Entwicklerhandbuch für AWS IoT Device Defender

AWS IoT Device Defender



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS IoT Device Defender: Entwicklerhandbuch für AWS IoT Device Defender

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS IoT Device Defender? 1
Verwenden Sie AWS IoT Device Defender zum ersten Mal? 2
Funktionsweise von AWS IoT Device Defender 2
Features von AWS IoT Device Defender 3
Erste Schritte mit AWS IoT Device Defender 5
Zugehörige Services 5
Zugriff auf AWS IoT Device Defender 6
Preise für AWS IoT Device Defender 6
Erste Schritte mit AWS IoT Device Defender 7
Einrichtung 7
Registrieren Sie sich für ein AWS-Konto 7
Leitfaden für Audits 7
Voraussetzungen 8
Aktivieren von Auditprüfungen 8
Anzeigen von Prüfungsergebnissen 9
Erstellen von Abhilfemaßnahmen für Audits 9
Anwenden von Abhilfemaßnahmen auf Ihre Prüfungsergebnisse 10
Erstellen einer AWS IoT Device Defender Audit-IAM-Rolle (optional) 10
Aktivieren von SNS-Benachrichtigungen (optional) 12
Konfigurieren von Berechtigungen für kundenseitig verwaltete Schlüssel (optional) 13
Aktivieren der Protokollierung (optional) 14
ML Detect-Handbuch 14
Voraussetzungen 15
So verwenden Sie ML Detect auf der Konsole 15
So verwenden Sie ML Detect mit der CLI 33
Anpassen, wann und wie Sie die AWS IoT Device Defender-Prüfungsergebnisse anzeigen 48
Erste Schritte 48
Anpassen Ihrer Prüfungsergebnisse in der Konsole 49
Anpassen Ihrer Prüfungsergebnisse in der CLI 52
Audit 60
Schweregrad des Problems 60
Nächste Schritte 61
Auditprüfungen 61
Zwischenzertifizierungsstelle für aktive Gerätezertifikate gesperrt 62

Das gesperrte Zertifikatstellen-Zertifikat ist immer noch aktiv. 63
Gerätezertifikat geteilt 64
Gerätezertifikat-Schlüsselqualität 66
Qualität der Zertifizierungsstellen–Zertifikatschlüssel 68
Übermäßig permissive nicht-authentifizierte Amazon Cognito-Rolle 70
Übermäßig permissive authentifizierte Cognito-Rolle 78
Übermäßig permissive AWS IoT-Richtlinien 88
Die AWS IoT-Richtlinie ist möglicherweise falsch konfiguriert 95
Zu permissiver Rollenalias 100
Der Rollenalias ermöglicht den Zugriff auf ungenutzte Dienste 102
Zertifizierungsstellen-Zertifikat läuft ab 103
Widersprüchliche MQTT-Client-IDs 104
Gerätezertifikat läuft ab. 105
Prüfung des Gerätezertifikatalters 107
Ein gesperrtes Gerätezertifikat ist weiterhin aktiv 108
Die Protokollierung ist deaktiviert 109
Prüfungsbefehle 110
Verwalten von Prüfungseinstellungen 110
Planen von Audits 118
Ausführen einer On-Demand-Prüfung 132
Verwalten von Prüfungs-Instances 134
Prüfen der Prüfungsergebnisse 144
Unterdrückungen von Prüfergebnissen 153
So funktionieren Unterdrückungen von Prüfergebnissen 154
So verwenden Sie die Unterdrückung von Prüfungsergebnissen auf der Konsole 154
So verwenden Sie die Unterdrückung von Prüfungsergebnissen in der CLI 162
APIs zur Suche nach Unterdrückungen von Prüfungsergebnissen 164
Detect 165
Überwachung der Verhaltensweise nicht registrierter Geräte 167
Änderungen an der Verfügbarkeit von Detect 167
Alternativen zu AWS IoT Device Defender Detect 168
Häufig gestellte Fragen 170
Anwendungsfälle für Sicherheit 171
Anwendungsfälle auf der Cloud-Seite 171
Geräteseitige Anwendungsfälle 174
Konzepte 179

Verhaltensweisen ... 181
ML Detect ... 185
Anwendungsfälle von ML Detect ... 186
So funktioniert ML Detect ... 186
Mindestanforderungen ... 187
Einschränkungen ... 187
Markierung von Fehlalarmen und anderen Bestätigungszuständen in Alarmen ... 188
Unterstützte Metriken ... 188
Service Quotas ... 189
CLI-Befehle von ML Detect ... 189
ML Detect APIs ... 190
Anhalten oder Löschen eines ML Detect-Sicherheitsprofils ... 190
Benutzerdefinierte Metriken ... 192
So verwenden Sie benutzerdefinierte Metriken auf der Konsole ... 192
So verwenden Sie benutzerdefinierte Metriken von der CLI ... 195
CLI-Befehle für benutzerdefinierte Metriken ... 200
Benutzerdefinierte Metriken-APIs ... 200
Geräteseitige Metriken ... 201
Ausgehende Bytes (`aws:all-bytes-out`) ... 201
Bytes in (`aws:all-bytes-in`) ... 202
Überwachen der Anzahl an TCP-Ports (`aws:num-listening-tcp-ports`) ... 204
Überwachen der Anzahl an UDP-Ports (`aws:num-listening-udp-ports`) ... 206
Ausgehende Pakete (`aws:all-packets-out`) ... 207
Pakete in (`aws:all-packets-in`) ... 209
Ziel-IPs (`aws:destination-ip-addresses`) ... 210
Überwachen von TCP-Ports (`aws:listening-tcp-ports`) ... 211
Überwachen von UDP-Ports (`aws:listening-udp-ports`) ... 212
Anzahl etablierter TCP-Verbindungen (`aws:num-established-tcp-connections`) ... 212
Spezifikationen für Gerätemetriken ... 214
Senden von Metriken von Geräten ... 223
Cloudseitige Metriken ... 224
Nachrichtengröße (aws:message-byte-size) ... 225
Gesendete Nachrichten (aws:num-messages-sent) ... 226
Empfangene Nachrichten (aws:num-messages-received) ... 228
Autorisierungsfehler (aws:num-authorization-failures) ... 229
Quell-IP (aws:source-ip-address) ... 231

Verbindungsversuche (aws:num-connection-attempts) 232
Unterbricht die Verbindung (aws:num-disconnects) 233
Dauer der Verbindung (aws:disconnect-duration) 235
Detect-Metrikexport 235
So funktioniert der Detect-Metrikenexport 237
Schema zum Exportieren von Metriken 238
Preisgestaltung für den Detect-Metrikexport 239
Berechtigungen 240
Einrichten des Detect-Metrikenexports auf der AWS IoT-Konsole 242
Erstellen eines Sicherheitsprofils zum Aktivieren des Metrikexports 244
Erstellen eines Sicherheitsprofils zum Aktivieren des Metrikexports (CLI) 245
Aktualisieren eines Sicherheitsprofils zum Deaktivieren des Metrikexports (CLI) 247
CLI-Befehle für den Export von Metriken 248
API-Operationen für den Metrikexport 248
Bereichsbestimmung für Metriken in Sicherheitsprofilen mithilfe von Dimensionen 248
So verwenden Sie Dimensionen in der Konsole 249
So verwenden Sie Dimensionen in AWS CLI 251
Berechtigungen 256
Gewähren AWS IoT Device Defender der Berechtigung zum Veröffentlichen von Warnungen in einem SNS-Thema 256
Detect-Befehle 258
Funktionsweise von AWS IoT Device Defender Detect 260
Abschwächungsaktionen 264
Abschwächungsaktionen für Audits 264
Detect-Abschwächungsaktionen 269
Verfahren zum Definieren und Verwalten von Abschwächungsaktionen 269
Erstellen von Abschwächungsaktionen 269
Anwenden von Abschwächungsaktionen 271
Berechtigungen 278
Befehle für Abschwächungsaktionen 284
Verwenden von AWS IoT Device Defender mit anderen AWS-Services 286
Verwenden von AWS IoT Device Defender auf Geräten, die AWS IoT Greengrass ausführen .. 286
Verwenden von AWS IoT Device Defender mit FreeRTOS und eingebetteten Geräten 286
Verwenden von AWS IoT Device Defender mit AWS IoT Device Management 287
Integration in Security Hub CSPM 287
Aktivieren und Konfigurieren der Integration 288

So sendet AWS IoT Device Defender Erkenntnisse an Security Hub CSPM 288
Typische Erkenntnis von AWS IoT Device Defender 291
So geben Sie an, dass keine Erkenntnisse mehr von AWS IoT Device Defender an Security Hub CSPM gesendet werden 296
Serviceübergreifende Confused-Deputy-Prävention 296
Bewährte Sicherheitsmethoden für Geräteagenten 298
AWS IoT Device DefenderAnleitung zur Fehlerbehebung in 301
Sicherheit 307
Datenschutz 308
Identity and Access Management 309
Zielgruppe 309
Authentifizierung mit Identitäten 310
Verwalten des Zugriffs mit Richtlinien 311
Funktionsweise von AWS IoT Device Defender mit IAM 313
Beispiele für identitätsbasierte Richtlinien 319
Fehlerbehebung 322
Compliance-Validierung 324
Ausfallsicherheit 324
Dokumentverlauf 326

Was ist AWS IoT Device Defender?

Verwenden Sie AWS IoT Device Defender, einen Sicherheits- und Überwachungsservice, mit dem Sie die Konfiguration Ihrer Geräte prüfen, vernetzte Geräte überwachen und Sicherheitsrisiken minimieren können. Mit AWS IoT Device Defender können Sie konsistente Sicherheitsrichtlinien in Ihrer AWS-IoT-Geräteflotte durchsetzen und schnell reagieren, wenn Geräte gefährdet sind. IoT-Flotten können aus einer großen Anzahl von Geräten mit unterschiedlichsten Funktionen bestehen, sind langlebig und geografisch verteilt. Aufgrund dieser Merkmale ist die Flotteneinrichtung komplex und fehleranfällig. Da Geräte bezüglich Rechenleistung, Arbeitsspeicher und Speicherkapazitäten eingeschränkt sind, können Verschlüsselung und andere Formen der Sicherheit auf den Geräten selbst nur begrenzt eingesetzt werden.

Geräte verwenden häufig Software mit bekannten Schwachstellen. Diese Faktoren machen IoT-Flotten zu einem sicheren Ziel für Hacker und erschweren eine anhaltende Sicherung Ihrer Geräteflotte. AWS IoT Device Defender bewältigt diese Herausforderungen, indem es Tools bereitstellt, um Sicherheitsprobleme und Abweichungen von bewährten Methoden zu identifizieren. AWS IoT Device Defender kann Geräteflotten prüfen, um sicherzustellen, dass sie die bewährten Sicherheitsmethoden einhalten, und ungewöhnliches Verhalten auf Geräten erkennen. Das folgende Diagramm zeigt die grundlegende Architektur von AWS IoT Device Defender und wie sie sich zu Services wie AWS IoT Core, Amazon CloudWatch und Amazon SNS verhält.



Themen

- Verwenden Sie AWS IoT Device Defender zum ersten Mal?
- Funktionsweise von AWS IoT Device Defender

- Features von AWS IoT Device Defender
- Erste Schritte mit AWS IoT Device Defender
- Zugehörige Services
- Zugriff auf AWS IoT Device Defender
- Preise für AWS IoT Device Defender

Verwenden Sie AWS IoT Device Defender zum ersten Mal?

Wenn Sie AWS IoT Device Defender zum ersten Mal verwenden, empfehlen wir Ihnen, dass Sie zunächst die folgenden Abschnitte lesen:

- Funktionsweise von AWS IoT Device Defender
- Features von AWS IoT Device Defender
- Erste Schritte mit AWS IoT Device Defender
- Zugehörige Services
- Zugriff auf AWS IoT Device Defender
- Preise für AWS IoT Device Defender

Funktionsweise von AWS IoT Device Defender

AWS IoT Device Defender ist ein vollständig verwalteter Sicherheits- und Überwachungsservice, der Sie beim Schutz Ihrer IoT-Geräteflotte unterstützt. AWS IoT Device Defender prüft Ihren Geräten zugeordnete IoT-Ressourcen, um sicherzustellen, dass sie den bewährten Sicherheitsmethoden entsprechen. Audit-Prüfungen geben bei erkannten Sicherheitsrisiken Warnungen aus und stellen für die Problembehebung relevante Informationen zur Verfügung. AWS IoT Device Defender überwacht auch kontinuierlich Sicherheitsmetriken aus der Cloud und von Geräten, um unerwartetes Verhalten zu erkennen und mögliche kompromittierte Geräte zu identifizieren. Sie können Auditprüfungen auf Abruf oder nach Plan starten, um Ihre IoT-Gerätekonfigurationen zu bewerten.

AWS IoT Device Defender arbeitet mit AWS IoT Core zusammen, um den Kontext von Geräteinteraktionen zu integrieren und die Genauigkeit von Prüfungen zu erhöhen. AWS IoT Device Defender sammelt und analysiert hochwertige Sicherheitsmetriken von Ihren verbundenen Geräten, um anormales Verhalten zu erkennen. Wenn Sie Rules Detect verwenden, werden die Metrikdaten kontinuierlich anhand benutzerdefinierter Verhaltensweisen ausgewertet. Wenn Sie ML Detect

verwenden, werden die Metrikdaten kontinuierlich von automatisch erstellten Machine Learning (ML)-Modellen ausgewertet, um Anomalien zu identifizieren.

Die Ergebnisse geplanter Prüfungsaufgaben und aller erkannten anomalen Geräteaktivitäten werden in der AWS-IoT-Konsole und AWS IoT Device Defender API veröffentlicht. Sie sind über Amazon CloudWatch zugänglich. Darüber hinaus können Sie AWS IoT Device Defender so konfigurieren, dass Ergebnisse zur Integration in Sicherheits-Dashboards oder zum Starten automatisierter Korrektur-Workflows an Amazon-SNS-Themen gesendet werden.

AWS IoT Device Defender unterstützt eine Vielzahl von Anwendungsfällen, darunter die folgenden:

- Schutz Ihrer Geräte: Sie können Ihre gerätebezogenen Ressourcen anhand bewährter AWS-IoT-Sicherheitsmethoden überprüfen, um Schwachstellen von Geräten zu erkennen. AWS IoT Device Defender-Prüfungen können Ihnen helfen, Risiken für Ihre Geräte zu identifizieren und aufzudecken und zu überprüfen, dass entsprechende Sicherheitsmaßnahmen etabliert wurden.
- Erkennung ungewöhnlichen Geräteverhaltens: Sie können veränderte Verbindungsmuster erkennen, Gerätekommunikation mit nicht autorisierten Endpunkten aufdecken und Änderungen an ein- und ausgehenden Geräteverkehrsmustern identifizieren.
- Einblicke zur Risikominderung: Sie können Maßnahmen ergreifen, um Probleme zu beheben, die in einem Prüfungsergebnis oder einem Detect-Alarm aufgedeckt wurden.
- Aufrechterhaltung der Gerätesicherheit: Sie können Erkenntnisse aus Audit- und Detect-Prüfungen verwenden, um mögliche Sicherheitsverstöße zu diagnostizieren und zu beheben.
- Verbesserte Gerätesicherheit: Sie können ein falsch konfiguriertes Gerät erkennen, den Zustand Ihrer Geräteflotten untersuchen und unerwartete Metriken zum Geräteverhalten finden.

Features von AWS IoT Device Defender

Im Folgenden finden Sie einige der wichtigsten Features von AWS IoT Device Defender.

Wichtigste Funktionen

Audit	AWS IoT Device Defender überprüft Ihre gerätebezogenen Ressourcen anhand bewährter AWS-IoT-Sicherheitsmethoden. Im IAM-Benutzerhandbuch meldet AWS IoT

	Device Defender Konfigurationen, die die bewährten Sicherheitsmethoden nicht erfüllen, z. B. übermäßig freizügige Richtlinien, mit denen ein Gerät Daten für viele andere Geräte lesen und aktualisieren kann.
Rules Detect	AWS IoT Device Defender erkennt ungewöhnliches Geräteverhalten, das auf eine Kompromittierung hinweisen kann, indem hochwertige Sicherheitsmetriken vom Gerät und AWS IoT Core kontinuierlich überwacht werden. Sie können das normale Geräteverhalten für eine Gruppe von Geräten angeben, indem Sie Verhaltensweisen (Regeln) für diese Metriken einrichten. AWS IoT Device Defender überwacht und wertet jeden für diese Metriken gemeldeten Datenpunkt anhand benutzerdefinierter Verhaltensweisen (Regeln) aus und warnt Sie, wenn eine Anomalie erkannt wird.
ML Detect	AWS IoT Device Defender legt automatisch das Geräteverhalten für Sie mit Machine Learning (ML)-Modellen fest, die Gerätedaten über sechs cloudseitige Metriken und sieben geräteseitige Metriken der letzten 14 Tage verwenden. Anschließend werden die Modelle jeden Tag neu trainiert (sofern genügend Daten zum Trainieren des Modells vorliegen), um das erwartete Geräteverhalten auf der Grundlage der letzten 14 Tage nach der Erstellung der Modelle zu aktualisieren. AWS IoT Device Defender überwacht und identifiziert anomale Datenpunkte für diese Metriken mit den ML-Modellen und löst einen Alarm aus, wenn eine Anomalie erkannt wird.

Warnfunktion	AWS IoT Device Defender veröffentlicht Alarme in der AWS-IoT-Konsole, Amazon CloudWatch und Amazon SNS.
Abhilfe	AWS IoT Device Defender kann verwendet werden, um Probleme zu untersuchen, indem kontextbezogene und historische Informati onen über das Gerät bereitgestellt werden, z. B. Gerätemetadaten, Gerätestatistiken und historische Warnungen für das Gerät. Sie können auch in AWS IoT Device Defender integrierte Abschwächungsaktionen vornehmen , um Abschwächungsschritte für Audit- und Detect-Alarme durchzuführen, z. B. Hinzufügen von Objekten zu einer Objektgruppe, Ersetzen der Standardrichtlinienversion und Aktualisieren des Gerätezertifikats.

Erste Schritte mit AWS IoT Device Defender

Die folgenden Tutorials helfen Ihnen bei den ersten Schritten mit AWS IoT Device Defender.

- Einrichtung
- ML-Detect-Handbuch
- Audit-Leitfaden
- Anpassen, wann und wie Sie die AWS IoT Device Defender-Prüfungsergebnisse anzeigen

Zugehörige Services

- AWS IoT Greengrass: AWS IoT Greengrass bietet eine vordefinierte Integration in AWS IoT Device Defender zur kontinuierlichen Überwachung des Geräteverhaltens.

- AWS IoT Device Management: Sie können die Flottenindizierung von AWS IoT Device Management verwenden, um Ihre AWS IoT Device Defender-Detect-Verstöße zu indizieren, zu durchsuchen und zu aggregieren.

Zugriff auf AWS IoT Device Defender

Sie können die AWS IoT Device Defender-Konsole oder die API verwenden, um auf AWS IoT Device Defender zuzugreifen.

Preise für AWS IoT Device Defender

Mit AWS IoT Device Defender zahlen Sie nur für das, was Sie tatsächlich nutzen. Es fallen keine Mindestgebühren an und es bestehen eine Mindestnutzungsanforderungen für den Service. Die Audit- und Detect-Features werden Ihnen jedoch separat in Rechnung gestellt. Die Auditpreise gelten pro Gerät und pro Monat. Wenn Sie Audit aktivieren, werden Ihnen Gebühren basierend auf der Anzahl der aktiven Geräteprinzipale in einem Monat berechnet. Daher hat das Hinzufügen oder Entfernen von Auditprüfungen keine Auswirkungen auf Ihre monatliche Rechnung, wenn Sie diese Funktion verwenden. Sie können Ihre AWS IoT Device Defender und die Architekturkosten mit dem AWS-Preisrechner in einer einzigen Schätzung berechnen.

- AWS Pricing Calculator

Erste Schritte mit AWS IoT Device Defender

Sie können die folgenden Tutorials verwenden, um mit zu arbeiten AWS IoT Device Defender.

Themen

- Einrichtung
- Leitfaden für Audits
- ML Detect-Handbuch
- Anpassen, wann und wie Sie die AWS IoT Device Defender-Prüfungsergebnisse anzeigen

Einrichtung

Führen Sie die folgenden Schritte aus, bevor Sie AWS IoT Device Defender zum ersten Mal verwenden:

Themen

- Registrieren Sie sich für ein AWS-Konto

Registrieren Sie sich für ein AWS-Konto

Für die ersten Schritte mit AWS ist ein AWS-Konto erforderlich. Weitere Informationen zum Erstellen eines AWS-Konto finden Sie unter Erste Schritte mit einem AWS-Konto im AWS -Kontenverwaltung-Referenzhandbuch.

Leitfaden für Audits

Dieses Tutorial enthält Anweisungen zur Konfiguration eines wiederkehrenden Audits, zur Einrichtung von Alarmen, zur Überprüfung der Prüfungsergebnisse und zur Behebung von Prüfungsproblemen.

Themen

- Voraussetzungen
- Aktivieren von Auditprüfungen
- Anzeigen von Prüfungsergebnissen

- Erstellen von Abhilfemaßnahmen für Audits
- Anwenden von Abhilfemaßnahmen auf Ihre Prüfungsergebnisse
- Erstellen einer AWS IoT Device Defender Audit-IAM-Rolle (optional)
- Aktivieren von SNS-Benachrichtigungen (optional)
- Konfigurieren von Berechtigungen für kundenseitig verwaltete Schlüssel (optional)
- Aktivieren der Protokollierung (optional)

Voraussetzungen

Zum Durcharbeiten dieses Tutorials ist Folgendes erforderlich:

- Ein(e) AWS-Konto. Wenn Sie darüber noch nicht verfügen, finden Sie unter Einrichten weitere Informationen.

Aktivieren von Auditprüfungen

Im folgenden Verfahren aktivieren Sie Auditprüfungen, bei denen Konto- und Geräteeinstellungen sowie Richtlinien geprüft werden, um sicherzustellen, dass Sicherheitsmaßnahmen getroffen wurden. In diesem Tutorial weisen wir Sie an, alle Auditprüfungen zu aktivieren; Sie können die Prüfungen jedoch nach Belieben auswählen.

Die Prüfungspreise beziehen sich auf die Anzahl der Geräte pro Monat (mit verbundenen Flottengeräte AWS IoT). Daher hat das Hinzufügen oder Entfernen von Auditprüfungen keine Auswirkungen auf Ihre monatliche Rechnung, wenn Sie diese Funktion verwenden.

1. Öffnen Sie die AWS IoT-Konsole. Wählen Sie im Navigationsbereich Sicherheit und dann Einführung.
2. Wählen Sie AWS IoT Sicherheitsüberprüfung automatisieren. Auditprüfungen sind automatisch aktiviert.
3. Erweitern Sie Audit, und wählen Sie Einstellungen, um Ihre Auditprüfungen einzusehen. Wählen Sie einen Namen für die Auditprüfung aus, um zu erfahren, was die Auditprüfung bewirkt. Weitere Informationen zu Auditprüfungen finden Sie unter Auditprüfungen.
4. (Optional) Wenn Sie bereits über eine Rolle verfügen, die Sie verwenden möchten, wählen Sie Dienstberechtigungen verwalten, wählen Sie die Rolle aus der Liste, und klicken Sie dann auf Aktualisieren.

Anzeigen von Prüfungsergebnissen

Nachstehend wird veranschaulicht, wie Sie Ihre Prüfungsergebnisse einsehen können. In diesem Tutorial sehen Sie die Prüfungsergebnisse der Auditprüfungen, die im Aktivieren von Auditprüfungen Tutorial eingerichtet wurden.

So zeigen Sie die Prüfungsergebnisse an

1. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Audit, und wählen Sie dann Ergebnisse.
2. Wählen Sie den Namen des Prüfplans aus, den Sie untersuchen möchten.
3. Wählen Sie unter Nichtkonforme Prüfungen unter Abhilfe die Informationsschaltflächen aus, um Informationen darüber zu erhalten, warum der Vorgang nicht konform ist. Hinweise darüber, wie Sie Ihre nicht konformen Prüfungen regelkonform gestalten können, finden Sie unter Auditprüfungen.

Erstellen von Abhilfemaßnahmen für Audits

Im folgenden Verfahren erstellen Sie eine AWS IoT Device Defender-Abhilfemaßnahme für Prüfungen, um die AWS IoT-Protokollierung zu aktivieren. Jeder Auditprüfung sind Abhilfemaßnahmen zugeordnet, die sich darauf auswirken, welchen Aktionstyp Sie für die Prüfung wählen, die Sie korrigieren möchten. Weitere Informationen finden Sie unter Abhilfemaßnahmen.

So verwenden Sie die AWS IoT-Konsole zum Erstellen von Abhilfemaßnahmen

1. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Erkennen, und wählen Sie dann Abhilemaßnahmen.
2. Wählen Sie auf der Seite Abhilfemaßnahmen die Option Erstellen.
3. Geben Sie Ihrer Abhilfemaßnahme auf der Seite Neue Abhilfemaßnahme erstellen unter Aktionsname einen eindeutigen Namen, wie beispielsweise *`EnableErrorLoggingAction`*.
4. Wählen Sie als Aktionstyp die Option AWS IoT-Protokollierung aktivieren.
5. Wählen Sie unter Berechtigungen die Option Rolle erstellen. Verwenden Sie als Rollenname *`IoTMitigationActionErrorLoggingRole`*. Wählen Sie dann die Option Erstellen.
6. Wählen Sie unter Parameter unter Rolle für die Protokollierung die Option `IoTMitigationActionErrorLoggingRole`. Wählen Sie als Protokollebene die Option `Error`.

7. Wählen Sie Erstellen.

Anwenden von Abhilfemaßnahmen auf Ihre Prüfungsergebnisse

Nachstehend wird veranschaulicht, wie Sie Abhilfemaßnahmen auf Ihre Prüfungsergebnisse anwenden können.

So wirken Sie nicht konformen Prüfungsergebnissen entgegen

1. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Audit, und wählen Sie dann Ergebnisse.
2. Wählen Sie ein Prüfergebnis aus, auf das Sie reagieren möchten.
3. Überprüfen Sie Ihre Ergebnisse.
4. Wählen Sie Abhilfemaßnahmen starten.
5. Wählen Sie für Protokollierung deaktiviert die Abhilfemaßnahme, die Sie zuvor erstellt haben, `EnableErrorLoggingAction`. Sie können für jedes nicht konforme Ergebnis die entsprechenden Maßnahmen auswählen, um die Probleme zu beheben.
6. Wählen Sie unter Ursachencodes auswählen den Ursachencode aus, der bei der Prüfung zurückgegeben wurde.
7. Wählen Sie Aufgabe starten. Die Ausführung der Abhilfemaßnahmen kann einige Minuten dauern.

So überprüfen Sie, ob die Abhilfemaßnahme funktioniert hat

1. Wählen Sie im Navigationsbereich der AWS IoT-Konsole die Option Einstellungen.
2. Vergewissern Sie sich im Dienstprotokoll, dass die Protokollebene `Error (least verbosity)` ist.

Erstellen einer AWS IoT Device Defender Audit-IAM-Rolle (optional)

Im folgenden Verfahren erstellen Sie eine AWS IoT Device Defender Audit-IAM-Rolle, die AWS IoT Device Defender-Lesezugriff auf AWS IoT bietet.

So erstellen Sie eine Servicerolle für AWS IoT Device Defender (IAM-Konsole)

1. Melden Sie sich bei der AWS-Managementkonsole an, und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich der IAM-Konsole auf Rollen, und wählen Sie dann Rolle erstellen.
3. Wählen Sie den Rollentyp AWS-Service.
4. Wählen Sie unter Anwendungsfälle für andere AWS Dienste verwenden die Option AWS IoT. Wählen Sie dann IoT — Device Defender Audit.
5. Wählen Sie Weiter.
6. (Optional) Legen Sie eine Berechtigungsgrenze fest. Dies ist ein erweitertes Feature, das für Servicerollen verfügbar ist, aber nicht für servicegebundene Rollen.

   Öffnen Sie den Abschnitt Permissions boundary (Berechtigungsgrenze) und wählen Sie Use a permissions boundary to control the maximum role permissions (Eine Berechtigungsgrenze verwenden, um die maximalen Rollen-Berechtigungen zu steuern). IAM enthält eine Liste der von AWS verwalteten und vom Kunden verwaltete Richtlinien in Ihrem Konto. Wählen Sie die Richtlinie aus, die für die Berechtigungsgrenze verwendet werden soll, oder wählen Create policy (Richtlinie erstellen), um eine neue Registerkarte im Browser zu öffnen und eine vollständig neue Richtlinie zu erstellen. Weitere Informationen finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch. Nachdem Sie die Richtlinie erstellt haben, schließen Sie die Registerkarte und kehren zur ursprünglichen Registerkarte zurück, um die Richtlinie auszuwählen, die für die Berechtigungsgrenze verwendet werden soll.
7. Wählen Sie Weiter aus.
8. Geben Sie unter Rollenname einen Rollennamen ein, der Ihnen hilft, den Zweck dieser Rolle zu identifizieren. Rollennamen müssen innerhalb Ihres eindeutig sein AWS-Konto. Es wird hierbei nicht zwischen Groß- und Kleinschreibung unterschieden. z. B. können Sie keine Rollen erstellen, die **`PRODROLE`** bzw. **`prodrole`** heißen. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung nicht bearbeitet werden.
9. (Optional) Geben Sie unter Description (Beschreibung) eine Beschreibung für die neue Rolle ein.
10. Wählen Sie in den Abschnitten Step 1: Select trusted entities (Schritt 1: Vertrauenswürdige Entitäten auswählen) oder Step 2: Add permissions (Schritt 2: Berechtigungen hinzufügen) die Option Edit (Bearbeiten), um die Anwendungsfälle und Berechtigungen für die Rolle zu bearbeiten.

11. (Optional) Fügen Sie dem Benutzer Metadaten hinzu, indem Sie Markierungen als Schlüssel-Wert-Paare anfügen. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Markieren von IAM-Ressourcen im IAM-Benutzerhandbuch.
12. Prüfen Sie die Rolle und klicken Sie dann auf Rolle erstellen.

Aktivieren von SNS-Benachrichtigungen (optional)

Im folgenden Verfahren aktivieren Sie Amazon SNS (SNS)-Benachrichtigungen, die Sie darüber benachrichtigen, wenn bei Ihren Audits nicht-konforme Ressourcen identifiziert werden. In diesem Tutorial richten Sie Benachrichtigungen für die im Aktivieren von Auditprüfungen-Tutorial aktivierten Auditprüfungen ein.

1. Falls Sie dies noch nicht getan haben, fügen Sie eine Richtlinie hinzu, die den Zugriff auf SNS über ermöglicht AWS-Managementkonsole. Folgen Sie dazu den Anweisungen unter Anhängen einer Richtlinie an eine IAM-Benutzergruppe im IAM-Benutzerhandbuch und wählen Sie die Richtlinie AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction aus.
2. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Audit, und wählen Sie dann Einstellungen.
3. Wählen Sie unten auf der Seite mit den Device Defender-Überwachungseinstellungen die Option SNS-Benachrichtigungen aktivieren.
4. Wählen Sie Aktiviert.
5. Wählen Sie Themen und danach Neues Thema erstellen. Geben Sie dem Thema den Namen *`IoTDDNotifications`* und wählen Sie Erstellen. Wählen Sie für Role die Rolle aus, die Sie in Erstellen einer AWS IoT Device Defender Audit-IAM-Rolle (optional) erstellt haben.
6. Wählen Sie Aktualisieren.
7. Wenn Sie E-Mails oder Textnachrichten auf Ihren Ops-Plattformen über Amazon SNS erhalten möchten, finden Sie weitere Informationen unter Verwenden von Amazon Simple Notification Service für Benutzerbenachrichtigungen.

Konfigurieren von Berechtigungen für kundenseitig verwaltete Schlüssel (optional)

> **Note**
>
> Diese Konfiguration ist nur erforderlich, wenn Sie sich für kundenseitig verwaltete Schlüssel für AWS IoT Core entschieden haben. Weitere Informationen zur Verschlüsselung im Ruhezustand in AWS IoT Core finden Sie unter Datenverschlüsselung im Ruhezustand in AWS IoT Core.

Wenn Sie kundenseitig verwaltete Schlüssel (CMK) für die Verschlüsselung im Ruhezustand in AWS IoT Core aktiviert haben, erfordert die von AWS IoT Device Defender Audit verwendete IAM-Rolle zusätzliche Berechtigungen zum Entschlüsseln von Daten. Ohne diese Berechtigungen schlagen Ihre Audit-Operationen fehl.

Die von `AWSIoTDeviceDefenderAudit` verwaltete Richtlinie beinhaltet standardmäßig keine `kms:Decrypt`-Berechtigungen und folgt damit dem Prinzip der geringsten Berechtigung. Sie müssen diese Berechtigungen manuell zu Ihrer Audit-Rolle hinzufügen, wenn Sie kundenseitig verwaltete Schlüssel verwenden.

So fügen Sie KMS-Berechtigungen zu Ihrer IAM-Rolle von AWS IoT Device Defender Audit hinzu

1. Melden Sie sich bei der AWS-Managementkonsole an, und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus und suchen Sie dann nach der Rolle, die Sie in Erstellen einer AWS IoT Device Defender Audit-IAM-Rolle (optional) erstellt haben, oder nach der Rolle, die Sie bei der Konfiguration der Audit-Einstellungen angegeben haben.
3. Wählen Sie den Rollennamen aus, um seine Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Berechtigungen die Option Berechtigungen hinzufügen und dann Inline-Richtlinie erstellen aus.
5. Wählen Sie die Registerkarte JSON aus und geben Sie die folgende Richtlinie ein. Ersetzen Sie die Beispielregion *us-east-1*, die Konto-ID *111122223333* und *KEY_ID* durch Ihre AWS KMS-Schlüsseldetails:

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-east-1:111122223333:key/KEY_ID"
    }
  ]
}
```

6. Wählen Sie Weiter aus.
7. Geben für Richtlinienname einen beschreibenden Namen wie **`DeviceDefenderAuditKMSDecrypt`** ein.
8. Wählen Sie Richtlinie erstellen aus.

Aktivieren der Protokollierung (optional)

In diesem Verfahren wird beschrieben, wie Sie AWS IoT aktivieren, um Informationen in CloudWatch-Protokollen zu protokollieren. Auf diese Weise können Sie Ihre Prüfungsergebnisse einsehen. Durch die Protokollierung können Gebühren anfallen.

So aktivieren Sie die Protokollierung

1. Öffnen Sie die AWS IoT-Konsole. Klicken Sie im Navigationsbereich auf Einstellungen.
2. Wählen Sie unter Protokolle die Option Protokolle verwalten.
3. Wählen Sie unter Rolle auswählen die Option Rolle erstellen. Geben Sie der Rolle den Namen *`awsiotLoggingRole`* und wählen Sie Erstellen. Es wird automatisch eine Richtlinie angehängt.
4. Wählen Sie für Protokollstufe die Option Debuggen (größte Ausführlichkeit).
5. Wählen Sie Aktualisieren.

ML Detect-Handbuch

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect

finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

> **Note**
>
> ML Detect ist in den folgenden Regionen nicht verfügbar:
>
> - Asien-Pazifik (Malaysia)

In diesem Handbuch „Erste Schritte“ erstellen Sie ein ML Detect-Sicherheitsprofil, das Machine Learning (ML) verwendet, um Modelle des erwarteten Verhaltens auf der Grundlage historischer Metrikdaten Ihrer Geräte zu erstellen. ML Detect erstellt das ML-Modell, Sie können den Fortschritt überwachen. Nachdem das ML-Modell erstellt wurde, können Sie kontinuierlich Alarme anzeigen und untersuchen sowie identifizierte Probleme beheben.

Weitere Informationen zu ML Detect und den entsprechenden API- und CLI-Befehlen finden Sie unterML Detect.

Dieses Kapitel enthält die folgenden Abschnitte:

- Voraussetzungen
- So verwenden Sie ML Detect auf der Konsole
- So verwenden Sie ML Detect mit der CLI

Voraussetzungen

- Ein(e) AWS-Konto. Wenn Sie darüber noch nicht verfügen, finden Sie unter Einrichten weitere Informationen.

So verwenden Sie ML Detect auf der Konsole

Tutorials

- Aktivieren von ML Detect
- Überwachen des Status Ihres ML-Modells
- Überprüfen Ihrer ML Detect-Alarme

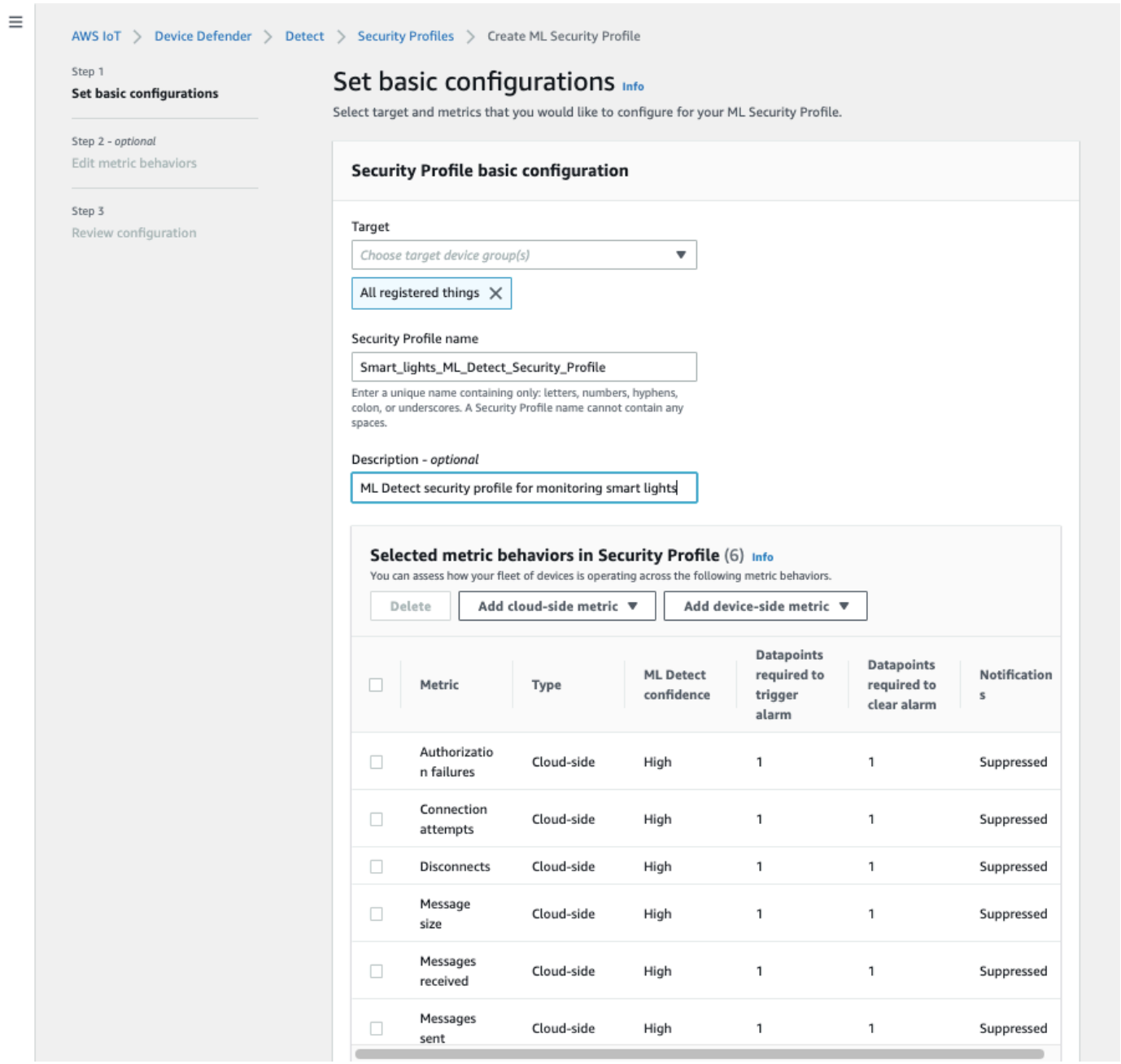
- Optimieren Ihrer ML-Alarme
- Markieren des Bestätigungsstatus Ihres Alarms
- Beseitigen von identifizierten Geräteproblemen

Aktivieren von ML Detect

In den folgenden Verfahren wird detailliert beschrieben, wie ML Detect auf der Konsole eingerichtet wird.

1. Stellen Sie zunächst sicher, dass Ihre Geräte die Mindestdatenpunkte erzeugen, die gemäß den Mindestanforderungen von ML Detect für das kontinuierliche Training und die kontinuierliche Aktualisierung des Modells erforderlich sind. Stellen Sie, damit die Datenerfassung vorangeht, sicher, dass Ihr Sicherheitsprofil an ein Ziel angehängt ist, bei dem es sich um ein Objekt oder eine Objektgruppe handeln kann.
2. Erweitern Sie auf der AWS IoT-Konsole im Navigationsbereich die Option Verteidigen. Wählen Sie Erkennen, Sicherheitsprofile, Sicherheitsprofil erstellen und anschließend Profil zur Erkennung von ML-Anomalien erstellen.
3. Führen Sie auf der Seite Grundlegende Konfigurationen festlegen die folgenden Schritte aus.

   - Wählen Sie Ihre Zielgerätegruppen unter Ziel.
   - Geben Sie unter Sicherheitsprofilname einen Namen für Ihr Sicherheitsprofil ein.
   - (Optional) Unter Beschreibung können Sie eine kurze Beschreibung des ML-Profils eingeben.
   - Wählen Sie unter Ausgewählte Metrikverhalten im Sicherheitsprofil die Metriken, die Sie überwachen möchten.

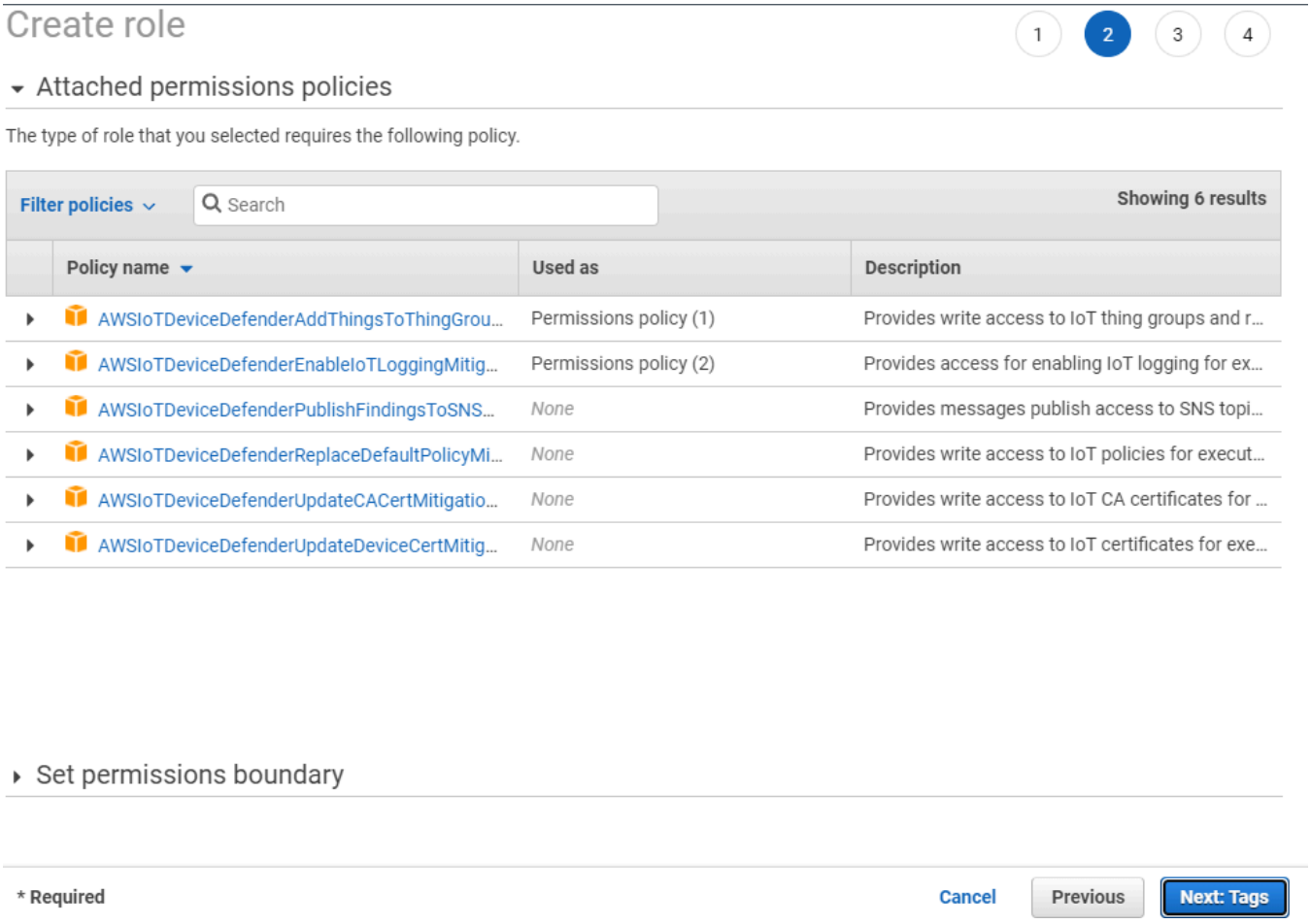


Wählen Sie abschließend Weiter.

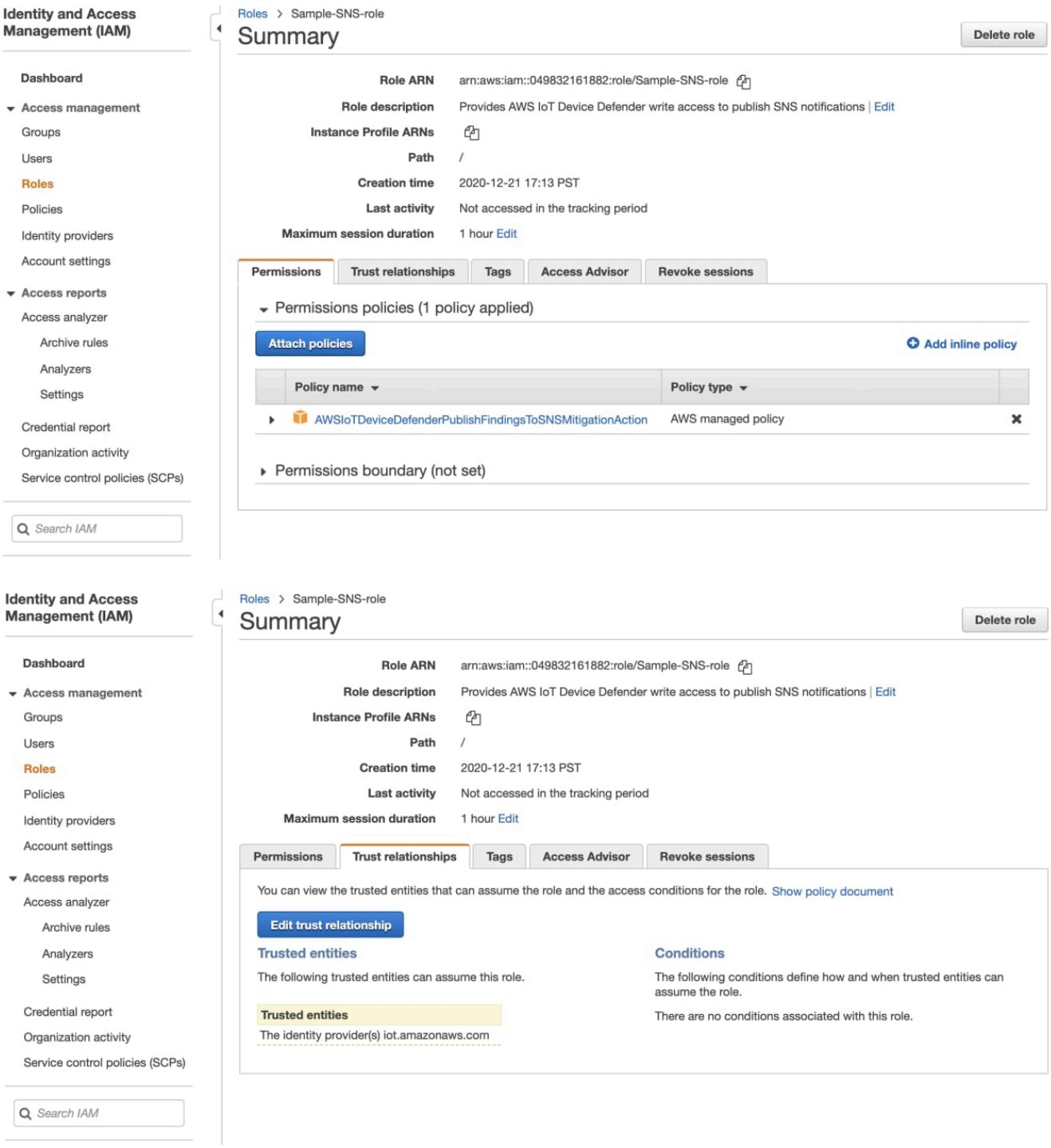
4. Geben Sie auf der Seite SNS einrichten (optional) ein SNS-Thema für Alarmbenachrichtigungen an, wenn ein Gerät gegen ein Verhalten in Ihrem Profil verstößt. Wählen Sie eine IAM-Rolle, die Sie für Veröffentlichungen zum ausgewählten SNS-Thema verwenden möchten.

   Wenn Sie noch keine SNS-Rolle haben, gehen Sie wie folgt vor, um eine Rolle mit den erforderlichen Berechtigungen und Vertrauensbeziehungen zu erstellen.

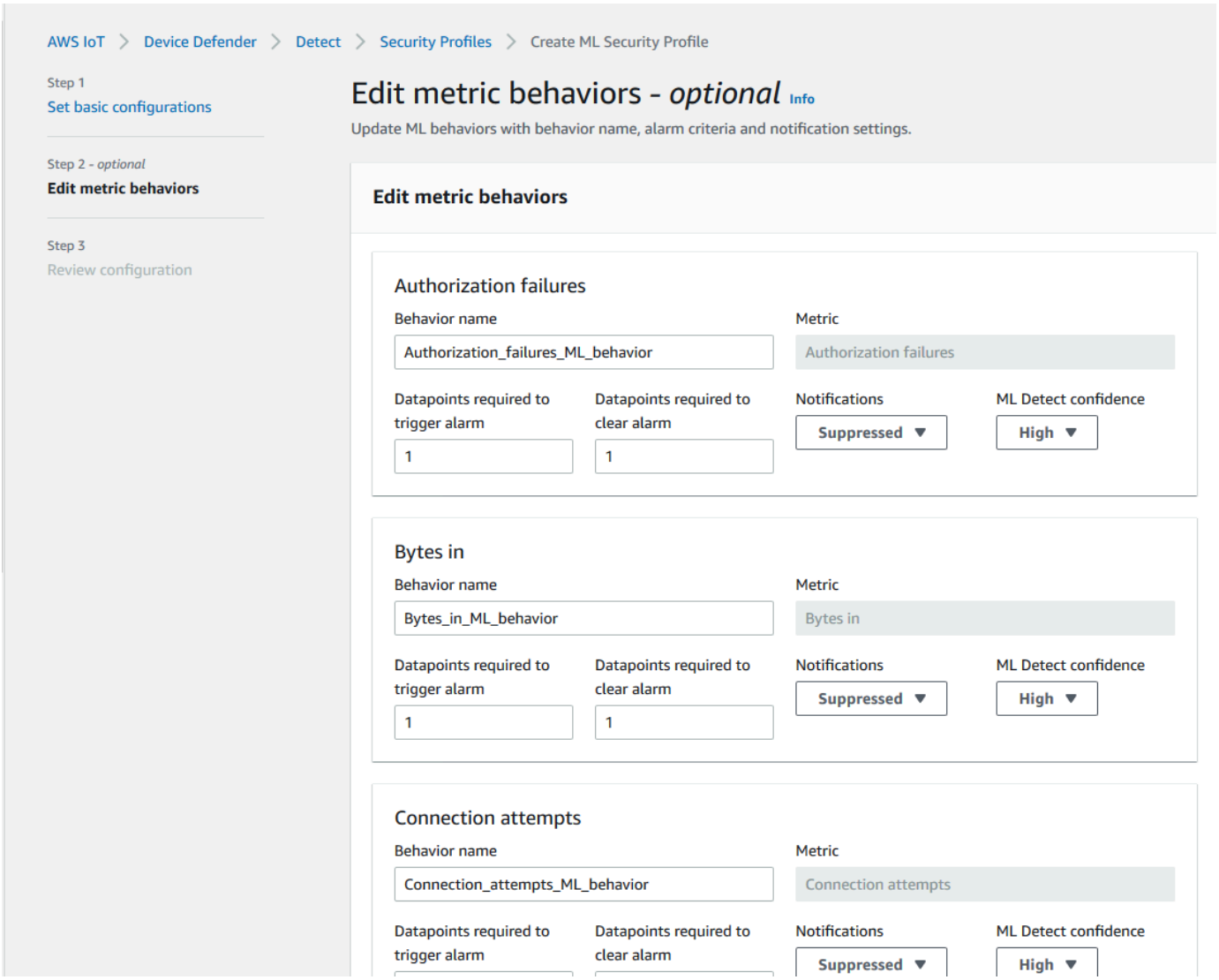
- Navigieren Sie zur IAM-Konsole. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen.
- Wählen Sie unter Typ der vertrauenswürdigen Entität auswählen die Option AWS-Service. Wählen Sie dann unter Anwendungsfall wählen die Option IoT und unter Anwendungsfall auswählen die Option IoT — Device-Defender-Abhilfemaßnahmen. Wählen Sie abschließend Weiter: Berechtigungen.
- Stellen Sie sicher, dass unter Richtlinien für angehängte Berechtigungen die Option AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction ausgewählt ist, und wählen Sie dann Weiter: Tags.



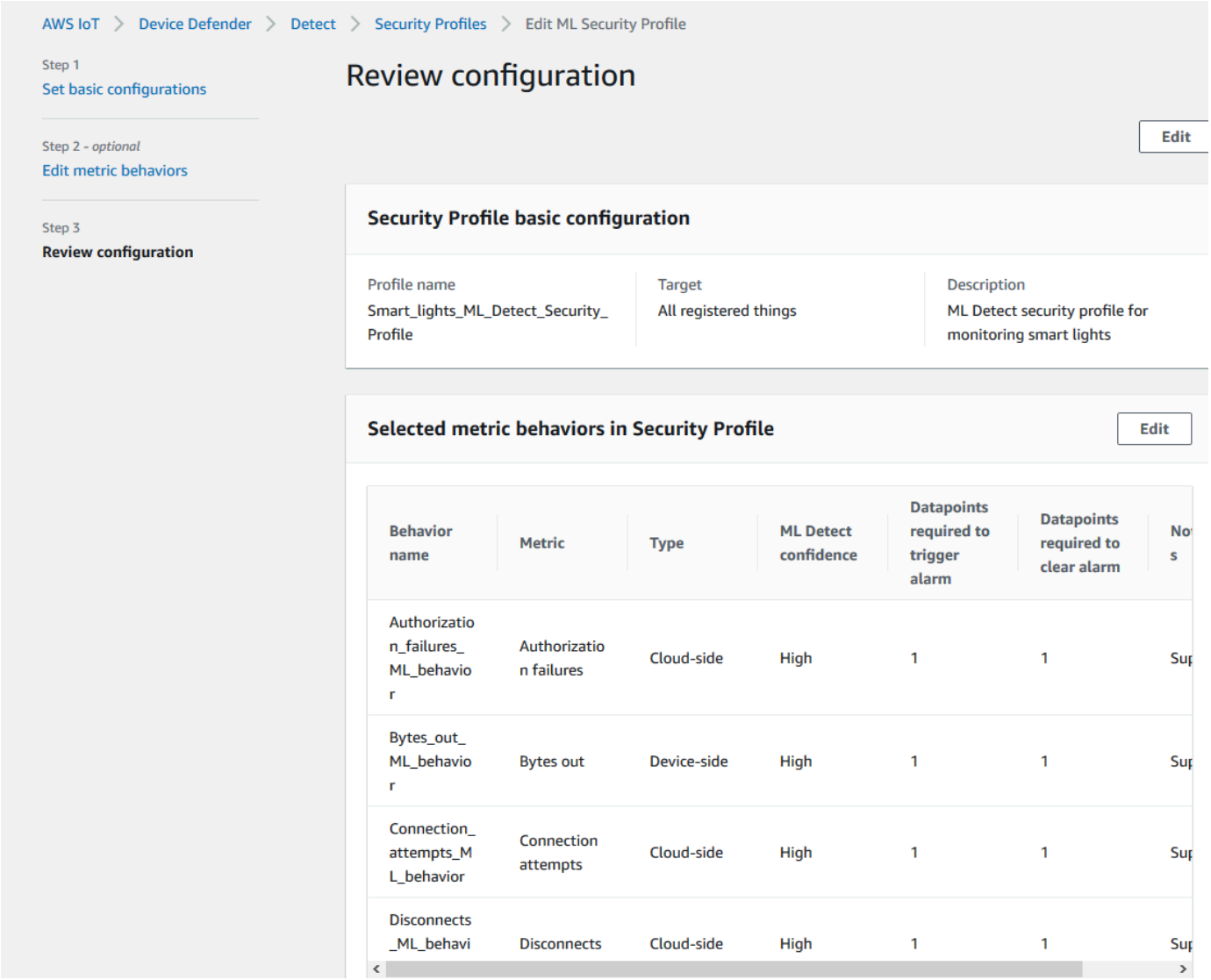
- Unter Stichwörter hinzufügen (optional) können Sie beliebige Tags hinzufügen, die Sie Ihrer Rolle zuordnen möchten. Klicken Sie abschließend auf Weiter: Überprüfen.
- Geben Sie Ihrer Rolle unter Überprüfen einen Namen und stellen Sie sicher, dass AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction unter Berechtigungen und AWS-Service: iot.amazonaws.com unter Vertrauensbeziehungen aufgeführt ist. Wählen Sie anschließend Rolle erstellen.



5. Auf der Seite Metrikverhalten bearbeiten können Sie Ihre ML-Verhaltenseinstellungen anpassen.



6. Wählen Sie abschließend Weiter.
7. Überprüfen Sie auf der Seite Konfiguration überprüfen die Verhaltensweisen, die Machine Learning überwachen soll, und wählen Sie dann Weiter.



8. Nachdem Sie Ihr Sicherheitsprofil erstellt haben, werden Sie zur Seite Sicherheitsprofile weitergeleitet, auf der das neu erstellte Sicherheitsprofil angezeigt wird.

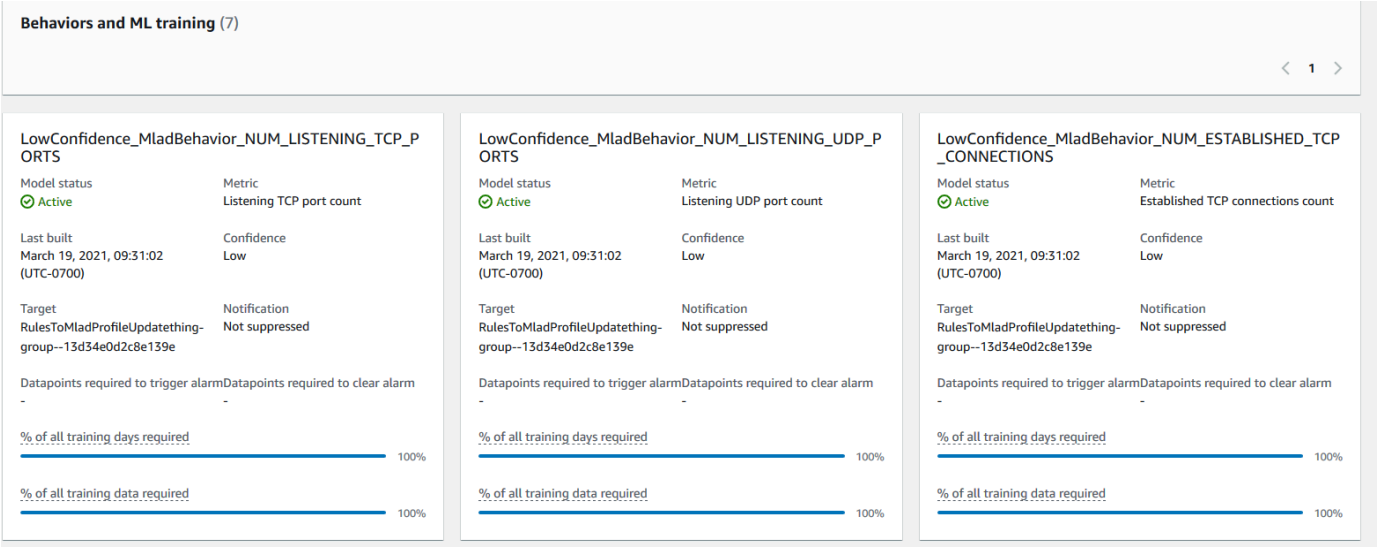
> **Note**
>
> Das erste Training und die Erstellung des ML-Modells dauern 14 Tage. Sie können damit rechnen, dass nach Abschluss des Vorgangs Alarme angezeigt werden, falls auf Ihren Geräten ungewöhnliche Aktivitäten auftreten.

Überwachen des Status Ihres ML-Modells

Während sich Ihre ML-Modelle in der ersten Trainingsphase befinden, können Sie ihren Fortschritt jederzeit überwachen, indem Sie die folgenden Schritte ausführen.

1. Erweitern Sie auf der AWS IoT-Konsole im Navigationsbereich die Option Verteidigen, und wählen Sie dann Erkennen, Sicherheitsprofile.
2. Wählen Sie auf der Seite Sicherheitsprofile das Sicherheitsprofil, das Sie überprüfen möchten. Wählen Sie dann Verhalten und ML-Training.
3. Überprüfen Sie auf der Seite Verhalten und ML-Training den Trainingsfortschritt Ihrer ML-Modelle.

   Sobald Ihr Modell den Status Aktiv hat, werden anhand Ihrer Nutzung Erkennungsentscheidungen getroffen und das Profil wird täglich aktualisiert.

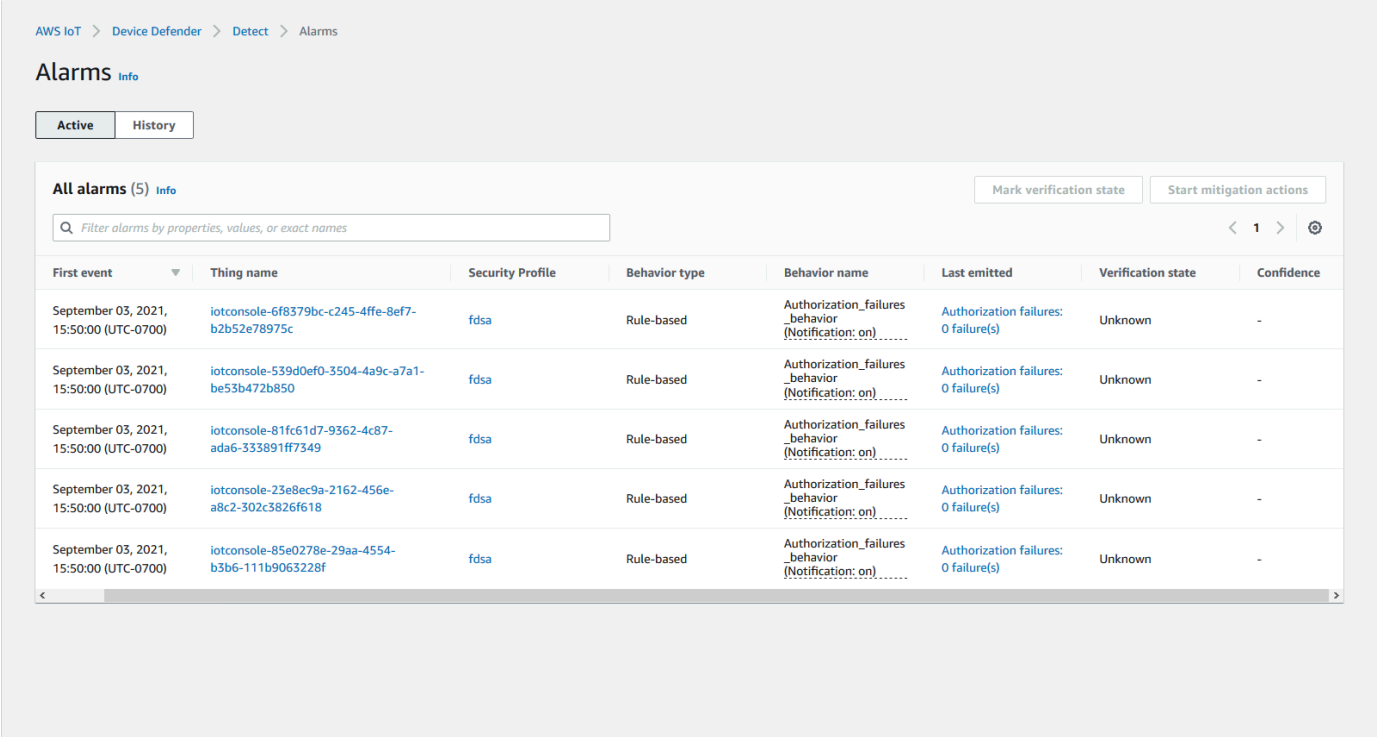


> **Note**
>
> Wenn sich Ihr Modell nicht wie erwartet entwickelt, stellen Sie sicher, dass Ihre Geräte den Mindestanforderungen entsprechen.

Überprüfen Ihrer ML Detect-Alarme

Nachdem Ihre ML-Modelle erstellt und für die Dateninferenz bereit sind, können Sie die anhand der Modelle identifizierten Alarme regelmäßig einsehen und untersuchen.

1. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoT-Konsole, und wählen Sie dann Erkennen, Alarme.

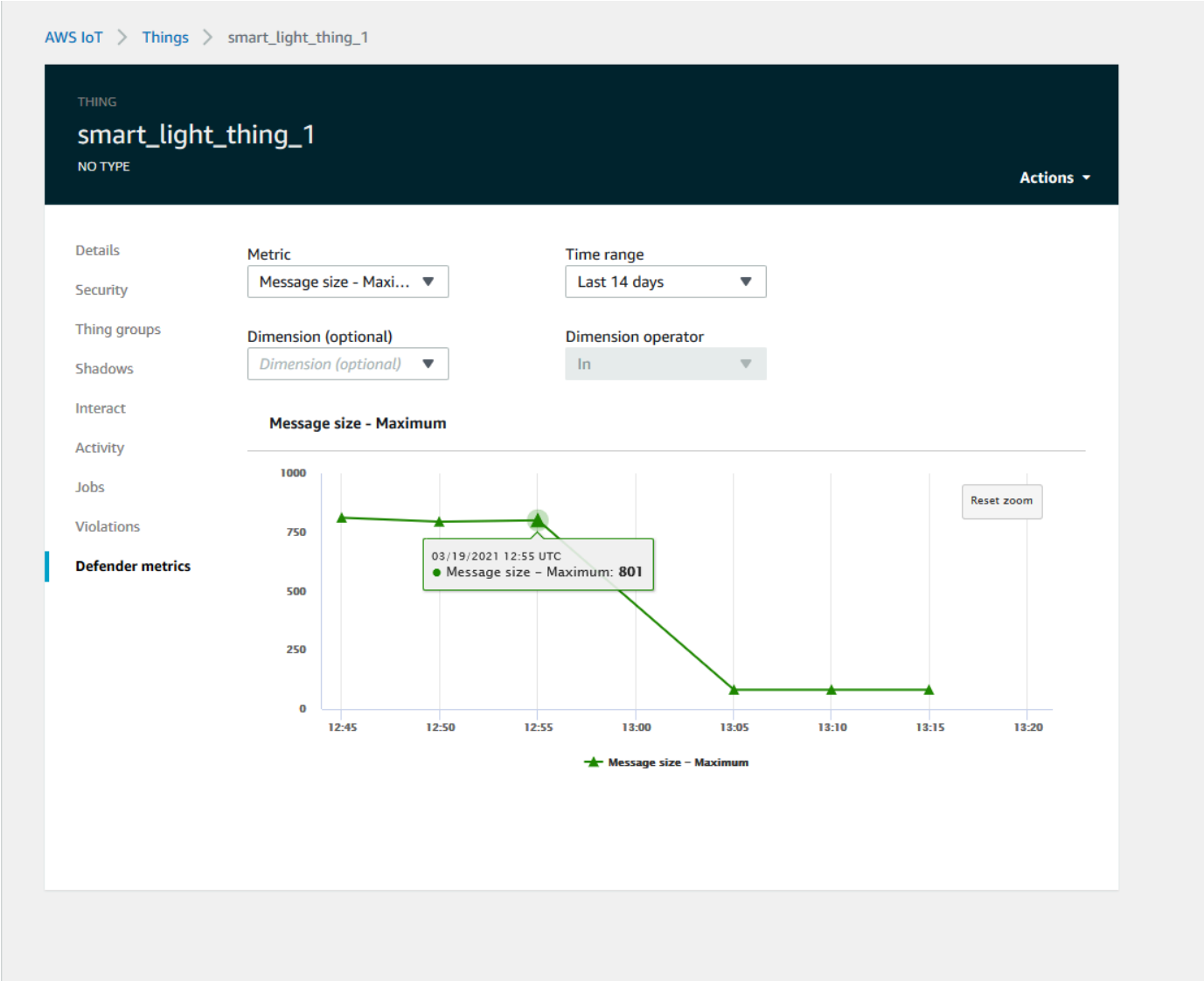


2. Wenn Sie zur Registerkarte Verlauf wechseln, können Sie sich ferner Details zu Ihren Geräten ansehen, für die keine Alarme mehr aktiviert wurden.



Um weitere Informationen zu erhalten, wählen Sie unter Verwalten die Option Objekte. Wählen Sie dann das Objekt aus, für das Sie weitere Details sehen möchten, und navigieren Sie

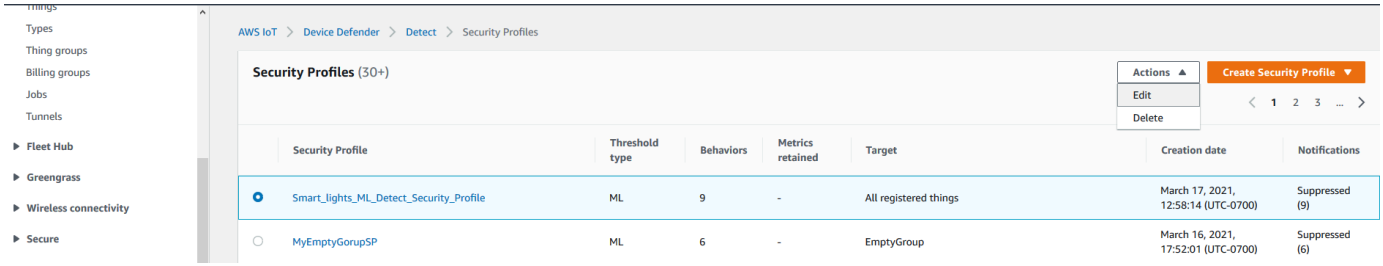
anschließend zu Defender-Metriken. Auf der Registerkarte Aktiv können Sie auf das Defender-Metrikendiagramm zugreifen und Ihre Untersuchung aller Alarmmeldungen durchführen. In diesem Fall zeigt das Diagramm einen Anstieg der Nachrichtengröße, der den Alarm ausgelöst hat. Sie können sehen, dass der Alarm anschließend gelöscht wurde.



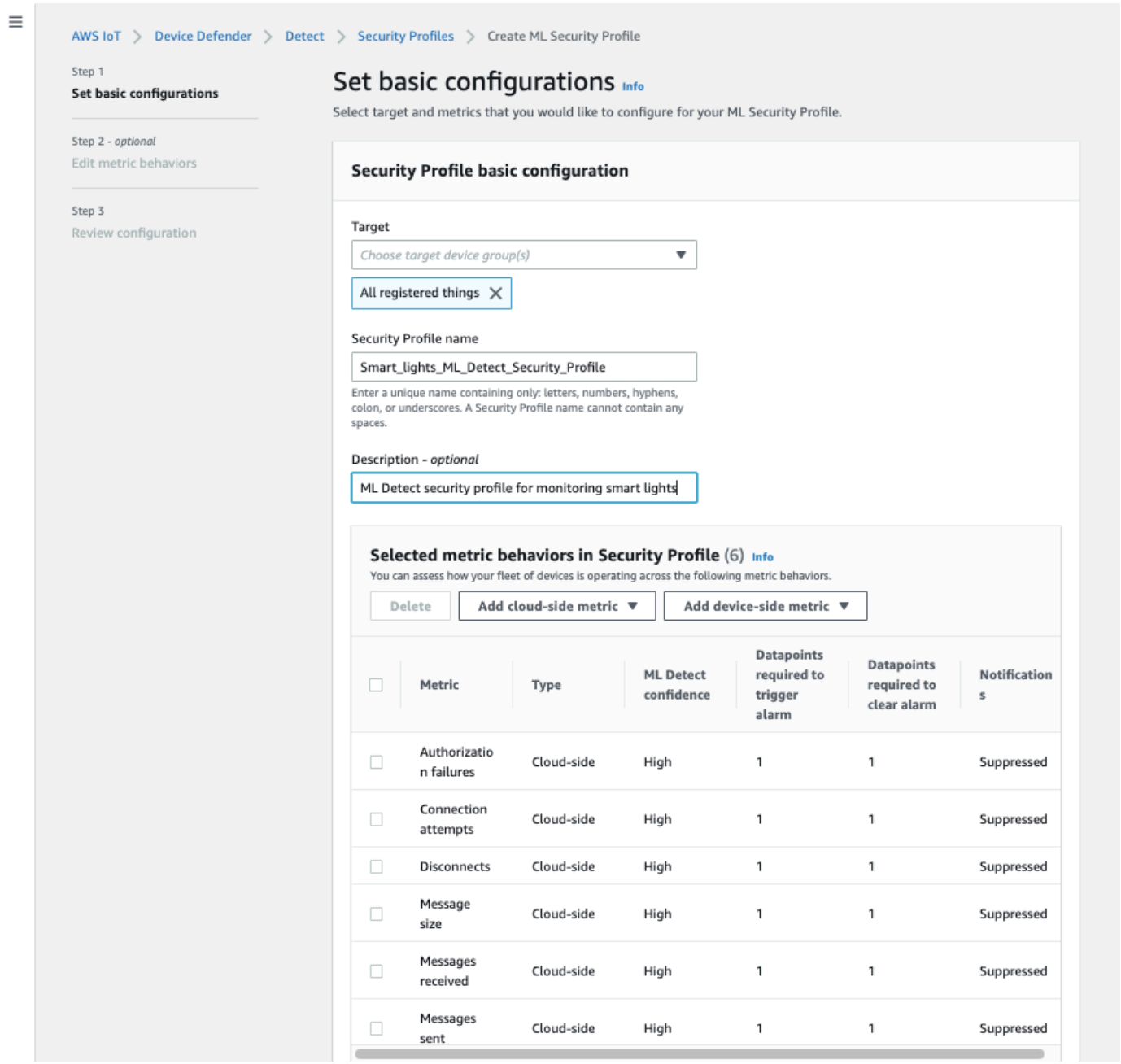
Optimieren Ihrer ML-Alarme

Nachdem Ihre ML-Modelle erstellt und für Datenauswertungen bereit sind, können Sie die ML-Verhaltenseinstellungen Ihres Sicherheitsprofils aktualisieren, um die Konfiguration zu ändern. Das folgende Verfahren zeigt Ihnen, wie Sie die ML-Verhaltenseinstellungen Ihres Sicherheitsprofils in der aktualisieren AWS CLI.

1. Erweitern Sie auf der AWS IoT-Konsole im Navigationsbereich die Option Verteidigen, und wählen Sie dann Erkennen, Sicherheitsprofile.
2. Aktivieren Sie auf der Seite Sicherheitsprofile das Kontrollkästchen neben dem Sicherheitsprofil, das Sie überprüfen möchten. Wählen Sie Aktionen und dann Bearbeiten.

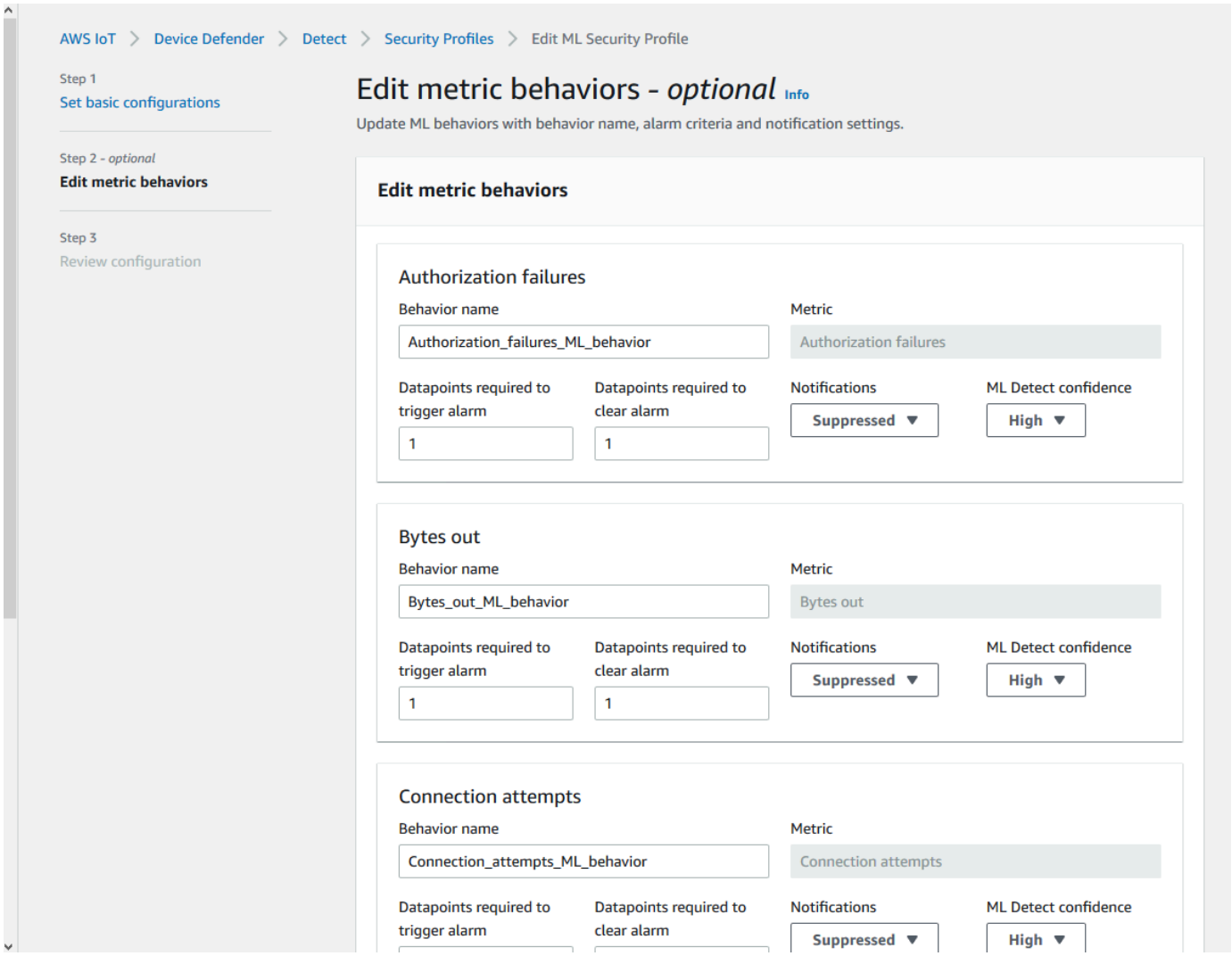


3. Unter Grundlegende Konfigurationen festlegen können Sie die Zielgruppen des Sicherheitsprofils anpassen oder ändern, welche Metriken Sie überwachen möchten.



4. Sie können jede der folgenden Optionen aktualisieren, indem Sie zu Verhalten von Metriken bearbeiten navigieren.

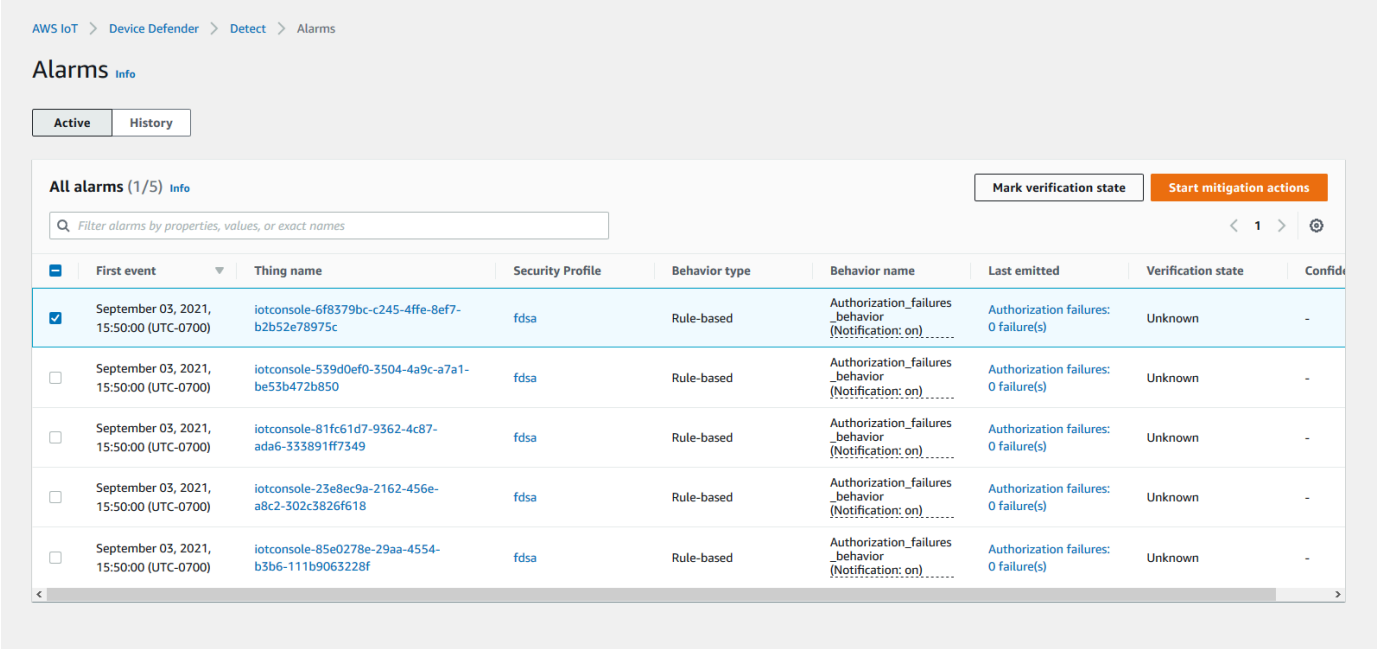
   - Ihre erforderlichen ML-Modell-Datenpunkte, um einen Alarm auszulösen
   - Ihre erforderlichen ML-Modell-Datenpunkte, um einen Alarm zu quittieren
   - Ihr ML Detect-Konfidenzniveau
   - Ihre ML Detect-Benachrichtigungen (z. B. Nicht unterdrückt, Unterdrückt)



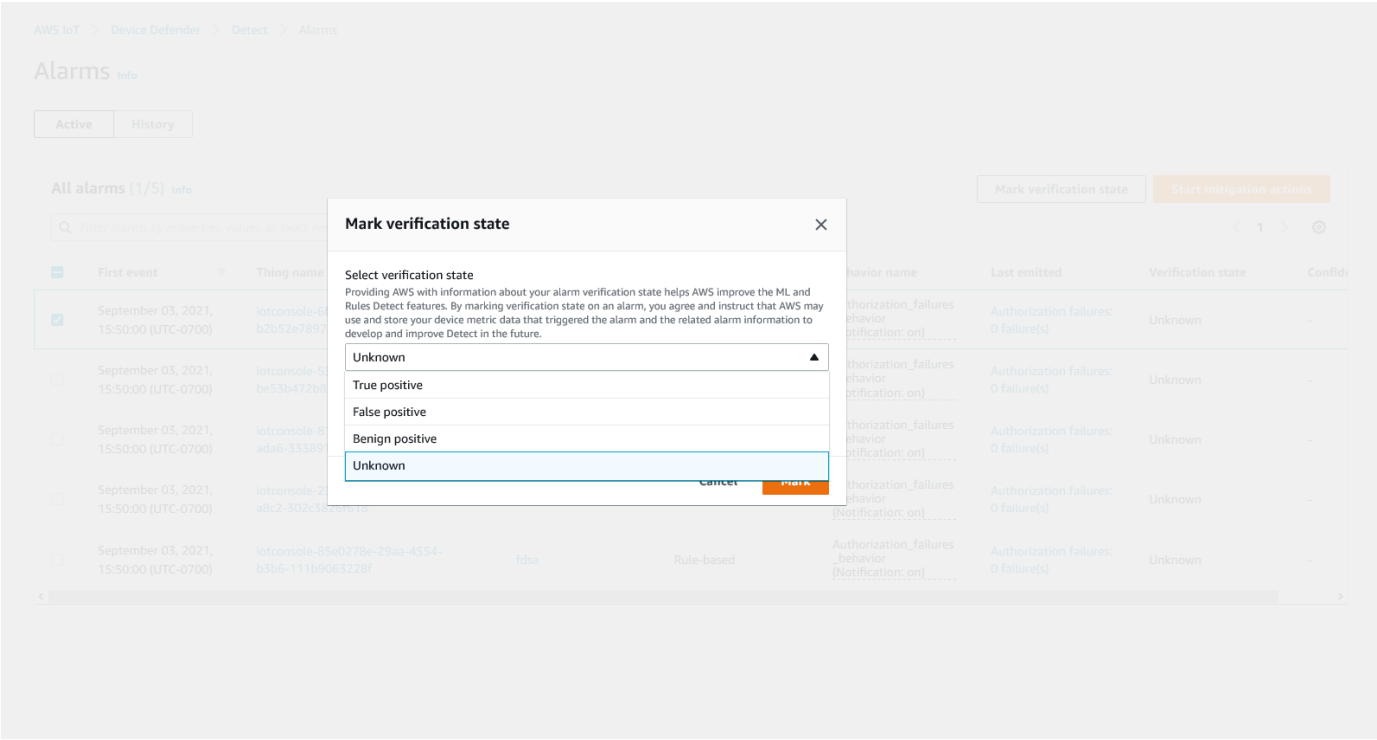
Markieren des Bestätigungsstatus Ihres Alarms

Markieren Sie Ihre Alarme, indem Sie den Bestätigungsstatus festlegen und eine Beschreibung dieses Bestätigungsstatus angeben. Dies hilft Ihnen und Ihrem Team, Alarme zu identifizieren, auf die Sie nicht reagieren müssen.

1. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoT-Konsole, und wählen Sie dann Erkennen, Alarme. Wählen Sie einen Alarm aus, um seinen Bestätigungsstatus zu kennzeichnen.



2. Wählen Sie Bestätigungsstatus markieren. Das Modal mit dem Bestätigungsstatus wird geöffnet.
3. Wählen Sie den entsprechenden Bestätigungsstatus, geben Sie eine Beschreibung der Überprüfung ein (optional), und wählen Sie dann Markieren. Diese Aktion weist dem ausgewählten Alarm einen Bestätigungsstatus und eine Beschreibung zu.

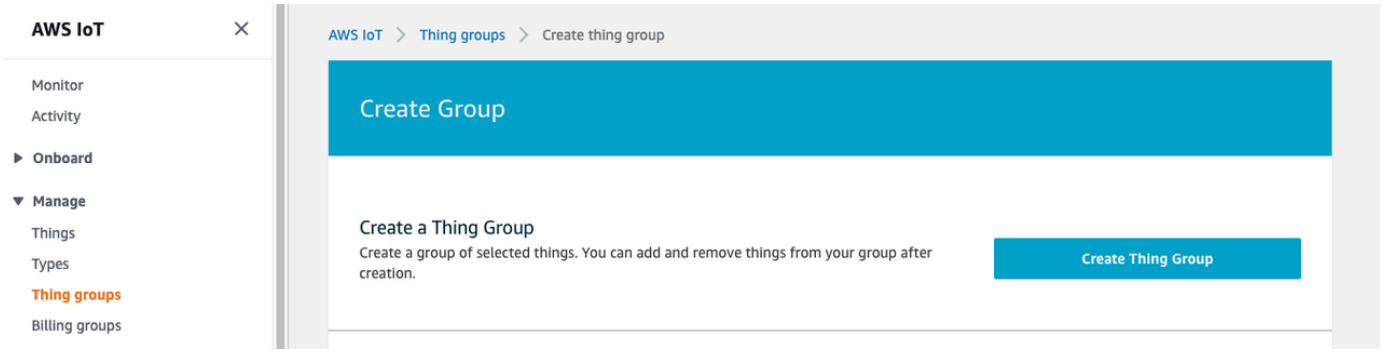


Beseitigen von identifizierten Geräteproblemen

1. (Optional) Bevor wir Maßnahmen zur Eindämmung der Quarantäne einrichten, richten wir zunächst eine Quarantänegruppe ein, in die wir das Gerät verschieben, das gegen die Quarantäne verstößt. Sie können auch eine vorhandene Gruppe verwenden.
2. Navigieren Sie zu Verwalten, Objektgruppen und dann zu Objektgruppe erstellen. Benennen Sie Ihre Objektgruppe. In diesem Tutorial geben wir unserer Objektgruppe den Namen `Quarantine_group`. Wenden Sie unter Objektgruppe, Sicherheit die folgende Richtlinie auf die Objektgruppe an.

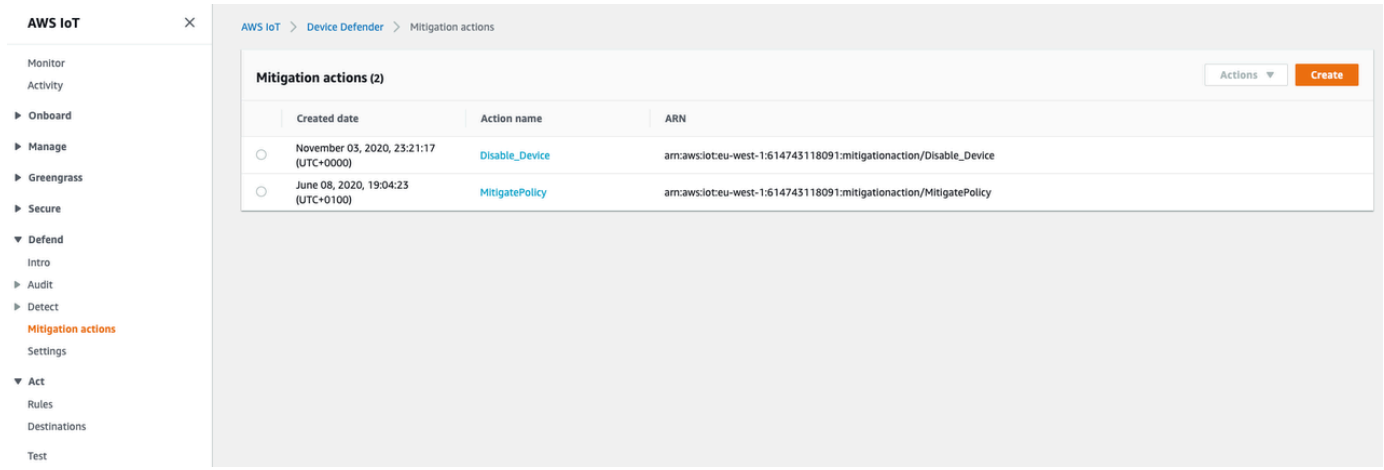
   JSON

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

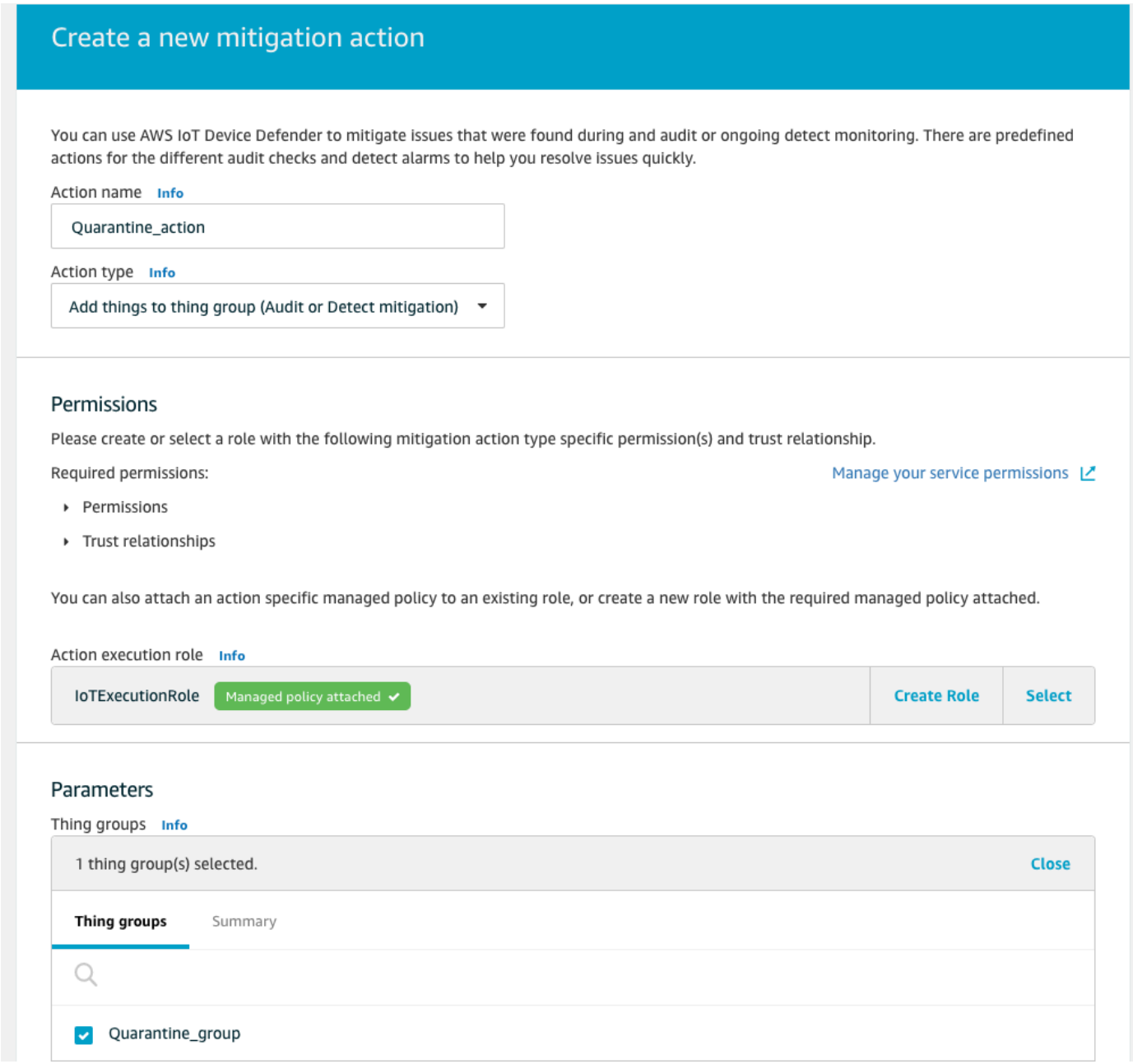
   Wählen Sie anschließend Erstellen.
3. Nachdem wir nun eine Objektgruppe erstellt haben, erstellen wir eine Abhilfemaßnahme, mit der Geräte, bei denen ein Alarm ausgelöst wird, in die `Quarantine_group` verschoben werden.

   Wählen Sie unter Verteidigen, Abhilfemaßnahmen die Option Erstellen.



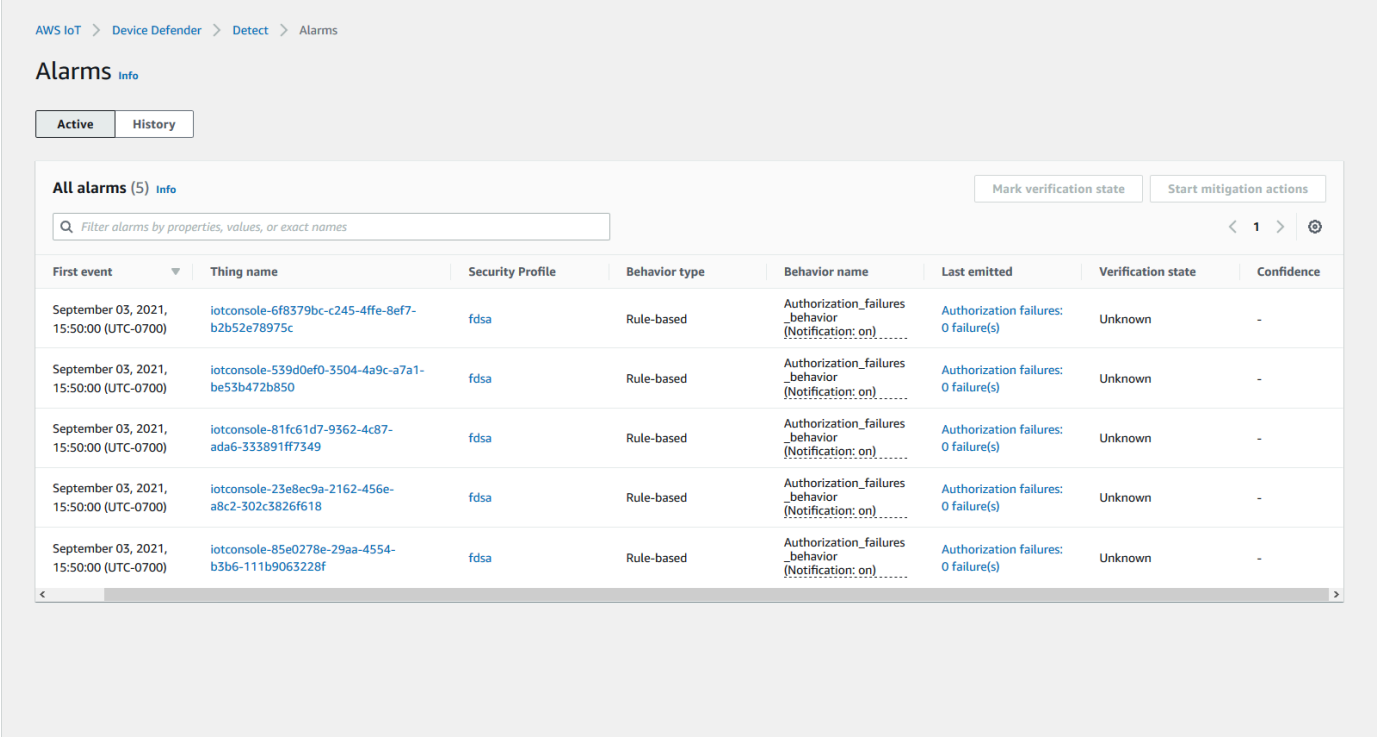
4. Geben Sie auf der Seite Neue Abhilfemaßnahme erstellen die folgenden Informationen ein.

   - Aktionsname: Geben Sie Ihrer Abhilfemaßnahme einen Namen, z. B. **`Quarantine_action`**.
   - Aktionstyp: Wählen Sie die Art der Aktion. Wir wählen Objekte zur Objektgruppe hinzufügen (Audit oder Detect-Abhilfemaßnahme).
   - Rolle zur Aktionsausführung: Erstellen Sie eine Rolle, oder wählen Sie eine vorhandene Rolle, falls Sie zuvor eine erstellt haben.
   - Parameter: Wählen Sie eine Objektgruppe. Wir können `Quarantine_group` verwenden, die wir zuvor erstellt haben.



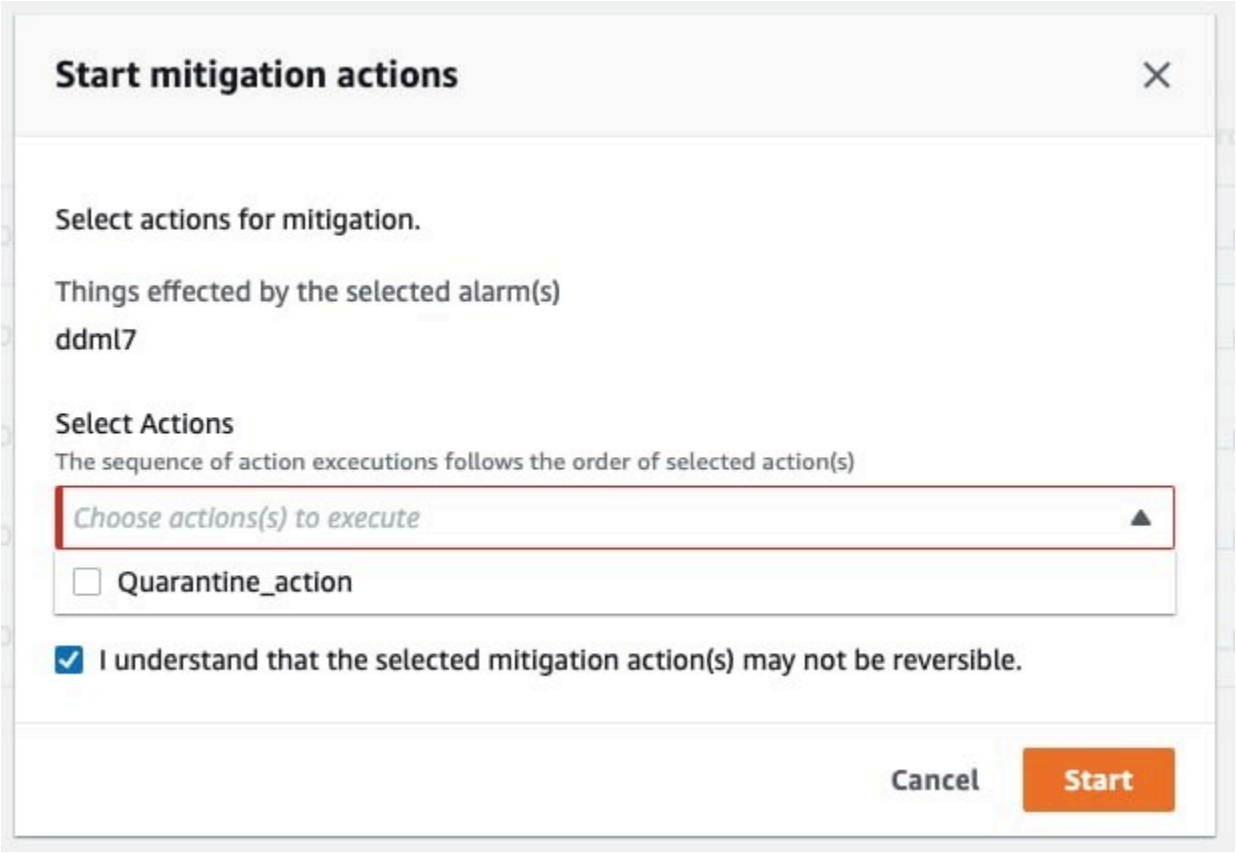
Klicken Sie abschließend auf Speichern. Sie verfügen jetzt über eine Abhilfemaßnahme, die Geräte, die sich im Alarmzustand befinden, in eine Quarantäne-Objektgruppe verschiebt, und über eine Abhilfemaßnahme, um das Gerät zu isolieren, während Sie die Untersuchung durchführen.

5. Navigieren Sie zu Defender, Erkennen, Alarme. Unter Aktiv können Sie sehen, welche Geräte sich im Alarmzustand befinden.

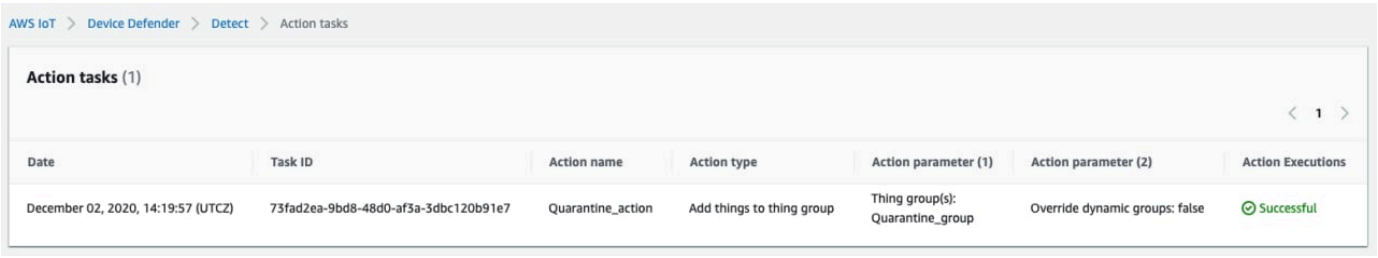


Wählen Sie das Gerät aus, das Sie in die Quarantänegruppe verschieben möchten, und wählen Sie Abhilfemaßnahmen starten.

6. Wählen Sie unter Abhilfemaßnahmen starten, Aktionen starten die Schutzmaßnahme aus, die Sie zuvor erstellt haben. Wir wählen zum Beispiel **`Quarantine_action`** und dann Start. Die Seite Aktionsaufgaben wird geöffnet.



7. Das Gerät ist jetzt in der **`Quarantine_group`** isoliert und Sie können die Ursache des Problems untersuchen, das den Alarm ausgelöst hat. Nachdem Sie die Untersuchung abgeschlossen haben, können Sie das Gerät aus der Objektgruppe entfernen oder weitere Maßnahmen ergreifen.



So verwenden Sie ML Detect mit der CLI

Im Folgenden erfahren Sie, wie Sie ML Detect mithilfe der CLI einrichten.

Tutorials

- Aktivieren von ML Detect

- Überwachen des Status Ihres ML-Modells
- Überprüfen Ihrer ML Detect-Alarme
- Optimieren Ihrer ML-Alarme
- Markieren des Bestätigungsstatus Ihres Alarms
- Beseitigen von identifizierten Geräteproblemen

Aktivieren von ML Detect

Das folgende Verfahren zeigt Ihnen, wie Sie ML Detect in der aktivieren AWS CLI.

1. Stellen Sie sicher, dass Ihre Geräte die Mindestdatenpunkte erzeugen, die gemäß den Mindestanforderungen von ML Detect für das kontinuierliche Training und die kontinuierliche Aktualisierung des Modells erforderlich sind. Damit die Datenerfassung fortgesetzt werden kann, stellen Sie sicher, dass sich Ihre Objekte in einer Objektgruppe befinden, die an ein Sicherheitsprofil angehängt ist.
2. Erstellen Sie ein ML Detect-Sicherheitsprofil mit dem Befehl `create-security-profile`. Im folgenden Beispiel wird ein Sicherheitsprofil mit dem Namen *`security-profile-for-smart-lights`* erstellt, das die Anzahl der gesendeten Nachrichten, die Anzahl der Autorisierungsfehler, die Anzahl der Verbindungsversuche und die Anzahl der Verbindungsabbrüche überprüft. Im Beispiel wird `mlDetectionConfig` verwendet, um festzulegen, dass die Metrik das ML Detect-Modell verwendet.

```
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
  }]'
```

Ausgabe:

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

3. Ordnen Sie als Nächstes Ihr Sicherheitsprofil einer oder mehreren Objektgruppen zu. Verwenden Sie den Befehl attach-security-profile, um Ihrem Sicherheitsprofil eine Objektgruppe anzuhängen. Im folgenden Beispiel wird eine Objektgruppe namens *ML_Detect_Beta_Static_Group* dem Sicherheitsprofil *security-profile-for-smart-lights* zugewiesen.

```
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
```

   Ausgabe:

   Keine.

4. Nachdem Sie Ihr vollständiges Sicherheitsprofil erstellt haben, beginnt das ML-Modell mit dem Training. Das erste Training und die Erstellung des ML-Modells dauern 14 Tage. Wenn nach 14 Tagen ungewöhnliche Aktivitäten auf Ihrem Gerät auftreten, können Sie damit rechnen, dass Alarme angezeigt werden.

Überwachen des Status Ihres ML-Modells

Das folgende Verfahren zeigt Ihnen, wie Sie das laufende Training Ihrer ML-Modelle überwachen können.

- Verwenden Sie den Befehl get-behavior-model-training-summaries, um den Fortschritt Ihres ML-Modells zu überprüfen. Im folgenden Beispiel wird die Zusammenfassung des Trainingsfortschrittes des ML-Modells für das Sicherheitsprofil *security-profile-for-smart-lights* abgerufen. `modelStatus` zeigt Ihnen, ob ein Modell das Training abgeschlossen hat oder ob die Erstellung eines Modells für ein bestimmtes Verhalten noch aussteht.

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

   Ausgabe:

```
{
    "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**
>
> Wenn sich Ihr Modell nicht wie erwartet entwickelt, stellen Sie sicher, dass Ihre Geräte den Mindestanforderungen entsprechen.

Überprüfen Ihrer ML Detect-Alarme

Nachdem Ihre ML-Modelle erstellt wurden und für die Datenauswertung bereit sind, können Sie regelmäßig alle Alarme anzeigen, die von den Modellen abgeleitet werden. Nachstehend wird veranschaulicht, wie Sie Ihre Alarme in der einsehen können AWS CLI.

- Verwenden Sie den Befehl `list-active-violations`, um alle aktiven Alarme anzuzeigen.

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

  Ausgabe:

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

  Alternativ können Sie mit dem Befehl `list-violation-events` alle Verstöße anzeigen, die in einem bestimmten Zeitraum entdeckt wurden. Im folgenden Beispiel werden Verstöße vom 22. September 2020, 5:42:13 Uhr GMT bis 26. Oktober 2020, 5:42:13 Uhr (GMT) aufgeführt.

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

  Ausgabe:

```
{
    "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"
}
```

Optimieren Ihrer ML-Alarme

Sobald Ihre ML-Modelle erstellt und für Datenauswertungen bereit sind, können Sie die ML-Verhaltenseinstellungen Ihres Sicherheitsprofils aktualisieren, um die Konfiguration zu ändern. Das folgende Verfahren zeigt Ihnen, wie Sie die ML-Verhaltenseinstellungen Ihres Sicherheitsprofils in der aktualisieren AWS CLI.

- Verwenden Sie den Befehl `update-security-profile`, um die ML-Verhaltenseinstellungen Ihres Sicherheitsprofils zu ändern. Im folgenden Beispiel wird das Verhalten des Sicherheitsprofils *`security-profile-for-smart-lights`* aktualisiert, indem das `confidenceLevel` einiger Verhaltensweisen geändert und die Unterdrückung von Benachrichtigungen für alle Verhaltensweisen aufgehoben wird.

```
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

    }]'
```

Ausgabe:

```
 {
    "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
}
```

Markieren des Bestätigungsstatus Ihres Alarms

Sie können Ihre Alarme mit Bestätigungsstatus kennzeichnen, um Alarme besser klassifizieren und Anomalien untersuchen zu können.

- Kennzeichnen Sie Ihre Alarme mit einem Bestätigungsstatus und einer Beschreibung dieses Status. Um beispielsweise den Bestätigungsstatus eines Alarms auf Falsch positiv zu setzen, verwenden Sie den folgenden Befehl:

```
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
```

Ausgabe:

Keine.

Beseitigen von identifizierten Geräteproblemen

1. Verwenden Sie den Befehl create-thing-group, um eine Objektgruppe für die Abhilfemaßnahme zu erstellen. Im folgenden Beispiel erstellen wir eine Objektgruppe namens ThingGroupForDetectMitigationAction.

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

Ausgabe:

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

2. Verwenden Sie dann den Befehl create-mitigation-action zum Erstellen Ihrer Abhilfemaßnahme. Im folgenden Beispiel erstellen wir eine Abhilfemaßnahme namens detect_mitigation_action mit dem ARN der IAM-Rolle, die zur Anwendung der Abhilfemaßnahme verwendet wird. Darüber hinaus definieren wir den Aktionstyp und die Parameter für diese Aktion. In diesem Fall verschiebt unsere Schadensbegrenzung Objekte in unsere zuvor erstellte Objektgruppe namens ThingGroupForDetectMitigationAction.

```
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
     }
 }'
```

Ausgabe:

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

3. Verwenden Sie den Befehl [start-detect-mitigation-actions-task](), um Ihre Abhilfemaßnahmen-Aufgabe zu starten. `task-id`, `target` und `actions` sind erforderliche Parameter.

```
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
```

Ausgabe:

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

4. (Optional) Verwenden Sie den Befehl [list-detect-mitigation-actions-executions](), um die in einer Aufgabe enthaltenen Abhilfemaßnahme-Ausführungen anzuzeigen.

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

Ausgabe:

```
{
    "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. (Optional) Verwenden Sie den Befehl describe-detect-mitigation-actions-task, um Informationen zu einer Abhilfemaßnahme abzurufen.

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

Ausgabe:

```
{
    "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. (Optional) Verwenden Sie den Befehl [list-detect-mitigation-actions-tasks](), um eine Liste Ihrer Abhilfemaßnahmen-Aufgaben abzurufen.

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

Ausgabe:

```
{
    "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. (Optional) Verwenden Sie den Befehl [cancel-detect-mitigation-actions-task](), um eine Abhilfemaßnahmen-Aufgabe abzubrechen.

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

Ausgabe:

Keine.

Anpassen, wann und wie Sie die AWS IoT Device Defender-Prüfungsergebnisse anzeigen

AWS IoT Device Defender Audit bietet regelmäßige Sicherheitsprüfungen, um sicherzustellen, dass AWS IoT-Geräte und -Ressourcen den bewährten Methoden entsprechen. Bei jeder Prüfung werden die Prüfungsergebnisse als konform oder nonkonform eingestuft, wobei bei Compliance-Abweichungen Warnsymbole in der Konsole angezeigt werden. Um den Geräuschen, die durch die Wiederholung bekannter Probleme entstehen, entgegenzuwirken, können Sie mit dem Feature zur Unterdrückung von Prüfungsergebnissen diese Benachrichtigungen zu Compliance-Abweichungen vorübergehend stummschalten.

Sie können ausgewählte Audit-Prüfungen für eine bestimmte Ressource oder ein bestimmtes Konto für einen bestimmten Zeitraum unterdrücken. Ein Prüfungsergebnis, das unterdrückt wurde, wird unabhängig von den Kategorien „konform“ und „nonkonform“ als unterdrückte Erkenntnis eingestuft. Diese neue Kategorie löst im Gegensatz zu einem nonkonformen Ergebnis keinen Alarm aus. Auf diese Weise können Sie Störungen durch Benachrichtigungen zu Compliance-Abweichungen während bekannter Wartungsperioden oder bis zum geplanten Abschluss eines Updates reduzieren.

Erste Schritte

In den folgenden Abschnitten wird detailliert beschrieben, wie Sie mit Unterdrückungen von Prüfungsergebnissen eine `Device certificate expiring`-Prüfung in der Konsole und in der CLI unterdrücken können. Wenn Sie eines der Beispiele nachvollziehen möchten, müssen Sie zunächst zwei ablaufende Zertifikate erstellen, die Device Defender erkennen kann.

Erstellen Sie die Zertifikate wie folgt:

- Erstellen und Registrieren eines CA-Zertifikats im AWS IoT Core-Entwicklerhandbuch
- Erstellen eines Clientzertifikats mit Ihrem CA-Zertifikat. Legen Sie in Schritt 3 Ihren `days`-Parameter auf **1** fest.

Wenn Sie die CLI zum Erstellen Ihrer Zertifikate verwenden, geben Sie den folgenden Befehl ein.

