}
```

リソースの指定に、接続に使用されるデバイス名と一致する変数が含まれます。条件ステートメントは、MQTT クライアントにより使用される証明書が、使用される名前を持つモノにアタッチされた証明書と一致することを確認することにより、アクセス許可をさらに制限します。

発行

- 不適合:

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

これにより、デバイスはあらゆるデバイスのシャドウを更新できるようになります (* = すべてのデバイス)。

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

これにより、デバイスはあらゆるデバイスのシャドウを読み取り/更新/削除できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Publish" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
      ],
    }
  ]
}
```

リソース仕様にはワイルドカードが含まれていますが、接続するためにモノの名前が使用されているデバイスのシャドウ関連のトピックにのみ一致します。

サブスクライブ

- 不適合:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

これにより、デバイスはすべてのデバイスの予約済みシャドウまたはジョブトピックにサブスクライブできます。

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

前の例と同じですが、# ワイルドカードを使用します。

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/+/shadow/update
```

これにより、デバイスはあらゆるデバイスのシャドウ更新を参照できるようになります (+ = すべてのデバイス)。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Subscribe" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
      ],
    }
  ]
}
```

リソース仕様にはワイルドカードが含まれていますが、接続するためにモノの名前が使用されているデバイスのシャドウ関連のトピックとジョブ関連のトピックにのみ一致します。

受信

- 適合:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

これは、サブスクライブするアクセス許可を持っているトピックからのみデバイスがメッセージを受信するため、許可されます。

シャドウまたはジョブデータを読み取る/変更する

デバイスが API アクションを実行してデバイスシャドウやジョブ実行データにアクセスまたは変更するアクセス許可を付与するポリシーは、これらのアクションを特定のリソースに制限します。API アクションは次のとおりです。

- DeleteThingShadow
- GetThingShadow
- UpdateThingShadow
- DescribeJobExecution
- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

Example

- 不適合:

```
arn:aws:iot:region:account-id:thing/*
```

これにより、デバイスがあらゆるモノで指定されたアクションを実行できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DeleteThingShadow",
          "iot:GetThingShadow",
```

```
            "iot:UpdateThingShadow",
            "iotjobsdata:DescribeJobExecution",
            "iotjobsdata:GetPendingJobExecutions",
            "iotjobsdata:StartNextPendingJobExecution",
            "iotjobsdata:UpdateJobExecution"
        ],
        "Resource": [
          "arn:aws:iot:region:account-id:/thing/MyThing1",
          "arn:aws:iot:region:account-id:/thing/MyThing2"
        ]
      }
    ]
}
```

これにより、デバイスが 2 つのモノでのみ指定されたアクションを実行できるようになります。

認証された Cognito ロールの権限が過剰です

認証済み Amazon Cognito ID プールロールにアタッチされたポリシーは、以下の AWS IoT アクションを実行できるアクセス許可が付与されるため、許可範囲が広過ぎるとみなされます。

- モノを管理または変更する。
- モノ以外に関連するデータやリソースを管理する。

または、さまざまなデバイスで以下の AWS IoT アクションを実行するアクセス許可が付与されるためです。

- モノの管理データを読み取る。
- MQTT を使用して予約済みトピック (シャドウまたはジョブ実行データを含む) に接続/発行/サブスクライブする。
- API コマンドを使用してシャドウまたはジョブ実行データを読み取るか変更する。

一般に、認証済み Amazon Cognito ID プールロールを使用して接続するデバイスには、モノ固有の管理データを読み取る、モノ固有の MQTT トピックを発行/サブスクライブする、または API コマンドを使用してシャドウまたはジョブ実行データに関連するモノ固有のデータを読み取る/変更する限定されたアクセス許可のみ付与してください。

このチェックは、CLI および API で
AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK として表示されます。

重要度: 非常事態

詳細

このチェック項目の場合、AWS IoT Device Defender は、監査の実行前 31 日間に AWS IoT メッセージブローカーに接続するために使用されたすべての Amazon Cognito ID プールを監査します。接続先の認証済みまたは未認証 Amazon Cognito ID のすべての Amazon Cognito ID プールが監査に含められます。

このチェックにより不適合の認証済み Amazon Cognito ID プールロールが見つかった場合、次の理由コードが返されます。

- ALLOWS_BROAD_ACCESS_TO_IOT_THING_ADMIN_READ_ACTIONS
- ALLOWS_ACCESS_TO_IOT_NON_THING_ADMIN_ACTIONS
- ALLOWS_ACCESS_TO_IOT_THING_ADMIN_WRITE_ACTIONS

重要な理由

認証されている ID が侵害された場合、管理アクションを使用して、アカウント設定の変更、リソースの削除、機密データへのアクセスを行う可能性があります。

修正方法

認証済みの Amazon Cognito ID プールロールにアタッチされたポリシーには、デバイスがそのジョブを実行するのに必要なアクセス許可のみ付与してください。次のステップを推奨します。

1. 新しい適合ロールを作成します。
2. Amazon Cognito ID プールを作成し、適合ロールをアタッチします。
3. ID が新しいプールを使用して AWS IoT にアクセスできることを確認します。
4. 検証が完了したら、不適合としてフラグが設定された Amazon Cognito ID プールにロールをアタッチします。

緩和アクションを使用して、以下を行うこともできます。

- Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

詳細については、「緩和アクション」を参照してください。

モノを管理または変更する

次の AWS IoT API アクションは、モノを管理または変更するために使用されるため、これらのアクションを実行するアクセス許可は、認証済みの Amazon Cognito ID プールを介して接続するデバイスには付与しないでください。

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`
- `DeleteThing`
- `DetachThingPrincipal`
- `ListThings`
- `ListThingsInThingGroup`
- `RegisterThing`
- `RemoveThingFromThingGroup`
- `UpdateThing`
- `UpdateThingGroupsForThing`

これらのアクションを実行するアクセス許可を付与するロールは、単一のリソースだけが持っていても、不適合とみなされます。

モノ以外を管理する

認証済みの Amazon Cognito ID プールを介して接続するデバイスには、これらのセクションで説明した AWS IoT API アクション以外のアクションを実行するアクセス許可を付与しないでください。認証済みの Amazon Cognito ID プールを介して接続するアプリケーションを使用してアカウントを管理するには、デバイスにより使用されていない別個の ID プールを作成します。

モノの管理データを読み取る

次の AWS IoT API アクションは、モノのデータを読み取るために使用されるため、認証済みの Amazon Cognito ID プールを介して接続するデバイスには限定されたモノのセットでのみこれらのアクションを実行するアクセス許可を付与してください。

- DescribeThing
- ListJobExecutionsForThing
- ListThingGroupsForThing
- ListThingPrincipals

- 不適合:

```
arn:aws:iot:region:account-id:thing/*
```

これにより、デバイスがあらゆるモノで指定されたアクションを実行できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing"
      ]
    }
  ]
}
```

これにより、デバイスが 1 つのモノでのみ指定されたアクションを実行できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
            "iot:DescribeThing",
            "iot:ListJobExecutionsForThing",
            "iot:ListThingGroupsForThing",
            "iot:ListThingPrincipals"
        ],
        "Resource": [
          "arn:aws:iot:region:account-id:/thing/MyThing*"
        ]
      }
    ]
}
```

リソースがワイルドカード (*) を使用して指定されていますが、前に特定の文字列が付いており、指定されたプレフィックスを持つ名前が付いたモノのセットへのアクセスに制限されるため、これは適合しています。

- 不適合:

```
arn:aws:iot:region:account-id:thing/*
```

これにより、デバイスがあらゆるモノで指定されたアクションを実行できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing"
      ]
    }
  ]
}
```

これにより、デバイスが 1 つのモノでのみ指定されたアクションを実行できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing*"
      ]
    }
  ]
}
```

リソースがワイルドカード (*) を使用して指定されていますが、前に特定の文字列が付いており、指定されたプレフィックスを持つ名前が付いたモノのセットへのアクセスに制限されるため、これは適合しています。

MQTT トピックにサブスクライブ/発行する

MQTT メッセージは、AWS IoT メッセージブローカー経由で送信され、シャドウステータスやジョブ実行ステータスへのアクセスや変更などのさまざまなアクションを実行するためにデバイスにより使用されます。MQTT メッセージに接続、発行、またはサブスクライブするアクセス許可をデバイスに付与するポリシーは、以下のようにこれらのアクションを特定のリソースに制限します。

接続

- 不適合:

```
arn:aws:iot:region:account-id:client/*
```

ワイルドカード * を使用すると、どのデバイスでも AWS IoT に接続できます。

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

条件キーで iot:Connection.Thing.IsAttached が true に設定されていない限り、これは前の例のワイルドカード * と同等の設定です。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Connect" ],
      "Resource": [
        "arn:aws:iot:region:account-id:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": { "iot:Connection.Thing.IsAttached": "true" }
      }
    }
  ]
}
```

リソース仕様には、接続に使用されるデバイス名に一致する変数が含まれており、条件ステートメントは、MQTT クライアントにより使用される証明書が、使用される名前を持つモノにアタッチされた証明書と一致することを確認することにより、アクセス許可をさらに制限します。

発行

- 不適合:

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

これにより、デバイスはあらゆるデバイスのシャドウを更新できるようになります (* = すべてのデバイス)。

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

これにより、デバイスはあらゆるデバイスのシャドウを読み取り/更新/削除できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Publish" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
      ],
    }
  ]
}
```

リソース仕様にはワイルドカードが含まれていますが、接続するためにモノの名前が使用されているデバイスのシャドウ関連のトピックにのみ一致します。

サブスクライブ

- 不適合:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

これにより、デバイスはすべてのデバイスの予約済みシャドウまたはジョブトピックにサブスクライブできます。

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/#
```

前の例と同じですが、# ワイルドカードを使用します。

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/+/shadow/update
```

これにより、デバイスはあらゆるデバイスのシャドウ更新を参照できるようになります (+ = すべてのデバイス)。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Subscribe" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
      ],
    }
  ]
}
```

リソース仕様にはワイルドカードが含まれていますが、接続するためにモノの名前が使用されているデバイスのシャドウ関連のトピックとジョブ関連のトピックにのみ一致します。

受信

- 適合:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

これは、サブスクライブするアクセス許可を持っているトピックからのみデバイスがメッセージを受信するため、問題ありません。

シャドウまたはジョブデータを読み取る/変更する

デバイスが API アクションを実行してデバイスシャドウやジョブ実行データにアクセスまたは変更するアクセス許可を付与するポリシーは、これらのアクションを特定のリソースに制限します。API アクションは次のとおりです。

- `DeleteThingShadow`
- `GetThingShadow`
- `UpdateThingShadow`
- `DescribeJobExecution`
- `GetPendingJobExecutions`

- StartNextPendingJobExecution
- UpdateJobExecution

例

- 不適合:

```
arn:aws:iot:region:account-id:thing/*
```

これにより、デバイスがあらゆるモノで指定されたアクションを実行できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DeleteThingShadow",
          "iot:GetThingShadow",
          "iot:UpdateThingShadow",
          "iot:DescribeJobExecution",
          "iot:GetPendingJobExecutions",
          "iot:StartNextPendingJobExecution",
          "iot:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing1",
        "arn:aws:iot:region:account-id:/thing/MyThing2"
      ]
    }
  ]
}
```

これにより、デバイスが 2 つのモノでのみ指定されたアクションを実行できるようになります。

AWS IoT ポリシーの権限が過剰です

AWS IoT ポリシーで付与されるアクセス許可は、広すぎるか、制限がありません。さまざまなデバイスとの間で MQTT メッセージを送受信するアクセス許可を付与するか、さまざまなデバイスのシャドウおよびジョブ実行データにアクセスまたは変更するアクセス許可を付与します。

一般に、デバイスのポリシーは、そのデバイスにのみ関連付けられており、他のデバイスにはまったく、あるいはわずかしか関連付けられていないリソースへのアクセスを付与する必要があります。一部例外はありますが、そのようなポリシーでリソースを指定するためにワイルドカード (「*」など) を使用すると、広すぎるか、制限がないとみなされます。

このチェックは、CLI および API で `IOT_POLICY_OVERLY_PERMISSIVE_CHECK` として表示されます。

重要度: 非常事態

詳細

このチェックにより非準拠の AWS IoT ポリシーが見つかると、次の理由コードが返されます。

- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

重要な理由

アクセス許可が過剰なポリシーを持つ証明書、Amazon Cognito ID、またはモノのグループは、侵害された場合、アカウント全体のセキュリティに影響を与える可能性があります。攻撃者がそのような広範なアクセス権を使用してすべてのデバイスのシャドウ、ジョブ、またはジョブ実行を読み取りまたは変更する可能性があります。または、攻撃者が侵害された証明書を使用し、悪意のあるデバイスに接続したり、ネットワーク上で DDoS 攻撃を開始したりする可能性があります。

修正方法

モノ、モノのグループ、その他のエンティティにアタッチされた不適合のポリシーを修正するには、以下のステップを実行してください。

1. ポリシーの新しい準拠バージョンを作成するには、CreatePolicyVersion を使用します。`setAsDefault` フラグを true に設定します。(これにより、この新しいバージョンは、ポリシーを使用するすべてのエンティティで動作します。)

2. ポリシーのアタッチ先のターゲット (証明書、モノのグループ) のリストを取得し、グループに含まれているデバイスや、接続に証明書を使用するデバイスを確認するには、ListTargetsForPolicy を使用します。
3. 関連付けられているすべてのデバイスが AWS IoT に接続できることを確認します。デバイスが接続できない場合は、SetPolicyVersion を使用してデフォルトのポリシーを以前のバージョンにロールバックして、もう一度試してください。

緩和アクションを使用して、以下を実行できます。

- 監査結果に REPLACE_DEFAULT_POLICY_VERSION 緩和アクションを適用して、この変更を行います。
- Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

詳細については、「緩和アクション」を参照してください。

AWS IoT Core ポリシー変数を使用して、ポリシー内の AWS IoT リソースを動的に参照します。

MQTT のアクセス許可

MQTT メッセージは、AWS IoT メッセージブローカー経由で送信され、シャドウステータスやジョブ実行ステータスへのアクセスや変更などのアクションを実行するためにデバイスにより使用されます。MQTT メッセージに接続、発行、またはサブスクライブするアクセス許可をデバイスに付与するポリシーは、以下のようにこれらのアクションを特定のリソースに制限します。

接続

- 不適合:

  ```
  arn:aws:iot:region:account-id:client/*
  ```

  ワイルドカード * を使用すると、どのデバイスでも AWS IoT に接続できます。

  ```
  arn:aws:iot:region:account-id:client/${iot:ClientId}
  ```

  条件キーで iot:Connection.Thing.IsAttached が true に設定されていない限り、これは前の例のワイルドカード * と同等の設定です。
- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Connect" ],
      "Resource": [
        "arn:aws:iot:region:account-id:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": { "iot:Connection.Thing.IsAttached": "true" }
      }
    }
  ]
}
```

リソースの指定に、接続に使用されるデバイス名と一致する変数が含まれます。条件ステートメントは、MQTT クライアントにより使用される証明書が、使用される名前を持つモノにアタッチされた証明書と一致することを確認することにより、アクセス許可をさらに制限します。

発行

- 不適合:

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

これにより、デバイスはあらゆるデバイスのシャドウを更新できるようになります (* = すべてのデバイス)。

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

これにより、デバイスはあらゆるデバイスのシャドウを読み取り/更新/削除できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [ "iot:Publish" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
      ],
    }
  ]
}
```

リソース仕様にはワイルドカードが含まれていますが、接続するためにモノの名前が使用されているデバイスのシャドウ関連のトピックにのみ一致します。

サブスクライブ

- 不適合:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

これにより、デバイスはすべてのデバイスの予約済みシャドウまたはジョブトピックにサブスクライブできます。

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

前の例と同じですが、# ワイルドカードを使用します。

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/+/shadow/update
```

これにより、デバイスはあらゆるデバイスのシャドウ更新を参照できるようになります (+ = すべてのデバイス)。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Subscribe" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
```

```
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
      ],
    }
  ]
}
```

リソース仕様にはワイルドカードが含まれていますが、接続するためにモノの名前が使用されているデバイスのシャドウ関連のトピックとジョブ関連のトピックにのみ一致します。

受信

- 適合:

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

これは、サブスクライブするアクセス許可を持っているトピックからのみデバイスがメッセージを受信するため、問題ありません。

シャドウとジョブのアクセス許可

デバイスが API アクションを実行してデバイスシャドウやジョブ実行データにアクセスまたは変更するアクセス許可を付与するポリシーは、これらのアクションを特定のリソースに制限します。API アクションは次のとおりです。

- DeleteThingShadow
- GetThingShadow
- UpdateThingShadow
- DescribeJobExecution
- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

例

- 不適合:

```
arn:aws:iot:region:account-id:thing/*
```

これにより、デバイスがあらゆるモノで指定されたアクションを実行できるようになります。

- 適合:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DeleteThingShadow",
          "iot:GetThingShadow",
          "iot:UpdateThingShadow",
          "iotjobsdata:DescribeJobExecution",
          "iotjobsdata:GetPendingJobExecutions",
          "iotjobsdata:StartNextPendingJobExecution",
          "iotjobsdata:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing1",
        "arn:aws:iot:region:account-id:/thing/MyThing2"
      ]
    }
  ]
}
```

これにより、デバイスが 2 つのモノでのみ指定されたアクションを実行できるようになります。

AWS IoT ポリシーが誤って構成されている可能性がある

AWS IoT ポリシーが誤って構成されている可能性があることが確認されました。過度に寛容なポリシーなど、誤って構成されたポリシーは、意図しないリソースへのデバイスアクセスを許可するなどのセキュリティインシデントを引き起こす可能性があります。

AWS IoT ポリシーが誤って構成されている可能性があるチェックは、ポリシーを更新する前に、意図したアクションのみが許可されていることを確認するための警告です。

このチェックは、CLI および API で `IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK` として表示されます。

重大度: 中

詳細

このチェックで AWS IoT ポリシーが誤って設定されている可能性があることが検出されると、AWS IoT は次の理由コードを返します。

- POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT
- TOPIC_FILTERS_INTENDED_TO_DENY_ALLOWED_USING_WILDCARDS

重要な理由

ポリシーの設定を誤ると、必要以上のアクセス許可がデバイスに与えられ、意図しない結果につながる可能性があります。リソースへのアクセスを制限し、セキュリティ上の脅威を防ぐために、ポリシーを慎重に検討することをお勧めします。

ポリシーの拒否ステートメントの例に MQTT ワイルドカードが含まれています

AWS IoT ポリシーが誤って構成されている可能性があるチェックでは、拒否ステートメント内の MQTT ワイルドカード文字 (+ または #) を検査します。ワイルドカードは AWS IoT ポリシーによってリテラル文字列として扱われるため、ポリシーが過度に制限される可能性があります。

次の例は、ポリシーで MQTT ワイルドカード # を使用して、`building/control_room` に関連するトピックへのサブスクライブを拒否することを目的としています。ただし、MQTT ワイルドカードは AWS IoT ポリシーではワイルドカードの意味を持たず、デバイスは `building/control_room/data1` にサブスクライブできます。

AWS IoT ポリシーが誤って構成されている可能性があるチェックでは、このポリシーに理由コード `POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT` のフラグが付けられます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/control_room/#"
    },
```

```
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:region:account-id:topic/building/*"
    }
  ]
}
```

以下は、適切に構成されたポリシーの例です。デバイスには、building/control_room/ のサブトピックにサブスクライブするアクセス許可がなく、building/control_room/ のサブトピックからメッセージを受信するアクセス許可がありません。

```
{
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:region:account-id:topicfilter/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/control_room/*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:region:account-id:topic/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:region:account-id:topic/building/control_room/*"
    }
  ]
}
```

ワイルドカードの使用を許可することを拒否することを目的したトピック フィルターの例

次のポリシー例は、リソース building/control_room/* を拒否することにより、building/control_room に関連するトピックへのサブスクライブを拒否することを目的としています。

ただし、デバイスは building/# にサブスクライブするリクエストを送信し、building/control_room/data1 を含む building に関連するすべてのトピックからメッセージを受信できます。

AWS IoT ポリシーが誤って構成されている可能性があるチェックでは、このポリシーに理由コード TOPIC_FILTERS_INTENDED_TO_DENY_ALLOWED_USING_WILDCARDS のフラグが付けられます。

次のポリシーの例には、building/control_room topics でメッセージを受信するアクセス許可が含まれています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/control_room/*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:region:account-id:topic/building/*"
    }
  ]
}
```

以下は、適切に構成されたポリシーの例です。デバイスには、building/control_room/ のサブトピックにサブスクライブするアクセス許可がなく、building/control_room/ のサブトピックからメッセージを受信するアクセス許可がありません。

```
{
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/*"
```

```
    },
    {
      "Effect": "Deny",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/control_room/*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:region:account-id:topic/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:region:account-id:topic/building/control_room/*"
    }
  ]
}
```

> **Note**
>
> このチェックでは、誤検出が報告される場合があります。フラグが立てられたポリシーをすべて評価し、監査抑制を使用して誤検出のリソースにマークを付けることをお勧めします。

修正方法

このチェックにより、誤って構成されている可能性のあるポリシーにフラグが付けられるため、誤検出が発生する可能性があります。今後フラグが立てられないように、監査抑制を使用して誤検出をマークします。

以下のステップに従ってモノ、モノのグループ、その他のエンティティにアタッチされた不適合のポリシーを修正することもできます。

1. ポリシーの新しい準拠バージョンを作成するには、CreatePolicyVersion を使用します。`setAsDefault` フラグを true に設定します。(これにより、この新しいバージョンは、ポリシーを使用するすべてのエンティティで動作します。)

   一般的なユースケース向けの AWS IoT ポリシーの作成例については、AWS IoT Core 開発者ガイドの「パブリッシュ/サブスクライブポリシーの例」を参照してください。

2. 関連付けられているすべてのデバイスが AWS IoT に接続できることを確認します。デバイスが接続できない場合は、SetPolicyVersion を使用してデフォルトのポリシーを以前のバージョンにロールバックして、もう一度試してください。

緩和アクションを使用して、以下を実行できます。

- 監査結果に REPLACE_DEFAULT_POLICY_VERSION 緩和アクションを適用して、この変更を行います。
- Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

詳細については、「緩和アクション」を参照してください。

「AWS IoT Core デベロッパーガイド」の IoT Core ポリシー変数を使用して、ポリシー内の AWS IoT リソースを動的に参照します。

ロールエイリアスの権限が過剰です

AWS IoT ロールエイリアスは、接続されたデバイスが X.509 証明書を使用して AWS IoT を認証し、AWS IoT ロールエイリアスに関連付けられた IAM ロールから短期間の AWS 認証情報を取得するためのメカニズムを提供します。これらの認証情報のアクセス許可は、認証コンテキスト変数を持つアクセスポリシーを使用して範囲を限定する必要があります。ポリシーが正しく設定されていない場合、特権攻撃のエスカレーションにさらされる可能性があります。この監査チェックにより、AWS IoT ロールエイリアスによって提供される一時的な認証情報が過度に許容されないことが保証されます。

このチェックは、次のいずれかの条件が見つかった場合にトリガーされます。

- このポリシーは、このロールエイリアスによって過去 1 年に使用されたすべてのサービスに対する管理アクセス許可を提供します (例えば、「iot:*」、「dynamodb:*」、「iam:*」など)。
- このポリシーは、モノメタデータアクションへの幅広いアクセス、制限された AWS IoT アクションへのアクセス、または AWS IoT データプレーンアクションへの幅広いアクセスを提供します。
- このポリシーは、「iam」、「cloudtrail」、「guardduty」、「inspector」、「trustedadvisor」などのセキュリティ監査サービスへのアクセスを提供します。

このチェックは、CLI および API で IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK として表示されます。

重要度: 非常事態

詳細

このチェックにより不適合の IoT ポリシーが見つかった場合、次の理由コードが返されます。

- ALLOWS_BROAD_ACCESS_TO_USED_SERVICES
- ALLOWS_ACCESS_TO_SECURITY_AUDITING_SERVICES
- ALLOWS_BROAD_ACCESS_TO_IOT_THING_ADMIN_READ_ACTIONS
- ALLOWS_ACCESS_TO_IOT_NON_THING_ADMIN_ACTIONS
- ALLOWS_ACCESS_TO_IOT_THING_ADMIN_WRITE_ACTIONS
- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

重要な理由

デバイスが通常のオペレーションを実行するために必要なアクセス許可を制限することで、デバイスが侵害された場合のアカウントのリスクを軽減できます。

修正方法

モノ、モノのグループ、その他のエンティティにアタッチされた不適合のポリシーを修正するには、以下のステップを実行してください。

1. 「AWS IoT Core 認証情報プロバイダーを使用して、AWS サービスの直接呼び出しを認証」の手順に従って、ロールエイリアスにより制限の厳しいポリシーを適用します。

緩和アクションを使用して、以下を実行できます。

- Amazon SNS メッセージに対するレスポンスとしてカスタムアクションを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

詳細については、「緩和アクション」を参照してください。

ロールエイリアスが未使用サービスへのアクセスを許可します

AWS IoT ロールエイリアスは、接続されたデバイスが X.509 証明書を使用して AWS IoT を認証し、AWS IoT ロールエイリアスに関連付けられた IAM ロールから短期間の AWS 認証情報を取得

するためのメカニズムを提供します。これらの認証情報のアクセス許可は、認証コンテキスト変数を持つアクセスポリシーを使用して範囲を限定する必要があります。ポリシーが正しく設定されていない場合、特権攻撃のエスカレーションにさらされる可能性があります。この監査チェックにより、AWS IoT ロールエイリアスによって提供される一時的な認証情報が過度に許容されないことが保証されます。

このチェックは、昨年 AWS IoT デバイスで使用されていないサービスに、ロールエイリアスがアクセスできる場合にトリガーされます。例えば、過去 1 年間に AWS IoT のみを使用したロールエイリアスにリンクされた IAM ロールがある場合、監査はレポートしますが、ロールにアタッチされたポリシーは `"iam:getRole"` および `"dynamodb:PutItem"` にもアクセス許可を付与します。

このチェックは、CLI および API で `IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK` として表示されます。

重大度: 中

詳細

このチェックにより非準拠の AWS IoT ポリシーが見つかった場合、次の理由コードが返されます。

- ALLOWS_ACCESS_TO_UNUSED_SERVICES

重要な理由

デバイスが通常のオペレーションを実行するために必要なサービスにアクセス許可を制限することで、デバイスが侵害された場合のアカウントのリスクを軽減できます。

修正方法

モノ、モノのグループ、その他のエンティティにアタッチされた不適合のポリシーを修正するには、以下のステップを実行してください。

1. 「AWS IoT Core 認証情報プロバイダーを使用して、AWS サービスの直接呼び出しを認証」の手順に従って、ロールエイリアスにより制限の厳しいポリシーを適用します。

緩和アクションを使用して、以下を実行できます。

- Amazon SNS メッセージに対するレスポンスとしてカスタムアクションを実装する場合は、`PUBLISH_FINDINGS_TO_SNS` 緩和アクションを適用します。

詳細については、「緩和アクション」を参照してください。

CA 証明書の有効期限が切れます

CA 証明書が 30 日以内に有効期限が切れるか、既に切れています。

このチェックは、CLI および API で CA_CERTIFICATE_EXPIRING_CHECK として表示されます。

重大度: 中

詳細

このチェック項目は、「ACTIVE」または「PENDING_TRANSFER」になっている CA 証明書に適用されます。

このチェックにより不適合の CA 証明書が見つかった場合、次の理由コードが返されます。

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

重要な理由

期限切れ CA 証明書は、新しいデバイス証明書への署名に使用しないでください。

修正方法

処理方法については、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。以下を行うことができます。

1. AWS IoT に新しい CA 証明書を登録します。
2. 新しい CA 証明書を使用して、デバイス証明書に署名できることを確認します。
3. UpdateCACertificate を使用して、AWS IoT で古い CA 証明書を INACTIVE としてマークします。緩和アクションを使用して、次の操作を行うこともできます。
   - 監査結果に UPDATE_CA_CERTIFICATE 緩和アクションを適用して、この変更を行います。
   - Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

   詳細については、「緩和アクション」を参照してください。

MQTT クライアント ID の競合

複数のデバイスが同じクライアント ID を使用して接続します。

このチェックは、CLI および API で CONFLICTING_CLIENT_IDS_CHECK として表示されます。

重大度: 高

詳細

同じクライアント ID を使用して複数の接続が行われ、既に接続されたデバイスが切断されます。MQTT 仕様では、クライアント ID あたり 1 つのアクティブな接続のみが許可されるため、同じクライアント ID を使用して別のデバイスが接続した場合、前の接続が中断されます。

オンデマンド監査の一部として実行された場合、このチェック項目は、監査の開始前 31 日間に接続するために clientID がどのように使用されたかを調べます。スケジュールされた監査では、このチェック項目は、前回監査が実行された時間からこの監査インスタンスが開始された時間までのデータを調べます。チェック時にこの条件を緩和するステップを実行した場合、問題が残っているかどうかを判断するために接続/切断がいつ行われたかに注目してください。

このチェックにより不適合が見つかった場合、次の理由コードが返されます。

- DUPLICATE_CLIENT_ID_ACROSS_CONNECTIONS

このチェック項目によって返される結果には、接続に使用された clientID、プリンシパル ID、切断時間も含まれます。最新の結果が最初に一覧表示されます。

重要な理由

ID が競合するデバイスは、継続して強制的に再接続されるため、メッセージが失われたり、デバイスが接続できなくなる可能性があります。

これは、デバイスまたはデバイスの認証情報が侵害されたか、DDoS 攻撃の一部であることを示している可能性があります。アカウントでデバイスが正しく設定されていない可能性や、デバイスが接続不良のために 1 分あたり数回再接続を強制された可能性もあります。

修正方法

AWS IoT で各デバイスを一意のモノとして登録し、モノの名前をクライアント ID として使用して接続します。または、MQTT を介してデバイスを接続するときに、クライアント ID として UUID を使用します。緩和アクションを使用して、以下を行うこともできます。

- Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

詳細については、「緩和アクション」を参照してください。

デバイス証明書の有効期限が切れます

デバイス証明書が設定されたしきい値期間内に期限切れになるか、または既に期限切れになっています。証明書の有効期限チェックのしきい値は、デフォルト値 30 日で、30 日 (最小) から 3,652 日 (最大 10 年) の間で設定できます。

このチェックは、CLI および API で DEVICE_CERTIFICATE_EXPIRING_CHECK として表示されます。

重大度: 中

詳細

このチェック項目は、「ACTIVE」または「PENDING_TRANSFER」になっているデバイス証明書に適用されます。

このチェックにより不適合のデバイス証明書が見つかった場合、次の理由コードが返されます。

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

重要な理由

デバイス証明書は、有効期限が切れたら使用しないでください。

デバイス証明書の有効期限チェックの設定

この設定により、デバイスフリート全体で有効期限が近づいている証明書をモニタリングし、アラートを受信できます。例えば、証明書の有効期限が 30 日以内になった際に通知を受け取る場合は、次のようにチェックを設定できます。

```
{
    "roleArn": "your-audit-role-arn",
    "auditCheckConfigurations": {
```

```
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": true,
            "configuration": {
                "CERT_EXPIRATION_THRESHOLD_IN_DAYS": "30"
            }
        }
    }
}
```

修正方法

処理方法については、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。以下を行うことができます。

1. 新しい証明書をプロビジョニングし、デバイスにアタッチします。
2. 新しい証明書が有効で、デバイスが接続するためにその証明書を使用できることを確認します。
3. UpdateCertificate を使用して、AWS IoT で古い証明書を INACTIVE としてマークします。緩和アクションを使用して、以下を行うこともできます。
   - 監査結果に UPDATE_DEVICE_CERTIFICATE 緩和アクションを適用して、この変更を行います。
   - ADD_THINGS_TO_THING_GROUP 緩和アクションを適用して、アクションを実行できるグループにデバイスを追加します。
   - Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

   詳細については、「緩和アクション」を参照してください。
4. 古い証明書をデバイスからデタッチします。(「DetachThingPrincipal」を参照してください)。

デバイス証明書の経過時間チェック

この監査チェックでは、指定した日数以上の期間、デバイス証明書がアクティブになったときにアラートが表示されます。このチェックにより、証明書のステータスを常に把握し、証明書の有効期限が切れる時期に関係なく、定期的にタイムリーなアクションが可能になり、証明書の侵害のリスクを軽減することでセキュリティを向上させるができます。

証明書の経過時間チェックのしきい値は、デフォルト値 365 日で、30 日 (最小) から 3,652 日 (最大 10 年) の間で設定できます。

このチェックは、CLI および API で DEVICE_CERTIFICATE_AGE_CHECK として表示されます。このチェックはデフォルトで無効になっています。重要度: 低

詳細

このチェック項目は、「ACTIVE」または「PENDING_TRANSFER」になっているデバイス証明書に適用されます。このチェックにより不適合のデバイス証明書が見つかった場合、次の理由コードが返されます。

- CERTIFICATE_PAST_AGE_THRESHOLD

デバイス証明書の経過時間チェックの設定

この設定により、フリートの特定のニーズに合わせて証明書ローテーションアラートを調整できるため、すべてのデバイスで強力なセキュリティ体制を維持できます。このチェックは、UpdateAccountAuditConfiguration API を使用して設定できます。例えば、証明書が 365 日以上アクティブになったときにアラートを受け取る場合は、次のようにチェックを設定できます。

```
{
    "roleArn": "your-audit-role-arn",
    "auditCheckConfigurations": {
        "DEVICE_CERTIFICATE_AGE_CHECK": {
            "enabled": true,
            "configuration": {
                "CERT_AGE_THRESHOLD_IN_DAYS": "365"
            }
        }
    }
}
```

取り消されたデバイス証明書がアクティブのままです

取り消されたデバイス証明書がアクティブのままです。

このチェックは、CLI および API で REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK として表示されます。

重大度: 中

詳細

デバイス証明書は、CA の 証明書失効リストに含まれていますが、AWS IoT ではまだアクティブなままです。

このチェック項目は、「ACTIVE」または「PENDING_TRANSFER」になっているデバイス証明書に適用されます。

このチェックにより不適合が見つかった場合、次の理由コードが返されます。

- CERTIFICATE_REVOKED_BY_ISSUER

重要な理由

デバイス証明書が取り消されるのは、通常侵害されたためです。エラーや見落としのため、AWS IoT でまだ取り消されていない可能性があります。

修正方法

デバイス証明書が侵害されていないことを確認します。侵害されている場合は、セキュリティのベストプラクティスに従って状況を軽減します。以下を行うことができます。

1. デバイスの新しい証明書をプロビジョニングします。
2. 新しい証明書が有効で、デバイスが接続するためにその証明書を使用できることを確認します。
3. UpdateCertificate を使用して、AWS IoT で古い証明書を REVOKED としてマークします。緩和アクションを使用して、以下を行うこともできます。
   - 監査結果に UPDATE_DEVICE_CERTIFICATE 緩和アクションを適用して、この変更を行います。
   - ADD_THINGS_TO_THING_GROUP 緩和アクションを適用して、アクションを実行できるグループにデバイスを追加します。
   - Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、PUBLISH_FINDINGS_TO_SNS 緩和アクションを適用します。

   詳細については、「緩和アクション」を参照してください。
4. 古い証明書をデバイスからデタッチします。(「DetachThingPrincipal」を参照してください)。

ログ記録が無効です

Amazon CloudWatch で AWS IoT ログが有効になっていません。V1 および V2 ログ記録の両方を確認します。

このチェックは、CLI および API で `LOGGING_DISABLED_CHECK` として表示されます。

重大度: 低

詳細

このチェックにより不適合が見つかった場合、次の理由コードが返されます。

- LOGGING_DISABLED

重要な理由

CloudWatch の AWS IoT ログでは、認証の失敗、予期しない接続や切断など (デバイスが侵害されていることを示している可能性があります)、AWS IoT 内での動作がわかります。

修正方法

CloudWatch で AWS IoT ログを有効にします。「AWS IoT Core デベロッパーガイド」の「モニタリングとログ記録」を参照してください。緩和アクションを使用して、以下を行うこともできます。

- 監査結果に `ENABLE_IOT_LOGGING` 緩和アクションを適用して、この変更を行います。
- Amazon SNS メッセージに対する応答としてカスタムレスポンスを実装する場合は、`PUBLISH_FINDINGS_TO_SNS` 緩和アクションを適用します。

詳細については、「緩和アクション」を参照してください。

監査コマンド

監査設定の管理

アカウントの監査設定を行うには、`UpdateAccountAuditConfiguration` を使用します。このコマンドを使用すると、監査で使用するこれらのチェック項目を有効にする、オプションの通知を設定する、アクセス許可を設定することができます。

`DescribeAccountAuditConfiguration` でこれらの設定を確認します。

監査設定を削除するには、DeleteAccountAuditConfiguration を使用します。これにより、すべてのデフォルト値が復元されます。すべてのチェック項目はデフォルトで無効になっているため、監査が事実上無効になります。

UpdateAccountAuditConfiguration

このアカウントの Device Defender 監査設定を設定または再設定します。設定には、監査通知が送信される方法と有効または無効な監査チェック項目が含まれています。