```
openssl x509 -req \
    -in device_cert_csr_filename \
    -CA root_ca_pem_filename \
    -CAkey root_ca_key_filename \
```

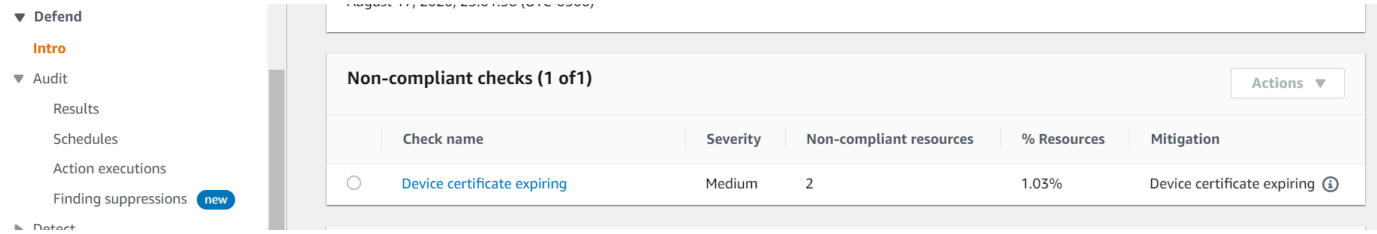
```
    -CAcreateserial \
    -out device_cert_pem_filename \
    -days 1 -sha256
```

Anpassen Ihrer Prüfungsergebnisse in der Konsole

In der folgenden exemplarischen Vorgehensweise wird ein Konto mit zwei abgelaufenen Gerätezertifikaten verwendet, die eine Prüfung auf Compliance-Abweichungen auslösen. In diesem Szenario möchten wir die Warnung deaktivieren, da unsere Entwickler ein neues Feature testen, mit dem das Problem behoben werden kann. Wir erstellen für jedes Zertifikat eine Unterdrückung von Prüfungsergebnissen, um zu verhindern, dass das Prüfergebnis in der nächsten Woche Compliance-Abweichungen anzeigt.

1. Zunächst führen wir eine On-Demand-Prüfung durch, um nachzuweisen, dass die Prüfung auf abgelaufene Gerätezertifikate nonkonform ist.

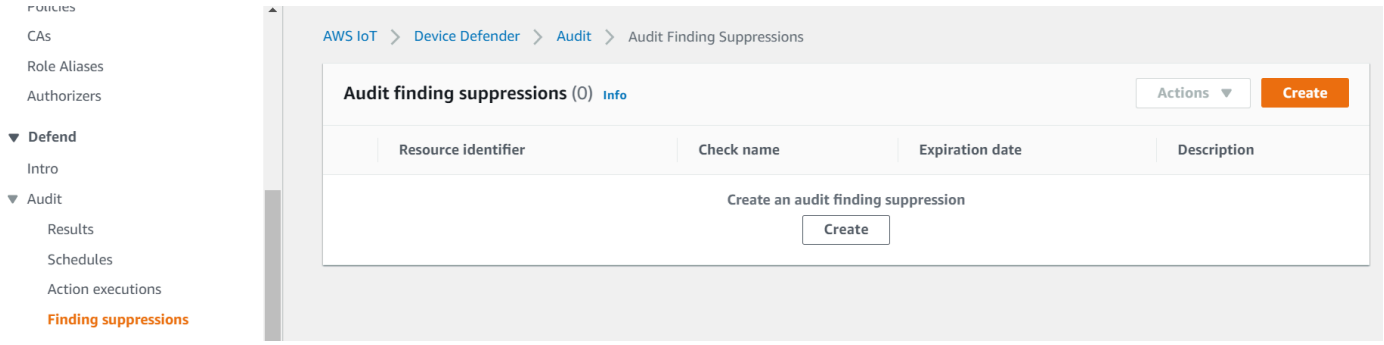
   Wählen Sie in der AWS IoT-Konsole in der linken Seitenleiste Defend, dann Audit und anschließend Ergebnisse aus. Klicken Sie auf der Seite Prüfungsergebnisse auf Erstellen. Das Fenster Neue Prüfung erstellen wird geöffnet. Wählen Sie Erstellen aus.



   Aus den Ergebnissen der On-Demand-Prüfung geht hervor, dass „Gerätezertifikat läuft ab“ für zwei Ressourcen nonkonform ist.

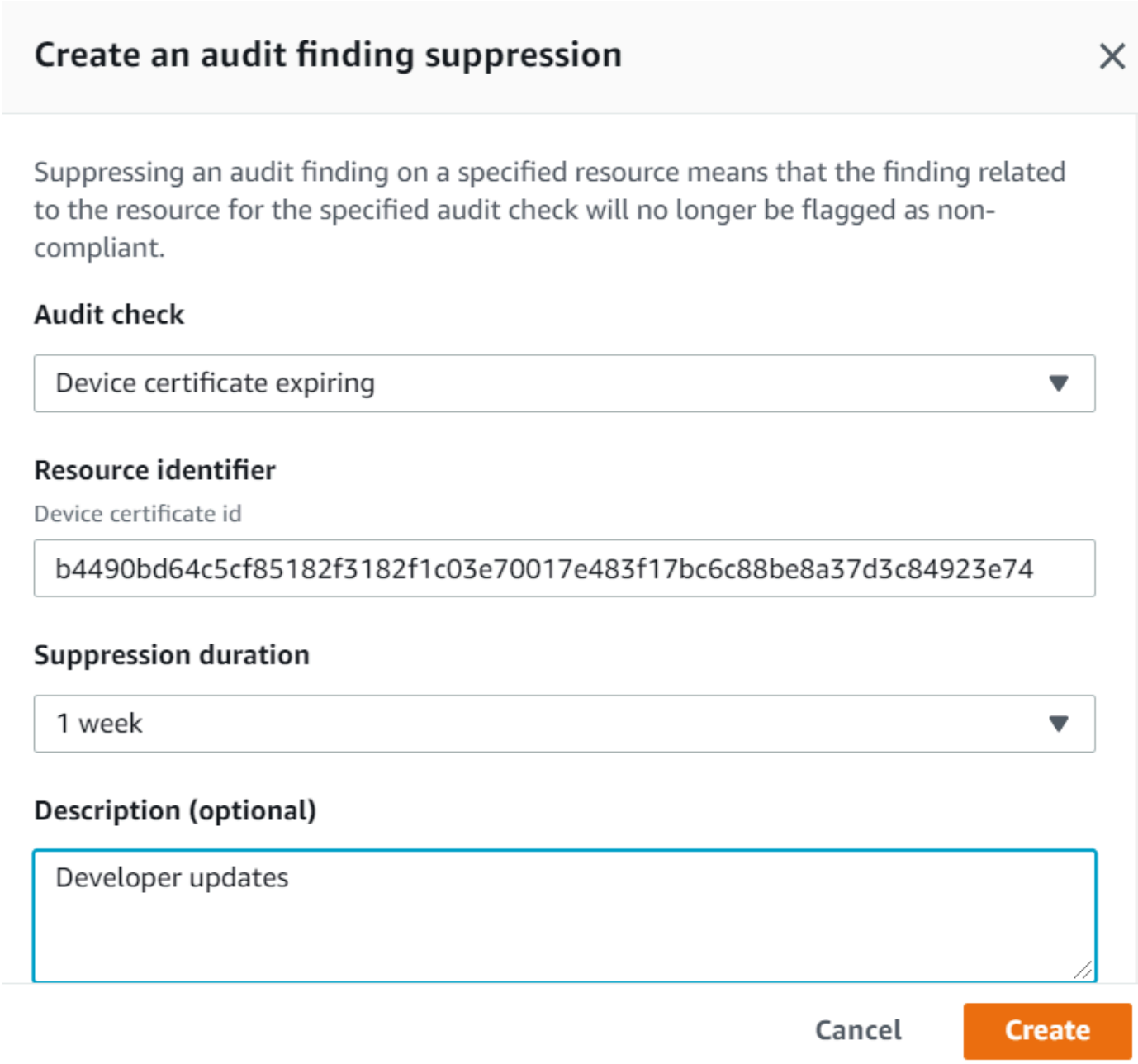
2. Jetzt möchten wir die Warnung „Gerätezertifikat läuft ab“ zur Prüfung auf Compliance-Abweichungen deaktivieren, da unsere Entwickler neue Features testen, mit denen diese Warnung behoben werden kann.

   Wählen Sie in der linken Seitenleiste unter Defend die Option Audit und anschließend Unterdrückungen des Prüfungsergebnisses aus. Wählen Sie auf der Seite Unterdrückungen des Prüfungsergebnisses die Option Erstellen aus.



3. Im Fenster Unterdrückung des Prüfungsergebnisses erstellen muss Folgendes ausgefüllt werden.

   - Audit-Prüfung: Wir wählen `Device certificate expiring` aus, denn dies ist die Prüfung, die wir unterdrücken möchten.
   - Ressourcen-ID: Wir geben die Gerätezertifikat-ID eines der Zertifikate ein, für die wir die Prüfungsergebnisse unterdrücken möchten.
   - Dauer der Unterdrückung: Wir wählen `1 week` aus, weil wir die Prüfung `Device certificate expiring` für diese Dauer unterdrücken möchten.
   - Beschreibung (optional): Wir fügen einen Hinweis hinzu, der beschreibt, warum wir dieses Prüfungsergebnis unterdrücken.

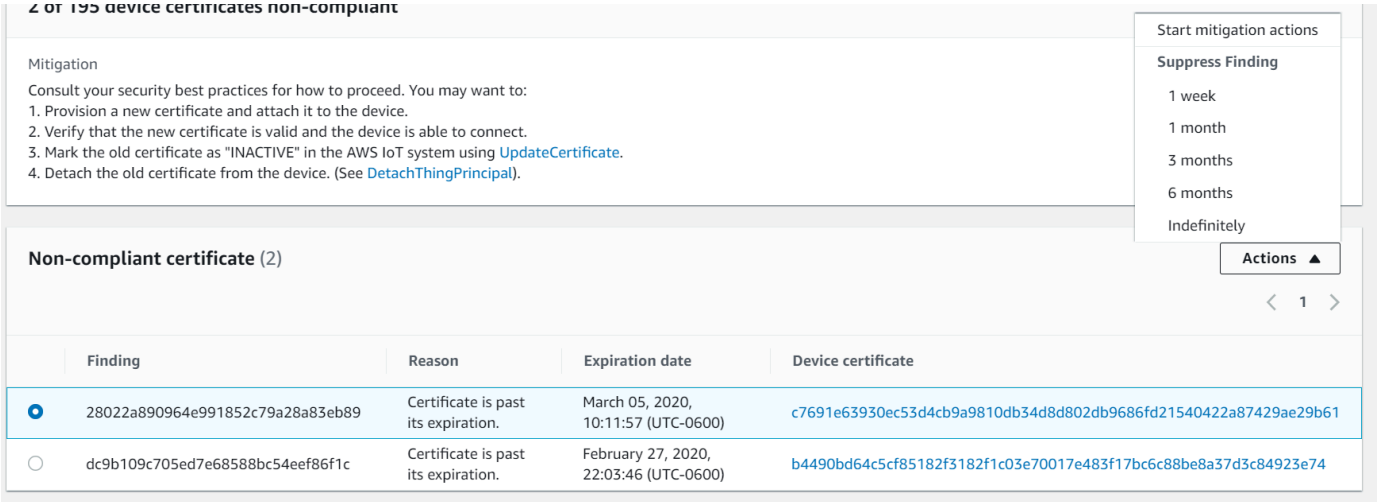


Nachdem wir die Felder ausgefüllt haben, wählen Sie Erstellen aus. Nachdem die Unterdrückung der Prüfungsergebnisse erstellt wurde, wird ein Erfolgsbanner angezeigt.

4. Wir haben ein Prüfergebnis für eines der Zertifikate unterdrückt und müssen nun das Prüfergebnis für das zweite Zertifikat unterdrücken. Wir könnten dieselbe Methode zur Unterdrückung verwenden, die wir in Schritt 3 verwendet haben, wir werden jedoch zu Demonstrationszwecken eine andere Methode nutzen.

Wählen Sie in der linken Seitenleiste unter Defend die Option Audit und anschließend Ergebnisse aus. Wählen Sie auf der Seite Prüfungsergebnisse die Prüfung mit der nonkonformen Ressource aus. Wählen Sie dann unter Prüfungen mit Compliance-Abweichungen die Ressource aus. In diesem Fall wählen wir „Gerätezertifikat läuft ab“ aus.

5. Wählen Sie auf der Seite Gerätezertifikat läuft ab unter Nicht konforme Richtlinie die Optionsschaltfläche neben der Erkenntnis aus, die unterdrückt werden muss. Wählen Sie als Nächstes das Dropdown-Menü Aktionen und dann die Dauer aus, für welche die Erkenntnis unterdrückt werden soll. In unserem Fall wählen wir `1 week` aus, wie wir es auch für das andere Zertifikat getan haben. Wählen Sie im Fenster Unterdrückung bestätigen die Option Unterdrückung aktivieren aus.



Nachdem die Unterdrückung der Prüfungsergebnisse erstellt wurde, wird ein Erfolgsbanner angezeigt. Beide Prüfungsergebnisse sind nun für eine Woche unterdrückt, während unsere Entwickler an einer Lösung zur Behebung der Warnung arbeiten.

Anpassen Ihrer Prüfungsergebnisse in der CLI

In der folgenden exemplarischen Vorgehensweise wird ein Konto mit einem abgelaufenen Gerätezertifikat verwendet, das eine Prüfung auf Compliance-Abweichungen auslöst. In diesem Szenario möchten wir die Warnung deaktivieren, da unsere Entwickler ein neues Feature testen, mit dem das Problem behoben werden kann. Wir erstellen für das Zertifikat eine Unterdrückung von Prüfungsergebnissen, um zu verhindern, dass das Prüfergebnis in der nächsten Woche Compliance-Abweichungen anzeigt.

Wir verwenden die folgenden CLI-Befehle.

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

1. Verwenden Sie den folgenden Befehl, um die Prüfung zu aktivieren.

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

   Ausgabe:

   Keine.

2. Verwenden Sie den folgenden Befehl, um ein On-Demand-Audit auszuführen, das auf die `DEVICE_CERTIFICATE_EXPIRING_CHECK`-Prüfung abzielt.

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

   Ausgabe:

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

3. Verwenden Sie den Befehl describe-account-audit-configuration, um die Audit-Konfiguration zu beschreiben. Wir möchten bestätigen, dass wir die Prüfung für `DEVICE_CERTIFICATE_EXPIRING_CHECK` aktiviert haben.

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

   Ausgabe:

```
{
```

```
    "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` sollte einen Wert von `true` haben.

4. Verwenden Sie den Befehl list-audit-task, um die abgeschlossenen Audit-Aufgaben zu identifizieren.

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

Ausgabe:

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

Die `taskId` der Prüfung, die Sie in Schritt 1 ausgeführt haben, sollte einen `taskStatus` von `COMPLETED` haben.

5. Verwenden Sie den Befehl describe-audit-task, um anhand der `taskId`-Ausgabe aus dem vorherigen Schritt Details zur abgeschlossenen Prüfung abzurufen. Mit diesem Befehl werden Details zu Ihrer Prüfung aufgelistet.

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

Ausgabe:

```
{
    "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. Verwenden Sie den Befehl list-audit-findings, um die nonkonforme Zertifikat-ID zu ermitteln, sodass wir die Prüfungswarnungen für diese Ressource aussetzen können.

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

Ausgabe:

```
{
    "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. Verwenden Sie den Befehl create-audit-suppression, um Benachrichtigungen für die DEVICE_CERTIFICATE_EXPIRING_CHECK-Audit-Prüfung für ein Gerätezertifikat mit der ID *c7691e<shortened>* bis *2020-08-20* zu unterdrücken.

```
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. Verwenden Sie den Befehl list-audit-suppression, um die Einstellung für die Prüfungsunterdrückung zu bestätigen und Einzelheiten zur Unterdrückung abzurufen.

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

Ausgabe:

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

9. Der Befehl update-audit-suppression kann verwendet werden, um die Unterdrückung der Prüfungsergebnisse zu aktualisieren. Im folgenden Beispiel wird das `expiration-date` auf `08/21/20` aktualisiert.

```
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. Der Befehl delete-audit-suppression kann verwendet werden, um die Unterdrückung der Prüfungsergebnisse zu entfernen.

```
aws iot delete-audit-suppression \
```

```
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>"
```

Verwenden Sie den Befehl list-audit-suppressions, um die Löschung zu bestätigen.

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

Ausgabe:

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

In diesem Tutorial haben wir Ihnen gezeigt, wie Sie eine Prüfung vom Typ `Device certificate expiring` in der Konsole und der CLI unterdrücken. Weitere Informationen zur Unterdrückung von Prüfungsergebnissen finden Sie unter Unterdrückungen von Prüfergebnissen.

Audit

Ein AWS IoT Device Defender-Audit untersucht die konto- und gerätebezogenen Einstellungen und Richtlinien, um sicherzustellen, dass Sicherheitsmaßnahmen vorhanden sind. Ein Audit kann Ihnen helfen, Abweichungen von bewährten Sicherheitsmethoden oder Zugriffsrichtlinien zu erkennen (z. B. mehrere Geräte mit derselben Identität oder übermäßig tolerante Richtlinien, die es einem Gerät ermöglichen, Daten für viele andere Geräte zu lesen und zu aktualisieren). Sie können Audits nach Bedarf (On-Demand-Audits) durchführen oder regelmäßig einplanen (geplante Audits).

Ein AWS IoT Device Defender-Audit führt eine Reihe von vordefinierten Prüfungen für bewährte Methoden für die IoT-Sicherheit und Geräteschwachstellen durch. Beispiele für vordefinierte Prüfungen sind Richtlinien, die das Lesen oder Aktualisieren von Daten auf mehreren Geräten erlauben, Geräte, die eine gemeinsame Identität teilen (X.508-Zertifikat), oder Zertifikate, die ablaufen oder widerrufen wurden, aber noch aktiv sind.

Schweregrad des Problems

Der Schweregrad des Problems gibt den Grad der Bedenken an, der mit jedem festgestellten Fall der Nichteinhaltung verbunden ist, und die empfohlene Zeit für die Behebung.

Kritisch

Nonkonforme Audit-Prüfungen mit diesem Schweregrad identifizieren Probleme, die dringend Aufmerksamkeit erfordern. Kritische Probleme ermöglichen es Angreifern, mit wenig Raffinesse und ohne Insiderwissen oder spezielle Anmeldeinformationen einfachen Zugriff auf Ihre Komponenten oder die Kontrolle darüber zu erlangen.

Hoch

Nonkonforme Audit-Prüfungen mit diesem Schweregrad erfordern eine dringende Untersuchungs- und Korrekturplanung, nachdem kritische Probleme behoben wurden. Wie kritische Probleme ermöglichen Probleme mit hohem Schweregrad Angreifern häufig Zugriff auf Ihre Komponenten oder die Kontrolle darüber. Probleme mit hohem Schweregrad sind jedoch oft schwieriger auszunutzen. Möglicherweise sind spezielle Tools, Insiderwissen oder bestimmte Einrichtungen erforderlich.

Mittel

Nonkonforme Audit-Prüfungen mit diesem Schweregrad stellen Probleme dar, die im Rahmen der kontinuierlichen Wartung der Sicherheitseinstellungen beachtet werden müssen. Probleme

mit mittlerem Schweregrad können negative Auswirkungen auf den Betrieb haben, wie z. B. ungeplante Ausfälle aufgrund von Fehlfunktionen der Sicherheitskontrollen. Diese Probleme können auch Angreifern eingeschränkten Zugriff auf oder die Kontrolle über Ihre Komponenten verschaffen oder Aspekte ihrer böswillige Aktivitäten erleichtern.

Niedrig

Nonkonforme Audit-Prüfungen mit diesem Schweregrad weisen häufig darauf hin, dass bewährte Sicherheitsmethoden übersehen oder umgangen wurden. Obwohl sie selbst keine unmittelbaren Auswirkungen auf die Sicherheit haben, können diese Lücken von Angreifern ausgenutzt werden. Wie Probleme mit mittlerem Schweregrad erfordern Probleme mit geringem Schweregrad im Rahmen der kontinuierlichen Wartung der Sicherheitseinstellungen Aufmerksamkeit.

Nächste Schritte

Informationen zu den Arten von Audit-Prüfungen, die durchgeführt werden können, finden Sie unter Auditprüfungen. Informationen zu Service Quotas, die für Audits gelten, finden Sie unter Service Quotas.

Auditprüfungen

> **Note**
>
> Wenn Sie eine Prüfung aktivieren, beginnt die Datenerfassung sofort. Wenn in Ihrem Konto eine große Datenmenge erfasst werden muss, sind die Ergebnisse der Prüfung möglicherweise erst nach einiger Zeit verfügbar.

Folgende Auditprüfungen werden unterstützt:

- Zwischenzertifizierungsstelle für aktive Gerätezertifikate gesperrt
- Das gesperrte Zertifikatstellen-Zertifikat ist immer noch aktiv.
- Gerätezertifikat geteilt
- Gerätezertifikat-Schlüsselqualität
- Qualität der Zertifizierungsstellen–Zertifikatschlüssel
- Übermäßig permissive nicht-authentifizierte Amazon Cognito-Rolle
- Übermäßig permissive authentifizierte Cognito-Rolle

- Übermäßig permissive AWS IoT-Richtlinien
- Die AWS IoT-Richtlinie ist möglicherweise falsch konfiguriert
- Zu permissiver Rollenalias
- Der Rollenalias ermöglicht den Zugriff auf ungenutzte Dienste
- Zertifizierungsstellen-Zertifikat läuft ab
- Widersprüchliche MQTT-Client-IDs
- Gerätezertifikat läuft ab.
- Prüfung des Gerätezertifikatalters
- Ein gesperrtes Gerätezertifikat ist weiterhin aktiv
- Die Protokollierung ist deaktiviert

Zwischenzertifizierungsstelle für aktive Gerätezertifikate gesperrt

Verwenden Sie diese Prüfung, um alle zugehörigen Gerätezertifikate zu identifizieren, die trotz Widerruf einer Zwischenzertifizierungsstelle noch aktiv sind.

Diese Prüfung wird wie `INTERMEDIATE_CA_REVOKED_FOR_ACTIVE_DEVICE_CERTIFICATES_CHECK` in der CLI und API angezeigt.

Schweregrad: Kritisch

Details

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung Nichtkonformität findet:

- INTERMEDIATE_CA_REVOKED_BY_ISSUER

Warum dies wichtig ist

Bei der Prüfung der temporären Zertifizierungsstelle, die für aktive Gerätezertifikate gesperrt wurde, werden Geräteidentität und Vertrauen geprüft, indem festgestellt wird, ob aktive Gerätezertifikate in AWS IoT Core vorhanden sind, in denen die ausstellenden Zwischenzertifizierungsstellen in der Zertifizierungsstellenkette gesperrt wurden.

Eine gesperrte Zwischenzertifizierungsstelle sollte nicht mehr zum Signieren anderer Zertifizierungsstellen oder Gerätezertifikate in der Zertifizierungsstellenkette verwendet werden. Neu

hinzugefügte Geräte mit Zertifikaten, die nach dem Widerruf des CA-Zwischenzertifikats mit diesem CA-Zertifikat signiert wurden, stellen möglicherweise ein Sicherheitsrisiko dar.

So lässt es sich beheben

Überprüfen Sie die Aktivitäten zur Registrierung von Gerätezertifikaten für die Zeit nach dem Widerruf des Zertifizierungsstellen-Zertifikats. Befolgen Sie Ihre bewährten Sicherheitsmethoden, um die Situation zu entschärfen. Mögliche Aktionen:

1. Stellen Sie für die betroffenen Geräte neue Zertifikate bereit, die von einer anderen Zertifizierungsstelle signiert sind.
2. Überprüfen Sie, ob die neuen Zertifikate gültig sind und ob die Geräte mit ihnen eine Verbindung herstellen können.
3. Verwenden Sie UpdateCertificate zum Kennzeichnen des alten Zertifikats als REVOKED in AWS IoT. Sie können Abhilfemaßnahmen auch für Folgendes verwenden:
   - Wenden Sie die Abhilfemaßnahme `UPDATE_DEVICE_CERTIFICATE` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
   - Wenden Sie die Abhilfemaßnahme `ADD_THINGS_TO_THING_GROUP` an, um das Geräts zu einer Gruppe hinzuzufügen, über der Aktionen ausgeführt werden können.
   - Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.
   - Überprüfen Sie die Gerätezertifikat-Registrierungsaktivität für die Zeit nach dem Widerruf des Zertifizierungsstellen-Zwischenzertifikats und ziehen Sie in Betracht, alle Gerätezertifikate, die während dieser Zeit ausgestellt wurden, zu widerrufen. Verwenden Sie ListRelatedResourcesForAuditFinding zum Auflisten der mit dem Zertifikatstellen-Zertifikat signierten Gerätezertifikate und UpdateCertificate zum Sperren eines Gerätezertifikats.
   - Entfernen Sie das alte Zertifikat von dem Gerät. (Siehe DetachThingPrincipal.)

   Weitere Informationen finden Sie unter Abschwächungsaktionen.

Das gesperrte Zertifikatstellen-Zertifikat ist immer noch aktiv.

Ein CA-Zertifikat wurde gesperrt, ist in aber weiterhin aktiv AWS IoT.

Diese Prüfung wird wie `REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK` in der CLI und API angezeigt.

Schweregrad: Kritisch

Details

Ein CA-Zertifikat ist in der von der ausstellenden Behörde geführten Zertifikatssperrliste als gesperrt gekennzeichnet, ist in jedoch weiterhin als ACTIVE oder PENDING TRANSFER markiert AWS IoT.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung ein nicht-konformes Zertifizierungsstellen-Zertifikat findet:

- CERTIFICATE_REVOKED_BY_ISSUER

Warum dies wichtig ist

Ein gesperrtes CA-Zertifikat sollte nicht mehr zum Signieren von Gerätezertifikaten verwendet werden. Es wurde möglicherweise widerrufen, da es kompromittiert wurde. Neu hinzugefügte Geräte mit Zertifikaten, die mit diesem CA-Zertifikat signiert wurden, stellen möglicherweise ein Sicherheitsrisiko dar.

So lässt es sich beheben

1. Verwenden Sie UpdateCACertificate zum Kennzeichnen des CA-Zertifikats als INACTIVE in AWS IoT. Sie können Abhilfemaßnahmen auch für Folgendes verwenden:
   - Wenden Sie die Abhilfemaßnahme `UPDATE_CA_CERTIFICATE` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
   - Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, um eine benutzerdefinierte Antwort als Reaktion auf die Amazon SNS-Nachricht zu implementieren.

   Weitere Informationen finden Sie unter Abschwächungsaktionen.
2. Überprüfen Sie die Gerätezertifikat-Registrierungsaktivität für die Zeit nach dem Widerruf des CA-Zertifikats und ziehen Sie in Betracht, alle Gerätezertifikate, die während dieser Zeit ausgestellt wurden, zu widerrufen. Verwenden Sie ListCertificatesByCA zum Auflisten der mit dem CA-Zertifikat signierten Gerätezertifikate und UpdateCertificate zum Sperren eines Gerätezertifikats.

Gerätezertifikat geteilt

Mehrere gleichzeitige Verbindungen verwenden dasselbe X.509-Zertifikat zur Authentifizierung bei AWS IoT.

Diese Prüfung wird wie `DEVICE_CERTIFICATE_SHARED_CHECK` in der CLI und API angezeigt.

Schweregrad: Kritisch

Details

Wenn diese Prüfung im Rahmen eines On-Demand-Audits durchgeführt wird, untersucht sie die Zertifikate und Client-IDs, mit denen Geräte innerhalb der letzten 31 Tage vor dem Start des Audits eine Verbindung hergestellt haben. Bei geplanten Audits untersucht diese Prüfung die Daten ab 2 Stunden vor dem Zeitpunkt, an dem der Audit zuletzt ausgeführt wurde, bis zu 2 Stunden vor dem Zeitpunkt, an dem diese Instance des Audits gestartet wurde. Wenn Sie innerhalb des geprüften Zeitraums Abhilfemaßnahmen ergriffen haben, vermerken Sie, wann die gleichzeitigen Verbindungen hergestellt wurden, um zu bestimmen, ob das Problem weiterhin besteht.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung ein nicht-konformes Zertifikat findet:

- CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES

Außerdem enthalten die von dieser Prüfung zurückgegebenen Ergebnisse die ID des freigegebenen Zertifikats, die IDs der Clients, die mittels des Zertifikats eine Verbindung hergestellt haben, sowie den Zeitpunkt, an dem die Verbindung hergestellt oder getrennt wurde. Die neuesten Ergebnisse werden zuerst aufgelistet.

Warum dies wichtig ist

Jedes Gerät sollte sich mit einem eindeutigen Zertifikat bei authentifizieren AWS IoT. Wenn mehrere Geräte dasselbe Zertifikat verwenden, kann dies darauf hindeuten, dass ein Gerät kompromittiert wurde. Seine Identität wurde möglicherweise geklont, um das System noch mehr zu kompromittieren.

So lässt es sich beheben

Stellen Sie sicher, dass das Gerätezertifikat nicht kompromittiert wurde. Ist dies der Fall, befolgen Sie Ihre bewährten Sicherheitsmethoden, um die Situation zu entschärfen.

Wenn Sie dasselbe Zertifikat auf mehreren Geräten verwenden, sind die folgenden Maßnahmen zu empfehlen:

1. Stellen Sie neue, eindeutige Zertifikate bereit und fügen Sie sie an jedes Gerät an.

2. Überprüfen Sie, ob die neuen Zertifikate gültig sind und ob die Geräte mit ihnen eine Verbindung herstellen können.
3. Verwenden Sie UpdateCertificate zum Kennzeichnen des alten Zertifikats als REVOKED in AWS IoT. Sie können auch Maßnahmen zur Schadensbegrenzung verwenden, um Folgendes zu tun:
   - Wenden Sie die Abhilfemaßnahme `UPDATE_DEVICE_CERTIFICATE` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
   - Wenden Sie die Abhilfemaßnahme `ADD_THINGS_TO_THING_GROUP` an, um das Geräts zu einer Gruppe hinzuzufügen, über der Aktionen ausgeführt werden können.
   - Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

   Weitere Informationen finden Sie unter Abschwächungsaktionen.
4. Trennen Sie das alte Zertifikat von jedem der Geräte.

Gerätezertifikat-Schlüsselqualität

AWS IoT-Kunden verlassen sich häufig auf die gegenseitige TLS-Authentifizierung mit X.509-Zertifikaten für die Authentifizierung beim AWS IoT-Nachrichtenbroker. Diese Zertifikate und ihre Zertifizierungsstellenzertifikate müssen im AWS IoT-Konto registriert werden, bevor sie verwendet werden. AWS IoT führt grundlegende Integritätsprüfungen für diese Zertifikate durch, wenn sie registriert sind. Folgendes wird überprüft:

- Sie müssen ein gültiges Format haben.
- Sie müssen von einer registrierten Zertifizierungsstelle signiert sein.
- Sie müssen sich noch innerhalb ihrer Gültigkeitsdauer befinden (mit anderen Worten, sie dürfen noch nicht abgelaufen sein).
- Ihre kryptografischen Schlüssel müssen eine erforderliche Mindestgröße aufweisen (RSA-Schlüssel müssen mindestens 2048 Bit groß sein).

Diese Audit-Prüfung bietet die folgenden zusätzlichen Tests zur Qualität Ihres kryptografischen Schlüssels:

- CVE-2008-0166: Überprüft, ob der Schlüssel mit OpenSSL 0.9.8c-1 bis zu Versionen vor 0.9.8g-9 auf einem Debian-basierten Betriebssystem generiert wurde. Diese Versionen von OpenSSL

verwenden einen Zufallszahlengenerator, der vorhersehbare Zahlen generiert. So wird es Angreifern erleichtert, Brute-Force-Rate-Angriffe gegen kryptografische Schlüssel durchzuführen.

- CVE-2017-15361: Überprüft, ob der Schlüssel von der Infineon RSA-Bibliothek 1.02.013 in der Infineon Trusted Platform Module-Firmware (TPM) generiert wurde, z. B. Versionen vor 0000000000000422 – 4.34, vor 000000000000062b – 6.43 und vor 0000000000008521 – 133.33. Diese Bibliothek handhabt die Generierung von RSA-Schlüsseln falsch, was es Angreifern erleichtert, einige kryptographische Schutzmechanismen durch gezielte Angriffe zu umgehen. Beispiele für betroffene Technologien sind BitLocker mit TPM 1.2, YubiKey 4 (vor 4.3.5) PGP-Schlüsselgenerierung und die Verschlüsselungsfunktion für zwischengespeicherte Benutzerdaten in Chrome OS.

AWS IoT Device Defender meldet Zertifikate als nicht konform, wenn sie diese Tests nicht bestehen.

Diese Prüfung wird wie `DEVICE_CERTIFICATE_KEY_QUALITY_CHECK` in der CLI und API angezeigt.

Schweregrad: Kritisch

Details

Diese Prüfung gilt für Gerätezertifikate mit dem Status ACTIVE oder PENDING_TRANSFER.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung ein nicht-konformes Zertifikat findet:

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

Warum dies wichtig ist

Wenn ein Gerät ein anfälliges Zertifikat verwendet, können Angreifer dieses Gerät leichter gefährden.

So lässt es sich beheben

Aktualisieren Sie Ihre Gerätezertifikate, um diese durch bekannte Schwachstellen zu ersetzen.

Wenn Sie dasselbe Zertifikat auf mehreren Geräten verwenden, sind die folgenden Maßnahmen zu empfehlen:

1. Stellen Sie neue, eindeutige Zertifikate bereit und fügen Sie sie an jedes Gerät an.

2. Überprüfen Sie, ob die neuen Zertifikate gültig sind und ob die Geräte mit ihnen eine Verbindung herstellen können.
3. Verwenden Sie UpdateCertificate zum Kennzeichnen des alten Zertifikats als REVOKED in AWS IoT. Sie können Abhilfemaßnahmen auch für Folgendes verwenden:
   - Wenden Sie die Abhilfemaßnahme `UPDATE_DEVICE_CERTIFICATE` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
   - Wenden Sie die Abhilfemaßnahme `ADD_THINGS_TO_THING_GROUP` an, um das Geräts zu einer Gruppe hinzuzufügen, über der Aktionen ausgeführt werden können.
   - Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

   Weitere Informationen finden Sie unter Abschwächungsaktionen.
4. Trennen Sie das alte Zertifikat von jedem der Geräte.

Qualität der Zertifizierungsstellen–Zertifikatschlüssel

AWS IoT-Kunden verlassen sich häufig auf die gegenseitige TLS-Authentifizierung mit X.509-Zertifikaten für die Authentifizierung beim AWS IoT-Nachrichtenbroker. Diese Zertifikate und ihre Zertifizierungsstellenzertifikate müssen im AWS IoT-Konto registriert werden, bevor sie verwendet werden. AWS IoT führt grundlegende Integritätsprüfungen für diese Zertifikate durch, wenn sie registriert sind, einschließlich:

- Die Zertifikate haben ein gültiges Format.
- Die Zertifikate befinden sich innerhalb ihrer Gültigkeitsdauer (mit anderen Worten, nicht abgelaufen).
- Ihre kryptografischen Schlüssel weisen eine erforderliche Mindestgröße auf (RSA-Schlüssel müssen mindestens 2048 Bit groß sein).

Diese Audit-Prüfung bietet die folgenden zusätzlichen Tests zur Qualität Ihres kryptografischen Schlüssels:

- CVE-2008-0166: Überprüft, ob der Schlüssel mit OpenSSL 0.9.8c-1 bis zu Versionen vor 0.9.8g-9 auf einem Debian-basierten Betriebssystem generiert wurde. Diese Versionen von OpenSSL verwenden einen Zufallszahlengenerator, der vorhersehbare Zahlen generiert. So wird es Angreifern erleichtert, Brute-Force-Rate-Angriffe gegen kryptografische Schlüssel durchzuführen.

- CVE-2017-15361: Überprüft, ob der Schlüssel von der Infineon RSA-Bibliothek 1.02.013 in der Infineon Trusted Platform Module-Firmware (TPM) generiert wurde, z. B. Versionen vor 0000000000000422 – 4.34, vor 000000000000062b – 6.43 und vor 0000000000008521 – 133.33. Diese Bibliothek handhabt die Generierung von RSA-Schlüsseln falsch, was es Angreifern erleichtert, einige kryptographische Schutzmechanismen durch gezielte Angriffe zu umgehen. Beispiele für betroffene Technologien sind BitLocker mit TPM 1.2, YubiKey 4 (vor 4.3.5) PGP-Schlüsselgenerierung und die Verschlüsselungsfunktion für zwischengespeicherte Benutzerdaten in Chrome OS.

AWS IoT Device Defender meldet Zertifikate als nicht konform, wenn sie diese Tests nicht bestehen.

Diese Prüfung wird wie `CA_CERTIFICATE_KEY_QUALITY_CHECK` in der CLI und API angezeigt.

Schweregrad: Kritisch

Details

Diese Prüfung gilt für Zertifizierungsstellen-Zertifikate mit dem Status ACTIVE oder PENDING_TRANSFER.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung ein nicht-konformes Zertifikat findet:

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

Warum dies wichtig ist

Neu hinzugefügte Geräte, die mit diesem CA-Zertifikat signiert wurden, stellen möglicherweise ein Sicherheitsrisiko dar.

So lässt es sich beheben

1. Verwenden Sie UpdateCACertificate zum Kennzeichnen des CA-Zertifikats als INACTIVE in AWS IoT. Sie können Abhilfemaßnahmen auch für Folgendes verwenden:
   - Wenden Sie die Abhilfemaßnahme `UPDATE_CA_CERTIFICATE` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
   - Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

2. Überprüfen Sie die Gerätezertifikat-Registrierungsaktivität für die Zeit nach dem Widerruf des CA-Zertifikats und ziehen Sie in Betracht, alle Gerätezertifikate, die während dieser Zeit ausgestellt wurden, zu widerrufen. (Verwenden Sie ListCertificatesByCA zum Auflisten der mit dem CA-Zertifikat signierten Gerätezertifikate und UpdateCertificate zum Sperren eines Gerätezertifikats.)

Übermäßig permissive nicht-authentifizierte Amazon Cognito-Rolle

Eine Richtlinie, die an eine nicht authentifizierte Amazon-Cognito-Identitätspool-Rolle angefügt ist, wird als übermäßig permissiv angesehen, da sie zum Ausführen folgender AWS IoT-Aktionen berechtigt:

- Objekte verwalten oder ändern
- Objekt-Verwaltungsdaten lesen
- Nicht auf das Objekt bezogene Daten oder Ressourcen verwalten

Oder da sie zum Ausführen der folgenden AWS IoT-Aktionen für eine breite Palette von Geräten berechtigt:

- MQTT zum Verbinden mit sowie zum Veröffentlichen und Abonnieren von reservierten Themen (einschließlich Schatten- oder Aufgabenausführungsdaten) verwenden
- API-Befehle zum Lesen und Ändern von Schatten- oder Auftragsausführungsdaten verwenden

Im Allgemeinen sollten Geräte, die eine Verbindung über eine nicht authentifizierte Amazon Cognito-Identitätspool-Rolle herstellen, nur eingeschränkt zum Veröffentlichen und Abonnieren von objektspezifischen MQTT-Themen oder zum Lesen und Ändern objektspezifischer Daten bezüglich Shadow- oder Auftragsausführungsdaten mithilfe von API-Befehlen berechtigt sein.

Diese Prüfung wird wie `UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK` in der CLI und API angezeigt.

Schweregrad: Kritisch

Details

Für diese Prüfung überprüft AWS IoT Device Defender alle Amazon-Cognito-Identitätspools, die verwendet wurden, um innerhalb der letzten 31 Tage vor Ausführung der Prüfung eine Verbindung mit dem AWS IoT-Nachrichten-Broker aufzubauen. Alle Amazon Cognito-Identitätspools, über die eine authentifizierte oder eine nicht authentifizierte Amazon Cognito-Identität eine Verbindung hergestellt hat, werden in den Audit eingeschlossen.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung eine nicht-konforme, nicht authentifizierte Amazon Cognito-Identitätspoolrolle findet:

- ALLOWS_ACCESS_TO_IOT_ADMIN_ACTIONS
- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

Warum dies wichtig ist

Da nicht authentifizierte Identitäten niemals vom Benutzer authentifiziert werden, stellen sie ein viel größer Risiko als authentifizierte Amazon Cognito-Identitäten dar. Wenn eine nicht authentifizierte Identität kompromittiert wird, könnte sie mit administrativen Aktionen Kontoeinstellungen ändern, Ressourcen löschen oder Zugriff auf vertrauliche Daten erhalten. Oder mit umfassendem Zugriff auf Geräte-Einstellungen kann sie Schatten und Aufträge für alle Geräte in Ihrem Konto aufrufen oder ändern. Eine Gastbenutzer könnte die Berechtigungen verwenden, um Ihre gesamte Flotte zu kompromittieren oder einen DDoS-Angriff mit Nachrichten zu starten.

So lässt es sich beheben

Eine Richtlinie, die an eine nicht authentifizierte Amazon Cognito-Identitätspool-Rolle angefügt ist, sollte einem Gerät nur die Berechtigungen gewähren, die es benötigt, um seine Arbeit erledigen zu können. Wir empfehlen die folgenden Schritte:

1. Erstellen Sie eine neue regelkonforme Rolle.
2. Erstellen Sie einen neuen Amazon Cognito-Identitätspool und ordnen Sie ihm die regelkonforme Rolle zu.
3. Vergewissern Sie sich, dass Ihre Identitäten über den neuen Pool Zugriff auf AWS IoT zugreifen können.
4. Sobald die Verifizierung abgeschlossen ist, fügen Sie die konforme Rolle an den Amazon Cognito-Identitätspool an, der als nicht konform markiert wurde.

Sie können Abhilfemaßnahmen auch für Folgendes verwenden:

- Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, um eine benutzerdefinierte Antwort als Reaktion auf die Amazon SNS-Nachricht zu implementieren.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

Objekte verwalten oder ändern

Die folgenden AWS IoT-API-Aktionen werden verwendet, um Objekte zu verwalten oder zu ändern. Daher sollte die Berechtigung zum Ausführen dieser Aktionen keinen Geräten gewährt werden, die eine Verbindung über einen nicht authentifizierten Amazon Cognito-Identitätspool herstellen.

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

Jede Rolle, die Berechtigungen zum Ausführen dieser Aktionen auf sogar nur einer einzigen Ressource erteilt, gilt als nicht konform.

Objekt-Verwaltungsdaten lesen

Die folgenden AWS IoT-API-Aktionen werden zum Lesen oder Ändern von Objektdaten verwendet. Geräte, die eine Verbindung über einen nicht authentifizierten Amazon Cognito-Identitätspool herstellen, sollten nicht zum Ausführen dieser Aktionen berechtigt werden.

- `DescribeThing`

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

Example

- Nicht regelkonform:

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowIoTThingOperations",
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:thing/name-of-thing"
      ]
    }
  ]
}
```

  Dies berechtigt das Gerät dazu, die angegebenen Aktionen auszuführen, auch wenn es nur für ein Objekt berechtigt ist.

Verwalten von Nicht-Objekten

Geräte, die sich über einen nicht authentifizierten Amazon Cognito-Identitätspool verbinden, sollten nicht zum Ausführen von anderen AWS IoT-API-Aktionen als denen, die in diesen Abschnitten angesprochen werden, berechtigt werden. Um Ihr Konto mit einer Anwendung zu verwalten, die eine Verbindung über einen nicht authentifizierten Amazon Cognito-Identitätspool herstellt, erstellen Sie einen separaten Identitätspool, der nicht von Geräten genutzt wird.

Abonnieren von/Veröffentlichen in MQTT-Themen

MQTT-Nachrichten werden über den AWS IoT Message Broker gesendet und von Geräten für die Ausführung zahlreicher Aktionen verwendet, darunter zum Abrufen und Ändern des Schattenstatus und des Aufgabenausführungsstatus. Eine Richtlinie, die ein Gerät zum Verbinden mit und Veröffentlichen oder Abonnieren von MQTT-Nachrichten berechtigt, sollte diese Aktionen wie folgt auf bestimmte Ressourcen einschränken:

Verbinden

- Nicht regelkonform:

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

  Das Platzhalterzeichen „*“ erlaubt jedem Gerät, eine Verbindung mit herzustellen AWS IoT.

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

  Außer wenn `iot:Connection.Thing.IsAttached` in den Bedingungsschlüsseln auf „true“ gesetzt wurde, ist dies gleichbedeutend mit dem Platzhalter „*“ im vorherigen Beispiel.

- Regelkonform:

  JSON

  ```
  {
    "Version":"2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/
  ${iot:Connection.Thing.ThingName}"
        ]
      }
    ]
  }
  ```

Die Ressourcenspezifikation enthält eine Variable, die dem Gerätenamen entspricht, der zum Herstellen der Verbindung verwendet wurde. Die Bedingungsanweisung schränkt die Berechtigung weiter ein, indem überprüft wird, ob das vom MQTT-Client verwendete Zertifikat mit dem übereinstimmt, das dem Objekt mit dem verwendeten Namen zugewiesen ist.

Veröffentlichen

- Nicht regelkonform:

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

Dies berechtigt das Gerät dazu, den Shadow eines jeden beliebigen Geräts zu aktualisieren (* = alle Geräte).

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

Dies berechtigt das Gerät dazu, den Schatten eines beliebigen Geräts zu lesen, aktualisieren oder löschen.

- Regelkonform:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
            ]
        }
    ]
}
```

Die Ressourcenspezifikation enthält einen Platzhalter, dieser findet als Übereinstimmung aber nur schattenbezogene Themen für das Gerät, dessen Objektname zum Herstellen der Verbindung verwendet wird.

Abonnieren

- Nicht regelkonform:

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

Dies berechtigt das Gerät dazu, reservierte Shadow- oder Auftragsthemen für alle Geräte zu abonnieren.

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

Vergleichbar mit dem vorherigen Beispiel, aber mit dem #-Platzhalter.

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

Dies berechtigt das Gerät dazu, Schattenaktualisierungen auf jedem beliebigen Gerät (+ = alle Geräte) anzuzeigen.

- Regelkonform:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*",
                "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
            ]
        }
    ]
```

```
}
```

Die Ressourcenspezifikationen enthalten Platzhalter, diese finden als Übereinstimmung aber nur schattenbezogene Themen und auftragsbezogene Themen für das Gerät, dessen Objektname zum Herstellen der Verbindung verwendet wird.

Empfangen

- Regelkonform:

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

Dies ist zulässig, da das Gerät nur Nachrichten von Themen empfangen kann, für die es über die Berechtigung zum Abonnieren verfügt.

Lesen/Ändern von Schatten- oder Auftragsdaten

Eine Richtlinie, die ein Gerät dazu berechtigt, eine API-Aktion zum Aufrufen oder Ändern von Device Shadows oder Auftragsausführungsdaten auszuführen, sollte diese Aktionen auf bestimmte Ressourcen einschränken. Im Folgenden sind die API-Aktionen aufgeführt:

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

Example

- Nicht regelkonform:

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

Dies berechtigt das Gerät dazu, die angegebene Aktion für ein beliebiges Objekt auszuführen.

- Regelkonform:

JSON

```
{
  "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:us-east-1:123456789012:thing/MyThing1",
        "arn:aws:iot:us-east-1:123456789012:thing/MyThing2"
      ]
    }
  ]
}
```

Auf diese Weise kann das Gerät die angegebenen Aktionen nur für zwei Objekte ausführen.

Übermäßig permissive authentifizierte Cognito-Rolle

Eine Richtlinie, die an eine authentifizierte Amazon Cognito-Identitätspool-Rolle angefügt ist, wird als übermäßig permissiv angesehen, da sie zum Ausführen folgender AWS IoT-Aktionen berechtigt:

- Objekte verwalten oder ändern
- Nicht auf das Objekt bezogene Daten oder Ressourcen verwalten

Oder da sie zum Ausführen der folgenden AWS IoT-Aktionen für eine breite Palette von Geräten berechtigt:

- Objekt-Verwaltungsdaten lesen

- MQTT zum Verbinden mit und zum Veröffentlichen und Abonnieren von reservierten Themen (einschließlich Shadow- oder Auftragsausführungsdaten) verwenden
- API-Befehle zum Lesen und Ändern von Schatten- oder Auftragsausführungsdaten verwenden

Im Allgemeinen sollten Geräte, die eine Verbindung über eine authentifizierte Amazon Cognito-Identitätspool-Rolle herstellen, nur eingeschränkt zum Lesen objektspezfischer Verwaltungsdaten, zum Veröffentlichen und Abonnieren von objektspezifischen MQTT-Themen oder zum Lesen und Ändern objektspezischer Daten bezüglich Schatten- oder Auftragsausführungsdaten mithilfe von API-Befehlen berechtigt sein.

Diese Prüfung wird wie `AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK` in der CLI und API angezeigt.

Schweregrad: Kritisch

Details

Für diese Prüfung überprüft AWS IoT Device Defender alle Amazon-Cognito-Identitätspools, die verwendet wurden, um innerhalb der letzten 31 Tage vor Ausführung der Prüfung eine Verbindung mit dem AWS IoT-Nachrichten-Broker aufzubauen. Alle Amazon Cognito-Identitätspools, über die eine authentifizierte oder eine nicht authentifizierte Amazon Cognito-Identität eine Verbindung hergestellt hat, werden in den Audit eingeschlossen.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung eine nicht-konforme, authentifizierte Amazon Cognito-Identitätspool-Rolle findet:

- 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

Warum dies wichtig ist

Wenn eine authentifizierte Identität kompromittiert wird, könnte sie mit administrativen Aktionen Kontoeinstellungen ändern, Ressourcen löschen oder Zugriff auf vertrauliche Daten erlangen.

So lässt es sich beheben

Eine Richtlinie, die an eine authentifizierte Amazon Cognito-Identitätspool-Rolle angefügt ist, sollte einem Gerät nur die Berechtigungen gewähren, die es benötigt, um seine Arbeit erledigen zu können. Wir empfehlen die folgenden Schritte:

1. Erstellen Sie eine neue regelkonforme Rolle.
2. Erstellen Sie einen neuen Amazon Cognito-Identitätspool und ordnen Sie ihm die regelkonforme Rolle zu.
3. Vergewissern Sie sich, dass Ihre Identitäten über den neuen Pool Zugriff auf AWS IoT zugreifen können.
4. Sobald die Verifizierung abgeschlossen ist, fügen Sie die konforme Rolle an den Amazon Cognito-Identitätspool an, der als nicht konform markiert wurde.

Sie können Abhilfemaßnahmen auch für Folgendes verwenden:

- Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, um eine benutzerdefinierte Antwort als Reaktion auf die Amazon SNS-Nachricht zu implementieren.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

Objekte verwalten oder ändern

Die folgenden AWS IoT-API-Aktionen werden verwendet, um Objekte zu verwalten oder zu ändern. Daher sollte die Berechtigung zum Ausführen dieser Aktionen keinen Geräten gewährt werden, die eine Verbindung über einen authentifizierten Amazon Cognito-Identitätspool herstellen:

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

- `UpdateThing`
- `UpdateThingGroupsForThing`

Jede Rolle, die Berechtigungen zum Ausführen dieser Aktionen auf sogar nur einer einzigen Ressource erteilt, gilt als nicht konform.

Verwalten von Nicht-Objekten

Geräte, die sich über einen authentifizierten Amazon Cognito-Identitätspool verbinden, sollten nicht zum Ausführen von anderen AWS IoT-API-Aktionen als denen, die in diesen Abschnitten angesprochen werden, berechtigt werden. Wenn Sie Ihr Konto mit einer Anwendung verwalten möchten, die eine Verbindung über einen authentifizierten Amazon Cognito-Identitätspool herstellt, erstellen Sie einen separaten Identitätspool, der nicht von Geräten genutzt wird.

Objekt-Verwaltungsdaten lesen

Die folgenden AWS IoT-API-Aktionen werden zum Lesen von Objektdaten verwendet. Daher sollten Geräte, die eine Verbindung über einen authentifizierten Amazon Cognito-Identitätspool herstellen, zum Ausführen dieser Aktionen für nur eine begrenzte Objektgruppe berechtigt werden:

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

- Nicht regelkonform:

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

  Dies berechtigt das Gerät dazu, die angegebene Aktion für ein beliebiges Objekt auszuführen.
- Regelkonform:

  JSON

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

```
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:thing/MyThing"
      ]
    }
  ]
}
```

Dies berechtigt das Gerät dazu, die angegebenen Aktionen für nur ein bestimmtes Objekt auszuführen.

- Regelkonform:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeThing",
                "iot:ListJobExecutionsForThing",
                "iot:ListThingGroupsForThing",
                "iot:ListThingPrincipals"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:thing/MyThing*"
            ]
        }
    ]
}
```

Dies ist konform, da der Ressource, obwohl sie mit einem Platzhalter (*) angegeben wird, eine bestimmte Zeichenfolge vorangestellt ist. Dies schränkt die aufgerufene Objektgruppe auf Objekte ein, deren Namen das angegebene Präfix enthalten.

- Nicht regelkonform:

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

Dies berechtigt das Gerät dazu, die angegebene Aktion für ein beliebiges Objekt auszuführen.

- Regelkonform:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:thing/MyThing"
      ]
    }
  ]
}
```

Dies berechtigt das Gerät dazu, die angegebenen Aktionen für nur ein bestimmtes Objekt auszuführen.

- Regelkonform:

JSON

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

```
                "iot:ListThingGroupsForThing",
                "iot:ListThingPrincipals"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:thing/MyThing*"
            ]
        }
    ]
}
```

Dies ist konform, da der Ressource, obwohl sie mit einem Platzhalter (*) angegeben wird, eine bestimmte Zeichenfolge vorangestellt ist. Dies schränkt die aufgerufene Objektgruppe auf Objekte ein, deren Namen das angegebene Präfix enthalten.

Abonnieren von/Veröffentlichen in MQTT-Themen

MQTT-Nachrichten werden über den AWS IoT Message Broker gesendet und von Geräten für die Ausführung vieler verschiedener Aktionen verwendet, darunter zum Abrufen und Ändern des Schattenstatus und des Auftragsausführungsstatus. Eine Richtlinie, die ein Gerät zum Verbinden mit und Veröffentlichen oder Abonnieren von MQTT-Nachrichten berechtigt, sollte diese Aktionen wie folgt auf bestimmte Ressourcen einschränken:

Verbinden

- Nicht regelkonform:

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

  Das Platzhalterzeichen „*“ erlaubt jedem Gerät, eine Verbindung mit herzustellen AWS IoT.

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

  Außer wenn `iot:Connection.Thing.IsAttached` in den Bedingungsschlüsseln auf „true“ gesetzt wurde, ist dies gleichbedeutend mit dem Platzhalter „*“ im vorherigen Beispiel.

- Regelkonform:

  JSON

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
      ]
    }
  ]
}
```

Die Ressourcenspezifikation enthält eine Variable, die dem Gerätenamen der Verbindungsherstellung entspricht. Die Bedingungsanweisung schränkt die Berechtigung noch weiter ein, indem geprüft wird, ob das vom MQTT-Client verwendete Zertifikat mit dem Zertifikat übereinstimmt, das dem Objekt mit dem verwendeten Namen zugeordnet ist.

Veröffentlichen

- Nicht regelkonform:

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

Dies berechtigt das Gerät dazu, den Shadow eines jeden beliebigen Geräts zu aktualisieren (* = alle Geräte).

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

Dies berechtigt das Gerät dazu, den Schatten eines jeden beliebigen Geräts zu lesen/aktualisieren/löschen.

- Regelkonform:

  JSON

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

```
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
            ]
        }
    ]
}
```

Die Ressourcenspezifikation enthält einen Platzhalter, dieser findet als Übereinstimmung aber nur schattenbezogene Themen für das Gerät, dessen Objektname zum Herstellen der Verbindung verwendet wird.

Abonnieren

- Nicht regelkonform:

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

Dies berechtigt das Gerät dazu, reservierte Shadow- oder Auftragsthemen für alle Geräte zu abonnieren.

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

Vergleichbar mit dem vorherigen Beispiel, aber mit dem #-Platzhalter.

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

Dies berechtigt das Gerät dazu, Schattenaktualisierungen auf jedem beliebigen Gerät (+ = alle Geräte) anzuzeigen.

- Regelkonform:

  JSON

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

```
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*",
                "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
            ]
        }
    ]
}
```

Die Ressourcenspezifikationen enthalten Platzhalter, diese finden als Übereinstimmung aber nur schattenbezogene Themen und auftragsbezogene Themen für das Gerät, dessen Objektname zum Herstellen der Verbindung verwendet wird.

Empfangen

- Regelkonform:

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

  Dies ist konform, da das Gerät nur Nachrichten von Themen empfangen kann, für die es über die Berechtigung zum Abonnieren verfügt.

Lesen oder Ändern von Schatten- oder Auftragsdaten

Eine Richtlinie, die ein Gerät dazu berechtigt, eine API-Aktion zum Aufrufen oder Ändern von Device Shadows oder Auftragsausführungsdaten auszuführen, sollte diese Aktionen auf bestimmte Ressourcen einschränken. Im Folgenden sind die API-Aktionen aufgeführt:

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

Beispiele

- Nicht regelkonform:

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

  Dies berechtigt das Gerät dazu, die angegebene Aktion für ein beliebiges Objekt auszuführen.