概要

```
aws iot  update-account-audit-configuration \
    [--role-arn <value>] \
    [--audit-notification-target-configurations <value>] \
    [--audit-check-configurations <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json 形式

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": {
    "string": {
      "targetArn": "string",
      "roleArn": "string",
      "enabled": "boolean"
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

cli-input-json フィールド

名前	型	説明
roleArn	string	監査を実行するときに、デバイス、ポリシー、証明書、

名前	型	説明
	長さ - 最大:2048 最小:20	他の項目に関する情報にアクセスするアクセス許可を AWS IoT に付与するロールの ARN。
auditNotificationTargetConfigurations	map	監査通知が送信されるターゲットに関する情報。
targetArn	string	監査通知が送信されるターゲット (SNS トピック) の ARN。
roleArn	string 長さ - 最大:2048 最小:20	ターゲットに通知を送信するアクセス許可を付与するロールの ARN。
有効	boolean	ターゲットへの通知が有効な場合は true。

名前	型	説明
auditCheckConfigurations	map	このアカウントで有効および無効になっている監査チェック項目が指定されます。現在有効になっているチェック項目を含むすべてのチェック項目のリストを表示するには、DescribeAccountAuditConfiguration を使用します。 特定のチェック項目を有効にすると、一部のデータ収集がすぐに開始されます。チェックが無効になると、そのチェック項目に関連してこれまでに収集されたデータがすべて削除されます。 スケジュールによる監査により使用されている場合、チェック項目を無効にすることはできません。まず、スケジュールによる監査からチェック項目を削除するか、スケジュールによる監査自体を削除する必要があります。 UpdateAccountAuditConfiguration の最初の呼び出しでは、このパラメータは必須であり、少なくとも 1 つの有効なチェック項目を指定する必要があります。

名前	型	説明
有効	boolean	このアカウントでこの監査チェック項目が有効になっている場合は true。
設定	マップ	(オプション) CERT_AGE_THRESHOLD_IN_DAYS や CERT_EXPIRATION_THRESHOLD_IN_DAYS など、特定の監査チェックのカスタム設定。証明書の経過時間と有効期限についてアラートを受け取るタイミングを定義できます。

Output

なし

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

DescribeAccountAuditConfiguration

このアカウントの Device Defender 監査設定の情報を取得します。設定には、監査通知が送信される方法と有効または無効な監査チェック項目が含まれています。

概要

```
aws iot  describe-account-audit-configuration  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json 形式

```
{
}
```

出力

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": {
    "string": {
      "targetArn": "string",
      "roleArn": "string",
      "enabled": "boolean"
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

CLI 出力フィールド

名前	型	説明
roleArn	string 長さ - 最大:2048 最小:20	監査を実行するときに、デバイス、ポリシー、証明書、他の項目に関する情報にアクセスするアクセス許可を AWS IoT に付与するロールの ARN。 UpdateAccountAudit Configuration の最初の

名前	型	説明
		呼び出しでは、このパラメータは必須です。
auditNotificationTargetConfigurations	map	監査通知が送信されるこのアカウントのターゲットに関する情報。
targetArn	string	監査通知が送信されるターゲット (SNS トピック) の ARN。
roleArn	string 長さ - 最大:2048 最小:20	ターゲットに通知を送信するアクセス許可を付与するロールの ARN。
有効	boolean	ターゲットへの通知が有効な場合は true。
auditCheckConfigurations	map	このアカウントで有効および無効になっている監査チェック項目。
有効	boolean	このアカウントでこの監査チェック項目が有効になっている場合は true。
設定	マップ	(オプション) 特定の監査チェックに特定の設定を提供します。(例: 証明書の最大許容経過時間やアラートをトリガーする有効期限までの日数など)

エラー

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

DeleteAccountAuditConfiguration

このアカウントの Device Defender の監査のデフォルト設定を復元します。入力した設定データがすべて削除され、すべての監査チェック項目が無効にリセットされます。

概要

```
aws iot  delete-account-audit-configuration \
    [--delete-scheduled-audits | --no-delete-scheduled-audits]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

`cli-input-json` 形式

```
{
  "deleteScheduledAudits": "boolean"
}
```

`cli-input-json` フィールド

名前	型	説明
deleteScheduledAudits	boolean	true の場合、スケジュールによる監査がすべて削除されます。

出力

なし

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ResourceNotFoundException`

指定されたリソースは存在しません

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

監査のスケジュール

1 つ以上のスケジュールによる監査を作成するには、`CreateScheduledAudit` を使用します。このコマンドを使用すると、監査中に実行するチェック項目と監査の実行頻度を指定できます。

`ListScheduledAudits` と `DescribeScheduledAudit` を使用してスケジュールによる監査を追跡します。

`UpdateScheduledAudit` を使用して既存のスケジュールによる監査を変更するか、`DeleteScheduledAudit` を使用して削除します。

CreateScheduledAudit

指定された間隔で実行される、スケジュールによる監査を作成します。

概要

```
aws iot  create-scheduled-audit \
    --frequency <value> \
    [--day-of-month <value>] \
    [--day-of-week <value>] \
    --target-check-names <value> \
    [--tags <value>] \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

`cli-input-json` 形式

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "scheduledAuditName": "string"
}
```

cli-input-json フィールド

名前	型	説明
frequency	string	スケジュールによる監査が行われる頻度。「DAILY」、「WEEKLY」、「BIWEEKLY」、「MONTHLY」のいずれかを選択できます。各監査の実際の開始時刻は、システムによって決定されます。 列挙値: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	string pattern: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	スケジュールによる監査が行われる毎月の日付。「1」から「31」、または「LAST」を選択できます。このフィールドは、`frequency` パラメータが「MONTHLY」に設定されている場合に必要です。29 ~ 31 日が指定されていて、その月にその日付がない場合、監

名前	型	説明
		査はその月の末日 (LAST) に行われます。
dayOfWeek	string	スケジュールによる監査が行われる曜日。「SUN」、「MON」、「TUE」、「WED」、「THU」、「FRI」、「SAT」のいずれかに設定できます。このフィールドは、`frequency` パラメータが「WEEKLY」または「BIWEEKLY」に設定されている場合に必要です。 列挙値: SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	リスト メンバー: AuditCheckName	スケジュールによる監査中に実行されるチェック項目。アカウントでチェック項目が有効になっている必要があります。(有効なチェック項目を含むすべてのチェック項目のリストを参照するには `DescribeAccountAuditConfiguration` 、有効になっているチェック項目を選択するには `UpdateAccountAuditConfiguration` を使用します)。
タグ	リスト メンバー: Tag java class: java.util.List	スケジュールによる監査を管理するために使用できるメタデータ。

名前	型	説明
キー	string	タグのキー。
値	string	タグの値。
scheduledAuditName	string 長さ - 最大:128 最小:1 pattern: [a-zA-Z0-9_-]+	スケジュールによる監査に割り当てる名前。(最大 128 文字)

出力

```
{
  "scheduledAuditArn": "string"
}
```

CLI 出力フィールド

名前	型	説明
scheduledAuditArn	string	スケジュールによる監査の ARN。

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

`LimitExceededException`

制限を超過しました。

ListScheduledAudits

スケジュールによる監査をすべて一覧表示します。

概要

```
aws iot  list-scheduled-audits \
    [--next-token <value>] \
    [--max-results <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json 形式

```
{
  "nextToken": "string",
  "maxResults": "integer"
}
```

cli-input-json フィールド

名前	型	説明
nextToken	string	次の結果セットのトークン。
maxResults	integer 範囲 - 最大: 250 最小: 1	一度に返す結果の最大数。デフォルトは 25 です。

出力

```
{
  "scheduledAudits": [
    {
      "scheduledAuditName": "string",
      "scheduledAuditArn": "string",
      "frequency": "string",
      "dayOfMonth": "string",
      "dayOfWeek": "string"
    }
  ],
  "nextToken": "string"
```

```
}
```

CLI 出力フィールド

名前	型	説明
scheduledAudits	リスト メンバー: Scheduled AuditMetadata java class: java.util.List	スケジュールによる監査のリスト。
scheduledAuditName	string 長さ - 最大:128 最小:1 pattern: [a-zA-Z0-9_-]+	スケジュールによる監査の名前。
scheduledAuditArn	string	スケジュールによる監査の ARN。
frequency	string	スケジュールによる監査が行われる頻度。 列挙値: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	string pattern: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	スケジュールによる監査が実行される月の日付 (`frequency` が「MONTHLY」の場合)。29 ~ 31 日が指定されていて、その月にその日付がない場合、監査はその月の末日 (LAST) に行われます。
dayOfWeek	string	スケジュールによる監査が実行される曜日 (`frequency` が「WEEKLY」または「BIWEEKLY」の場合)。

名前	型	説明
		列挙値: SUN | MON | TUE | WED | THU | FRI | SAT
nextToken	string	次の結果セットの取得に使用できるトークン、または他に結果がない場合は null です。

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

DescribeScheduledAudit

スケジュールによる監査に関する情報を取得します。

概要

```
aws iot  describe-scheduled-audit \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json 形式

```
{
  "scheduledAuditName": "string"
}
```

cli-input-json フィールド

名前	型	説明
scheduledAuditName	string 長さ - 最大:128 最小:1 pattern: [a-zA-Z0-9_-]+	情報を取得するスケジュールによる監査の名前。

出力

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "scheduledAuditName": "string",
  "scheduledAuditArn": "string"
}
```

CLI 出力フィールド

名前	型	説明
frequency	string	スケジュールによる監査が行われる頻度。「DAILY」、「WEEKLY」、「BIWEEKLY」、「MONTHLY」のいずれか。各監査の実際の開始時刻は、システムによって決定されます。 列挙値: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	string	スケジュールによる監査が行われる毎月の日付。「1」から

名前	型	説明
	pattern: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	「31」、または「LAST」を選択できます。29 ~ 31 日が指定されていて、その月にその日付がない場合、監査はその月の末日 (LAST) に行われます。
dayOfWeek	string	スケジュールによる監査が行われる曜日。「SUN」、「MON」、「TUE」、「WED」、「THU」、「FRI」、「SAT」のいずれか。 列挙値: SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	リスト メンバー: AuditCheckName	スケジュールによる監査中に実行されるチェック項目。アカウントでチェック項目が有効になっている必要があります。(有効なチェック項目を含むすべてのチェック項目のリストを参照するには `DescribeAccountAuditConfiguration` 、有効になっているチェック項目を選択するには `UpdateAccountAuditConfiguration` を使用します)。
scheduledAuditName	string 長さ - 最大:128 最小:1 pattern: [a-zA-Z0-9_-]+	スケジュールによる監査の名前。

名前	型	説明
scheduledAuditArn	string	スケジュールによる監査の ARN。

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ResourceNotFoundException`

指定されたリソースは存在しません

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

UpdateScheduledAudit

実行されるチェック項目や監査が実行される頻度など、スケジュールによる監査が更新されます。

概要

```
aws iot  update-scheduled-audit \
    [--frequency <value>] \
    [--day-of-month <value>] \
    [--day-of-week <value>] \
    [--target-check-names <value>] \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

`cli-input-json` 形式

```
{
  "frequency": "string",
  "dayOfMonth": "string",
```

```
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "scheduledAuditName": "string"
}
```

cli-input-json フィールド

名前	型	説明
frequency	string	スケジュールによる監査が行われる頻度。「DAILY」、「WEEKLY」、「BIWEEKLY」、「MONTHLY」のいずれかを選択できます。各監査の実際の開始時刻は、システムによって決定されます。 列挙値: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	string pattern: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	スケジュールによる監査が行われる毎月の日付。「1」から「31」、または「LAST」を選択できます。このフィールドは、`frequency` パラメータが「MONTHLY」に設定されている場合に必要です。29 ~ 31 日が指定されていて、その月にその日付がない場合、監査はその月の末日 (LAST) に行われます。
dayOfWeek	string	スケジュールによる監査が行われる曜日。「SUN」、「MON」、「TUE」、「WED」、「THU」、「FRI」、「SAT」のいずれか

名前	型	説明
		を選択できます。このフィールドは、`frequency` パラメータが「WEEKLY」または「BIWEEKLY」に設定されている場合に必要です。 列挙値: SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	リスト メンバー: AuditCheckName	スケジュールによる監査中に実行されるチェック項目。アカウントでチェック項目が有効になっている必要があります。(有効なチェック項目を含むすべてのチェック項目のリストを参照するには `DescribeAccountAuditConfiguration` 、有効になっているチェック項目を選択するには `UpdateAccountAuditConfiguration` を使用します)。
scheduledAuditName	string 長さ - 最大:128 最小:1 pattern: [a-zA-Z0-9_-]+	スケジュールによる監査の名前。(最大 128 文字)

出力

```
{
  "scheduledAuditArn": "string"
}
```

CLI 出力フィールド

名前	型	説明
scheduledAuditArn	string	スケジュールによる監査の ARN。

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ResourceNotFoundException`

指定されたリソースは存在しません

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

DeleteScheduledAudit

スケジュールによる監査を削除します。

概要

```
aws iot  delete-scheduled-audit \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

`cli-input-json` 形式

```
{
  "scheduledAuditName": "string"
}
```

cli-input-json フィールド

名前	型	説明
scheduledAuditName	string 長さ - 最大:128 最小:1 pattern: [a-zA-Z0-9_-]+	削除するスケジュールによる監査の名前。

出力

なし

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ResourceNotFoundException`

指定されたリソースは存在しません

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

オンデマンド監査の実行

実行するチェック項目を指定し、監査の実行をすぐに開始するには、`StartOnDemandAuditTask` を使用します。

StartOnDemandAuditTask

オンデマンドの Device Defender 監査を開始します。

概要

```
aws iot  start-on-demand-audit-task \
    --target-check-names <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json 形式

```
{
  "targetCheckNames": [
    "string"
  ]
}
```

cli-input-json フィールド

名前	型	説明
targetCheckNames	リスト メンバー: AuditCheckName	監査中に実行されるチェック項目。指定するチェック項目は、アカウントで有効になっている必要があります。有効になっていないと、例外が発生します。(有効なチェック項目を含むすべてのチェック項目のリストを参照するには DescribeAccountAuditConfiguration 、有効になっているチェック項目を選択するには UpdateAccountAuditConfiguration を使用します)。

出力

```
{
  "taskId": "string"
}
```

CLI 出力フィールド

名前	型	説明
taskId	string 長さ - 最大:40 最小:1 pattern: [a-zA-Z0-9-]+	開始したオンデマンド監査の ID。

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

`LimitExceededException`

制限を超過しました。

監査インスタンスの管理

特定の監査インスタンスに関する情報を取得するには、`DescribeAuditTask` を使用します。すでに実行されている場合、結果には失敗したチェック項目と成功したチェック項目、システムが完了できなかったチェック項目、監査がまだ進行中の場合は処理中のチェック項目が含まれます。

特定の期間内に実行された監査を調べるには、`ListAuditTasks` を使用します。

進行中の監査を停止するには、`CancelAuditTask` を使用します。

DescribeAuditTask

Device Defender の監査に関する情報を取得します。

概要

```
aws iot  describe-audit-task \
    --task-id <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json 形式

```
{
  "taskId": "string"
}
```

cli-input-json フィールド

名前	型	説明
taskId	string 長さ - 最大:40 最小:1 pattern: [a-zA-Z0-9-]+	情報を取得する監査の ID。

出力

```
{
  "taskStatus": "string",
  "taskType": "string",
  "taskStartTime": "timestamp",
  "taskStatistics": {
    "totalChecks": "integer",
    "inProgressChecks": "integer",
    "waitingForDataCollectionChecks": "integer",
    "compliantChecks": "integer",
    "nonCompliantChecks": "integer",
    "failedChecks": "integer",
    "canceledChecks": "integer"
  },
  "scheduledAuditName": "string",
  "auditDetails": {
    "string": {
```

```
      "checkRunStatus": "string",
      "checkCompliant": "boolean",
      "totalResourcesCount": "long",
      "nonCompliantResourcesCount": "long",
      "errorCode": "string",
      "message": "string"
    }
  }
}
```

CLI 出力フィールド

名前	型	説明
taskStatus	string	監査のステータス: 「IN_PROGRESS」、「COMPLETED」、「FAILED」、「CANCELED」のいずれか。 列挙値: IN_PROGRESS | COMPLETED | FAILED | CANCELED
taskType	string	監査のタイプ: 「ON_DEMAND_AUDIT_TASK」または「SCHEDULED_AUDIT_TASK」。 列挙値: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStartTime	timestamp	監査の開始時刻。
taskStatistics	TaskStatistics	監査に関する統計情報。
totalChecks	integer	この監査のチェック項目の数。
inProgressChecks	integer	進行中のチェック項目の数。

名前	型	説明
waitingForDataCollectionChecks	integer	データ収集を待機しているチェック項目の数。
compliantChecks	integer	適合しているリソースが見つかったチェック項目の数。
nonCompliantChecks	integer	不適合リソースが見つかったチェック項目の数。
failedChecks	integer	チェック項目の数。
canceledChecks	integer	監査がキャンセルされたために実行されなかったチェック項目の数。
scheduledAuditName	string 長さ - 最大:128 最小:1 pattern: [a-zA-Z0-9_-]+	スケジュールによる監査の名前 (監査がスケジュールによる監査の場合のみ)。
auditDetails	map	この監査中に実行された各チェック項目に関する詳細情報。

名前	型	説明
checkRunStatus	string	このチェック項目の完了ステータス:「IN_PROGRESS」、「WAITING_FOR_DATA_COLLECTION」、「CANCELED」、「COMPLETED_COMPLIANT」、「COMPLETED_NON_COMPLIANT」、「FAILED」のいずれか。 列挙値: IN_PROGRESS | WAITING_FOR_DATA_COLLECTION | CANCELED | COMPLETED_COMPLIANT | COMPLETED_NON_COMPLIANT | FAILED
checkCompliant	boolean	チェックが完了し、すべてのリソースが適合していることが分かった場合は true。
totalResourcesCount	long	チェック項目が実行されたリソースの数。
nonCompliantResourcesCount	long	チェックで不適合が検出されたリソースの数。
errorCode	string	この監査中にこのチェック項目を実行する際に発生したエラーのコード。「INSUFFICIENT_PERMISSIONS」または「AUDIT_CHECK_DISABLED」。

名前	型	説明
メッセージ	string 長さ - 最大:2048	この監査中にこのチェック項目を実行する際に発生したエラーに関連付けられたメッセージ。

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ResourceNotFoundException`

指定されたリソースは存在しません

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

ListAuditTasks

指定された期間中に実行された Device Defender の監査を一覧表示します。

概要

```
aws iot  list-audit-tasks \
    --start-time <value> \
    --end-time <value> \
    [--task-type <value>] \
    [--task-status <value>] \
    [--next-token <value>] \
    [--max-results <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

`cli-input-json` 形式

```
{
  "startTime": "timestamp",
  "endTime": "timestamp",
  "taskType": "string",
  "taskStatus": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

cli-input-json フィールド

名前	型	説明
startTime	timestamp	期間の開始。監査情報は限られた期間内 (180 日) のみ保持されます。保持される内容より前に開始時間をリクエストすると、InvalidRequestException が生成されます。
endTime	timestamp	期間の終了。
taskType	string	指定されたタイプの監査に出力を制限するフィルタ: 「ON_DEMAND_AUDIT_TASK」または「SCHEDULED__AUDIT_TASK」を選択ことができます。 列挙値: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStatus	string	指定された完了ステータスを持つ監査に出力を制限するフィルタ: 「IN_PROGRESS」、「COMPLETED」、「FAILED」、「

名前	型	説明
		CANCELED」のいずれかを選択することができます。 列挙値: IN_PROGRESS | COMPLETED | FAILED | CANCELED
nextToken	string	次の結果セットのトークン。
maxResults	integer 範囲 - 最大: 250 最小: 1	一度に返す結果の最大数。デフォルトは 25 です。

出力

```
{
  "tasks": [
    {
      "taskId": "string",
      "taskStatus": "string",
      "taskType": "string"
    }
  ],
  "nextToken": "string"
}
```

CLI 出力フィールド

名前	型	説明
tasks	リスト メンバー: AuditTaskMetadata java class: java.util.List	指定された期間中に実行された監査。
taskId	string 長さ - 最大:40 最小:1	この監査の ID。

名前	型	説明
	pattern: [a-zA-Z0-9-]+	
taskStatus	string	この監査のステータス: 「IN_PROGRESS」、「COMPLETED」、「FAILED」、「CANCELED」のいずれか。 列挙値: IN_PROGRESS | COMPLETED | FAILED | CANCELED
taskType	string	この監査のタイプ: 「ON_DEMAND_AUDIT_TASK」または「SCHEDULED_AUDIT_TASK」。 列挙値: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
nextToken	string	次の結果セットの取得に使用できるトークン、または追加の結果がない場合は `null` です。

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

CancelAuditTask

進行中の監査をキャンセルします。監査は、スケジュールすることもオンデマンドにすることもできます。監査が進行中でない場合は、InvalidRequestException が発生します。

概要

```
aws iot  cancel-audit-task \
    --task-id <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

cli-input-json 形式

```
{
  "taskId": "string"
}
```

cli-input-json フィールド

名前	型	説明
taskId	string 長さ - 最大:40 最小:1 pattern: [a-zA-Z0-9-]+	キャンセルする監査の ID です。「IN_PROGRESS」の監査のみキャンセルすることができます。

出力

なし

エラー

ResourceNotFoundException

指定されたリソースは存在しません

InvalidRequestException

リクエストの内容が無効です。

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

監査結果のチェック

監査の結果を表示するには、`ListAuditFindings` を使用します。チェック項目の種類、特定のリソース、または監査の時間によって結果をフィルタリングすることができます。この情報を使用して、検知された問題を軽減することができます。

緩和アクションを定義し、監査の結果に適用できます。詳細については、「緩和アクション」を参照してください。

ListAuditFindings

Device Defender の監査または指定された期間に実行された監査の結果を一覧表示します。(結果は 180 日間保持されます)。

概要

```
aws iot  list-audit-findings \
    [--task-id <value>] \
    [--check-name <value>] \
    [--resource-identifier <value>] \
    [--max-results <value>] \
    [--next-token <value>] \
    [--start-time <value>] \
    [--end-time <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

`cli-input-json` 形式

```
{
  "taskId": "string",
  "checkName": "string",
  "resourceIdentifier": {
    "deviceCertificateId": "string",
    "caCertificateId": "string",
```

```
    "cognitoIdentityPoolId": "string",
    "clientId": "string",
    "policyVersionIdentifier": {
      "policyName": "string",
      "policyVersionId": "string"
    },

    "roleAliasArn": "string",
    "account": "string"
  },
  "maxResults": "integer",
  "nextToken": "string",
  "startTime": "timestamp",
  "endTime": "timestamp"
}
```

cli-input-json フィールド

名前	型	説明
taskId	string 長さ - 最大:40 最小:1 pattern: [a-zA-Z0-9-]+	指定された ID を持つ監査に結果を制限するフィルタ。taskId または startTime と endTime のどちらかを指定する必要がありますが、両方を指定することはできません。
checkName	string	指定された監査チェック項目の結果に結果を制限するフィルタ。
resourceIdentifier	ResourceIdentifier	不適合リソースを識別する情報。
deviceCertificateId	string 長さ - 最大:64 最小:64 pattern: (0x)?[a-fA-F0-9]+	リソースにアタッチされた証明書の ID。
caCertificateId	string	証明書を承認するために使用する CA 証明書の ID。

名前	型	説明
	長さ - 最大:64 最小:64 pattern: (0x)?[a-fA-F0-9]+	
cognitoIdentityPoolId	string	Amazon Cognito ID プールの ID。
clientId	string	クライアント ID。
policyVersionIdentifier	PolicyVersionIdentifier	リソースに関連付けられているポリシーのバージョン。
policyName	string 長さ - 最大:128 最小:1 pattern: [w+=,.@-]+	ポリシーの名前。
policyVersionId	string pattern: [0-9]+	リソースに関連付けられているポリシーのバージョンの ID。
roleAliasArn	string	過度に許容されたアクションを持つロールエイリアスの ARN。 長さ - 最大: 2048 最小: 1
account	string 長さ - 最大:12 最小:12 pattern: [0-9]+	リソースが関連付けられているアカウント。
maxResults	integer 範囲 - 最大: 250 最小: 1	一度に返す結果の最大数。デフォルトは 25 です。
nextToken	string	次の結果セットのトークン。

名前	型	説明
startTime	timestamp	指定された時刻後に見つかった結果に結果を制限するフィルタ。startTime と endTime または taskId のどちらかを指定する必要がありますが、両方を指定することはできません。
endTime	timestamp	指定された時刻前に見つかった結果に結果を制限するフィルタ。startTime と endTime または taskId のどちらかを指定する必要がありますが、両方を指定することはできません。

出力

```
{
  "findings": [
    {
      "taskId": "string",
      "checkName": "string",
      "taskStartTime": "timestamp",
      "findingTime": "timestamp",
      "severity": "string",
      "nonCompliantResource": {
        "resourceType": "string",
        "resourceIdentifier": {
          "deviceCertificateId": "string",
          "caCertificateId": "string",
          "cognitoIdentityPoolId": "string",
          "clientId": "string",
          "policyVersionIdentifier": {
            "policyName": "string",
            "policyVersionId": "string"
          },
```

```
          "account": "string"
        },
        "additionalInfo": {
          "string": "string"
        }
      },
      "relatedResources": [
        {
          "resourceType": "string",
          "resourceIdentifier": {
            "deviceCertificateId": "string",
            "caCertificateId": "string",
            "cognitoIdentityPoolId": "string",
            "clientId": "string",

            "iamRoleArn": "string",

            "policyVersionIdentifier": {
              "policyName": "string",
              "policyVersionId": "string"
            },
            "account": "string"
          },

          "roleAliasArn": "string",

          "additionalInfo": {
            "string": "string"
          }
        }
      ],
      "reasonForNonCompliance": "string",
      "reasonForNonComplianceCode": "string"
    }
  ],
  "nextToken": "string"
}
```

CLI 出力フィールド

名前	型	説明
findings	リスト	監査の結果。

名前	型	説明
	メンバー: AuditFinding	
taskId	string 長さ - 最大:40 最小:1 pattern: [a-zA-Z0-9-]+	この結果 (所見) を生成した監査の ID。
checkName	string	この結果を生成した監査チェック項目。
taskStartTime	timestamp	監査の開始時刻。
findingTime	timestamp	結果が検出された時刻。
severity	string	結果の重要度。 列挙値: CRITICAL | HIGH | MEDIUM | LOW
nonCompliantResource	NonCompliantResource	監査チェックの結果、適合していないことが判明したリソース。
resourceType	string	不適合リソースのタイプ。 列挙値: DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	不適合リソースを識別する情報。

名前	型	説明
deviceCertificateId	string 長さ - 最大:64 最小:64 pattern: (0x)?[a-fA-F0-9]+	リソースにアタッチされた証明書の ID。
caCertificateId	string 長さ - 最大:64 最小:64 pattern: (0x)?[a-fA-F0-9]+	証明書を承認するために使用する CA 証明書の ID。
cognitoIdentityPoolId	string	Amazon Cognito ID プールの ID。
clientId	string	クライアント ID。
policyVersionIdentifier	PolicyVersionIdentifier	リソースに関連付けられているポリシーのバージョン。
policyName	string 長さ - 最大:128 最小:1 pattern: [w+=,.@-]+	ポリシーの名前。
policyVersionId	string pattern: [0-9]+	リソースに関連付けられているポリシーのバージョンの ID。
account	string 長さ - 最大:12 最小:12 pattern: [0-9]+	リソースが関連付けられているアカウント。
additionalInfo	map	不適合リソースに関するその他の情報。

名前	型	説明
relatedResources	リスト メンバー: RelatedResource	関連リソースのリスト。
resourceType	string	リソースのタイプ。 列挙値: DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	リソースを識別する情報。
deviceCertificateId	string 長さ - 最大:64 最小:64 pattern: (0x)?[a-fA-F0-9]+	リソースにアタッチされた証明書の ID。
caCertificateId	string 長さ - 最大:64 最小:64 pattern: (0x)?[a-fA-F0-9]+	証明書を承認するために使用する CA 証明書の ID。
cognitoIdentityPoolId	string	Amazon Cognito ID プールの ID。
clientId	string	クライアント ID。
policyVersionIdentifier	PolicyVersionIdentifier	リソースに関連付けられているポリシーのバージョン。
iamRoleArn	string 長さ - 最大:2048 最小:20	過度に許容されたアクションを持つ IAM ロールの ARN。

名前	型	説明
policyName	string 長さ - 最大:128 最小:1 pattern: [w+=,.@-]+	ポリシーの名前。
policyVersionId	string pattern: [0-9]+	リソースに関連付けられているポリシーのバージョンの ID。
roleAliasArn	string 長さ - 最大: 2048 最小: 1	過度に許容されたアクションを持つロールエイリアスの ARN。
account	string 長さ - 最大:12 最小:12 pattern: [0-9]+	リソースが関連付けられているアカウント。
additionalInfo	map	リソースに関するその他の情報。
reasonForNonCompliance	string	リソースが不適合だった理由。
reasonForNonCompliianceCode	string	リソースが不適合だった理由を示すコード。
nextToken	string	次の結果セットの取得に使用できるトークン、または追加の結果がない場合は `null` です。

エラー

`InvalidRequestException`

リクエストの内容が無効です。

`ThrottlingException`

レートが制限を超えています。

`InternalFailureException`

予期しないエラーが発生しました。

監査の所見の抑制

監査を実行すると、すべての不適合リソースの所見が報告されます。つまり、監査レポートには、問題の緩和に取り組んでいるリソースの所見と、テストやデバイスの破損など、不適合であることがわかっているリソースの所見が含まれます。監査では、後続の監査実行で不適合のままであるリソースに関する所見が引き続きレポートされるため、レポートに不要な情報が追加される可能性があります。監査所見を抑制すると、指定した期間、リソースが修正されるまでの期間、またはテストもしくは破損したデバイスに関連付けられたリソースに対して、無期限に所見を抑制または除外できます。

> **Note**
>
> 抑制された監査所見に対しては、緩和アクションは使用できません。緩和アクションの詳細については、緩和アクション を参照してください。

監査所見の抑制クォータの詳細については、「AWS IoT Device Defender のエンドポイントとクォータ」を参照してください。

監査所見の抑制の仕組み

不適合のリソース用に監査所見の抑制を作成すると、監査レポートと通知の動作が異なります。

監査レポートには、レポートに関連する、抑制されたすべての所見を一覧表示する新しいセクションが含まれます。監査チェックが適合であるかどうかを評価する場合、抑制された所見は考慮されません。コマンドラインインターフェイス (CLI) で describe-audit-task コマンドを使用すると、各監査チェックの抑制されたリソース数も返されます。

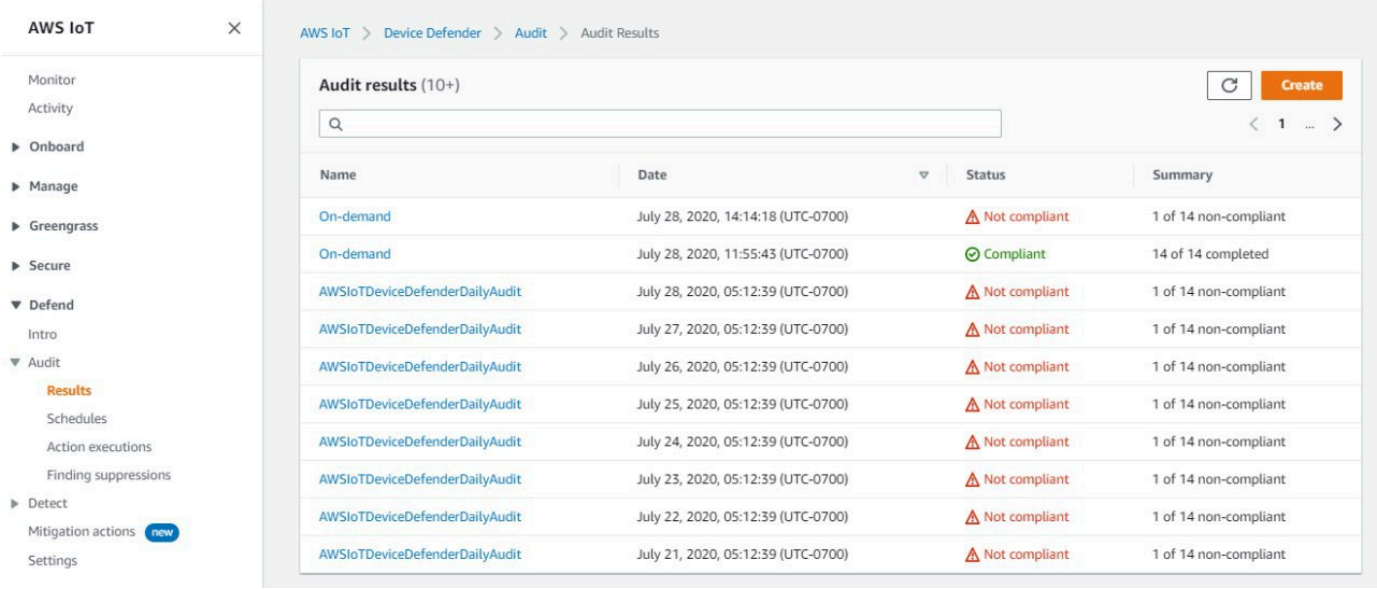
監査通知では、監査チェックが適合であるかどうかを評価する場合、抑制された所見は考慮されません。抑制されたリソース数は、AWS IoT Device Defender が Amazon CloudWatch および Amazon Simple Notification Service (Amazon SNS) に発行する、各監査チェック通知にも含まれます。

コンソールで監査所見の抑制を使用する方法

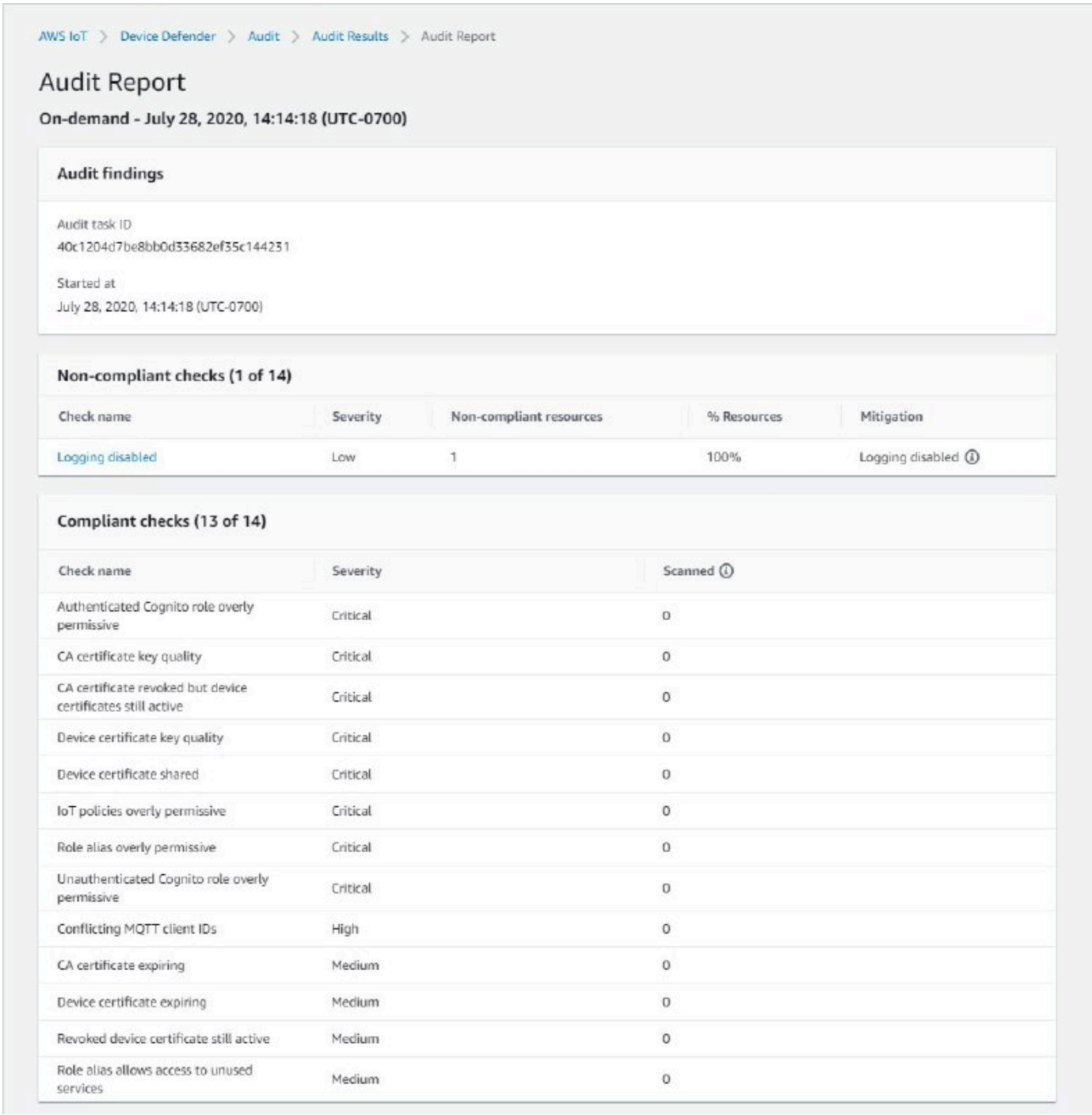
監査レポートからの所見を抑制するには

次の手順は、AWS IoT コンソールで監査所見の抑制を作成する方法を示しています。

1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Audit] (監査)、[Results] (結果) の順に選択します。
2. 確認する監査レポートを選択します。



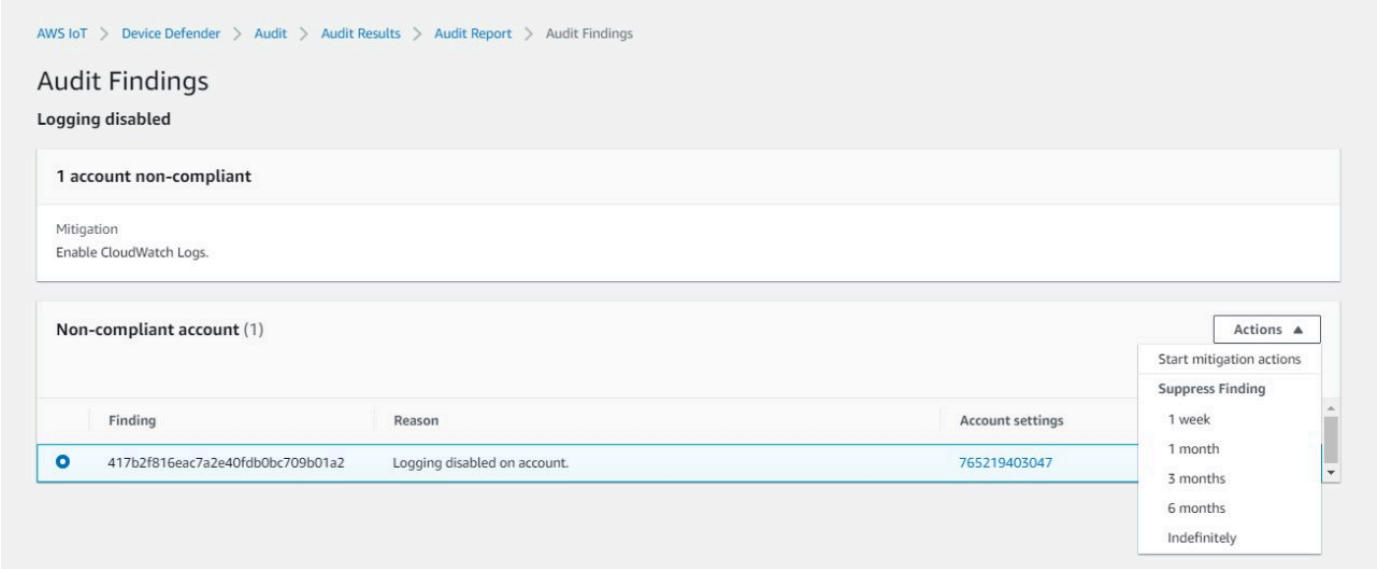
3. [非準拠のチェック] セクションの [チェック名] で、関心のある監査チェックを選択します。



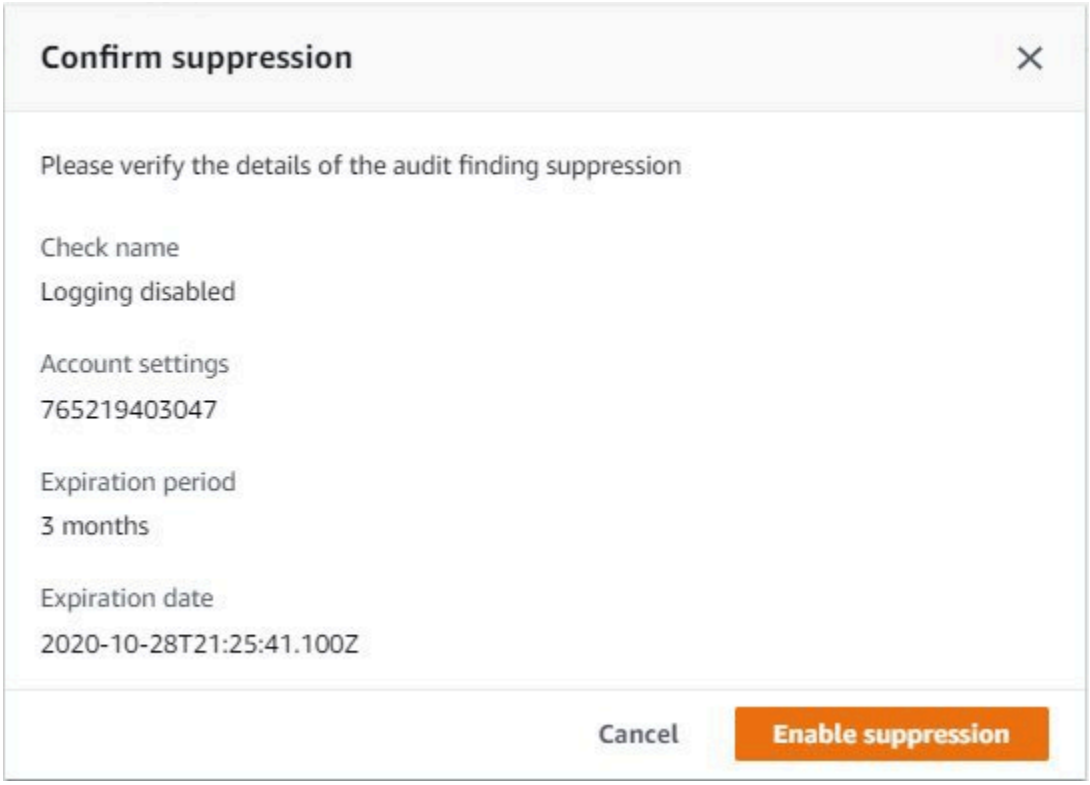
4. 監査チェックの詳細画面で、表示したくない所見がある場合は、所見の横にあるオプションボタンを選択します。次に、[アクション] を選択し、監査結果を継続する期間を選択します。

> **Note**
>
> コンソールでは、監査所見の抑制の有効期限として、[1 week] (1 週間)、[1 month] (1 か月)、[3 months] (3 か月)、[6 months] (6 か月)、または [Indefinitely] (無期限) を選択でき

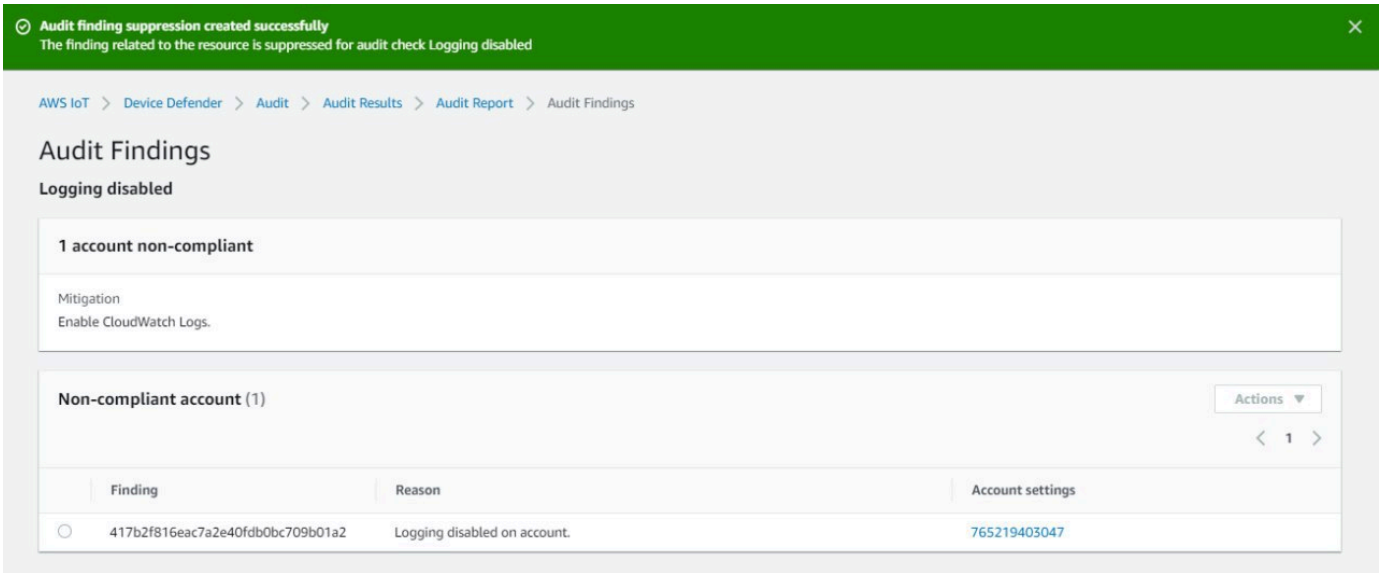
ます。特定の有効期限を設定する場合は、CLI または API でのみ設定できます。監査所見の抑制は、有効期限にかかわらず、いつでも取り消すこともできます。



5. 抑制の詳細を確認してから、[抑制を有効化] を選択します。

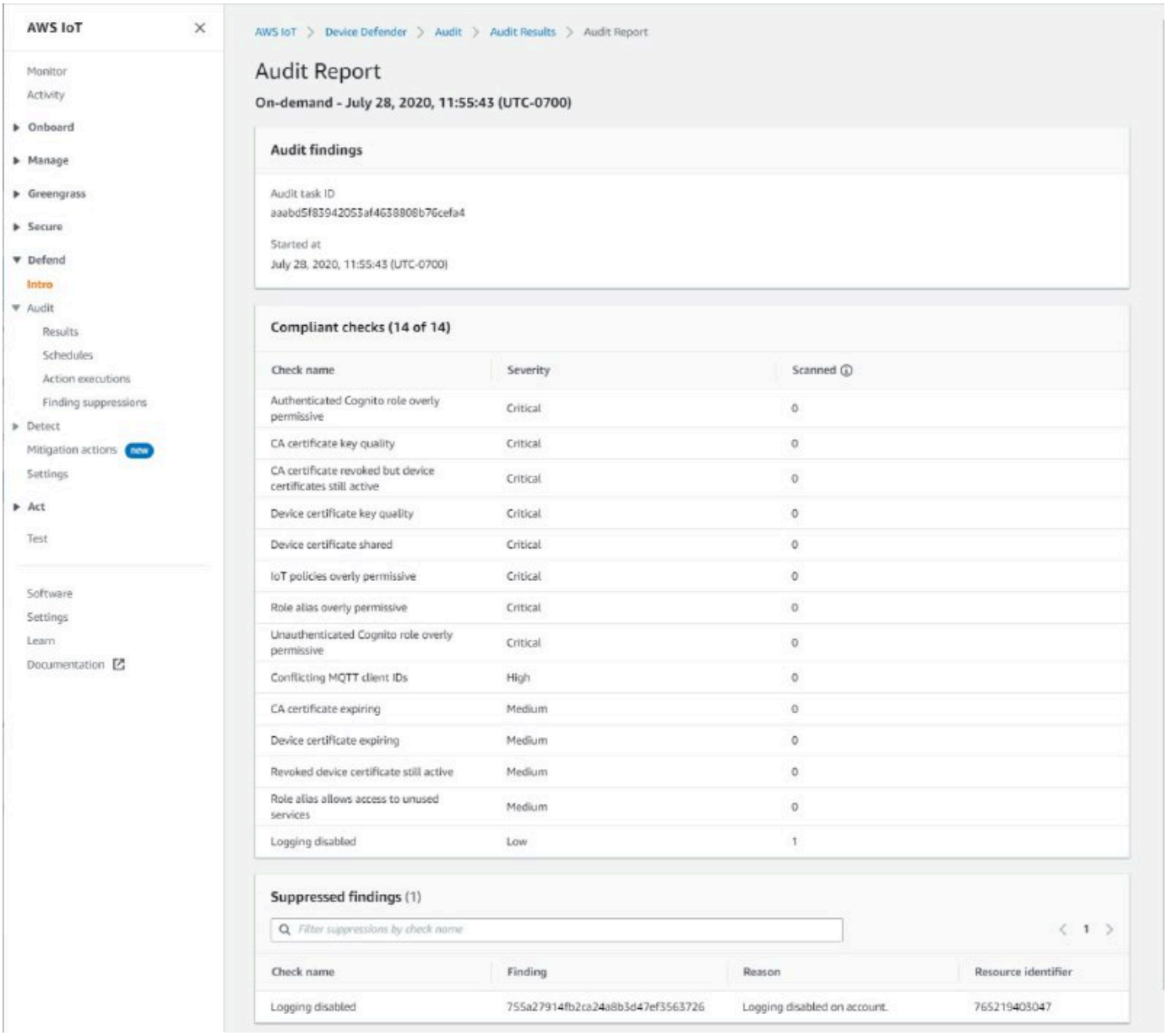


6. 監査所見の抑制を作成すると、監査所見の抑制が作成されたことを確認するバナーが表示されます。



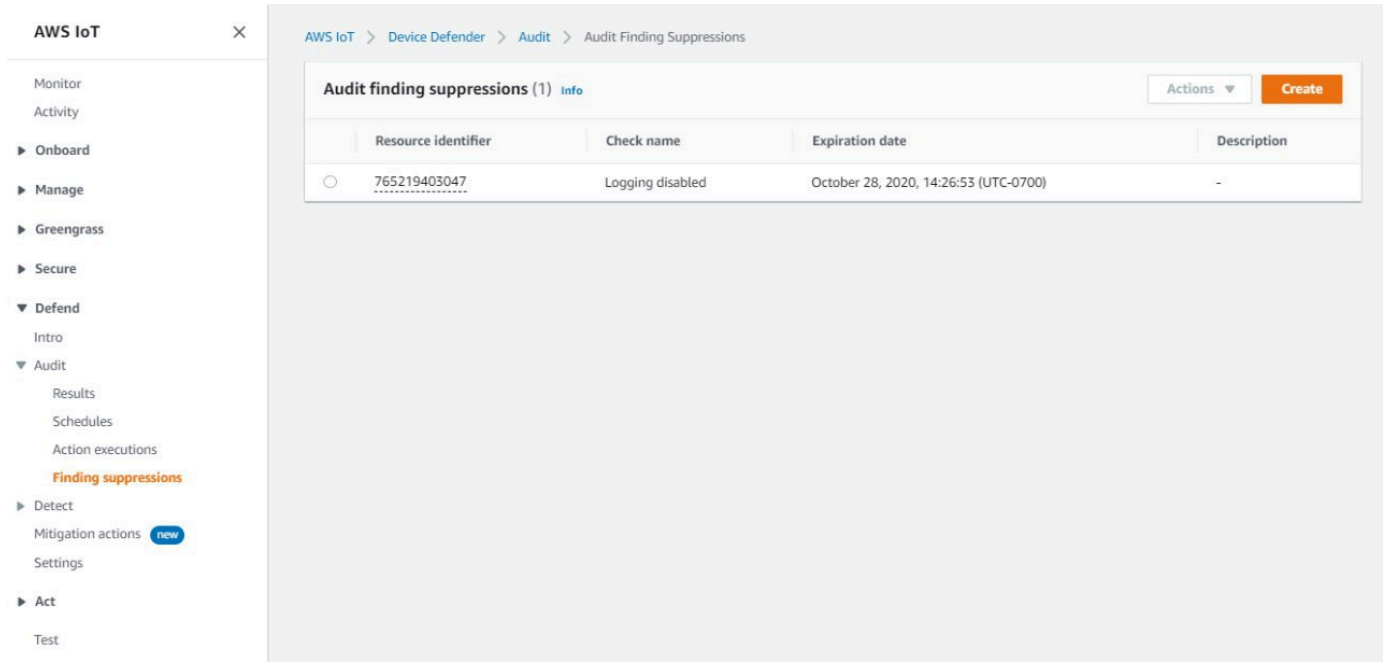
抑制された所見を監査レポートで表示するには

1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Audit] (監査)、[Results] (結果) の順に選択します。
2. 確認する監査レポートを選択します。
3. [抑制された結果] セクションで、選択した監査レポート用に抑制された監査結果を表示します。



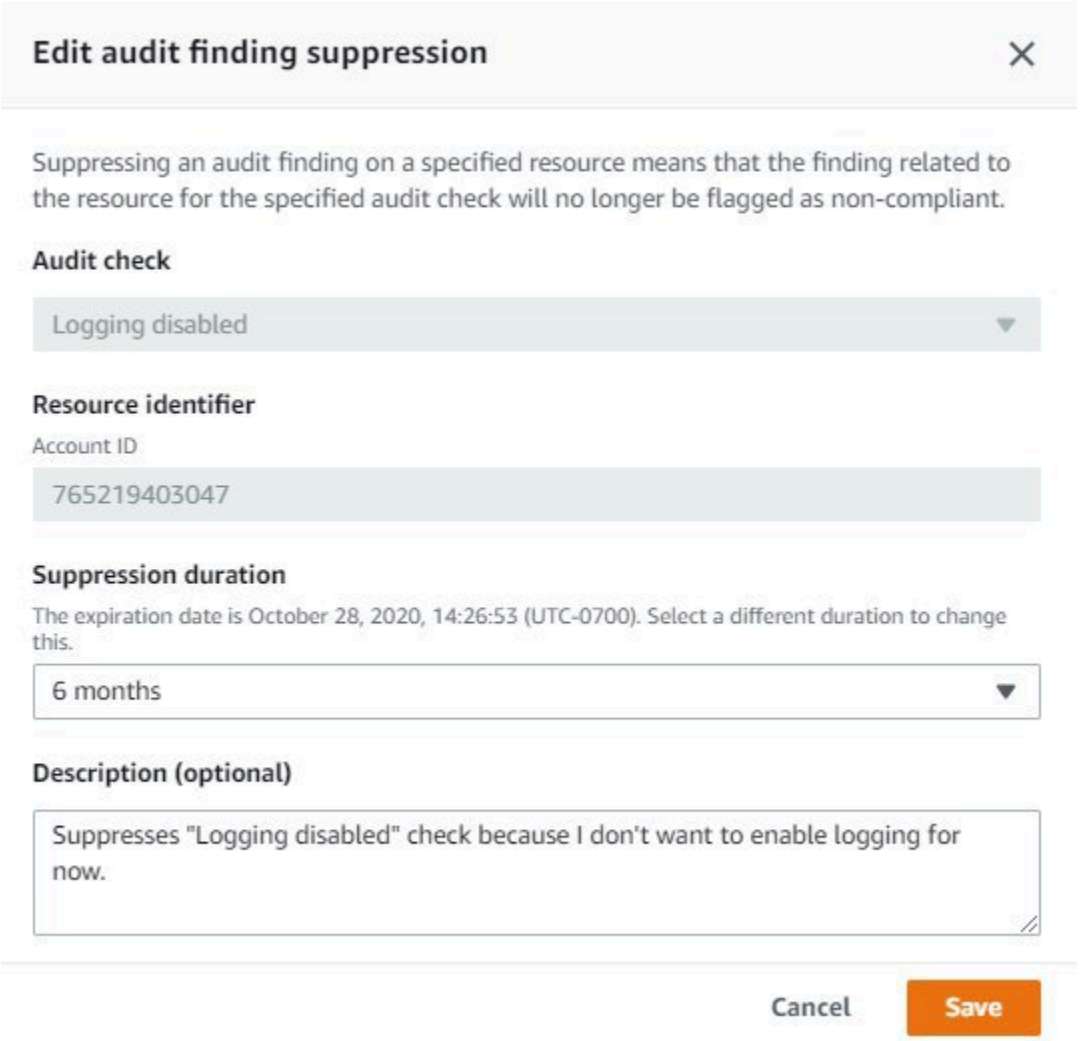
監査所見の抑制をリスト化するには

- AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Audit] (監査)、[Finding suppressions] (所見の抑制) の順に選択します。



監査所見の抑制を編集するには

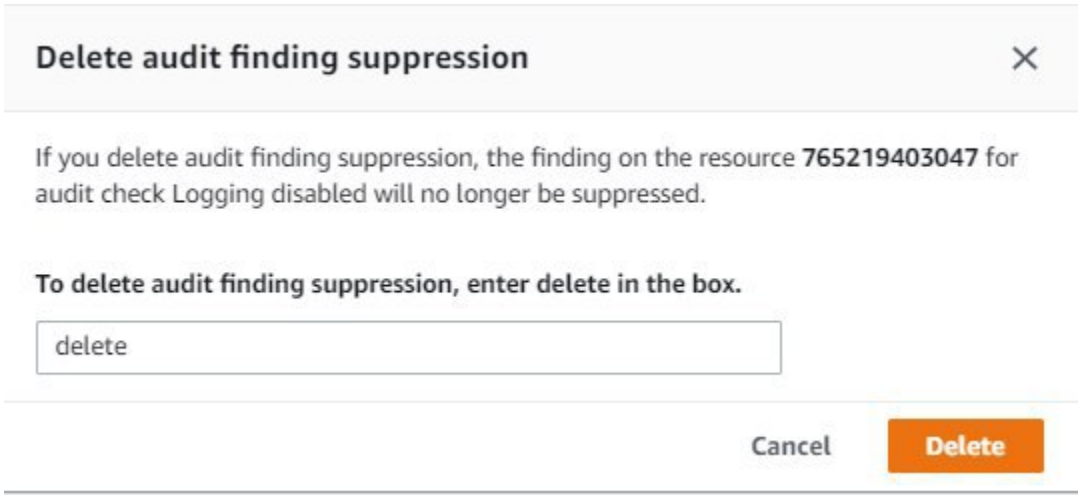
1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Audit] (監査)、[Finding suppressions] (所見の抑制) の順に選択します。
2. 編集する監査所見の抑制の横にあるオプションボタンを選択します。次に、[アクション]、[編集] の順に選択します。
3. [監査結果の抑制を編集する] ウィンドウで、[抑制期間] または [説明] (オプション) を変更できます。



4. 変更を行ったら、[保存] を選択します。[検索結果の抑制] ウィンドウが開きます。

監査所見の抑制を削除するには

1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Audit] (監査)、[Finding suppressions] (所見の抑制) の順に選択します。
2. 削除する監査所見の抑制の横にあるオプションボタンを選択し、[Actions] (アクション)、[Delete] (削除) の順に選択します。
3. [Delete audit finding suppression] (監査所見の抑制の削除) ウィンドウで、テキストボックスに `delete` と入力して削除を確認し、[Delete] (削除) を選択します。[検索結果の抑制] ウィンドウが開きます。



CLI での監査所見の抑制の使用方法

次の CLI コマンドを使用して、監査所見の抑制を作成および管理できます。

- create-audit-suppression
- describe-audit-suppression
- update-audit-suppression
- delete-audit-suppression
- list-audit-suppressions

入力する `resource-identifier` は、結果を抑制する `check-name` によって異なります。次の表では、各チェックが抑制を作成および編集するために必要な `resource-identifier` の詳細を示しています。

> **Note**
>
> 抑制コマンドは、監査をオフにするものではありません。監査は引き続き AWS IoT デバイスで実行されます。抑制は、監査所見にのみ適用されます。

check-name	resource-identifier
AUTHENTICATE_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	cognitoIdentityPoolId

check-name	resource-identifier
CA_CERT_APPROACHING_EXPIRATION_CHECK	caCertificateId
CA_CERTIFICATE_KEY_QUALITY_CHECK	caCertificateId
CONFLICTING_CLIENT_IDS_CHECK	clientId
DEVICE_CERT_APPROACHING_EXPIRATION_CHECK	deviceCertificateId
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK	deviceCertificateId
DEVICE_CERTIFICATE_SHARED_CHECK	deviceCertificateId
IOT_POLICY_OVERLY_PERMISSIVE_CHECK	policyVersionIdentifier
IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK	roleAliasArn
IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK	roleAliasArn
LOGGING_DISABLED_CHECK	account
REVOKED_CA_CERT_CHECK	caCertificateId
REVOKED_DEVICE_CERT_CHECK	deviceCertificateId
UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	cognitoIdentityPoolId

監査所見の抑制を作成して適用するには

次の手順は、AWS CLI コンソールで監査所見の抑制を作成する方法を示しています。

- `create-audit-suppression` コマンドを使用して、監査所見の抑制を作成します。次の例では、[*Logging disabled*] (ログ記録が無効化されました) のチェックに基づいて、AWS アカウント123456789012 の監査所見の抑制を作成します。

```
aws iot create-audit-suppression \
    --check-name LOGGING_DISABLED_CHECK \
    --resource-identifier account=123456789012 \
    --client-request-token 28ac32c3-384c-487a-a368-c7bbd481f554 \
    --suppress-indefinitely \
    --description "Suppresses logging disabled check because I don't want to enable
 logging for now."
```

このコマンドの出力はありません。

監査所見の抑制 API

次の API を使用して、監査所見の抑制を作成および管理できます。

- CreateAuditSuppression
- DescribeAuditSuppression
- UpdateAuditSuppression
- DeleteAuditSuppression
- ListAuditSuppressions

特定の監査の所見をフィルタリングするには、ListAuditFindings API を使用できます。

検出

AWS IoT Device Defender Detect を使用すると、デバイスの動作を監視することにより、侵害されたデバイスを示す可能性がある異常な動作を特定できます。クラウド側のメトリクス (AWS IoT から取得) とデバイス側のメトリクス (デバイスにインストールしたエージェントから取得) を組み合わせて使用すると、次のことを検出できます。

- 接続パターンの変更。
- 認証されていないエンドポイントまたは認識されていないエンドポイントと通信するデバイス。
- インバウンドおよびアウトバウンドのデバイストラフィックパターンの変更。

予期されるデバイス動作の定義が含まれるセキュリティプロファイルを作成し、フリート内のデバイスのグループまたはすべてのデバイスに割り当てます。AWS IoT Device DefenderDetect は、これらのセキュリティプロファイルを使用して異常を検出し、Amazon CloudWatch メトリクスと Amazon Simple Notification Service 通知を介してアラームを送信します。

AWS IoT Device Defender Detect は、接続されたデバイスで頻繁に見られるセキュリティ上の問題を検出できます。

- 潜在的な悪意のあるコマンドおよびコントロールチャネルを示す、既知の悪意のある IP アドレスまたは許可されていないエンドポイントへのデバイスからのトラフィック。
- デバイスが DDoS 攻撃に関与していることを示す、アウトバウンドトラフィックのスパイクなど、変則的なトラフィック。
- リモートでアクセスできるリモート管理インターフェイスとポートを持つデバイス。
- アカウントに送信されるメッセージ量のスパイク (たとえば、メッセージあたりの料金が過剰になる可能性のある非認証デバイスから)。

ユースケース:

攻撃領域を測定する

AWS IoT Device Defender Detect を使用すると、デバイスの攻撃領域を測定できます。たとえば、攻撃キャンペーンの対象となりやすいサービスポートを持つデバイスを特定できます (ポート 23/2323 で実行される telnet サービス、ポート 22 で実行される SSH サービス、ポート 80/443/8080/8081 で実行される HTTP/S サービス)。デバイスでこれらのサービスポートを使用する正当な理由がある場合もありますが、攻撃者やキャリア関連のリスクによって攻撃領域の一

部ともよくなります。AWS IoT Device Defender Detect によって攻撃領域のアラートが生成されたら、攻撃領域を最小限に抑えるか (未使用のネットワークサービスをなくすことにより)、追加の評価を実行してセキュリティの脆弱性を特定する (たとえば、よくあるパスワード、デフォルトのパスワード、弱いパスワードで構成された telnet など) ことができます。

セキュリティの根本原因を使用したデバイス動作の異常の検出

AWS IoT Device Defender Detect を使用すると、セキュリティ侵害を示す可能性がある予期しないデバイス動作メトリクス (開いているポートの数、接続の数、予期せず開いているポート、予期しない IP アドレスへの接続) についてのアラームを生成できます。たとえば、TCP 接続の数が予想より多くなった場合、デバイスが DDoS 攻撃に使用されていることを示している可能性があります。予期されるポート以外のポートでリッスンしているプロセスは、リモート制御のためにバックドアがデバイスにインストールされていることを示している可能性があります。AWS IoT Device Defender Detect を使用すると、デバイスフリートの正常性を調べ、セキュリティ上の前提を確認できます (たとえば、ポート 23 または 2323 でリッスンしているデバイスがないこと)。

機械学習 (ML) ベースの脅威検出を有効にして、潜在的な脅威を自動的に識別できます。

正しく設定されていないデバイスを検出する

デバイスからアカウントに送信されるメッセージの数またはサイズのスパイクは、正しく設定されていないデバイスを示している可能性があります。そのようなデバイスがあると、メッセージ単位の料金が上昇する可能性があります。同様に、認証エラーの多いデバイスには、再設定されたポリシーが必要になる可能性があります。

未登録のデバイスの動作をモニタリングする

AWS IoT Device Defender Detect では、AWS IoT レジストリに登録されていないデバイスの異常な動作を識別することができます。以下のいずれかのターゲットタイプに固有のセキュリティプロファイルを定義することができます。

- すべてのデバイス
- 登録済みのすべてのデバイス (AWS IoT レジストリ内のモノ)
- 未登録のすべてのデバイス
- モノグループ内のデバイス

セキュリティプロファイルは、アカウントの予期されるデバイスの動作を定義し、異常が検出されたときに実行するアクションを指定します。セキュリティプロファイルは、そのプロファイルに対して

評価するデバイスをきめ細かく制御できるように、最も合ったターゲットにアタッチする必要があります。

未登録のデバイスの場合は、すべての違反やメトリクスが同じデバイスとして扱われるように、デバイスの有効期間にわたって一貫した MQTT クライアント識別子またはモノの名前 (デバイスメトリクスをレポートするデバイス用) を指定する必要があります。

> **Important**
>
> モノ名に制御文字が含まれている場合、またはモノ名が 128 バイトの UTF-8 エンコード文字より長い場合、デバイスによって報告されたメッセージは拒否されます。

セキュリティのユースケース

このセクションでは、デバイスフリートを脅かすさまざまな種類の攻撃と、これらの攻撃を監視するために使用できる推奨メトリクスについて説明します。セキュリティの問題を調査する出発点としてメトリクスの異常を使用することをお勧めしますが、メトリクスの異常のみに基づいてセキュリティ上の脅威を決定しないでください。

異常アラームを調査するには、アラームの詳細をデバイス属性、デバイスメトリクスの履歴トレンド、Security Profile メトリクスの履歴トレンド、カスタムメトリクス、ログなどのコンテキスト情報と関連付けて、セキュリティの脅威が存在するかどうかを判断します。

クラウド側のユースケース

Device Defender は、AWS IoT クラウド側で以下のユースケースを監視できます。

[知的財産の盗難:]

知的財産の盗難には、企業秘密、ハードウェア、ソフトウェアなどの個人または企業の知的財産を盗むことが含まれます。デバイスの製造段階で発生することがよくあります。知的財産の盗難は、著作権侵害、デバイスの盗難、またはデバイス証明書の盗難などの形で発生することがあります。クラウドベースの知的財産の盗難は、IoT リソースへの意図しないアクセスを許可するポリシーが存在することによって発生する可能性があります。IoT ポリシーを確認し、[Audit overly permissive checks] (権限が過剰のチェックの監査) をオンにして、権限が過剰であるポリシーを特定する必要があります。

[関連メトリクス:]

メトリクス	根拠
送信元 IP	デバイスが盗まれた場合、その送信元 IP アドレスは、通常のサプライチェーンで流通しているデバイスの通常想定される IP アドレス範囲外になります。
受信したメッセージの数 メッセージサイズ	攻撃者はクラウドベースの IP 盗難でデバイスを使用する可能性があるため、AWS IoT クラウドからデバイスに送信されたメッセージ数またはメッセージサイズに関連するメトリクスが急増する場合があり、これはセキュリティ上の問題があることを示しています。

[MQTT ベースのデータ流出:]

データ流出は、悪意のある行為者が IoT のデプロイまたはデバイスから不正なデータ転送を実行したときに発生します。攻撃者は、クラウド側のデータソースに対して MQTT を通じてこの種の攻撃を開始します。

[関連メトリクス:]

メトリクス	根拠
送信元 IP	デバイスが盗まれた場合、その送信元 IP アドレスは、標準のサプライチェーンで流通しているデバイスの通常想定される IP アドレス範囲外になります。
受信したメッセージの数 メッセージサイズ	攻撃者は MQTT ベースのデータ流出でデバイスを使用する可能性があるため、AWS IoT クラウドからデバイスに送信されたメッセージ数またはメッセージサイズに関連するメトリクスが急増する場合があり、これはセキュリティ上の問題があることを示しています。

[なりすまし:]

なりすまし攻撃は、AWS IoT のクラウド側のサービス、アプリケーション、データにアクセスしたり、IoT デバイスの指揮および制御に関与したりするために、攻撃者が既知のまたは信頼できるエンティティとして振る舞う攻撃です。

[関連メトリクス:]

メトリクス	根拠
認証エラー	攻撃者が盗まれた ID を使用して信頼できるエンティティとして振る舞うと、認証情報が無効になったり、信頼できるデバイスによって既に使用されていたりするため、接続関連のメトリクスが急増することがよくあります。承認の失敗、接続の試行、または切断における異常な動作は、潜在的ななりすましのシナリオを示唆します。
接続の試行	
切断	

[クラウドインフラストラクチャの悪用:]

AWS IoT クラウドサービスの不正利用は、メッセージ量が多いトピックやサイズの大きいトピックを発行またはサブスクライブするときに発生します。コマンドおよび制御に対する過度に寛容なポリシーまたはデバイスの脆弱性の悪用も、クラウドインフラストラクチャの悪用を引き起こす可能性があります。この攻撃の主な目的の 1 つは、AWS の請求額を増やすことです。IoT ポリシーを確認し、[Audit overly permissive checks] (権限が過剰のチェックの監査) をオンにして、権限が過剰であるポリシーを特定する必要があります。

[関連メトリクス:]

メトリクス	根拠
受信したメッセージの数	この攻撃の目的は、AWS の請求額を増やすことです。メッセージ数、受信したメッセージ、メッセージサイズなどのアクティビティを監視するメトリクスが急増します。
送信されたメッセージの数	
メッセージサイズ	

メトリクス	根拠
送信元 IP	疑わしい送信元 IP リストが表示される場合があります。攻撃者は、これらの IP からメッセージボリュームを生成します。

デバイス側のユースケース

Device Defender は、デバイス側で次のユースケースを監視できます。

[サービス拒否攻撃:]

Denial-of-Service (DoS; サービス拒否) 攻撃は、デバイスまたはネットワークをシャットダウンし、目的のユーザがデバイスまたはネットワークにアクセスできないようにすることを目的としています。DoS 攻撃は、ターゲットをトラフィックでフラッディングしたり、システムの起動を遅らせ、もしくはシステムの障害を引き起こすリクエストを送信したりすることで、アクセスをブロックします。IoT デバイスは DoS 攻撃に使用できます。

[関連メトリクス:]

メトリクス	根拠
出力パケット数	通常、DoS 攻撃では、特定のデバイスからの送信通信速度が高くなります。また、DoS 攻撃のタイプによっては、出力パケット数と出力バイト数のいずれかまたは両方が増加する可能性があります。
出力バイト数	
送信先 IP	デバイスが通信する IP アドレス/CIDR 範囲を定義した場合、送信先 IP における異常は、デバイスからの認証されていない IP 通信を示唆する場合があります。
リッスンする TCP ポート	DoS 攻撃には、通常、より大きなコマンドおよび制御インフラストラクチャが必要です。このインフラストラクチャでは、デバイスにインストールされているマルウェアが、攻撃する相手と攻撃するタイミングに関するコマ
リッスンする TCP ポート数	
リッスンする UDP ポート	

メトリクス	根拠
リッスンする UDP ポート数	ンドや情報を受信します。したがって、このような情報を受信するために、マルウェアは通常、デバイスによって通常使用されていないポートでリッスンします。

[横方向の脅威のエスカレーション:]

横方向の脅威のエスカレーションは通常、攻撃者がネットワークの 1 つのポイント (コネクテッドデバイスなど) にアクセスできるようになることから始まります。その後、攻撃者は、盗まれた認証情報や脆弱性の悪用などの方法で、特権レベルまたは他のデバイスへのアクセス権を強化しようとします。

[関連メトリクス:]

メトリクス	根拠
出力パケット数	一般的な状況では、攻撃者はローカルエリアネットワーク上でスキャンを実行し、攻撃対象の選択を絞り込むために、利用可能なデバイスを偵察および特定する必要があります。この種のスキャンでは、出力バイト数および出力パケット数が急増する場合があります。
出力バイト数	
送信先 IP	デバイスが一連の既知の IP アドレスまたは CIDR と通信することになっている場合、異常な IP アドレスとの通信の試行を特定できます。このアドレスは、横方向の脅威のエスカレーションのユースケースでは、ローカルネットワーク上のプライベート IP アドレスであることがよくあります。
認証エラー	攻撃者は、IoT ネットワーク全体で権限のレベルを上げるために、取り消され、または執行した盗まれた認証情報を使用する可能性があります。これにより、承認の失敗が増える可能性があります。

[データ流出または監視:]

データ流出は、マルウェアまたは悪意のある行為者がデバイスまたはネットワークエンドポイントから不正なデータ転送を実行したときに発生します。データ流出は、通常、攻撃者にとってデータや知的財産の取得、またはネットワークの偵察の 2 つの目的を果たすものです。監視 (Surveillance) とは、認証情報を盗み、情報を収集する目的で、悪意のあるコードを使用してユーザーの活動をモニタリングすることを意味します。以下のメトリクスは、いずれかのタイプの攻撃を調査するための出発点となります。

[関連メトリクス:]

メトリクス	根拠
出力パケット数	データ流出や監視攻撃が発生すると、攻撃者は単にデータをリダイレクトするのではなく、デバイスから送信されたデータをミラーリングすることがよくあります。これは、意図したデータが届かないときに防御者によって識別されます。このようなミラーリングされたデータは、デバイスから送信されるデータの総量を大幅に増加させ、出力パケット数および出力バイト数の急増を引き起こします。
出力バイト数	
送信先 IP	攻撃者がデータ流出攻撃や監視攻撃でデバイスを使用している場合、攻撃者が管理する異常な IP アドレスにデータが送信されることになります。送信先 IP をモニタリングすると、このような攻撃を特定できます。

[仮想通貨のマイニング]

攻撃者は、デバイスの処理能力を活用して暗号通貨をマイニングします。暗号通貨マイニングはコンピューティングを多用するプロセスであり、通常、他のマイニングピアやプールとのネットワーク通信を必要とします。

[関連メトリクス:]

メトリクス	根拠
送信先 IP	通常、暗号通貨マイニング中は、ネットワーク通信が必要です。デバイスが通信する必要がある IP アドレスのリストを厳密に制御すると、暗号通貨マイニングなどのデバイス上の意図しない通信を識別するのに役立ちます。
CPU の使用状況のカスタムメトリクス	暗号通貨マイニングは、デバイス CPU の高い使用率をもたらす多くのコンピューティングを必要とします。このメトリクスの収集と監視を選択した場合、通常よりも高い CPU 使用率は、暗号通貨マイニング活動を示唆している可能性があります。