- Regelkonform:

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:DeleteThingShadow",
        "iot:GetThingShadow",
        "iot:UpdateThingShadow",
        "iot:DescribeJobExecution",
        "iotjobsdata:DescribeJobExecution",
        "iotjobsdata:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:thing/MyThing1",
        "arn:aws:iot:us-east-1:123456789012:thing/MyThing2"
      ]
    }
  ]
}
```

  Dies berechtigt das Gerät dazu, die angegebenen Aktionen für nur zwei bestimmte Objekte auszuführen.

Übermäßig permissive AWS IoT-Richtlinien

Eine AWS IoT-Richtlinie gewährt Berechtigungen, die zu weit gefasst oder uneingeschränkt sind. Sie berechtigt zum Senden oder Empfangen von MQTT-Nachrichten für eine breite Palette von Geräten

oder zum Aufrufen oder Ändern von Shadow- und Auftragsausführungsdaten für eine breite Palette von Geräten.

Im Allgemeinen sollte eine Richtlinie für ein Gerät den Zugriff auf Ressourcen gewähren, die nur dem betreffenden Gerät und keinem anderen oder nur sehr wenigen anderen Geräten zugeordnet sind. Bis auf einige Ausnahmen gilt die Verwendung eines Platzhalters (z. B. „*“) zur Angabe von Ressourcen in einer solchen Richtlinie als zu weit gefasst oder uneingeschränkt.

Diese Prüfung wird wie `IOT_POLICY_OVERLY_PERMISSIVE_CHECK` in der CLI und API angezeigt.

Schweregrad: Kritisch

Details

Der folgende Ursachencode wird zurückgegeben, wenn diese Prüfung eine nicht-konforme AWS IoT-Richtlinie findet:

- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

Warum dies wichtig ist

Ein Zertifikat, eine Amazon Cognito-Identität oder eine Objektgruppe mit einer übermäßig großzügigen Richtlinie kann sich, sofern kompromittiert, auf die Sicherheit Ihres gesamten Kontos auswirken. Ein Angreifer könnte einen solchen umfassenden Zugriff zum Lesen oder Ändern von Schatten, Aufträgen oder Auftragsausführungen für alle Ihre Geräte verwenden. Alternativ könnte ein Angreifer ein kompromittiertes Zertifikat verwenden, um böswillige Geräte zu verbinden oder einen DDoS-Angriffen in Ihrem Netzwerk zu starten.

So lässt es sich beheben

Gehen Sie wie folgt vor, um alle nicht konformen Richtlinien zu korrigieren, die an Objekte, Objektgruppen oder andere Entitäten angefügt sind:

1. Verwenden Sie CreatePolicyVersion zum Erstellen einer neuen, konformen Version der Richtlinie. Legen Sie das Flag `setAsDefault` auf „true“ fest. (Dies macht diese neue Version operativ für alle Entitäten, die die Richtlinie verwenden.)
2. Fordern Sie mit ListTargetsForPolicy eine Liste von Zielen (Zertifikate, Objektgruppen) an, an die die Richtlinie angefügt ist, und bestimmen Sie, welche Geräte in den Gruppen enthalten sein oder welche die Zertifikate zum Herstellen einer Verbindung verwenden sollen.

3. Stellen Sie sicher, dass alle zugeordneten Geräte in der Lage sind, eine Verbindung mit herzustellen AWS IoT. Wenn ein Gerät keine Verbindung herstellen kann, führen Sie mit SetPolicyVersion ein Rollback der Standardrichtlinie auf die vorherige Version durch, überarbeiten Sie die Richtlinie und versuchen Sie es erneut.

Sie können mit Abhilfemaßnahmen für Folgendes verwenden:

- Wenden Sie die Abhilfemaßnahme `REPLACE_DEFAULT_POLICY_VERSION` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
- Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

Verwenden Sie AWS IoT Core-Richtlinienvariablen, um dynamisch auf spezifische AWS IoT-Ressourcen in Ihren Richtlinien zu verweisen.

MQTT-Berechtigungen

MQTT-Nachrichten werden über den AWS IoT Message Broker gesendet und von Geräten für die Ausführung zahlreicher Aktionen verwendet, darunter zum Abrufen und Ändern des Schattenstatus und des Aufgabenausführungsstatus. Eine Richtlinie, die ein Gerät zum Verbinden mit und Veröffentlichen oder Abonnieren von MQTT-Nachrichten berechtigt, sollte diese Aktionen wie folgt auf bestimmte Ressourcen einschränken:

Verbinden

- Nicht regelkonform:

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

  Das Platzhalterzeichen „*“ erlaubt jedem Gerät, eine Verbindung mit herzustellen AWS IoT.

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

  Außer wenn `iot:Connection.Thing.IsAttached` in den Bedingungsschlüsseln auf „true“ gesetzt wurde, ist dies gleichbedeutend mit dem Platzhalter „*“ im vorherigen Beispiel.
- Regelkonform:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
      ]
    }
  ]
}
```

Die Ressourcenspezifikation enthält eine Variable, die dem Gerätenamen entspricht, der zum Herstellen der Verbindung verwendet wurde. Die Bedingungsanweisung schränkt die Berechtigung weiter ein, indem überprüft wird, ob das vom MQTT-Client verwendete Zertifikat mit dem übereinstimmt, das dem Objekt mit dem verwendeten Namen zugewiesen ist.

Veröffentlichen

- Nicht regelkonform:

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

Dies berechtigt das Gerät dazu, den Shadow eines jeden beliebigen Geräts zu aktualisieren (* = alle Geräte).

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

Dies berechtigt das Gerät dazu, den Schatten eines beliebigen Geräts zu lesen, aktualisieren oder löschen.

- Regelkonform:

  JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
            ]
        }
    ]
}
```

Die Ressourcenspezifikation enthält einen Platzhalter, dieser findet als Übereinstimmung aber nur schattenbezogene Themen für das Gerät, dessen Objektname zum Herstellen der Verbindung verwendet wird.

Abonnieren

- Nicht regelkonform:

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

  Dies berechtigt das Gerät dazu, reservierte Shadow- oder Auftragsthemen für alle Geräte zu abonnieren.

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

  Vergleichbar mit dem vorherigen Beispiel, aber mit dem #-Platzhalter.

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

  Dies berechtigt das Gerät dazu, Schattenaktualisierungen auf jedem beliebigen Gerät (+ = alle Geräte) anzuzeigen.

- Regelkonform:

  JSON

  ```
  {
  ```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*",
                "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
            ]
        }
    ]
}
```

Die Ressourcenspezifikationen enthalten Platzhalter, diese finden als Übereinstimmung aber nur schattenbezogene Themen und auftragsbezogene Themen für das Gerät, dessen Objektname zum Herstellen der Verbindung verwendet wird.

Empfangen

- Regelkonform:

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

  Dies ist konform, da das Gerät nur Nachrichten von Themen empfangen kann, für die es über die Berechtigung zum Abonnieren verfügt.

Schatten- und Auftragsberechtigungen

Eine Richtlinie, die ein Gerät dazu berechtigt, eine API-Aktion zum Aufrufen oder Ändern von Device Shadows oder Auftragsausführungsdaten auszuführen, sollte diese Aktionen auf bestimmte Ressourcen einschränken. Im Folgenden sind die API-Aktionen aufgeführt:

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

- `GetPendingJobExecutions`
- `StartNextPendingJobExecution`
- `UpdateJobExecution`

Beispiele

- Nicht regelkonform:

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

  Dies berechtigt das Gerät dazu, die angegebene Aktion für ein beliebiges Objekt auszuführen.

- Regelkonform:

  JSON

```
{
  "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:us-east-1:123456789012:thing/MyThing1",
        "arn:aws:iot:us-east-1:123456789012:thing/MyThing2"
      ]
    }
  ]
}
```

  Dies berechtigt das Gerät dazu, die angegebenen Aktionen für nur zwei bestimmte Objekte auszuführen.

Die AWS IoT-Richtlinie ist möglicherweise falsch konfiguriert

Eine AWS IoT-Richtlinie wurde als potenziell falsch konfiguriert identifiziert. Falsch konfigurierte Richtlinien, einschließlich übermäßig permissiver Richtlinien, können zu Sicherheitsvorfällen führen, z. B. wenn Geräten der Zugriff auf unbeabsichtigte Ressourcen ermöglicht wird.

Die Überprüfung auf potenziell falsch konfigurierte AWS IoT-Richtlinien ist eine Warnung, mit der Sie sicherstellen müssen, dass nur beabsichtigte Aktionen zulässig sind, bevor Sie die Richtlinie aktualisieren.

In der CLI und API wird diese Prüfung als `IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK` angezeigt.

Schweregrad: Mittel

Details

AWS IoT gibt den folgenden Ursachencode zurück, wenn bei dieser Prüfung eine potenziell falsch konfigurierte AWS IoT-Richtlinie gefunden wird:

- POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT
- TOPIC_FILTERS_INTENDED_TO_DENY_ALLOWED_USING_WILDCARDS

Warum dies wichtig ist

Falsch konfigurierte Richtlinien können unbeabsichtigte Folgen haben, da sie Geräten mehr Berechtigungen gewähren als erforderlich. Wir empfehlen, die Richtlinie sorgfältig zu prüfen, um den Zugriff auf Ressourcen einzuschränken und Sicherheitsbedrohungen zu verhindern.

Die Richtlinie enthält MQTT-Platzhalter im Beispiel für eine Anweisung zur Zugriffsverweigerung (Deny-Anweisung)

Die AWS IoT-Richtlinie ist möglicherweise falsch konfiguriert und prüft MQTT-Platzhalterzeichen (+ oder #) in Anweisungen zur Zugriffsverweigerung. Platzhalter werden von AWS IoT-Richtlinien als wörtliche Zeichenfolgen behandelt und können dazu führen, dass die Richtlinie zu permissiv ist.

Das folgende Beispiel soll verhindern, dass Sie Themen abonnieren, die sich auf die Verwendung von `building/control_room` unter Verwendung des MQTT-Platzhalters # in Richtlinien beziehen. MQTT-Platzhalter haben jedoch keine Platzhalterbedeutung in AWS IoT-Richtlinien und Geräten, die für `building/control_room/data1` abonniert werden können.

Bei der Prüfung, ob die AWS IoT-Richtlinie möglicherweise falsch konfiguriert ist, wird diese Richtlinie mit einem Ursachencode `POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT` gekennzeichnet.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iot:Subscribe",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/building/
*"
        },
        {
            "Effect": "Deny",
            "Action": "iot:Subscribe",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/building/
control_room/#"
        },
        {
            "Effect": "Allow",
            "Action": "iot:Receive",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topic/building/*"
        }
    ]
}
```

Nachstehend finden Sie ein Beispiel für eine ordnungsgemäß konfigurierte Richtlinie. Geräte sind nicht berechtigt, Unterthemen von `building/control_room/` zu abonnieren, und sie verfügen nicht über die Berechtigung, Nachrichten aus Unterthemen von `building/control_room/` zu empfangen.

JSON

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

```
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/building/
control_room/*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:us-east-1:123456789012:topic/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:us-east-1:123456789012:topic/building/
control_room/*"
    }
  ]
}
```

Themenfilter, die das Zulassen verbieten sollen (Beispiel: Platzhalter)

Die folgende Beispielrichtlinie soll verhindern, dass Sie das Abonnieren von Themen in Bezug auf `building/control_room` ablehnen, indem Sie die Ressource `building/control_room/*` ablehnen. Geräte können jedoch Anfragen zum Abonnieren an `building/#` senden, und Nachrichten zu allen Themen in Bezug auf `building` empfangen, einschließlich `building/control_room/data1`.

Bei der Prüfung, ob die AWS IoT-Richtlinie möglicherweise falsch konfiguriert ist, wird diese Richtlinie mit einem Ursachencode `TOPIC_FILTERS_INTENDED_TO_DENY_ALLOWED_USING_WILDCARDS` gekennzeichnet.

Die folgende Beispielrichtlinie verfügt über Berechtigungen zum Empfangen von Nachrichten auf `building/control_room topics`:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iot:Subscribe",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/building/
*"
        },
        {
            "Effect": "Deny",
            "Action": "iot:Subscribe",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/building/
control_room/*"
        },
        {
            "Effect": "Allow",
            "Action": "iot:Receive",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topic/building/*"
        }
    ]
}
```

Nachstehend finden Sie ein Beispiel für eine ordnungsgemäß konfigurierte Richtlinie. Geräte sind nicht berechtigt, Unterthemen von `building/control_room/` zu abonnieren, und sie verfügen nicht über die Berechtigung, Nachrichten aus Unterthemen von `building/control_room/` zu empfangen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iot:Subscribe",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/building/
*"
```

```
        },
        {
            "Effect": "Deny",
            "Action": "iot:Subscribe",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/building/
control_room/*"
        },
        {
            "Effect": "Allow",
            "Action": "iot:Receive",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topic/building/*"
        },
        {
            "Effect": "Deny",
            "Action": "iot:Receive",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topic/building/
control_room/*"
        }
    ]
}
```

> **Note**
>
> Bei dieser Prüfung werden möglicherweise falsch positive Ergebnisse gemeldet. Wir empfehlen Ihnen, alle markierten Richtlinien zu überprüfen und Ressourcen mithilfe von Prüfungsunterdrückungen als falsch positiv zu kennzeichnen.

So lässt es sich beheben

Diese Prüfung kennzeichnet potenziell falsch konfigurierte Richtlinien, sodass es zu falsch positiven Ergebnissen kommen kann. Markieren Sie alle falsch positiven Ergebnisse mithilfe von Prüfunterdrückungen, damit sie künftig nicht mehr gemeldet werden.

Sie können auch folgendermaßen vorgehen, um alle nicht konformen Richtlinien zu korrigieren, die an Objekte, Objektgruppen oder andere Entitäten angefügt sind:

1. Verwenden Sie CreatePolicyVersion zum Erstellen einer neuen, konformen Version der Richtlinie. Legen Sie das Flag `setAsDefault` auf „true“ fest. (Dies macht diese neue Version operativ für alle Entitäten, die die Richtlinie verwenden.)

Beispiele für das Erstellen von AWS IoT-Richtlinien für allgemeine Anwendungsfälle finden Sie unter Richtlinienbeispiele zum Veröffentlichen/Abonnieren im AWS IoT Core-Entwicklerhandbuch.

2. Stellen Sie sicher, dass alle zugeordneten Geräte in der Lage sind, eine Verbindung mit herzustellen AWS IoT. Wenn ein Gerät keine Verbindung herstellen kann, führen Sie mit SetPolicyVersion ein Rollback der Standardrichtlinie auf die vorherige Version durch, überarbeiten Sie die Richtlinie und versuchen Sie es erneut.

Sie können mit Abhilfemaßnahmen für Folgendes verwenden:

- Wenden Sie die Abhilfemaßnahme `REPLACE_DEFAULT_POLICY_VERSION` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
- Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

Verwenden Sie IoT-Core-Richtlinienvariablen im AWS IoT Core-Entwicklerhandbuch, um dynamisch auf spezifische AWS IoT-Ressourcen in Ihren Richtlinien zu verweisen.

Zu permissiver Rollenalias

Ein AWS IoT-Rollenalias stellt einen Mechanismus für verbundene Geräte zur Authentifizierung bei der AWS IoT mithilfe von X.509-Zertifikaten bereit. Anschließend erhalten Sie AWS-Anmeldeinformationen mit kurzer Lebensdauer von einer IAM-Rolle, die einem AWS IoT-Rollenalias zugeordnet ist. Die Berechtigungen für diese Anmeldeinformationen müssen mithilfe von Zugriffsrichtlinien mit Authentifizierungskontextvariablen eingeschränkt werden. Wenn Ihre Richtlinien nicht korrekt konfiguriert sind, können Sie einer Eskalation von Privilegien ausgesetzt sein. Diese Audit-Prüfung stellt sicher, dass die temporären Anmeldeinformationen, die von AWS IoT-Rollenaliasen bereitgestellt werden, nicht übermäßig großzügig sind.

Diese Prüfung wird ausgelöst, wenn eine der folgenden Bedingungen gefunden wird:

- Die Richtlinie stellt Administratorberechtigungen für alle Dienste bereit, die im vergangenen Jahr von diesem Rollenalias verwendet wurden (z. B. „iot:*“, „dynamodb:*“, „iam:*“ usw.).
- Die Richtlinie bietet breiten Zugriff auf Metadatenaktionen, Zugriff auf eingeschränkte AWS IoT-Aktionen oder umfassenden Zugriff auf AWS IoT-Aktionen auf Datenebene.

- Die Richtlinie bietet Zugriff auf Sicherheitsüberwachungsdienste wie „iam“, „cloudtrail“, „guardduty“, „inspector“ oder „trustedadvisor“.

Diese Prüfung wird wie `IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK` in der CLI und API angezeigt.

Schweregrad: Kritisch

Details

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung eine nicht-konforme IoT-Richtlinie findet:

- 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

Warum dies wichtig ist

Indem Sie Berechtigungen auf diejenigen beschränken, die zum normalen Betrieb eines Geräts erforderlich sind, reduzieren Sie die Risiken für Ihr Konto, wenn ein Gerät gefährdet ist.

So lässt es sich beheben

Gehen Sie wie folgt vor, um alle nicht konformen Richtlinien zu korrigieren, die an Objekte, Objektgruppen oder andere Entitäten angefügt sind:

1. Führen Sie die Schritte unter Autorisieren von direkten Aufrufen von AWS-Services mithilfe des AWS IoT Core-Anmeldeinformationsanbieters aus, um eine restriktivere Richtlinie auf Ihren Rollenalias anzuwenden.

Sie können mit Abhilfemaßnahmen für Folgendes verwenden:

- Wenden Sie die `PUBLISH_FINDINGS_TO_SNS`-Abhilfemaßnahme an, wenn Sie eine benutzerdefinierte Aktion als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

Der Rollenalias ermöglicht den Zugriff auf ungenutzte Dienste

Ein AWS IoT-Rollenalias stellt einen Mechanismus für verbundene Geräte zur Authentifizierung bei der AWS IoT mithilfe von X.509-Zertifikaten bereit. Anschließend erhalten Sie AWS-Anmeldeinformationen mit kurzer Lebensdauer von einer IAM-Rolle, die einem AWS IoT-Rollenalias zugeordnet ist. Die Berechtigungen für diese Anmeldeinformationen müssen mithilfe von Zugriffsrichtlinien mit Authentifizierungskontextvariablen eingeschränkt werden. Wenn Ihre Richtlinien nicht korrekt konfiguriert sind, können Sie einer Eskalation von Privilegien ausgesetzt sein. Diese Audit-Prüfung stellt sicher, dass die temporären Anmeldeinformationen, die von AWS IoT-Rollenaliasen bereitgestellt werden, nicht übermäßig großzügig sind.

Diese Prüfung wird ausgelöst, wenn der Rollenalias Zugriff auf Dienste hat, die im letzten Jahr nicht für das AWS IoT-Gerät verwendet wurden. Die Prüfung meldet beispielsweise, wenn Sie eine mit dem Rollenalias verknüpfte IAM-Rolle haben, die nur AWS IoT im letzten Jahr verwendet hat, wobei die der Rolle angehängte Richtlinie jedoch auch Berechtigungen für `"iam:getRole"` und `"dynamodb:PutItem"` gewährt.

Diese Prüfung wird wie `IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK` in der CLI und API angezeigt.

Schweregrad: Mittel

Details

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung eine nicht-konforme AWS IoT-Richtlinie findet:

- ALLOWS_ACCESS_TO_UNUSED_SERVICES

Warum dies wichtig ist

Indem Sie Berechtigungen auf jene Dienste beschränken, die zum normalen Betrieb eines Geräts erforderlich sind, reduzieren Sie die Risiken für Ihr Konto, wenn ein Gerät gefährdet ist.

So lässt es sich beheben

Gehen Sie wie folgt vor, um alle nicht konformen Richtlinien zu korrigieren, die an Objekte, Objektgruppen oder andere Entitäten angefügt sind:

1. Führen Sie die Schritte unter Autorisieren von direkten Aufrufen von AWS-Services mithilfe des AWS IoT Core-Anmeldeinformationsanbieters aus, um eine restriktivere Richtlinie auf Ihren Rollenalias anzuwenden.

Sie können mit Abhilfemaßnahmen für Folgendes verwenden:

- Wenden Sie die `PUBLISH_FINDINGS_TO_SNS`-Abhilfemaßnahme an, wenn Sie eine benutzerdefinierte Aktion als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

Zertifizierungsstellen-Zertifikat läuft ab

Ein Zertifizierungsstellen-Zertifikat läuft in 30 Tagen ab oder ist abgelaufen.

Diese Prüfung wird wie `CA_CERTIFICATE_EXPIRING_CHECK` in der CLI und API angezeigt.

Schweregrad: Mittel

Details

Diese Prüfung gilt für Zertifizierungsstellen-Zertifikate mit dem Status ACTIVE oder PENDING_TRANSFER.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung ein nicht-konformes Zertifizierungsstellen-Zertifikat findet:

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

Warum dies wichtig ist

Ein abgelaufenes Zertifizierungsstellen-Zertifikat sollte nicht zum Signieren neuer Gerätezertifikate verwendet werden.

So lässt es sich beheben

Lesen Sie in Ihren bewährten Methoden das weitere Vorgehen nach. Mögliche Aktionen:

1. Registrieren Sie ein neues Zertifizierungsstellen-Zertifikat bei AWS IoT.

2. Überprüfen Sie, ob Sie Gerätezertifikate mit dem neuen Zertifizierungsstellen-Zertifikat signieren können.
3. Verwenden Sie UpdateCACertificate zum Kennzeichnen des alten Zertifizierungsstellen-Zertifikats als INACTIVE in AWS IoT. Sie können auch Maßnahmen zur Schadensbegrenzung verwenden, um Folgendes zu tun:
   - Wenden Sie die Abhilfemaßnahme `UPDATE_CA_CERTIFICATE` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
   - Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

   Weitere Informationen finden Sie unter Abschwächungsaktionen.

Widersprüchliche MQTT-Client-IDs

Mehrere Geräte stellen eine Verbindung mittels derselben Client-ID her.

Diese Prüfung wird wie `CONFLICTING_CLIENT_IDS_CHECK` in der CLI und API angezeigt.

Schweregrad: Hoch

Details

Es wurden mehrere Verbindungen mit der gleichen Client-ID hergestellt, wodurch ein bereits verbundenes Gerät getrennt wurde. Die MQTT-Spezifikation gestattet nur eine aktive Verbindung pro Client-ID, d. h. wenn sich ein anderes Gerät mit derselben Client-ID verbindet, wird die vorherige Verbindung abgebrochen.

Wenn diese Prüfung im Rahmen eines On-Demand-Audits durchgeführt wird, untersucht sie, wie Client-IDs verwendet werden, um innerhalb der letzten 31 Tage vor dem Start des Audits eine Verbindung herzustellen. Bei geplanten Audits untersucht diese Prüfung die Daten ab dem Zeitpunkt, an dem der Audit zuletzt ausgeführt wurde, bis zu dem Zeitpunkt, an dem diese Instance des Audits gestartet wurde. Wenn Sie innerhalb des geprüften Zeitraums Abhilfemaßnahmen ergriffen haben, vermerken Sie, wann Verbindungen hergestellt/getrennt wurden, um zu bestimmen, ob das Problem weiterhin besteht.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung Nichtkonformität findet:

- DUPLICATE_CLIENT_ID_ACROSS_CONNECTIONS

Die durch diese Prüfung zurückgegebenen Ergebnisse umfassen die Client-ID, mit der die Verbindung hergestellt wurde, die Prinzipal-IDs und die Trennungszeiten. Die neuesten Ergebnisse werden zuerst aufgelistet.

Warum dies wichtig ist

Geräte mit widersprüchlichen IDs werden gezwungen, sich ständig neu zu verbinden. Dies kann dazu führen, dass Nachrichten verloren gehen oder ein Gerät keine Verbindung herstellen kann.

Dies kann darauf hinweisen, dass ein Gerät oder die Anmeldeinformationen eines Geräts kompromittiert wurden und möglicherweise Teil eines DDoS-Angriffs sind. Es ist auch möglich, dass Geräte im Konto nicht ordnungsgemäß konfiguriert sind oder ein Gerät aufgrund einer schlechten Verbindung gezwungen ist, sich mehrmals pro Minute neu zu verbinden.

So lässt es sich beheben

Registrieren Sie jedes Gerät als eindeutiges Objekt in AWS IoT und verwenden Sie den Objektnamen als Client-ID für die Verbindung. Verwenden Sie alternativ eine UUID als eine Client-ID, wenn Sie eine Geräteverbindung über MQTT herstellen. Sie können Abhilfemaßnahmen auch für Folgendes verwenden:

- Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

Gerätezertifikat läuft ab.

Ein Gerätezertifikat läuft innerhalb des konfigurierten Schwellenwerts für den Zeitraum ab oder ist bereits abgelaufen. Der Schwellenwert für die Prüfung des Zertifikatablaufs kann mit einem Wert zwischen 30 Tagen (mindestens) und 3652 Tagen (maximal; 10 Jahre) konfiguriert werden. Der Standardwert ist 30 Tage.

Diese Prüfung wird wie `DEVICE_CERTIFICATE_EXPIRING_CHECK` in der CLI und API angezeigt.

Schweregrad: Mittel

Details

Diese Prüfung gilt für Gerätezertifikate mit dem Status ACTIVE oder PENDING_TRANSFER.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung ein nicht-konformes Gerätezertifikat findet:

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

Warum dies wichtig ist

Ein Gerätezertifikat sollte nicht verwendet werden, nachdem es abgelaufen ist.

Konfiguration der Prüfung des Gerätezertifikatablaufs

Mit dieser Konfiguration können Sie die Zertifikate für Ihre gesamte Geräteflotte überwachen und Warnungen für Zertifikate erhalten, die sich dem Ablaufdatum nähern. Wenn Sie beispielsweise benachrichtigt werden möchten, wenn Zertifikate innerhalb von 30 Tagen ablaufen, können Sie die Prüfung wie folgt konfigurieren:

```
{
    "roleArn": "your-audit-role-arn",
    "auditCheckConfigurations": {
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": true,
            "configuration": {
                "CERT_EXPIRATION_THRESHOLD_IN_DAYS": "30"
            }
        }
    }
}
```

So lässt es sich beheben

Lesen Sie in Ihren bewährten Methoden das weitere Vorgehen nach. Mögliche Aktionen:

1. Stellen Sie ein neues Zertifikat bereit und fügen Sie es an das Gerät an.
2. Überprüfen Sie, ob das neue Zertifikat gültig ist und das Gerät damit eine Verbindung herstellen kann.
3. Verwenden Sie UpdateCertificate zum Kennzeichnen des alten Zertifikats als INACTIVE in AWS IoT. Sie können Abhilfemaßnahmen auch für Folgendes verwenden:

- Wenden Sie die Abhilfemaßnahme `UPDATE_DEVICE_CERTIFICATE` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
- Wenden Sie die Abhilfemaßnahme `ADD_THINGS_TO_THING_GROUP` an, um das Geräts zu einer Gruppe hinzuzufügen, über der Aktionen ausgeführt werden können.
- Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

4. Entfernen Sie das alte Zertifikat von dem Gerät. (Siehe DetachThingPrincipal.)

Prüfung des Gerätezertifikatalters

Bei dieser Prüfung werden Sie benachrichtigt, wenn ein Gerätezertifikat für eine Anzahl von Tagen aktiv ist, die größer als oder gleich der von Ihnen angegebenen Anzahl von Tagen ist. Mithilfe dieser Prüfung bleiben Sie hinsichtlich des Status Ihrer Zertifikate auf dem Laufenden, sodass Sie in regelmäßigen Abständen rechtzeitig handeln können, unabhängig davon, wann das Zertifikat abläuft. Dies verbessert die Sicherheit, da das Risiko für eine Kompromittierung des Zertifikats verringert wird.

Der Schwellenwert für die Prüfung des Zertifikatsalters kann für einen Wert zwischen 30 Tagen (mindestens) und 3652 Tagen (maximal; 10 Jahre) konfiguriert werden. Der Standardwert ist 365 Tage.

Diese Prüfung wird wie `DEVICE_CERTIFICATE_AGE_CHECK` in der CLI und API angezeigt. Diese Prüfung ist standardmäßig deaktiviert; Schweregrad:Niedrig

Details

Diese Prüfung gilt für Gerätezertifikate mit dem Status ACTIVE oder PENDING_TRANSFER.
Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung ein nicht-konformes Gerätezertifikat findet:

- CERTIFICATE_PAST_AGE_THRESHOLD

Konfiguration der Prüfung des Gerätezertifikatalters

Mithilfe dieser Konfiguration können Sie Warnungen für die Zertifikatsrotation an die spezifischen Anforderungen Ihrer Flotte anpassen. So erzielen Sie für alle Geräte ein hohes Maß an Sicherheit. Sie können diese Prüfung mithilfe der `UpdateAccountAuditConfiguration`-API konfigurieren.

Wenn Sie beispielsweise benachrichtigt werden möchten, wenn Zertifikate seit mehr als 365 Tagen aktiv sind, können Sie die Prüfung wie folgt konfigurieren:

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

Ein gesperrtes Gerätezertifikat ist weiterhin aktiv

Ein gesperrtes Gerätezertifikat ist weiterhin aktiv.

Diese Prüfung wird wie `REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK` in der CLI und API angezeigt.

Schweregrad: Mittel

Details

Ein Gerätezertifikat befindet sich auf der Zertifikatsperrliste der Zertifizierungsstelle (CA), ist in AWS IoT aber weiterhin aktiv.

Diese Prüfung gilt für Gerätezertifikate mit dem Status ACTIVE oder PENDING_TRANSFER.

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung Nichtkonformität findet:

- CERTIFICATE_REVOKED_BY_ISSUER

Warum dies wichtig ist

Ein Gerätezertifikat wird gewöhnlich Regel gesperrt, wenn es kompromittiert wurde. Es ist möglich, dass es aufgrund eines Fehlers oder aus Versehen noch nicht in AWS IoT gesperrt wurde.

So lässt es sich beheben

Stellen Sie sicher, dass das Gerätezertifikat nicht kompromittiert wurde. Ist dies der Fall, befolgen Sie Ihre bewährten Sicherheitsmethoden, um die Situation zu entschärfen. Mögliche Aktionen:

1. Stellen Sie ein neues Zertifikat für das Gerät bereit.
2. Überprüfen Sie, ob das neue Zertifikat gültig ist und das Gerät damit eine Verbindung herstellen kann.
3. Verwenden Sie UpdateCertificate zum Kennzeichnen des alten Zertifikats als REVOKED in AWS IoT. Sie können Abhilfemaßnahmen auch für Folgendes verwenden:
   - Wenden Sie die Abhilfemaßnahme `UPDATE_DEVICE_CERTIFICATE` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
   - Wenden Sie die Abhilfemaßnahme `ADD_THINGS_TO_THING_GROUP` an, um das Geräts zu einer Gruppe hinzuzufügen, über der Aktionen ausgeführt werden können.
   - Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

   Weitere Informationen finden Sie unter Abschwächungsaktionen.
4. Entfernen Sie das alte Zertifikat von dem Gerät. (Siehe DetachThingPrincipal.)

Die Protokollierung ist deaktiviert

In Amazon CloudWatch sind keine AWS IoT-Protokolle aktiviert. Überprüft sowohl die V1- als auch die V2-Protokollierung.

Diese Prüfung wird wie `LOGGING_DISABLED_CHECK` in der CLI und API angezeigt.

Schweregrad: Niedrig

Details

Die folgenden Ursachencodes werden zurückgegeben, wenn diese Prüfung Nichtkonformität findet:

- LOGGING_DISABLED

Warum dies wichtig ist

AWS IoT-Protokolle in CloudWatch bieten eine Übersicht über Verhaltensweisen in AWS IoT, darunter Authentifizierung und unerwartetes Herstellen und Trennen von Verbindungen, woran zu erkennen ist, dass ein Gerät kompromittiert wurde.

So lässt es sich beheben

Aktivieren Sie AWS IoT Protokolle in CloudWatch. Weitere Informationen finden Sie unter Protokollierung und Überwachung im AWS IoT Core-Entwicklerhandbuch. Sie können Abhilfemaßnahmen auch für Folgendes verwenden:

- Wenden Sie die Abhilfemaßnahme `ENABLE_IOT_LOGGING` auf Ihre Prüfungsergebnisse an, um diese Änderung vorzunehmen.
- Wenden Sie die Abhilfemaßnahme `PUBLISH_FINDINGS_TO_SNS` an, wenn Sie eine benutzerdefinierte Antwort als Antwort auf die Amazon SNS-Nachricht implementieren möchten.

Weitere Informationen finden Sie unter Abschwächungsaktionen.

Prüfungsbefehle

Verwalten von Prüfungseinstellungen

Konfigurieren Sie mit `UpdateAccountAuditConfiguration` die Audit-Einstellungen für Ihr Konto. Mit diesem Befehl können Sie die Prüfungen aktivieren, die für Audits verfügbar sein sollen, optionale Benachrichtigungen einrichten und Berechtigungen konfigurieren.

Überprüfen Sie diese Einstellungen mit `DescribeAccountAuditConfiguration`.

Verwenden Sie `DeleteAccountAuditConfiguration` zum Löschen Ihrer Audit-Einstellungen. Damit werden alle Standardwerte wiederhergestellt und Audits auf effiziente Weise deaktiviert, da alle Prüfungen standardmäßig deaktiviert sind.

UpdateAccountAuditConfiguration

Konfiguriert erstmals oder von neuem die Device Defender-Audit-Einstellungen für dieses Konto. Die Einstellungen legen u. a. fest, wie Audit-Benachrichtigungen gesendet werden und welche Prüfungen aktiviert oder deaktiviert sind.

Syntax

```
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`-Format

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

`cli-input-json`-Felder

Name	Typ	Beschreibung
roleArn	Zeichenfolge Länge max. 2048, min. 20	Der ARN der Rolle, die AWS IoT zum Zugriff auf Informationen zu Ihren Geräten, Richtlinien, Zertifikaten und anderen Elemente berechtigt, wenn ein Audit durchgeführt wird.
auditNotificationTargetConfigurations	map	Informationen über die Ziele, an die Audit-Benachrichtigungen gesendet werden.

Name	Typ	Beschreibung
targetArn	Zeichenfolge	Der ARN des Ziels (SNS-Thema), an das Audit-Ben achrichtigungen gesendet werden.
roleArn	Zeichenfolge Länge max. 2048, min. 20	Der ARN der Rolle, die die Berechtigung zum Senden von Benachrichtigungen an das Ziel erteilt.
aktiviert	boolesch	True, wenn Benachric htigungen an das Ziel aktiviert sind.

Name	Typ	Beschreibung
auditCheckConfigurations	map	Gibt an, welche Audit-Prüfungen für dieses Konto aktiviert und deaktiviert sind. Zeigen Sie mit `DescribeAccountAuditConfiguration` die Liste aller Prüfungen an, einschließlich derjenigen, die derzeit aktiviert sind. Mit dem Sammeln bestimmter Daten wird möglicherweise sofort begonnen, wenn bestimmte Prüfungen aktiviert werden. Wenn eine Prüfung deaktiviert ist, werden alle bisher gesammelten Informationen bezüglich der Prüfung gelöscht wird. Sie können keine Prüfungen deaktivieren, die Teil eines geplanten Audits sind. Sie müssen zuerst die Prüfung aus dem geplanten Audit oder das geplante Audit selbst löschen. Dieser Parameter ist beim ersten Aufruf von `UpdateAccountAuditConfiguration` erforderlich und muss mindestens eine aktivierte Prüfung angeben.

Name	Typ	Beschreibung
aktiviert	boolesch	True, wenn diese Audit-Prüfung für dieses Konto aktiviert ist.
configuration	map	(Optional) Benutzerdefinierte Konfigurationen für bestimmte Auditprüfungen, z. B. `CERT_AGE_THRESHOLD_IN_DAYS` und `CERT_EXPIRATION_THRESHOLD_IN_DAYS`, mit denen Sie festlegen können, wann Sie über das Alter eines Zertifikats und dessen bevorstehenden Ablauf benachrichtigt werden möchten.

Ausgabe

Keine

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

DescribeAccountAuditConfiguration

Ruft Informationen zu den Device Defender Audit-Einstellungen für dieses Konto ab. Die Einstellungen legen u. a. fest, wie Audit-Benachrichtigungen gesendet werden und welche Prüfungen aktiviert oder deaktiviert sind.

Syntax

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

`cli-input-json`-Format

```
{
}
```

Ausgabe

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

CLI-Ausgabefelder

Name	Typ	Beschreibung
roleArn	Zeichenfolge	Der ARN der Rolle, die AWS IoT zum Zugriff auf Informati

Name	Typ	Beschreibung
	Länge max. 2048, min. 20	onen zu Ihren Geräten, Richtlinien, Zertifikaten und anderen Elemente berechtig t, wenn ein Audit durchgeführt wird. Beim ersten Aufruf von `UpdateAccountAudit Configuration` ist dieser Parameter erforderlich.
auditNotificationTargetConf igurations	map	Informationen über die Ziele, an die Audit-Benachrichti gungen für dieses Konto gesendet werden.
targetArn	Zeichenfolge	Der ARN des Ziels (SNS-Thema), an das Audit-Ben achrichtigungen gesendet werden.
roleArn	Zeichenfolge Länge max. 2048, min. 20	Der ARN der Rolle, die die Berechtigung zum Senden von Benachrichtigungen an das Ziel erteilt.
aktiviert	boolesch	True, wenn Benachric htigungen an das Ziel aktiviert sind.
auditCheckConfigurations	map	Gibt an, welche Audit-Prü fungen für dieses Konto aktiviert und deaktiviert sind.
aktiviert	boolesch	True, wenn diese Audit-Prü fung für dieses Konto aktiviert ist.

Name	Typ	Beschreibung
configuration	map	(Optional) Stellt spezielle Konfigurationen für bestimmte Auditprüfungen bereit, z. B. zum maximal zulässigen Zertifikatalter oder zur Anzahl der Tage vor dem Ablauf, bei deren Erreichen eine Warnung ausgelöst werden soll.

Fehler

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

DeleteAccountAuditConfiguration

Stellt die Standardeinstellungen für Device Defender-Audits für dieses Konto wieder her. Alle Konfigurationsdaten, die Sie eingegeben haben, werden gelöscht, und alle Auditprüfungen werden auf „Deaktiviert“ zurückgesetzt.

Syntax

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

`cli-input-json`-Format

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

`cli-input-json`-Felder

Name	Typ	Beschreibung
deleteScheduledAudits	boolesch	Ist diese Option auf „true“ gesetzt, werden alle geplanten Audits gelöscht.

Ausgabe

Keine

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ResourceNotFoundException`

Die angegebene Ressource ist nicht vorhanden.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

Planen von Audits

Erstellen Sie einen oder mehrere geplante Audits mithilfe von `CreateScheduledAudit`. Mit diesem Befehl können Sie die Prüfungen angeben, die Sie während eines Audits durchführen möchten, und wie oft der Audit ausgeführt werden soll.

Verfolgen Sie Ihre geplanten Audits mit `ListScheduledAudits` und `DescribeScheduledAudit`.

Ändern Sie einen vorhandenen geplanten Audit mit mit `UpdateScheduledAudit` oder löschen Sie ihn mit `DeleteScheduledAudit`.

CreateScheduledAudit

Erstellt einen geplanten Audit, der in einem angegebenen Zeitintervall ausgeführt wird.

Syntax

```
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`-Format

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

`cli-input-json`-Felder

Name	Typ	Beschreibung
frequency	Zeichenfolge	Wie oft der geplante Audit stattfindet. Kann entweder DAILY, WEEKLY, BIWEEKLY oder MONTHLY sein. Der

Name	Typ	Beschreibung
		tatsächliche Startzeitpunkt der einzelnen Prüfungen wird vom System bestimmt. enum: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	Zeichenfolge pattern: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	Das Tag des Monats, an dem der geplante Audit stattfindet. Kann 1 bis 31 oder LAST sein. Dieses Feld ist erforderlich, wenn der Parameter `frequency` auf MONTHLY eingestellt ist. Wenn die Tage 29-31 angegeben werden und der Monat nicht so viele Tage hat, erfolgt der Audit am Tag LAST des Monats.
dayOfWeek	Zeichenfolge	Das Tag der Woche, an dem der geplante Audit stattfindet. Kann entweder SUN, MON, TUE,WED, THU, FRI oder SAT sein. Dieses Feld ist erforderlich, wenn der Parameter `frequency` auf WEEKLY oder BIWEEKLY eingestellt ist. enum: SUN | MON | TUE | WED | THU | FRI | SAT

Name	Typ	Beschreibung
targetCheckNames	list member: AuditCheckName	Gibt an, welche Prüfungen während des geplanten Audits ausgeführt werden. Die Prüfungen müssen für Ihr Konto aktiviert sein. (Verwenden Sie `DescribeAccountAuditConfiguration`, um die Liste aller Prüfungen, einschließlich aller aktivierten, anzuzeigen, oder `UpdateAccountAuditConfiguration`, um auszuwählen, welche Prüfungen aktiviert sind.)
tags	list member: Tag java class: java.util.List	Metadaten, die zum Verwalten der geplanten Prüfung verwendet werden können.
Schlüssel	Zeichenfolge	Der Tag-Schlüssel.
Wert	Zeichenfolge	Der Tag-Wert.
scheduledAuditName	Zeichenfolge length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	Der Name, den Sie dem geplanten Audit zuweisen möchten. (Maximal 128 Zeichen)

Ausgabe

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

CLI-Ausgabefelder

Name	Typ	Beschreibung
scheduledAuditArn	Zeichenfolge	Der ARN des geplanten Audits.

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

`LimitExceededException`

Ein Grenzwert wurde überschritten.

ListScheduledAudits

Listet alle Ihre geplanten Audits auf.

Syntax

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

`cli-input-json`-Format

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

`cli-input-json`-Felder

Name	Typ	Beschreibung
nextToken	Zeichenfolge	Das Token für den nächsten Ergebnissatz.
maxResults	Ganzzahl range- max:250 min:1	Die maximale Anzahl der Ergebnisse, die auf einmal zurückgegeben werden sollen. Der Standardwert ist 25.

Ausgabe

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

CLI-Ausgabefelder

Name	Typ	Beschreibung
scheduledAudits	list member: ScheduledAuditMetadata java class: java.util.List	Die Liste der geplanten Audits.
scheduledAuditName	Zeichenfolge length- max:128 min:1	Der Name des geplanten Audits.

Name	Typ	Beschreibung
	pattern: [a-zA-Z0-9_-]+	
scheduledAuditArn	Zeichenfolge	Der ARN des geplanten Audits.
frequency	Zeichenfolge	Wie oft der geplante Audit stattfindet. enum: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	Zeichenfolge pattern: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	Der Tag des Monats, an dem der geplante Audit ausgeführt wird (wenn der Wert für `frequency` MONTHLY lautet). Wenn die Tage 29-31 angegeben werden und der Monat nicht so viele Tage hat, erfolgt der Audit am Tag LAST des Monats.
dayOfWeek	Zeichenfolge	Der Tag der Woche, an dem der geplante Audit ausgeführt wird (wenn der Wert für `frequency` WEEKLY oder BIWEEKLY lautet). enum: SUN | MON | TUE | WED | THU | FRI | SAT
nextToken	Zeichenfolge	Ein Token, mit dem der nächste Ergebnissatzes abgerufen werden kann, bzw. `null`, wenn keine weiteren Ergebnisse vorliegen.

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

DescribeScheduledAudit

Ruft Informationen zu einem geplanten Audit ab.

Syntax

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

`cli-input-json`-Format

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

`cli-input-json`-Felder

Name	Typ	Beschreibung
scheduledAuditName	Zeichenfolge length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	Der Name des geplanten Audits, dessen Daten Sie abrufen möchten.

Ausgabe

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

CLI-Ausgabefelder

Name	Typ	Beschreibung
frequency	Zeichenfolge	Wie oft der geplante Audit stattfindet. Entweder DAILY, WEEKLY, BIWEEKLY oder MONTHLY. Der tatsächliche Startzeitpunkt der einzelnen Prüfungen wird vom System bestimmt. enum: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	Zeichenfolge pattern: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	Das Tag des Monats, an dem der geplante Audit stattfindet. Kann 1 bis 31 oder LAST sein. Wenn die Tage 29-31 angegeben werden und der Monat nicht so viele Tage hat, erfolgt der Audit am Tag LAST des Monats.
dayOfWeek	Zeichenfolge	Das Tag der Woche, an dem der geplante Audit stattfindet. Entweder SUN, MON, TUE, WED, THU, FRI oder SAT.

Name	Typ	Beschreibung
		enum: SUN | MON | TUE | WED | THU | FRI | SAT
targetCheckNames	list member: AuditCheckName	Gibt an, welche Prüfungen während des geplanten Audits ausgeführt werden. Die Prüfungen müssen für Ihr Konto aktiviert sein. (Verwenden Sie `DescribeAccountAuditConfiguration` , um die Liste aller Prüfungen, einschließlich aller aktivierten, anzuzeigen, oder `UpdateAccountAuditConfiguration` , um auszuwählen, welche Prüfungen aktiviert sind.)
scheduledAuditName	Zeichenfolge length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	Der Name des geplanten Audits.
scheduledAuditArn	Zeichenfolge	Der ARN des geplanten Audits.

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ResourceNotFoundException`

Die angegebene Ressource ist nicht vorhanden.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

UpdateScheduledAudit

Aktualisiert einen geplanten Audit, einschließlich welche Prüfungen durchgeführt werden und wie oft der Audit stattfindet.

Syntax

```
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`-Format

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

`cli-input-json`-Felder

Name	Typ	Beschreibung
frequency	Zeichenfolge	Wie oft der geplante Audit stattfindet. Kann entweder

Name	Typ	Beschreibung
		DAILY, WEEKLY, BIWEEKLY oder MONTHLY sein. Der tatsächliche Startzeitpunkt der einzelnen Prüfungen wird vom System bestimmt. enum: DAILY | WEEKLY | BIWEEKLY | MONTHLY
dayOfMonth	Zeichenfolge pattern: ^([1-9]|[12][0-9]|3[01])$\|^LAST$	Das Tag des Monats, an dem der geplante Audit stattfindet. Kann 1 bis 31 oder LAST sein. Dieses Feld ist erforderlich, wenn der Parameter `frequency` auf MONTHLY eingestellt ist. Wenn die Tage 29-31 angegeben werden und der Monat nicht so viele Tage hat, erfolgt der Audit am Tag LAST des Monats.
dayOfWeek	Zeichenfolge	Das Tag der Woche, an dem der geplante Audit stattfindet. Kann entweder SUN, MON, TUE, WED, THU, FRI oder SAT sein. Dieses Feld ist erforderlich, wenn der Parameter `frequency` auf WEEKLY oder BIWEEKLY eingestellt ist. enum: SUN | MON | TUE | WED | THU | FRI | SAT

Name	Typ	Beschreibung
targetCheckNames	list member: AuditCheckName	Gibt an, welche Prüfungen während des geplanten Audits ausgeführt werden. Die Prüfungen müssen für Ihr Konto aktiviert sein. (Verwenden Sie `DescribeAccountAuditConfiguration`, um die Liste aller Prüfungen, einschließlich aller aktivierten, anzuzeigen, oder `UpdateAccountAuditConfiguration`, um auszuwählen, welche Prüfungen aktiviert sind.)
scheduledAuditName	Zeichenfolge length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	Der Name des geplanten Audits. (Maximal 128 Zeichen)

Ausgabe

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

CLI-Ausgabefelder

Name	Typ	Beschreibung
scheduledAuditArn	Zeichenfolge	Der ARN des geplanten Audits.

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ResourceNotFoundException`

Die angegebene Ressource ist nicht vorhanden.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

DeleteScheduledAudit

Löscht ein geplantes Audit.

Syntax

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

`cli-input-json`-Format

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

`cli-input-json`-Felder

Name	Typ	Beschreibung
scheduledAuditName	Zeichenfolge length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	Der Name des geplanten Audits, das Sie löschen möchten.

Ausgabe

Keine

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ResourceNotFoundException`

Die angegebene Ressource ist nicht vorhanden.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

Ausführen einer On-Demand-Prüfung

Geben Sie mit `StartOnDemandAuditTask` die Prüfungen an, die Sie ausführen möchten, und starten Sie einen sofort ausgeführten Audit.

StartOnDemandAuditTask

Startet einen On-Demand-Device Defender-Audit.

Syntax

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

`cli-input-json`-Format

```
{
  "targetCheckNames": [
    "string"
```

```
  ]
}
```

cli-input-json-Felder

Name	Typ	Beschreibung
targetCheckNames	list member: AuditCheckName	Gibt an, welche Prüfungen während des Audits ausgeführt werden. Die Prüfungen, die Sie angeben, müssen für Ihr Konto aktiviert sein. Andernfalls tritt eine Ausnahme auf. Verwenden Sie `DescribeAccountAuditConfiguration`, um die Liste aller Prüfungen, einschließlich aller aktivierten, anzuzeigen, oder `UpdateAccountAuditConfiguration`, um auszuwählen, welche Prüfungen aktiviert sind.

Ausgabe

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

CLI-Ausgabefelder

Name	Typ	Beschreibung
taskId	Zeichenfolge length- max:40 min:1 pattern: [a-zA-Z0-9-]+	Die ID des On-Demand-Audits, den Sie gestartet haben.

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

`LimitExceededException`

Ein Grenzwert wurde überschritten.

Verwalten von Prüfungs-Instances

Mit `DescribeAuditTask` können Sie Informationen zu einer bestimmten Audit-Instance anzufordern. Wenn der Audit bereits ausgeführt wurde, umfassen die Ergebnisse, welche Prüfungen fehlgeschlagen sind und welche bestanden wurden und welche Prüfungen das System nicht durchführen konnte. Wenn der Audit noch nicht beendet wurde, zeigen die Ergebnisse auf, welche Prüfungen derzeit noch durchgeführt werden.

Mit `ListAuditTasks` können Sie die Prüfungen finden, die ausgeführt während eines angegebenen Zeitintervalls ausgeführt wurden.

Mit `CancelAuditTask` können Sie die derzeit laufende Prüfung unterbrechen.

DescribeAuditTask

Ruft Informationen zu einer Device Defender-Prüfung ab.

Syntax

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

`cli-input-json`-Format

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

cli-input-json-Felder

Name	Typ	Beschreibung
taskId	Zeichenfolge length- max:40 min:1 pattern: [a-zA-Z0-9-]+	Die ID des Audits, dessen Daten Sie abrufen möchten.

Ausgabe

```
{
  "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-Ausgabefelder

Name	Typ	Beschreibung
taskStatus	Zeichenfolge	Der Status der Prüfung: entweder IN_PROGRESS, COMPLETED, FAILED oder CANCELED. enum: IN_PROGRESS | COMPLETED | FAILED | CANCELED
taskType	Zeichenfolge	Der Typ von Audit: ON_DEMAND_AUDIT_TASK oder SCHEDULED_AUDIT_TASK. enum: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStartTime	Zeitstempel	Der Zeitpunkt, zu dem das Audit begann.
taskStatistics	TaskStatistics	Statistische Informationen zum Audit.
totalChecks	Ganzzahl	Die Anzahl der Prüfungen in diesem Audit.
inProgressChecks	Ganzzahl	Die Anzahl der laufenden Prüfungen.
waitingForDataCollectionChecks	Ganzzahl	Die Anzahl der Prüfungen mit anstehender Datensammlung.
compliantChecks	Ganzzahl	Die Anzahl der Prüfungen, mit denen konforme Ressourcen gefunden wurden.

Name	Typ	Beschreibung
nonCompliantChecks	Ganzzahl	Die Anzahl der Prüfungen , mit denen nicht-konforme Ressourcen gefunden wurden.
failedChecks	Ganzzahl	Die Anzahl der Prüfungen.
canceledChecks	Ganzzahl	Die Anzahl der Prüfungen, die nicht ausgeführt werden, da der Audit abgebrochen wurde.
scheduledAuditName	Zeichenfolge length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	Der Name des geplanten Audits (nur bei geplanten Audits).
auditDetails	map	Detaillierte Informationen zu den einzelnen Prüfungen, die während dieses Audits durchgeführt werden.
checkRunStatus	Zeichenfolge	Der Erledigungsstatus dieser Prüfung; entweder IN_PROGRESS, WAITING_FOR_DATA_COLLECTION, CANCELED, COMPLETED_COMPLIANT, COMPLETED_NON_COMPLIANT oder FAILED. enum: IN_PROGRESS | WAITING_FOR_DATA_COLLECTION | CANCELED | COMPLETED_COMPLIANT | COMPLETED_NON_COMPLIANT | FAILED

Name	Typ	Beschreibung
checkCompliant	boolesch	"True", wenn die Überprüfung abgeschlossen wurde und alle konformen Ressourcen gefunden hat.
totalResourcesCount	long	Die Anzahl der Ressourcen, für die die Prüfung ausgeführt wurde.
nonCompliantResourcesCount	long	Die Anzahl der Ressourcen, die von der Prüfungen als nicht-konforme Ressourcen befunden wurden.
errorCode	Zeichenfolge	Der Code eines jeden Fehlers, der beim Durchführen dieser Prüfung während dieses Audits aufgetreten ist. Entweder INSUFFICIENT_PERMISSIONS oder AUDIT_CHECK_DISABLED.
Nachricht	Zeichenfolge length- max:2048	Die Nachricht, die zu den Fehlern gehört, die beim Durchführen dieser Prüfung während dieses Audits aufgetreten sind.

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ResourceNotFoundException`

Die angegebene Ressource ist nicht vorhanden.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

ListAuditTasks

Listet die Device Defender-Audits auf, die während eines bestimmten Zeitraums durchgeführt wurden.

Syntax

```
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`-Format

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

`cli-input-json`-Felder

Name	Typ	Beschreibung
startTime	Zeitstempel	Der Beginn des Zeitraums . Audit-Daten werden für

Name	Typ	Beschreibung
		einen begrenzten Zeitraum (180 Tage) aufbewahrt. Wenn die angeforderte Startzeit vor dem Aufbewahrungszeitraum liegt, führt dies zu einer `InvalidRequestException`.
endTime	Zeitstempel	Das Ende des Zeitraums.
taskType	Zeichenfolge	Ein Filter, um die Auswertungsergebnisse auf die angegebene Art von Audit: entweder ON_DEMAND_AUDIT_TASK oder SCHEDULED__AUDIT_TASK. enum: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStatus	Zeichenfolge	Ein Filter, um die Ausgabe auf Audits mit dem angegebenen Erledigungsstatus einzuschränken: entweder IN_PROGRESS, COMPLETED, FAILED oder CANCELED. enum: IN_PROGRESS | COMPLETED | FAILED | CANCELED
nextToken	Zeichenfolge	Das Token für den nächsten Ergebnissatz.

Name	Typ	Beschreibung
maxResults	Ganzzahl range- max:250 min:1	Die maximale Anzahl der Ergebnisse, die auf einmal zurückgegeben werden sollen. Der Standardwert ist 25.

Ausgabe

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

CLI-Ausgabefelder

Name	Typ	Beschreibung
Aufgaben	list member: AuditTaskMetadata java class: java.util.List	Die Audits, die während des angegebenen Zeitraums durchgeführt wurden.
taskId	Zeichenfolge length- max:40 min:1 pattern: [a-zA-Z0-9-]+	Die ID dieses Audits.
taskStatus	Zeichenfolge	Der Status dieser Audits: entweder IN_PROGRESS, COMPLETED, FAILED oder CANCELED.

Name	Typ	Beschreibung
		enum: IN_PROGRESS | COMPLETED | FAILED | CANCELED
taskType	Zeichenfolge	Der Typ dieses Audits: entweder ON_DEMAND _AUDIT_TASK oder SCHEDULED_AUDIT_TASK. enum: ON_DEMAND _AUDIT_TASK | SCHEDULED _AUDIT_TASK
nextToken	Zeichenfolge	Ein Token, mit dem der nächste Ergebnissatzes abgerufen werden kann, bzw. `null`, wenn keine weiteren Ergebnisse vorliegen.

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

CancelAuditTask

Bricht einen Audit ab, der gerade ausgeführt wird. Es kann sich um einen planmäßigen oder um einen On-Demand-Audit handeln. Wenn der Audit nicht ausgeführt wird, tritt eine `InvalidRequestException` auf.

Syntax

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

`cli-input-json`-Format

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

`cli-input-json`-Felder

Name	Typ	Beschreibung
taskId	Zeichenfolge length- max:40 min:1 pattern: [a-zA-Z0-9-]+	Die ID des Audits, das Sie abbrechen möchten. Sie können nur einen Audit mit dem Status IN_PROGRESS abbrechen.

Ausgabe

Keine

Fehler

`ResourceNotFoundException`

Die angegebene Ressource ist nicht vorhanden.

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

Prüfen der Prüfungsergebnisse

Mit `ListAuditFindings` können Sie die Ergebnisse eines Audits anzeigen. Sie können die Ergebnisse nach der Art von Prüfung, einer bestimmten Ressource oder dem Zeitpunkt des Audits filtern. Sie können diese Informationen verwenden, um für gefundene Probleme Abhilfe zu schaffen.

Sie können Abhilfemaßnahmen definieren und sie auf die Ergebnisse Ihrer Prüfung anwenden. Weitere Informationen finden Sie unter Abschwächungsaktionen.

ListAuditFindings

Listet die Ergebnisse eines Device Defender-Audits oder der während eines bestimmten Zeitraums durchgeführten Audits. (Die Ergebnisse 180 Tage lang aufbewahrt.)

Syntax

```
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`-Format

```
{
  "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-Felder

Name	Typ	Beschreibung
taskId	Zeichenfolge length- max:40 min:1 pattern: [a-zA-Z0-9-]+	Ein Filter zum Einschränken der Ergebnisse auf das Audit mit der angegebenen ID. Sie müssen entweder die taskId oder den Wert für startTime und endTime angeben, aber nicht beides.
checkName	Zeichenfolge	Ein Filter zum Einschrän ken der Ergebnisse auf eine bestimmte Audit-Prüfung.
resourceIdentifier	ResourceIdentifier	Informationen zur Identifiz ierung der nicht konformen Ressource.
deviceCertificateId	Zeichenfolge length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	Die ID des Zertifikats, das der Ressource angehängt ist.

Name	Typ	Beschreibung
caCertificateId	Zeichenfolge length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	Die ID des CA-Zertifikats, mit dem das Zertifikat autorisiert wird.
cognitoIdentityPoolId	Zeichenfolge	Die ID des Amazon Cognito-Identitäten-Pools.
clientId	Zeichenfolge	Die Client-ID.
policyVersionIdentifier	PolicyVersionIdentifier	Die Version der Richtlinie, die der Ressource zugeordnet ist.
policyName	Zeichenfolge length- max:128 min:1 pattern: [w+=,.@-]+	Der Name der Richtlinie.
policyVersionId	Zeichenfolge pattern: [0-9]+	Die ID der Richtlinienversion, die der Ressource zugeordnet ist.
roleAliasArn	Zeichenfolge	Der ARN des Rollenalias, der Aktionen mit zu weitreichenden Berechtigungen hat. length- max:2048 min:1
Konto	Zeichenfolge length- max:12 min:12 pattern: [0-9]+	Das Konto, dem die Ressource zugeordnet ist.

Name	Typ	Beschreibung
maxResults	Ganzzahl range- max:250 min:1	Die maximale Anzahl der Ergebnisse, die auf einmal zurückgegeben werden sollen. Der Standardwert ist 25.
nextToken	Zeichenfolge	Das Token für den nächsten Ergebnissatz.
startTime	Zeitstempel	Ein Filter, mit dem die Ergebnisse auf jene einzuschränken, die nach dem angegebenen Zeitpunkt gefunden wurden. Sie müssen entweder die den Wert für startTime und endTime oder die taskID angeben, aber nicht beides.
endTime	Zeitstempel	Ein Filter, mit dem die Ergebnisse auf diejenigen eingeschränkt werden, die vor dem angegebenen Zeitpunkt gefunden wurden. Sie müssen entweder die den Wert für startTime und endTime oder die taskID angeben, aber nicht beides.