[コマンドと制御、マルウェア、ランサムウェア]

マルウェアやランサムウェアは、デバイスの制御を制限し、デバイスの機能を制限します。ランサムウェア攻撃の場合、ランサムウェアが使用する暗号化により、データへのアクセスが失われます。

[関連メトリクス:]

メトリクス	根拠
送信先 IP	ネットワーク攻撃またはリモート攻撃は、IoT デバイスに対する攻撃の大部分を占めています。デバイスが通信する IP アドレスのリストを厳密に制御することで、マルウェアやランサムウェアの攻撃に起因する異常な送信先 IP を特定できます。
リッスンする TCP ポート リッスンする TCP ポート数	いくつかのマルウェア攻撃には、デバイスで実行するコマンドを送信するコマンド & コントロールサーバーの起動が含まれます。このタイプのサーバーは、マルウェアやランサ

メトリクス	根拠
リッスンする UDP ポート	ムウェアのオペレーションにとって重要であり、開いている TCP/UDP ポートとポート数を厳しく監視することで識別できます。
リッスンする UDP ポート数	

概念

メトリクス

AWS IoT Device Defender Detect は、メトリクスを使用してデバイスの変則的な動作を検出します。AWS IoT Device DefenderDetect は、報告されたメトリクスの値を指定された予測値と比較します。これらのメトリクスは、クラウド側メトリクスとデバイス側メトリクスの 2 つのソースから取得できます。ML Detect では、6 つのクラウド側メトリクスと 7 つのデバイス側メトリクスがサポートされています。ML Detect でサポートされているメトリクスのリストについては、サポートされるメトリクス を参照してください。

AWS IoT ネットワークでの異常な動作は、認証エラーの数や、デバイスが AWS IoT を通じて送受信するメッセージの数またはサイズなど、クラウド側メトリクスを使用して検出されます。

AWS IoT Device Defender Detect は、デバイスがリッスンしているポート、送信されたバイトまたはパケットの数、デバイスの TCP 接続など、AWS IoT デバイスにより生成されたメトリクスを収集、集計、監視することもできます。

クラウド側メトリクスだけに AWS IoT Device Defender Detect を使用することができます。デバイス側メトリクスを使用するには、まず AWS IoT によって接続されたデバイスまたはデバイスゲートウェイに AWS IoT SDK をデプロイしてメトリクスを収集し、AWS IoT に送信する必要があります。「」を参照してくださいデバイスからのメトリクスの送信

セキュリティプロファイル

セキュリティプロファイルは、デバイスのグループ (モノの静的グループ) やアカウント内のすべてのデバイスの変則的な動作を定義し、異常が検出されたときに実行するアクションを指定します。セキュリティプロファイルを作成し、デバイスのグループに関連付けるには、AWS IoT コンソールまたは API コマンドを使用します。AWS IoT Device DefenderDetect は、セキュリティ関連データの記録を開始し、セキュリティプロファイルで定義された動作を使用してデバイスの動作の異常を検出します。

behavior

動作は、デバイスの動作が異常であるかどうかを AWS IoT Device Defender Detect が検出する方法を指定します。動作に一致しないアクションをデバイスが行うと、アラートがトリガーされます。Rules Detect の動作は、メトリクス、および絶対値または統計しきい値 (例えば、次の値以下、次の値以上) で構成されます。これらのしきい値は、予想されるデバイスの動作を表します。ML Detect 動作は、メトリクスと ML Detect 設定で構成されます。これは、デバイスの通常の動作を学習するように ML モデルを設定します。

ML モデル

ML モデルは、お客様が設定する各動作を監視するために作成される機械学習モデルです。このモデルは、ターゲットデバイスグループからのメトリクスデータパターンに基づいてトレーニングし、メトリクスベースの動作の 3 つの異常信頼しきい値 (高、中、低) を生成します。デバイスレベルで取り込まれたメトリクスデータに基づいて異常を推測します。ML Detect のコンテキストでは、1 つの ML モデルが作成され、1 つのメトリクスベースの動作が評価されます。詳細については、「ML 検出」を参照してください。

信頼度レベル

ML Detect は、`High`、`Medium`、`Low` といった 3 つの信頼度レベルをサポートします。`High` の信頼度は、異常動作評価における感度が低く、しばしばアラームの数が少なくなります。`Medium` の信頼度は中程度の感度を意味し、`Low` の信頼度は高感度であり、アラームの数が多くなります。

ディメンション

ディメンションを定義して、動作の範囲を調整できます。たとえば、パターンに一致する MQTT トピックに動作を適用するトピックフィルタディメンションを定義できます。セキュリティプロファイルで使用するディメンションの定義については、「CreateDimension」を参照してください。

アラーム

異常が検出されると、CloudWatch メトリクス (「AWS IoT Core デベロッパーガイド」の「Amazon CloudWatch を使用した AWS IoT アラームとメトリクスのモニタリング」を参照) または SNS 通知を通じてアラーム通知を送信できます。アラーム通知は、アラームの情報、デバイスのアラームの履歴とともに、AWS IoT コンソールにも表示されます。またアラームは、監視対象のデバイスが変則的な動作をやめたときや、アラームを発生させていたが長期間にわたって報告を停止したときも送信されます。

検証状態

アラームが作成されたら、アラームを True positive、良性 positive、False positive、または Unknown として検証できます。また、アラーム検証の状態に説明を追加することもできます。4 の検証状態の1つを使い、AWS IoT Device Defenderアラームを表示、整理、およびフィルタ処理する事ができます。アラーム確認の状態および関連する説明を使用して、チームのメンバーに通知できます。これは、True Positive アラームに対する緩和アクションの実行、良性のポジティブアラームのスキップ、不明なアラームに対する調査の継続など、チームがフォローアップアクションを実行するのに役立ちます。すべてのアラームのデフォルトの検証状態は Unknown です。

アラームの抑制

動作通知を on または `suppressed` に設定して、Detect アラーム SNS 通知を管理します。アラームを抑制しても、Detect はデバイス動作評価の実行を停止しません。Detect は異常動作を違反アラームとしてフラグ付けし続けます。ただし、抑制されたアラームは SNS 通知のために転送されません。AWS IoT コンソールまたは API からのみアクセスできます。

動作

セキュリティプロファイルには一連の動作が含まれています。各動作には、デバイスのグループまたはアカウント内のすべてのデバイスの正常な動作を指定するメトリクスが含まれています。動作は、Rules Detect 動作と ML Detect 動作の 2 つのカテゴリに分類されます。Rules Detect 動作では、デバイスの動作を定義します。一方、ML Detect では、デバイス履歴データに基づいて構築された ML モデルを使用して、デバイスの動作を評価します。

セキュリティプロファイルは、ML またはルールベースという 2 つのしきい値タイプのいずれかになります。ML セキュリティプロファイルは、過去のデータから学習することで、フリート全体のデバイスレベルの運用およびセキュリティの異常を自動的に検出します。ルールベースのセキュリティプロファイルでは、デバイスの動作を監視するために静的ルールを手動で設定する必要があります。

`behavior` の定義に使用されるフィールドには次のようなものがあります。

Rules Detect および ML Detect に共通の事項

`name`

動作の名前。

metric

使用するメトリクスの名前 (つまり、動作によって測定される内容)。

consecutiveDatapointsToAlarm

指定された数の連続するデータポイントの動作にデバイスが違反している場合、アラームが発生します。指定されなかった場合、デフォルト値は 1 です。

consecutiveDatapointsToClear

アラームが発生し、問題のデバイスの動作が、指定された数の連続するデータポイントに違反しなくなった場合、アラームはクリアされます。指定されなかった場合、デフォルト値は 1 です。

threshold type

セキュリティプロファイルは、ML またはルールベースといった 2 つのしきい値タイプのいずれかになります。ML セキュリティプロファイルは、過去のデータから学習することで、フリート全体のデバイスレベルの運用およびセキュリティの異常を自動的に検出します。ルールベースのセキュリティプロファイルでは、デバイスの動作を監視するために静的ルールを手動で設定する必要があります。

alarm suppressions

動作通知を on または `suppressed` に設定して、アラーム検出の Amazon SNS 通知を管理できます。アラームを抑制しても、Detect はデバイス動作評価の実行を停止しません。Detect は異常動作を違反アラームとしてフラグ付けし続けます。ただし、抑制されたアラームは Amazon SNS 通知のために転送されません。AWS IoT コンソールまたは API からアクセスできます。

Rules Detect

`dimension`

ディメンションを定義して、動作の範囲を調整できます。たとえば、パターンに一致する MQTT トピックに動作を適用するトピックフィルタディメンションを定義できます。セキュリティプロファイルで使用するディメンションを定義するには、「CreateDimension」を参照してください。Rules Detect にのみ適用されます。

`criteria`

デバイスが `metric` に関して正常に動作しているかどうかを判断する条件。

> **Note**
>
> AWS IoT コンソールで [アラートを通知] を選択すると、AWS IoT Device Defender がデバイスの異常な動作を検出したきに、Amazon SNS を通じて通知を受けることができます。

`comparisonOperator`

測定対象のモノ (`metric`) を条件 (`value` または `statisticalThreshold`) に関連付ける演算子。

有効な値は、"less-than"、"less-than-equals"、"greater-than"、"greater-than-equals"、"in-cidr-set"、"not-in-cidr-set"、"in-port-set"、"not-in-port-set" です。すべての演算子をすべてのメトリクスで使用できるわけではありません。たとえば、CIDR セットやポートの演算子は、それらを含むメトリクスでのみ使用できます。

`value`

`metric` と比較する値。メトリクスの種類に応じて、`count` (値)、`cidrs` (CIDR のリスト)、`ports` (ポートのリスト) のいずれかを含める必要があります。

`statisticalThreshold`

動作違反が決定される統計的しきい値。このフィールドには、以下の値を指定できる `statistic` フィールドが含まれています。「p0」、「p0.1」、「p0.01」、「p1」、「p10」、「p50」、「p90」、「p99」、「p99.9」、「p99.99」、または「p100」。

この `statistic` はパーセンタイルを示します。動作への適合が決定される値に解決されます。このセキュリティプロファイルに関連付けられたすべての報告デバイスから、指定された期間 (`durationSeconds`) にわたり 1 回以上メトリクスが収集され、そのデータに基づいてパーセンタイルが計算されます。その後、デバイスに対して測定が収集され、同じ期間にわたり蓄積されます。そのデバイスの結果の値が、指定されたパーセンタイルに関連付けられている値 (`comparisonOperator`) を上回ったり、下回ったりする場合、デバイスは動作に準拠していると見なされます。それ以外の場合、デバイスは動作の違反になります。

パーセンタイルは、関連付けられた値を下回る、考慮されたすべての測定の割合を示します。たとえば、「p90」(90 パーセンタイル) に関連付けられた値が 123 である場合、すべての測定の 90% が 123 未満です。

`durationSeconds`

時間ディメンション (NUM_MESSAGES_SENT など) を持つ条件に対して動作が評価される期間を指定するには、これを使用します。statisticalThreshhold メトリクスの比較では、これは statisticalThreshold の値を決定するためにすべてのデバイスの測定が収集され、比較におけるその動作のランクを各デバイスが決定するための期間です。

ML 検出

`ML Detect confidence`

ML Detect は、High、Medium、Low といった 3 つの信頼度レベルをサポートします。High の信頼度は、異常動作評価における感度が低く、しばしばアラームの数が少なくなります。Medium の信頼度は中程度の感度を意味し、Low の信頼度は高感度であり、アラームの数が多くなります。

ML 検出

機械学習 Detect (ML Detect) では、デバイス履歴データに基づいてモデルを自動的に作成することで、機械学習を使用して想定されるデバイスの動作を学習するセキュリティプロファイルを作成し、これらのプロファイルをデバイスのグループまたはフリート内のすべてのデバイスに割り当てます。そして、AWS IoT Device Defender は異常を識別し ML モデルを使用してアラームをトリガーします。

ML Detect の開始方法の詳細については、ML Detect ガイド を参照してください。

この章には、以下のセクションが含まれています。

- ML Detect セキュリティプロファイルを一時停止または削除する

ML Detect のユースケース

ML Detect を使用すると、想定されるデバイスの動作を設定することが困難な場合に、フリートデバイスを監視できます。例えば、切断数のメトリクスを監視する場合、許容可能なしきい値とみなされる値が明確でない場合があります。この場合、ML Detect を有効にして、デバイスから報告された履歴データに基づいて、異常な切断メトリクスデータポイントを特定できます。

ML Detect のもう 1 つのユースケースは、時間の経過とともに動的に変化するデバイスの動作を監視することです。ML Detect は、デバイスからのデータパターンの変化に基づいて、想定される動的なデバイス動作を定期的に学習します。例えば、デバイスメッセージの送信量は、平日と週末の間で変化し、ML Detect はこの動的な動作を学習します。

ML Detect の仕組み

ML Detect を使用すると、6 個のクラウド側メトリクスと 7 個のデバイス側メトリクスで運用およびセキュリティの異常を識別するための動作を作成できます。初期モデルトレーニング期間後、ML Detect は、後続の 14 日間のデータに基づいてモデルを毎日更新します。ML モデルでこれらのメトリクスのデータポイントを監視し、異常が検出されるとアラームをトリガーします。

ML Detect は、想定される動作が似ている一連のデバイスにセキュリティプロファイルをアタッチする場合に最適です。例えば、一部のデバイスが顧客の自宅で使用され、その他のデバイスが事業所で使用されている場合、デバイスの動作パターンが 2 つのグループ間で大きく異なる場合があります。デバイスを home-device のモノのグループと office-device のモノグループに整理できます。異常検出を適切なものとするためには、各モノのグループを個別の ML Detect セキュリティプロファイルにアタッチします。

ML Detect が初期モデルを構築している間、モデルを生成するには、後続の 14 日間で、メトリクスあたり 14 日間以上、かつ少なくとも 25,000 のデータポイントが必要です。その後、最小数のメトリクスデータポイントがあるモデルを毎日更新します。最小要件が満たされない場合、ML Detect はモデルの作成を翌日に試行し、評価のためのモデルの使用を中止する前に、次の 30 日間毎日再試行します。

最小要件

ML Detect の初期モデルのトレーニングと作成のために、ML Detect は次の最小要件を満たす必要があります。

最小トレーニング期間

初期モデルの構築には 14 日かかります。その後、モデルは 14 日間の追跡期間のメトリクスデータで毎日更新されます。

データポイントの合計

ML モデルを構築するために必要なデータポイントの最小数は、過去 14 日間でメトリクスあたり 25,000 データポイントです。モデルの継続的なトレーニングと更新のために、ML Detect では、監視対象デバイスからの最小データポイントが必要です。これは、次の設定とほぼ同等です。

- 60 台のデバイスが AWS IoT 上で45 分間隔で接続しアクティビティを実行
- 40 台のデバイスによる30 分間隔での接続と実行
- 15 台のデバイスによる10 分間隔での接続と実行
- 7 台のデバイスによる5 分間隔での接続と実行

デバイスグループのターゲット

データを収集するには、セキュリティプロファイルにおけるターゲットのモノのグループにモノが含まれている必要があります。

初期モデルが作成されると、ML モデルは毎日更新され、14 日間の後続期間に少なくとも 25,000 のデータポイントが必要になります。

制限事項

次のクラウド側のメトリクスのディメンションで ML Detect を使用できます。

- 認可の失敗 (aws:num-authorization-failures)
- 受信したメッセージ (aws:num-messages-received)
- 送信されたメッセージ (aws:num-messages-sent)
- メッセージサイズ (aws:message-byte-size)

次のメトリクスは、ML Detect ではサポートされていません。

ML Detect でサポートされていないクラウド側のメトリクス:

- 送信元 IP (aws:source-ip-address)

ML Detect でサポートされていないデバイス側のメトリクス:

- 送信先 IP (aws:destination-ip-addresses)
- リッスンする TCP ポート (aws:listening-tcp-ports)
- リッスンする UDP ポート (aws:listening-udp-ports)

カスタムメトリクスは、数値タイプのみをサポートします。

アラームでの誤検とその他の検証状態のマーキング

調査を通じて ML Detect アラームが誤検出であることを確認した場合は、アラームの検証状態を False positive に設定できます。これは、ユーザーとユーザーチームが、応答する必要のないアラームを特定するのに役立ちます。またアラームを真正、良性正、または未知としてマークすることもできます。

AWS IoT Device Defenderconsoleを使用するか、PutVerificationStateon違反API アクションを使用して、アラームをマークする事ができます。

サポートされるメトリクス

ML Detect では、次のクラウド側のメトリクスを使用できます。

- 認可の失敗 (aws:num-authorization-failures)
- 接続試行 (aws:num-connection-attempts)
- 切断 (aws:num-disconnects)
- メッセージサイズ (aws:message-byte-size)
- 送信されたメッセージ (aws:num-messages-sent)
- 受信したメッセージ (aws:num-messages-received)

ML Detect では、次のデバイス側のメトリクスを使用できます。

- 出力バイト数 (aws:all-bytes-out)
- 入力バイト数 (aws:all-bytes-in)
- リスニング TCP ポート数 (aws:num-listening-tcp-ports)
- リスニング UDP ポート数 (aws:num-listening-udp-ports)
- 出力パケット (aws:all-packets-out)
- 入力パケット (aws:all-packets-in)

- 確立された TCP 接続数 (aws:num-established-tcp-connections)

サービスクォータ

ML Detect サービスのクォータと制限については、「AWS IoT Device Defender のエンドポイントとクォータ」を参照してください。

ML Detect CLI コマンド

以下の CLI コマンドを使用して ML Detect を作成および管理できます。

- create-security-profile
- attach-security-profile
- list-security-profiles
- describe-security-profile
- update-security-profile
- delete-security-profile
- get-behavior-model-training-summaries
- list-active-violations
- list-violation-events

ML Detect API

ML Detect セキュリティプロファイルの作成と管理には、次の API を使用できます。

- CreateSecurityProfile
- AttachSecurityProfile
- ListSecurityProfiles
- DescribeSecurityProfile
- UpdateSecurityProfile
- DeleteSecurityProfile
- GetBehaviorModelTrainingSummaries
- ListActiveViolations
- ListViolationEvents

- PutVerificationStateon違反

ML Detect セキュリティプロファイルを一時停止または削除する

ML Detect セキュリティプロファイルを一時停止してデバイス動作の監視を一時的に停止したり、ML Detect セキュリティプロファイルを削除してデバイス動作の監視を長期間停止したりできます。

コンソールを使用して ML Detect セキュリティプロファイルを一時停止する

コンソールを使用して ML Detect セキュリティプロファイルを一時停止するには、まず空のモノのグループが必要です。空のモノのグループを作成するには、「AWS IoT Core デベロッパーガイド」の「モノの静的グループ」を参照してください。空のモノのグループを作成した場合は、空のモノのグループを ML Detect セキュリティプロファイルのターゲットとして設定します。

> **Note**
>
> セキュリティプロファイルのターゲットを 30 日以内にデバイスを持つデバイスグループに戻すよう設定する必要があります。この設定を行わない場合、セキュリティプロファイルを再度有効化できません。

コンソールを使用して ML Detect セキュリティプロファイルを削除する

セキュリティプロファイルを削除するには、次の手順を実行します。

1. AWS IoT コンソールでサイドバーに移動し、[Defend] (防御) セクションを選択します。
2. [Defend] (防御) で、[Detect] (検出)、[Security Profiles] (セキュリティプロファイル) の順に選択します。
3. 削除する ML Detect セキュリティプロファイルを選択します。
4. [Actions] (アクション) を選択し、オプションから [Delete] (削除) を選択します。

> **Note**
>
> ML Detect セキュリティプロファイルを削除すると、セキュリティプロファイルを再度有効化できなくなります。

CLI を使用して ML Detect セキュリティプロファイルを一時停止する

CLI を使用して ML Detect セキュリティプロファイルを一時停止するには、`detach-security-security-profile` コマンドを使用します。

```
$aws iot detach-security-profile --security-profile-name SecurityProfileName --
security-profile-target-arn arn:aws:iot:us-east-1:123456789012:all/registered-things
```

> **Note**
>
> このオプションは、AWS CLI でのみ使用できます。コンソールのワークフローと同様に、セキュリティプロファイルのターゲットを 30 日以内にデバイスを持つデバイスグループに戻すよう設定する必要があります。この設定を行わない場合、セキュリティプロファイルを再度有効化できません。セキュリティプロファイルをデバイスグループにアタッチするには、attach-security-profileコマンドを使用します。

CLI を使用して ML Detect セキュリティプロファイルを削除する

以下の `delete-security-profile` コマンドを実行することで、セキュリティプロファイルを削除できます。

```
delete-security-profile --security-profile-name SecurityProfileName
```

> **Note**
>
> ML Detect セキュリティプロファイルを削除すると、セキュリティプロファイルを再度有効化できなくなります。

カスタムメトリクス

AWS IoT Device Defender カスタムメトリクスを使用すると、Wi-Fi ゲートウェイに接続されたデバイスの数、バッテリの充電レベル、スマートプラグの電源サイクル数など、フリートやユースケースに固有のメトリクスを定義およびモニタリングできます。カスタムメトリクスの動作は、セキュリティプロファイルで定義されています。セキュリティプロファイルでは、デバイスのグループ (モノのグループ) の想定される動作を指定します。アラームを設定することで、動作を監視できます。アラームを使用すると、デバイスに固有の問題を検出して対応できます。

この章には、以下のセクションが含まれています。

コンソールでのカスタムメトリクスの使用方法

チュートリアル

- AWS IoT Device Defender Agent SDK (Python)
- カスタムメトリクスを作成し、セキュリティプロファイルに追加する
- カスタムメトリクスの詳細を表示する
- カスタムメトリクスを更新する
- カスタムメトリクスを削除する

AWS IoT Device Defender Agent SDK (Python)

使用を開始するには、AWS IoT Device Defender Agent SDK (Python) サンプルエージェントをダウンロードします。エージェントはメトリクスを収集し、レポートを発行します。デバイス側のメトリクスを発行すると、収集されるメトリクスを表示して、アラームを設定するためのしきい値を決定できます。デバイスエージェントの設定手順については、「AWS IoT Device Defender Agent SDK (Python) Readme」を参照してください。詳細については、「AWS IoT Device Defender Agent SDK (Python)」を参照してください。

カスタムメトリクスを作成し、セキュリティプロファイルに追加する

次の手順は、コンソールでカスタムメトリクスを作成する方法を示しています。

1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Metrics] (メトリクス) の順に選択します。
2. [Custom metrics] (カスタムメトリクス) ページで、[Create] (作成) を選択します。
3. [Create custom metric] (カスタムメトリクスを作成) ページで、次の操作を行います。

    1. [Name] (名前) で、カスタムメトリクスの名前を入力します。カスタムメトリクスを作成した後は、この名前を変更できません。

2. [Display name] (表示名) (オプション) で、カスタムメトリクスのわかりやすい名前を入力できます。名前は一意である必要はなく、作成後に変更することができます。
3. [Type] (タイプ) で、監視するメトリクスのタイプを選択します。メトリクスのタイプには、string-list、ip-address-list、number-list、および number が含まれます。タイプは、作成後に変更することはできません。

   > **Note**
   >
   > ML Detect では、数値タイプのみ使用可能です。

4. [Tags] (タグ) で、リソースに関連付けるタグを選択できます。

   完了したら、[Confirm] (確認) を選択します。

4. カスタムメトリクスを作成すると、[Custom metrics] (カスタムメトリクス) ページが表示され、新しく作成したカスタムメトリクスを確認できます。
5. 次に、カスタムメトリクスをセキュリティプロファイルに追加する必要があります。AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Security profiles] (セキュリティプロファイル) の順に選択します。
6. カスタムメトリクスを追加するセキュリティプロファイルを選択します。
7. [Actions]、[Edit] の順に選択します。
8. [Additional Metrics to retain] (保持する追加のメトリクス) を選択してから、カスタムメトリクスを選択します。[Confirm] (確認) ページが表示されるまで、次の画面で [Next] (次へ) を選択します。[Save] (保存) と [Continue] (続行) を選択します。カスタムメトリクスが正常に追加されると、セキュリティプロファイルの詳細ページが表示されます。

   > **Note**
   >
   > パーセンタイル統計は、メトリクス値が負の数値のメトリクスに対して使用することはできません。

カスタムメトリクスの詳細を表示する

次の手順では、カスタムメトリクスの詳細をコンソールで表示する方法を示します。

1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Metrics] (メトリクス) の順に選択します。
2. 詳細を表示するカスタムメトリクスの [Metric name] (メトリクス名) を選択します。

カスタムメトリクスを更新する

次の手順は、コンソールでカスタムメトリクスを更新する方法を示しています。

1. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Metrics] (メトリクス) の順に選択します。
2. 更新するカスタムメトリクスの横にあるオプションボタンを選択します。その後、[Actions] (アクション) で、[Edit] (編集) を選択します。
3. [Update custom metric] (カスタムメトリクスの更新) ページで、表示名を編集したり、タグを削除または追加したりできます。
4. 完了したら、[Update] (更新) を選択します。[Custom metrics] (カスタムメトリクス) ページ。

カスタムメトリクスを削除する

次の手順は、コンソールでカスタムメトリクスを削除する方法を示しています。

1. まず、カスタムメトリクスを参照元のセキュリティプロファイルから削除します。カスタムメトリクスの詳細ページで、カスタムメトリクスが含まれているセキュリティプロファイルを表示できます。AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Metrics] (メトリクス) の順に選択します。
2. 削除するカスタムメトリクスを選択します。カスタムメトリクスの詳細ページの [Security Profiles] (セキュリティプロファイル) の下にリストされているセキュリティプロファイルからカスタムメトリクスを削除します。
3. AWS IoT コンソールのナビゲーションペインで、[Defend] (防御) を展開し、[Detect] (検出)、[Metrics] (メトリクス) の順に選択します。
4. 削除するカスタムメトリクスの横にあるオプションボタンを選択します。その後、[Actions] (アクション) で、[Delete] (削除) を選択します。
5. [Are you sure you want to delete custom metric?] (カスタムメトリクスを削除してもよろしいですか?) というメッセージが表示されたら、[Delete custom metric] (カスタムメトリクスの削除) を選択します。

> **Warning**
>
> カスタムメトリクスを削除すると、そのメトリクスに関連付けられているすべてのデータが失われます。この操作は元に戻すことができません。

CLI からのカスタムメトリクスの使用方法

チュートリアル

- AWS IoT Device Defender Agent SDK (Python)
- カスタムメトリクスを作成し、セキュリティプロファイルに追加する
- カスタムメトリクスの詳細を表示する
- カスタムメトリクスを更新する
- カスタムメトリクスを削除する

AWS IoT Device Defender Agent SDK (Python)

使用を開始するには、AWS IoT Device Defender Agent SDK (Python) サンプルエージェントをダウンロードします。エージェントはメトリクスを収集し、レポートを発行します。デバイス側のメトリクスを発行したら、収集するメトリクスを表示して、アラームを設定するためのしきい値を決定できます。デバイスエージェントの設定手順については、「AWS IoT Device Defender Agent SDK (Python) Readme」を参照してください。詳細については、「AWS IoT Device Defender Agent SDK (Python)」を参照してください。

カスタムメトリクスを作成し、セキュリティプロファイルに追加する

次の手順では、カスタムメトリクスを作成し、CLI からセキュリティプロファイルに追加する方法について説明します。

1. `create-custom-metric` コマンドを使用して、カスタムメトリクスを作成します。次の例では、バッテリーの割合を測定するカスタムメトリクスを作成します。

```
aws iot create-custom-metric \
    --metric-name "batteryPercentage" \
    --metric-type "number" \
    --display-name "Remaining battery percentage." \
```

```
    --region us-east-1
    --client-request-token "02ccb92b-33e8-4dfa-a0c1-35b181ed26b0" \
```

出力:

```
{
    "metricName": "batteryPercentage",
    "metricArn": "arn:aws:iot:us-
east-1:1234564789012:custommetric/batteryPercentage"
}
```

2. カスタムメトリクスを作成したら、update-security-profile を使用して既存のプロファイルにカスタムメトリクスを追加するか、create-security-profile を使用してカスタムメトリクスを追加するために新しいセキュリティプロファイルを作成できます。ここでは、*batteryUsage* という新しいセキュリティプロファイルを作成して、新しい *batteryPercentage* カスタムメトリクスを追加します。また、*cellularBandwidth* と呼ばれる Rules Detect メトリクスを追加します。

```
aws iot create-security-profile \
    --security-profile-name batteryUsage \
    --security-profile-description "Shows how much battery is left in percentile."
 \
    --behaviors "[{\"name\":\"great-than-75\",\"metric\":\"batteryPercentage\",
\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"number
\":75},\"consecutiveDatapointsToAlarm\":5,\"consecutiveDatapointsToClear
\":1}},{\"name\":\"cellularBandwidth\",\"metric\":\"aws:message-byte-size\",
\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]" \
    --region us-east-1
```

出力:

```
{
    "securityProfileArn": "arn:aws:iot:us-east-1:1234564789012:securityprofile/
batteryUsage",
    "securityProfileName": "batteryUsage"
}
```

> **Note**
>
> パーセンタイル統計は、メトリクス値が負の数値のメトリクスに対して使用することはできません。

カスタムメトリクスの詳細を表示する

次の手順は、CLI からカスタムメトリクスの詳細を表示する方法を示しています。

- list-custom-metrics コマンドを使用して、すべてのカスタムメトリクスを表示します。

```
aws iot list-custom-metrics \
    --region us-east-1
```

このコマンドの出力は以下のようになります。

```
{
    "metricNames": [
        "batteryPercentage"
    ]
}
```

カスタムメトリクスを更新する

次の手順は、CLI からカスタムメトリクスを更新する方法を示しています。

- update-custom-metric コマンドを使用して、カスタムメトリクスを更新します。次の例では、display-name を更新します。

```
aws iot update-custom-metric \
    --metric-name batteryPercentage \
    --display-name 'remaining battery percentage on device' \
    --region us-east-1
```

このコマンドの出力は以下のようになります。

```
{
    "metricName": "batteryPercentage",
```

```
    "metricArn": "arn:aws:iot:us-
east-1:1234564789012:custommetric/batteryPercentage",
    "metricType": "number",
    "displayName": "remaining battery percentage on device",
    "creationDate": "2020-11-17T23:01:35.110000-08:00",
    "lastModifiedDate": "2020-11-17T23:02:12.879000-08:00"
}
```

カスタムメトリクスを削除する

次の手順は、CLI からカスタムメトリクスを削除する方法を示しています。

1. カスタムメトリクスを削除するには、まず、アタッチ先のセキュリティプロファイルからそのカスタムメトリクスを削除します。特定のカスタムメトリクスがアタッチされているセキュリティプロファイルを表示するには、list-security-profiles コマンドを使用します。
2. セキュリティプロファイルからカスタムメトリクスを削除するには、update-security-profiles コマンドを使用します。保持するすべての情報を入力します。ただし、カスタムメトリクスは除外します。

```
aws iot update-security-profile \
  --security-profile-name batteryUsage \
  --behaviors "[{\"name\":\"cellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

このコマンドの出力は以下のようになります。

```
{
  "behaviors": [{\"name\":\"cellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}],
  "securityProfileName": "batteryUsage",
  "lastModifiedDate": 2020-11-17T23:02:12.879000-09:00,
  "securityProfileDescription": "Shows how much battery is left in percentile.",
  "version": 2,
  "securityProfileArn": "arn:aws:iot:us-east-1:1234564789012:securityprofile/
batteryUsage",
  "creationDate": 2020-11-17T23:02:12.879000-09:00
}
```

3. カスタムメトリクスがデタッチされたら、delete-custom-metric コマンドを使用してカスタムメトリクスを削除します。

```
aws iot delete-custom-metric  \
  --metric-name batteryPercentage \
  --region us-east-1
```

このコマンドの出力は以下のようになります。

```
HTTP 200
```

カスタムメトリクス CLI コマンド

以下の CLI コマンドを使用してカスタムメトリクスを作成および管理できます。

- create-custom-metric
- describe-custom-metric
- list-custom-metrics
- update-custom-metric
- delete-custom-metric
- list-security-profiles

カスタムメトリクス API

次の API を使用して、カスタムメトリクスを作成および管理できます。

- CreateCustomMetric
- DescribeCustomMetric
- ListCustomMetrics
- UpdateCustomMetric
- DeleteCustomMetric
- ListSecurityProfiles

デバイス側のメトリクス

セキュリティプロファイルを作成するときに、IoT デバイスによって生成されるメトリクスの動作としきい値を設定することで、IoT デバイスの想定される動作を指定できます。以下は、デバイス側のメトリクスです。これは、デバイスにインストールしたエージェントから取得したメトリクスです。

出力バイト数 (**aws:all-bytes-out**)

一定期間内におけるデバイスからのアウトバウンドバイト数。

一定期間内にデバイスが送信する必要があるアウトバウンドトラフィックの最大量と最小量 (バイト単位で測定) を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: バイト

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 4096
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **statisticalThreshold** を使用した例

```
{
```

```
  "name": "TCP outbound traffic",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 900,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Outbound traffic ML behavior",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

入力バイト数 (**aws:all-bytes-in**)

一定期間内におけるデバイスへのインバウンドバイト数。

一定期間内にデバイスが受信する必要があるインバウンドトラフィックの最大量と最小量 (バイト単位で測定) を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: バイト

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-bytes-in",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 4096
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **statisticalThreshold** を使用した例

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-bytes-in",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Inbound traffic ML behavior",
  "metric": "aws:all-bytes-in",
  "criteria": {
```

```
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

リスニング TCP ポート数 (**aws:num-listening-tcp-ports**)

デバイスがリッスンしている TCP ポートの数。

各デバイスが監視する必要がある TCP ポートの最大数を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect | ML Detect

単位: エラー

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: エラー

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "Max TCP Ports",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
```

```
  "suppressAlerts": true
}
```

Example **statisticalThreshold** を使用した例

```
{
  "name": "Max TCP Ports",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Max TCP Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

リスニング UDP ポート数 (**aws:num-listening-udp-ports**)

デバイスがリッスンしている UDP ポートの数。

各デバイスが監視する必要がある UDP ポートの最大数を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect | ML Detect

単位: エラー

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: エラー

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "Max UDP Ports",
  "metric": "aws:num-listening-udp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`** を使用した例

```
{
  "name": "Max UDP Ports",
  "metric": "aws:num-listening-udp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
```

```
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Max UPD Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

出力パケット (**aws:all-packets-out**)

一定期間内におけるデバイスからのアウトバウンドパケット数。

一定期間内にデバイスが送信する必要がある合計アウトバウンドトラフィックの最大量と最小量を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: パケット

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-packets-out",
  "criteria": {
```

```
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 100
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **statisticalThreshold** を使用した例

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-packets-out",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Outbound sent ML behavior",
  "metric": "aws:all-packets-out",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

入力パケット (**aws:all-packets-in**)

一定期間内におけるデバイスへのインバウンドパケット数。

一定期間内にデバイスが受信する必要がある合計インバウンドトラフィックの最大量と最小量を指定するには、このメトリクスを使用します。

次と互換性あり: Rule Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: パケット

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-packets-in",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 100
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example

`statisticalThreshold` を使用した例

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-packets-in",
  "criteria": {
```

```
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Inbound sent ML behavior",
  "metric": "aws:all-packets-in",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

送信先 IP (**aws:destination-ip-addresses**)

IP 送信先のセット。

各デバイスが AWS IoT に接続する必要がある、または接続してはならない、許可された Classless Inter-Domain Routings (CIDR) のセット (以前はホワイトリストに登録されたとしていた) または拒否された CIDR のセット (以前はブラックリストに登録されたとしていた) を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect

演算子: in-cidr-set | not-in-cidr-set

値: CIDR のリスト

単位: 該当なし

Example

```
{
  "name": "Denied source IPs",
  "metric": "aws:destination-ip-address",
  "criteria": {
    "comparisonOperator": "not-in-cidr-set",
    "value": {
      "cidrs": [ "12.8.0.0/16", "15.102.16.0/24" ]
    }
  },
  "suppressAlerts": true
}
```

リッスンする TCP ポート (**aws:listening-tcp-ports**)

デバイスがリッスンしている TCP ポート。

各デバイスが接続する必要がある、または接続してはならない、許可された TCP ポートのセット (以前はホワイトリストに登録されたとしていた) または拒否された TCP ポートのセット (以前はブラックリストに登録されたとしていた) を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect

演算子: in-port-set | not-in-port-set

値: ポートのリスト

単位: 該当なし

Example

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 443, 80 ]
    }
  },
  "suppressAlerts": true
```

```
}
```

リッスンする UDP ポート (**aws:listening-udp-ports**)

デバイスがリッスンしている UDP ポート。

各デバイスが接続する必要がある、または接続してはならない、許可された UDP ポートのセット (以前はホワイトリストに登録されたとしていた) または拒否された UDP ポートのセット (以前はブラックリストに登録されたとしていた) を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect

演算子: in-port-set | not-in-port-set

値: ポートのリスト

単位: 該当なし

Example

```
{
  "name": "Listening UDP Ports",
  "metric": "aws:listening-udp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 1025, 2000 ]
    }
  }
}
```

確立された TCP 接続数 (**aws:num-established-tcp-connections**)

デバイスの TCP 接続の数。

各デバイスが必要とするアクティブな TCP 接続の最大値または最小数を指定するには、このメトリクスを使用します (すべての TCP 状態)。

次と互換性あり: Rules Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: 接続

Example

```
{
  "name": "TCP Connection Count",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 3
    },
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **statisticalThreshold** を使用した例

```
{
  "name": "TCP Connection Count",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 900,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Connection count ML behavior",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
```