Ausgabe

```
{
  "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-Ausgabefelder

Name	Typ	Beschreibung
findings	list member: AuditFinding	Die Ergebnisse des Audits.
taskId	Zeichenfolge length- max:40 min:1 pattern: [a-zA-Z0-9-]+	Die ID des Audits, der dieses Ergebnis generiert hat.
checkName	Zeichenfolge	Die Audit-Prüfung, die dieses Ergebnis generiert hat.
taskStartTime	Zeitstempel	Der Zeitpunkt, zu dem das Audit begann.
findingTime	Zeitstempel	Die Uhrzeit, zu der das Ergebnis entdeckt wurde.
severity	Zeichenfolge	Der Schweregrad des Ergebnisses. enum: CRITICAL | HIGH | MEDIUM | LOW
nonCompliantResource	NonCompliantResource	Die Ressource, die als nicht mit der Auditprüfung konform befunden wurde.
resourceType	Zeichenfolge	Der Typ der nichtkonformen Ressource.

Name	Typ	Beschreibung
		enum: DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	Informationen zur Identifizierung der nicht konformen Ressource.
deviceCertificateId	Zeichenfolge length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	Die ID des Zertifikats, das der Ressource angehängt ist.
caCertificateId	Zeichenfolge length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	Die ID des CA-Zertifikats, mit dem das Zertifikat autorisiert wird.
cognitoIdentityPoolId	Zeichenfolge	Die ID des Amazon Cognito-Identitäten-Pools.
clientId	Zeichenfolge	Die Client-ID.
policyVersionIdentifier	PolicyVersionIdentifier	Die Version der Richtlinie, die der Ressource zugeordnet ist.
policyName	Zeichenfolge length- max:128 min:1 pattern: [w+=,.@-]+	Der Name der Richtlinie.

Name	Typ	Beschreibung
policyVersionId	Zeichenfolge pattern: [0-9]+	Die ID der Richtlinienversion, die der Ressource zugeordnet ist.
Konto	Zeichenfolge length- max:12 min:12 pattern: [0-9]+	Das Konto, dem die Ressource zugeordnet ist.
additionalInfo	map	Weitere Informationen über die nicht konforme Ressource.
relatedResources	list member: RelatedResource	Die Liste zugehöriger Ressourcen.
resourceType	Zeichenfolge	Der Typ der Ressource. enum: DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	Informationen zum Identifizieren der Ressource.
deviceCertificateId	Zeichenfolge length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	Die ID des Zertifikats, das der Ressource angehängt ist.

Name	Typ	Beschreibung
caCertificateId	Zeichenfolge length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	Die ID des CA-Zertifikats, mit dem das Zertifikat autorisiert wird.
cognitoIdentityPoolId	Zeichenfolge	Die ID des Amazon Cognito-Identitäten-Pools.
clientId	Zeichenfolge	Die Client-ID.
policyVersionIdentifier	PolicyVersionIdentifier	Die Version der Richtlinie, die der Ressource zugeordnet ist.
iamRoleArn	Zeichenfolge Länge max. 2048, min. 20	Der ARN der IAM-Rolle, die Aktionen mit zu weitreichenden Berechtigungen hat.
policyName	Zeichenfolge length- max:128 min:1 pattern: [w+=,.@-]+	Der Name der Richtlinie.
policyVersionId	Zeichenfolge pattern: [0-9]+	Die ID der Richtlinienversion, die der Ressource zugeordnet ist.
roleAliasArn	Zeichenfolge length- max:2048 min:1	Der ARN des Rollenalias, der Aktionen mit zu weitreichenden Berechtigungen hat.
Konto	Zeichenfolge length- max:12 min:12 pattern: [0-9]+	Das Konto, dem die Ressource zugeordnet ist.

Name	Typ	Beschreibung
additionalInfo	map	Weitere Informationen über die Ressource.
reasonForNonCompliance	Zeichenfolge	Der Grund, weshalb die Ressource nicht konform war.
reasonForNonCompliance Code	Zeichenfolge	Ein Code, womit der Grund angegeben wird, weshalb die Ressource nicht konform war.
nextToken	Zeichenfolge	Ein Token, mit dem der nächste Ergebnissatzes abgerufen werden kann, bzw. `null`, wenn keine weiteren Ergebnisse vorliegen.

Fehler

`InvalidRequestException`

Der Inhalt der Anforderung war ungültig.

`ThrottlingException`

Die Rate überschreitet den Grenzwert.

`InternalFailureException`

Ein unerwarteter Fehler ist aufgetreten.

Unterdrückungen von Prüfergebnissen

Wenn Sie eine Prüfung durchführen, werden Ergebnisse für alle nicht konformen Ressourcen gemeldet. Das bedeutet, dass Ihre Prüfungsberichte Ergebnisse für Ressourcen enthalten, für die Sie Probleme beheben möchten, sowie für Ressourcen, die bekanntermaßen nicht konform sind, wie z. B. Testgeräte oder defekte Geräte. Im Rahmen der Prüfung werden weiterhin Ergebnisse für Ressourcen gemeldet, die in aufeinanderfolgenden Prüfungsläufen weiterhin nicht konform sind, wodurch Ihre Berichte möglicherweise um unerwünschte Informationen erweitert werden. Mithilfe der

Unterdrückungen von Prüfungsergebnissen können Sie Ergebnisse für einen bestimmten Zeitraum unterdrücken oder herausfiltern, bis die Ressource instandgesetzt ist, oder auf unbestimmte Zeit, wenn es sich um eine Ressource handelt, die einem Test oder einem defekten Gerät zugeordnet ist.

> **Note**
>
> Bei unterdrückten Prüfungsergebnissen sind keine Abschwächungsaktionen verfügbar. Weitere Informationen zu Abschwächungsaktionen finden Sie unter Abschwächungsaktionen.

Informationen zu Kontingenten für die Unterdrückung von Prüfungsergebnissen finden Sie unter AWS IoTDevice Defender-Endpunkte und Kontingente.

So funktionieren Unterdrückungen von Prüfergebnissen

Wenn Sie für eine nicht konforme Ressource eine Unterdrückung von Prüfungsergebnissen erstellen, verhalten sich Ihre Prüfberichte und Benachrichtigungen anders.

Ihre Prüfberichte werden einen neuen Abschnitt enthalten, in dem alle unterdrückten Feststellungen im Zusammenhang mit dem Bericht aufgeführt sind. Unterdrückte Ergebnisse werden nicht berücksichtigt, wenn wir beurteilen, ob eine Auditprüfung konform ist oder nicht. Darüber hinaus wird eine unterdrückte Ressourcenanzahl auch für jede Auditprüfung zurückgegeben, wenn Sie den Befehl describe-audit-task in der Befehlszeilenschnittstelle (CLI) verwenden.

Für Prüfungsbenachrichtigungen werden keine unterdrückten Ergebnisse berücksichtigt, wenn wir beurteilen, ob eine Auditprüfung konform ist oder nicht. Eine unterdrückte Ressourcenanzahl ist auch in jeder Prüfergebnisbenachrichtigung enthalten, die AWS IoT Device Defender in Amazon CloudWatch und Amazon Simple Notification Service (Amazon SNS) veröffentlicht.

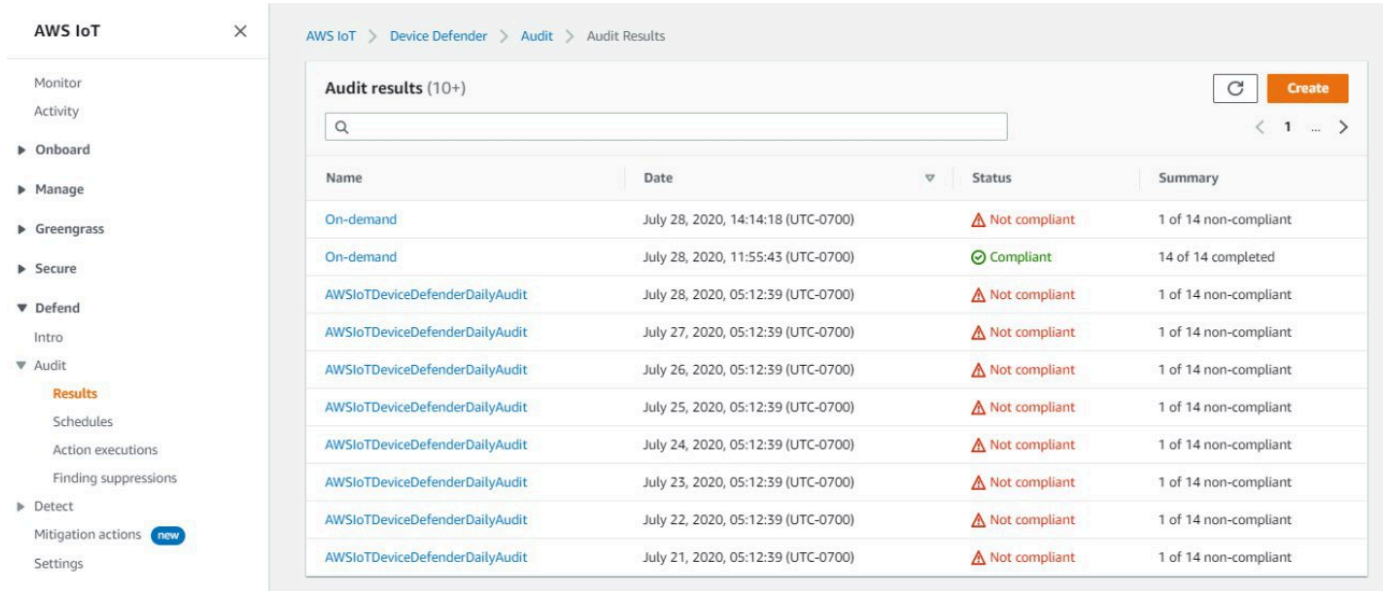
So verwenden Sie die Unterdrückung von Prüfungsergebnissen auf der Konsole

So unterdrücken Sie ein Ergebnis in einem Prüfungsbericht

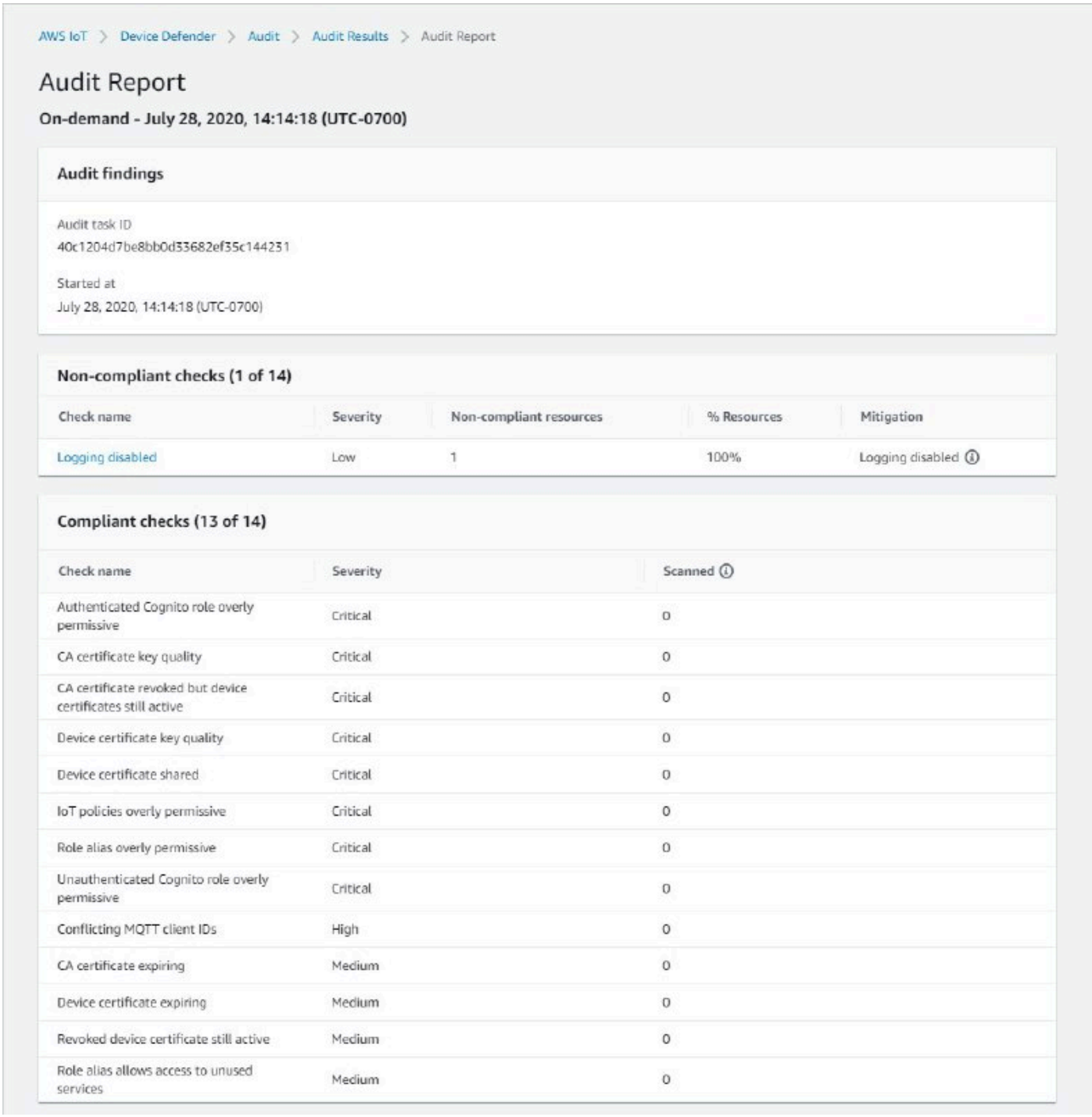
Im folgenden Verfahren wird gezeigt, wie Sie eine Unterdrückung von Prüfungsergebnissen auf der AWS IoT-Konsole erstellen.

1. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoT-Konsole, und wählen Sie dann Prüfung, Ergebnisse.

2. Wählen Sie einen Prüfbericht aus, den Sie überprüfen möchten.

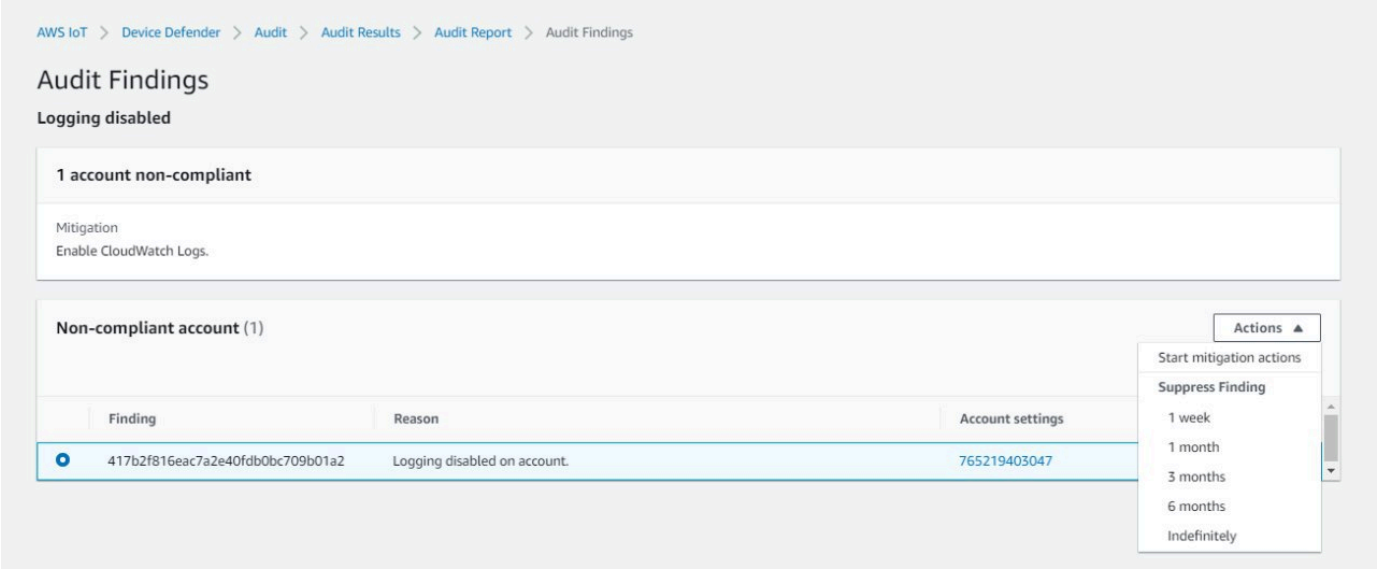


3. Wählen Sie im Abschnitt Prüfungen mit Compliance-Abweichungen unter Prüfungsname,die Prüfung, an der Sie interessiert sind.

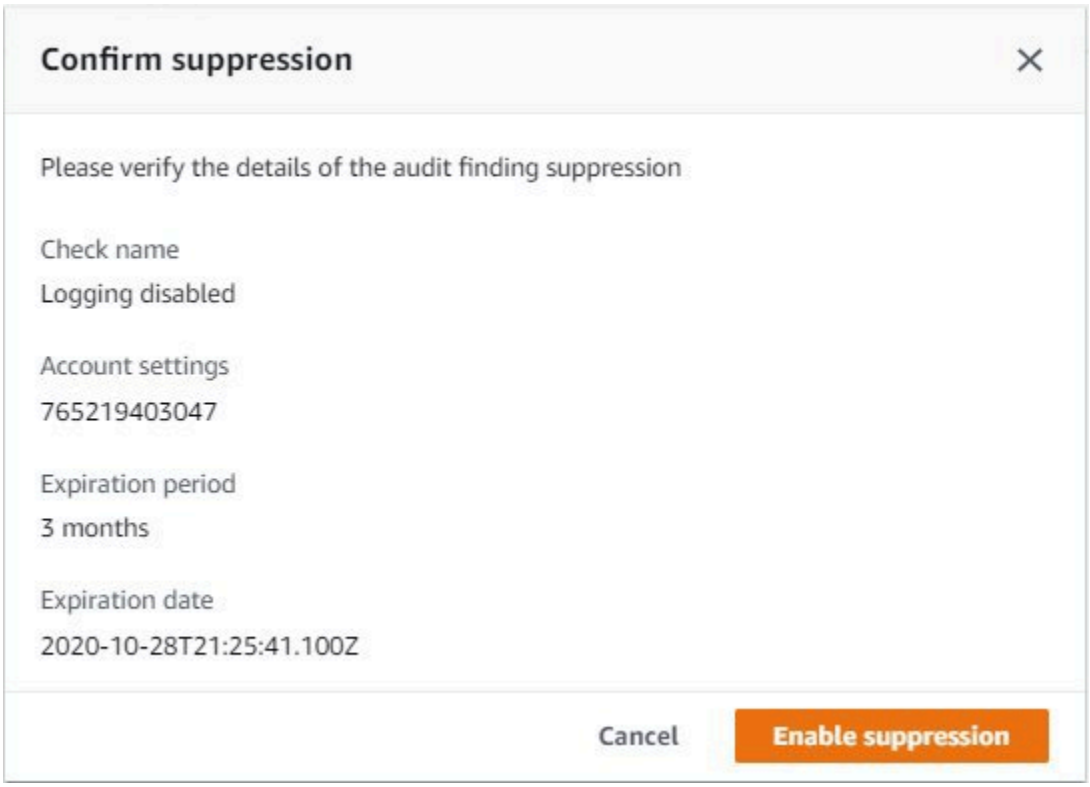


4. Wenn es Ergebnisse gibt, die Sie nicht sehen möchten, klicken Sie auf dem Auditprüfungs-Detailbildschirm auf das Optionsfeld neben dem Ergebnis. Wählen Sie als Nächstes Aktionen, und legen Sie dann fest, wie lange die Unterdrückung Ihrer Prüfungsergebnisse beibehalten werden soll.

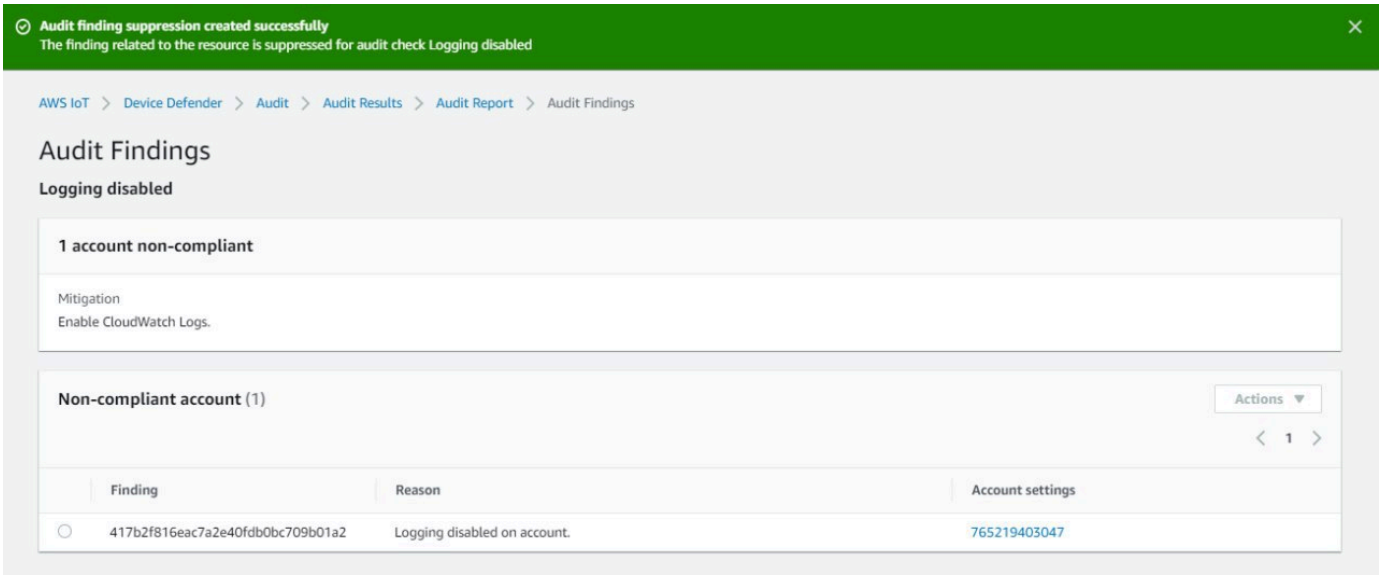
> **Note**
>
> Auf der Konsole können Sie 1 Woche, 1 Monat, 3 Monate, 6 Monate oder Unbegrenzt als Ablaufdaten für die Löschung Ihrer Prüfungsergebnisse auswählen. Wenn Sie ein bestimmtes Ablaufdatum festlegen möchten, können Sie dies nur in der CLI oder API tun. Unterdrückungen von Prüfungsergebnissen können zudem unabhängig vom Ablaufdatum jederzeit storniert werden.



5. Bestätigen Sie die Unterdrückungsdetails, und wählen Sie dann Unterdrückung aktivieren.



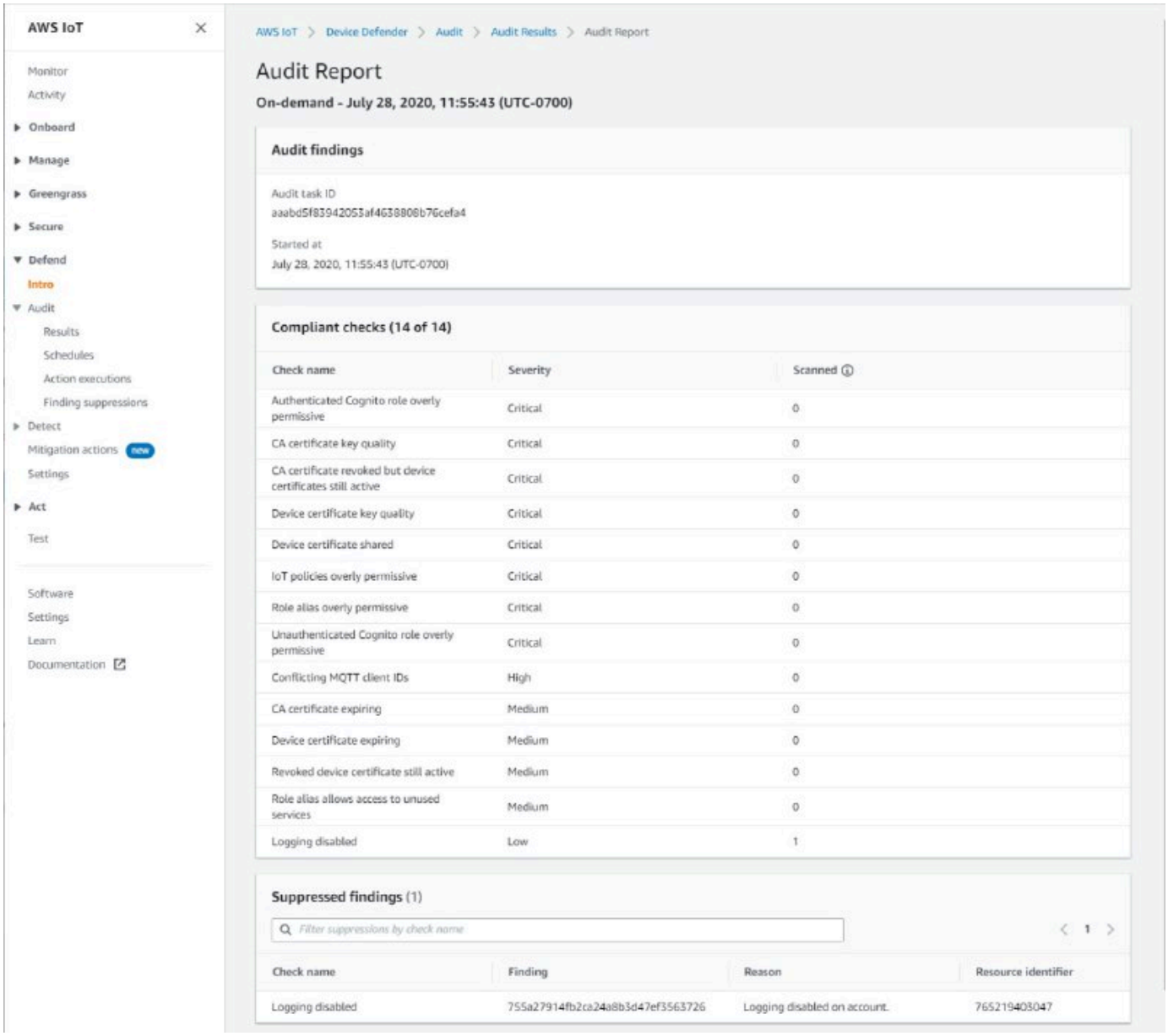
6. Nachdem Sie die Unterdrückung von Prüfungsergebnissen erstellt haben, wird ein Banner angezeigt, das bestätigt, dass die Unterdrückung Ihrer Prüfungsergebnisse erstellt wurde.



So zeigen Sie Ihre unterdrückten Ergebnisse in einem Prüfungsbericht an

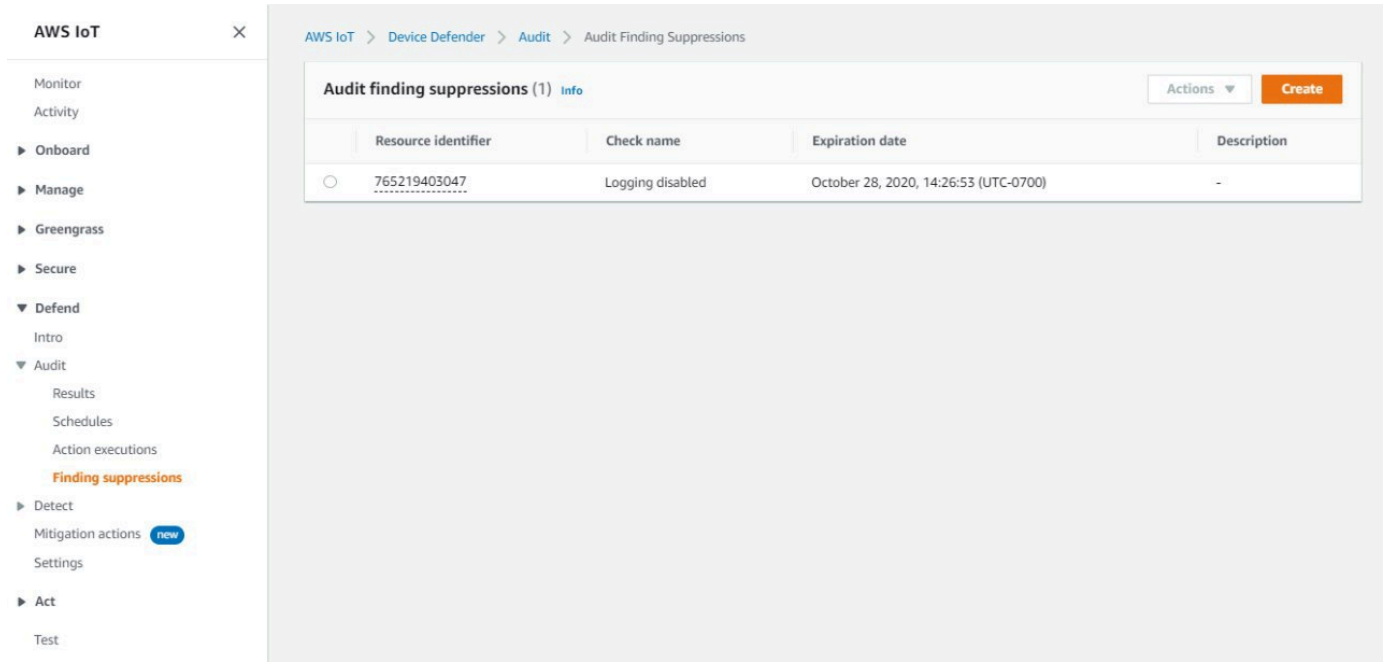
1. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoTKonsole, und wählen Sie dann Prüfung, Ergebnisse.

2. Wählen Sie einen Prüfbericht aus, den Sie überprüfen möchten.
3. Sehen Sie sich im Abschnitt Unterdrückte Ergebnisse an, welche Prüfungsergebnisse für den von Ihnen ausgewählten Prüfungsbericht unterdrückt wurden.



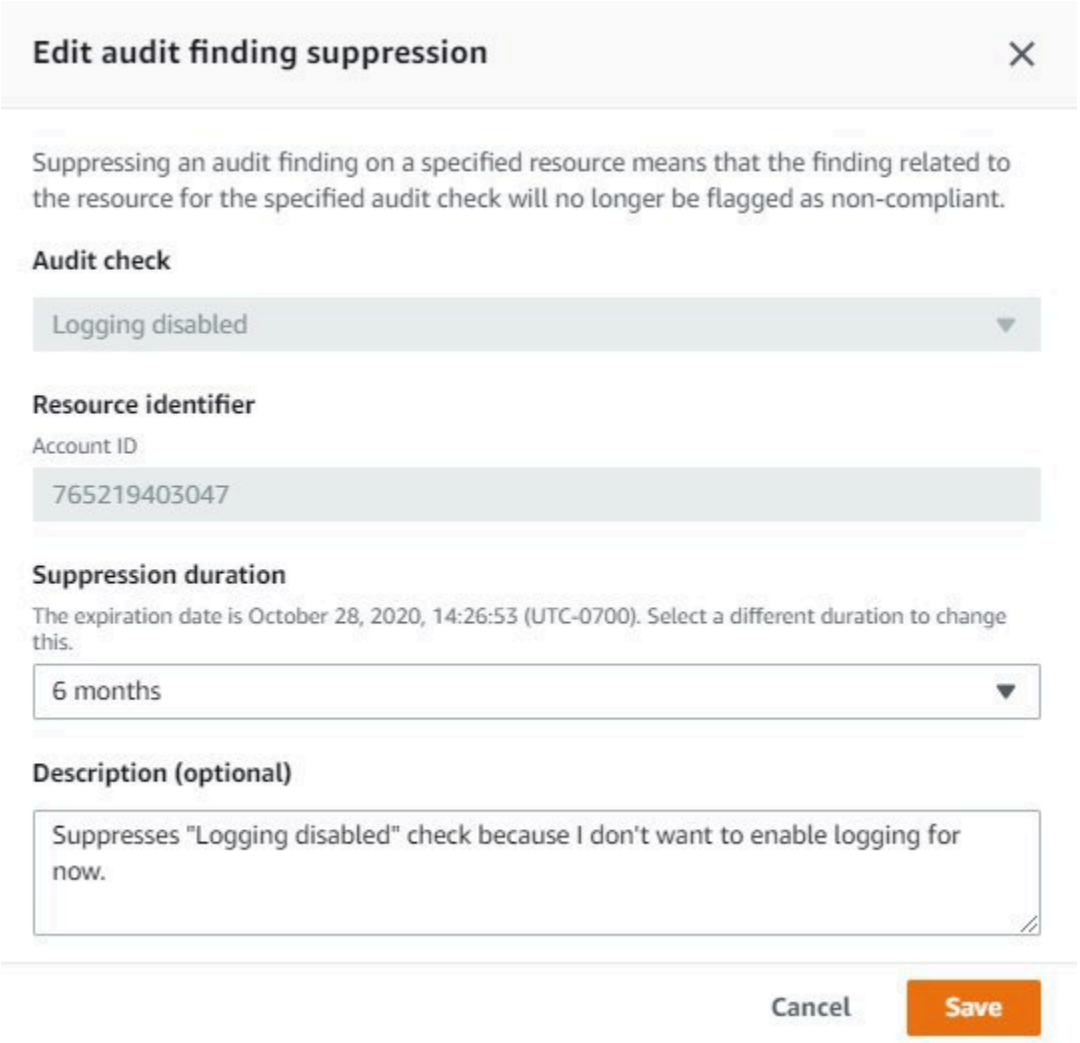
So listen Sie die Unterdrückungen Ihrer Prüfungsergebnisse auf

- Erweitern Sie Verteidigen im Navigationsbereich der AWS IoTKonsole, und wählen Sie dann Prüfung, Unterdrückungen suchen.



So bearbeiten Sie die Unterdrückung Ihrer Prüfungsergebnisse

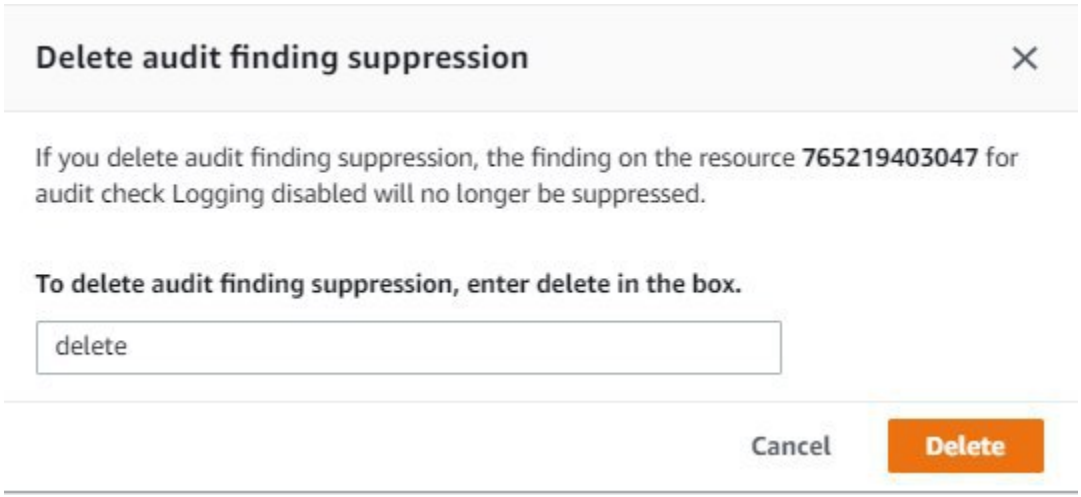
1. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoTKonsole, und wählen Sie dann Prüfung, Unterdrückungen von Ergebnissen.
2. Wählen Sie das Optionsfeld neben der Überprüfung von Prüfungsergebnissen aus, die bearbeiten möchten. Wählen Sie dann Aktionen, Bearbeiten.
3. Im Fenster Unterdrückung von Prüfungsergebnissen bearbeiten können Sie die Dauer der Unterdrückung oder die Beschreibung (optional) ändern.



4. Nachdem Sie Ihre Änderungen durchgeführt haben, wählen Sie Speichern. Das Fenster Unterdrückungen von Ergebnissen wird geöffnet.

So löschen Sie eine Unterdrückung von Prüfungsergebnissen

1. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoTKonsole, und wählen Sie dann Prüfung, Unterdrückungen von Ergebnissen.
2. Wählen Sie das Optionsfeld neben der Überprüfung von Prüfungsergebnissen aus, die Sie löschen möchten, und wählen Sie dann Aktionen, Löschen.
3. Geben Sie im Fenster Unterdrückung von Prüfungsergebnissen löschen `delete` in das Textfeld ein, um das Löschen zu bestätigen, und wählen Sie dann Löschen. Das Fenster Unterdrückungen suchen wird geöffnet.



So verwenden Sie die Unterdrückung von Prüfungsergebnissen in der CLI

Sie können die folgenden CLI-Befehle zum Erstellen und Verwalten von Unterdrückungen von Prüfungsergebnissen verwenden.

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

Ihre Eingabe von `resource-identifier` ist vom `check-name` abhängig, für den Sie Ergebnisse unterdrücken. In der folgenden Tabelle wird detailliert beschrieben, welche Prüfungen `resource-identifier` für die Erstellung und Bearbeitung von Unterdrückungen erfordern.

> **Note**
>
> Die Befehle zur Unterdrückung deuten nicht darauf hin, dass eine Prüfung deaktiviert ist. Auf Ihren AWS IoT-Geräten werden weiterhin Prüfungen ausgeführt. Unterdrückungen gelten nur für die Prüfungsergebnisse.

check-name	resource-identifier
AUTHENTICATE_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	cognitoIdentityPoolId
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

Erstellen und Anwenden einer Unterdrückung von Prüfungsergebnissen

Im folgenden Verfahren wird gezeigt, wie Sie eine Unterdrückung von Prüfergebnissen in der AWS CLI erstellen.

- Verwenden Sie den `create-audit-suppression`-Befehl, um eine Unterdrückung von Prüfungsergebnissen zu erstellen. Im folgenden Beispiel wird eine Unterdrückung der Prüfungsergebnisse für AWS-Konto *123456789012* auf der Grundlage der Prüfung Protokollierung deaktiviert erstellt.

  ```
  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."
  ```

  Für diesen Befehl gibt es keine Ausgabe.

APIs zur Suche nach Unterdrückungen von Prüfungsergebnissen

Die folgenden APIs können verwendet werden, um Unterdrückungen bei Prüfungsergebnissen zu erstellen und zu verwalten.

- CreateAuditSuppression
- DescribeAuditSuppression
- DescribeAuditSuppression
- DeleteAuditSuppression
- ListAuditSuppressions

Mit der ListAuditFindings-API können Sie nach bestimmten Prüfungsergebnissen filtern.

Detect

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

AWS IoT Device Defender Detect ermöglicht es Ihnen, durch Überwachung des Verhaltens Ihrer Geräte ungewöhnliches Verhalten zu erkennen, das auf ein kompromittiertes Gerät hindeuten kann. Mithilfe einer Kombination von cloudseitigen Metriken (von AWS IoT) und geräteseitigen Metriken (von Agenten, die Sie auf Ihren Geräten installieren) können Sie Folgendes erkennen:

- Änderungen in Verbindungsmustern.
- Geräte, die mit nicht autorisierten oder nicht erkannten Endpunkten kommunizieren.
- Änderungen der Muster des eingehenden und ausgehenden Datenverkehrs der Geräte.

Sie erstellen Sicherheitsprofile mit Definitionen der erwarteten Geräte-Verhaltensweisen und weisen sie einer Gruppe von Geräten oder allen Geräten in Ihrer Flotte zu. AWS IoT Device Defender Detect verwendet diese Sicherheitsprofile zum Erkennen von Anomalien und Senden von Alarmen über Amazon CloudWatch-Metriken und Amazon Simple Notification Service-Benachrichtigungen.

AWS IoT Device Defender Detect kann häufige Sicherheitsprobleme von verbundenen Geräten erkennen:

- Datenverkehr von einem Gerät zu einer bekannten böswilligen IP-Adresse oder zu einem nicht autorisierten Endpunkt, der auf einen potenziell böswilligen Befehl und Kontrollkanal hindeutet.
- Anomaler Datenverkehr, wie beispielsweise Verkehrsspitzen im ausgehenden Datenverkehr, der darauf hindeutet, dass ein Gerät an einem DDoS-Angriff beteiligt ist.
- Geräte mit Remote-Management-Schnittstellen und Ports, auf die remote zugegriffen werden kann.
- Ein Spitzenwert der Anzahl der an Ihr Konto gesendeten Nachrichten (z. B. von einem nicht autorisierten Gerät, was dazu führen kann, dass übermäßige Gebühren pro Nachricht anfallen).

Anwendungsfälle:

Messen der Angriffsfläche

Sie können mit AWS IoT Device Defender Detect die Angriffsfläche Ihrer Geräte messen. Sie können beispielsweise Geräte mit Service-Ports identifizieren, die häufig das Ziel von Angriffskampagnen sind (Ausführung von Telnet-Service auf Ports 23/2323, SSH-Service auf Port 22, HTTP/S-Services auf Ports 80/443/8080/8081). Obwohl der Einsatz dieser Service-Ports auf den Geräten gute Gründe haben kann, sind sie in der Regel aber auch Teil der Angriffsfläche für Angreifer und bergen die damit verbundenen Risiken. Nachdem AWS IoT Device Defender Detect Sie auf die Angriffsfläche aufmerksam gemacht hat, können Sie diese minimieren (indem Sie nicht verwendete Netzwerkdienste eliminieren) oder zusätzliche Bewertungen zur Identifizierung von Sicherheitslücken (z. B. Telnet-Konfiguration mit gemeinsamen, Standard- oder schwachen Passwörtern) durchführen.

Erkennen von Anomalien im Geräteverhalten mit möglichen die Sicherheit betreffenden Ursachen

Sie können sich von AWS IoT Device Defender Detect auf unerwartete Geräte-Verhaltensmetriken (die Anzahl der offenen Ports, die Anzahl der Verbindungen, ein unerwarteter offener Port, Verbindungen mit unerwarteten IP-Adressen) aufmerksam machen lassen, die auf eine Sicherheitsverletzung hindeuten können. Beispiel: eine höhere Anzahl von TCP-Verbindungen als erwartet könnte darauf hinweisen, dass ein Gerät für einen DDoS-Angriff verwendet wird. Ein Prozess, der einen anderen als den von Ihnen erwarteten Port überwacht, könnte auf eine auf dem Gerät für Remote-Steuerung installierte Backdoor hinweisen. Sie können mit AWS IoT Device Defender Detect den Zustand Ihrer Geräteflotte prüfen und Ihre Sicherheitsannahmen (z. B. kein Gerät überwacht Port 23 oder 2323) überprüfen.

Sie können die Machine Learning (ML)-basierte Bedrohungserkennung aktivieren, um potenzielle Bedrohungen automatisch zu identifizieren.

Erkennen eines nicht korrekt konfigurierten Geräts

Ein starker Anstieg in der Anzahl oder Größe der von einem Gerät zu Ihrem Konto gesendeten Nachrichten kann auf ein falsch konfiguriertes Gerät hinweisen. Bei Verwendung eines solchen Geräts könnten sich Ihre Gebühren pro Nachricht erhöhen. Ebenso könnte ein Gerät mit vielen Autorisierungsfehlern eine neu konfigurierte Richtlinie benötigen.

Überwachung der Verhaltensweise nicht registrierter Geräte

Mit AWS IoT Device Defender Detect können Sie ungewöhnliche Verhaltensweisen für Geräte identifizieren, die nicht in der AWS IoT-Registrierung registriert sind. Sie können Sicherheitsprofile speziell für einen der folgenden Zieltypen definieren:

- Alle Geräte
- Alle registrierten Geräte (Objekte in der AWS IoT-Registrierung)
- Alle nicht registrierten Geräte
- Geräte in einer Objektgruppe

Ein Sicherheitsprofil definiert eine Reihe von erwarteten Verhaltensweisen für Geräte in Ihrem Konto und legt die auszuführenden Aktionen fest, wenn eine Anomalie erkannt wird. Sicherheitsprofile sollten den spezifischsten Zielen angefügt werden, damit Sie präzise steuern können, welche Geräte anhand dieses Profils ausgewertet werden.

Nicht registrierte Geräte müssen eine konsistente MQTT-Client-ID oder einen konsistenten Objektnamen (für Geräte, die Gerätemetriken ausgeben) innerhalb der gesamten Lebensdauer des Geräts angeben, sodass alle Verstöße und Metriken dem demselben Gerät zugeordnet werden.

> **Important**
>
> Gerätemeldungen werden abgelehnt, wenn der Objektname Steuerzeichen enthält oder wenn der Objektname länger als 128 Byte an UTF-8-codierten Zeichen ist.

Änderungen an der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender

Die Erkennungsfunktion von AWS IoT Device Defender wird für Neukunden ab dem 31. August 2026 nicht mehr verfügbar sein. Bestandskunden von AWS IoT Device Defender Detect können die Funktion weiterhin verwenden.

Wenn Sie die Erkennungsfunktion von AWS IoT Device Defender derzeit aktiv nutzen, können Sie die regelbasierte Erkennung und ML Detect weiterhin wie gewohnt verwenden. Die Erkennungsfunktion wird ab dem 31. August 2026 in den Wartungsmodus versetzt. Das bedeutet, dass Detect keine neuen Funktionen mehr hinzugefügt werden und die Erkennungsfunktion für Neukunden nicht mehr

verfügbar sein wird. AWS wird weiterhin Sicherheitsupdates bereitstellen und die Verfügbarkeit der Funktion aufrechterhalten, damit Ihre Überwachungs-Workflows weiterhin reibungslos ausgeführt werden können.

> **Note**
>
> Die Audit-Funktion von AWS IoT Device Defender bleibt weiterhin uneingeschränkt verfügbar. Nur die Erkennungsfunktion (die regelbasierte Erkennung und ML Detect) wechselt in den Wartungsmodus und wird ab dem 31. August 2026 für Neukunden nicht mehr verfügbar sein.

Alternativen zu AWS IoT Device Defender Detect

Selbstverwaltete Alternative für die cloudseitige Überwachung von Metriken

Ausführliche Anweisungen und Beispielcode finden Sie im GitHub-Projekt Erkennung von IoT-Geräteanomalien. Diese Open-Source- und Serverless-Pipeline ermöglicht es Ihnen, vergleichbare Funktionen wie die Erkennungsfunktion von AWS IoT Device Defender für die cloudseitige Überwachung von Metriken in Ihrem eigenen AWS-Konto zu nutzen. Sie basiert auf Serverless-AWS-Services, darunter AWS IoT, Amazon Managed Service für Apache Flink, AWS Lambda, Amazon Kinesis, Amazon SageMaker AI, Amazon DynamoDB, Amazon API Gateway, Amazon Simple Notification Service und Amazon CloudWatch. Das Projekt umfasst eine Vorlage für das AWS Serverless Application Model (SAM), mit der Sie die Pipeline innerhalb weniger Minuten in Ihrem Konto bereitstellen können, eine REST-API für die Verwaltung von Sicherheitsprofilen sowie Unterstützung für regelbasierte Schwellenwerte und die Erkennung von Anomalien auf Basis von Machine Learning (ML). Sie können das Intervall des gleitenden Fensters für Metriken anpassen, um zwischen Erkennungslatenz und Kosten abzuwägen.

Hauptunterschiede zu AWS IoT Device Defender Detect

- Der Beispielcode ist eine Open-Source-Software-Pipeline, die Sie in Ihrem eigenen AWS-Konto bereitstellen und betreiben. Es handelt sich nicht um einen von AWS verwalteten Dienst.
- Cloudseitige Metriken werden sofort unterstützt. Für geräteseitige Metriken und benutzerdefinierte Metriken muss die Pipeline wie in der README-Datei des Projekts beschrieben erweitert werden.
- Die Konfiguration erfolgt über eine REST-API und umfasst keine Integration in die AWS-Managementkonsole.
- Abschwächungsaktionen (beispielsweise die Einordnung eines Geräts in eine Gerätegruppe oder der Austausch eines Zertifikats) sind nicht enthalten. Sie können dies jedoch implementieren,

indem Sie eine Lambda-Funktion für das Amazon SNS-Benachrichtigungsthema des Projekts abonnieren.

- Verbesserungen gegenüber der Erkennungsfunktion von AWS IoT Device Defender:
  - Anpassbar – Die vollständig anpassbare Pipeline ermöglicht es Ihnen, Dimensionen hinzuzufügen, Aggregationsfenster anzupassen oder die Erkennungslogik zu erweitern, ohne dass dabei Einschränkungen hinsichtlich der Services auftreten.
  - Kostenkontrolle – Der Beispielcode bietet Möglichkeiten zur Kostenoptimierung auf Grundlage der gewünschten Metriken. Die Pipeline ermöglicht Ihnen die Nutzung vorhandener Regeln und unterstützt Nachrichten, die in AWS IoT Rules Basic Ingest Topics oder an den AWS IoT Message Broker veröffentlicht werden. Sie können außerdem das Intervall des gleitenden Fensters für die Auswertung von Metriken steuern, um zwischen höherer Latenz und geringeren Kosten abzuwägen. Bei ML-Metriken ist die Pipeline im großen Maßstab zudem kosteneffizienter als ML Detect, da sie auf einer festen Infrastruktur ausgeführt wird und keine Kosten pro Datenpunkt anfallen.
  - Vereinfachte Überwachung – Der Beispielcode ersetzt CloudWatch-Metriken pro Sicherheitsprofil durch eine einzige aggregierte Metrik für die gesamte Flotte, um die Überwachung zu vereinfachen und gleichzeitig die Verhaltenserkennung auf Geräteebene aufrechtzuerhalten.
  - Verbesserte Granularität von ML-Modellen – Im Gegensatz zu ML Detect, bei dem ein einzelnes ML-Modell für alle Verhaltensweisen innerhalb eines Sicherheitsprofils trainiert wird, trainiert der Beispielcode für jedes Verhalten ein separates Modell. Dadurch erhält jedes Modell eine gezieltere Ausgangsbasis für die Verteilung einer einzelnen Metrik, und es wird sichergestellt, dass etwaige Fehler im Modell hinsichtlich eines bestimmten Verhaltens die Erkennungsgenauigkeit bei anderen Verhaltensweisen nicht beeinträchtigen.

Open-Source-Geräteagenten für geräteseitige Metriken

Um geräteseitige Metriken (wie beispielsweise TCP- und UDP-Ports im Listening-Modus, bestehende TCP-Verbindungen, Ziel-IP-Adressen sowie Paket- oder Byte-Zähler) zu erfassen, können Sie einen Open-Source-Agenten auf Ihren Geräten ausführen und die Metriken über MQTT an AWS IoT übermitteln. Zu den Optionen gehören der AWS IoT Device Client, Telegraf (unter Verwendung seines MQTT-Ausgabe-Plugins) oder osquery in Kombination mit einem Eclipse-Paho-MQTT-Client. Die README-Datei des GitHub-Projekts enthält zudem Anweisungen dazu, wie die von den Geräten veröffentlichten Metriken über AWS IoT-Regeln erfasst und die im Beispielcode enthaltenen Sicherheitsprofilschemas erweitert werden können.

Häufig gestellte Fragen

Was bedeutet das für die Erkennungsfunktion von AWS IoT Device Defender?

Die Erkennungsfunktion von AWS IoT Device Defender wird für Neukunden ab dem 31. August 2026 nicht mehr verfügbar sein. Die Funktion wird für Bestandskunden weiterhin verfügbar sein, es werden jedoch keine weiteren Funktionen hinzugefügt.

Wie wirkt sich das auf Bestandskunden aus?

Bei Bestandskunden wird es zu keinerlei Beeinträchtigungen ihrer Workloads kommen. Sie können die Erkennungsfunktion von AWS IoT Device Defender weiterhin wie gewohnt verwenden und es werden weiterhin Sicherheitsupdates bereitgestellt, um die Zuverlässigkeit der Funktionen zu gewährleisten. Es ergeben sich keine Auswirkungen auf Ihren Sicherheitsstatus oder Ihre Workflows.

Wann muss ich zur selbstverwalteten Alternative migrieren?

Die Erkennungsfunktion von AWS IoT Device Defender steht Bestandskunden im Wartungsmodus weiterhin zur Verfügung. Falls Änderungen an der Verfügbarkeit der Funktion geplant sind, werden Sie vorab darüber informiert. Wenn Sie an einer besser anpassbaren Lösung interessiert sind, informieren Sie sich über die selbstverwaltete Alternative.

Ist AWS IoT Device Defender Audit betroffen?

Nein. Nur die Erkennungsfunktion wechselt in den Wartungsmodus. Die Audit-Funktion von AWS IoT Device Defender bleibt weiterhin uneingeschränkt verfügbar.

Kann ich schrittweise migrieren?

Ja. Sie können den Beispielcode in Ihrem Konto bereitstellen und zusammen mit AWS IoT Device Defender Detect ausführen. Beide Lösungen werten dieselben AWS IoT-Daten aus, sodass Sie die Ergebnisse vergleichen und Sicherheitsprofile schrittweise migrieren können.

Wie kann ich Hilfe erhalten, wenn Probleme auftreten?

Wenn bei Ihnen Probleme auftreten, wenden Sie sich an den AWS Support.

Anwendungsfälle für Sicherheit

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

In diesem Abschnitt werden die verschiedenen Arten von Angriffen beschrieben, die Ihre Geräteflotte bedrohen, sowie die empfohlenen Metriken, mit denen Sie diese Angriffe überwachen können. Wir empfehlen, metrische Anomalien als Ausgangspunkt für die Untersuchung von Sicherheitsproblemen zu verwenden. Sie sollten Ihre Einschätzung von Sicherheitsbedrohungen jedoch nicht ausschließlich auf eine metrische Anomalie stützen.

Um einen Anomaliealarm zu untersuchen, korrelieren Sie die Alarmdetails mit anderen Kontextinformationen wie Geräteattributen, historischen Trends bei Gerätemetriken, historischen Trends der Sicherheitsprofilmetrik, benutzerdefinierten Metriken und Protokollen, um festzustellen, ob eine Sicherheitsbedrohung vorliegt.

Anwendungsfälle auf der Cloud-Seite

Device Defender kann die folgenden Anwendungsfälle auf der AWS IoT-Cloud-Seite überwachen.

Diebstahl geistigen Eigentums:

Beim Diebstahl geistigen Eigentums wird das geistige Eigentum einer Person oder eines Unternehmens gestohlen, einschließlich Geschäftsgeheimnissen, Hardware oder Software. Er tritt häufig während der Herstellungsphase von Geräten auf. Der Diebstahl geistigen Eigentums kann in Form von Piraterie, Gerätediebstahl oder Diebstahl von Gerätezertifikaten erfolgen. Cloud-basierter Diebstahl von geistigem Eigentum kann aufgrund von Richtlinien erfolgen, die einen unbeabsichtigten Zugriff auf IoT-Ressourcen ermöglichen. Sie sollten Ihre IoT-Richtlinien überprüfen und die Option Übermäßig freizügige Berechtigungen prüfen aktivieren, um zu tolerante Richtlinien zu identifizieren.

Verwandte Metriken:

Metrik	Begründung
Quell-IP	Wenn ein Gerät gestohlen wird, würde seine Quell-IP-Adresse außerhalb des normalerw eise zu erwartenden IP-Adressbereichs für Geräte liegen, die in einer normalen Lieferket te im Umlauf sind.
Anzahl der empfangenen Nachrichten	Da ein Angreifer ein Gerät für Cloud-bas ierten IP-Diebstahl verwenden kann, können die Metriken in Bezug auf die Anzahl der Nachrichten oder die Nachrichtengröße, die aus der AWS IoT-Cloud an das Gerät gesendet werden, stark ansteigen. Das deutet auf ein mögliches Sicherheitsproblem hin.
Nachrichtengröße	

MQTT-basierte Datenexfiltration:

Datenexfiltration tritt auf, wenn ein böswilliger Akteur eine unbefugte Datenübertragung von einer IoT-Bereitstellung oder von einem Gerät aus durchführt. Der Angreifer startet diese Art von Angriffen über MQTT gegen cloud-seitige Datenquellen.

Verwandte Metriken:

Metrik	Begründung
Quell-IP	Wenn ein Gerät gestohlen wird, würde seine Quell-IP-Adresse außerhalb des normalerw eise zu erwartenden IP-Adressbereichs für Geräte liegen, die in einer Standardlieferkette im Umlauf sind.
Anzahl der empfangenen Nachrichten	Da ein Angreifer ein Gerät in einer MQTT-basierten Datenexfiltration verwenden kann, können die Metriken in Bezug auf die Anzahl der Nachrichten oder die Nachrichtengröße, die aus der AWS IoT-Cloud an das Gerät
Nachrichtengröße	

Metrik	Begründung
	gesendet werden, stark ansteigen. Das deutet auf ein mögliches Sicherheitsproblem hin.

Identitätswechsel:

Bei einem Identitätsmissbrauch geben sich Angreifer als bekannte oder vertrauenswürdige Entitäten aus, um auf cloud-seitige AWS IoT-Services, -Anwendungen und -Daten zuzugreifen oder IoT-Geräte zu steuern und zu kontrollieren.

Verwandte Metriken:

Metrik	Begründung
Autorisierungsfehler	Wenn sich Angreifer mit gestohlenen Identitäten als vertrauenswürdige Entitäten ausgeben, steigen die Verbindungsmetriken häufig an, da die Anmeldeinformationen möglicherweise nicht mehr gültig sind oder bereits von einem vertrauenswürdigen Gerät verwendet werden. Anormales Verhalten bei Autorisierungsfehlern, Verbindungsversuchen oder Verbindungsabbrüchen deutet auf ein potenzielles Identitätsmissbrauchsszenario hin.
Verbindungsversuche	
Verbindungsabbrüche	

Missbrauch der Cloud-Infrastruktur:

Ein Missbrauch von AWS IoT-Cloud-Services liegt vor, wenn Themen mit einem hohen Nachrichtenvolumen oder mit Nachrichten in großem Umfang veröffentlicht oder abonniert werden. Übermäßig freizügige Richtlinien oder die Ausnutzung von Geräteschwachstellen zur Steuerung und Kontrolle können ebenfalls zu einem Missbrauch der Cloud-Infrastruktur führen. Eines der Hauptziele dieses Angriffs besteht darin, Ihre AWS-Rechnung zu erhöhen. Sie sollten Ihre IoT-Richtlinien überprüfen und die Option Übermäßig freizügige Berechtigungen prüfen aktivieren, um zu tolerante Richtlinien zu identifizieren.

Verwandte Metriken:

Metrik	Begründung
Anzahl der empfangenen Nachrichten	Ziel dieses Angriffs ist es, Ihre AWS-Rechnung in die Höhe zu treiben. Metriken, mit denen Aktivitäten wie die Anzahl der Nachrichten, die empfangenen Nachrichten und die Nachrichtengröße überwacht werden, werden in die Höhe schnellen.
Anzahl der gesendeten Nachrichten	
Nachrichtengröße	
Quell-IP	Es können verdächtige Quell-IP-Listen vorkommen, aus denen Angreifer ihr Nachrichtenvolumen generieren.

Geräteseitige Anwendungsfälle

Device Defender kann die folgenden Anwendungsfälle auf Ihrer Geräteseite überwachen.

Denial-of-Service-Angriff:

Ein Denial-of-Service (DoS)-Angriff zielt darauf ab, ein Gerät oder Netzwerk herunterzufahren, sodass das Gerät oder Netzwerk für die vorgesehenen Benutzer unzugänglich wird. DoS-Angriffe blockieren den Zugriff, indem sie das Ziel mit Datenverkehr überfluten oder Anfragen senden, die das System verlangsamen oder zum Ausfall des Systems führen. Ihre IoT-Geräte können bei DoS-Angriffen verwendet werden.

Verwandte Metriken:

Metrik	Begründung
Ausgegangene Pakete	DoS-Angriffe beinhalten in der Regel höhere Raten ausgehender Kommunikation von einem bestimmten Gerät aus, und je nach Art des DoS-Angriffs kann es zu einem Anstieg der Anzahl der ausgegangenen Pakete und der ausgehenden Bytes kommen.
Ausgehende Bytes	

Metrik	Begründung
Ziel-IP	Wenn Sie die IP-Adressen/CIDR-Bereiche definieren, mit denen Ihre Geräte kommunizieren sollen, kann eine Anomalie in der Ziel-IP auf eine unautorisierte IP-Kommunikation von Ihren Geräten hinweisen.
TCP-Überwachungsports	Ein DoS-Angriff erfordert in der Regel eine größere Befehls- und Kontrollinfrastruktur, in der auf Ihren Geräten installierte Malware Befehle und Informationen darüber erhält, wer und wann angegriffen werden soll. Um solche Informationen zu erhalten, überwacht die Malware daher in der Regel Ports, die normalerweise nicht von Ihren Geräten verwendet werden.
Anzahl der TCP-Überwachungsports	
UDP-Überwachungsports	
Anzahl der UDP-Überwachungsports	

Seitliche Bedrohungseskalation:

Seitliche Bedrohungseskalation beginnt in der Regel damit, dass sich ein Angreifer Zugriff auf einen Punkt im Netzwerk verschafft, z. B. auf ein verbundenes Gerät. Der Angreifer versucht dann, seine Rechte oder seinen Zugriff auf andere Geräte durch Methoden wie gestohlene Anmeldeinformationen oder das Ausnutzen von Schwachstellen zu erhöhen.

Verwandte Metriken:

Metrik	Begründung
Ausgegangene Pakete	In typischen Situationen müsste der Angreifer einen Scan im lokalen Netzwerk durchführen, um die verfügbaren Geräte ausfindig zu machen und die Auswahl seiner Angriffsziele einzugrenzen. Diese Art von Scan könnte zu einem Anstieg der Anzahl von ausgehenden Bytes und ausgegangenen Paketen führen.
Ausgehende Bytes	

Metrik	Begründung
Ziel-IP	Wenn ein Gerät mit einem bekannten Satz von IP-Adressen oder CIDRs kommunizieren soll, können Sie feststellen, ob es versucht, mit einer abnormalen IP-Adresse zu kommunizieren, bei der es sich bei einer seitlichen Bedrohungseskalation häufig um eine private IP-Adresse im lokalen Netzwerk handelt.
Autorisierungsfehler	Wenn der Angreifer versucht, seine Rechte in einem IoT-Netzwerk zu erhöhen, verwendet er möglicherweise gestohlene Anmeldeinformationen, die gesperrt wurden oder abgelaufen sind, was zu vermehrten Autorisierungsfehlern führen würde.

Datenexfiltration oder -überwachung:

Datenexfiltration tritt auf, wenn Malware oder ein böswilliger Akteur eine unbefugte Datenübertragung von einem Gerät oder einem Netzwerkendpunkt aus durchführt. Eine Datenexfiltration dient dem Angreifer in der Regel zwei Zwecken: der Beschaffung von Daten oder geistigem Eigentum oder der Erkundung eines Netzwerks. Überwachung bedeutet, dass bösartiger Code verwendet wird, um Benutzeraktivitäten zu überwachen, um Anmeldeinformationen zu stehlen und Informationen zu sammeln. Die folgenden Metriken können als Ausgangspunkt für die Untersuchung beider Arten von Angriffen dienen.

Verwandte Metriken:

Metrik	Begründung
Ausgegangene Pakete	Bei Angriffen durch Datenexfiltration oder -überwachung spiegelt der Angreifer häufig die vom Gerät gesendeten Daten wider, anstatt die Daten einfach umzuleiten, was der Defender erkennen würde, wenn er die beabsichtigten Daten nicht sieht.
Ausgehende Bytes	

Metrik	Begründung
	Solche gespiegelten Daten würden die Gesamtmenge der vom Gerät gesendete n Daten erheblich erhöhen, was zu einem Anstieg der Anzahl von ausgegangenen Paketen und ausgehenden Bytes führen würde.
Ziel-IP	Wenn ein Angreifer ein Gerät für Angriffe mit Datenexfiltration oder -überwachung verwendet, müssten die Daten an eine abnormale IP-Adresse gesendet werden, die vom Angreifer kontrolliert wird. Die Überwachung der Ziel-IP kann helfen, einen solchen Angriff zu identifizieren.

Mining von Kryptowährungen

Angreifer nutzen die Rechenleistung von Geräten zum Mining von Kryptowährungen. Krypto-Mining ist ein rechenintensiver Prozess, der in der Regel eine Netzwerkkommunikation mit anderen Mining-Peers und -Pools erfordert.

Verwandte Metriken:

Metrik	Begründung
Ziel-IP	Netzwerkkommunikation ist in der Regel eine Anforderung beim Krypto-Mining. Eine streng kontrollierte Liste von IP-Adressen, mit denen das Gerät kommunizieren soll, kann dabei helfen, unbeabsichtigte Kommunikation auf einem Gerät zu identifizieren, z. B. beim Mining von Kryptowährungen.
Benutzerdefinierte Metrik zur CPU-Auslastung	Das Mining von Kryptowährungen erfordert intensive Berechnungen, was zu einer hohen Auslastung der Geräte-CPU führt. Wenn

Metrik	Begründung
	Sie sich dafür entscheiden, diese Metrik zu erfassen und zu überwachen, könnte eine höhere CPU-Auslastung als im Normalfall ein Indikator für Krypto-Mining-Aktivitäten sein.

Steuerung und Kontrolle, Malware und Ransomware

Malware oder Ransomware schränkt Ihre Kontrolle über Ihre Geräte ein und beschränkt die Funktionalität Ihrer Geräte. Im Falle eines Ransomware-Angriffs würde der Datenzugriff aufgrund der von der Ransomware verwendeten Verschlüsselung verloren gehen.

Verwandte Metriken:

Metrik	Begründung
Ziel-IP	Netzwerk- oder Remoteangriffe machen einen großen Teil der Angriffe auf IoT-Geräte aus. Eine streng kontrollierte Liste von IP-Adress en, mit denen das Gerät kommunizieren soll, kann dabei helfen, abnormale Ziel-IPs zu identifizieren, die auf einen Malware- oder Ransomware-Angriff zurückzuführen sind.
TCP-Überwachungsports	Bei mehreren Malware-Angriffen wird ein Command-and-Control-Server gestartet, der Befehle zur Ausführung auf ein Gerät sendet. Dieser Servertyp ist für einen Malware- oder Ransomware-Vorgang von entscheidender Bedeutung und kann identifiziert werden, indem die offenen TCP/UDP-Ports und die Anzahl der Ports genau überwacht werden.
Anzahl der TCP-Überwachungsports	
UDP-Überwachungsports	
Anzahl der UDP-Überwachungsports	

Konzepte

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

Metrik

AWS IoT Device Defender Detect verwendet Metriken zur Erkennung ungewöhnlicher Verhaltensweisen. AWS IoT Device Defender Detect vergleicht den gemeldeten Wert einer Metrik mit dem von Ihnen angegebenen erwarteten Wert. Diese Metriken können aus zwei Quellen stammen: cloudseitige Metriken und geräteseitige Metriken. ML Detect unterstützt 6 cloudseitige Metriken und 7 geräteseitige Metriken. Eine Liste der unterstützten Metriken für ML Detect finden Sie unter Unterstützte Metriken.

Ungewöhnliche Verhaltenswesen im AWS IoT-Netzwerk werden mithilfe von cloudseitigen Metriken, wie z. B. der Anzahl der Autorisierungsfehler oder der Anzahl oder Größe der von einem Gerät über AWS IoT gesendeten oder empfangenen Nachrichten erkannt.

AWS IoT Device Defender Detect kann auch die von AWS IoT-Geräten generierten Metrikdaten (z. B. die von einem Gerät überwachten Ports, die Anzahl der gesendeten Bytes oder Pakete oder die TCP-Verbindungen des Geräts) erkennen, erfassen und aggregieren.

Sie können AWS IoT Device Defender Detect nur mit cloudseitigen Metriken verwenden. Wenn Sie geräteseitige Metriken verwenden möchten, müssen Sie auf Ihren über AWS IoT verbundenen Geräten oder Geräte-Gateways zunächst einen das AWS IoT SDK zum Erfassen und Senden der Metriken an AWS IoT bereitstellen. Siehe Senden von Metriken von Geräten.

Sicherheitsprofil

Ein Sicherheitsprofil definiert anomale Verhaltensweisen für eine Gruppe von Geräten (eine statische Objektgruppe) oder für alle Geräte in Ihrem Konto und gibt an, welche Aktionen unternommen werden sollen, wenn eine Anomalie erkannt wird. Sie können mithilfe der AWS IoT-Konsole oder API-Befehle ein Sicherheitsprofil erstellen und es einer Gruppe von Geräten zuordnen. AWS IoT Device Defender Detect startet mit der Aufzeichnung sicherheitsrelevanter

Daten und erkennt anhand der im Sicherheitsprofil definierten Verhaltensweisen Anomalien im Verhalten der Geräte.

Verhalten

Ein Verhalten teilt AWS IoT Device Defender Detect mit, wie es erkennt, wenn sich ein Gerät ungewöhnlich verhält. Jede Geräteaktion, die nicht mit einer Verhaltensweise übereinstimmt, löst einen Alarm aus. Eine Verhaltensweise von Rules Detect besteht aus einer Metrik und einem absoluten Wert oder einem statistischen Schwellenwert mit einem Operator (z. B. kleiner als oder gleich, größer als oder gleich), der das erwartete Geräteverhalten beschreibt. Ein ML-Erkennungsverhalten besteht aus einer Metrik und einer ML Detect-Konfiguration, die ein ML-Modell so einrichten, dass es das normale Verhalten von Geräten lernt.

ML-Modell

Ein ML-Modell ist ein Machine Learning-Modell, das erstellt wurde, um jedes Verhalten zu überwachen, das ein Kunde konfiguriert. Das Modell trainiert anhand von metrischen Datenmustern bestimmter Gerätegruppen und generiert drei Schwellenwerte für die Zuverlässigkeit von Anomalien (hoch, mittel und niedrig) für das metrikbasierte Verhalten. Es leitet auf der Grundlage aufgenommener metrischer Daten auf Geräteebene auf Anomalien ab. Im Kontext von ML Detect wird ein ML-Modell erstellt, um ein metrikbasiertes Verhalten zu bewerten. Weitere Informationen finden Sie unter ML Detect.

Konfidenzniveau

ML Detect unterstützt drei Konfidenzstufen:`High`, `Medium` und `Low`. Konfidenz von `High` bedeutet eine geringe Sensitivität bei der Bewertung von anomalem Verhalten und häufig eine geringere Anzahl an Alarmen; eine Konfidenz von `Medium` bedeutet eine mittlere Empfindlichkeit, und eine Konfidenz von `Low` bedeutet eine hohe Empfindlichkeit und häufig eine höhere Anzahl an Alarmen.

Dimension

Sie können eine Dimension definieren, um den Bereich eines Verhaltens anzupassen. Sie können beispielsweise eine Themenfilterdimension definieren, die ein Verhalten auf MQTT-Themen anwendet, die einem Muster entsprechen. Informationen zum Definieren einer Dimension für die Verwendung in einem Sicherheitsprofil finden Sie unter CreateDimension.

Alarm

Wenn eine Anomalie erkannt wird, kann über eine Alarmbenachrichtigung über eine CloudWatch-Metrik (siehe Überwachen von AWS IoT-Alarmen und Metriken mithilfe von Amazon CloudWatch

im AWS IoT Core-Entwicklerhandbuch) oder eine SNS-Benachrichtigung gesendet werden. Eine Alarmbenachrichtigung wird auch in der AWS IoT-Konsole zusammen mit Informationen über den Alarm und einem Verlauf der Alarme für das Gerät angezeigt. Ein Alarm wird auch gesendet, wenn ein überwachtes Gerät kein ungewöhnliches Verhalten mehr zeigt oder wenn es über einen längeren Zeitraum keine Daten mehr meldet, nachdem wegen ihm ein Alarm ausgelöst wurde.

Alarmverifizierungsstatus

Nachdem ein Alarm erstellt wurde, können Sie den Alarm als „Wahr positiv“, „Gutartig positiv“, „Falsch positiv“ oder „Unbekannt“ überprüfen. Sie können Ihrem Alarmverifizierungsstatus auch eine Beschreibung hinzufügen. Sie können AWS IoT Device Defender-Alarme anzeigen, organisieren und filtern, indem Sie einen der vier Verifizierungsstatus verwenden. Sie können den Status der Alarmverifizierung und zugehörige Beschreibungen verwenden, um Mitglieder Ihres Teams zu informieren. Dies hilft Ihrem Team, Folgemaßnahmen zu ergreifen, z. B. Abhilfemaßnahmen bei Alarmen mit dem Status „Wahr positiv“ durchzuführen, Alarme mit dem Status „Gutartig positiv“ zu überspringen oder die Untersuchung bei Alarmen mit dem Status „Unbekannt“ fortzusetzen. Der Standard-Verifizierungsstatus für alle Alarme ist Unbekannt.

Unterdrückung von Alarmen

Sie können Amazon SNS-Benachrichtigungen über Detect Alarm verwalten, indem Sie die Verhaltensbenachrichtigung auf on oder `suppressed` festlegen. Das Unterdrücken von Alarmen hindert Detect nicht daran, das Geräteverhalten zu bewerten. Detect kennzeichnet anomale Verhaltensweisen weiterhin als Alarme bei Verstößen. Unterdrückte Alarme werden jedoch nicht für SNS-Benachrichtigungen weitergeleitet. Auf sie kann nur über die AWS IoT-Konsole oder API zugegriffen werden.

Verhaltensweisen

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

Ein Sicherheitsprofil enthält eine Reihe von Verhaltensweisen. Jede Verhaltensweise enthält eine Kennzahl, die das normale Verhalten für eine Gruppe von Geräten oder für alle Geräte in Ihrem

Konto angibt. Verhalten lässt sich in zwei Kategorien einteilen: Rules Detect-Verhalten und ML Detect-Verhalten. Mit den Rules Detect-Verhaltensweisen definieren Sie, wie sich Ihre Geräte verhalten sollen, wohingegen ML Detect zur Bewertung, wie sich Ihre Geräte verhalten sollen, ML-Modelle verwendet.

Bei einem Sicherheitsprofil kann es sich um einen von zwei Schwellenwerttypen handeln: ML oder Regelbasiert. ML-Sicherheitsprofile erkennen durch Lernen aus früheren Daten automatisch operative und Sicherheitsanomalien auf Geräteebene in Ihrer Flotte. Regelbasierte Sicherheitsprofile erfordern, dass Sie manuell statische Regeln festlegen, um das Verhalten Ihres Geräts zu überwachen.

Im Folgenden sind einige der Felder beschrieben, die in der `behavior`-Definition verwendet werden:

Gemeinsamkeiten von Rules Detect und ML Detect

`name`

Der Name für die Verhaltensweise.

`metric`

Der Name der verwendeten Metrik (d. h. das, was anhand der Verhaltensweise gemessen wird).

`consecutiveDatapointsToAlarm`

Wenn ein Gerät bei der angegebenen Anzahl an aufeinanderfolgenden Datenpunkten gegen das Verhalten verstößt, wird ein Alarm ausgegeben. Wenn nichts angegeben ist, ist der Standardwert 1.

`consecutiveDatapointsToClear`

Wenn ein Alarm aufgetreten ist und das betreffende Gerät nicht mehr gegen das Verhalten für die angegebene Anzahl aufeinander folgender Datenpunkte verstößt, wird der Alarm deaktiviert. Wenn nichts angegeben ist, ist der Standardwert 1.

`threshold type`

Bei einem Sicherheitsprofil kann es sich um einen von zwei Schwellenwerttypen handeln: ML oder Regelbasiert. ML-Sicherheitsprofile erkennen durch Lernen aus früheren Daten automatisch operative und Sicherheitsanomalien auf Geräteebene in Ihrer Flotte. Regelbasierte Sicherheitsprofile erfordern, dass Sie manuell statische Regeln festlegen, um das Verhalten Ihres Geräts zu überwachen.

`alarm suppressions`

Sie können Amazon SNS-Benachrichtigungen über Detect Alarm verwalten, indem Sie die Verhaltensbenachrichtigung auf on oder `suppressed` setzen. Das Unterdrücken von Alarmen hindert Detect nicht daran, das Geräteverhalten zu bewerten. Detect kennzeichnet anomale Verhaltensweisen weiterhin als Alarme bei Verstößen. Unterdrückte Alarme werden jedoch nicht für Amazon SNS-Benachrichtigungen weitergeleitet. Auf sie kann nur über die AWS IoT-Konsole oder API zugegriffen werden.

Rules Detect

`dimension`

Sie können eine Dimension definieren, um den Bereich eines Verhaltens anzupassen. Sie können beispielsweise eine Themenfilterdimension definieren, die ein Verhalten auf MQTT-Themen anwendet, die einem Muster entsprechen. Informationen zum Definieren einer Dimension für die Verwendung in einem Sicherheitsprofil finden Sie unter CreateDimension. Gilt nur für Rules Detect.

`criteria`

Die Kriterien, die bestimmen, ob sich ein Gerät im Hinblick auf die `metric` normal verhält.

> **Note**
>
> Auf der AWS IoT-Konsole können Sie Mich warnen auswählen, um über Amazon SNS benachrichtigt zu werden, wenn AWS IoT Device Defender feststellt, dass sich ein Gerät ungewöhnlich verhält.

`comparisonOperator`

Der Operator, der das gemessene Objekt (`metric`) zu den Kriterien (`value` oder `statisticalThreshold`) in Beziehung setzt.

Mögliche Werte sind: „less-than“, „less-than-equals“, „greater-than“, „greater-than-equals“, „in-cidr-set“, „not-in-cidr-set“, „in-port-set“ und „not-in-port-set“. Nicht alle Operatoren sind für jede Metrik gültig. Operatoren für CIDR-Sets und Ports können nur mit Metriken mit solchen Entitäten verwendet werden.

`value`

Der Wert, der mit `metric` verglichen werden soll. Abhängig von der Art der Metrik sollte dies `count` (ein Wert), `cidrs` (eine Liste von CIDRs) oder `ports` (eine Liste von Ports) enthalten.

`statisticalThreshold`

Der statistische Schwellenwert, durch den eine Verhaltensverletzung bestimmt wird. Dieses Feld enthält ein `statistic`-Feld mit den folgenden möglichen Werten: „p0“, „p0.1“, „p0.01“, „p1“, „p10“, „p50“, „p90“, „p99“, „p99.9“, „p99.99“ oder „p100“.

Diese `statistic` gibt an, dass es sich um ein Perzentil handelt. Sie wird in einen Wert aufgelöst, über den die Compliance mit dem Verhalten bestimmt wird. Metriken werden über die angegebene Dauer (`durationSeconds`) einmal oder mehrfach von allen Meldegeräten erfasst, die diesem Sicherheitsprofil zugeordnet sind. Die Berechnung der Perzentile erfolgt auf Basis dieser Daten. Danach werden die Metriken für ein Gerät erfasst und über die gleiche Dauer akkumuliert. Wenn das Ergebnis für das Gerät den dem angegebenen Perzentil zugeordneten Wert über- oder unterschreitet (`comparisonOperator`), entspricht das Gerät dem Verhalten. Andernfalls liegt ein Verstoß gegen das Verhalten vor.

Ein Perzentil gibt den Prozentsatz aller berücksichtigen Messwerte an, die unterhalb des zugeordneten Werts liegen. Beispiel: Wenn der „p90“ zugeordnete Wert (das 90. Perzentil) „123“ beträgt, lagen 90 % aller Messwerte unterhalb des Werts „123“.

`durationSeconds`

Hiermit können Sie den Zeitraum angeben, über den hinweg die Verhaltensweise hinsichtlich solcher Parameter bewertet wird, die eine Zeitdimension (z. B.) aufweisen., `NUM_MESSAGES_SENT`). Bei einem metrischen `statisticalThreshhold`-Vergleich ist dies der Zeitraum, in dem für alle Geräte Metriken zum Bestimmen der `statisticalThreshold`-Werte erfasst werden. Dann werden die Metriken für jedes einzelne Gerät ermittelt, um das Verhalten der verschiedenen Geräte vergleichen zu können.

ML Detect

`ML Detect confidence`

ML Detect unterstützt drei Konfidenzstufen:`High`, `Medium` und `Low`. Konfidenz von `High` bedeutet eine geringe Sensitivität bei der Bewertung von anormalem Verhalten und häufig eine geringere Anzahl an Alarmen, eine Konfidenz von `Medium` bedeutet eine mittlere Empfindlichkeit und

eine Konfidenz von Low bedeutet eine hohe Empfindlichkeit und häufig eine höhere Anzahl an Alarmen.

ML Detect

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

> **Note**
>
> ML Detect ist in den folgenden Regionen nicht verfügbar:
>
> - Asien-Pazifik (Malaysia)

Mit Machine Learning Detect (ML Detect) erstellen Sie Sicherheitsprofile, die Machine Learning nutzen, um das erwartete Geräteverhalten zu ermitteln. Dabei erstellen sie automatisch Modelle auf der Grundlage historischer Gerätedaten und weisen diese Profile einer Gruppe von Geräten oder allen Geräten in Ihrer Flotte zu. Dann identifiziert AWS IoT Device Defender mithilfe der ML-Modelle Anomalien und löst Alarme aus.

Weitere Informationen über die ersten Schritte mit ML Detect finden Sie unter ML Detect-Handbuch.

Dieses Kapitel enthält die folgenden Abschnitte:

- Anwendungsfälle von ML Detect
- So funktioniert ML Detect
- Mindestanforderungen
- Einschränkungen
- Markierung von Fehlalarmen und anderen Bestätigungszuständen in Alarmen
- Unterstützte Metriken
- Service Quotas

- CLI-Befehle von ML Detect
- ML Detect APIs
- Anhalten oder Löschen eines ML Detect-Sicherheitsprofils

Anwendungsfälle von ML Detect

Sie können ML Detect verwenden, um Ihre Flottengeräte zu überwachen, wenn es schwierig ist, die erwartete Verhaltensweise von Geräten festzulegen. So ist bei der Überwachung der Metrik „Anzahl der Verbindungsabbrüche“ möglicherweise nicht klar, welcher Schwellenwert als akzeptabel angesehen wird. In diesem Fall können Sie ML Detect aktivieren, um anhand von historischen Daten, die von Geräten gemeldet wurden, ungewöhnliche Datenpunkte der Metrik für Verbindungsabbrüche zu identifizieren.

Ein weiterer Anwendungsfall von ML Detect ist die Überwachung des Geräteverhaltens, das sich im Laufe der Zeit dynamisch ändert. ML Detect lernt in regelmäßigen Abständen das erwartete dynamische Geräteverhalten auf der Grundlage von sich ändernden Datenmustern von Geräten. So kann das Volumen der gesendeten Gerätenachrichten beispielsweise zwischen Wochentagen und Wochenenden variieren, und ML Detect lernt dieses dynamische Verhalten.

So funktioniert ML Detect

Mit ML Detect können Sie Verhaltensweisen erstellen, um Betriebs- und Sicherheitsanomalien anhand von 6 cloudseitigen Metriken und 7 geräteseitigen Metriken zu identifizieren. Nach der ersten Modelltrainingsphase aktualisiert ML Detect die Modelle täglich auf der Grundlage der Daten der letzten 14 Tage. Es überwacht Datenpunkte für diese Metriken mit den ML-Modellen und löst einen Alarm aus, wenn eine Anomalie erkannt wird.

ML Detect funktioniert am besten, wenn Sie ein Sicherheitsprofil an eine Sammlung von Geräten mit einer ähnlichen erwarteten Verhaltensweise anhängen. Wenn beispielsweise einige Ihrer Geräte bei Kunden zu Hause und andere Geräte in Geschäftsbüros verwendet werden, können sich die Verhaltensmuster der Geräte zwischen den beiden Gruppen erheblich unterscheiden. Sie können die Geräte in eine Objektgruppe für Privatgeräte und eine Objektgruppe für Bürogeräte klassifizieren. Um eine optimale Wirksamkeit bei der Erkennung von Anomalien zu erzielen, fügen Sie jede Objektgruppe einem separaten ML Detect-Sicherheitsprofil hinzu.

Während ML Detect das erste Modell erstellt, benötigt es 14 Tage und mindestens 25.000 Datenpunkte pro Metrik in den letzten 14 Tagen, um ein Modell zu generieren. Danach

aktualisiert es das Modell an jedem Tag, an dem eine Mindestanzahl an metrischen Datenpunkten vorhanden ist. Wenn die Mindestanforderung nicht erfüllt ist, versucht ML Detect, das Modell am nächsten Tag zu erstellen, und versucht es in den nächsten 30 Tagen täglich erneut, bevor das Modell für Evaluierungen eingestellt wird.

Mindestanforderungen

Für das Training und die Erstellung des ersten ML-Modells gelten für ML Detect folgende Mindestanforderungen.

Mindestausbildungsdauer

Es dauert 14 Tage, bis die ersten Modelle erstellt sind. Danach wird das Modell täglich mit metrischen Daten aus einem Zeitraum von 14 Tagen aktualisiert.

Mindestgesamtanzahl an Datenpunkten

Für die Erstellung eines ML-Modells sind für die letzten 14 Tage mindestens 25.000 Datenpunkte pro Metrik erforderlich. Für das kontinuierliche Training und die Aktualisierung des Modells setzt ML Detect voraus, dass die Mindestanzahl an Datenpunkten der überwachten Geräte erfüllt wird. Das entspricht in etwa den folgenden Konfigurationen:

- 60 Geräte stellen eine Verbindung her und sind auf AWS IoT in Intervallen von 45 Minuten aktiv.
- 40 Geräte in 30-Minuten-Intervallen.
- 15 Geräte in 10-Minuten-Intervallen.
- 7 Geräte in 5-Minuten-Intervallen.

Gerätegruppenziele

Um Daten zu erfassen, müssen Sie in den Zielgruppen für das Sicherheitsprofil über Objekte verfügen.

Nach der Erstellung des ersten Modells werden ML-Modelle täglich aktualisiert und benötigen für den Zeitraum von 14 Tagen mindestens 25.000 Datenpunkte.

Einschränkungen

Mit ML Detect können Sie die folgenden cloudseitigen Metriken mit Dimensionen verwenden:

- Autorisierungsfehler (aws:num-authorization-failures)

- Empfangene Nachrichten (aws:num-messages-received)
- Gesendete Nachrichten (aws:num-messages-sent)
- Nachrichtengröße (aws:message-byte-size)

Die folgenden Metriken werden von ML Detect nicht unterstützt.

Nicht von ML Detect unterstützte cloudseitige Metriken:

- Quell-IP (aws:source-ip-address)

Nicht von ML Detect unterstützte geräteseitige Metriken:

- Ziel-IPs (`aws:destination-ip-addresses`)
- Überwachen von TCP-Ports (`aws:listening-tcp-ports`)
- Überwachen von UDP-Ports (`aws:listening-udp-ports`)

Benutzerdefinierte Metriken unterstützen nur den Typ Zahlen.

Markierung von Fehlalarmen und anderen Bestätigungszuständen in Alarmen

Wenn Sie im Rahmen Ihrer Untersuchung sicherstellen, dass ein ML Detect-Alarm falsch positiv ist, können Sie den Bestätigungsstatus des Alarms auf Falsch positiv setzen. Dies kann Ihnen und Ihrem Team helfen, Alarme zu identifizieren, auf die Sie nicht reagieren müssen. Sie können Alarme auch als „Wahr positiv“, „Gutartig positiv“ oder „Unbekannt“ markieren.

Sie können Alarme über die AWS IoT Device Defender Konsole oder mithilfe der API-Aktion PutVerificationStateOnViolation markieren.

Unterstützte Metriken

Mit ML Detect können Sie die folgenden cloudseitigen Metriken verwenden:

- Autorisierungsfehler (aws:num-authorization-failures)
- Verbindungsversuche (aws:num-connection-attempts)
- Unterbricht die Verbindung (aws:num-disconnects)

- Nachrichtengröße (aws:message-byte-size)
- Gesendete Nachrichten (aws:num-messages-sent)
- Empfangene Nachrichten (aws:num-messages-received)

Mit ML Detect können Sie die folgenden geräteseitigen Metriken verwenden:

- Ausgehende Bytes (`aws:all-bytes-out`)
- Bytes in (`aws:all-bytes-in`)
- Überwachen der Anzahl an TCP-Ports (`aws:num-listening-tcp-ports`)
- Überwachen der Anzahl an UDP-Ports (`aws:num-listening-udp-ports`)
- Ausgehende Pakete (`aws:all-packets-out`)
- Pakete in (`aws:all-packets-in`)
- Anzahl etablierter TCP-Verbindungen (`aws:num-established-tcp-connections`)

Service Quotas

Weitere Informationen zu Service Quotas in ML Detect finden Sie unter AWS IoT Device DefenderEndpunkte und Kontingente.

CLI-Befehle von ML Detect

Mit den folgenden CLI-Befehlen können Sie ML Detect erstellen und verwalten.

- 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 APIs

Die folgenden APIs können verwendet werden, um ML Detect-Sicherheitsprofile zu erstellen und zu verwalten.

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

Anhalten oder Löschen eines ML Detect-Sicherheitsprofils

Sie können Ihr ML Detect-Sicherheitsprofil anhalten, um die Überwachung des Geräteverhaltens vorübergehend zu beenden, oder Ihr ML Detect-Sicherheitsprofil löschen, um die Überwachung des Geräteverhaltens über einen längeren Zeitraum zu beenden.

ML Detect-Sicherheitsprofil mit der Konsole anhalten

Um ein ML Detect-Sicherheitsprofil mithilfe der Konsole anzuhalten, müssen Sie zunächst über eine leere Objektgruppe verfügen. Informationen zum Erstellen einer leeren Objektgruppe finden Sie unter Statische Objektgruppen im AWS IoT Core-Entwicklerhandbuch. Wenn Sie eine leere Objektgruppe erstellt haben, legen Sie die leere Objektgruppe als Ziel des ML Detect-Sicherheitsprofils fest.

> **Note**
>
> Sie müssen das Ziel Ihres Sicherheitsprofils innerhalb von 30 Tagen wieder auf eine Gerätegruppe mit Geräten festlegen. Andernfalls können Sie das Sicherheitsprofil nicht reaktivieren.

ML Detect-Sicherheitsprofil mit der Konsole löschen

Gehen Sie folgendermaßen vor, um ein Sicherheitsprofil zu löschen:

1. Navigieren Sie in der AWS IoT-Konsole zur Seitenleiste, und wählen Sie den Bereich Verteidigen.
2. Wählen Sie unter Verteidigen die Option Erkennen und dann Security Profiles.
3. Wählen Sie das ML Detect-Sicherheitsprofil aus, das Sie löschen möchten.
4. Wählen Sie Aktionen und anschießend Löschen aus dem Menü.

> **Note**
>
> Nachdem ein ML Detect-Sicherheitsprofil gelöscht wurde, können Sie das Sicherheitsprofil nicht mehr reaktivieren.

ML Detect-Sicherheitsprofil mit der CLI anhalten

Verwenden Sie den Befehl `detach-security-security-profile`, um ein ML Detect-Sicherheitsprofil mithilfe der CLI anzuhalten:

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

> **Note**
>
> Diese Option ist nur in der AWS CLI verfügbar. Ähnlich wie beim Konsolen-Workflow müssen Sie das Ziel Ihres Sicherheitsprofils innerhalb von 30 Tagen wieder auf eine Gerätegruppe mit Geräten festlegen. Andernfalls können Sie das Sicherheitsprofil nicht reaktivieren. Verwenden Sie den Befehl `attach-security-profile`, um ein Sicherheitsprofil an eine Gerätegruppe anzuhängen.

ML Detect-Sicherheitsprofil mit der CLI löschen

Sie können ein Sicherheitsprofil mit dem Befehl `delete-security-profile` unten löschen:

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

> **Note**
>
> Nachdem ein ML Detect-Sicherheitsprofil gelöscht wurde, können Sie das Sicherheitsprofil nicht mehr reaktivieren.

Benutzerdefinierte Metriken

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

Mit AWS IoT Device Defender benutzerdefinierten Metriken können Sie Kennzahlen definieren und überwachen, die für Ihre Flotte oder Ihren Anwendungsfall spezifisch sind, z. B. die Anzahl der mit Wi-Fi-Gateways verbundenen Geräte, den Ladezustand von Akkus oder die Anzahl der Netzzyklen für intelligente Steckverbinder. Benutzerdefiniertes metrisches Verhalten wird in Sicherheitsprofilen definiert, die das erwartete Verhalten für eine Gruppe von Geräten (eine Objektgruppe) oder für alle Geräte spezifizieren. Sie können das Verhalten überwachen, indem Sie Alarme einrichten, anhand derer Sie gerätespezifische Probleme erkennen und darauf reagieren können.

Dieses Kapitel enthält die folgenden Abschnitte:

- So verwenden Sie benutzerdefinierte Metriken auf der Konsole
- So verwenden Sie benutzerdefinierte Metriken von der CLI
- CLI-Befehle für benutzerdefinierte Metriken
- Benutzerdefinierte Metriken-APIs

So verwenden Sie benutzerdefinierte Metriken auf der Konsole

Tutorials

- AWS IoT Device Defender Agent SDK (Python)
- Erstellen Sie eine benutzerdefinierte Metrik, und fügen Sie sie einem Sicherheitsprofil hinzu.

- Anzeigen der Details von benutzerdefinierten Metriken
- Aktualisieren einer benutzerdefinierten Metrik
- Löschen einer benutzerdefinierten Metrik

AWS IoT Device Defender Agent SDK (Python)

Laden Sie zunächst den Beispielagenten des AWS IoT Device Defender Agenten SDK (Python) herunter. Der Agent erfasst die Metriken und veröffentlicht Berichte. Sobald Ihre geräteseitigen Metriken veröffentlicht wurden, können Sie sich die erfassten Messwerte ansehen und Schwellenwerte für die Einrichtung von Alarmen festlegen. Anweisungen zur Einrichtung des Device Agents finden Sie in der Readme-Datei des AWS IoT Device Defender Agent SDK (Python). Weitere Informationen finden Sie unter AWS IoT Device Defender Agent SDK (Python).

Erstellen Sie eine benutzerdefinierte Metrik, und fügen Sie sie einem Sicherheitsprofil hinzu.

Das folgende Verfahren zeigt Ihnen, wie Sie eine benutzerdefinierte Metrik auf der Konsole erstellen können.

1. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoTKonsole, und wählen Sie dann Detect, Metriken.
2. Wählen Sie auf der Seite Benutzerdefinierte Metriken die Option Erstellen.
3. Führen Sie auf der Seite Benutzerdefinierte Metrik erstellen die folgenden Schritte aus:

    1. Geben Sie unter Name einen Namen für Ihre benutzerdefinierte Metrik ein. Sie können diesen Namen nicht mehr ändern, nachdem Sie die benutzerdefinierte Metrik erstellt haben.
    2. Unter Anzeigename (optional) können Sie einen Anzeigenamen für Ihre benutzerdefinierte Metrik eingeben. Er muss nicht eindeutig sein und kann nach der Erstellung geändert werden.
    3. Wählen Sie unter Typ den Typ der Metrik, die Sie überwachen möchten. Zu den Metriktypen gehören string-list, ip-address-list, number-list und number. Der Typ kann nach der Erstellung nicht geändert werden.

        > **Note**
        >
        > ML Detect erlaubt nur den Typ number.

    4. Unter Tags können Sie Tags auswählen, die der Ressource zugeordnet werden sollen.

Wählen Sie abschließend Bestätigen.

4. Nachdem Sie Ihre benutzerdefinierte Metrik erstellt haben, wird die Seite Benutzerdefinierte Metrik angezeigt, auf der Sie Ihre neu erstellte benutzerdefinierte Metrik sehen können.
5. Als Nächstes müssen Sie Ihre benutzerdefinierte Metrik einem Sicherheitsprofil hinzufügen. Erweitern Sie auf der AWS IoT-Konsole im Navigationsbereich die Option Verteidigen, und wählen Sie dann Erkennen, Sicherheitsprofile.
6. Wählen Sie das Sicherheitsprofil, zu dem Sie Ihre benutzerdefinierte Metrik hinzufügen möchten.
7. Wählen Sie Aktionen und Bearbeiten.
8. Wählen Sie Zusätzliche Metriken zum Beibehalten, und wählen Sie dann Ihre benutzerdefinierte Metrik. Wählen Sie auf den folgenden Bildschirmen Weiter, bis Sie zur Seite Bestätigen gelangen. Wählen Sie Speichern und Fortfahren. Nachdem Ihre benutzerdefinierte Metrik erfolgreich hinzugefügt wurde, wird die Sicherheitsprofil-Detailseite angezeigt.

> **Note**
>
> Perzentil-Statistiken für Metriken sind nicht verfügbar, wenn es Metrik-Werte gibt, die negative Zahlen enthalten.

Anzeigen der Details von benutzerdefinierten Metriken

Das folgende Verfahren zeigt Ihnen, wie Sie eine benutzerdefinierte Metrik auf der Konsole anzeigen können.

1. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoTKonsole, und wählen Sie dann Detect, Metriken.
2. Wählen Sie den Metriknamen der benutzerdefinierten Metrik, deren Details Sie anzeigen möchten.

Aktualisieren einer benutzerdefinierten Metrik

Das folgende Verfahren zeigt Ihnen, wie Sie eine benutzerdefinierte Metrik auf der Konsole aktualisieren können.

1. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoTKonsole, und wählen Sie dann Detect, Metriken.
2. Wählen Sie das Optionsfeld neben der benutzerdefinierten Metrik, die Sie aktualisieren möchten. Wählen Sie dann unter Aktionen die Option Bearbeiten.
3. Auf der Seite Benutzerdefinierte Metrik aktualisieren können Sie den Anzeigenamen bearbeiten und Tags entfernen oder hinzufügen.
4. Klicken Sie abschließend auf Aktualisieren. Die Seite Benutzerdefinierte Metriken.

Löschen einer benutzerdefinierten Metrik

Das folgende Verfahren zeigt Ihnen, wie Sie eine benutzerdefinierte Metrik von der Konsole löschen können.

1. Entfernen Sie zunächst Ihre benutzerdefinierte Metrik aus allen Sicherheitsprofilen, die darauf verweisen. Sie können auf der Detailseite Ihrer benutzerdefinierten Metrik sehen, welche Sicherheitsprofile Ihre benutzerdefinierte Metrik enthalten. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoTKonsole, und wählen Sie dann Detect, Metriken.
2. Wählen Sie die benutzerdefinierte Metrik, die Sie entfernen möchten. Entfernen Sie die benutzerdefinierte Metrik aus allen Sicherheitsprofilen, die auf der Detailseite der benutzerdefinierten Metrik unter Sicherheitsprofile aufgeführt sind.
3. Erweitern Sie Verteidigen im Navigationsbereich der AWS IoTKonsole, und wählen Sie dann Detect, Metriken.
4. Wählen Sie das Optionsfeld neben der benutzerdefinierten Metrik, die Sie löschen möchten. Wählen Sie dann unter Aktionen die Option Löschen.
5. Wählen Sie in der Nachricht Soll die benutzerdefinierte Metrik wirklich gelöscht werden? die Option Benutzerdefinierte Metrik löschen.

> **Warning**
>
> Nachdem Sie eine benutzerdefinierte Metrik gelöscht haben, verlieren Sie alle mit der Metrik verknüpften Daten. Diese Aktion kann nicht mehr rückgängig gemacht werden.

So verwenden Sie benutzerdefinierte Metriken von der CLI

Tutorials

- AWS IoT Device Defender Agent SDK (Python)
- Erstellen Sie eine benutzerdefinierte Metrik, und fügen Sie sie einem Sicherheitsprofil hinzu.
- Anzeigen der Details von benutzerdefinierten Metriken
- Aktualisieren einer benutzerdefinierten Metrik
- Löschen einer benutzerdefinierten Metrik

AWS IoT Device Defender Agent SDK (Python)

Laden Sie zunächst den Beispielagenten des AWS IoT Device Defender Agenten SDK (Python) herunter. Der Agent erfasst die Metriken und veröffentlicht Berichte. Sobald Ihre geräteseitigen Metriken veröffentlicht wurden, können Sie sich die erfassten Metriken ansehen und Schwellenwerte für die Einrichtung von Alarmen festlegen. Anweisungen zur Einrichtung des Device Agents finden Sie in der Readme-Datei des AWS IoT Device Defender Agent SDK (Python). Weitere Informationen finden Sie unter AWS IoT Device Defender Agent SDK (Python).

Erstellen Sie eine benutzerdefinierte Metrik, und fügen Sie sie einem Sicherheitsprofil hinzu.

Das folgende Verfahren zeigt Ihnen, wie Sie eine benutzerdefinierte Metrik erstellen und sie über die CLI zu einem Sicherheitsprofil hinzufügen können.

1. Verwenden Sie den Befehl `create-custom-metric`, um Ihre benutzerdefinierte Metrik zu erstellen. Im folgenden Beispiel wird eine benutzerdefinierte Metrik erstellt, die den Akkuladestand misst.

```
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" \
```

Ausgabe:

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

```
}
```

2. Nachdem Sie Ihre benutzerdefinierte Metrik erstellt haben, können Sie die benutzerdefinierte Metrik entweder mit `update-security-profile` zu einem vorhandenen Profil hinzufügen oder mit `create-security-profile` ein neues Sicherheitsprofil erstellen, um die benutzerdefinierte Metrik hinzuzufügen. Hier erstellen wir ein neues Sicherheitsprofil namens *`BatteryUsage`*, zu dem wir unsere neue benutzerdefinierte *`BatteryPercentage`*-Metrik hinzufügen können. Wir fügen ferner eine Rules Detect-Metrik namens *`CellularBandwidth`* hinzu.

```
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
```

Ausgabe:

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

> **Note**
>
> Perzentil-Statistiken für Metriken sind nicht verfügbar, wenn es Metrik-Werte gibt, die negative Zahlen enthalten.

Anzeigen der Details von benutzerdefinierten Metriken

Das folgende Verfahren zeigt Ihnen, wie Sie die Details einer benutzerdefinierten Metrik in der CLI anzeigen können.

- Verwenden Sie den Befehl `list-custom-metrics`, um alle Ihre benutzerdefinierten Metriken anzuzeigen.

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

Die Ausgabe dieses Befehls sieht wie folgt aus:

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

Aktualisieren einer benutzerdefinierten Metrik

Das folgende Verfahren zeigt Ihnen, wie Sie eine benutzerdefinierte Metrik in der CLI aktualisieren können.

- Verwenden Sie den Befehl `update-custom-metric`, um eine benutzerdefinierte Metrik zu aktualisieren. Im folgenden Beispiel wird der `display-name` aktualisiert.

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

Die Ausgabe dieses Befehls sieht wie folgt aus:

```
{
    "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"
}
```

Löschen einer benutzerdefinierten Metrik

Das folgende Verfahren zeigt Ihnen, wie Sie eine benutzerdefinierte Metrik aus der CLI löschen können.

1. Um eine benutzerdefinierte Metrik zu löschen, entfernen Sie sie zunächst aus allen Sicherheitsprofilen, denen sie angefügt ist. Mit dem Befehl `list-security-profiles` können Sie Sicherheitsprofile mit einer bestimmten Dimension anzeigen.
2. Mit dem Befehl `update-security-profiles` können Sie eine Dimension aus einem Sicherheitsprofil entfernen. Geben Sie alle Informationen ein, die Sie behalten möchten, jedoch nicht die Dimension.

   ```
   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}}]"
   ```

   Die Ausgabe dieses Befehls sieht wie folgt aus:

   ```
   {
     "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. Nachdem die benutzerdefinierte Metrik getrennt wurde, verwenden Sie den Befehl `delete-custom-metric`, um die benutzerdefinierte Metrik zu löschen.

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

Die Ausgabe dieses Befehls sieht wie folgt aus:

```
HTTP 200
```

CLI-Befehle für benutzerdefinierte Metriken

Mit den folgenden CLI-Befehlen können Sie benutzerdefinierte Metriken erstellen und verwalten.

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

Benutzerdefinierte Metriken-APIs

Die folgenden APIs können verwendet werden, um benutzerdefinierte Metriken zu erstellen und zu verwalten.

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

Geräteseitige Metriken

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

Bei der Erstellung eines Sicherheitsprofils können Sie das erwartete Verhalten Ihres IoT-Geräts festlegen, indem Sie Verhalten und Schwellenwerte für von IoT-Geräten generierte Metriken konfigurieren. Bei den folgenden Kennzahlen handelt es sich um geräteseitige Metriken, bei denen es sich um Messwerte von Agenten handelt, die Sie auf Ihren Geräten installieren.

Ausgehende Bytes (**`aws:all-bytes-out`**)

Die Anzahl der ausgehenden Bytes von einem Gerät während eines bestimmten Zeitraums.

Mit dieser Metrik geben Sie die maximale oder minimale Menge des ausgehenden Datenverkehrs an, die ein Gerät in einem bestimmten Zeitraum senden soll, gemessen in Byte.

Kompatibel mit: Rules Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: Byte

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{
  "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 Beispiel mit ML Detect

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

Bytes in (**`aws:all-bytes-in`**)

Die Anzahl der eingehenden Bytes zu einem Gerät während eines bestimmten Zeitraums.

Mit dieser Metrik geben Sie die maximale oder minimale Menge an eingehendem Datenverkehr an, die ein Gerät in einem bestimmten Zeitraum empfangen soll, gemessen in Byte.

Kompatibel mit: Rules Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: Byte

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{
  "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 Beispiel mit ML Detect

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

Überwachen der Anzahl an TCP-Ports (**`aws:num-listening-tcp-ports`**)

Die Anzahl der TCP-Ports, die das Gerät überwacht.

Mit diesem Parameter geben Sie die maximale oder minimale Anzahl von TCP-Ports an, die jedes Gerät überwachen soll.

Kompatibel mit: Rules Detect | ML Detect

Einheit: Fehler

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheit: Fehler

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{
  "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 Beispiel mit ML Detect

```
{
  "name": "Max TCP Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
```

```
}
```

Überwachen der Anzahl an UDP-Ports (**aws:num-listening-udp-ports**)

Die Anzahl der UDP-Ports, die das Gerät überwacht.

Mit diesem Parameter geben Sie die maximale Anzahl an UDP-Ports an, die jedes Gerät überwachen soll.

Kompatibel mit: Rules Detect | ML Detect

Einheit: Fehler

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheit: Fehler

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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 Beispiel für die Nutzung von **statisticalThreshold**

```
{
  "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 Beispiel mit ML Detect

```
{
  "name": "Max UPD Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

Ausgehende Pakete (**`aws:all-packets-out`**)

Die Anzahl der ausgehenden Pakete von einem Gerät während eines bestimmten Zeitraums.

Mit dieser Metrik geben Sie die maximale oder minimale Menge an gesamtem ausgehendem Datenverkehr an, die ein Gerät in einem bestimmten Zeitraum senden soll.

Kompatibel mit: Rules Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: Pakete

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{
  "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 Beispiel mit ML Detect

```
{
  "name": "Outbound sent ML behavior",
  "metric": "aws:all-packets-out",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
```

```
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Pakete in (**`aws:all-packets-in`**)

Die Anzahl der eingehenden Pakete zu einem Gerät während eines bestimmten Zeitraums.

Mit dieser Metrik geben Sie die maximale oder minimale Menge an insgesamt eingehendem Datenverkehr an, die ein Gerät in einem bestimmten Zeitraum empfangen soll.

Kompatibel mit: Rule Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: Pakete

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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

Beispiel für die Nutzung von `statisticalThreshold`

```
{
  "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 Beispiel mit ML Detect

```
{
  "name": "Inbound sent ML behavior",
  "metric": "aws:all-packets-in",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Ziel-IPs (**`aws:destination-ip-addresses`**)

Ein Satz von IP-Zieladressen.

Mit dieser Metrik geben Sie einen Satz von zugelassenen (vormals als auf einer Whitelist stehend bezeichnet) bzw. verbotenen (vormals als auf einer Blacklist stehend bezeichnet) Classless Inter-Domain Routings (CIDR) an, von denen jedes Gerät eine Verbindung mit AWS IoT herstellen darf bzw. keine Verbindung herstellen darf.

Kompatibel mit: Rules Detect

Operatoren: in-cidr-set | not-in-cidr-set

Werte: eine Liste von CIDRs

Einheiten: –

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
}
```

Überwachen von TCP-Ports (**`aws:listening-tcp-ports`**)

Die TCP-Ports, die das Gerät überwacht.

Mit dieser Metrik geben Sie einen Satz von zugelassenen (vormals als auf einer Whitelist stehend bezeichnet) bzw. verbotenen (vormals als auf einer Blacklist stehend bezeichnet) TCP-Ports an, auf denen jedes Gerät horchen darf bzw. nicht horchen darf.

Kompatibel mit: Rules Detect

Operatoren: in-port-set | not-in-port-set

Werte: eine Liste von Ports

Einheiten: –

Example

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 443, 80 ]
    }
```

```
  },
  "suppressAlerts": true
}
```

Überwachen von UDP-Ports (**aws:listening-udp-ports**)

Die UDP-Ports, die das Gerät überwacht.

Mit dieser Metrik geben Sie einen Satz von zugelassenen (vormals als auf einer Whitelist stehend bezeichnet) bzw. verbotenen (vormals als auf einer Blacklist stehend bezeichnet) UDP-Ports an, auf denen jedes Gerät horchen darf bzw. nicht horchen darf.

Kompatibel mit: Rules Detect

Operatoren: in-port-set | not-in-port-set

Werte: eine Liste von Ports

Einheiten: –

Example

```
{
  "name": "Listening UDP Ports",
  "metric": "aws:listening-udp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 1025, 2000 ]
    }
  }
}
```

Anzahl etablierter TCP-Verbindungen (**aws:num-established-tcp-connections**)

Die Anzahl der TCP-Verbindungen für ein Gerät.

Mit diesem Parameter geben Sie die maximale oder minimale Anzahl aktiver TCP-Verbindungen an, die jedes Gerät haben sollte (alle TCP-Status).

Kompatibel mit: Rules Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: Verbindungen

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{
  "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 Beispiel mit ML Detect

```
{
  "name": "Connection count ML behavior",
```

```
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Spezifikationen für Gerätemetriken

Gesamtstruktur

Langer Name	Kurzname	Erforderlich	Typ	Einschränkungen	Hinweise
header	hed	Y	Objekt		Vollständiger Block für gültiges Berichtsformat erforderlich.
Metriken	met	Y	Objekt		Ein Bericht kann mindestens einen `metrics`- oder `custom_metrics` -Block oder beide enthalten.
custom_metrics	comet	Y	Objekt		Ein Bericht kann mindesten

Langer Name	Kurzname	Erforderlich	Typ	Einschränkungen	Hinweise
					s einen – `metrics`oder `custom_metrics` - Block oder beide enthalten.

Header-Block

Langer Name	Kurzname	Erforderlich	Typ	Einschränkungen	Hinweise
report_id	rid	Y	Ganzzahl		Der Wert wird monotonisch erhöht. Ein Epoche-Zeitstempel wird empfohlen.
version	V	Y	Zeichenfolge	Major.Minor	Kleine Schritte mit dem Hinzufügen eines Feldes. Große Schritte, wenn Metriken entfernt werden.

Metriken-Block:

TCP-Verbindungen

Langer Name	Kurzname	Übergeordnetes Element	Erforderlich	Typ	Einschränkungen	Hinweise
tcp_connections	tc	Metriken	N	Objekt		
established_connections	ec	tcp_connections	N	Objekt		Eingerichtete TCP-Status-
Verbindungen	cs	established_connections	N	List<Object>		
remote_addr	rad	Verbindungen	Y	Zahl	ip:port	IP kann IPv6 oder IPv4 sein
local_port	lp	Verbindungen	N	Zahl	>= 0	
local_interface	li	Verbindungen	N	Zeichenfolge		Schnittstellenname
total	t	established_connections	N	Zahl	>= 0	Anzahl der eingerichteten Verbindungen

Überwachen von TCP-Ports

Langer Name	Kurzname	Übergeordnetes Element	Erforderlich	Typ	Einschränkungen	Hinweise
listening_tcp_ports	tp	Metriken	N	Objekt		
ports	pts	listening_tcp_ports	N	List<Object>	> 0	
port	pt	ports	N	Zahl	> 0	Ports sollten Zahlen größer 0 sein
interface	if	ports	N	Zeichenfolge		Schnittstellenname
total	t	listening_tcp_ports	N	Zahl	>= 0	

Überwachen von UDP-Ports

Langer Name	Kurzname	Übergeordnetes Element	Erforderlich	Typ	Einschränkungen	Hinweise
listening_udp_ports	up	Metriken	N	Objekt		
ports	pts	listening_udp_ports	N	Liste<Port>	> 0	
port	pt	ports	N	Zahl	> 0	Ports sollten Zahlen

Langer Name	Kurzname	Übergeordnetes Element	Erforderlich	Typ	Einschränkungen	Hinweise
						größer 0 sein
interface	if	ports	N	Zeichenfolge		Schnittstellenname
total	t	listening_udp_ports	N	Zahl	>= 0	

Netzwerkstatistik

Langer Name	Kurzname	Übergeordnetes Element	Erforderlich	Typ	Einschränkungen	Hinweise
network_stats	ns	Metriken	N	Objekt		
bytes_in	bi	network_stats	N	Zahl	Delta Metric, >= 0	
bytes_out	bo	network_stats	N	Zahl	Delta Metric, >= 0	
packets_in	pi	network_stats	N	Zahl	Delta Metric, >= 0	
packets_out	po	network_stats	N	Zahl	Delta Metric, >= 0	

Example

Die folgende JSON-Struktur verwendet lange Namen.

```
{
  "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 Beispiel für eine JSON-Struktur mit kurzen Namen

```
{
  "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"
        ]
      }
    ]
  }
}
```

Senden von Metriken von Geräten

AWS IoT Device Defender Detect kann von AWS IoT-Geräten generierte Metrikdaten sammeln, aggregieren und überwachen, um Geräte zu identifizieren, die ein ungewöhnliches Verhalten zeigen. In diesem Abschnitt erfahren Sie, wie Sie Metriken von einem Gerät an senden AWS IoT Device Defender.

Sie müssen auf Ihren über AWS IoT verbundenen Geräten oder Geräte-Gateways auf sichere Weise die AWS IoT SDK Version 2 zum Erfassen von geräteseitigen Metriken bereitstellen. Die vollständige Liste der SDKs finden Sie hier.

Sie können den AWS IoT Geräteclient zum Veröffentlichen von Metriken verwenden, da er einen einzigen Agent bietet, der alle Funktionen des Gerätemanagements von AWS IoT Device Defender und AWS IoT abdeckt. Zu diesen Funktionen gehören Jobs, sicheres Tunneling, die Veröffentlichung von AWS IoT Device Defender Metriken uvm.

Sie veröffentlichen geräteseitige Metriken zum Erfassen und Evaluieren unter dem reservierten Thema in AWS IoT für AWS IoT Device Defender.

Verwenden des AWS IoT Geräteclients zum Veröffentlichen von Metriken

Um den AWS IoT Geräteclient zu installieren, können Sie ihn von Github herunterladen. Nachdem Sie den AWS IoT Geräteclient auf dem Gerät installiert haben, für das Sie geräteseitige Daten

sammeln möchten, müssen Sie ihn so konfigurieren, dass er geräteseitige Messdaten an AWS IoT Device Defender sendet. Stellen Sie sicher, dass in der Konfigurationsdatei des AWS IoT Geräteclients die folgenden Parameter im Abschnitt `device-defender` festgelegt sind:

```
"device-defender":    {
      "enabled":    true,
      "interval-in-seconds": 300
  }
```

> **Warning**
>
> Sie sollten das Zeitintervall auf mindestens 300 Sekunden festlegen. Wenn Sie das Zeitintervall auf weniger als 300 Sekunden festlegen, werden Ihre Metrikdaten möglicherweise gedrosselt.

Nachdem Sie Ihre Konfiguration aktualisiert haben, können Sie auf der AWS IoT Device Defender Konsole Sicherheitsprofile und Sicherheitsverhalten erstellen, um die Messwerte zu überwachen, die Ihre Geräte in der Cloud veröffentlichen. Sie finden veröffentlichte Metriken auf der AWS IoT Core-Konsole, indem Sie Defend, Detect und dann Metrics auswählen.

Cloudseitige Metriken

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

Bei der Erstellung eines Sicherheitsprofils können Sie das erwartete Verhalten Ihres IoT-Geräts festlegen, indem Sie Verhalten und Schwellenwerte für von IoT-Geräten generierte Metriken konfigurieren. Bei den folgenden Kennzahlen handelt es sich um cloudseitige Metriken, bei denen es sich um Metriken von handelt AWS IoT.

Nachrichtengröße (aws:message-byte-size)

Die Anzahl der Bytes in einer Nachricht. Mit dieser Metrik geben Sie die maximale oder minimale Größe (in Byte) der einzelnen von einem Gerät an übertragenen Nachrichten an AWS IoT.

Kompatibel mit: Rules Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: Byte

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{

  "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 Beispiel mit ML Detect

```
{
  "name": "Message size ML behavior",
  "metric": "aws:message-byte-size",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

Für ein Gerät wird ein Alarm ausgegeben, wenn es während drei aufeinanderfolgender Fünf-Minuten-Zeiträume Nachrichten überträgt, deren kumulative Größe größer ist als bei 90 % aller anderen Geräte, die dieses Sicherheitsprofilverhalten protokollieren.