```
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

デバイスメトリクスドキュメントの仕様

全体構造

長い名前	短縮名	必須	タイプ	制約	コメント
ヘッダー	hed	Y	オブジェクト		正しい形式のレポートに必要なブロックを完成させます。
メトリクス	met	Y	オブジェクト		レポートは、metrics と custom_metrics ブロックの両方または少なくともいずれか 1 つを持つことができます。
custom_metrics	cmet	Y	オブジェクト		レポートは、metrics と custom_metrics ブロックの両方または少なくともいずれ

長い名前	短縮名	必須	タイプ	制約	コメント
					か 1 つを持つことができます。

ヘッダーブロック

長い名前	短縮名	必須	タイプ	制約	コメント
report_id	rid	Y	整数		一定間隔で増加する値。エポックタイムスタンプが推奨されています。
バージョン	v	Y	文字列	Major.Minor	フィールドの追加があるマイナー増分。メトリクスが削除された場合はメジャー増分。

メトリクスブロック:

TCP 接続

長い名前	短縮名	親要素	必須	タイプ	制約	コメント
tcp_connections	tc	メトリクス	N	オブジェクト		
established_connections	ec	tcp_connections	N	オブジェクト		確立された TCP 状態

長い名前	短縮名	親要素	必須	タイプ	制約	コメント
接続	cs	established_connections	N	List<Object>		
remote_addr	rad	接続	Y	数値	ip:port	IP は IPv6 または IPv4
local_port	lp	接続	N	数値	>= 0	
local_interface	li	接続	N	文字列		インターフェイス名
total	t	established_connections	N	数値	>= 0	確立した接続の数

リッスンする TCP ポート

長い名前	短縮名	親要素	必須	タイプ	制約	コメント
listening_tcp_ports	tp	メトリクス	N	オブジェクト		
ports	pts	listening_tcp_ports	N	List<Object>	> 0	
port	pt	ports	N	数値	> 0	ポートは 0 より大きい数値にする必要があります
インターフェイスか	if	ports	N	文字列		インターフェイス名

長い名前	短縮名	親要素	必須	タイプ	制約	コメント
らリクエスト						
total	t	listening_tcp_ports	N	数値	>= 0	

リッスンする UDP ポート

長い名前	短縮名	親要素	必須	タイプ	制約	コメント
listening_udp_ports	up	メトリクス	N	オブジェクト		
ports	pts	listening_udp_ports	N	List<Port>	> 0	
port	pt	ports	N	数値	> 0	ポートは 0 より大きい数値にする必要があります
インターフェイスからリクエスト	if	ports	N	文字列		インターフェイス名
total	t	listening_udp_ports	N	数値	>= 0	

ネットワーク統計

長い名前	短縮名	親要素	必須	タイプ	制約	コメント
network_stats	ns	メトリクス	N	オブジェクト		

長い名前	短縮名	親要素	必須	タイプ	制約	コメント
bytes_in	bi	network_stats	N	数値	デルタメトリクス、>= 0	
bytes_out	bo	network_stats	N	数値	デルタメトリクス、>= 0	
packets_in	pi	network_stats	N	数値	デルタメトリクス、>= 0	
packets_out	po	network_stats	N	数値	デルタメトリクス、>= 0	

Example

次の JSON 構造では、長い名前を使用します。

```
{
  "header": {
    "report_id": 1530304554,
    "version": "1.0"
  },
  "metrics": {
    "listening_tcp_ports": {
      "ports": [
        {
          "interface": "eth0",
          "port": 24800
        },
        {
          "interface": "eth0",
          "port": 22
        },
        {
          "interface": "eth0",
```

```
          "port": 53
        }
      ],
      "total": 3
    },
    "listening_udp_ports": {
      "ports": [
        {
          "interface": "eth0",
          "port": 5353
        },
        {
          "interface": "eth0",
          "port": 67
        }
      ],
      "total": 2
    },
    "network_stats": {
      "bytes_in": 29358693495,
      "bytes_out": 26485035,
      "packets_in": 10013573555,
      "packets_out": 11382615
    },
    "tcp_connections": {
      "established_connections": {
        "connections": [
          {
            "local_interface": "eth0",
            "local_port": 80,
            "remote_addr": "192.168.0.1:8000"
          },
          {
            "local_interface": "eth0",
            "local_port": 80,
            "remote_addr": "192.168.0.1:8000"
          }
        ],
        "total": 2
      }
    }
  },
  "custom_metrics": {
    "MyMetricOfType_Number": [
```

```
      {
        "number": 1
      }
    ],
    "MyMetricOfType_NumberList": [
      {
        "number_list": [
          1,
          2,
          3
        ]
      }
    ],
    "MyMetricOfType_StringList": [
      {
        "string_list": [
          "value_1",
          "value_2"
        ]
      }
    ],
    "MyMetricOfType_IpList": [
      {
        "ip_list": [
          "172.0.0.0",
          "172.0.0.10"
        ]
      }
    ]
  }
}
```

Example 短い名前を使用したサンプル JSON 構造

```
{
  "hed": {
    "rid": 1530305228,
    "v": "1.0"
  },
  "met": {
    "tp": {
      "pts": [
        {
```

```
          "if": "eth0",
          "pt": 24800
        },
        {
          "if": "eth0",
          "pt": 22
        },
        {
          "if": "eth0",
          "pt": 53
        }
      ],
      "t": 3
    },
    "up": {
      "pts": [
        {
          "if": "eth0",
          "pt": 5353
        },
        {
          "if": "eth0",
          "pt": 67
        }
      ],
      "t": 2
    },
    "ns": {
      "bi": 29359307173,
      "bo": 26490711,
      "pi": 10014614051,
      "po": 11387620
    },
    "tc": {
      "ec": {
        "cs": [
          {
            "li": "eth0",
            "lp": 80,
            "rad": "192.168.0.1:8000"
          },
          {
            "li": "eth0",
            "lp": 80,
```

```
            "rad": "192.168.0.1:8000"
          }
        ],
        "t": 2
      }
    }
  },
  "cmet": {
    "MyMetricOfType_Number": [
      {
        "number": 1
      }
    ],
    "MyMetricOfType_NumberList": [
      {
        "number_list": [
          1,
          2,
          3
        ]
      }
    ],
    "MyMetricOfType_StringList": [
      {
        "string_list": [
          "value_1",
          "value_2"
        ]
      }
    ],
    "MyMetricOfType_IpList": [
      {
        "ip_list": [
          "172.0.0.0",
          "172.0.0.10"
        ]
      }
    ]
  }
}
```

デバイスからのメトリクスの送信

AWS IoT Device Defender Detect は、AWS IoT デバイスによって生成されたメトリクスデータを収集、集計、監視し、異常な動作をしているデバイスを特定できます。このセクションでは、デバイスから AWS IoT Device Defender にメトリクスを送信する方法について説明します。

デバイス側のメトリクスを収集するには、AWS IoT コネクテッドデバイスまたはデバイスゲートウェイに AWS IoT SDK バージョン 2 を安全な方法でデプロイする必要があります。SDK の完全なリストはこちらをご覧ください。

AWS IoT Device Client は、AWS IoT Device Defender と AWS IoT Device Management の両方に存在する機能を力バーする単一のエージェントを提供することから、メトリクスの発行に使用できます。これらの機能には、ジョブ、セキュアトンネリング、AWS IoT Device Defender メトリクスの発行などがあります。

AWS IoT Device Defender が収集および評価するために、デバイス側のメトリックを AWS IoT の予約済みトピックに公開します。

AWS IoT デバイスクライアントを使用したメトリクスの発行

AWS IoT Device Client をインストールするには、Github からダウンロードできます。デバイス側のデータを収集するデバイスに AWS IoT Device Client をインストールしたら、デバイス側のメトリクスを AWS IoT Device Defender に送信するように設定する必要があります。AWS IoT Device Client 設定ファイルの `device-defender` セクションで次のパラメータが設定されていることを確認します。

```
"device-defender":    {
      "enabled":    true,
      "interval-in-seconds": 300
  }
```

> **Warning**
>
> 少なくとも、時間間隔を 300 秒に設定する必要があります。時間間隔を 300 秒未満に設定すると、メトリクスデータがスロットリングされることがあります。

設定を更新したら、AWS IoT Device Defender コンソールでセキュリティプロファイルと動作を作成して、デバイスがクラウドに発行するメトリクスをモニタリングできます。AWS IoT Core コンソー

ルで、[Defend] (防御)、[Detect] (検出)、[Metrics] (メトリクス) の順に選択することで、発行されたメトリクスを確認できます。

クラウド側のメトリクス

セキュリティプロファイルを作成するときに、IoT デバイスによって生成されるメトリクスの動作としきい値を設定することで、IoT デバイスの想定される動作を指定できます。以下は、AWS IoT のメトリクスであるクラウド側のメトリクスです。

メッセージサイズ (aws:message-byte-size)

メッセージのバイト数。デバイスから AWS IoT に送信される各メッセージの最大サイズと最小サイズ (バイト単位) を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: バイト

Example

```
{
  "name": "Max Message Size",
  "metric": "aws:message-byte-size",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 1024
    },
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`** を使用した例

```
{
```

```
  "name": "Large Message Size",
  "metric": "aws:message-byte-size",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Message size ML behavior",
  "metric": "aws:message-byte-size",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

3 つの連続する 5 分間に、このセキュリティプロファイルの動作を報告する他のすべてのデバイスの 90% で測定された累積サイズよりも大きい累積サイズのメッセージを送信すると、デバイスでアラームが発生します。

送信されたメッセージ (aws:num-messages-sent)

一定期間内にデバイスが送信したメッセージの数。

一定期間に AWS IoT と各デバイスの間で送信できるメッセージの最大数と最小数を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: メッセージ

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{

  "name": "Out bound message count",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 50
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
    },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`** を使用した例

```
{

  "name": "Out bound message rate",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Messages sent ML behavior",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

受信したメッセージ (aws:num-messages-received)

一定期間内にデバイスが受信したメッセージの数。

一定期間に AWS IoT と各デバイスの間で受信できるメッセージの最大数と最小数を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: メッセージ

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "In bound message count",
  "metric": "aws:num-messages-received",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 50
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
```

```
    "consecutiveDatapointsToClear": 1
    },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`** を使用した例

```
{
  "name": "In bound message rate",
  "metric": "aws:num-messages-received",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Messages received ML behavior",
  "metric": "aws:num-messages-received",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

認可の失敗 (aws:num-authorization-failures)

一定期間内に各デバイスに許容される認証エラーの最大数を指定するには、このメトリクスを使用します。デバイスが十分にアクセス許可を持たないトピックに発行しようとした場合など、デバイスから AWS IoT へのリクエストが拒否されると認証エラーが発生します。

次と互換性あり: Rules Detect | ML Detect

単位: エラー

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "Authorization Failures",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "comparisonOperator": "less-than",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`** を使用した例

```
{
  "name": "Authorization Failures",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Authorization failures ML behavior",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

送信元 IP (aws:source-ip-address)

デバイスが AWS IoTに接続したときの接続元 IP アドレス。

各デバイスが AWS IoT に接続する必要がある、または接続してはならない、許可された Classless Inter-Domain Routings (CIDR) のセット (以前はホワイトリストに登録されたとしていた) または拒否された CIDR のセット (以前はブラックリストに登録されたとしていた) を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect

演算子: in-cidr-set | not-in-cidr-set

値: CIDR のリスト

単位: 該当なし

Example

```
{
  "name": "Denied source IPs",
  "metric": "aws:source-ip-address",
  "criteria": {
    "comparisonOperator": "not-in-cidr-set",
    "value": {
      "cidrs": [ "12.8.0.0/16", "15.102.16.0/24" ]
    }
```

```
  },
  "suppressAlerts": true
}
```

接続試行 (aws:num-connection-attempts)

一定期間内にデバイスが接続を試行する回数。

各デバイスの接続試行回数の最大値または最小値を指定するには、このメトリクスを使用します。成功した場合も失敗した場合もカウントされます。

次と互換性あり: Rules Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: 接続試行回数

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "Connection Attempts",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **`statisticalThreshold`** を使用した例

```
{
  "name": "Connection Attempts",
```

```
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Connection attempts ML behavior",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": false
}
```

切断 (aws:num-disconnects)

一定期間内にデバイスが AWS IoT から切断される回数。

一定期間内にデバイスが AWS IoT から切断される回数の最大値または最小値を指定するには、このメトリクスを使用します。

次と互換性あり: Rules Detect | ML Detect

演算子: less-than | less-than-equals | greater-than | greater-than-equals

値: 負でない整数

単位: 切断回数

期間: 負でない整数 有効な値は、300、600、900、1800、3600 秒です。

Example

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example **statisticalThreshold** を使用した例

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example ML Detect を使用した例

```
{
  "name": "Disconnects ML behavior",
  "metric": "aws:num-disconnects",
  "criteria": {
```

```
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

切断時間 (aws:disconnect-duration)

デバイスの接続が AWS IoT から切断されたままになる時間。

このメトリクスを使用して、AWS IoT からデバイスの接続が切断されたままになる最大時間を指定します。

次と互換性あり: Rules Detect

演算子: less-than | less-than-equals

値: 負でない整数 (分単位)

Example

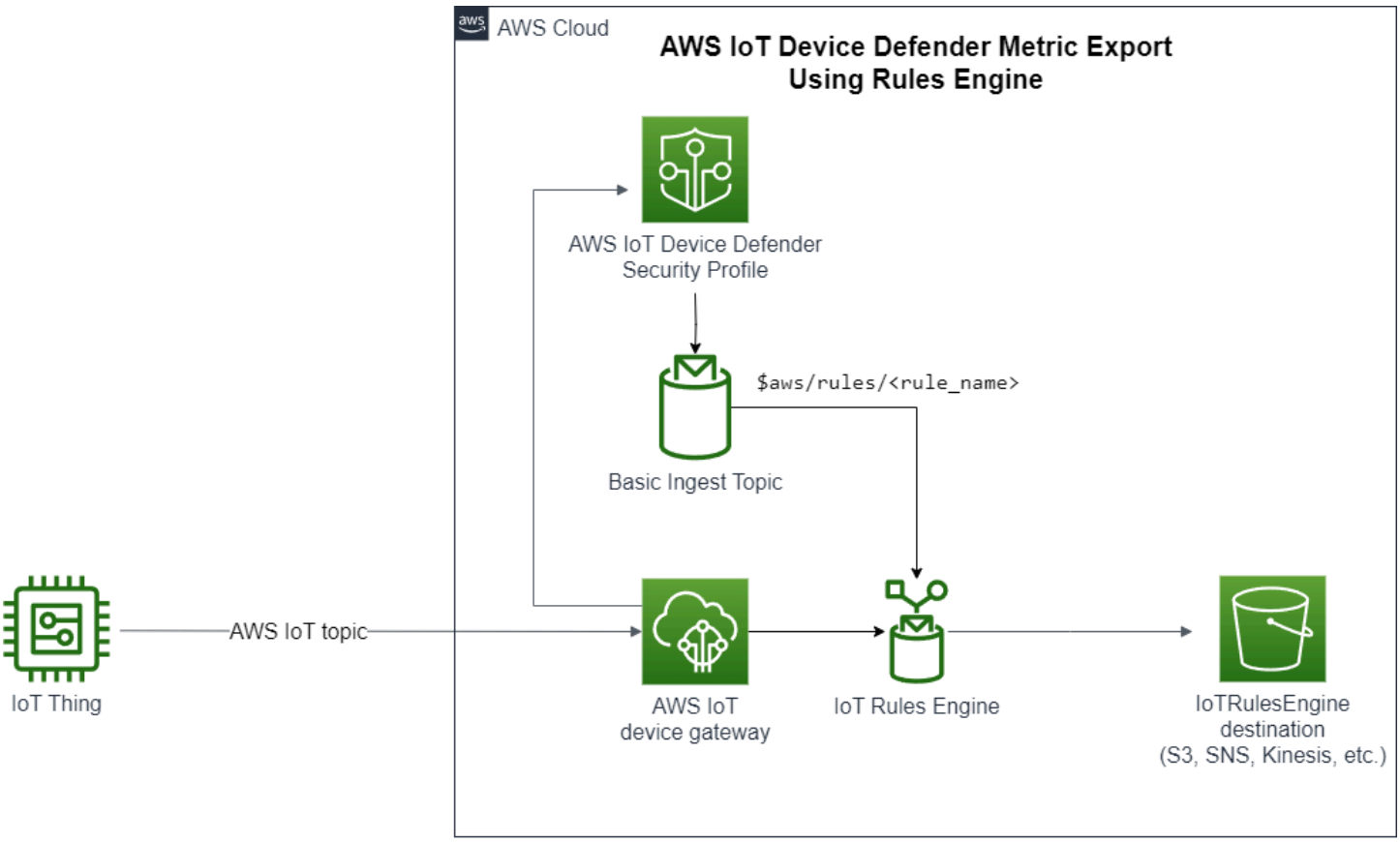
```
{
"name": "DisconnectDuration",
  "metric": "aws:disconnect-duration",
  "criteria": {
"comparisonOperator": "less-than-equals",
    "value": {
"count": 5
    }
  },
  "suppressAlerts": true
}
```

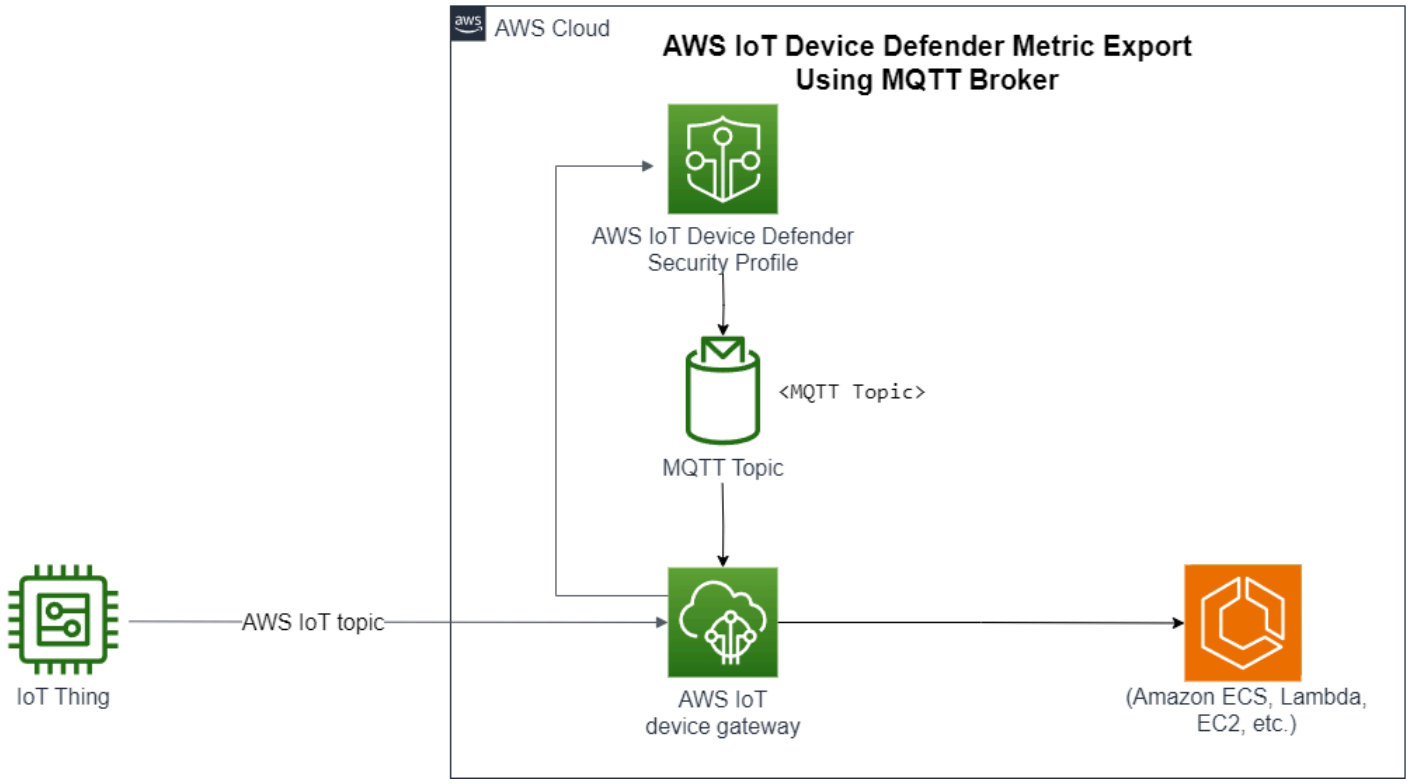
Detect メトリクスのエクスポート

メトリクスのエクスポートを使用すると、クラウド側、デバイス側、またはカスタムメトリクスを AWS IoT Device Defender からエクスポートし、設定した MQTT トピックに発行できます。この

機能により、Detect メトリクスの一括エクスポートがサポートされます。これにより、データのレポートと分析が効率化されるだけでなく、コスト管理もしやすなります。MQTT トピックは、AWS IoT ルールの基本的な取り込みトピックとして選択することも、独自の MQTT トピックを作成してサブスクライブすることもできます。AWS IoT Device Defender コンソール、API、または CLI を使用してメトリクスのエクスポートを設定します。この機能は、AWS IoT Device Defender を利用できる全 AWS リージョンで利用可能です。

次の図は、メトリクスをエクスポートするための AWS IoT Device Defender の設定方法を示しています。最初の図は、基本的な取り込みトピックでメトリクスのエクスポートを設定する方法を示しています。その後、エクスポートされたメトリクスを AWS IoT ルールでサポートされているさまざまな送信先にルーティングできます。2 番目の図は、MQTT トピックにデータを発行するための AWS IoT Device Defender 設定方法を示しています。次に、MQTT クライアントは、そのトピックにサブスクライブします。Amazon Elastic Container Service、Lambda、または同じ MQTT トピックにサブスクライブする Amazon EC2 インスタンスのコンテナで MQTT クライアントを実行できます。AWS IoT Device Defender がデータを発行するたびに、MQTT クライアントはデータを受信して処理します。詳細については、「MQTT のトピック」を参照してください。


AWS Cloud
AWS IoT Device Defender Metric Export
Using Rules Engine
AWS IoT Device Defender
Security Profile
$aws/rules/<rule_name>
Basic Ingest Topic
IoT Thing
AWS IoT topic
AWS IoT
device gateway
IoT Rules Engine
IoTRulesEngine
destination
(S3, SNS, Kinesis, etc.)


AWS Cloud
AWS IoT Device Defender Metric Export
Using MQTT Broker
AWS IoT Device Defender
Security Profile
<MQTT Topic>
MQTT Topic
IoT Thing
AWS IoT topic
AWS IoT
device gateway
(Amazon ECS, Lambda,
EC2, etc.)

Detect メトリクスのエクスポートの仕組み

セキュリティプロファイルを設定するときは、エクスポートするメトリクスを選択し、MQTT トピックを指定します。また、設定された MQTT トピックにメッセージを発行するために必要なアクセス許可を AWS IoT Device Defender Detect に付与する IAM ロールを設定します。AWS IoT ルールの基本取り込み MQTT トピックを設定し、エクスポートされたメトリクスを AWS IoT ルールがサポートする送信先に送信できます。AWS IoT ルールの設定と構成の手順については、「AWS IoT デベロッパーガイド」の「AWS IoT のルール」を参照してください。

AWS IoT Device Defender Detect は、設定された各メトリクスのメトリクス値をバッチ処理し、設定された MQTT トピックに定期的に発行します。メッセージバイトサイズと合計バイトサイズを除き、クラウド側のメトリクスはバッチ処理期間のメトリクス値を合計して集計されます。カスタムメトリクスおよびデバイス側のメトリクスは集計されません。メッセージバイトサイズの場合、エクスポート値はバッチ処理期間の最小、最大、合計バイトサイズです。切断時間の場合、エクスポート値はすべての追跡されたデバイスの切断時間を秒単位で表したものです。これは 1 時間ごとに、また、接続イベントまたは切断イベントがある都度、実施されます。接続されたデバイスまたは接続イベントの場合、値は 0 です。クラウド側のメトリクス、デバイス側のメトリクス、カスタムメトリクスの詳細については、「AWS IoT Device Defender デベロッパーガイド」の以下のトピックを参照してください。

- カスタムメトリクス
- クラウド側のメトリクス
- デバイス側のメトリクス

バッチ処理されたメトリクスは、AWS IoT ルールを使用して異なる送信先にエクスポートできます。サポートされている送信先のリストについては、「AWS IoT ルールアクション」を参照してください。バッチ処理されたエクスポートメッセージ内の個々のメトリクスを、サポートされている送信先に送信するには、AWS IoT ルールアクションに batchMode オプションを使用します。希望する AWS IoT ルールの送信先に `batchMode` サポートがない場合は、Lambda や Kinesis データストリームなどの中間アクションを使用して、バッチ処理されたメッセージ内で個々のメトリクスを送信できます。

メトリクスのエクスポートスキーマ

バッチ処理されたメトリクスのエクスポートデータについては、次のスキーマを参照してください。

```
{
```

```
 "version": "1.0",
 "metrics": [
 {
 "name": "{metricName}",
 "thing": "{thingName}",
 "value": {
 # a list of Classless Inter-Domain Routings (CIDR) specifying metric
# source-ip-address and destination-ip-address
 "cidrs": ["string"],
 # a single metric value for cloud/device metrics
 "count": number,
 # a single metric value for custom metric
 "number": number,
 # a list of numbers for custom metrics
 "numbers": [number],
 # a list of ports for cloud/device metrics
 "ports": [number],
 # a list of strings for custom metrics
 "strings": ["string"]
 },
 # In some rare cases we may send multiple values for the same thing, metric and
 timestamp.
 # When there are multiple values, please use the value with highest version number
 # and discard other values.
 "version": number,
 # For cloud-side metrics, this is the time when AWS IoT Device Defender Detect
 aggregates the
 # metrics data received from AWS IoT.
 # For device-side and custom metrics, this is the time at which the metrics data
 # is reported by the devices.
 "timestamp": number,
 # The dimension parameters are optional. It's set only if
 # the metrics are configured with a dimension in the security profile.
 "dimension": {
 "name": "{dimensionName}",
 "operator": "{dimensionOperator}"
 }
 }
 ]
}
```

Detect メトリクスのエクスポート料金

設定した MQTT トピックにクラウド側のメトリクス、デバイス側のメトリクス、またはカスタムメトリクスを発行しても、エクスポートプロセスのこのステップでは料金は発生しません。ただし、発行済みメトリクスを任意の送信先に転送する以降のステップでは、ルールエンジンまたはメッセージングを使用することにより、選択した転送方法に基づいてコストが発生します。AWS IoT Device Defender は、複数のデバイスのメトリクスデータを含む 1 つのメッセージとしてバッチ処理されたメトリクスを MQTT トピックに発行するため、コスト管理に役立ちます。料金の詳細については、「AWS 料金計算ツール」を参照してください。

アクセス許可

このセクションでは、AWS IoT Device Defender Detect メトリクスのエクスポートを管理するために必要な IAM ロールとポリシーをセットアップする方法について説明します。詳細については、IAM ユーザーガイドを参照してください。

MQTT トピックに対してメッセージを発行するアクセス許可を AWS IoT Device Defender Detect に付与します。

CreateSecurityProfile でメトリクスのエクスポートを有効にする場合、2 つのポリシー (アクセス許可ポリシーと信頼ポリシー) を持つ IAM ロールを指定する必要があります。アクセス許可ポリシーは、メトリクスを含むメッセージを MQTT トピックに発行するアクセス許可を AWS IoT Device Defender に付与します。信頼ポリシーは、必要なロールを引き受けるアクセス許可を AWS IoT Device Defender に付与します。

アクセス許可ポリシー

```
{
  "Version":"2012-10-17",
  "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "iot:Publish"
        ],
        "Resource":[
          "arn:aws:iot:region:account-id:topic/your-topic-name"
        ]
    }
  ]
```

```
}
```

信頼ポリシー

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Pass ロールポリシー

そのほかに、IAM のアクセス許可ポリシーを IAM ユーザーにアタッチして、ユーザーがロールを渡せるようにする必要もあります。「AWS サービスにロールを渡すアクセス許可をユーザーに付与する」を参照してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
          "iam:GetRole",
          "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/Role_To_Pass"
    }
  ]
}
```

AWS IoT コンソールで Detect メトリクスのエクスポートを設定する

コンソールでメトリクスのエクスポートを含む新しいセキュリティプロファイルを作成、表示、編集します。

前提条件

Detect メトリクスのエクスポートを設定する前に、次の前提条件を満たしていることを確認します。

- IAM ロール。IAM ロールの作成方法の詳細については、「IAM ユーザーガイド」の「IAM ロールを作成する」を参照してください。
- 正しいアクセス許可を持つ AWS Identity and Access Management (IAM) ユーザーとしてサインインできる AWS アカウント。AWS IoT Device Defender Detect のアクセス許可の詳細については、「AWS IoT Core デベロッパーガイド」の「アクセス許可」を参照してください。

メトリクスのエクスポートを含む新しいセキュリティプロファイルの作成 (コンソール)

メトリクス動作データをエクスポートするには、まずメトリクスのエクスポートを含むためのセキュリティプロファイルを設定します。次の手順では、Detect メトリクスのエクスポートを含むルールベースのセキュリティプロファイルを設定する方法について詳しく説明します。

メトリクスのエクスポートを含む新しいセキュリティプロファイルを作成するには

1. AWS IoT コンソールを開きます。ナビゲーションバーで、[セキュリティ]、[検出]、[セキュリティプロファイル] の順に展開します。
2. [セキュリティプロファイルを作成] で、[ルールに基づいた異常検出プロファイルを作成] を選択します。
3. セキュリティプロファイルのプロパティを指定するには、[セキュリティプロファイル名] を入力し、[ターゲット] で、異常のターゲットになるデバイスのグループを選択します。(オプション) AWS リソースにラベルを付ける説明とタグを含めます。[Next] を選択します。
4. [メトリクス] で、デバイスの動作を定義するメトリクスを選択します。デバイスが目的の動作を満たさない場合に警告する動作のしきい値を定義できます。
5. 動作異常のアラートを受信するには、[アラートを送信 (メトリクスの動作を定義)] を選択し、[動作名] と [条件] を指定します。アラートなしでメトリクスを保持するには、[アラートを送信しない (リテンションメトリクス)] を選択します。[Next] を選択します。

6. メトリクスのエクスポートを設定するには、[メトリクスのエクスポートをオンにする] を選択します。
7. メトリクスデータを AWS IoT Core に発行するための MQTT トピック名を入力します。IAM ロールを選択して、設定されたトピックにメッセージを発行するアクセス許可「AWS IoT:Publish」を AWS IoT に付与します。エクスポートするメトリクスを選択し、[次へ] を選択します。

> **Note**
>
> MQTT トピック名を入力するときに、スラッシュを使用して階層情報を表します。例えば、`$AWS/rules/rule-name/` と指定します。

8. デバイスが、設定された動作に違反したときに AWS コンソールにアラートを送信するには、Amazon SNS トピックと IAM ロールを選択または作成します。[Next] を選択します。
9. 構成を確認し、[次へ] を選択します。

セキュリティプロファイルの詳細の表示と編集 (コンソール)

セキュリティプロファイルの詳細を表示および編集するには

1. AWS IoT コンソールを開きます。ナビゲーションバーで、[セキュリティ]、[検出]、[セキュリティプロファイル] の順に展開します。
2. メトリクスのエクスポートを含めるために作成したセキュリティプロファイルを選択し、[アクション] で [編集] を選択します。
3. [ターゲット] で、編集するターゲットデバイスグループを選択し、[次へ] を選択します。
4. メトリクスの動作設定を編集するには、[アラートを出す (メトリクスの動作を定義)] を選択し、メトリクスの動作が満たされたときの条件を定義します。[Next] を選択します。
5. メトリクスのエクスポート設定をオフにするには、[メトリクスのエクスポートをオフにする] を選択します。[Next] を選択します。
6. デバイスが、設定された動作に違反したときに AWS IoT コンソールにアラートを送信するように Amazon SNS を設定するには、Amazon SNS トピックと IAM ロールを選択または作成します。[Next] を選択します。
7. 設定を確認し、[次へ] を選択します。

メトリクスのエクスポートを有効にするセキュリティプロファイルの作成

`create-security-profile` コマンドを使用してセキュリティプロファイルを作成し、メトリクスのエクスポートを有効にします。

メトリクスのエクスポートを含むセキュリティプロファイルを作成するには

1. メトリクスのエクスポートを有効にして、Detect が、対応するメトリクスをエクスポートする必要があるかどうかを示すには、`Behavior` と `AdditionalMetricsToRetainV2` の両方で値 `exportMetric` を true に設定します。
2. `MetricsExportConfig` の値を含めます。これは、メトリクスのエクスポートに必要な MQTT トピック Amazon リソースネーム (ARN) を指定します。

> **Note**
>
> AWS IoT Device Defender Detect がメッセージを発行できるように `mqttTopic` を含めます。ロール ARN には MQTT メッセージを公開するアクセス許可があり、その後 AWS IoT Device Defender Detect がそのロールを引き受け、ユーザーに代わってメッセージを公開できます。

```
aws iot create-security-profile \
    --security-profile-name CreateSecurityProfileWithMetricsExport \
    --security-profile-description "create security profile with metrics export
 enabled"  \
    --behaviors "[{\"name\":\"BehaviorNumAuthz\",\"metric\":\"aws:num-authorization-
failures\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count
\":5}, \"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1,
\"durationSeconds\":300},\"exportMetric\":true}]" \
    --metrics-export-config "{\"mqttTopic\":\"\$aws/rules/metricsExportRule\",\"roleArn
\":\"arn:aws:iam::123456789012:role/iot-test-role\"}" \
    --region us-east-1
```

出力:

```
{
    "securityProfileName": "CreateSecurityProfileWithMetricsExport",
    "securityProfileArn": "arn:aws:iot:us-east-1:123456789012:securityprofile/
CreateSecurityProfileWithMetricsExport"
```

```
}
```

メトリクスのエクスポートを有効にするためのセキュリティプロファイルの更新 (CLI)

update-security-profile コマンドを使用して既存のセキュリティプロファイルを更新し、メトリクスのエクスポートを有効にします。

セキュリティプロファイルを更新してメトリクスのエクスポートを有効にするには

1. メトリクスのエクスポートを有効にして、Detect が、対応するメトリクスをエクスポートする必要があるかどうかを示すには、Behavior と AdditionalMetricsToRetainV2 の両方で値 exportMetric を true に設定します。
2. MetricsExportConfig の値を含めます。これは、メトリクスのエクスポートに必要な MQTT トピック Amazon リソースネーム (ARN) を指定します。

> **Note**
>
> AWS IoT Device Defender Detect がメッセージを発行できるように mqttTopic を含めます。ロール ARN には MQTT メッセージを公開するアクセス許可があり、その後 AWS IoT Device Defender Detect がそのロールを引き受け、ユーザーに代わってメッセージを公開できます。

```
aws iot update-security-profile \
    --security-profile-name UpdateSecurityProfileWithMetricsExport \
    --security-profile-description "update an existing security profile to enable
 metrics export"  \
    --behaviors "[{\"name\":\"BehaviorNumAuthz\",\"metric\":\"aws:num-authorization-
failures\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count
\":5}, \"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1,
\"durationSeconds\":300},\"exportMetric\":true}]" \
    --metrics-export-config "{\"mqttTopic\":\"\$aws/rules/metricsExportRule\",\"roleArn
\":\"arn:aws:iam::123456789012:role/iot-test-role\"}" \
    --region us-east-1
```

出力:

```
{
```

```
    "securityProfileName": "UpdateSecurityProfileWithMetricsExport",
    "securityProfileArn": "arn:aws:iot:us-east-1:123456789012:securityprofile/
UpdateSecurityProfileWithMetricsExport",
    "securityProfileDescription": "update an existing security profile to enable
 metrics export",
    "behaviors": [
        {
            "name": "BehaviorNumAuthz",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "comparisonOperator": "less-than",
                "value": {
                    "count": 5
                },
                "durationSeconds": 300,
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            },
            "exportMetric": true
        }
    ],
    "version": 2,
    "creationDate": "2023-11-09T16:18:37.183000-08:00",
    "lastModifiedDate": "2023-11-09T16:20:15.486000-08:00",
    "metricsExportConfig": {
        "mqttTopic": "$aws/rules/metricsExportRule",
        "roleArn": "arn:aws:iam::123456789012:role/iot-test-role"
    }
}
```

セキュリティプロファイルを更新してメトリクスのエクスポートをオフにする (CLI)

update-security-profile コマンドを使用して既存のセキュリティプロファイルを更新し、メトリクスのエクスポートをオフにします。

セキュリティプロファイルを更新してメトリクスのエクスポートをオフにするには

- セキュリティプロファイルを更新し、メトリクスのエクスポート設定を削除するには、コマンド --delete-metrics-export-config を使用します。

```
aws iot update-security-profile \
    --security-profile-name UpdateSecurityProfileToDisableMetricsExport \
    --security-profile-description "update an existing security profile to disable
 metrics export"  \
    --behaviors "[{\"name\":\"BehaviorNumAuthz\",\"metric\":\"aws:num-authorization-
failures\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count
\":5}, \"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1,
\"durationSeconds\":300}}]" \
    --delete-metrics-export-config \
    --region us-east-1
```

出力:

```
{
    "securityProfileName": "UpdateSecurityProfileToDisableMetricsExport",
    "securityProfileArn": "arn:aws:iot:us-east-1:123456789012:securityprofile/
UpdateSecurityProfileWithMetricsExport",
    "securityProfileDescription": "update an existing security profile to disable
 metrics export",
    "behaviors": [
        {
            "name": "BehaviorNumAuthz",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "comparisonOperator": "less-than",
                "value": {
                    "count": 5
                },
                "durationSeconds": 300,
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        }
    ],
    "version": 2,
    "creationDate": "2023-11-09T16:18:37.183000-08:00",
    "lastModifiedDate": "2023-11-09T16:31:16.265000-08:00"
}
```

詳細については、「AWS IoT デベロッパーガイド」の「Detect コマンド」を参照してください。

メトリクスのエクスポート CLI コマンド

以下の CLI コマンドを使用して Detect メトリクスのエクスポートを作成および管理できます。

- CreateSecurityProfile
- UpdateSecurityProfile
- DescribeSecurityProfile

メトリクスのエクスポート API オペレーション

次の API オペレーションを使用して、Detect メトリクスのエクスポートを作成および管理できます。

- CreateSecurityProfile
- UpdateSecurityProfile
- DescribeSecurityProfile

ディメンションを使用したセキュリティプロファイルでのメトリクスの範囲設定

ディメンションは、メトリクスと動作に関するより正確なデータを取得するためにセキュリティプロファイルで定義できる属性です。範囲を定義するには、フィルタとして使用する値またはパターンを指定します。たとえば、「data/bulb/+/activity」など、特定の値と一致する MQTT トピックにのみメトリクスを適用するトピックフィルタディメンションを定義できます。セキュリティプロファイルで使用できるディメンションの定義については、CreateDimension を参照してください。

ディメンション値は、MQTT ワイルドカードをサポートしています。MQTT ワイルドカードを使用すると、複数のトピックに同時にサブスクライブできます。ワイルドカードには、シングルレベル (+) とマルチレベル (#) の 2 種類があります。たとえば、ディメンション値 `Data/bulb/+/activity` によって、+ と同じレベルに存在するすべてのトピックに一致するサブスクリプションが作成されます。ディメンション値は、MQTT クライアント ID 代替変数 ${iot:ClientId} もサポートしています。

TOPIC_FILTER タイプのディメンションは、次の一連のクラウド側メトリクスと互換性があります。

- 認証エラーの数
- メッセージのサイズ (バイト)
- 受信したメッセージの数
- 送信されたメッセージの数
- 送信元 IP アドレス (Rules Detect でのみ使用可能)

コンソールでディメンションを使用する方法

ディメンションを作成してセキュリティプロファイルの動作に適用するには

1. AWS IoT コンソールを開きます。ナビゲーションペインで、[セキュリティ]、[検出] の順に展開し、[セキュリティプロファイル] を選択します。
2. [セキュリティプロファイル] ページで、[セキュリティプロファイルを作成]、[ルールに基づいた異常検出プロファイルを作成] の順に選択します。または、既存のルールベースのセキュリティプロファイルにディメンションを適用するには、セキュリティプロファイルを選択し、[編集] を選択します。
3. [セキュリティプロファイルのプロパティを指定する] ページで、セキュリティプロファイルの名前を入力します。
4. 異常についてターゲットにするデバイスのグループを選択します。
5. [Next] を選択します。
6. [メトリクス動作の設定] ページの [メトリクスタイプ] で、クラウド側のメトリクスディメンションのいずれかを選択します。
7. [メトリクスの動作] で [アラートを送信 (メトリクスの動作を定義)] を選択して、予想されるメトリクスの動作を定義します。
8. デバイスの異常な動作に関するアラートをいつ受信するかを選択します。
9. [Next] を選択します。
10. セキュリティプロファイルの設定を確認し、[作成] を選択します。

アラームを表示するには

1. AWS IoT コンソールを開きます。ナビゲーションペインで、[セキュリティ]、[検出] の順に展開し、[アラーム] を選択します。
2. [モノの名前] 列でモノを選択すると、アラームの原因に関する情報が表示されます。

ディメンションを表示および更新するには

1. AWS IoT コンソールを開きます。ナビゲーションペインで、[セキュリティ]、[検出] の順に展開し、[ディメンション] を選択します。
2. ディメンションを選択してから、[編集] を選択します。
3. ディメンションを編集してから、[更新] を選択します。

ディメンションを削除するには

1. AWS IoT コンソールを開きます。ナビゲーションペインで、[セキュリティ]、[検出] の順に展開し、[ディメンション] を選択します。
2. ディメンションを削除する前に、そのディメンションを参照するメトリクス動作を削除する必要があります。[セキュリティプロファイル] 列をチェックして、ディメンションがセキュリティプロファイルにアタッチされていないことを確認します。ディメンションがセキュリティプロファイルにアタッチされている場合、左側の [セキュリティプロファイル] ページを開き、ディメンションがアタッチされているセキュリティプロファイルを編集します。次に、動作の削除を続行できます。別のディメンションを削除する場合は、このセクションのステップに従います。
3. ディメンションを選択してから、[削除] を選択します。
4. ディメンション名を入力して確認し、[削除] を選択します。

AWS CLI でディメンションを使用する方法

ディメンションを作成してセキュリティプロファイルの動作に適用するには

1. セキュリティプロファイルにアタッチする前に、まずディメンションを作成します。CreateDimension コマンドを使用してディメンションを作成します。

```
aws iot create-dimension \
  --name TopicFilterForAuthMessages \
  --type TOPIC_FILTER \
  --string-values device/+/auth
```

このコマンドの出力は以下のようになります。

```
{
    "arn": "arn:aws:iot:us-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
```

```
    "name": "TopicFilterForAuthMessages"
}
```

2. UpdateSecurityProfile を使用して既存のセキュリティプロファイルにディメンションを追加するか、CreateSecurityProfile を使用して新しいセキュリティプロファイルにディメンションを追加します。次の例では、TopicFilterForAuthMessages へのメッセージが 128 バイト未満かどうかをチェックし、非認証トピックに送信されるメッセージ数を保持する新しいセキュリティプロファイルを作成します。

```
aws iot create-security-profile \
  --security-profile-name ProfileForConnectedDevice \
  --security-profile-description "Check to see if messages to
 TopicFilterForAuthMessages are under 128 bytes and retains the number of messages
 sent to non-auth topics." \
  --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":10},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]" \
  --additional-metrics-to-retain-v2 "[{\"metric\": \"aws:num-authorization-failures
\",\"metricDimension\": {\"dimensionName\": \"TopicFilterForAuthMessages\",
\"operator\": \"NOT_IN\"}}]"
```

このコマンドの出力は以下のようになります。

```
{
    "securityProfileArn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice",
    "securityProfileName": "ProfileForConnectedDevice"
}
```

時間を節約するため、パラメータをコマンドラインパラメータ値として入力する代わりに、ファイルからロードすることもできます。詳細については、「ファイルから AWS CLI パラメータをロードする」を参照してください。次に、拡張された JSON 形式の behavior パラメータを示します。

```
[
  {
    "criteria": {
      "comparisonOperator": "less-than",
```

```
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "value": {
        "count": 128
      }
    },
    "metric": "aws:message-byte-size",
    "metricDimension": {
      "dimensionName:": "TopicFilterForAuthMessages"
    },
    "name": "CellularBandwidth"
  }
]
```

または、次の例のように ML でディメンションを使用して、CreateSecurityProfile を使用します。

```
aws iot create-security-profile --security-profile-name ProfileForConnectedDeviceML
 \
   --security-profile-description “Check to see if messages to
 TopicFilterForAuthMessages are abnormal”  \
   --behaviors “[{\“name\“:\“test1\“,\“metric\“:\“aws:message-byte-size\“,
\“metricDimension\“:{\“dimensionName\“: \“TopicFilterForAuthMessages\“,\“operator
\“: \“IN\“},\“criteria\“:{\“mlDetectionConfig\“:{\“confidenceLevel\“:\“HIGH\“},
\“consecutiveDatapointsToAlarm\“:1,\“consecutiveDatapointsToClear\“:1}}]” \
   --region us-west-2
```

ディメンションがアタッチされたセキュリティプロファイルを表示するには

- 特定のディメンションがアタッチされたセキュリティプロファイルを表示するには、ListSecurityProfiles コマンドを使用します。

```
aws iot list-security-profiles \
  --dimension-name TopicFilterForAuthMessages
```

このコマンドの出力は以下のようになります。

```
{
    "securityProfileIdentifiers": [
        {
```

```
            "name": "ProfileForConnectedDevice",
            "arn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice"
        }
    ]
}
```

ディメンションを更新するには

- UpdateDimension コマンドを使用してディメンションを更新します。

```
aws iot update-dimension \
  --name TopicFilterForAuthMessages \
  --string-values device/${iot:ClientId}/auth
```

このコマンドの出力は以下のようになります。

```
{
    "name": "TopicFilterForAuthMessages",
    "lastModifiedDate": 1585866222.317,
    "stringValues": [
        "device/${iot:ClientId}/auth"
    ],
    "creationDate": 1585854500.474,
    "type": "TOPIC_FILTER",
    "arn": "arn:aws:iot:us-west-2:1234564789012:dimension/
TopicFilterForAuthMessages"
}
```

ディメンションを削除するには

1. ディメンションを削除するには、まず、ディメンションがアタッチされているセキュリティプロファイルからディメンションをデタッチします。特定のディメンションがアタッチされているセキュリティプロファイルを表示するには、ListSecurityProfiles コマンドを使用します。
2. セキュリティプロファイルからディメンションを削除するには、UpdateSecurityProfile コマンドを使用します。保持するすべての情報を入力します。ただしディメンションは除外します。

```
aws iot update-security-profile \
  --security-profile-name ProfileForConnectedDevice \
```

```
  --security-profile-description "Check to see if authorization fails 10 times in 5
 minutes or if cellular bandwidth exceeds 128" \
  --behaviors "[{\"name\":\"metric\":\"aws:message-byte-size\",\"criteria
\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\comparisonOperator\":\"less-than\",\"value\"{\"count\":10},\"durationSeconds
\":300,\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

このコマンドの出力は以下のようになります。

```
{
  "behaviors": [
    {
      "metric": "aws:message-byte-size",
      "name": "CellularBandwidth",
      "criteria": {
        "consecutiveDatapointsToClear": 1,
        "comparisonOperator": "less-than",
        "consecutiveDatapointsToAlarm": 1,
        "value": {
          "count": 128
        }
      }
    },
    {
      "metric": "aws:num-authorization-failures",
      "name": "Authorization",
      "criteria": {
        "durationSeconds": 300,
        "comparisonOperator": "less-than",
        "consecutiveDatapointsToClear": 1,
        "consecutiveDatapointsToAlarm": 1,
        "value": {
          "count": 10
        }
      }
    }
  ],
  "securityProfileName": "ProfileForConnectedDevice",
  "lastModifiedDate": 1585936349.12,
  "securityProfileDescription": "Check to see if authorization fails 10 times in 5
 minutes or if cellular bandwidth exceeds 128",
```

```
  "version": 2,
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Preo/
ProfileForConnectedDevice",
  "creationDate": 1585846909.127
}
```

3. ディメンションをデタッチした後、DeleteDimension コマンドを使用してディメンションを削除します。

```
aws iot delete-dimension \
  --name TopicFilterForAuthMessages
```

アクセス許可

このセクションでは、AWS IoT Device Defender Detect を管理するために必要な IAM ロールとポリシーをセットアップする方法について説明します。詳細については、IAM ユーザーガイドを参照してください。

AWS IoT Device Defender Detect に、SNS トピックに対してアラームを発行するアクセス許可を付与します。

CreateSecurityProfile で `alertTargets` パラメータを使用する場合、2 つのポリシー (アクセス許可ポリシーと信頼ポリシー) を持つ IAM ロールを指定する必要があります。アクセス許可ポリシーは、SNS トピックに通知を発行するアクセス許可を AWS IoT Device Defender に付与します。信頼ポリシーは、必要なロールを引き受けるアクセス許可を AWS IoT Device Defender に付与します。

アクセス許可ポリシー

```
{
  "Version":"2012-10-17",
  "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "sns:Publish"
        ],
        "Resource":[
          "arn:aws:sns:region:account-id:your-topic-name"
        ]
```

```
    }
  ]
}
```

信頼ポリシー

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Pass ロールポリシー

そのほかに、IAM のアクセス許可ポリシーを IAM ユーザーにアタッチして、ユーザーがロールを渡せるようにする必要もあります。「AWS サービスにロールを渡すアクセス許可をユーザーに付与する」を参照してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
          "iam:GetRole",
          "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/Role_To_Pass"
    }
  ]
}
```

検出コマンド

このセクションの Detect コマンドを使用して、ML Detect または Rules Detect セキュリティプロファイルを設定し、デバイスが侵害されたことを示唆している可能性のある異常な動作を特定および監視できます。

DetectMitigation アクションコマンド

Detect 実行を開始および管理する
CancelDetectMitigationActionsTask
DescribeDetectMitigationActionsTask
ListDetectMitigationActionsTasks
StartDetectMitigationActionsTask
ListDetectMitigationActionsExecutions

ディメンションアクションコマンド

ディメンション実行を開始および管理する
CreateDimension
DescribeDimension
ListDimensions
DeleteDimension
UpdateDimension

CustomMetric アクションコマンド

CustomMetric の実行を開始および管理する
CreateCustomMetric

CustomMetric の実行を開始および管理する
UpdateCustomMetric
DescribeCustomMetric
ListCustomMetrics
DeleteCustomMetric

セキュリティプロファイルアクションコマンド

セキュリティプロファイルの実行を開始および管理する
CreateSecurityProfile
AttachSecurityProfile
DetachSecurityProfile
DeleteSecurityProfile
DescribeSecurityProfile
ListTargetsForSecurityProfile
UpdateSecurityProfile
ValidateSecurityProfileBehaviors
ListSecurityProfilesForTarget

アラームアクションコマンド

アラームとターゲットを管理する
ListActiveViolations
ListViolationEvents

アラームとターゲットを管理する
PutVerificationStateon 違反

ML Detect アクションコマンド

ML モデルトレーニングデータを一覧表示する
GetBehaviorModelTrainingSummaries

AWS IoT Device Defender Detect を使用する方法

1. クラウド側メトリクスだけに AWS IoT Device Defender Detect を使用できますが、デバイスでレポートされたメトリクスを使用する予定の場合、まず AWS IoT によって接続されたデバイスまたはデバイスゲートウェイに AWS IoT SDK をデプロイする必要があります。詳細については、「デバイスからのメトリクスの送信」を参照してください。
2. 動作を定義してアラームを作成する前に、デバイスが生成するメトリクスを表示することを検討してください。AWS IoT では、デバイスから測定基準を収集することができるため、デバイスのグループ、またはアカウント内のすべてのデバイスに対する通常の動作または異常な動作を最初に識別することができます。CreateSecurityProfile を使用しますが、目的の `additionalMetricsToRetain` のみ指定してください。この時点では `behaviors` を指定しないでください。

   デバイスの一般的な動作を構成する内容を確認するには、AWS IoT コンソールを使用してデバイスメトリクスを探します。
3. セキュリティプロファイルの一連の動作を作成します。動作には、デバイスのグループまたはアカウント内のすべてのデバイスの正常な動作を指定するメトリクスが含まれています。詳細な説明と例については、「クラウド側のメトリクス」および「デバイス側のメトリクス」を参照してください。一連の動作を作成したら、ValidateSecurityProfileBehaviors を使用して検証できます。
4. 動作が含まれるセキュリティプロファイルを作成するには、CreateSecurityProfile アクションを使用します。デバイスが動作に違反した場合にアラームをターゲット (SNS トピック) に送信するには、`alertTargets` パラメータを使用します。(SNS を使用してアラームを送信する場合、これらは AWS アカウント の SNS トピッククォータに対してカウントされる点に注意してください。違反が大幅に増加すると、SNS トピッククォータを超過する可能性があり

ます。CloudWatch メトリクスを使用して違反を検出することもできます。詳細については、「AWS IoT Core デベロッパーガイド」の「Amazon CloudWatch を使用した AWS IoT アラームとメトリクスのモニタリング」を参照してください。

5. AttachSecurityProfile アクションを使用して、セキュリティプロファイルをデバイスのグループ (モノのグループ)、アカウント内の登録済みのすべてのモノ、未登録のすべてのモノ、またはすべてのデバイスにアタッチします。AWS IoT Device DefenderDetect は異常な動作のチェックを開始し、何らかの動作違反が検出された場合はアラームを送信します。たとえば、アカウントのモノレジストリに登録されていないモバイルデバイスとやり取りする場合は、セキュリティプロファイルを未登録のすべてのモノにアタッチすることをお勧めします。ニーズに合わせて、デバイスのグループごとに異なる動作セットを定義できます。

   デバイスのグループにセキュリティプロファイルをアタッチするには、そのデバイスのグループが含まれるモノのグループの ARN を指定する必要があります。モノのグループの ARN の形式を以下に示します。

   ```
   arn:aws:iot:region:account-id:thinggroup/thing-group-name
   ```

   セキュリティプロファイルを AWS アカウント 内の登録済みのすべてのモノにアタッチするには (未登録のモノを無視)、次の形式を使用して ARN を指定する必要があります。

   ```
   arn:aws:iot:region:account-id:all/registered-things
   ```

   セキュリティプロファイルを未登録のモノにアタッチするには、次の形式を使用して ARN を指定する必要があります。

   ```
   arn:aws:iot:region:account-id:all/unregistered-things
   ```

   セキュリティプロファイルをデバイスにアタッチするには、次の形式を使用して ARN を指定する必要があります。

   ```
   arn:aws:iot:region:account-id:all/things
   ```

6. ListActiveViolations アクションを使用して違反を追跡することもできます。これにより、特定のセキュリティプロファイルまたはターゲットデバイスで検出された違反を確認できるようになります。

指定した期間中に検出された違反を確認するために、ListViolationEvents アクションを使用します。これらの結果は、セキュリティプロファイルまたはデバイスによってフィルタリングすることができます。

7. アラームの検証状態をマークし、その検証状態の説明を入力し、PutVerificationStateOOnViolationaction を使う事で、アラームの検証、整理、管理を行う事ができます。
8. デバイスが定義された動作に違反する頻度が高すぎる、またはそれほど頻繁にならない場合は、動作定義を微調整する必要があります。
9. 設定したセキュリティプロファイルと監視対象のデバイスを確認するには、ListSecurityProfiles、ListSecurityProfilesForTarget、および ListTargetsForSecurityProfile アクションを使用します。

   セキュリティプロファイルに関する詳細を取得するには、DescribeSecurityProfile アクションを使用します。
10. セキュリティプロファイルを更新するには、UpdateSecurityProfile アクションを使用します。アカウントまたはターゲットのモノのグループからセキュリティプロファイルをデタッチするには、DetachSecurityProfile アクションを使用します。セキュリティプロファイル全体を削除するには、DeleteSecurityProfile アクションを使用します。

緩和アクション

AWS IoT Device Defender を使用して、監査所見または Detect アラームで検出された問題を緩和するためのアクションを実行できます。

> **Note**
>
> 抑制された監査所見に対しては、緩和アクションは実行されません。監査所見の抑制の詳細については、「監査の所見の抑制」を参照してください。

緩和アクションの監査

AWS IoT Device Defender では、さまざまな監査チェック用に事前定義されたアクションが提供されます。これらのアクションを AWS アカウント に設定し、一連の結果に適用します。これらの結果は以下のとおりです。

- 監査からのすべての結果。このオプションは、AWS IoT コンソールでも AWS CLI の使用によっても利用できます。
- 個々の結果のリスト。このオプションは、AWS CLI の使用によってのみ利用できます。
- 監査からフィルタリングされた一連の結果。

次の表は、監査チェックのタイプと、それぞれでサポートされている緩和アクションの一覧です。

監査チェックから緩和アクションへのマッピング

監査チェック	サポートされている緩和アクション
REVOKED_CA_CERT_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
INTERMEDIATE_CA_REVOKED_FOR_ACTIVE_DEVICE_CERTIFICATES_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP

監査チェック	サポートされている緩和アクション
DEVICE_CERTIFICATE_SHARED_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS
AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS
IOT_POLICY_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS, REPLACE_DEFAULT_POLICY_VERSION
IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK	PUBLISH_FINDING_TO_SNS, REPLACE_DEFAULT_POLICY_VERSION
CA_CERTIFICATE_EXPIRING_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
CONFLICTING_CLIENT_IDS_CHECK	PUBLISH_FINDING_TO_SNS
DEVICE_CERTIFICATE_EXPIRING_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
LOGGING_DISABLED_CHECK	PUBLISH_FINDING_TO_SNS, ENABLE_IOT_LOGGING
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP

監査チェック	サポートされている緩和アクション
CA_CERTIFICATE_KEY_QUALITY_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS
IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK	PUBLISH_FINDING_TO_SNS

すべての監査チェックでは、Amazon SNS への監査結果の発行がサポートされるため、通知に応じてカスタムアクションを実行できます。各タイプの監査チェックでは、追加の緩和アクションをサポートできます。

REVOKED_CA_CERT_CHECK

- 証明書の状態を変更して、AWS IoT で非アクティブとしてマークします。

DEVICE_CERTIFICATE_SHARED_CHECK

- デバイス証明書の状態を変更して、AWS IoT で非アクティブとしてマークします。
- その証明書を使用するデバイスをモノのグループに追加します。

UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- 追加でサポートされているアクションはありません。

AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- 追加でサポートされているアクションはありません。

IOT_POLICY_OVERLY_PERMISSIVE_CHECK

- アクセス許可を制限するには、空の AWS IoT ポリシーバージョンを追加します。

IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK

- AWS IoT ポリシーの潜在的な設定ミスを特定します。

CA_CERT_APPROACHING_EXPIRATION_CHECK

- 証明書の状態を変更して、AWS IoT で非アクティブとしてマークします。

CONFLICTING_CLIENT_IDS_CHECK

- 追加でサポートされているアクションはありません。

DEVICE_CERT_APPROACHING_EXPIRATION_CHECK

- デバイス証明書の状態を変更して、AWS IoT で非アクティブとしてマークします。
- その証明書を使用するデバイスをモノのグループに追加します。

DEVICE_CERTIFICATE_KEY_QUALITY_CHECK

- デバイス証明書の状態を変更して、AWS IoT で非アクティブとしてマークします。
- その証明書を使用するデバイスをモノのグループに追加します。

CA_CERTIFICATE_KEY_QUALITY_CHECK

- 証明書の状態を変更して、AWS IoT で非アクティブとしてマークします。

REVOKED_DEVICE_CERT_CHECK

- デバイス証明書の状態を変更して、AWS IoT で非アクティブとしてマークします。
- その証明書を使用するデバイスをモノのグループに追加します。

LOGGING_DISABLED_CHECK

- ログ作成を有効化します。

AWS IoT Device Defender では、監査所見に関する以下のタイプの緩和アクションがサポートされています。

アクションの種類	コメント
ADD_THINGS_TO_THING_GROUP	デバイスを追加するグループを指定します。また、モノが属することができるグループの最大数を超える場合、1 つ以上の動的グループのメンバーシップをオーバーライドするかどうかも指定します。
ENABLE_IOT_LOGGING	ログ記録レベルとログ記録のアクセス許可を持つロールを指定します。DISABLED のログ記録レベルを指定することはできません。
PUBLISH_FINDING_TO_SNS	結果の発行先となるトピックを指定します。
REPLACE_DEFAULT_POLICY_VERSION	テンプレート名を指定します。ポリシーバージョンをデフォルトまたは空白のポリシーに置

アクションの種類	コメント
	き換えます。現在、BLANK_POLICY の値のみがサポートされています。
UPDATE_CA_CERTIFICATE	CA 証明書の新しい状態を指定します。現在、DEACTIVATE の値のみがサポートされています。
UPDATE_DEVICE_CERTIFICATE	デバイス証明書の新しい状態を指定します。現在、DEACTIVATE の値のみがサポートされています。

監査中に問題が見つかったときの標準アクションを設定することで、それらの問題に一貫して対応できます。これらの定義済みの緩和アクションを使用すると、問題をより迅速に解決し、人為的ミスの可能性を低く抑えることができます。

> **Important**
>
> 証明書の変更、新しいモノのグループへのモノの追加、またはポリシーの置き換えを行う緩和アクションを適用すると、デバイスとアプリケーションに影響する可能性があります。たとえば、デバイスが接続できなくなる場合があります。緩和アクションは、適用する前に、その影響を考慮してください。デバイスやアプリケーションが正常に機能できるようになる前に、問題を解決するための他のアクションを実行する必要がある場合があります。たとえば、更新されたデバイス証明書を提供する必要がある場合があります。緩和アクションはリスクをすばやく制限するのに役立ちますが、根本的な問題に対処するための是正措置を講じる必要があります。

デバイス証明書の再アクティブ化などの一部のアクションは、手動でのみ実行できます。AWS IoT Device Defender には、適用された緩和アクションを自動的にロールバックするメカニズムはありません。

Detect 緩和アクション

AWS IoT Device Defender では、Detect アラームに関する以下のタイプの緩和アクションがサポートされています。

アクションの種類	コメント
ADD_THINGS_TO_THING_GROUP	デバイスを追加するグループを指定します。また、モノが属することができるグループの最大数を超える場合、1 つ以上の動的グループのメンバーシップをオーバーライドするかどうかも指定します。

緩和アクションを定義および管理する方法

AWS IoT コンソールまたは AWS CLI を使用して、AWS アカウント の緩和アクションを定義および管理できます。

緩和アクションの作成

定義する各緩和アクションは、事前定義されたアクションタイプとアカウントに固有のパラメータの組み合わせです。

AWS IoT コンソールを使用して緩和アクションを作成するには

1. AWS IoT コンソールの [Mitigation actions page] (緩和アクション) ページを開きます。
2. [Mitigation actions] (緩和アクション) ページで、[Create] (作成) を選択します。
3. [Create a new mitigation action] (新規の緩和アクションの作成) ページの [Action name] (アクション名) に、緩和アクションの一意の名前を入力します。
4. [アクションの種類] で、定義するアクションのタイプを指定します。
5. [Permissions] (アクセス権限) で、アクションが適用されるアクセス権限の IAM ロールを選択します。
6. 各アクションタイプは、異なる一連のパラメータをリクエストします。アクションのパラメータを入力します。たとえば、[モノのグループにモノを追加] アクションタイプを選択した場合は、追加先グループを選択し、[Override dynamic groups (動的グループのオーバーライド)] を選択またはオフにします。
7. [Create] (作成) を選択して、緩和アクションを AWS アカウントに保存します。

AWS CLI を使用して緩和アクションを作成するには

- CreateMitigationAction コマンドを使用して、緩和アクションを作成します。アクションに付ける一意の名前は、そのアクションを監査の結果に適用するときに使用されます。わかりやすい名前を選択します。

AWS IoT コンソールを使用して緩和アクションを表示および変更するには

1. AWS IoT コンソールの [Mitigation actions page] (緩和アクション) ページを開きます。

   [Mitigation actions] (緩和アクション) ページには、AWS アカウント に対して定義されたすべての緩和アクションのリストが表示されます。
2. 変更する緩和アクションのアクション名リンクを選択します。
3. [Edit] (編集) を選択して、緩和アクションを変更します。緩和アクションの名前は識別に使用されるため、名前を変更することはできません。
4. [Update] (更新) を選択して、緩和アクションに対する変更を AWS アカウント に保存します。

AWS CLI を使用して緩和アクションを一覧表示するには

- ListMitigationAction コマンドを使用して、緩和アクションを一覧表示します。緩和アクションを変更または削除する場合は、名前をメモします。

AWS CLI を使用して緩和アクションを更新するには

- 緩和アクションを変更するには、UpdateMitigationAction コマンドを使用します。

AWS IoT コンソールを使用して緩和アクションを削除するには

1. AWS IoT コンソールの [Mitigation actions page] (緩和アクション) ページを開きます。

   [Mitigation actions] (緩和アクション) ページには、AWS アカウント に対して定義されたすべての緩和アクションが表示されます。
2. 削除する緩和アクションを選択し、[Delete] (削除) を選択します。
3. [Are you sure you want to delete] (本当に削除しますか) ウィンドウで、[Delete] (削除) を選択します。

AWS CLI を使用して緩和アクションを削除するには

- 緩和アクションを変更するには、UpdateMitigationAction コマンドを使用します。

AWS IoT コンソールを使用して緩和アクションの詳細を表示するには

1. AWS IoT コンソールの [Mitigation actions page] (緩和アクション) ページを開きます。

   [Mitigation actions] (緩和アクション) ページには、AWS アカウント に対して定義されたすべての緩和アクションが表示されます。
2. 表示する緩和アクションのアクション名リンクを選択します。

AWS CLI を使用して緩和アクションの詳細を表示するには

- 緩和アクションの詳細を表示するには、DescribeMitigationAction コマンドを使用します。

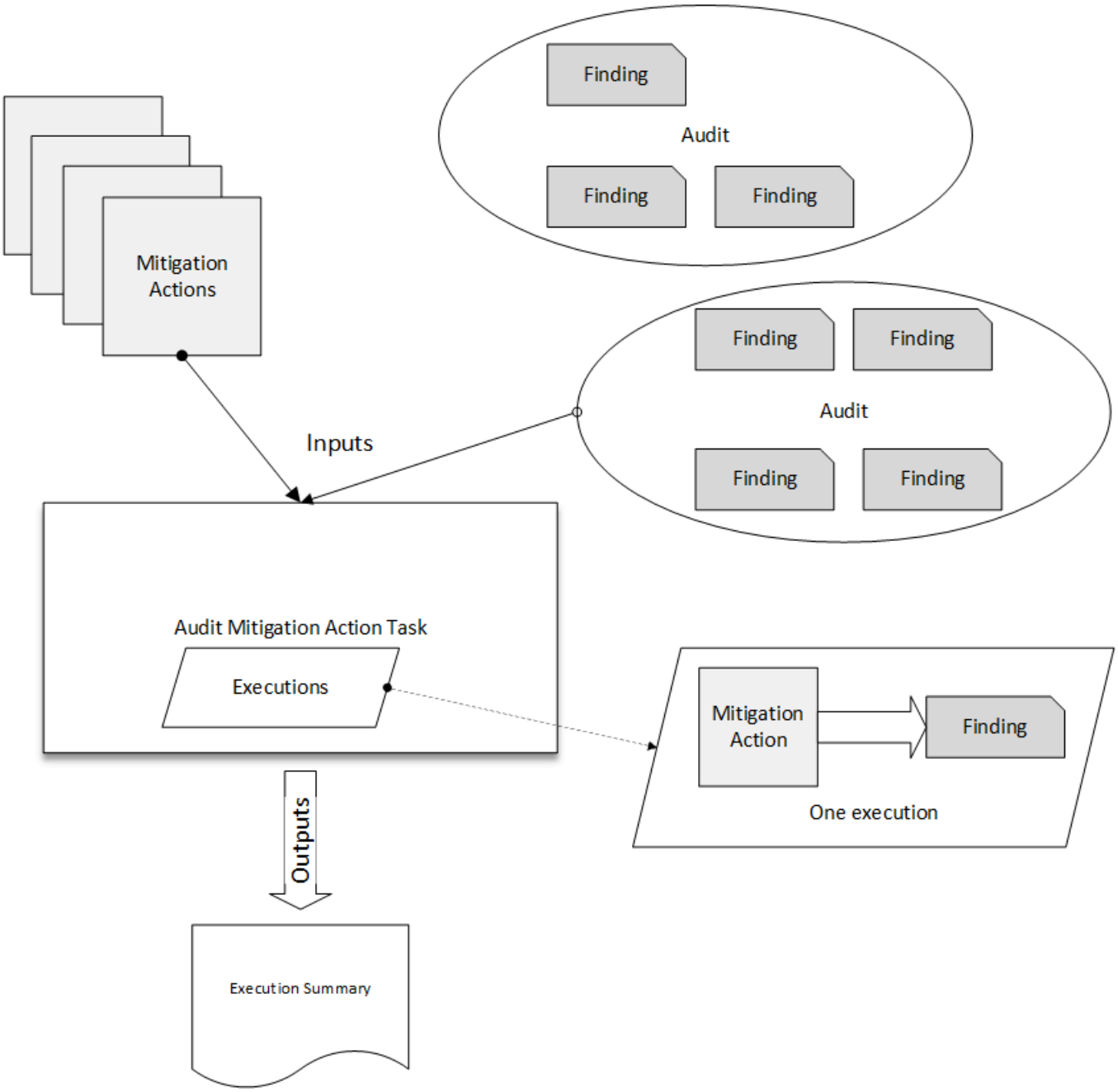
緩和アクションの適用

一連の緩和アクションを定義したら、それらのアクションを監査の結果に適用できます。アクションを適用すると、監査緩和アクションタスクを開始します。このタスクは、結果のセットとそれに適用するアクションによっては、完了までに時間がかかることがあります。たとえば、証明書の有効期限が切れた大規模なデバイスプールがある場合、それらの証明書をすべて無効にするか、それらのデバイスを隔離グループに移動するまでに時間がかかることがあります。ログ記録の有効化など、他のアクションはすばやく完了できます。

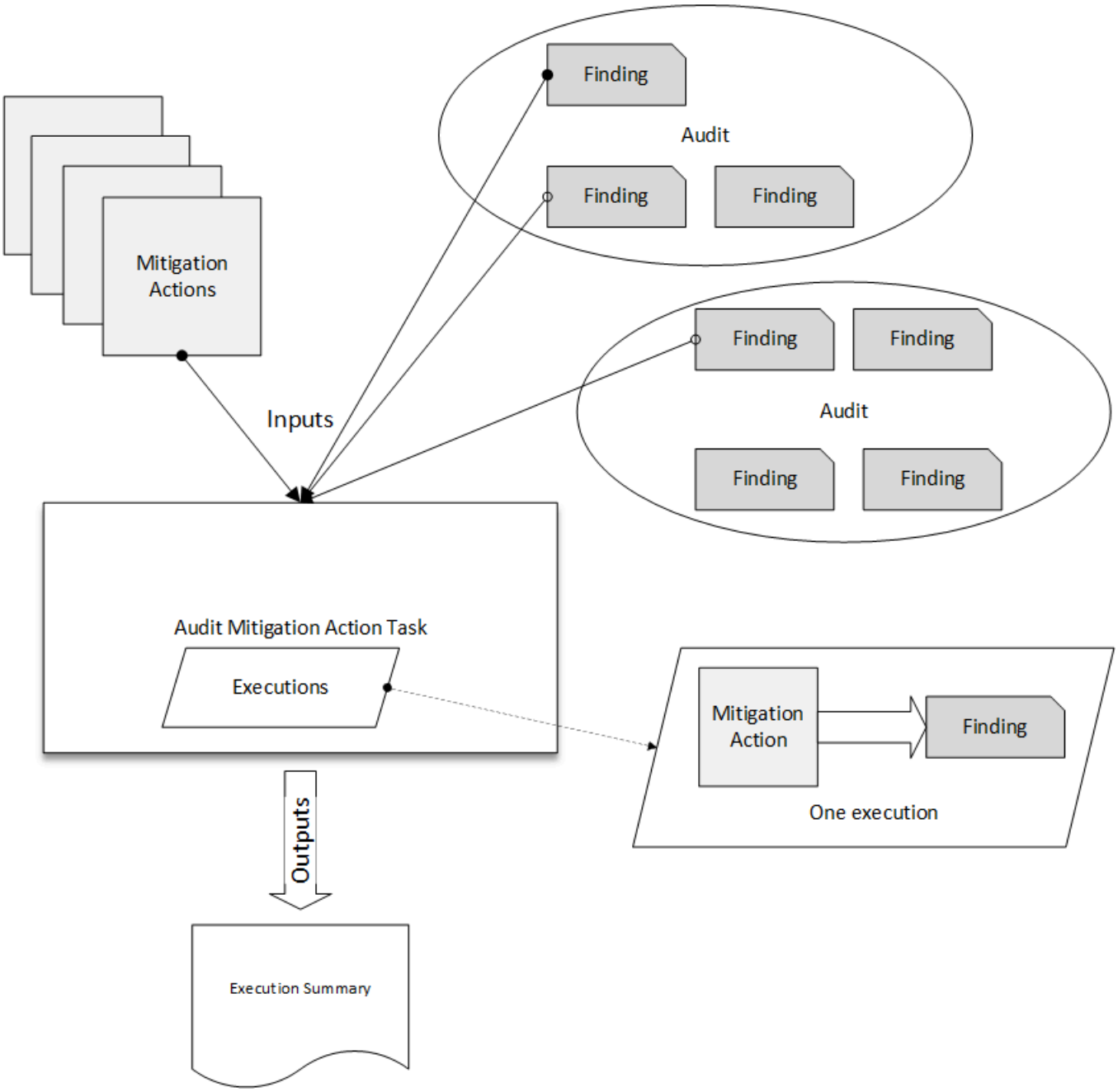
アクション実行のリストを表示し、まだ完了していない実行をキャンセルできます。キャンセルされたアクション実行の一部として既に実行されているアクションはロールバックされません。結果のセットに複数のアクションを適用していて、アクションの 1 つが失敗した場合、その結果に対する後続のアクションはスキップされます (ただし、他の結果には適用されます)。結果のタスクのステータスは FAILED です。結果に適用したときに 1 つ以上のアクションが失敗した場合、 `taskStatus` は失敗に設定されます。アクションは、指定された順序で適用されます。

それぞれのアクション実行では、一連のアクションがターゲットに適用されます。そのターゲットは、結果のリストまたは監査からのすべての結果にすることができます。

次の図は、1 つの監査からすべての結果を取得し、その結果に一連のアクションを適用する監査緩和タスクを定義する方法を示しています。1 回の実行で、1 つのアクションが 1 つの結果に適用されます。監査緩和アクションタスクは、実行の概要を出力します。



次の図は、1 つ以上の監査から個々の結果のリストを取得し、その結果に一連のアクションを適用する監査緩和タスクを定義する方法を示しています。1 回の実行で、1 つのアクションが 1 つの結果に適用されます。監査緩和アクションタスクは、実行の概要を出力します。



緩和アクションを適用するには、AWS IoT コンソールまたは AWS CLI を使用できます。

AWS IoT コンソールを使用して、アクション実行を開始することで緩和アクションを適用するには

1. AWS IoT コンソールの [Audit results] (監査結果) ページを開きます。
2. アクションを適用する監査の名前を選択します。
3. [Start mitigation actions] (緩和アクションの開始) を選択します。すべてのチェック項目が適合している場合、このボタンは使用できません。

4. [Start a new mitigation action] (新規の緩和アクションの開始) では、タスク名はデフォルトで監査 ID になりますが、わかりやすいものに変更できます。
5. 監査で 1 つ以上の不適合の結果があったチェックのタイプごとに、適用する 1 つ以上のアクションを選択できます。チェックタイプに有効なアクションのみが表示されます。

   > **Note**
   >
   > AWS アカウント にアクションを設定していない場合、アクションのリストは空になります。[Create mitigation action] (緩和アクションの作成) リンクを選択して、1 つまたは複数の緩和アクションを作成できます。

6. 適用するすべてのアクションを指定したら、[Start task] (タスクの開始) を選択します。

AWS CLI を使用して、監査緩和アクションの実行を開始して緩和アクションを適用するには

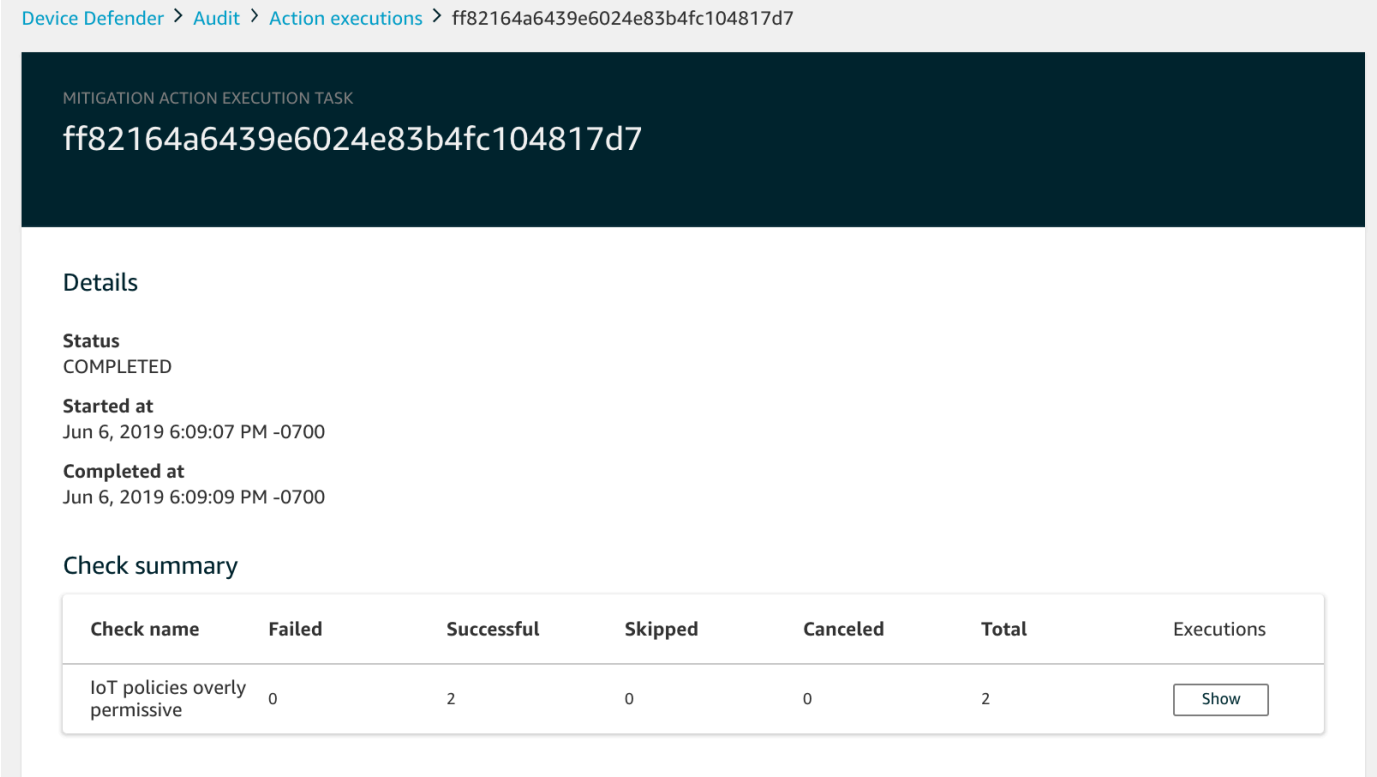
1. 監査のすべての結果にアクションを適用する場合は、ListAuditTasks コマンドを使用してタスク ID を見つけます。
2. 選択した結果にのみアクションを適用する場合は、ListAuditFindings コマンドを使用して結果 ID を取得します。
3. ListMitigationActions コマンドを使用して、適用する緩和アクションの名前を書き留めます。
4. StartAuditMitigationActionsTask コマンドを使用して、ターゲットにアクションを適用します。タスク ID を書き留めます。ID を使用して、アクション実行の状態の確認、詳細の確認、またはキャンセルを行うことができます。

AWS IoT コンソールを使用してアクションの実行を表示するには

1. AWS IoT コンソールの [Action tasks] (アクションタスク) ページを開きます。

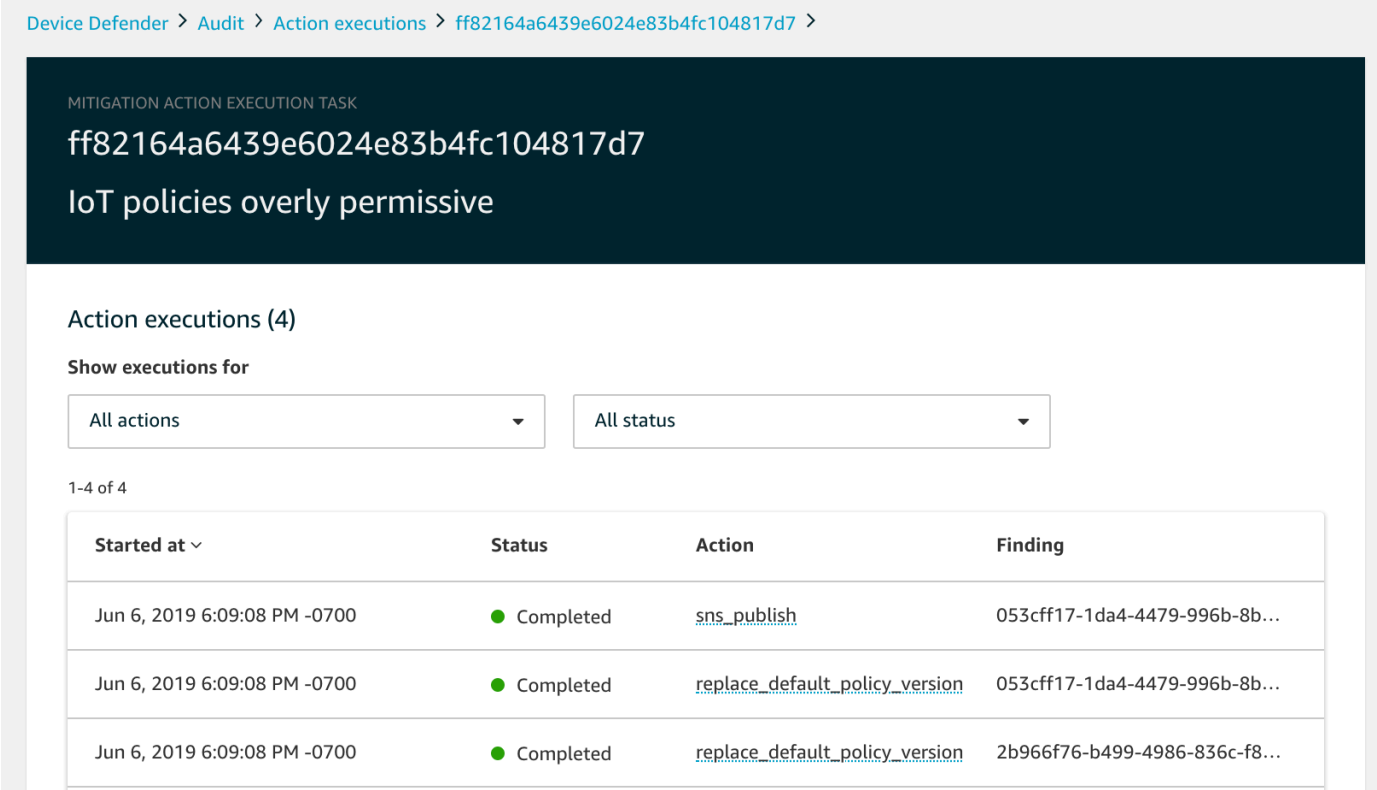
   アクションタスクのリストには、それぞれが開始された時刻と現在のステータスが表示されます。

2. [名前] リンクを選択して、タスクの詳細を表示します。詳細には、タスクによって適用されるすべてのアクション、そのターゲット、およびステータスが含まれます。



[Show executions for (実行を表示)] フィルターを使用して、アクションの種類やアクションの状態に注目できます。

3. タスクの詳細を表示するには、[Executions (実行)] で [Show (表示)] を選択します。



AWS CLI を使用して、開始したタスクを一覧表示するには

1. ListAuditMitigationActionsTasks を使用して、監査結果の緩和アクションタスクを表示します。フィルターを指定して結果を絞り込むことができます。タスクの詳細を表示する場合は、タスク ID をメモしておきます。
2. 特定の監査緩和アクションタスクの実行の詳細を表示するには、ListAuditMitigationActionsExecutions を使用します。
3. 開始時に指定されたパラメータなど、タスクの詳細を表示するには、DescribeAuditMitigationActionsTask を使用します。

AWS CLI を使用して、実行中の監査緩和アクションタスクをキャンセルするには

1. ListAuditMitigationActionsTasks コマンドを使用して、実行をキャンセルするタスクのタスク ID を見つけます。フィルターを指定して結果を絞り込むことができます。
2. タスク ID を使用して ListDetectMitigationActionsExecutions コマンドを使用し、監査緩和アクションタスクをキャンセルします。完了したタスクはキャンセルできません。タスクをキャンセルすると、残りのアクションは適用されませんが、既に適用された緩和アクションはロールバックされません。

アクセス許可

定義する緩和アクションごとに、そのアクションを適用するために使用するロールを指定する必要があります。

緩和アクションのアクセス許可

アクションの種類	アクセス許可ポリシーテンプレート	

UPDATE_DEVICE_CERTIFICATE	<pre>{
	    "Version":"2012-10-17",
	    "Statement":[
	        {
	            "Effect": "Allow",
	            "Action":[
	"iot:UpdateCertificate"
	            ],
	            "Resource":[
	"*"
	            ]
	        }
	    ]
	}</pre>	

UPDATE_CA_CERTIFICATE	<pre>{
	    "Version":"2012-10-17",
	    "Statement":[
	        {
	            "Effect": "Allow",
	            "Action":[</pre>	

アクションの種類	アクセス許可ポリシーテンプレート	

	<pre>"iot:UpdateCACertificate"
	],
	"Resource":
	[
	"*"
	]
	}
	]
	}</pre>	

ADD_THINGS_TO_THING_GROUP	<pre>{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":
	"Allow",
	"Action":[
	"iot:ListPrincipalThings",
	"iot:AddThingToThingGroup"
	],
	"Resource":
	[
	"*"
	]
	}
	]
	}</pre>	

アクションの種類	アクセス許可ポリシーテンプレート	

REPLACE_DEFAULT_POLICY_VERSION	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"iot:CreatePolicyVersion"
	],
	"Resource":[
	"*"
	]
	}
	]
	}	

アクションの種類	アクセス許可ポリシーテンプレート	

ENABLE_IOT_LOGGING	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect": "Allow",
	"Action":[
	"iot:SetV2LoggingOptions"
	],
	"Resource": [
	"*"
	]
	},
	{
	"Effect": "Allow",
	"Action":[
	"iam:PassRole"
	],
	"Resource": [
	"*<IAM role ARN used for setting up logging>*"
	]
	}
	]
	}	

アクションの種類	アクセス許可ポリシーテンプレート	

PUBLISH_FINDING_TO_SNS	<pre>{
	    "Version":"2012-10-17",
	    "Statement":[
	        {
	            "Effect": "Allow",
	            "Action":[
	"sns:Publish"
	            ],
	            "Resource":[
	                "<The SNS topic to which the finding is published> "
	            ]
	        }
	    ]
	}</pre>	

すべての緩和アクションタイプに、次の信頼ポリシーテンプレートを使用します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iot:*:111122223333::*"
        },
```

```
        "StringEquals": {
          "aws:SourceAccount": "111122223333:"
        }
      }
    }
  ]
}
```