Gesendete Nachrichten (aws:num-messages-sent)

Die Anzahl der Nachrichten, die von einem Gerät während eines bestimmten Zeitraums gesendet werden.

Mit dieser Metrik geben Sie die maximale oder minimale Anzahl von Nachrichten an, die während eines bestimmten Zeitraums zwischen AWS IoT und jedem Gerät gesendet werden können.

Kompatibel mit: Rules Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: Nachrichten

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{

  "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 Beispiel mit ML Detect

```
{
  "name": "Messages sent ML behavior",
  "metric": "aws:num-messages-sent",
  "criteria": {
```

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

Empfangene Nachrichten (aws:num-messages-received)

Die Anzahl der Nachrichten, die von einem Gerät während eines bestimmten Zeitraums gesendet werden.

Mit dieser Metrik geben Sie die maximale oder minimale Anzahl von Nachrichten an, die während eines bestimmten Zeitraums zwischen AWS IoT und jedem Gerät empfangen werden können.

Kompatibel mit: Rules Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: Nachrichten

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{
  "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 Beispiel mit ML Detect

```
{
  "name": "Messages received ML behavior",
  "metric": "aws:num-messages-received",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Autorisierungsfehler (aws:num-authorization-failures)

Mit diesem Parameter geben Sie die maximale Anzahl an Autorisierungsfehlern an, die für jedes Gerät in einem bestimmten Zeitraum zulässig ist. Ein Autorisierungsfehler tritt auf, wenn eine Anforderung von einem Gerät an AWS IoT abgelehnt wird, z. B. wenn ein Gerät versucht, ein Thema zu veröffentlichen, für das es nicht über ausreichende Berechtigungen verfügt.

Kompatibel mit: Rules Detect | ML Detect

Einheit: Fehler

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{
  "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 Beispiel mit ML Detect

```
{
```

```
  "name": "Authorization failures ML behavior",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Quell-IP (aws:source-ip-address)

Die IP-Adresse, von der ein Gerät eine Verbindung zu AWS IoT hergestellt hat.

Mit dieser Metrik geben Sie einen Satz von zugelassenen (vormals als auf einer Whitelist stehend bezeichnet) bzw. verbotenen (vormals als auf einer Blacklist stehend bezeichnet) Classless Inter-Domain Routings (CIDR) an, von denen jedes Gerät eine Verbindung mit herstellen darf bzw. keine Verbindung herstellen darf AWS IoT.

Kompatibel mit: Rules Detect

Operatoren: in-cidr-set | not-in-cidr-set

Werte: eine Liste von CIDRs

Einheiten: –

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
}
```

Verbindungsversuche (aws:num-connection-attempts)

Gibt an, wie oft ein Gerät versucht, innerhalb eines bestimmten Zeitraums eine Verbindung herzustellen.

Mit dieser Metrik geben Sie die Höchst- oder Mindestanzahl an Verbindungsversuchen für jedes Gerät an. Erfolgreiche und fehlgeschlagene Versuche werden gezählt.

Kompatibel mit: Rules Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: Verbindungsversuche

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

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 Beispiel für die Nutzung von **`statisticalThreshold`**

```
{
  "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 Beispiel mit ML Detect

```
{
  "name": "Connection attempts ML behavior",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": false
}
```

Unterbricht die Verbindung (aws:num-disconnects)

Gibt an, wie oft die Verbindung eines Geräts mit AWS IoT innerhalb eines bestimmten Zeitraums getrennt wird.

Mit dieser Metrik geben Sie die Höchst- oder Mindestanzahl der getrennten Verbindungen eines Geräts mit AWS IoT innerhalb eines bestimmten Zeitraums an.

Kompatibel mit: Rules Detect | ML Detect

Operatoren: less-than | less-than-equals | greater-than | greater-than-equals

Wert: eine nicht negative Ganzzahl

Einheiten: getrennte Verbindungen

Dauer: eine nicht negative Ganzzahl. Gültige Werte sind 300, 600, 900, 1800 oder 3600 Sekunden.

Example

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Beispiel für die Nutzung von **`statisticalThreshold`**

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Beispiel mit ML Detect

```
{
  "name": "Disconnects ML behavior",
  "metric": "aws:num-disconnects",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
```

```
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Dauer der Verbindung (aws:disconnect-duration)

Die Dauer, für die ein Gerät von getrennt bleibt AWS IoT.

Verwenden Sie diese Metrik, um die maximale Dauer anzugeben, für die ein Gerät von getrennt bleibt AWS IoT.

Kompatibel mit: Rules Detect

Operatoren: weniger als | weniger als gleich

Wert: eine nicht negative Ganzzahl (in Minuten)

Example

```
{
"name": "DisconnectDuration",
  "metric": "aws:disconnect-duration",
  "criteria": {
"comparisonOperator": "less-than-equals",
    "value": {
"count": 5
    }
  },
  "suppressAlerts": true
}
```

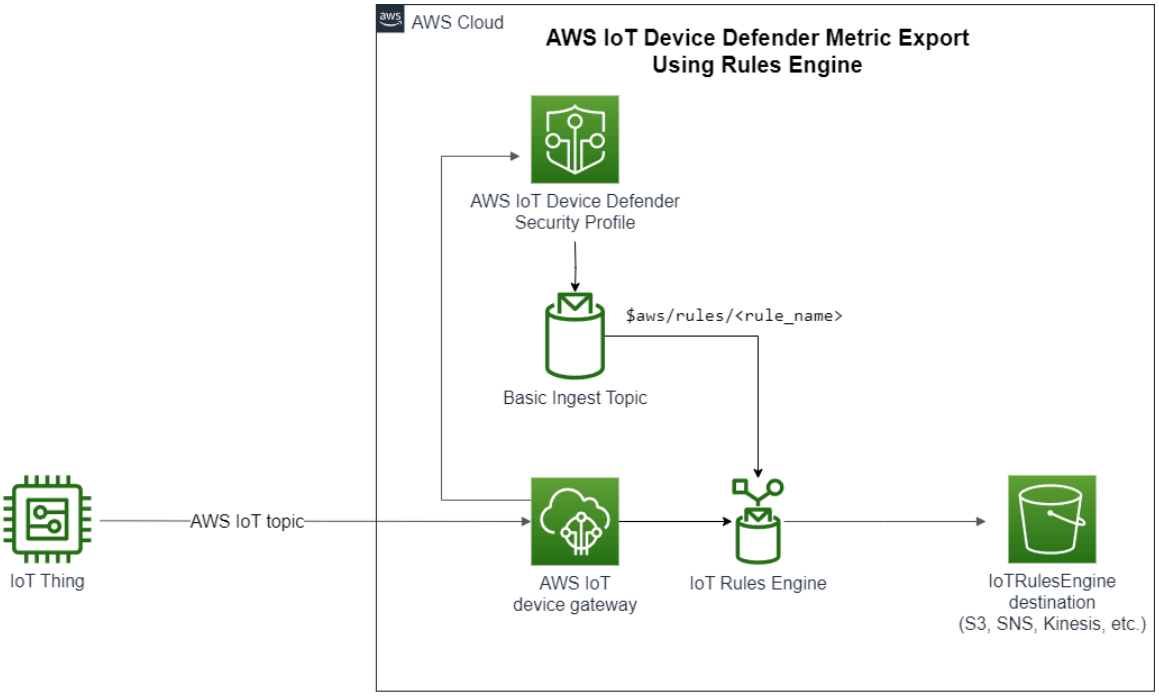
Detect-Metrikexport

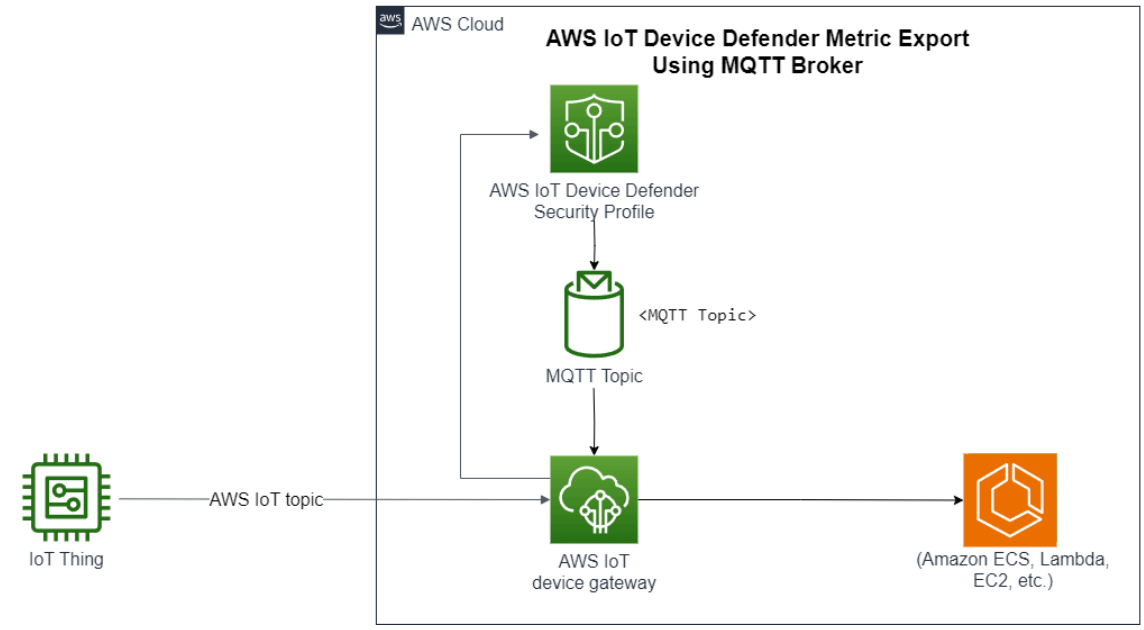
> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect

finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

Mit dem Metrikexport können Sie cloudseitige, geräteseitige oder benutzerdefinierte Metriken aus AWS IoT Device Defender exportieren und sie in einem von Ihnen konfigurierten MQTT-Thema veröffentlichen. Dieses Feature unterstützt den Massenexport von Detect-Metriken, was nicht nur eine effizientere Datenberichterstattung und -analyse ermöglicht, sondern auch zur Kostenkontrolle beiträgt. Sie können Ihr MQTT-Thema als grundlegendes Thema der Erfassung von AWS IoT-Regeln auswählen oder Ihr eigenes MQTT-Thema erstellen und abonnieren. Konfigurieren Sie einen Metrikexport über die AWS IoT Device Defender-Konsole, die API oder die CLI. Diese Funktion ist in allen AWS Regionen verfügbar, in denen AWS IoT Device Defender verfügbar ist.

Die folgende Abbildung zeigt, wie Sie AWS IoT Device Defender für den Export von Metriken konfigurieren können. Das erste Diagramm zeigt, wie Sie den Metrikexport konfigurieren, um Metriken zu einem grundlegenden Thema der Erfassung zu exportieren. Anschließend können Sie die exportierten Metriken an verschiedene Ziele weiterleiten, die von AWS IoTRegeln unterstützt werden. Das zweite Diagramm zeigt, wie Sie AWS IoT Device Defender so konfigurieren, dass Daten in einem MQTT-Thema veröffentlicht werden. Der Client abonniert anschließend dieses MQTT-Thema: Sie können einen MQTT-Client in einem Container auf Amazon Elastic Container Service, Lambda oder einer Amazon EC2 EC2-Instance ausführen, die dasselbe MQTT-Thema abonniert. Wann immer AWS IoT Device Defender Daten veröffentlicht, werden sie vom MQTT-Client empfangen und verarbeitet. Weitere Informationen finden Sie unter MQTT-Themen.





So funktioniert der Detect-Metrikenexport

Wenn Sie ein Sicherheitsprofil einrichten, wählen Sie die Metriken für den Export aus und geben das MQTT-Thema an. Sie konfigurieren auch eine IAM-Rolle, die AWS IoT Device Defender Detect die erforderlichen Berechtigungen zum Veröffentlichen von Nachrichten im konfigurierten MQTT-Thema gewährt. Sie können ein AWS IoT Rules Basic Ingest MQTT-Thema konfigurieren und die

exportierten Metriken an Ziele senden, die von AWS IoT Rules unterstützt werden. Ausführliche Anweisungen zum Einrichten und Konfigurieren von AWS IoT-Regeln finden Sie in den Regeln für AWS IoT im AWS IoT-Entwicklerhandbuch.

AWS IoT Device Defender stapelt Metrikwerte für jede konfigurierte Metrik und veröffentlicht sie in regelmäßigen Abständen im konfigurierten MQTT-Thema. Mit Ausnahme der Nachrichten-Bytegröße und der Gesamtbytegröße werden cloudseitige Metriken durch Summieren von Metrikwerten für die Batchdauer aggregiert. Benutzerdefinierte und geräteseitige Metriken werden nicht aggregiert. Bei der Nachrichten-Bytegröße handelt es sich bei den Exportwerten um die Mindest-, Höchst- und Gesamtbytegröße für die Batchdauer. Für die Dauer der Verbindung entspricht der Exportwert der Dauer der Unterbrechung – in Sekunden – für alle verfolgten Geräte. Dies tritt in einstündigen Intervallen sowie bei Verbindungen oder Verbindungsabbrüchen auf. Bei verbundenen Geräten oder Verbindungsereignissen ist der Wert null. Weitere Informationen zu cloudseitigen Metriken, geräteseitigen Metriken und benutzerdefinierten Metriken finden Sie im AWS IoT Device Defender-Entwicklerhandbuch:

- Benutzerdefinierte Metriken
- Cloudseitige Metriken
- Geräteseitige Metriken

Sie können Batchmetriken mit AWS IoT-Regeln an verschiedene Ziele exportieren. Eine Liste der unterstützten Ziele finden Sie unter AWS IoT-Regelaktionen. Verwenden Sie die batchMode-Option für AWS IoT-Regelaktionen, um einzelne Metriken innerhalb einer Batch-Exportnachricht an ein unterstütztes Ziel zu senden. Wenn Ihr bevorzugtes Ziel für AWS IoT-Regeln keine `batchMode`-Unterstützung bietet, können Sie trotzdem einzelne Metriken innerhalb einer Batchnachricht senden, indem Sie Zwischenaktionen wie Lambda oder Kinesis Data Streams verwenden.

Schema zum Exportieren von Metriken

Im folgenden Schema finden Sie Informationen zum Export von Metriken im Batchformat.

```
{
 "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}"
 }
 }
 ]
}
```

Preisgestaltung für den Detect-Metrikexport

Wenn Sie cloudseitige, geräteseitige oder benutzerdefinierte Metriken zu einem von Ihnen konfigurierten MQTT-Thema veröffentlichen, fallen für diesen Schritt des Exportprozesses keine Gebühren an. In den nachfolgenden Schritten, wenn Sie die veröffentlichten Metriken mithilfe der Rules Engine oder von Messaging an ein Ziel Ihrer Wahl übertragen, fallen jedoch Kosten an, die auf der von Ihnen gewählten Übertragungsmethode basieren. AWS IoT Device Defender veröffentlicht

gebündelte Metriken zu MQTT-Themen als einzelne Nachricht, die Metrikdaten für mehrere Geräte enthält, was zur Kostensenkung beiträgt. Weitere Informationen zur Preisgestaltung finden Sie im AWSPreisrechner.

Berechtigungen

Dieser Abschnitt enthält Informationen darüber, wie die IAM-Rollen und -Richtlinien eingerichtet werden, die zum Verwalten des AWS IoT Device Defender Detect-Metrikenexports erforderlich sind. Weitere Informationen finden Sie im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zum Veröffentlichen von Nachrichten in einem MQTT-Thema an AWS IoT Device Defender Detect

Wenn Sie den Parameter in CreateSecurityProfile verwenden, müssen Sie eine IAM-Rolle mit zwei Richtlinien angeben: einer Berechtigungsrichtlinie und einer Vertrauensrichtlinie. Die Berechtigungsrichtlinie gewährt AWS IoT Device Defender die Berechtigung, Nachrichten zu veröffentlichen, die Metriken zu einem MQTT-Thema enthalten. Die Vertrauensrichtlinie erteilt AWS IoT Device Defender die Berechtigung zur Übernahme der erforderlichen Rolle.

Berechtigungsrichtlinie

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/your-topic-name"
            ]
        }
    ]
}
```

Vertrauensrichtlinie

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Richtlinie zum Übergeben von Rollen

Sie benötigen auch eine IAM-Berechtigungsrichtlinie, die dem IAM-Benutzer zugeordnet ist und es diesem ermöglicht, Rollen zu übergeben. Beachten Sie hierzu den Abschnitt Gewähren von Berechtigungen, mit denen ein Benutzer eine Rolle an einen AWS-Service übergeben kann.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::123456789012:role/Role_To_Pass"
        }
    ]
}
```

Einrichten des Detect-Metrikenexports auf der AWS IoT-Konsole

Sie können ein neues Sicherheitsprofil erstellen, anzeigen und bearbeiten, das den Metrikexport auf der Konsole beinhaltet.

Voraussetzungen

Bevor Sie den Detect-Metrikexport einrichten, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Eine IAM-Rolle. Weitere Informationen zum Erstellen von IAM-Rollen finden Sie unter Erstellen von IAM-Rollen im IAM-Benutzerhandbuch.
- Ein AWS-Konto, bei dem Sie sich als AWS Identity and Access ManagementIAM-Benutzer mit entsprechenden Berechtigungen anmelden können. Weitere Informationen zu AWS IoT Device Defender-Detect-Berechtigungen finden Sie unter Berechtigungen im AWS IoT Core-Entwicklerhandbuch.

Erstellen eines neuen Sicherheitsprofils, das den Metrikexport beinhaltet (Konsole)

Konfigurieren Sie zum Exportieren von Daten zum Verhalten von Metriken zunächst ein Sicherheitsprofil, das den Metrikexport beinhaltet. Im folgenden Verfahren wird beschrieben, wie Sie ein regelbasiertes Sicherheitsprofil einrichten, das den Export von Detect-Metriken beinhaltet.

Erstellen eines neuen Sicherheitsprofils, das den Metrikexport beinhaltet

1. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Erkennen und Sicherheitsprofile.
2. Wählen Sie unter Sicherheitsprofil erstellen die Option Regelbasiertes Erkennungsprofil für Anomalien erstellen.
3. Geben Sie den Namen Ihres Sicherheitsprofils ein, um die Eigenschaften Ihres Sicherheitsprofils anzugeben, und wählen Sie unter Ziel eine Gruppe von Geräten, die wegen Anomalien analysiert werden soll. (Optional) Fügen Sie eine Beschreibung und Tags hinzu, um AWS-Ressourcen zu kennzeichnen. Wählen Sie Next.
4. Wählen Sie für Metrik die Metriken aus, um das Geräteverhalten zu definieren. Sie können den Schwellenwert für das Verhalten definieren, um eine Warnung zu erhalten, wenn Ihr Gerät die Verhaltenserwartungen nicht erfüllt.

5. Wählen Sie Eine Warnung senden (metrisches Verhalten definieren), um Warnungen für Verhaltensanomalien zu erhalten, und geben Sie dann den Namen des Verhaltens und die Bedingungen an. Wählen Sie Keine Warnung senden (Metrik beibehalten), um die Metriken ohne Warnungen beizubehalten. Wählen Sie Weiter aus.
6. Wählen Sie Metrikexport einschalten, um die Konfigurationen für den Export von Metriken zu deaktivieren.
7. Geben Sie einen Namen für das MQTT-Thema ein, der für die Veröffentlichung Ihrer Metrikdaten in verwendet werden soll AWS IoT Core. Wählen Sie eine IAM-Rolle, um AWS IoT die Berechtigung „AWS IoT:Publish“ zum Veröffentlichen von Nachrichten zum konfigurierten Thema zu erteilen. Wählen Sie die Metriken aus, die Sie exportieren möchten, und klicken Sie dann auf Weiter.

   > **Note**
   >
   > Verwenden Sie den Schrägstrich, um bei der Eingabe Ihres MQTT-Themas hierarchische Informationen darzustellen. Beispiel, `$AWS/rules/rule-name/`.

8. Wählen oder erstellen Sie ein Amazon-SNS-Thema und eine IAM-Rolle, um die Amazon-SNS-Benachrichtigungen so zu konfigurieren, dass Benachrichtigungen an Ihre AWS-Konsole gesendet werden, wenn ein Gerät gegen ein festgelegtes Verhalten verstößt. Wählen Sie Next.
9. Überprüfen Sie Ihre Konfigurationen, und wählen Sie dann Weiter.

Anzeigen und Bearbeiten von Sicherheitsprofildetails (Konsole)

Anzeigen und Bearbeiten von Sicherheitsprofildetails

1. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Erkennen und Sicherheitsprofile.
2. Wählen Sie das Sicherheitsprofil aus, das Sie für den Metrikexport erstellt haben, und wählen Sie dann für Aktionen die Option Bearbeiten.
3. Wählen Sie unter Ziel die Zielgerätegruppen aus, die Sie bearbeiten möchten, und klicken Sie dann auf Weiter.
4. Wählen Sie Mich benachrichtigen (metrisches Verhalten definieren), um die Konfigurationen des metrischen Verhaltens zu bearbeiten, und definieren Sie dann die Bedingungen, unter denen das metrische Verhalten erfüllt ist. Wählen Sie Next.

5. Wählen Sie Export von Metriken deaktivieren, um die Konfigurationen für den Export von Metriken zu deaktivieren. Wählen Sie Next.
6. Wählen oder erstellen Sie ein Amazon-SNS-Thema und eine IAM-Rolle, um die Amazon-SNS-Benachrichtigungen so zu konfigurieren, dass Benachrichtigungen an Ihre AWS IoT-Konsole gesendet werden, wenn ein Gerät gegen ein festgelegtes Verhalten verstößt. Wählen Sie Next.
7. Überprüfen Sie Ihre Konfigurationen, und wählen Sie dann Weiter.

Erstellen eines Sicherheitsprofils zum Aktivieren des Metrikexports

Verwenden Sie den Befehl `create-security-profile`, um Ihr Sicherheitsprofil zu erstellen und den Metrikexport zu aktivieren.

Erstellen eines Sicherheitsprofils, das den Metrikexport beinhaltet

1. Setzen Sie den Wert `exportMetric` bei `Behavior` und `AdditionalMetricsToRetainV2` auf „war“, um den Metrikexport zu aktivieren und anzugeben, ob Detect die entsprechenden Metriken exportieren soll.
2. Fügen Sie den Wert für `MetricsExportConfig` ein. Gibt das MQTT-Thema und den Rollen-ARN (Amazon Resource Name) an, die für den Metrikexport erforderlich sind.

> **Note**
>
> Schließen Sie `mqttTopic` ein, damit AWS IoT Device Defender Detect Nachrichten veröffentlichen kann. Die Rollen-ARN hat die Berechtigung, MQTT-Nachrichten zu veröffentlichen. Danach kann AWS IoT Device Defender Detect die Rolle übernehmen und Nachrichten in Ihrem Namen veröffentlichen.

```
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
```

Ausgabe:

```
{
    "securityProfileName": "CreateSecurityProfileWithMetricsExport",
    "securityProfileArn": "arn:aws:iot:us-east-1:123456789012:securityprofile/
CreateSecurityProfileWithMetricsExport"
}
```

Erstellen eines Sicherheitsprofils zum Aktivieren des Metrikexports (CLI)

Verwenden Sie den `update-security-profile`-Befehl, um ein vorhandenes Sicherheitsprofil zu aktualisieren und den Metrikexport zu aktivieren.

Aktualisieren Sie ein Sicherheitsprofil, um den Metrikexport zu aktivieren.

1. Setzen Sie den Wert `exportMetric` bei `Behavior` und `AdditionalMetricsToRetainV2` auf „war“, um den Metrikexport zu aktivieren und anzugeben, ob Detect die entsprechenden Metriken exportieren soll.
2. Fügen Sie den Wert für `MetricsExportConfig` ein. Gibt das MQTT-Thema und den Rollen-ARN (Amazon Resource Name) an, die für den Metrikexport erforderlich sind.

> **Note**
>
> Schließen Sie `mqttTopic` ein, damit AWS IoT Device Defender Detect Nachrichten veröffentlichen kann. Die Rollen-ARN hat die Berechtigung, MQTT-Nachrichten zu veröffentlichen. Danach kann AWS IoT Device Defender Detect die Rolle übernehmen und Nachrichten in Ihrem Namen veröffentlichen.

```
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
```

Ausgabe:

```
{
    "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"
    }
}
```

Aktualisieren eines Sicherheitsprofils zum Deaktivieren des Metrikexports (CLI)

Verwenden Sie den `update-security-profile`-Befehl, um ein vorhandenes Sicherheitsprofil zu aktualisieren und den Metrikexport zu deaktivieren.

Aktualisieren eines Sicherheitsprofils zum Deaktivieren des Metrikexports (CLI)

- Verwenden Sie den Befehl `--delete-metrics-export-config`, um Ihr Sicherheitsprofil zu aktualisieren und die Konfiguration für den Metrikexport zu entfernen.

```
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
```

Ausgabe:

```
{
    "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"
}
```

Weitere Informationen finden Sie unter Detect-Befehle im AWS IoT Entwicklerhandbuch.

CLI-Befehle für den Export von Metriken

Mit den folgenden CLI-Befehlen können Sie einen Detect-Metrikenexport erstellen und verwalten.

- CreateSecurityProfile
- UpdateSecurityProfile
- DescribeSecurityProfile

API-Operationen für den Metrikexport

Mit den folgenden API-Operationen können Sie einen Detect-Metrikexport erstellen und verwalten.

- CreateSecurityProfile
- UpdateSecurityProfile
- DescribeSecurityProfile

Bereichsbestimmung für Metriken in Sicherheitsprofilen mithilfe von Dimensionen

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

Dimensionen sind Attribute, die Sie definieren können, um genauere Daten über Metriken und Verhaltensweisen in Ihrem Sicherheitsprofil zu erhalten. Sie definieren den Bereich durch Angabe eines Werts oder Musters, der/das als Filter verwendet wird. Sie können beispielsweise eine Themenfilterdimension definieren, die eine Metrik nur auf MQTT-Themen anwendet, die einem bestimmten Wert entsprechen, wie „data/bulb/+/activity“. Informationen zum Definieren einer Dimension, die Sie in Ihrem Sicherheitsprofil verwenden können, finden Sie unter CreateDimension.

Dimensionswerte unterstützen MQTT-Platzhalter. Mithilfe von MQTT-Platzhaltern können Sie mehrere Themen gleichzeitig abonnieren. Es gibt zwei verschiedene Arten von Platzhaltern: einstufige (+) und mehrstufige (#). Der Dimensionswert `Data/bulb/+/activity` beispielsweise erstellt ein Abonnement, das allen Themen entspricht, die auf derselben Ebene wie + vorhanden sind. Dimensionswerte unterstützen auch die MQTT-Client-ID-Ersetzungsvariable ${iot:ClientId}.

Dimensionen des Typs TOPIC_FILTER sind mit den folgenden cloudseitigen Metriken kompatibel:

- Anzahl der Autorisierungsfehler
- Nachrichten-Bytegröße
- Anzahl der empfangenen Nachrichten
- Anzahl der gesendeten Nachrichten
- Quell-IP-Adresse (nur für Rules Detect verfügbar)

So verwenden Sie Dimensionen in der Konsole

So erstellen Sie eine Dimension und wenden sie auf ein Sicherheitsprofilverhalten an

1. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Erkennen, und wählen Sie dann Sicherheitsprofile.
2. Wählen Sie auf der Seite Sicherheitsprofile die Option Sicherheitsprofil erstellen und dann Regelbasiertes Profil zur Erkennung von Anomalien erstellen. Sie können auch eine Dimension auf ein vorhandenes regelbasiertes Sicherheitsprofil anwenden. Dazu wählen Sie das Sicherheitsprofil aus und klicken auf Bearbeiten.
3. Geben Sie auf der Seite Eigenschaften des Sicherheitsprofils angeben einen Namen für das Sicherheitsprofil ein.
4. Wählen Sie die Gerätegruppe aus, die Sie im Hinblick auf Anomalien analysieren möchten.
5. Wählen Sie Weiter.

6. Wählen Sie auf der Seite Metrikverhalten konfigurieren unter Metriktyp eine der cloudseitigen Metrikdimensionen.
7. Wählen Sie für Metrikverhalten die Option Warnung senden (metrisches Verhalten definieren), um das erwartete Verhalten der Metrik zu definieren.
8. Wählen Sie aus, wann Sie bei ungewöhnlichem Geräteverhalten benachrichtigt werden möchten.
9. Wählen Sie Weiter aus.
10. Prüfen Sie die Konfiguration des Sicherheitsprofils, und wählen Sie Erstellen.

So zeigen Sie Ihre Alarme an

1. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Erkennen, und wählen Sie dann Sicherheitsprofile.
2. Wählen Sie das Element in der Spalte Elementname, um Informationen darüber zu erhalten, was den Alarm ausgelöst hat.

So zeigen Sie Ihre Dimensionen an und aktualisieren sie

1. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Erkennen, und wählen Sie dann Dimensionen.
2. Wählen Sie die LAG, und klicken Sie dann auf Bearbeiten.
3. Bearbeiten Sie die Dimension, und wählen Sie dann Aktualisieren.

So löschen Sie eine Dimension

1. Öffnen Sie die AWS IoT-Konsole. Erweitern Sie im Navigationsbereich die Optionen Sicherheit, Erkennen, und wählen Sie dann Dimensionen.
2. Bevor Sie eine Dimension löschen, müssen Sie das metrische Verhalten löschen, das auf die Dimension verweist. Vergewissern Sie sich, dass die Dimension keinem Sicherheitsprofil angefügt ist. Prüfen Sie dazu die Spalte Sicherheitsprofile. Wenn die Dimension einem Sicherheitsprofil angefügt ist, öffnen Sie die Seite Sicherheitsprofile links und bearbeiten die Sicherheitsprofile, denen die Dimension angefügt ist. Dann können Sie mit dem Löschen des Verhaltens fortfahren. Wenn Sie eine andere Dimension löschen möchten, führen Sie die Schritte in diesem Abschnitt aus.
3. Wählen Sie die Dimension und dann die Option Löschen.

4. Geben Sie zur Bestätigung den Dimensionsnamen in das Feld ein. Wählen Sie anschließend Löschen.

So verwenden Sie Dimensionen in AWS CLI

So erstellen Sie eine Dimension und wenden sie auf ein Sicherheitsprofilverhalten an

1. Sie müssen die Dimension erstellen, bevor Sie sie an ein Sicherheitsprofil anfügen können. Erstellen Sie eine Dimension mit dem Befehl CreateDimension:

```
aws iot create-dimension \
  --name TopicFilterForAuthMessages \
  --type TOPIC_FILTER \
  --string-values device/+/auth
```

   Die Ausgabe dieses Befehls sieht wie folgt aus:

```
{
    "arn": "arn:aws:iot:us-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
    "name": "TopicFilterForAuthMessages"
}
```

2. Fügen Sie die Dimension mit UpdateSecurityProfile einem vorhandenen Sicherheitsprofil oder mit CreateSecurityProfile einem neuen Sicherheitsprofil hinzu. Im folgenden Beispiel wird ein neues Sicherheitsprofil erstellt, das überprüft, ob Nachrichten an `TopicFilterForAuthMessages` weniger als 128 Bytes enthalten. Die Anzahl der Nachrichten, die an andere als Authentifizierungsthemen gesendet werden, wird gespeichert.

```
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\"}}]"
```

Die Ausgabe dieses Befehls sieht wie folgt aus:

```
{
    "securityProfileArn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice",
    "securityProfileName": "ProfileForConnectedDevice"
}
```

Um Zeit zu sparen können Sie einen Parameter auch aus einer Datei laden, statt ihn als Befehlszeilen-Parameterwert einzugeben. Weitere Informationen finden Sie unter Laden von AWS CLI Parametern aus einer Datei. Im Folgenden wird der Parameter `behavior` im erweiterten JSON-Format gezeigt:

```
[
  {
    "criteria": {
      "comparisonOperator": "less-than",
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "value": {
        "count": 128
      }
    },
    "metric": "aws:message-byte-size",
    "metricDimension": {
      "dimensionName:": "TopicFilterForAuthMessages"
    },
    "name": "CellularBandwidth"
  }
]
```

Oder verwenden Sie CreateSecurityProfile mithilfe von Dimension mit ML wie im folgenden Beispiel:

```
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
```

So zeigen Sie Sicherheitsprofile mit einer Dimension an

- Mit dem Befehl ListSecurityProfiles können Sie Sicherheitsprofile mit einer bestimmten Dimension anzeigen:

  ```
  aws iot list-security-profiles \
    --dimension-name TopicFilterForAuthMessages
  ```

  Die Ausgabe dieses Befehls sieht wie folgt aus:

  ```
  {
      "securityProfileIdentifiers": [
          {
              "name": "ProfileForConnectedDevice",
              "arn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
  ProfileForConnectedDevice"
          }
      ]
  }
  ```

So aktualisieren Sie Ihre Dimension

- Mit dem Befehl UpdateDimension können Sie eine Dimension aktualisieren:

  ```
  aws iot update-dimension \
    --name TopicFilterForAuthMessages \
    --string-values device/${iot:ClientId}/auth
  ```

  Die Ausgabe dieses Befehls sieht wie folgt aus:

  ```
  {
  ```

```
    "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"
}
```

So löschen Sie eine Dimension

1. Um eine Dimension zu löschen, trennen Sie sie zunächst von allen Sicherheitsprofilen, denen sie angefügt ist. Mit dem Befehl ListSecurityProfiles können Sie Sicherheitsprofile mit einer bestimmten Dimension anzeigen.
2. Mit dem Befehl UpdateSecurityProfile können Sie eine Dimension aus einem Sicherheitsprofil entfernen. Geben Sie alle Informationen ein, die Sie behalten möchten, jedoch nicht die Dimension:

   ```
   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}}]"
   ```

   Die Ausgabe dieses Befehls sieht wie folgt aus:

   ```
   {
     "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. Nach der Trennung der Dimension können Sie die Dimension mit dem Befehl DeleteDimension löschen:

```
aws iot delete-dimension \
  --name TopicFilterForAuthMessages
```

Berechtigungen

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

Dieser Abschnitt beschreibt, wie die erforderlichen IAM-Rollen und -Richtlinien zum Verwalten von AWS IoT Device Defender Detect eingerichtet werden. Weitere Informationen finden Sie im IAM-Benutzerhandbuch.

Gewähren AWS IoT Device Defender der Berechtigung zum Veröffentlichen von Warnungen in einem SNS-Thema

Wenn Sie den Parameter `alertTargets` in CreateSecurityProfile verwenden, müssen Sie eine IAM-Rolle mit zwei Richtlinien angeben: einer Berechtigungsrichtlinie und einer Vertrauensrichtlinie. Die Berechtigungsrichtlinie erteilt AWS IoT Device Defender die Berechtigung zum Veröffentlichen von Benachrichtigungen in Ihrem SNS-Thema. Die Vertrauensrichtlinie erteilt AWS IoT Device Defender die Berechtigung zur Übernahme der erforderlichen Rolle.

Berechtigungsrichtlinie

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": [
                "arn:aws:sns:us-east-1:123456789012:your-topic-name"
            ]
        }
```

```
    ]
}
```

Vertrauensrichtlinie

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Richtlinie zum Übergeben von Rollen

Sie benötigen auch eine IAM-Berechtigungsrichtlinie, die dem IAM-Benutzer zugeordnet ist und es diesem ermöglicht, Rollen zu übergeben. Beachten Sie hierzu den Abschnitt Gewähren von Berechtigungen, mit denen ein Benutzer eine Rolle an einen AWS-Service übergeben kann.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole"
            ],
```

```
            "Resource": "arn:aws:iam::123456789012:role/Role_To_Pass"
        }
    ]
}
```

Detect-Befehle

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

Sie können die Detect-Befehle in diesem Abschnitt verwenden, um die Sicherheitsprofile ML Detect oder Rules Detect so zu konfigurieren, dass sie ungewöhnliche Verhaltensweisen, die auf ein kompromittiertes Gerät hinweisen könnten, identifizieren und überwachen.

Befehle für DetectMitigation-Aktionen

Starten und Verwalten der Detect-Ausführung
CancelDetectMitigationActionsTask
DescribeDetectMitigationActionsTask
ListDetectMitigationActionsTasks
StartDetectMitigationActionsTask
ListDetectMitigationActionsExecutions

Befehle für Dimensionsaktionen

Starten und Verwalten der Dimensionsausführung
CreateDimension

Starten und Verwalten der Dimensionsausführung
DescribeDimension
ListDimensions
DeleteDimension
UpdateDimension

CustomMetric-Action-Befehle

Starten und Verwalten der CustomMetric-Ausführung
CreateCustomMetric
UpdateCustomMetric
DescribeCustomMetric
ListCustomMetrics
DeleteCustomMetric

Befehle für Sicherheitsprofil-Aktionen

Starten und Verwalten der Sicherheitsprofil-Ausführung
CreateSecurityProfile
AttachSecurityProfile
DetachSecurityProfile
DeleteSecurityProfile
DescribeSecurityProfile
ListTargetsForSecurityProfile

Starten und Verwalten der Sicherheitsprofil-Ausführung
UpdateSecurityProfile
ValidateSecurityProfileBehaviors
ListSecurityProfilesForTarget

Befehle für Alarmaktionen

Verwalten von Alarmen und Zielen
ListActiveViolations
ListViolationEvents
PutVerificationStateOnViolation

Befehle für MLDetect-Aktionen

Auflisten der Trainingsdaten des ML-Modells
GetBehaviorModelTrainingSummaries

Funktionsweise von AWS IoT Device Defender Detect

> **Note**
>
> Die Erkennungsfunktion von AWS IoT Device Defender ist für Neukunden nicht mehr verfügbar. Weitere Informationen zu Alternativen zu AWS IoT Device Defender Detect finden Sie unter Änderung der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender. An der Verfügbarkeit der AWS-IoT-Device-Defender-Audits ändert sich nichts.

1. Sie können AWS IoT Device Defender Detect nur mit cloudseitigen Metriken verwenden. Wenn Sie jedoch vom Gerät gemeldete Metriken verwenden möchten, müssen Sie auf Ihren über AWS

IoT verbundenen Geräten oder Geräte-Gateways zunächst ein AWS IoT SDK bereitstellen. Weitere Informationen finden Sie unter Senden von Metriken von Geräten.

2. Sie sollten die von Ihren Geräten generierten Metriken anzeigen, bevor Sie Verhaltensweisen definieren und Warnungen erstellen. AWS IoT kann Metriken von Ihren Geräte erfassen, sodass Sie zuerst normale oder ungewöhnliche Verhaltensweisen für eine Gruppe von Geräten oder für alle Geräte in Ihrem Konto bestimmen können. Verwenden Sie CreateSecurityProfile, geben Sie jedoch nur die `additionalMetricsToRetain` an, die für Sie relevant sind. Geben Sie `behaviors` zu diesem Zeitpunkt noch nicht an.

   Zeigen Sie die Gerätemetriken mit der AWS IoT-Konsole an, um zu ergründen, welche Verhaltensweisen für Ihre Geräte typisch sind.

3. Erstellen Sie eine Gruppe von Verhaltensweisen für Ihr Sicherheitsprofil. Verhaltensweisen enthalten Metriken, die normales Verhalten für eine Gruppe von Geräten oder für alle Geräte in Ihrem Konto angeben. Weitere Informationen und Beispiele finden Sie unter Cloudseitige Metriken und Geräteseitige Metriken. Nachdem Sie eine Gruppe von Verhaltensweisen erstellt haben, können Sie diese mithilfe von ValidateSecurityProfileBehaviors validieren.

4. Erstellen Sie mit der Aktion CreateSecurityProfile ein Sicherheitsprofil mit Ihren Verhaltensweisen. Sie können mit dem Parameter `alertTargets` veranlassen, dass Alarme an ein Ziel (ein SNS-Thema) gesendet werden, wenn ein Gerät gegen eine Verhaltensweise verstößt. (Wenn Sie mit SNS Alarme senden, sollten Sie daran denken, dass diese auf das SNS-Themenkontingent Ihres AWS-Kontos angerechnet werden. Es kann sein, dass bei einem starken Anstieg an Verstößen Ihr SNS-Themenkontingent überschritten wird. Sie können auch CloudWatch-Metriken auf Verstöße überprüfen. Weitere Informationen finden Sie unter Überwachen von AWS IoT-Alarmen und Metriken mithilfe von Amazon CloudWatch im AWS IoT Core-Entwicklerhandbuch.

5. Verwenden Sie die Aktion AttachSecurityProfile, um das Sicherheitsprofil einer Gerätegruppe (einer Objektgruppe), allen in Ihrem Konto registrierten Objekten, allen nicht registrierten Objekten oder allen Geräten anzufügen. AWS IoT Device Defender Detect beginnt mit der Überprüfung auf anormale Verhaltensweisen und sendet Alarme, falls Verstöße gegen normale Verhaltensweisen erkannt werden. Möglicherweise möchten Sie allen nicht registrierten Objekten ein Sicherheitsprofil zuweisen, wenn Sie z. B. vorhaben, mit mobilen Geräten zu interagieren, die sich nicht in der Objektregistrierung Ihres Kontos befinden. Sie können verschiedene Sätze von Verhaltensweisen für verschiedene Gruppen von Geräten definieren, um Ihren Anforderungen gerecht zu werden.

Um ein Sicherheitsprofil einer Gruppe von Geräten zuzuweisen, müssen Sie den ARN der Objektgruppe angeben, in der sie enthalten sind. Der ARN einer Objektgruppe weist folgendes Format auf.

```
arn:aws:iot:region:account-id:thinggroup/thing-group-name
```

Wenn Sie allen registrierten Objekten in einem AWS-Konto ein Sicherheitsprofil anfügen möchten (nicht registrierte Objekte werden ignoriert), müssen Sie einen ARN im folgenden Format angeben.

```
arn:aws:iot:region:account-id:all/registered-things
```

Wenn Sie allen nicht registrierten Objekten ein Sicherheitsprofil anfügen möchten, müssen Sie einen ARN im folgenden Format angeben.

```
arn:aws:iot:region:account-id:all/unregistered-things
```

Wenn Sie allen Geräten ein Sicherheitsprofil anfügen möchten, müssen Sie einen ARN im folgenden Format angeben.

```
arn:aws:iot:region:account-id:all/things
```

6. Sie können Verstöße auch mit der Aktion ListActiveViolations nachverfolgen. Auf diese Weise können Sie sehen, welche Verstöße für ein bestimmtes Sicherheitsprofil oder Zielgerät erkannt wurden.

   Verwenden Sie die Aktion ListViolationEvents, um festzustellen, welche Verstöße während eines bestimmten Zeitraums entdeckt wurden. Sie können diese Ergebnisse nach einem bestimmten Sicherheitsprofil, Gerät oder Alarm-Verifizierungsstatus filtern.
7. Mit der Aktion PutVerificationStateOnViolation können Sie Ihre Alarme überprüfen, organisieren und verwalten, indem Sie ihren Verifizierungsstatus markieren und eine Beschreibung dieses Verifizierungsstatus angeben.
8. Wenn Ihre Geräte zu oft oder nicht oft genug gegen die definierten Verhaltensweisen verstoßen, sollten Sie die Verhaltensweisen genauer definieren.

9. Verwenden Sie die Aktionen ListSecurityProfiles, ListSecurityProfilesForTarget und ListTargetsForSecurityProfile, um die von Ihnen eingerichteten Sicherheitsprofile und die überwachten Geräte zu überprüfen.

   Verwenden Sie die Aktion DescribeSecurityProfile, um weitere Details zu einem Sicherheitsprofil anzufordern.

10. Verwenden Sie die Aktion UpdateSecurityProfile, um ein Sicherheitsprofil zu aktualisieren. Verwenden Sie die Aktion DetachSecurityProfile, um ein Sicherheitsprofil von einem Konto oder einer Ziel-Objektgruppe zu trennen. Verwenden Sie die Aktion DeleteSecurityProfile, um ein Sicherheitsprofil vollständig zu löschen.

Abschwächungsaktionen

Sie können mit AWS IoT Device Defender Maßnahmen ergreifen, um Probleme zu beheben, die in einer Audit-Erkenntnis oder einem Detect-Alarm festgestellt wurden.

> **Note**
>
> Bei unterdrückten Audit-Ergebnissen werden keine Abschwächungsaktionen ergriffen. Weitere Informationen zur Unterdrückung von Prüfungsergebnissen finden Sie unter Unterdrückungen von Prüfergebnissen.

Abschwächungsaktionen für Audits

AWS IoT Device Defender bietet vordefinierte Aktionen für die verschiedenen Audit-Prüfungen. Sie konfigurieren diese Aktionen für Ihr AWS-Konto und übernehmen sie anschließend für eine Gruppe von Erkenntnissen. Diese Erkenntnisse sind möglich:

- Alle Erkenntnisse aus einer Prüfung. Diese Option ist über die AWS IoT-Konsole oder über die AWS CLI verfügbar.
- Eine Liste von einzelnen Erkenntnissen. Diese Option ist nur über die verfügbar AWS CLI.
- Eine gefilterte Gruppe von Erkenntnissen aus einer Prüfung.

Die folgende Tabelle listet die Arten von Audit-Prüfungen und die jeweils unterstützten Abschwächungsaktionen auf:

Prüfung der Zuweisung von Abschwächungsaktionen

Prüfung	Unterstützte Abschwächungsaktionen
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

Prüfung	Unterstützte Abschwächungsaktionen
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

Prüfung	Unterstützte Abschwächungsaktionen
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

Alle Audit-Prüfungen unterstützen die Veröffentlichung von Audit-Ergebnissen in Amazon SNS, daher können Sie Ihre Aktionen als Reaktion auf die Benachrichtigung anpassen. Jede Art von Audit-Prüfung kann zusätzliche Abschwächungsaktionen unterstützen:

REVOKED_CA_CERT_CHECK

- Ändern Sie den Status des Zertifikats in als inaktiv AWS IoT.

DEVICE_CERTIFICATE_SHARED_CHECK

- Ändern Sie den Status des Gerätezertifikat in als inaktiv AWS IoT.
- Fügen Sie die Geräte, die dieses Zertifikat verwenden, eine Objektgruppe hinzu.

UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- Keine zusätzliche Aktionen unterstützt.

AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- Keine zusätzliche Aktionen unterstützt.

IOT_POLICY_OVERLY_PERMISSIVE_CHECK

- Fügen Sie eine leere AWS IoT-Richtlinienversion hinzu, um Berechtigungen einzuschränken.

IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK

- Identifizieren Sie potenzielle Fehlkonfigurationen in AWS IoT-Richtlinien.

CA_CERT_APPROACHING_EXPIRATION_CHECK

- Ändern Sie den Status des Zertifikats in als inaktiv AWS IoT.

CONFLICTING_CLIENT_IDS_CHECK

- Keine zusätzliche Aktionen unterstützt.

DEVICE_CERT_APPROACHING_EXPIRATION_CHECK

- Ändern Sie den Status des Gerätezertifikat in als inaktiv AWS IoT.
- Fügen Sie die Geräte, die dieses Zertifikat verwenden, eine Objektgruppe hinzu.

DEVICE_CERTIFICATE_KEY_QUALITY_CHECK

- Ändern Sie den Status des Gerätezertifikat in als inaktiv AWS IoT.
- Fügen Sie die Geräte, die dieses Zertifikat verwenden, eine Objektgruppe hinzu.

CA_CERTIFICATE_KEY_QUALITY_CHECK

- Ändern Sie den Status des Zertifikats in als inaktiv AWS IoT.

REVOKED_DEVICE_CERT_CHECK

- Ändern Sie den Status des Gerätezertifikat in als inaktiv AWS IoT.
- Fügen Sie die Geräte, die dieses Zertifikat verwenden, eine Objektgruppe hinzu.

LOGGING_DISABLED_CHECK

- Aktivieren Sie die Protokollierung.

AWS IoT Device Defender unterstützt die folgenden Arten von Abschwächungsaktionen in Audit-Ergebnissen:

Aktionstyp	Hinweise
ADD_THINGS_TO_THING_GROUP	Geben Sie die Gruppe an, der Sie die Geräte hinzufügen möchten. Geben Sie auch an, ob die Mitgliedschaft in einer oder mehreren dynamischen Gruppen überschrieben werden soll, wenn dies die maximale Anzahl von Gruppen, zu denen das Objekt gehören kann, überschreiten würde.
ENABLE_IOT_LOGGING	Sie geben die Protokollierungsstufe und die Rolle mit Berechtigungen für die Protokollierung an. Sie können nicht als Protokollierungsstufe angeben `DISABLED`.
PUBLISH_FINDING_TO_SNS	Sie geben das Thema an, zu dem die Erkenntnis veröffentlicht werden soll.

Aktionstyp	Hinweise
REPLACE_DEFAULT_POLICY_VERSION	Sie geben den Namen der Vorlage an. Ersetzt die Richtlinienversion mit einem Standard- oder leeren Richtlinie. Derzeit wird nur der Wert `BLANK_POLICY` unterstützt.
UPDATE_CA_CERTIFICATE	Sie geben den neuen Status für das CA-Zertifikat an. Derzeit wird nur der Wert `DEACTIVATE` unterstützt.
UPDATE_DEVICE_CERTIFICATE	Sie geben den neuen Status für das Gerätezertifikat an. Derzeit wird nur der Wert `DEACTIVATE` unterstützt.

Durch die Konfiguration von Standardaktionen für den Fall, dass bei einer Prüfung Probleme gefunden werden, können Sie auf derartige Probleme konsistent reagieren. Mit diesen definierten Abschwächungsaktionen können Sie auch Probleme schneller und mit geringerem Risiko menschlichen Versagens die beheben.

> **Important**
>
> Bei Anwendung von Abschwächungsaktionen, bei denen Zertifikate geändert werden, Objekte einer neuen Objektgruppe hinzugefügt werden oder die Richtlinie ersetzt wird, kann dies Auswirkungen auf Ihre Geräte und Anwendungen haben. Beispiel: Geräte können u. U. keine Verbindung mehr herstellen. Beachten Sie die Auswirkungen der Abschwächungsaktionen, bevor Sie sie anwenden. Möglicherweise müssen Sie andere Aktionen ausführen, um die Probleme zu lösen, bevor Ihre Geräte und Anwendungen wieder normal ausgeführt werden können. Beispielsweise müssen Sie möglicherweise aktualisierte Gerätezertifikate bereitstellen. Abschwächungsaktionen können Ihnen dabei helfen, Risiken schnell zu verringern, aber müssen Sie weiterhin korrigierende Maßnahmen ergreifen, um die zugrunde liegenden Probleme zu lösen.

Einige Aktionen, wie z. B. ein Gerätezertifikat erneut zu aktivieren, können nur manuell ausgeführt werden. AWS IoT Device Defender verfügt über keinen Mechanismus, um automatisch für die Abschwächungsaktionen nach deren Anwendung ein Rollback durchzuführen.

Detect-Abschwächungsaktionen

AWS IoT Device Defender unterstützt die folgenden Arten von Abschwächungsaktionen für Detect-Alarme:

Aktionstyp	Hinweise
ADD_THINGS_TO_THING_GROUP	Geben Sie die Gruppe an, der Sie die Geräte hinzufügen möchten. Geben Sie auch an, ob die Mitgliedschaft in einer oder mehreren dynamischen Gruppen überschrieben werden soll, wenn dies die maximale Anzahl von Gruppen, zu denen das Objekt gehören kann, überschreiten würde.

Verfahren zum Definieren und Verwalten von Abschwächungsaktionen

Sie können die AWS IoT-Konsole oder die AWS CLI verwenden, um Abschwächungsaktionen für Ihr AWS-Konto zu definieren und zu verwalten.

Erstellen von Abschwächungsaktionen

Jede Abschwächungsaktion, die Sie definieren, ist eine Kombination aus einem vordefinierten Aktionstyp und Parametern, die speziell für Ihr Konto gelten.

So verwenden Sie die AWS IoT-Konsole zum Erstellen der Abschwächungsaktionen

1. Öffnen Sie die Seite mit den Abschwächungsaktionen in der AWS IoT-Konsole.
2. Wählen Sie auf der Seite Abschwächungsaktionen die Option Erstellen.
3. Geben Sie auf der Seite Neue Abschwächungsaktion erstellen in Aktionsname einen eindeutigen Namen für Ihre Abschwächungsaktion an.
4. Geben Sie unter Aktionstyp den Typ der Aktion an, die Sie definieren möchten.
5. Wählen Sie unter Berechtigungen die IAM-Rolle aus, unter deren Berechtigungen die Aktion angewendet wird.

6. Jede Aktionsart benötigt einen eigenen Satz von Parametern. Geben Sie die Parameter für die Aktion ein. Wenn Sie beispielsweise den Aktionstyp Objekte zu einer Objektgruppe hinzufügen wählen, wählen Sie die Zielgruppe aus und aktivieren Sie bei Bedarf die Option Überschreiben dynamischer Gruppen.
7. Wählen Sie Erstellen aus, um Ihre Abschwächungsaktion zu Ihrem AWS-Konto hinzuzufügen.

So verwenden Sie die AWS CLI zum Erstellen von Abschwächungsaktionen

- Verwenden Sie den Befehl CreateMitigationAction zum Erstellen Ihrer Abschwächungsaktion. Der eindeutige Name, den Sie der Aktion geben, wird verwendet, wenn Sie diese Aktion auf Audit-Ergebnisse anwenden. Wählen Sie einen aussagekräftigen Namen.

So verwenden Sie die AWS IoT-Konsole zum Anzeigen und Ändern der Abschwächungsaktionen

1. Öffnen Sie die Seite mit den Abschwächungsaktionen in der AWS IoT-Konsole.

   Auf der Seite Abschwächungsaktionen wird eine Liste aller Abschwächungsaktionen angezeigt, die für Ihr AWS-Konto definiert sind.
2. Wählen Sie den Namenslink für die Abschwächungsaktion aus, die Sie ändern möchten.
3. Wählen Sie Bearbeiten aus, um die gewünschten Änderungen an der Abschwächungsaktion vorzunehmen. Sie können den Namen nicht ändern, da der Name der Abschwächungsaktion verwendet wird, um diese zu identifizieren.
4. Klicken Sie auf Aktualisieren um die Änderungen an der Abschwächungsaktion in Ihrem AWS-Konto zu speichern.

So verwenden Sie die AWS CLI zum Auflisten von Abschwächungsaktionen

- Verwenden Sie den Befehl ListMitigationAction, um die Liste Ihrer Abschwächungsaktionen anzuzeigen. Wenn Sie eine Abschwächungsaktion ändern oder löschen möchten, notieren Sie sich den Namen.

So verwenden Sie die AWS CLI zum Aktualisieren von Abschwächungsaktionen

- Verwenden Sie den Befehl UpdateMitigationAction, um Ihre Abschwächungsaktion zu ändern.

So verwenden Sie die AWS IoT-Konsole zum Löschen von Abschwächungsaktionen

1. Öffnen Sie die Seite mit den Abschwächungsaktionen in der AWS IoT-Konsole.

   Auf der Seite Abschwächungsaktionen werden alle Abschwächungsaktionen angezeigt, die für Ihr AWS-Konto definiert sind.
2. Wählen Sie die Abschwächungsaktion aus, die Sie löschen möchten, und wählen Sie dann Löschen aus.
3. Wählen Sie im Fenster Möchten Sie die CEV wirklich löschen die Option Löschen aus.

So verwenden Sie die AWS CLI, um Abschwächungsaktionen zu löschen

- Verwenden Sie den Befehl UpdateMitigationAction, um Ihre Abschwächungsaktion zu ändern.

So verwenden Sie die AWS IoT-Konsole, um Details zu Abschwächungsaktionen anzuzeigen

1. Öffnen Sie die Seite mit den Abschwächungsaktionen in der AWS IoT-Konsole.

   Auf der Seite Abschwächungsaktionen werden alle Abschwächungsaktionen angezeigt, die für Ihr AWS-Konto definiert sind.
2. Wählen Sie den Namenslink für die Abschwächungsaktion aus, die Sie anzeigen möchten.

So verwenden Sie AWS CLI, um die Details von Abschwächungsaktionen anzuzeigen

- Verwenden Sie den Befehl DescribeMitigationAction, um Details zu Ihren Abschwächungsaktionen anzuzeigen.