緩和アクションコマンド

これらの緩和アクションコマンドを使用して、後で 1 つ以上の監査所見のセットに適用できる一連のアクションを AWS アカウント に定義できます。コマンドカテゴリは 3 つあります。

- アクションを定義および管理するために使用するもの。
- 結果を監査するために、これらのアクションのアプリケーションを開始および管理するために使用するもの。
- アラームを検出するために、これらのアクションのアプリケーションを開始および管理するために使用するもの。

緩和アクションコマンド

アクションの定義と管理	Audit 実行を開始および管理する	Detect 実行を開始および管理する
CreateMitigationAction	CancelAuditMitigationActionsTask	CancelDetectMitigationActionsTask
DeleteMitigationAction	DescribeAuditMitigationActionsTask	DescribeDetectMitigationActionsTask
DescribeMitigationAction	ListAuditMitigationActionsTasks	ListDetectMitigationActionsTasks
ListMitigationActions	StartAuditMitigationActionsTask	StartDetectMitigationActionsTask
UpdateMitigationAction	ListAuditMitigationActionsExecutions	ListDetectMitigationActionsExecutions

AWS IoT Device Defender を他のAWSサービスと併用する

AWS IoT Greengrass を実行するデバイスでの AWS IoT Device Defender の使用

AWS IoT Greengrass は、デバイスの動作を継続的に監視するために、AWS IoT Device Defender との事前構築された統合を提供します。

- AWS IoT Greengrass V1 と Device Defender を統合
- AWS IoT Greengrass V2 と Device Defender を統合

FreeRTOSおよび組み込み機器とAWS IoT Device Defenderを使用

FreeRTOS デバイスでAWS IoT Device Defenderを使用するには、お使いのデバイスにFreeR Embedded C SDKまたはAWSIoT Device Defender ライブラリがインストールされている必要があります。FreeRTOS Embedded C SDKは、AWSIoT Device Defender ライブラリを含みます。AWS IoT Device Defenderを FreeRTOS デバイスに統合する方法については、以下のデモを参照してください：

- FreeRTOS標準メトリックとカスタムメトリクスデモ用AWS IoT Device Defender
- AWS IoT Device Defender にメトリクスを送信するための MQTT エージェントの使用
- AWS IoT Device Defender にメトリクスを送信するための MQTT コアライブラリの使用

FreeRTOS を使用せずに組み込みデバイスで AWS IoT Device Defender を使用するには、デバイスに AWS IoT Embedded C SDK または AWS IoT Device Defender ライブラリがインストールされている必要があります。AWSIoT Embedded C SDK はAWSIoT Device Defender ライブラリを含みます。AWS IoT Device Defender を組み込みデバイスに統合する方法については、次のデモ、AWS IoT Embedded SDK 標準およびカスタムメトリクスのデモのための AWS IoT Device Defender を参照してください。

AWS IoT Device Defenderと使用するAWS IoT Device Management

AWS IoT Device Management フリートのインデックス作成を使用して、AWS IoT Device Defender Detect 違反をインデックス化、検索、および集計できます。Device Defender 違反データがフリートインデックス作成でインデックス化された後で、Fleet Hub アプリケーションの Device Defender 違反データにアクセスしてクエリを実行したり、違反データに基づいてフリートアラームを作成してデバイスのフリート全体の異常をモニタリングしたり、Fleet Hub ダッシュボードでフリートアラームを表示したりできます。

> **Note**
>
> AWS IoT Device Defender 違反データのインデックス化をサポートするフリートインデックス機能は、AWS IoT Device Management のプレビューリリースにあり、変更される可能性があります。

- フリートインデックス作成の管理
- クエリ構文
- Fleet Hub アプリケーション用フリートインデックス作成の管理
- 入門

AWS Security Hub との統合

AWS Security Hub では、AWS のセキュリティ状態を包括的に把握し、セキュリティ業界標準および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、およびサポートされているサードパーティーパートナー製品全体からセキュリティデータを収集します。Security Hub 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。

Security Hub との AWS IoT Device Defender 統合により、AWS IoT Device Defender から Security Hub に結果を送信できます。Security Hub は、セキュリティ体制の分析にこれらの検出結果を含めます。

目次

統合の有効化と構成

AWS IoT Device Defender を Security Hub と統合する前に、まず Security Hub を有効にする必要があります。Security Hub を有効にする方法の詳細については、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。

AWS IoT Device Defender と Security Hub の両方を有効にしたら、Security Hub コンソールで [Integrations] (統合) ページを開き、[Audit] (監査)、[Detect] (検出)、またはその両方の [Accept findings] (結果を受け入れる) を選択します。AWS IoT Device Defender が結果を Security Hub へ送信します。

AWS IoT Device Defender から Security Hub に結果を送信する方法

Security Hub では、セキュリティの問題が調査結果として追跡されます。結果の中には、他の AWS のサービスやサードパーティー製品が検出した問題に由来するものもあります。

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。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 詳細については、AWS Security Hub ユーザーガイドの「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 のすべての調査結果で、AWS Security Finding 形式 (ASFF) と呼ばれる標準の JSON 形式が使用されます。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。詳細については、AWS Security Hub ユーザーガイドの「AWS Security Finding 形式 (ASFF)」を参照してください。

AWS IoT Device Defender は Security Hub に結果を送信する AWS サービスの 1 つです。

AWS IoT Device Defender が送信する検出結果の種類

Security Hub 統合を有効にすると、AWS IoT Device Defender 監査は生成した結果 (チェックサマリーと呼ばれる) を Security Hub に送信します。チェックサマリーは、特定の監査チェックタイプと特定の監査タスクに関する一般的な情報です。監査の詳細については、「監査チェック」を参照してください。

AWS IoT Device Defender 監査は、各監査タスクの監査チェックサマリーと監査結果の両方について、検出結果の更新を Security Hub に送信します。監査チェックで見つかったすべてのリソースが準拠している場合、または監査タスクがキャンセルされた場合、監査は Security Hub のチェックサマリーをアーカイブ済みのレコード状態に更新します。あるリソースが監査チェックで非準拠と報告されたが、最後の監査タスクで準拠していると報告された場合、監査はそのリソースを準拠に変更し、Security Hub での結果を ARCHIVED レコードの状態に更新します。

AWS IoT Device Defender Detect は、違反の検出結果を Security Hub に送信します。これらの違反結果には、機械学習 (ML)、統計的動作、静的動作が含まれます。

結果を Security Hub に送信するために、AWS IoT Device Defender は AWS Security Finding Format (ASFF) を使用します。ASFF では、Types フィールドが検出結果タイプを提供します。AWS IoT Device Defender の検出結果には、Types に対する次の値を指定できます。

異常な動作

競合する MQTT クライアント ID とデバイス証明書共有チェックの検出タイプ、および Detect の検出タイプ。

ソフトウェアと設定のチェック/脆弱性

その他すべての Audit チェックの結果タイプ。

結果が送信されるまでのレイテンシー

AWS IoT Device Defender Audit によって新しい結果が作成されると、その結果が Security Hub へ送信されます。レイテンシーは監査タスクで生成される結果の量によって異なります。Security Hub は通常 1 時間以内に結果を受け取ります。

AWS IoT Device Defender Detect は違反の検出結果を Security Hub へ送信します。違反がアラームに入ったり、アラームから外れたりすると (つまり、アラームが作成または削除された場合)、対応する Security Hub の結果がすぐに作成またはアーカイブされます。

Security Hub が使用できないときに再試行する

Security Hub が使用できない場合、AWS IoT Device Defender Audit と AWS IoT Device Defender Detect は検出結果が受信されるまでその結果を再送信し続けます。

Security Hub の既存の結果を更新する

AWS IoT Device Defender Audit 結果が Security Hub に送信されると、チェックされたリソース ID と監査チェックタイプによって識別できます。同じリソースと監査チェックの後続の監査タスクで新しい監査結果が生成された場合、AWS IoT Device Defender Audit 更新を送信して、検出アクティビティの追加の観測を Security Hub に反映します。同じリソースと監査チェックに対する後続の監査タスクで追加の監査結果が生成されない場合、リソースは監査チェックに準拠するように変更されます。AWS IoT Device DefenderAudit は検出結果を Security Hub へ送信します。

AWS IoT Device Defender Audit では、Security Hub チェックサマリーも更新されます。監査チェックで準拠していないリソースが見つかったり、チェックが失敗したりすると、Security Hub の検索結果のステータスがアクティブになります。それ以外の場合、AWS IoT Device Defender Audit は検出結果を Security Hub へ送信します。

AWS IoT Device Defender Detect は、違反があったとき (アラーム中など) に Security Hub の検出結果を作成します。この結果は、次のいずれかの条件が満たされた場合にのみ更新されます。

- Security Hub での検索結果はまもなく期限切れになるため、AWS IoT Device Defender は、更新を送信して結果を最新の状態に保ちます。結果は、最新の更新から 90 日後、または更新が行われない場合は作成日から 90 日後に削除されます。Security Hub クォータの詳細については、AWS Security Hub ユーザーガイドの「Security Hub クォータ」を参照してください。
- 該当する違反はアラーム終了となり、AWS IoT Device Defender は、検出ステータスを ARCHIVED に更新します。

AWS IoT Device Defender からの一般的な結果

AWS IoT Device Defender は、AWS Security Finding Format (ASFF) を使用して結果を Security Hub に送信します。

次の例は、監査結果についての Security Hub の一般的な結果を示しています。`ProductFields` の `ReportType` は `AuditFinding` です。

```
{
```

```
  "SchemaVersion": "2018-10-08",
  "Id": "336757784525/IOT_POLICY/policyexample/1/IOT_POLICY_OVERLY_PERMISSIVE_CHECK/
ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",
  "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/iot-device-defender-audit",
  "ProductName": "IoT Device Defender - Audit",
  "CompanyName": "AWS",
  "Region": "us-west-2",
  "GeneratorId": "1928b87ab338ee2f541f6fab8c41c4f5",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Check/Vulnerabilities"
  ],
  "CreatedAt": "2022-11-06T22:11:40.941Z",
  "UpdatedAt": "2022-11-06T22:11:40.941Z",
  "Severity": {
    "Label": "CRITICAL",
    "Normalized": 90
  },
  "Title": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK:
 ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",
  "Description": "IOT_POLICY policyexample:1 is reported as non-compliant for
 IOT_POLICY_OVERLY_PERMISSIVE_CHECK by Audit task 9f71b6e90cfb57d4ac671be3a4898e6a.
 The non-compliant reason is Policy allows broad access to IoT data plane actions:
 [iot:Connect].",
  "SourceUrl": "https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/
policy/policyexample",
  "ProductFields": {
    "CheckName": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",
    "TaskId": "9f71b6e90cfb57d4ac671be3a4898e6a",
    "TaskType": "ON_DEMAND_AUDIT_TASK",
    "PolicyName": "policyexample",
    "IsSuppressed": "false",
    "ReasonForNonComplianceCode": "ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",
    "ResourceType": "IOT_POLICY",
    "FindingId": "1928b87ab338ee2f541f6fab8c41c4f5",
    "PolicyVersionId": "1",
    "ReportType": "AuditFinding",
    "TaskStartTime": "1667772700554",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-west-2::product/
aws/iot-device-defender-audit/336757784525/IOT_POLICY/policyexample/1/
IOT_POLICY_OVERLY_PERMISSIVE_CHECK/ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",
    "aws/securityhub/ProductName": "IoT Device Defender - Audit",
    "aws/securityhub/CompanyName": "AWS"
  },
```

```
  "Resources": [
    {
      "Type": "AwsIotPolicy",
      "Id": "policyexample",
      "Partition": "aws",
      "Region": "us-west-2",
      "Details": {
        "Other": {
          "PolicyVersionId": "1"
        }
      }
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "CRITICAL"
    },
    "Types": [
      "Software and Configuration Check/Vulnerabilities"
    ]
  }
}
```

次の例は、監査チェックサマリーに関する Security Hub からの結果を示しています。ProductFields の ReportType は CheckSummary です。

```
  {
  "SchemaVersion": "2018-10-08",
  "Id": "615243839755/SCHEDULED_AUDIT_TASK/daily_audit_schedule_checks/
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/iot-device-defender-audit",
  "ProductName": "IoT Device Defender - Audit",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "f3021945485adf92487c273558fcaa51",
  "AwsAccountId": "123456789012",
```

```
  "Types": [
    "Software and Configuration Check/Vulnerabilities/CVE"
  ],
  "CreatedAt": "2022-10-18T14:20:13.933Z",
  "UpdatedAt": "2022-10-18T14:20:13.933Z",
  "Severity": {
    "Label": "CRITICAL",
    "Normalized": 90
  },
  "Title": "DEVICE_CERTIFICATE_KEY_QUALITY_CHECK Summary: Completed with 2 non-
compliant resources",
  "Description": "Task f3021945485adf92487c273558fcaa51 of weekly scheduled Audit
 daily_audit_schedule_checks completes. 2 non-cimpliant resources are found for
 DEVICE_CERTIFICATE_KEY_QUALITY_CHECK out of 1000 resources in the account. The
 percentage of non-compliant resources is 0.2%.",
  "SourceUrl": "https://us-east-1.console.aws.amazon.com/iot/home?region=us-east-1#/dd/
audit/results/f3021945485adf92487c273558fcaa51/DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
  "ProductFields": {
    "TaskId": "f3021945485adf92487c273558fcaa51",
    "TaskType": "SCHEDULED_AUDIT_TASK",
    "ScheduledAuditName": "daily_audit_schedule_checks",
    "CheckName": "DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
    "ReportType": "CheckSummary",
    "CheckRunStatus": "COMPLETED_NON_COMPLIANT",
    "NonComopliantResourcesCount": "2",
    "SuppressedNonCompliantResourcesCount": "1",
    "TotalResourcesCount": "1000",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/
iot-device-defender-audit/615243839755/SCHEDULED/daily_audit_schedule_checks/
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
    "aws/securityhub/ProductName": "IoT Device Defender - Audit",
    "aws/securityhub/CompanyName": "AWS"
  },
  "Resources": [
    {
      "Type": "AwsIotAuditTask",
      "Id": "f3021945485adf92487c273558fcaa51",
      "Region": "us-east-1"
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
```

```
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "CRITICAL"
    },
    "Types": [
      "Software and Configuration Check/Vulnerabilities/CVE"
    ]
  }
}
```

次の例は、AWS IoT Device Defender 検出違反に対する Security Hub からの一般的な検出結果を示しています。

```
  {
  "SchemaVersion": "2018-10-08",
  "Id": "e92a782593c6f5b1fc7cb6a443dc1a12",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/iot-device-defender-
detect",
  "ProductName": "IoT Device Defender - Detect",
  "CompanyName": "AWS",
  "Region": "us-east-1",
  "GeneratorId": "arn:aws:iot:us-east-1:123456789012:securityprofile/
MySecurityProfile",
  "AwsAccountId": "123456789012",
  "Types": [
    "Unusual Behaviors"
  ],
  "CreatedAt": "2022-11-09T22:45:00Z",
  "UpdatedAt": "2022-11-09T22:45:00Z",
  "Severity": {
    "Label": "MEDIUM",
    "Normalized": 40
  },
  "Title": "Registered thing MyThing is in alarm for STATIC behavior MyBehavior.",
  "Description": "Registered thing MyThing violates STATIC behavior MyBehavior of
 security profile MySecurityProfile. Violation was triggered because the device did not
 conform to aws:num-disconnects less-than 1.",
  "SourceUrl": "https://us-east-1.console.aws.amazon.com/iot/home?region=us-east-1#/dd/
securityProfile/MySecurityProfile?tab=violations",
  "ProductFields": {
```

```
    "ComparisonOperator": "less-than",
    "BehaviorName": "MyBehavior",
    "ViolationId": "e92a782593c6f5b1fc7cb6a443dc1a12",
    "ViolationStartTime": "1668033900000",
    "SuppressAlerts": "false",
    "ConsecutiveDatapointsToAlarm": "1",
    "ConsecutiveDatapointsToClear": "1",
    "DurationSeconds": "300",
    "Count": "1",
    "MetricName": "aws:num-disconnects",
    "BehaviorCriteriaType": "STATIC",
    "ThingName": "MyThing",
    "SecurityProfileName": "MySecurityProfile",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/iot-
device-defender-detect/e92a782593c6f5b1fc7cb6a443dc1a12",
    "aws/securityhub/ProductName": "IoT Device Defender - Detect",
    "aws/securityhub/CompanyName": "AWS"
  },
  "Resources": [
    {
      "Type": "AwsIotRegisteredThing",
      "Id": "MyThing",
      "Region": "us-east-1",
      "Details": {
        "Other": {
          "SourceUrl": "https://us-east-1.console.aws.amazon.com/iot/home?region=us-
east-1#/thing/MyThing?tab=violations",
          "IsRegisteredThing": "true",
          "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyThing"
        }
      }
    }
  ],
  "WorkflowState": "NEW",
  "Workflow": {
    "Status": "NEW"
  },
  "RecordState": "ACTIVE",
  "FindingProviderFields": {
    "Severity": {
      "Label": "MEDIUM"
    },
    "Types": [
      "Unusual Behaviors"
```

```
    ]
  }
}
```

AWS IoT Device Defender が検出結果を Security Hub に送信するのを停止する

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。

詳細については、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無効化と有効化 (コンソール)」または「統合からの検出結果のフローの無効化 (Security Hub API、AWS CLI)」を参照してください。

サービス間の混乱した代理の防止

混乱した代理問題とは、アクションを実行する許可を持たないエンティティが、より高い特権を持つエンティティにそのアクションの実行を強制できるというセキュリティ問題です。AWS では、サービス間でのなりすましによって、混乱した代理問題が発生する場合があります。サービス間でのなりすましは、1 つのサービス (呼び出し元サービス) が、別のサービス (呼び出し対象サービス) を呼び出すときに発生する可能性があります。呼び出し元サービスが操作され、それ自身のアクセス許可を使用して、本来アクセス許可が付与されるべきではない方法で呼び出したサービスを介して別の顧客のリソースに対して働きかけることがあります。これを防ぐため、AWS では、アカウントのリソースへのアクセス権が付与されたサービスプリンシパルで、すべてのサービスのデータを保護するために役立つツールを提供しています。

セキュリティ問題の混乱により影響を受ける可能性のあるお客様からの AWS IoT Device Defender アクセスは、監査の実行、セキュリティプロファイル違反のSNS通知の送信、緩和措置の実行の3つです。これらの各アクションについて、`aws:SourceArn` の値は次のようである必要があります :

- UpdateAccountAuditConfiguration API (RoleArn 属性および NotificationTarget RoleArn 属性) で渡されたリソースの場合、`aws:SourceArn` と `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:` を使用して、リソースポリシーのスコープを絞り込んでおく必要があります。
- CreateMitigationAction API (RoleArn 属性) で渡されたリソースの場合、`aws:SourceArn` および `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:mitigationaction/`*`mitigationActionName` を使用して、リソースポリシーのスコープを絞り込んでおく必要があります。

- CreateSecurityProfile API (alertTargets 属性) で渡されたリソースの場合、aws:SourceArn と arn:*arnPartition*:iot:*region*:*accountId*:securityprofile/*securityprofileName* を使用して、リソースポリシーのスコープを絞り込んでおく必要があります。

混乱した代理問題から保護するための最も効果的な方法は、リソースの完全な ARN を指定して aws:SourceArn グローバル条件コンテキストキーを使用することです。リソースの完全な ARN が不明な場合や、複数のリソースを指定する場合は、aws:SourceArn グローバルコンテキスト条件キーを使用して、ARN の未知部分をワイルドカード (*) で表します。例えば、arn:aws:*servicename*:*:*123456789012*:* です。

次の例では、AWS IoT Device Defender で aws:SourceArn および aws:SourceAccount グローバル条件コンテキストキーを使用して、混乱した代理問題を回避する方法を示します。

```
 {
"Version": "2012-10-17",
"Statement": {
  "Sid": "ConfusedDeputyPreventionExamplePolicy",
  "Effect": "Allow",
  "Principal": {
    "Service": "iot.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": "arn:aws:iot:*:123456789012::*"
    },
    "StringEquals": {
      "aws:SourceAccount": "123456789012:"
    }
  }
}
}
```

デバイスエージェントのセキュリティベストプラクティス

最小特権

エージェントプロセスには、役割を実行するために必要な最小限のアクセス許可のみを付与してください。

基本的なメカニズム

- エージェントは、root 以外のユーザーとして実行する必要があります。
- エージェントは、その独自のグループで専用ユーザーとして実行してください。
- ユーザー/グループには、メトリクスの収集と送信に必要な読み取り専用アクセス許可のみリソースで付与してください。
- 例: サンプルエージェントの /proc /sys での読み取り専用。
- アクセス許可を制限して実行するプロセスをセットアップする方法の例については、Python サンプルエージェントに含まれているセットアップ手順を参照してください。

エージェントプロセスをさらに制限/分離するためのよく知られている Linux メカニズムは多数あります。

高度なメカニズム

- CGroups
- SELinux
- Chroot
- Linux 名前空間

運用の耐障害性

エージェントプロセスには、クラッシュしたり完全に終了したりしないように予期しない動作エラーや例外からの回復力が必要です。コードは、例外を適切に処理する必要があり、予防策として、予期せずに終了した場合 (システムの再起動やキャッチされない例外などのため) は自動的に再起動するように設定されている必要があります。

最小の依存関係

エージェントは、その実装でできる限り少ない依存関係 (つまり、サードパーティーのライブラリ) を使用する必要があります。タスクが複雑なためにライブラリの使用が適切な場合 (Transport Layer Security など)、よく管理された依存関係のみを使用し、それらを最新の状態に保つメカニズムを確立します。追加された依存関係に、エージェントにより使用されず、デフォルトでアクティブな機能が含まれている場合 (たとえば、開いているポート、ドメインソケットなど)、コードで無効にするか、ライブラリの構成ファイルを使用して無効にします。

プロセスの分離

エージェントプロセスには、デバイスメトリクスの収集と送信の実行に必要な機能のみ含まれている必要があります。他のシステムプロセスの上でコンテナとして実行したり、他の範囲外ユースケースの機能を実装したりすることはできません。さらに、エージェントプロセスは、ドメイ

ンソケットやネットワークサービスポートなどのインバウンド通信を作成しないようにする必要があります。作成すると、ローカルまたはリモートプロセスが処理に干渉できるようになり、整合性と分離に影響が及ぶ可能性があります。

ステルス性

プロセスエージェントの名前に、その目的やセキュリティ上の価値を示すセキュリティ、モニタリング、監査などのキーワードを使用することはできません。汎用コード名やランダムかつ一意のデバイスごとのプロセス名が推奨されます。エージェントのバイナリが存在するディレクトリや、プロセス引数の名前と値の命名でも、同じ原則に従う必要があります。

最小限の情報共有

デバイスにデプロイされるエージェントアーティファクトには、特権認証情報、デバッグおよびデッドコード、インラインコメントまたはドキュメントファイルなどの機密情報を含めることはできません。エージェントが収集したメトリクスのサーバー側の処理の詳細やバックエンドシステムに関する詳細が開示されます。

Transport Layer Security

データ転送用の TLS セキュアチャネルを確立するには、エージェントプロセスは証明書チェーンやドメイン名検証などのクライアント側検証をすべてアプリケーションレベルで (デフォルトで有効になっていない場合) 強制する必要があります。さらに、エージェントは、信頼された機関が含まれていて、侵害された証明書発行元に属している証明書が含まれていないルート証明書ストアを使用する必要があります。

安全なデプロイ

コードのプッシュや同期、そのバイナリ、ソースコード、設定ファイル (信頼されたルート証明書を含む) を含むリポジトリなど、すべてのエージェントデプロイメカニズムは、許可されていないコードインジェクションや改ざんを防ぐためにアクセス制御される必要があります。デプロイメカニズムがネットワーク通信に依存している場合、暗号化方法を利用して転送中のデプロイアーティファクトの整合性を保護します。

詳細情報

- AWS IoT Device Defender のセキュリティ
- AWS IoT セキュリティモデルを理解する
- Redhat: A Bite of Python
- Python における10個の一般的なセキュリティ問題とそれを避ける方法
- 最も少ない特権とは何か & なぜそれが必要なのか？
- OWASP Embedded Security Top 10

- OWASP IoT Project

AWS IoT Device Defender トラブルシューティングガイド

このトピックの改善にご協力ください
より良いものにするために必要なことを教えてください

全般

Q: AWS IoT Device Defender を使用するための前提条件はありますか?

A: デバイスでレポートされたメトリクスを使用する場合はまず、AWS IoT に接続されたデバイスまたはデバイスゲートウェイにエージェントをデプロイする必要があります。デバイスで、一貫したクライアント識別子またはモノの名前を指定する必要があります。

監査

Q: チェック項目を有効にしましたが、監査には長時間「進行中」と表示されています。問題が発生したのですか? 結果はいつ確認できますか?

A: チェック項目が有効になると、データ収集がすぐに開始されます。ただし、アカウントに収集するデータが大量にある場合 (例、証明書、モノ、ポリシー) は、有効にしてからしばらくの間チェックの結果が生成されないことがあります。

検出

Q: AWS IoT Device Defender セキュリティプロファイルの動作で設定するしきい値はどうすれば分かりますか?

A: まず、しきい値の低いセキュリティプロファイル動作を作成し、対応するデバイスセットを含むモノのグループにそれをアタッチします。AWS IoT Device Defender を使用して現在のメトリクスを表示し、ユースケースに合わせてデバイスの動作のしきい値を調整することができます。

Q: 動作を作成しましたが、予想される場合でも違反がトリガーされません。解決策はありますか?

A: デバイスを定義する際に、通常のデバイスの動作を指定してください。たとえば、TCP ポート 8888 で 1 つの中央サーバーにのみ接続するセキュリティカメラがある場合、他の接続を行うことは期待しません。カメラが別のポートで接続を行った場合にアラートを生成するには、以下のように動作を定義します。

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 8888 ]
    }
  }
}
```

カメラが TCP ポート 443 で TCP 接続を行った場合、デバイス動作への違反となり、アラートがトリガーされます。

Q: 1 つまたは複数の動作が違反しています。違反をクリアするにはどうすればよいですか?

A: 定義した動作プロファイルで定義されているように、予期される動作に戻るとアラームはクリアされます。動作プロファイルは、デバイスのメトリクスデータを受信したときに評価されます。デバイスが 2 日を超えてメトリクスを発行しない場合は、自動的に違反イベントが `alarm-invalidated` に設定されます。

Q: 違反した動作を削除しましたが、アラートを停止するにはどうすればいいですか?

A: 動作を削除すると、その後のすべての違反と、その動作に対するアラートが停止されます。それ以前のアラートは、通知メカニズムから空にする必要があります。動作を削除すると、アカウント内の他のすべての違反と同じ期間、その動作の違反記録が保持されます。

デバイスメトリクス

Q: 動作に違反していることが分かっているメトリクスレポートを送信していますが、違反がトリガーされません。何が問題なのでしょうか?

A: 以下の MQTT トピックにサブスクライブし、メトリクスレポートが承諾されていることをチェックしてください。

```
$aws/things/THING_NAME/defender/metrics/FORMAT/rejected
$aws/things/THING_NAME/defender/metrics/FORMAT/accepted
```

ここで、`THING_NAME` はメトリクスを報告しているモノの名前であり、`FORMAT` はモノによって送信されるメトリクスレポートの形式に応じて「json」または「cbor」です。

サブスクライブすると、送信されたメトリクスレポートごとにこれらのトピックに関するメッセージを受け取ります。`rejected` メッセージは、メトリクスレポートの解析に問題があったことを示しています。エラーメッセージは、メトリクスレポートでエラーを修正できるようにするため、メッセージペイロードに含まれています。`accepted` メッセージは、メトリクスレポートが適切に解析されたことを示しています。

Q: メトリクスレポートで空のメトリクスを送信した場合はどうなりますか?

A: ポートまたは IP アドレスの空のリストは常に、対応する動作に準拠していると見なされます。対応する動作が違反している場合、違反がクリアされます。

Q: AWS IoT レジストリにないデバイスのメッセージがデバイスメトリクスレポートに含まれているのはなぜですか?

1 つ以上のセキュリティプロファイルがすべてのモノまたは未登録のすべてのモノにアタッチされている場合、AWS IoT Device Defender には未登録のモノからのメトリクスが含まれます。未登録のモノからメトリクスを除外する場合は、すべてのデバイスではなく登録済みのすべてのデバイスにプロファイルをアタッチします。

Q: セキュリティプロファイルを未登録のすべてのデバイスまたはすべてのデバイスに適用しても、未登録の 1 つ以上のデバイスからのメッセージが表示されません。どうすれば解決できますか?

サポートされているいずれかの形式を使用して、適切な形式のメトリクスレポートを送信していることを確認してください。詳細については、デバイスメトリクスドキュメントの仕様 を参照してください。未登録のデバイスで一貫したクライアント識別子またはモノの名前が使用されていることを確認します。モノ名に制御文字が含まれている場合、またはモノ名が 128 バイトの UTF-8 エンコード文字より長い場合、デバイスによって報告されたメッセージは拒否されます。

Q: 未登録のデバイスがレジストリに追加された場合、または登録済みのデバイスが未登録になった場合はどうなりますか?

A: レジストリにデバイスが追加または削除された場合は次のようになります。

- 違反の測定基準を引き続き公開すると、デバイスに対して 2 つの異なる違反が表示されます (1 つは登録済みのモノの名前の下、もう 1 つは未登録の ID の下)。古い ID に対するアクティブな違反は 2 日後には表示されなくなりますが、違反の履歴には最大 14 日間表示でされます。

Q: デバイスメトリクスレポートの ID フィールドにはどの値を指定する必要がありますか?

A: 正の整数で表された、メトリクスレポートごとに一意の値にする必要があります。一般的な手法として UNIX エポックタイムスタンプを使用できます。

Q: AWS IoT Device Defender メトリクスの専用 MQTT 接続を作成する必要はありますか?

A: 別個の MQTT 接続は必要ありません。

Q: デバイスメトリクスを発行するために接続するとき、どのクライアント ID を使用する必要がありますか?

AWS IoT レジストリにあるデバイス (モノ) の場合は、登録済みのモノの名前を使用します。AWS IoT レジストリにないデバイス (モノ) の場合は、AWS IoT に接続する際の一貫した ID を使用します。このプラクティスは、モノの名前の違反を一致させるのに役立ちます。

Q: 別のクライアント ID を持つデバイスのメトリクスを発行できますか?

別のモノに代わってメトリクスを公開することができます。そのためには、そのデバイスの AWS IoT Device Defender 予約トピックにメトリクスを発行します。たとえば、Thing-1 はそのモノ自体、また Thing-2 に代わって、メトリクスを公開するとします。Thing-1 は、そのモノ独自のメトリクスを収集し、MQTT トピックで発行します。

```
$aws/things/Thing-1/defender/metrics/json
```

Thing-1 は、Thing-2 からのメトリクスを取得し、それらのメトリクスを MQTT トピックで発行します。

```
$aws/things/Thing-2/defender/metrics/json
```

Q: アカウントにはいくつのセキュリティプロファイルと動作を保有できますか?

A: AWS IoT Device Defender エンドポイントとクォータを参照してください。

Q: アラートターゲットの典型的なターゲットロールはどのようなものですか?

A: AWS IoT Device Defender がアラートターゲット (SNS トピック) でアラートを発行できるようにするロールには、以下の 2 つが必要です。

- 信頼されたエンティティとして iot.amazonaws.com 指定する信頼関係。
- 指定された SNS トピックで発行する AWS IoT アクセス許可を付与するアタッチされたポリシー。例 :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "<sns-topic-arn>"
        }
```

```
    ]
}
```

- アラートを発行するために使用される SNS トピックが暗号化されたトピックである場合、SNS トピックに発行するためのアクセス許可とともに、AWS IoT にさらに 2 つのアクセス許可を付与する必要があります。例 :

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "sns:Publish",
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": "<sns-topic-arn>"
       }
    ]
}
```

Q: カスタムメトリクスタイプ number を使用したメトリクスレポートの送信時に、エラーメッセージ `Malformed metrics report` が表示されて失敗します。何が問題なのでしょうか?

A: タイプ number は入力として単一のメトリクス値のみを取りますが、DeviceMetrics レポートでメトリック値を送信するときは、単一の値を持つ配列として渡す必要があります。メトリクス値を配列として送信していることを確認します。

ペイロードエラー:

```
{"header":{"report_id":12334567,"version":"1.0"},"metrics":{"network_stats":
{"bytes_in":30680,"bytes_out":10652,"packets_in":113,"packets_out":118}},"custom_metrics":
{"my_custom_metric":{"number":0}}}
```

エラーメッセージ :

```
{"thingName":"myThing","status":"REJECTED","statusDetails":
{"ErrorCode":"InvalidPayload","ErrorMessage":"Malformed metrics
 report"},"timestamp":1635802047699}
```

エラーのないペイロード :

```
{"header":{"report_id":12334567,"version":"1.0"},"metrics":{"network_stats":
{"bytes_in":30680,"bytes_out":10652,"packets_in":113,"packets_out":118}},"custom_metrics":
{"my_custom_metric":[{"number":0}]}}
```

レスポンス：

```
{"thingName":"myThing","12334567":1635800375,"status":"ACCEPTED","timestamp":1635801636023}
```

AWS IoT Device Defender のセキュリティ

AWS でのクラウドセキュリティは最優先事項です。AWS のユーザーは、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を利用できます。

セキュリティは、AWS とユーザーの間の責任共有です。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のセキュリティとして説明しています。

- クラウドのセキュリティ - AWS は、AWS クラウド で AWS の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。また、AWS は、ユーザーが安全に使用できるサービスも提供します。AWSコンプライアンスプログラムg11AWSコンプライアンスプログラム/g11g10AWSコンプライアンスプログラム/g10の一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。AWS IoT Device Defender に適用す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による対象範囲内の AWS のサービス」「」を参照してください。
- クラウド内のセキュリティ - ユーザーの責任は、使用する AWS のサービスに応じて異なります。また、お客様は、データの機密性、お客様の会社の要件、および適用される法律および規制など、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、AWS IoT Device Defender を使用する際に責任共有モデルを適用する方法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の目的を達成するために AWS IoT Device Defender を設定する方法を示します。また、AWS IoT Device Defender リソースのモニタリングや保護に役立つ、その他 AWS サービスの使用方法についても説明します。AWS IoT Core のセキュリティの詳細については、「AWS IoT Core デベロッパーガイド」のセキュリティに関する章を参照してください。

トピック

- AWS IoT Device Defender におけるデータ保護
- AWS IoT Device Defender のためのアイデンティティおよびアクセス管理
- AWS IoT Device Defender のコンプライアンス検証
- AWS IoT Device Defender のレジリエンス

AWS IoT Device Defender におけるデータ保護

AWS IoT Device Defender におけるデータ保護には、AWS 責任共有モデルが適用されます。このモデルで説明されているように、AWS は、AWS クラウド のすべて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。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AWS 責任共有モデルおよび GDPRのブログ記事を参照してください。

データを保護するため、AWS アカウント 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セットアップ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。
- AWS CloudTrail で API とユーザーアクティビティロギングをセットアップします。CloudTrail 証跡を使用して AWS アクティビティをキャプチャする方法については、「AWS CloudTrail ユーザーガイド」の「CloudTrail 証跡の使用」を参照してください。
- AWS のサービス 内のすべてのデフォルトセキュリティ管理に加え、AWS 暗号化ソリューションを使用します。
- 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 を使用して AWS にアクセスする際に FIPS 140-3 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) 140-3」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、AWS IoT Device Defender で作業する場合や、コンソール、API、AWS CLI、または AWS SDK を使用しているその他の AWS のサービス で作業する場合も同様です。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。

AWS IoT Device Defender のためのアイデンティティおよびアクセス管理

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するために役立つ AWS のサービスです。IAM 管理者は、AWS IoT Device Defender リソースの使用を認証 (サインイン) し、認可 (アクセス許可を持つ) できるユーザーを制御します。IAM は、追加費用なしで使用できる AWS のサービスです。

トピック

対象者

AWS IoT Device Defender で行う作業によって異なる AWS Identity and Access Management (IAM) の使用方法。

サービスユーザー - ジョブを実行するために AWS IoT Device Defender サービスを使用する場合は、管理者が必要なアクセス許可と認証情報を用意します。作業を実行するためにさらに多くの AWS IoT Device Defender 機能を使用するとき、追加のアクセス許可が必要になる場合があります。アクセスの管理方法を理解すると、管理者から適切な権限をリクエストするのに役に立ちます。AWS IoT Device Defender の機能にアクセスできない場合は、「AWS IoT Device Defender ID 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 AWS IoT Device Defender リソースを担当している場合は、おそらく AWS IoT Device Defender へのフルアクセスがあります。サービスのユーザーがどの AWS IoT Device Defender 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。お客様の会社で AWS IoT Device Defender で IAM を利用する方法の詳細については、「AWS IoT Device Defender と IAM の連携方法」を参照してください。

IAM 管理者 - 管理者は、AWS IoT Device Defender 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 AWS IoT Device Defender の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、「AWS IoT Device Defender の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。

アイデンティティによる認証

認証とは、アイデンティティ認証情報を使用して AWSにサインインする方法です。ユーザーは、AWS アカウントのルートユーザー、IAM ユーザーとして、または IAM ロールを引き受けることによって、認証される (AWSにサインインする) 必要があります。

ID ソースから提供された認証情報を使用して、フェデレーティッドアイデンティティとして AWS にサインインできます。AWS IAM Identity Centerフェデレーティッドアイデンティティの例としては、(IAM アイデンティティセンター) ユーザー、貴社のシングルサインオン認証、Google または Facebook の認証情報などがあります。フェデレーティッド ID 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。フェデレーションを使用して AWSにアクセスする場合、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。AWS へのサインインの詳細については、AWS サインインユーザーガイドの「AWS アカウントにサインインする方法」を参照してください。

プログラムを使用して AWS にアクセスする場合、AWSは Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) 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。リクエストに自分で署名する際に推奨される方法の使用の詳細については、「IAM ユーザーガイド」の「API リクエストに対する AWS Signature Version 4」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 は、アカウントのセキュリティを強化するために多要素認証 (MFA) を使用す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」および「IAM ユーザーガイド」の「IAM の AWS 多要素認証」を参照してください。

AWS アカウント のルートユーザー

AWS アカウント を作成する場合は、このアカウントのすべての AWS のサービス とリソースに対して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。このアイデンティティは 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によってアクセスできます。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対し、ID プロバイダーとのフェデレーションを使用して、一時的な認証情報の使用により、AWS のサービスにアクセスすることを要求します。

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、AWS Directory Service、Identity Center ディレクトリのユーザーか、または ID ソースから提供された認証情報を使用して AWS のサービス にアクセスするユーザーです。フェデレーティッド ID が AWS アカウント にアクセスすると、ロールが継承され、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 アイデンティティセンター でユーザーとグループを作成するか、すべての AWS アカウント とアプリケーションで使用するために、独自の ID ソースで一連のユーザーとグループに接続して同期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、1 人のユーザーまたは 1 つのアプリケーションに対して特定の許可を持つ AWS アカウント 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、一時的な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「Rotate access keys regularly for use cases that require long-term credentials」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許可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 ユーザーのユースケース」を参照してください。

IAM ロール

IAM ロールは、特定の許可を持つ、AWS アカウント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。AWS Management Consoleで IAM ロールを一時的に引き受けるには、ユーザーから IAM ロール (コンソール) に切り替えます。ロールを引き受けるには、AWS CLI または AWS API 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「ロールを引き受けるための各種方法」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 ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールについては、「IAM ユーザーガイド」の「サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する」を参照してください。IAM Identity Center を使用する場合は、許可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。アクセス許可セットの詳細については、「AWS IAM Identity Center User Guide」の「Permission sets」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 AWS のサービス では、(ロールをプロキシとして使用する代わりに) リソースにポリシーを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 でのクロスアカウントのリソースへのアクセス」を参照してください。
- クロスサービスアクセス権 - 一部の AWS のサービス では、他の AWS のサービス の機能を使用します。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許可、サービスロール、またはサービスリンク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 AWS でアクションを実行するユーザーは、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、AWS のサービスを呼び出すプリンシパルの権限を、AWS のサービスのリクエストと合わせて使用し、ダウンストリームのサービスに対してリクエストを行います。FAS リクエストは、サービスが、完了するために他の AWS のサービス またはリソースとのやりとりを必要とするリクエストを受け取ったときにのみ行われます。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「AWS のサービス に許可を委任するロールを作成する」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、AWS のサービス にリンクされたサービスロールの一種です。サービスがロールを引き受け、ユーザーに代わってアクションを実行できるようになります。サービスにリンクされたロールは、AWS アカウント に表示され、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許可を表示できますが、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 - EC2 インスタンスで実行され、AWS CLI または AWS API 要求を行っているアプリケーションの一時的な認証情報を管理するには、IAM ロールを使用できます。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、インスタンスに添付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 ロールを使用して許可を付与する」を参照してください。

ポリシーを使用したアクセス権の管理

AWS でアクセスを制御するには、ポリシーを作成して AWS ID またはリソースにアタッチします。ポリシーは AWS のオブジェクトであり、アイデンティティやリソースに関連付けて、これらのアクセス許可を定義します。AWS 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) がリクエストを行うと、これらのポリシーを評価します。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。大半のポリシーは JSON ドキュメントとして AWS

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 AWSJSON ポリシーを使用して、だれ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、アクションの許可を定義します。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。このポリシーがあるユーザーは、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 ユーザーグループ、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 許可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「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

アイデンティティベースのポリシーは、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ます。マネージドポリシーは、AWS アカウント内の複数のユーザー、グループ、およびロールにアタッチできるスタンドアロンポリシーです。マネージドポリシーには、AWS マネージドポリシーとカスタマー管理ポリシーがあります。管理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「管理ポリシーとインラインポリシーのいずれかを選択する」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、フェデレーションユーザー、または AWS のサービス を含めることができます。

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は使用できません。

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許可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 S3、AWS WAF、および Amazon VPC は、ACL をサポートするサービスの例です。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 では、他の一般的ではないポリシータイプをサポートしています。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。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可境界」を参照してください。
- サービスコントロールポリシー (SCP) - SCP は、AWS Organizations で組織や組織単位 (OU) の最大許可を指定する JSON ポリシーです。AWS Organizations は、顧客のビジネスが所有する複数の AWS アカウント をグループ化し、一元的に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 はメンバーアカウントのエンティティに対する権限を制限します (各 AWS アカウントのルートユーザー など)。Organizations と SCP の詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー (SCP)」を参照してください。
- リソースコントロールポリシー (RCP) – RCP は、所有する各リソースにアタッチされた IAM ポリシーを更新することなく、アカウント内のリソースで利用可能な最大のアクセス許可を設定するために使用できる JSON ポリシーです。RCP は、メンバーアカウントのリソースの許可を制限し、組織に属するかどうかにかかわらず、AWS アカウントのルートユーザー を含む ID のための有効な許可に影響を及ぼす可能性があります。Organizations と RCP の詳細 (RCP をサポートす

る AWS のサービスのリストを含む) については、「AWS Organizations ユーザーガイド」の「リソースコントロールポリシー (RCP)」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連するとき、リクエストを許可するかどうかを AWS が決定する方法の詳細については、IAM ユーザーガイドのポリシーの評価ロジックを参照してください。

AWS IoT Device Defender と IAM の連携方法

IAM を使用して AWS IoT Device Defender へのアクセスを管理する前に、AWS IoT Device Defender で利用できる IAM の機能について学びます。

AWS IoT Device Defender で使用できる IAM の機能

IAM の機能	AWS IoT Device Defender サポート
アイデンティティベースポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	Yes
ポリシー条件キー	Yes
ACL	なし
ABAC (ポリシー内のタグ)	部分的

IAM の機能	AWS IoT Device Defender サポート
一時的な認証情報	あり
プリンシパル権限	あり
サービスロール	あり
サービスリンクロール	いいえ

AWS IoT Device Defender およびその他の AWS のサービスがほとんどの IAM 機能と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「IAM と連携する AWS のサービス」を参照してください。

AWS IoT Device Defender の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のポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザーグループ、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 許可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 ユーザーガイドのカスタマー管理ポリシーでカスタム IAM アクセス許可を定義する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学ぶには、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。

AWS IoT Device Defender のアイデンティティベースのポリシーの例

AWS IoT Device Defender 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、「AWS IoT Device Defender の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。

AWS IoT Device Defender 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: なし

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、フェデレーションユーザー、または AWS のサービス を含めることができます。

クロスアカウントアクセスを有効にするには、全体のアカウント、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 AWS アカウント にある場合、信頼できるアカウントの IAM 管理者は、リソースにアクセスするための権限をプリンシパルエンティティ (ユーザーまたはロール) に付与する必要もあります。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さらに付与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 でのクロスアカウントリソースアクセス」を参照してください。

AWS IoT Device Defender のポリシーアクション

ポリシーアクションのサポート: あり

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対して、どのような条件下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、関連する 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許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数の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。

AWS IoT Device Defender 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」を参照してください。

AWS IoT Device Defender のポリシーアクションは、アクションの前に以下のプレフィックス を使用します。

```
```

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": [
      ":action1",
      ":action2"
         ]
```

AWS IoT Device Defender 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、「AWS IoT Device Defender の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。

AWS IoT Device Defender のポリシーリソース

ポリシーリソースのサポート: あり

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ような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、リソースレベルの許可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": "*"
```

AWS IoT Device Defender リソースのタイプとその ARN のリストを確認するには、「サービス認証リファレンス」を参照してください。どのアクションで、各リソースの ARN を指定することができるかについては、「」を参照してください。

AWS IoT Device Defender 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、「AWS IoT Device Defender の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。

AWS IoT Device Defender 向け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: あり

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 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、AWS では AND 論理演算子を使用してそれらを評価します。単一の条件キーに複数の値を指定する場合、AWS では OR 論理演算子を使用して条件を評価します。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: 変数とタグ」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ています。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。

AWS IoT Device Defender の条件キーのリストを確認するには、「サービス認証リファレンス」を参照してください。どのアクションおよびリソースと条件キーを使用できるかについては、「」を参照してください。

AWS IoT Device Defender 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、「AWS IoT Device Defender の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。

AWS IoT Device Defender の ACL

ACL のサポート: なし

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許可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。

AWS IoT Device Defender を備えた ABAC

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。AWS では、これら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)、および多数の 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オペレーションを許可するように ABAC ポリシーをします。

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ありです。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 認可で属性に基づいてアクセス許可を定義する」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「IAM チュートリアル: タグに基づいて AWS リソースにアクセスするためのアクセス許可を定義する」を参照してください。

AWS IoT Device Defender 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: あり

AWS のサービスには、一時的な認証情報を使用してサインインしても機能しないものがあります。一時的な認証情報を利用できる AWS のサービスを含めた詳細情報については、「IAM ユーザーガイド」の「IAM と連携する AWS のサービス」を参照してください。

ユーザー名とパスワード以外の方法で AWS Management Console にサインインする場合は、一時的な認証情報を使用していることになります。例えば、会社のシングルサインオン (SSO) リンクを使用して AWS にアクセスすると、そのプロセスは自動的に一時認証情報を作成します。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「ユーザーから IAM ロールに切り替える (コンソール)」を参照してください。

一時認証情報は、AWS CLI または AWS API を使用して手動で作成できます。作成後、一時認証情報を使用して AWS にアクセスできるようになります。AWSは、長期的なアクセスキーを使用する代わりに、一時認証情報を動的に生成することをお勧めします。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。

AWS IoT Device Defender のクロスサービスプリンシパル権限

転送アクセスセッション (FAS) のサポート: あり

IAM ユーザーまたはロールを使用して AWS でアクションを実行するユーザーは、プリンシパルとみなされます。一部のサービスを使用する際に、アクションを実行してから、別のサービスの別のアクションを開始することがあります。FAS は、AWS のサービスを呼び出すプリンシパルの権限を、AWS のサービスのリクエストと合わせて使用し、ダウンストリームのサービスに対してリクエストを行います。FAS リクエストは、サービスが、完了するために他の AWS のサービス またはリソースとのやりとりを必要とするリクエストを受け取ったときにのみ行われます。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。

AWS IoT Device Defender のサービスロール

サービスロールのサポート: あり

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 に許可を委任するロールを作成する」を参照してください。

> **Warning**
>
> サービスロールの権限を変更すると、AWS IoT Device Defender の機能が破損する可能性があります。AWS IoT Device Defender 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。

AWS IoT Device Defender のサービスリンクロール

サービスにリンクされたロールのサポート: なし

サービスにリンクされたロールは、AWS のサービスにリンクされているサービスロールの一種です。サービスがロールを引き受け、ユーザーに代わってアクションを実行できるようになります。

サービスにリンクされたロールは、AWS アカウント に表示され、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アクセス許可を表示できますが、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「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。サービスリンク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、「はい」 リンクを選択します。

AWS IoT Device Defender の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、IAM ユーザーとロールには AWS IoT Device Defender リソースの作成や変更を行うアクセス許可はありません。また、AWS Management Console、AWS Command Line Interface (AWS CLI)、または AWS API を使用してタスクを実行することもできません。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継ぐことができます。

これらのサンプルの 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 ポリシーを作成する (コンソール)」を参照してください。

AWS IoT Device Defender 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「Actions, Resources, and Condition Keys for AWS IoT Device Defender」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
- AWS IoT Device Defender コンソールを使用する
- 自分の権限の表示をユーザーに許可する

ポリシーのベストプラクティス

アイデンティティベースのポリシーは、ユーザーのアカウントで誰かが AWS IoT Device Defender 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 マネージドポリシーを使用して開始し、最小特権の許可に移行する – ユーザーとワークロードへの許可の付与を開始するには、多くの一般的なユースケースのために許可を付与する AWS マネージドポリシーを使用します。これらは AWS アカウントで使用できます。ユースケースに応じた AWS カスタマー管理ポリシーを定義することで、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「ジョブ機能の AWS 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 ポリシーで許可を設定する場合は、タスクの実行に必要な許可のみを付与します。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、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、「IAM ユーザーガイド」の「AWS Identity and Access Management でのポリシーとアクセス許可」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。また、AWS CloudFormation などの特定の AWS のサービス を介して使用する場合、条件を使用して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。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 でポリシーを検証する」を参照してください。
- 多要素認証 (MFA) を要求する – AWS アカウントで IAM ユーザーまたはルートユーザーを要求するシナリオがある場合は、セキュリティを強化するために MFA をオンにします。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 を使用した安全な API アクセス」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。

AWS IoT Device Defender コンソールを使用する

AWS IoT Device Defender コンソールにアクセスするには、一連の最小限のアクセス許可が必要です。これらのアクセス許可により、AWS アカウント の AWS IoT Device Defender リソースの詳細

の一覧表示と表示ができます。最小限必要なアクセス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) では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権限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へのアクセスのみを許可します。

ユーザーとロールが引き続き AWS IoT Device Defender 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 AWS IoT Device Defender *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 か AWS API を使用してプログラム的に、このアクションを完了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": "*"
        }
    ]
}
```

AWS IoT Device Defender ID とアクセスのトラブルシューティング

次の情報は、AWS IoT Device Defender と IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

トピック

- AWS IoT Device Defender でアクションを実行する権限がない
- iam:PassRole を実行する権限がありません
- 自分の AWS アカウント 以外のユーザーに AWS IoT Device Defender リソースへのアクセスを許可したい

AWS IoT Device Defender でアクションを実行する権限がない

ある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、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:*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: :GetWidget on resource: my-example-widget
```

この場合、:*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 管理者に問い合わせてください。サインイン認証情報を提供した担当者が管理者です。

iam:PassRole を実行する権限がありません

`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 AWS IoT Device Defender にロールを渡すことができ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やサービスリンクロールを作成せずに、既存のロールをサービスに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。

以下の例のエラーは、`marymajor` という IAM ユーザーがコンソールを使用して AWS IoT Device Defender 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。メアリーには、ロールをサービスに渡す許可がありません。

```
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 管理者に問い合わせてください。サインイン認証情報を提供した担当者が管理者です。

自分の AWS アカウント 以外のユーザーに AWS IoT Device Defender リソース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ユーザーが、リソースへのアクセス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。

- AWS IoT Device Defender 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、「AWS IoT Device Defender と IAM の連携方法」を参照してください。
- 所有している AWS アカウント 全体のリソースへのアクセス権を提供する方法については、IAM ユーザーガイドの所有している別の AWS アカウント アカウントへのアクセス権を IAM ユーザーに提供を参照してください。

- サードパーティーの AWS アカウント にリソースへのアクセス権を提供する方法については、「IAM ユーザーガイド」の「サードパーティが所有する AWS アカウント へのアクセス権を付与する」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へのアクセス (ID フェデレーション)」を参照してください。
- 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使用方法の違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。

AWS IoT Device Defender 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対象で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による対象範囲内の AWS のサービス のサービスをご覧いただき、関心のあるコンプライアンスプログラムを選択してください。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。

AWS Artifact 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。詳細については、「AWS Artifact でレポートをダウンロードする」を参照してください。

AWS のサービス を使用する際のユーザーのコンプライアンス責任は、ユーザーのデータの機密性や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。AWS では、コンプライアンスに役立つ次のリソースを提供しています。

- セキュリティのコンプライアンスとガバナンス – これらのソリューション実装ガイドでは、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の機能をデプロイする手順を示します。
- HIPAA 対応サービスのリファレンス – HIPAA 対応サービスの一覧が提供されています。すべての AWS のサービスが HIPAA 適格であるわけではありません。
- 「AWS コンプライアンスのリソース」 – このワークブックおよびガイドのコレクションは、顧客の業界と拠点に適用されるものである場合があります。
- AWS Customer Compliance Guide - コンプライアンスの観点から見た責任共有モデルを理解できます。このガイドは、AWS のサービスを保護するためのベストプラクティスを要約したものであり、複数のフレームワーク (米国標準技術研究所 (NIST)、ペイメントカード業界セキュリティ標準評議会 (PCI)、国際標準化機構 (ISO) など) にわたるセキュリティ統制へのガイダンスがまとめられています。

- 「AWS Config デベロッパーガイド」の「ルールでのリソースの評価」 - AWS Config サービスは、自社のプラクティス、業界ガイドライン、および規制に対するリソースの設定の準拠状態を評価します。
- AWS Security Hub – この AWS のサービス は、AWS 内のセキュリティ状態の包括的なビューを提供します。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一覧については、Security Hub のコントロールリファレンスを参照してください。
- Amazon GuardDuty – この AWS のサービスは、環境をモニタリングして、疑わしいアクティビティや悪意のあるアクティビティがないか調べることで、AWS アカウント、ワークロード、コンテナ、データに対する潜在的な脅威を検出します。GuardDuty を使用すると、特定のコンプライアンスフレームワークで義務付けられている侵入検知要件を満たすことで、PCI DSS などのさまざまなコンプライアンス要件に対応できます。
- AWS Audit Manager - この AWS のサービスは、AWS の使用状況を継続的に監査して、リスクの管理方法や、規制および業界標準へのコンプライアンスの管理方法を簡素化するために役立ちます。

AWS IoT Device Defender のレジリエンス

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。AWS リージョン は、低レイテンシー、高スループット、そして高度な冗長ネットワークで接続される物理的に独立、隔離された複数のアベイラビリティーゾーンを提供します。アベイラビリティーゾーンを使用すると、中断することなくゾーン間で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。

AWS リージョン とアベイラビリティーゾーンの詳細については、「AWS グローバルインフラストラクチャ」を参照してください。

AWS グローバルインフラストラクチャに加えて、AWS IoT Device Defender では、データレジリエンスとバックアップのニーズをサポートする複数の機能を提供しています。

AWS IoT Device Defender ユーザーガイドのドキュメント履歴

以下の表は、AWS IoT Device Defender のドキュメントリリースの内容をまとめたものです。

変更	説明	日付
一般提供	これは、AWS IoT Device Defender の最初の一般リリースです。	2023 年 8 月 2 日
AWS IoT Device Defender がデバイスの切断時間のモニタリングのサポートを開始	AWS IoT Device Defender Rule Detect は、各デバイスの切断時間をモニタリングするための新たな切断時間メトリクスをサポートするようになりました。この新たなメトリクスを使用すると、デバイスの接続が切断されている時間を追跡して、期待どおりに動作しているかどうかを確認できます。また、事前に定義したしきい値レベルでアラームを設定し、デバイス接続の問題が続く場合にアラートを受け取ることもできます。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2023 年 7 月 20 日
AWS IoT Device Defender Audit 機能が、IoT ポリシーで設定ミスの可能性を特定	Audit 機能を使用して、欠陥の特定、問題のトラブルシューティング、必要な是正措置を行うことができます。この新	2022 年 12 月 6 日

機能は、デバイスが、意図しないリソースにアクセスできるように許容する許可ステートメントを含む IoT ポリシーを特定するのにも役立ちます。また、ワイルドカードを特定の文字列に置き換えるときにデバイスによって回避される可能性がある拒否ステートメントでの MQTT ワイルドカードの使用も検査します。詳細については、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。

AWS IoT Device Defender ML Detect カスタムメトリクスとディメンションのサポート	AWS IoT Device Defender は、取り消された中間認証機関 (CA) の新しい監査チェックをサポートするようになりました。CA が潜在的な侵害のために中間 CA を取り消すと、その中間 CA によって発行されたすべての証明書も侵害されている可能性があるとみなされ、無効となります。この新しい監査チェックは、取り消された中間 CA によって発行されたアクティブなデバイス証明書を特定し、お客様がこうした証明書を見直し、取り替えるのに役立ちます。詳細については、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2022 年 11 月 10 日

AWS IoT Device Defender ML Detect カスタムメトリクスとディメンションのサポート	ML Detect は、カスタムメトリクスのモニタリングをサポートするようになりました。これにより、フリートに固有の運用状態パラメータを評価できます。Rules Detect で静的アラームを手動で設定することに加えて、機械学習を使用して、カスタムメトリクスでフリートに予想される動作を自動的に学習できるようになりました。さらに、ML Detect で新しいディメンションフィルターがサポートされるようになり、ML セキュリティプロファイルでより正確なメトリクスを評価するための属性を定義できるようになりました。「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」	2022 年 9 月 14 日

AWS IoT Device Management と AWS IoT Device Defender が ListMetricValues API によるデバイスメトリクスのモニタリングのサポートを開始	ListMetricValues API を使用すると、セキュリティプロファイルに属する接続されたデバイスから、デバイス側のメトリクス、クラウド側のメトリクス、カスタムメトリクスの履歴にアクセスできます。AWS IoT 管理コンソールでデータを表示できるだけでなく、プログラムでモニタリングし、独自の視覚化を構築できる柔軟性が備わりました。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2022 年 4 月 5 日
AWS IoT Device Defender が Detect アラーム検証の状態のサポートを開始	検出された動作異常の調査に基づいてアラームを検証します。アラームを真検知、良性の検知、誤検知、または不明として検証し、検証の説明を提供できます。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2021 年 9 月 24 日

AWS IoT Device Defender Audit One-Click のリリース	Audit One-Click を使用すると、ワンクリックでアカウントと IoT デバイスの監査をセキュリティのベストプラクティスに照らして開始できるため、AWS IoT Core のお客様はセキュリティベースラインを簡単に改善できます。Audit One-Click を使用すると、利用可能なすべての監査チェックや日次監査スケジュールの有効化など、プリセット設定で AWS IoT Device Defender 監査を有効にできます。また、定期的なセキュリティ監査の利点について、コンテキストに応じた説明も提供します。Audit One-Click は AWS IoT コンソールからのみ使用できます。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2021 年 9 月 22 日

AWS IoT Device Defender CloudFormation のサポート	AWS IoT Device Defender Rules Detect は、時間をモニタリングするための新たな切断時間メトリクスをサポートするようになりました。AWS IoT Device Defender は、スケジュールされた監査やセキュリティプロファイルなどの AWS IoT Device Defender リソースを、安全で効率的で反復可能な方法で作成および設定するための AWS CloudFormation をサポートするようになりました。AWS IoT Device Defender がサポートする AWS CloudFormation リソースタイプの詳細については、「IoT リソースタイプのリファレンス」を参照してください。	2021 年 3 月 5 日
AWS IoT Device Defender がカスタムメトリクスのサポートを追加	AWS IoT Device Defender を使用して、フリートまたはユースケースに固有の運用状態メトリクスをモニタリングできます。アラートは、Device Defender コンソールで表示することも、AWSSimple Notification Service (SNS) を介して共有することもできます。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2020 年 12 月 15 日

AWS IoT Device Defender が監査所見の抑制を開始	監査結果の抑制機能を使用すると、表示する監査結果を選択し、特定のリソースに関して準拠していない監査結果を無効にすることができます。さらに、監査結果の抑制は、定義した期間について、または無期限に設定できます。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「監査」を参照してください。	2020 年 8 月 12 日
AWS IoT Device Defender がトピックベースのメトリクスモニタリング用のディメンションのサポートを開始	ディメンション機能を使用すると、Device Defender Detect が評価するメトリクスを MQTT トピックでフィルタリングできます。ディメンションは、受信したメッセージ数、メッセージバイトサイズ、送信されたメッセージ数、送信元 IP、認証エラーの数といったクラウド側のメトリクスをサポートします。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2020 年 4 月 2 日

AWS IoT Device Defender ML Detect の一般提供	AWS IoT Device Defender の ML Detect 機能は、過去のデータから学習することで、フリート全体のデバイスレベルの運用およびセキュリティの異常を自動的に検出します。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2020 年 3 月 24 日
AWS IoT Device Defender が Audit 機能に 4 つの新しいチェックを追加	AWS IoT Device Defender Audit を使用して、フリート内のデバイスに対し、過度に許容されているアクセス許可を持っている、365 日以上使用されていないサービスにアクセスしている、ブルートフォース攻撃の影響を受けやすい予測可能な暗号化キーを持つ Debian ベースのオペレーティングシステムで OpenSSL バージョンを使用している、ハッキングの影響を受けやすくなるように RSA キー生成を誤って処理するように識別された Infineon RSA ライブラリバージョンを使用しているかどうかどうかをチェックできます。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「監査」を参照してください。	2019 年 11 月 25 日

AWS IoT Device Defender が監査結果の緩和アクションをサポート	AWS IoT Device Defender は、お客様が緩和アクションを監査結果に適用できる機能をサポートしています。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「監査」を参照してください。	2019 年 8 月 6 日
AWS IoT Device Defender で未登録デバイスの動作のモニタリングをサポート	AWS IoT Core レジストリに登録されていないデバイスの異常な動作を特定できます。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2019 年 5 月 15 日
AWS IoT Device Defender が統計異常検出とデータ視覚化の提供を開始	統計異常検出を使用し、デバイスがパーセンタイルベースのしきい値内にない場合にアラートを受け取ります。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2019 年 2 月 19 日

AWS IoT Device Defender がデバイスの切断時間のモニタリングのサポートを開始	AWS IoT Device Defender は、2 つの追加のクラウド側のメトリクス、接続試行回数、切断回数をサポートするようになりました。ドキュメントについては、「AWS IoT Device Defender デベロッパーガイド」の「クラウド側のメトリクス」を参照してください。	2018 年 12 月 19 日