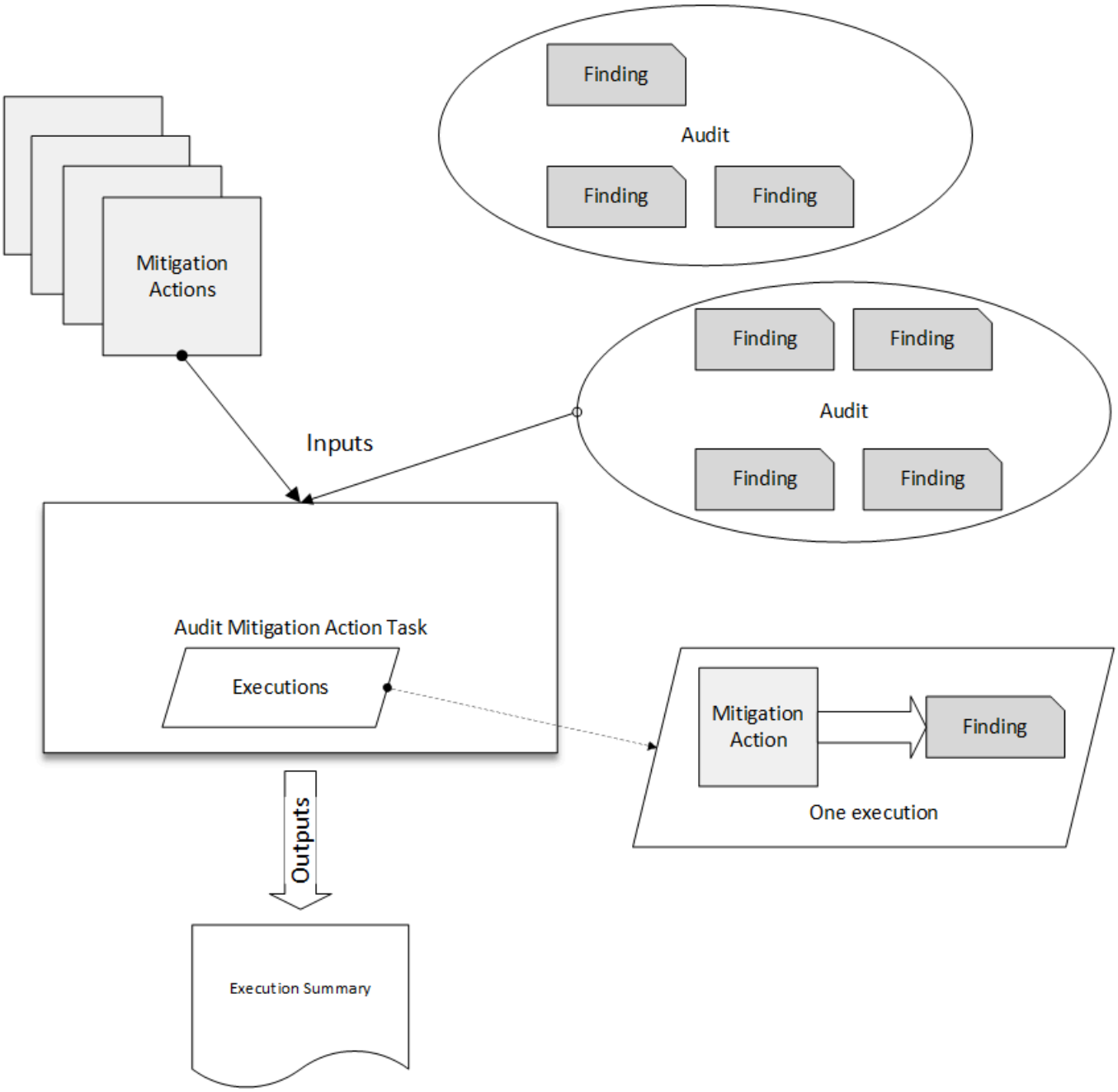
Anwenden von Abschwächungsaktionen

Nachdem Sie eine Reihe von Abschwächungsaktionen definiert haben, können Sie Aktionen auf Prüfungsergebnisse anwenden. Wenn Sie Aktionen anwenden, starten Sie eine Aufgabe zur Abschwächungsaktion für die Prüfung. Dieser Vorgang kann, abhängig von der Menge der Erkenntnisse und der Aktionen, die Sie auf sie anwenden möchten, einige Zeit in Anspruch nehmen. Wenn Sie zum Beispiel über einen großen Pool von Geräten verfügen, deren Zertifikate abgelaufen sind, kann es einige Zeit dauern, bis alle diese Zertifikate deaktiviert sind oder diese Geräte in eine Quarantänegruppe verschoben sind. Andere Aktionen, wie z. B. die Aktivierung der Protokollierung, können schnell abgeschlossen werden.

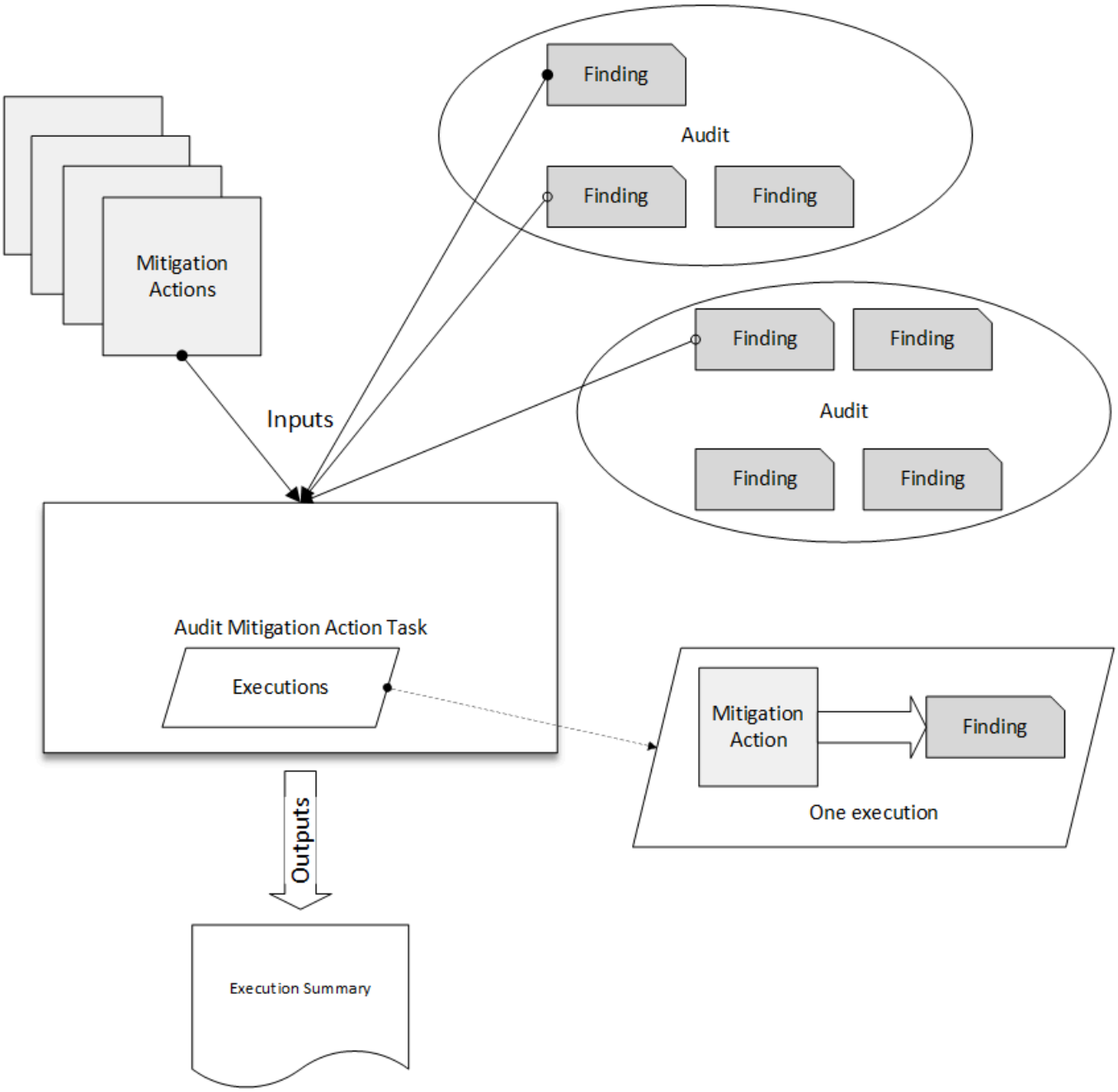
Sie können die Ausführungsliste der Aktion anzeigen und eine Ausführung abbrechen, die noch nicht abgeschlossen ist. Bereits im Rahmen der Ausführung der abgebrochenen Aktion ausgeführte Aktionen werden nicht rückgängig gemacht. Wenn Sie mehrere Aktionen auf eine Reihe von Erkenntnissen anwenden und eine dieser Aktionen fehlgeschlagen ist, werden die nachfolgenden Aktionen für diese Erkenntnis übersprungen (auf andere Erkenntnisse aber immer noch angewendet). Der Aufgabenstatus für die Erkenntnis ist FEHLGESCHLAGEN. Der `taskStatus` lautet „fehlgeschlagen“, wenn eine oder mehrere der Aktionen fehlgeschlagen, wenn die Aktionen auf die Erkenntnisse angewendet werden. Aktionen werden in der Reihenfolge angewendet, in der sie angegeben sind.

Jede Ausführung einer Aktion wendet eine Reihe von Aktionen auf ein Ziel an. Bei dem Ziel kann es sich um eine Liste von Erkenntnissen handeln oder um alle Erkenntnisse aus einer Prüfung.

Das folgende Diagramm zeigt, wie Sie eine Aufgabe als Abschwächungsaktion definieren, die alle Erkenntnisse aus einer Prüfung nimmt und eine Reihe von Aktionen auf diese Erkenntnisse anwendet. Eine Ausführung wendet jeweils eine Aktion auf eine Erkenntnis an. Die Aufgabe zur Abschwächungsaktion für die Prüfung gibt eine Ausführungszusammenfassung aus.



Das folgende Diagramm zeigt, wie Sie eine Aufgabe zur Abschwächungsaktion für die Prüfung definieren können, die eine Liste der einzelnen Erkenntnisse aus einer oder mehreren Prüfungen nimmt und eine Reihe von Aktionen auf diese Erkenntnisse anwendet. Eine Ausführung wendet jeweils eine Aktion auf eine Erkenntnis an. Die Aufgabe zur Abschwächungsaktion für die Prüfung gibt eine Ausführungszusammenfassung aus.



Sie können die AWS IoT-Konsole oder die AWS CLI verwenden, um Abschwächungsaktionen anzuwenden.

So verwenden Sie die AWS IoT-Konsole, um Abschwächungsaktionen anzuwenden, indem Sie die eine Aktion ausführen

1. Öffnen Sie die Seite mit den Prüfungsergebnissen in der AWS IoT-Konsole.
2. Wählen Sie den Namen für die Prüfung, auf die Sie Aktionen anwenden möchten.

3. Wählen Sie Abschwächungsaktion starten aus. Diese Schaltfläche ist nicht verfügbar, wenn alle Ihre Prüfungen konform sind.
4. In Neue Abschwächungsaktion starten wird standardmäßig der Name der Audit-ID vorgegeben, Sie können jedoch einen aussagekräftigeren Namen verwenden.
5. Für jede Art der Prüfung, die zu einem oder mehreren nonkonformen Erkenntnissen geführt hat, können Sie eine oder mehrere anzuwendende Aktionen auswählen. Es werden nur Aktionen, die für den Prüfungstyp gültig sind, angezeigt.

   > **Note**
   >
   > Wenn Sie keine Aktionen für Ihr AWS-Konto konfiguriert haben, ist die Liste der Aktionen leer. Sie können den Link Abschwächungsaktion erstellen wählen, um eine oder mehrere Abschwächungsaktionen zu erstellen.

6. Wenn Sie alle Aktionen angegeben haben, die Sie anwenden möchten, wählen Sie Aufgabe starten aus.

So verwenden Sie die AWS CLI, um Abschwächungsaktionen anzuwenden, indem Sie die Ausführung einer Abschwächungsaktion für eine Prüfung starten

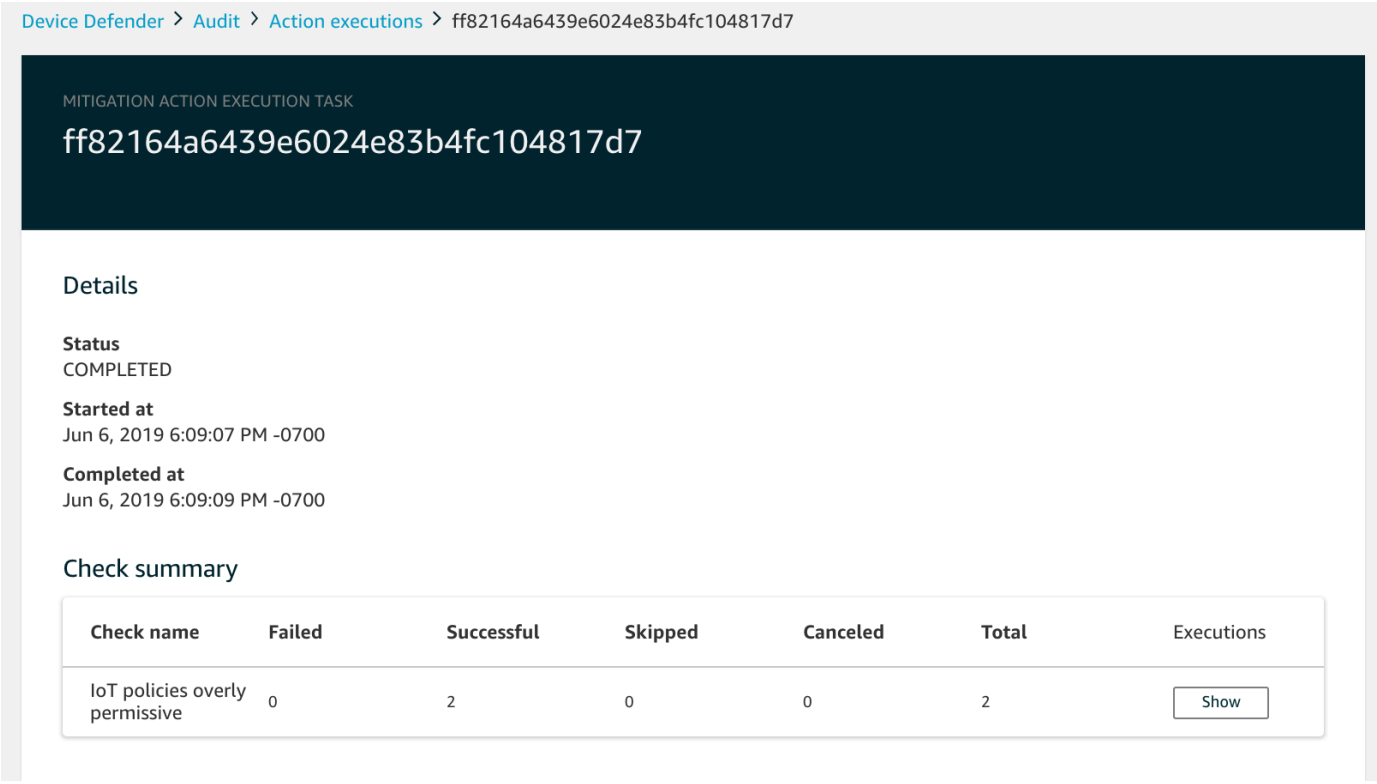
1. Wenn Sie möchten, dass Aktionen auf alle Erkenntnisse für die Prüfung angewendet werden, verwenden Sie den Befehl ListAuditTasks, um die Aufgaben-ID zu suchen.
2. Wenn Sie Aktionen nur auf ausgewählte Erkenntnisse anwenden möchten, verwenden Sie den Befehl ListAuditFindings, um die Ergebnis-IDs abzurufen.
3. Verwenden Sie den Befehl ListMitigationActions und notieren Sie sich die Namen der Abschwächungsaktionen, die Sie anwenden möchten.
4. Verwenden Sie den Befehl StartAuditMitigationActionsTask, um Aktionen auf das Ziel anzuwenden. Notieren Sie sich die Aufgaben-ID. Sie können die ID verwenden, um den Ausführungsstatus der Aktion zu überprüfen, die Details anzuzeigen oder die Aktion abzubrechen.

So verwenden Sie die AWS IoT-Konsole, um die Ausführung Ihrer Aktionen anzuzeigen

1. Öffnen Sie die Seite mit den Aktionsaufgaben in der AWS IoT-Konsole.

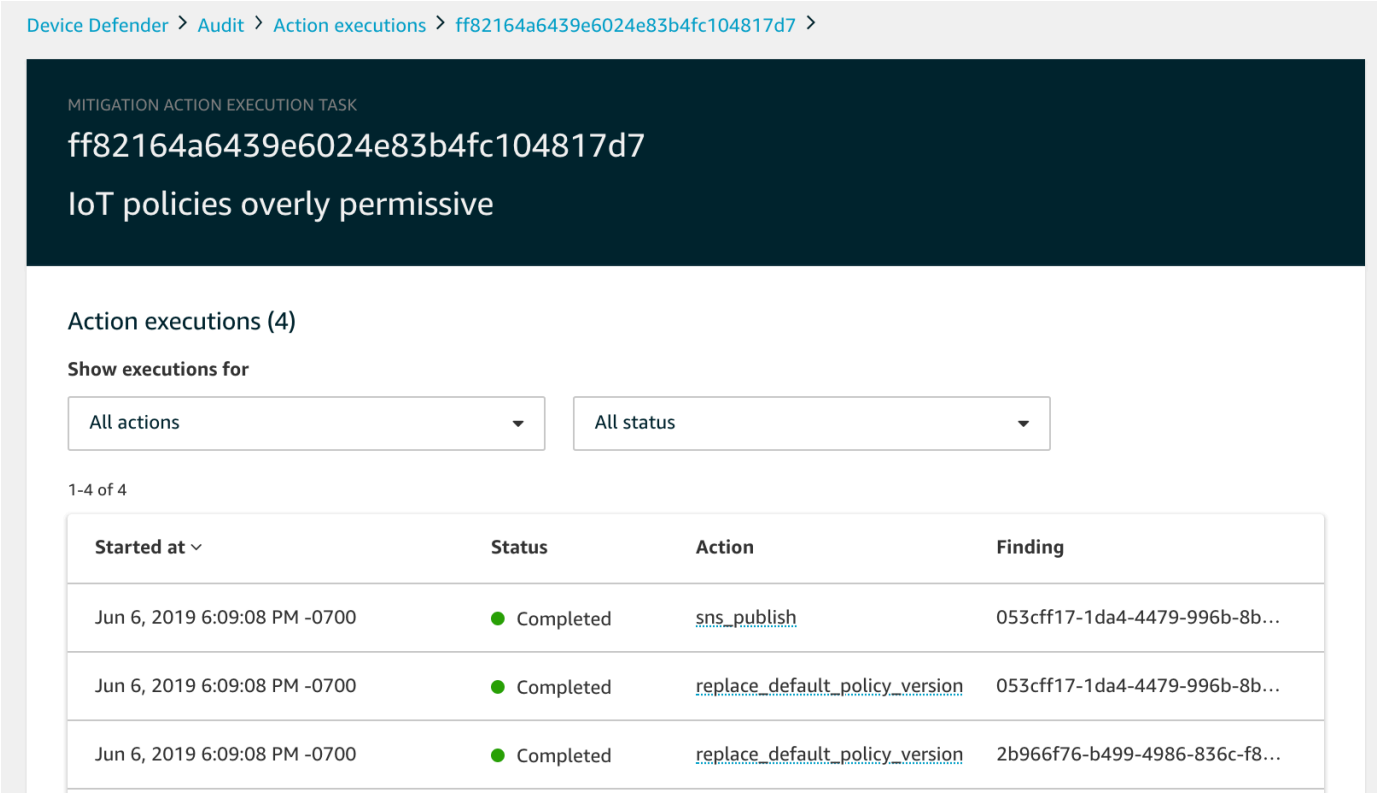
Es wird eine Liste der Aktionen angezeigt, zusammen mit dem Ausführungsstartzeitpunkt und dem aktuellen Status.

2. Wählen Sie den Link Name, um Details zu der Aufgabe anzuzeigen. Die Details umfassen alle Aktionen, die von der Aufgabe angewendet werden, sowie deren Ziel und deren Status.



Sie können die Filter unter Ausführungen anzeigen für verwenden, um nach Aktionstypen oder Aktionsstatus zu filtern.

3. Zum Anzeigen von Details für die Aufgabe wählen Sie unter Ausführungen die Option Anzeigen aus.



So verwenden Sie die AWS CLI, um Ihre gestarteten Aufgaben anzuzeigen

1. Verwenden Sie ListAuditMitigationActionsTasks, um Ihre Aufgaben zur Abschwächungsaktion für eine Prüfung anzuzeigen. Sie können Filter bereitstellen, um die Erkenntnisse einzugrenzen. Wenn Sie Details der Aufgabe anzeigen möchten, notieren Sie sich die Aufgaben-ID.
2. Verwenden Sie ListAuditMitigationActionsExecutions zum Anzeigen der Ausführungsdetails für eine bestimmte Aufgabe zur Abschwächungsaktion für eine Prüfung.
3. Verwenden Sie DescribeAuditMitigationActionsTask zur Anzeige von Details zur Aufgabe, wie z. B. die Parameter, mit der die Aufgabe gestartet wurde.

So verwenden Sie die AWS CLI, um eine aktuell ausgeführte Aufgabe zur Abschwächungsaktion für eine Prüfung abzubrechen

1. Verwenden Sie den Befehl ListAuditMitigationActionsTasks, um die Aufgaben-ID für die Aufgabe zu suchen, deren Ausführung Sie abbrechen möchten. Sie können Filter bereitstellen, um die Erkenntnisse einzugrenzen.
2. Verwenden Sie den Befehl ListDetectMitigationActionsExecutions mit der Aufgaben-ID, um die Aufgabe zur Abschwächungsaktion für eine Prüfung abzubrechen. Sie können keine

Aufgaben abbrechen, die abgeschlossen sind. Wenn Sie eine Aufgabe abbrechen, werden die verbleibenden Aktionen nicht angewendet, aber Abschwächungsaktionen, die bereits angewendet wurden, werden nicht rückgängig gemacht.

Berechtigungen

Für jede Abschwächungsaktion, die Sie definieren, müssen Sie die Rolle angeben, die bei der Anwendung der Aktion verwendet wird.

Berechtigungen für Abschwächungsaktionen

Aktionstyp	Berechtigungsrichtlinienvorlage	

UPDATE_DEVICE_CERTIFICATE	JSON
	
	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect": "Allow",
	"Action":[
	"iot:UpdateCertificate"
	],
	"Resource":[
	"*"
	]
	}
	]	

Aktionstyp	Berechtigungsrichtlinienvorlage	
	<pre>}</pre>	

UPDATE_CA_CERTIFICATE	JSON
	<pre>{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"iot:UpdateCACertificate"
	],
	"Resource":[
	"*"
	]
	}
	]
	}</pre>	

Aktionstyp	Berechtigungsrichtlinienvorlage	
ADD_THINGS_TO_THING_GROUP	JSON	

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

    "Statement":[
        {

  "Effect":
"Allow",

 "Action":[

    "iot:List
Principal
Things",

    "iot:AddT
hingToThi
ngGroup"
            ],

 "Resource":[

 "*"
            ]
        }
    ]
}
```

Aktionstyp	Berechtigungsrichtlinienvorlage	
REPLACE_DEFAULT_POLICY_VERSION	JSON	

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

    "Statement":[
        {

  "Effect":
"Allow",

 "Action":[

    "iot:Crea
tePolicyVersion"
            ],

 "Resource":[

 "*"
            ]
        }
    ]
}
```

Aktionstyp	Berechtigungsrichtlinienvorlage	
ENABLE_IOT_LOGGING	JSON	

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

  "Statement": [
    {
      "Effect":
 "Allow",
      "Action": [
        "iot:SetV
2LoggingOptions"
      ],
      "Resource":
 "*"
    },
    {
      "Effect":
 "Allow",
      "Action": [
        "iam:Pass
Role"
      ],
      "Resource
": "arn:aws:
iam::1234
56789012:role/
IoTLoggingRole"
    }
  ]
}
```

Aktionstyp	Berechtigungsrichtlinienvorlage	
PUBLISH_FINDING_TO_SNS	JSON	

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

    "Statement":[
        {

  "Effect":
"Allow",

 "Action":[

 "sns:Publish"
           ],

 "Resource":[

 "arn:aws:sns: us-
east-1 :12345678
9012: example-t
opic "
           ]
        }
    ]
}
```

Verwenden Sie für alle Arten von Abschwächungsaktionen die folgende Vertrauensrichtlinienvorlage:

JSON

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

```
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iot:*:123456789012::*"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012:"
        }
      }
    }
  ]
}
```

Befehle für Abschwächungsaktionen

Sie können diese Befehle für Abschwächungsaktionen zum Definieren einer Reihe von Aktionen für Ihr AWS-Konto verwenden, die zu einem späteren Zeitpunkt auf eine oder mehrere Gruppen von Prüfungsergebnissen angewendet werden können. Es gibt drei Befehlskategorien:

- Befehle zur Definition und Verwaltung von Aktionen.
- Befehle zum Starten und Verwalten der Anwendung dieser Aktionen auf Prüfungsergebnisse.
- Befehle zum Starten und Verwalten der Anwendung dieser Aktionen auf Detect-Alarme.

Befehle für Abschwächungsaktionen

Definieren und Verwalten von Aktionen	Starten und Verwalten der Audit-Ausführung	Starten und Verwalten der Detect-Ausführung
CreateMitigationAction	CancelAuditMitigationActionsTask	CancelDetectMitigationActionsTask

Definieren und Verwalten von Aktionen	Starten und Verwalten der Audit-Ausführung	Starten und Verwalten der Detect-Ausführung
DeleteMitigationAction	DescribeAuditMitigationActionsTask	DescribeDetectMitigationActionsTask
DescribeMitigationAction	ListAuditMitigationActionsTasks	ListDetectMitigationActionsTasks
ListMitigationActions	StartAuditMitigationActionsTask	StartDetectMitigationActionsTask
UpdateMitigationAction	ListAuditMitigationActionsExecutions	ListDetectMitigationActionsExecutions

Verwenden von AWS IoT Device Defender mit anderen AWS-Services

Verwenden von AWS IoT Device Defender auf Geräten, die AWS IoT Greengrass ausführen

AWS IoT Greengrass bietet eine vordefinierte Integration in AWS IoT Device Defender zur kontinuierlichen Überwachung des Geräteverhaltens.

- Integrieren von Device Defender in AWS IoT Greengrass V1
- Integrieren von Device Defender in AWS IoT Greengrass V2

Verwenden von AWS IoT Device Defender mit FreeRTOS und eingebetteten Geräten

Für die Verwendung von AWS IoT Device Defender auf einem FreeRTOS-Gerät muss auf Ihrem Gerät das FreeRTOS Embedded C SDK oder die AWS IoT Device Defender-Bibliothek installiert sein. Das FreeRTOS Embedded C SDK enthält die AWS IoT Device Defender-Bibliothek. Weitere Informationen zur Integration von AWS IoT Device Defender in Ihre FreeRTOS-Geräte finden Sie in den folgenden Demos:

- AWS IoT Device Defender Demos für für FreeRTOS-Standardmetriken und benutzerdefinierte Metriken
- Verwenden des MQTT-Agenten zum Senden von Metriken an AWS IoT Device Defender
- Verwendung der MQTT-Kernbibliothek zum Senden von Metriken an AWS IoT Device Defender

Für die Verwendung von AWS IoT Device Defender auf einem eingebetteten Gerät ohne FreeRTOS muss auf Ihrem Gerät das AWS IoT Embedded C SDK oder die AWS IoT Device Defender-Bibliothek installiert sein. Das AWS IoT Embedded C SDK enthält die AWS IoT Device Defender-Bibliothek. Informationen zur Integration von AWS IoT Device Defender in Ihre eingebetteten Geräte finden Sie in den folgenden Demos: Demos zu Standard- und benutzerdefinierten Metriken für AWS IoT Device Defender für AWS IoT Embedded SDK.

Verwenden von AWS IoT Device Defender mit AWS IoT Device Management

Sie können die AWS IoT Device Management-Flottenindizierung verwenden, um Ihre AWS IoT Device Defender Detect-Verstöße zu indizieren, zu durchsuchen und zu aggregieren.

> **Note**
>
> Die Flottenindizierungsfunktion zur Unterstützung der Indizierung von Daten zu AWS IoT Device Defender-Verstößen befindet sich in der Vorschauphase von AWS IoT Device Management und unterliegt Änderungen.

- Verwalten der Flottenindizierung
- Abfragesyntax

Integration mit AWS Security Hub CSPM

AWS Security Hub CSPM liefert einen umfassenden Überblick über den Sicherheitsstatus in AWS und hilft Ihnen dabei, Ihre Umgebung anhand von Sicherheitsstandards und bewährten Methoden der Branche zu überprüfen. Security Hub CSPM sammelt Sicherheitsdaten aus Ihren gesamten AWS-Konten, Services und unterstützten Produkten von Drittanbietern. Mit Security Hub CSPM können Sie Ihre Sicherheitstrends analysieren und Sicherheitsprobleme mit höchster Priorität identifizieren.

Mit der AWS IoT Device Defender-Integration in Security Hub CSPM können Sie Erkenntnisse aus AWS IoT Device Defender an Security Hub CSPM senden. Security Hub CSPM bezieht diese Erkenntnisse dann in die Analyse Ihrer Sicherheitslage ein.

Inhalt

- Aktivieren und Konfigurieren der Integration
- So sendet AWS IoT Device Defender Erkenntnisse an Security Hub CSPM
  - Arten von Erkenntnissen, die AWS IoT Device Defender sendet
  - Latenz für das Senden von Erkenntnissen
  - Wiederholung, wenn Security Hub CSPM nicht verfügbar ist
  - Aktualisieren von vorhandenen Erkenntnissen in Security Hub CSPM

- Typische Erkenntnis von AWS IoT Device Defender
- So geben Sie an, dass keine Erkenntnisse mehr von AWS IoT Device Defender an Security Hub CSPM gesendet werden

Aktivieren und Konfigurieren der Integration

Vor der Integration von AWS IoT Device Defender in Security Hub CSPM müssen Sie Security Hub CSPM zunächst aktivieren. Informationen zur Aktivierung von Security Hub CSPM finden Sie unter Einrichten von Security Hub im AWS Security Hub-Benutzerhandbuch.

Nachdem Sie sowohl AWS IoT Device Defender als auch Security Hub CSPM aktiviert haben, öffnen Sie die Seite „Integrationen“ in der Konsole von Security Hub CSPM und wählen dann Ergebnisse akzeptieren für Audit, Detect oder beides aus. AWS IoT Device Defender beginnt, Erkenntnisse an Security Hub CSPM zu senden.

So sendet AWS IoT Device Defender Erkenntnisse an Security Hub CSPM

In Security Hub CSPM werden Sicherheitsprobleme als Erkenntnisse verfolgt. Einige Erkenntnisse stammen von Problemen, die von anderen AWS-Services oder von Produkten von Drittanbietern erkannt werden.

Security Hub CSPM bietet Tools zur Verwaltung von Erkenntnissen aus all diesen Quellen. Sie können Listen mit Erkenntnissen anzeigen und filtern und Details zu einer Erkenntnis anzeigen. Weitere Informationen finden Sie unter Anzeigen der Erkenntnisse im AWS Security Hub-Benutzerhandbuch. Sie können auch den Status einer Untersuchung zu einer Erkenntnis nachverfolgen. Weitere Informationen finden Sie unter Ergreifen von Maßnahmen zu Erkenntnissen im AWS Security Hub-Benutzerhandbuch.

Alle Erkenntnisse in Security Hub CSPM verwenden ein Standard-JSON-Format, das so genannte AWS-Security Finding Format (ASFF). Das ASFF enthält Details über die Ursache des Problems, die betroffenen Ressourcen und den aktuellen Status der Erkenntnis. Weitere Informationen zu ASFF finden Sie unter AWS-Security Finding-Format (ASFF) im AWS Security Hub-Benutzerhandbuch.

AWS IoT Device Defender ist einer der AWS-Services, die Erkenntnisse an Security Hub CSPM senden.

Arten von Erkenntnissen, die AWS IoT Device Defender sendet

Nachdem Sie die Integration in Security Hub CSPM aktiviert haben, sendet AWS IoT Device Defender Audit die generierten Erkenntnisse (sogenannte Prüfungszusammenfassungen) an Security Hub CSPM. Bei Prüfungszusammenfassungen handelt es sich um allgemeine Informationen zu einem bestimmten Prüfungstyp und einer bestimmten Prüfungsaufgabe. Weitere Informationen finden Sie unter Audit-Prüfungen.

AWS IoT Device Defender Audit sendet in jeder Audit-Aufgabe sowohl für Audit-Prüfungszusammenfassungen als auch für Audit-Erkenntnisse Aktualisierungen an Security Hub CSPM. Wenn alle in Audit-Prüfungen gefundenen Ressourcen konform sind oder eine Audit-Aufgabe abgebrochen wird, aktualisiert Audit die Prüfzusammenfassungen in Security Hub CSPM auf den Datensatzstatus ARCHIVIERT. Wenn eine Ressource für eine Audit-Prüfung als nonkonform gemeldet wurde, aber in der letzten Audit-Aufgabe als konform gemeldet wurde, ändert Audit den Status dieser Ressource zu „konform“ und aktualisiert auch die Erkenntnis in Security Hub CSPM auf den Datensatzstatus ARCHIVIERT.

AWS IoT Device Defender Detect sendet Erkenntnisse zu Verstößen an Security Hub CSPM. Zu diesen Erkenntnissen zu Verstößen gehören Machine Learning (ML), Statistikdaten und statisches Verhalten.

AWS IoT Device Defender sendet die Erkenntnisse unter Verwendung des AWS-Security Finding Format (ASFF) an Security Hub CSPM. In ASFF gibt das `Types`-Feld die Art der Erkenntnis an. Die Erkenntnisse von AWS IoT Device Defender können die folgenden Werte für `Types` haben.

Ungewöhnliches Verhalten

Der Erkenntnistyp für widersprüchliche MQTT-Client-IDs und Prüfungen von geteilten Gerätezertifikaten und der Erkenntnistyp für Detect.

Software- und Konfigurationsprüfung/Schwachstellen

Der Erkenntnistyp für alle anderen Audit-Prüfungen.

Latenz für das Senden von Erkenntnissen

Wenn AWS IoT Device Defender Audit eine neue Erkenntnis erstellt, wird diese sofort an Security Hub CSPM gesendet, nachdem die Audit-Aufgabe abgeschlossen ist. Die Latenz hängt vom Umfang der bei der Autit-Aufgabe generierten Erkenntnisse ab. Security Hub CSPM erhält die Erkenntnisse in der Regel innerhalb einer Stunde.

AWS IoT Device Defender Detect sendet Erkenntnisse zu Verstößen nahezu in Echtzeit. Wenn bei einem Verstoß ein Alarm ausgelöst oder überschritten wird (d. h. der Alarm wurde erstellt oder gelöscht), wird die entsprechende Erkenntnis von Security Hub CSPM sofort erstellt oder archiviert.

Wiederholung, wenn Security Hub CSPM nicht verfügbar ist

Wenn Security Hub CSPM nicht verfügbar ist, versuchen AWS IoT Device Defender Audit und AWS IoT Device Defender Detect so lange erneut, die Erkenntnisse zu senden, bis sie empfangen wurden.

Aktualisieren von vorhandenen Erkenntnissen in Security Hub CSPM

Nachdem eine AWS IoT Device Defender-Audit-Erkenntnis an Security Hub CSPM gesendet wurde, können Sie sie anhand der geprüften Ressourcen-ID und des Typs von Audit-Prüfung identifizieren. Wenn eine neue Audit-Erkenntnis mit einer nachfolgenden Audit-Aufgabe für dieselbe Ressource und dieselbe Audit-Prüfung generiert wird, sendet AWS IoT Device Defender Audit Aktualisierungen, um zusätzliche Beobachtungen der Erkenntnisaktivität an Security Hub CSPM zu reflektieren. Wenn bei einer nachfolgenden Audit-Aufgabe für dieselbe Ressource und dieselbe Audit-Prüfung keine zusätzliche Audit-Erkenntnis generiert wird, wird im Status der Ressource die Konformität mit der Audit-Prüfung festgestellt. AWS IoT Device Defender Audit archiviert dann die Erkenntnisse in Security Hub CSPM.

AWS IoT Device Defender Audit aktualisiert auch die Prüfungszusammenfassungen in Security Hub CSPM. Wenn bei einer Audit-Prüfung nonkonforme Ressourcen gefunden werden oder die Prüfung fehlschlägt, wird für die Erkenntnis von Security Hub CSPM der Status „Aktiv“ festgelegt. Andernfalls archiviert AWS IoT Device Defender Audit die Erkenntnis in Security Hub CSPM.

AWS IoT Device Defender Detect erstellt eine Erkenntnis von Security Hub CSPM, wenn ein Verstoß festgestellt wird (z. B. bei einem Alarm). Diese Erkenntnis wird nur aktualisiert, wenn eines der folgenden Kriterien erfüllt ist:

- Die Erkenntnis läuft bald in Security Hub CSPM ab und AWS IoT Device Defender sendet daher eine Aktualisierung, um die Erkenntnis auf dem neuesten Stand zu halten. Erkenntnisse werden 90 Tage nach der letzten Aktualisierung gelöscht – oder 90 Tage nach ihrer Erstellung, wenn es keine Aktualisierungen gibt. Weitere Informationen finden Sie unter Kontingente für Security Hub CSPM im AWS Security Hub-Benutzerhandbuch.
- Für den entsprechenden Verstoß wird der Alarm aufgehoben, daher wird der Status der Erkenntnis von AWS IoT Device Defender zu ARCHIVIERT aktualisiert.

Typische Erkenntnis von AWS IoT Device Defender

AWS IoT Device Defender sendet die Erkenntnisse unter Verwendung des AWS-Security Finding Format (ASFF) an Security Hub CSPM.

Das folgende Beispiel zeigt eine typische Erkenntnis von Security Hub CSPM für eine Audit-Erkenntnis. Der `ReportType` in `ProductFields` lautet `AuditFinding`.


```
  {
  "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"
    ]
  }
}
```


Das folgende Beispiel zeigt eine Erkenntnis von Security Hub CSPM für eine Audit-Prüfungszusammenfassung. Der `ReportType` in `ProductFields` lautet `CheckSummary`.

```
  {
  "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"
    ]
  }
}
```

Das folgende Beispiel zeigt eine typische Erkenntnis von Security Hub CSPM für einen AWS IoT Device Defender-Detect-Verstoß.

```
  {
  "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"
    ]
  }
}
```

So geben Sie an, dass keine Erkenntnisse mehr von AWS IoT Device Defender an Security Hub CSPM gesendet werden

Um anzugeben, dass keine Erkenntnisse mehr an Security Hub CSPM gesendet werden, können Sie entweder die Konsole von Security Hub CSPM oder die API verwenden.

Weitere Informationen finden Sie unter Deaktivieren und Aktivieren des Flows von Erkenntnissen aus einer Integration (Konsole) oder Deaktivieren des Flows von Erkenntnissen aus einer Integration (API von Security Hub CSPM, AWS CLI) im AWS Security Hub-Benutzerhandbuch.

Serviceübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine juristische Stelle, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiertere juristische Stelle zwingen kann, die Aktion auszuführen. In AWS kann der serviceübergreifende Identitätswechsel zu Confused-Deputy-Problem führen. Ein serviceübergreifender Identitätswechsel kann auftreten, wenn ein Service (der Anruf-Service) einen anderen Service anruft (den aufgerufenen Service). Der Anruf-Service kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden über den aufgerufenen Service zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Es gibt drei Ressourcen von Ihnen, auf die AWS IoT Device Defender zugreift, die durch das Confused-Deputy-Sicherheitsproblem betroffen sein können: das Ausführen von Audits, das Senden von SNS-Benachrichtigungen bei Verstößen gegen das Sicherheitsprofil und das Ausführen von Abschwächungsaktionen. Für jede dieser Aktionen müssen die Werte für `aws:SourceArn` wie folgt lauten:

- Für Ressourcen, die in der UpdateAccountAuditConfiguration-API übergeben wurden (RoleArn- und notificationTarget-RoleArn-Attribute), sollten Sie die Ressourcenrichtlinie mithilfe von `aws:SourceArn` als `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:` einschränken.
- Für Ressourcen, die in der CreateMitigationAction-API (das RoleArn-Attribut) übergeben wurden, sollten Sie die Ressourcenrichtlinie mithilfe von `aws:SourceArn` als `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:mitigationaction/`*`mitigationActionName`* einschränken.
- Für Ressourcen, die in der CreateSecurityProfile-API (das alertTargets-Attribut) übergeben wurden, sollten Sie die Ressourcenrichtlinie mithilfe von `aws:SourceArn` als `arn:`*`arnPartition`*`:iot:`*`region`*`:`*`accountId`*`:securityprofile/`*`securityprofileName`* einschränken.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Bedingungskontext-Schlüssel `aws:SourceArn` mit Platzhaltern (*) für die unbekannten Teile des ARN. Beispiel, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Das folgende Beispiel zeigt, wie Sie die globalen Bedingungskontext-Schlüssel `aws:SourceArn` und `aws:SourceAccount` in AWS IoT Device Defender verwenden können, um das Confused-Deputy-Problem zu vermeiden.

JSON

```
 {
"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:"
    }
  }
}
}
```

Bewährte Sicherheitsmethoden für Geräteagenten

Geringste Berechtigung

Dem Agentenprozess sollten nur die Mindestberechtigungen gewährt werden, die zum Ausführen seiner Aufgaben erforderlich sind.

Grundlegende Mechanismen

- Der Agent sollte nicht als Stammbenutzer ausgeführt werden.
- Der Agent sollte als dedizierter Benutzer in seiner eigenen Gruppe ausgeführt werden.
- Benutzern/Gruppen sollten für die zum Sammeln und Übertragen von Metriken erforderlichen Ressourcen Leseberechtigungen erteilt werden.
- Beispiel: read-only on /proc /sys für den Beispiel-Agenten.
- Ein Beispiel für die Einrichtung eines Prozesses für die Ausführung mit eingeschränkten Berechtigungen finden Sie in den Anweisungen zur Einrichtung, die dem Python Beispiel-Agenten beiliegen.

Es gibt eine Reihe von bekannten Linux-Mechanismen, mit denen Sie Ihren Agentenprozess weiter einschränken oder isolieren können:

Fortgeschrittene Mechanismen

- CGroups
- SELinux

- Chroot
- Linux-Namespaces

Betriebliche Ausfallsicherheit

Ein Agentenprozess muss bei unerwarteten Betriebsfehlern und Ausnahmen stabil sein und darf nicht abstürzen oder permanent beendet werden. Der Code muss mit Ausnahmen ordnungsgemäß umgehen und als Vorsichtsmaßnahme für den Fall eines unerwarteten Abbruchs (z. B. bedingt durch Systemneustarts oder unerkannte Ausnahmen) so konfiguriert werden, dass er automatisch neu gestartet wird.

Geringste Abhängigkeiten

Ein Agent muss in seiner Implementierung die geringstmögliche Anzahl von Abhängigkeiten (z. B. Drittanbieter-Bibliotheken) verwenden. Wenn die Nutzung einer Bibliothek aufgrund der Komplexität einer Aufgabe (z. B. Transportschichtsicherheit) gerechtfertigt ist, verwenden Sie nur gut gewartete Abhängigkeiten und richten Sie einen Mechanismus ein, um sie auf dem neuesten Stand zu halten. Wenn die hinzugefügten Abhängigkeiten Funktionen enthalten, die vom Agenten nicht verwendet werden und standardmäßig aktiv sind (z. B. Öffnen von Ports, Sockets), deaktivieren Sie sie in Ihrem Code oder über die Konfigurationsdateien der Bibliothek.

Prozessisolation

Ein Agentenprozess darf nur Funktionen enthalten, die zum Ausführen der Sammlung und Übertragung von Metriken erforderlich sind. Er darf kein Piggyback-Container auf anderen Systemprozessen sein oder Funktionen für andere außerhalb des Bereichs liegenden Anwendungsfällen implementieren. Darüber hinaus darf der Agentenprozess keine eingehenden Kommunikationskanäle erstellen, wie z. B. Domänen-Socket- und Netzwerk-Service-Ports, die es lokalen oder Remote-Prozessen erlauben würden, seinen Betrieb zu stören und seine Integrität und Isolierung zu beeinträchtigen.

Verdecktheit

Ein Agentenprozess darf nicht mit Schlüsselwörtern wie Sicherheit, Überwachung oder Prüfung, aus denen sein Zweck und sein Sicherheitswert hervorgeht, benannt werden. Generische Code-Namen oder zufällige und pro Gerät eindeutige Prozessnamen werden bevorzugt. Das gleiche Prinzip ist bei der Benennung des Verzeichnisses mit den Binärdateien des Agenten und bei allen Namen und Werten der Prozessargumente einzuhalten.

Geringste freigegebene Informationen

Alle auf Geräten bereitgestellte Agenten-Artefakte dürfen keine vertraulichen Informationen enthalten, z. B. privilegierte Anmeldeinformationen, Debugging und toten Code oder Inline-

Kommentare oder Dokumentationsdateien, die Details zur serverseitigen Verarbeitung der von Agenten gesammelten Metriken oder andere Details zu Backend-Systemen offenlegen.

Transport Layer Security

Um TLS-sichere Kanäle für die Datenübertragung einzurichten, muss ein Agentenprozess alle clientseitigen Validierungen, wie z. B. Zertifikatketten- und Domänennamens-Validierungen, auf der Anwendungsebene erzwingen, sofern sie nicht standardmäßig aktiviert sind. Darüber hinaus muss ein Agent einen Root-Zertifikatspeicher verwenden, der vertrauenswürdige Behörden und keine Zertifikate von kompromittierten Zertifikatausstellern enthalten.

Sichere Bereitstellung

Der Zugriff auf alle Agenten-Bereitstellungsmechanismen, wie z. B. Codeübertragungen oder -synchronisierungen und Repositorys mit seinen Binärdateien, Quellcode und alle Konfigurationsdateien (einschließlich vertrauenswürdiger Stammzertifikate), muss kontrolliert werden, um eine unbefugte Injektion oder Manipulation des Codes zu verhindern. Wenn sich die Bereitstellungsmechanismus auf Netzwerkkommunikation stützen, müssen kryptografische Methoden genutzt werden, um die Integrität der Bereitstellungsartefakte bei der Übertragung zu schützen.

Weitere Informationen

- Sicherheit in AWS IoT Device Defender
- Grundlagen des AWS IoT-Sicherheitsmodells
- Redhat: ein Stück Python
- 10 gängige Fallstricke in Python und wie sie sich vermeiden lassen
- Was bedeutet das Konzept der geringsten Rechte und warum ist es erforderlich?
- Top 10 der integrierten OWASP-Sicherheit 10
- OWASP IoT-Projekt

AWS IoT Device DefenderAnleitung zur Fehlerbehebung in

Helfen Sie uns, dieses Thema zu verbessern

Lassen Sie uns wissen, was dazu beitragen würde, es besser zu machen

Allgemeines

F: Gibt es Voraussetzungen für die Verwendung von AWS IoT Device Defender?

A: Wenn Sie von Geräten gemeldete Metriken nutzen möchten, müssen Sie zunächst einen Agenten auf Ihren über AWS IoT verbundenen Geräten oder Geräte-Gateways bereitstellen. Die Geräte müssen eine konsistente Client-ID oder einen konsistenten Objektnamen bereitstellen.

Audit

F: Ich habe eine Prüfung aktiviert und für meinen Audit wird für längere Zeit "In Bearbeitung" angezeigt. Ist etwas falsch? Wann kann ich Ergebnisse erwarten?

A: Wenn eine Prüfung aktiviert ist, beginnt die Datenerfassung unverzüglich. Wenn in Ihrem Konto eine große Datenmenge erhoben werden muss (Zertifikate, Objekte, Richtlinien usw.), sind die Ergebnisse der Prüfung jedoch möglicherweise nicht gleich nach der Aktivierung verfügbar.

Detect

F: Wie kann ich wissen, welche Schwellenwerte für Verhaltensweisen in einem AWS IoT Device Defender-Sicherheitsprofil festzulegen sind?

A: Erstellen Sie eine Verhaltensweise im Sicherheitsprofil zunächst mit niedrigen Schwellenwerten und weisen Sie sie einer Objektgruppe mit einer repräsentativen Reihe von Geräten zu. Sie können mit AWS IoT Device Defender die aktuellen Metriken anzeigen und die Schwellenwerte des Geräteverhaltens dann entsprechend Ihrem Anwendungsfall anpassen.

F: Ich habe eine Verhaltenweise erstellt, sie löst wider Erwarten aber keinen Verstoß aus. Wie lässt sich dies beheben?

A: Wenn Sie eine Verhaltensweise definieren, geben Sie an, wie sich Ihr Gerät erwartungsgemäß normal verhält. Beispiel: Sie haben eine Überwachungskamera, die nur über TCP-Port 8888

eine Verbindung mit einem zentralen Server herstellt. In diesem Fall erwarten Sie von ihr nicht, dass sie andere Verbindungen herstellt. Um benachrichtigt zu werden, wenn die Kamera eine Verbindung über einen anderen Port herstellt, können Sie z. B. die folgende Verhaltensweise definieren:

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 8888 ]
    }
  }
}
```

Wenn die Kamera eine TCP-Verbindung auf TCP-Port 443 herstellt, wäre dies ein Verstoß gegen das Geräteverhalten und würde eine Warnung auslösen.

F: Es wird gegen eine oder mehrere meiner Verhaltensweisen verstoßen. Wie hebe ich den Verstoß auf?

A: Warnungen werden gelöscht, nachdem das Gerät sich wieder, wie in den Verhaltensprofilen definiert, wie erwartet verhält. Die Verhaltensprofile werden bei dem Eingang von Metrikdaten für Ihr Gerät ausgewertet. Wenn das Gerät länger als zwei Tage keine Metriken veröffentlicht, wird das Verletzungsereignis automatisch auf `alarm-invalidated` gesetzt.

F: Ich habe die Verhaltensweise, gegen die verstoßen wurde, gelöscht. Wie stoppe ich die Warnungen?

A: Durch das Löschen einer Verhaltensweise werden alle zukünftigen Verstöße und Warnungen für diese Verhaltensweise gelöscht. Frühere Warnungen müssen aus Ihrem Benachrichtigungsmechanismus entfernt werden. Wenn Sie eine Verhaltensweise löschen, wird die Aufzeichnung der Verstöße gegen die betreffende Verhaltensweise jedoch genauso lange wie alle anderen Verstöße in Ihrem Konto beibehalten.

Geräte-Metriken

F: Ich sende Metrikberichte, von denen ich weiß, dass sie gegen meine Verhaltensweisen verstoßen, es werden aber keine Verstöße ausgelöst. Was ist los?

A: Stellen Sie sicher, dass Ihre Metrikberichte akzeptiert werden; dazu dient das Abonnement der folgenden MQTT-Themen:

```
$aws/things/THING_NAME/defender/metrics/FORMAT/rejected
$aws/things/THING_NAME/defender/metrics/FORMAT/accepted
```

Dabei ist `THING_NAME` der Name des Objekts, das die Metrik meldet, `FORMAT` ist entweder „JSON“ oder „CBOR“. Dies ist vom Format des Metrikberichts abhängig, den das Objekt sendet.

Nachdem Sie sich angemeldet haben, sollten Sie für jeden gesendeten Metrikbericht Nachrichten zu diesen Tehmen erhalten. Die Nachricht `rejected` weist auf ein Problem bei der Analyse des Metrikberichts hin. In der Nutzlast der Nachricht ist eine Fehlermeldung enthalten, um Ihnen die Korrektur von Fehlern in Ihrem Metrikbericht zu erleichtern. Die Nachricht `accepted` weist darauf hin, dass der Metrikbericht ordnungsgemäß analysiert wurde.

F: Was geschieht, wenn ich in meinem Metrikbericht eine leere Metrik sende?

A: Eine leere Liste von Ports oder IP-Adressen wird immer als in Übereinstimmung mit der entsprechenden Verhaltenweise gedeutet. Wenn die entsprechende Verhaltenweise verletzt wurde, wird der Verstoß gelöscht.

F: Warum können meine Gerätemetrikberichte Meldungen für Geräte enthalten, die sich nicht in der AWS IoT-Registrierung befinden?

Wenn Sie allen Objekten oder allen nicht registrierten Objekten ein oder mehrere Sicherheitsprofile zugewiesen haben, schließt AWS IoT Device Defender Metriken von nicht registrierten Objekten ein. Wenn Metriken von nicht registrierten Objekten ausgeschlossen werden sollen, können Sie die Profile anstatt allen Geräten nur allen registrierten Geräten zuweisen.

F: Ich sehe keine Meldungen von einem oder mehreren nicht registrierten Geräten, obwohl ich ein Sicherheitsprofil auf alle nicht registrierten Geräte oder alle Geräte angewendet habe. Wie lässt sich dies beheben?

Vergewissern Sie sich, dass der gesendete Metrikbericht mit einem der unterstützten Formate richtig formatiert ist. Weitere Informationen finden Sie unter Spezifikationen für Gerätemetriken. Vergewissern Sie sich, dass für die nicht registrierten Geräte eine konsistente Client-ID oder ein

konsistenter Objektname verwendet wird. Wenn der Objektname Steuerzeichen enthält oder wenn der Objektname länger als 128 Byte an UTF-8-codierten Zeichen ist, werden von Geräten gemeldete Nachrichten zurückgewiesen.

F: Was geschieht, wenn ein nicht registriertes Gerät zur Registrierung hinzugefügt oder die Registrierung eines registrierten Geräts aufgehoben wird?

A: Wenn ein Gerät zur Registrierung hinzugefügt oder daraus entfernt wird:

- Sie sehen zwei separate Verstöße für das Gerät (einen unter seinem registrierten Objektnamen, einen unter seiner nicht registrierten Identität), wenn es mit dem Veröffentlichen von Metriken für Verstöße fortfährt. Nach zwei Tagen erscheinen keine aktiven Verstöße für die alte Identität mehr, sie bleiben aber bis zu 14 Tage im Verlauf der Verstöße verfügbar.

F: Welchen Wert sollte ich in dem Berichts-ID-Feld in meinem Geräte-Metrikbericht angeben?

A: Verwenden Sie einen für jeden Metrikbericht eindeutigen Wert, der als positive ganze Zahl ausgedrückt wird. Üblicherweise wird eine Unix-Epochen-Zeitstempel verwendet.

F: Sollte ich für AWS IoT Device Defender-Metriken eine dedizierte MQTT-Verbindung erstellen?

A: Eine separate MQTT-Verbindung ist nicht erforderlich.

F: Mit welcher Client-ID sollte eine Verbindung mit veröffentlichten Geräte-Metriken herstellen?

Verwenden Sie für Geräte (Objekte), die sich in der AWS IoT-Registrierung befinden, den registrierten Objektnamen. Verwenden Sie für Geräte, die sich nicht in der AWS IoT-Registrierung befinden, eine konsistente ID, wenn Sie eine Verbindung mit AWS IoT herstellen. Diese Vorgehensweise erleichtert die Zuordnung von Verstößen zu Objektnamen.

F: Kann ich Metriken für ein Gerät mit einer anderen Client-ID veröffentlichen?

Es ist möglich, Metriken im Namen eines anderen Objekts zu veröffentlichen. Sie können dazu die Metriken in dem von AWS IoT Device Defender für dieses Gerät reservierten Thema veröffentlichen. Beispiel: `Thing-1` möchte Metriken für sich selbst und auch im Namen von `Thing-2` veröffentlichen. `Thing-1` sammelt seine eigenen Metriken und veröffentlicht sie im MQTT-Thema:

```
$aws/things/Thing-1/defender/metrics/json
```

`Thing-1` ruft dann Metriken von `Thing-2` ab und veröffentlicht sie im MQTT-Thema:

```
$aws/things/Thing-2/defender/metrics/json
```

F: Wie viele Sicherheitsprofile und Verhaltensweisen sind in meinem Konto zulässig?

A: Siehe AWS IoT Device DefenderEndpunkte und Kontingente.

F: Wie sieht eine prototypische Ziel-Rolle für ein Warnungsziel aus?

A: Für eine Rolle, die AWS IoT Device Defender zum Veröffentlichen von Warnungen an einem Warnungsziel (SNS-Thema) berechtigt, müssen zwei Voraussetzungen erfüllt sein:

- Eine Vertrauensstellung, die iot.amazonaws.com als vertrauenswürdige Entität angibt
- Eine angefügte Richtlinie, die AWS IoT zum Veröffentlichen in einem angegebenen SNS-Thema berechtigt Beispiel:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "arn:aws:sns:us-east-1:123456789012:example-topic"
        }
    ]
}
```

- Wenn es sich bei dem für die Veröffentlichung von Benachrichtigungen verwendeten SNS-Thema um ein verschlüsseltes Thema handelt, müssen AWS IoT neben der Berechtigung zur Veröffentlichung im SNS-Thema zwei weitere Berechtigungen erteilt werden. Beispiel:

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "sns:Publish",
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": "arn:aws:sns:us-east-1:123456789012:example-topic"
       }
```

```
    ]
}
```

F: Meine Übermittlung eines Metrikberichts mit einem benutzerdefinierten Metriktyp `number` schlägt mit der Fehlermeldung `Malformed metrics report` fehl. Was ist los?

A: Der Typ `number` nimmt nur einen einzelnen Metrikwert als Eingabe, aber wenn Sie den Metrikwert im DeviceMetrics-Bericht einreichen, müssen Sie ihn als Array mit einem einzelnen Wert übergeben. Stellen Sie sicher, dass Sie den Metrikwert als Array einreichen.

Fehler-Nutzlast:

```
{"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}}}
```

Fehlermeldung:

```
{"thingName":"myThing","status":"REJECTED","statusDetails":
{"ErrorCode":"InvalidPayload","ErrorMessage":"Malformed metrics
 report"},"timestamp":1635802047699}
```

Nutzlast ohne Fehler:

```
{"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}]}}
```

Antwort:

```
{"thingName":"myThing","12334567":1635800375,"status":"ACCEPTED","timestamp":1635801636023}
```

Sicherheit in AWS IoT Device Defender

Die Sicherheit in der Cloud hat für AWS höchste Priorität. Als AWS-Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die eingerichtet wurden, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und als Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Informationen zu den Compliance-Programmen, die für AWS IoT Device Defender gelten, finden Sie unter Im Rahmen des Compliance-Programms zugelassene AWS-Services.
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der geteilten Verantwortung bei der Verwendung von AWS IoT Device Defender einsetzen können. Die folgenden Themen veranschaulichen, wie Sie AWS IoT Device Defender zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren außerdem, wie Sie andere AWS-Services verwenden, um Ihre AWS IoT Device Defender-Ressourcen zu überwachen und zu schützen. Weitere Informationen zur Sicherheit in AWS IoT Core finden Sie im Kapitel Sicherheit im AWS IoT Core-Entwicklerhandbuch.

Themen

- Datenschutz in AWS IoT Device Defender
- Identity and Access Management für AWS IoT Device Defender
- Compliance-Validierung für AWS IoT Device Defender
- Ausfallsicherheit in AWS IoT Device Defender

Datenschutz in AWS IoT Device Defender

Das AWS-Modell der geteilten Verantwortung gilt für den Datenschutz in AWS IoT Device Defender. Wie in diesem Modell beschrieben, ist AWS für den Schutz der globalen Infrastruktur verantwortlich, in der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Weitere Informationen zum Datenschutz in Europa finden Sie im Zentrum für die Datenschutz-Grundverordnung (DSGVO).

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail-Trails zur Erfassung von AWS-Aktivitäten finden Sie unter Arbeiten mit CloudTrail-Trails im AWS CloudTrail-Benutzerhandbuch.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie unter Verwendung der Konsole, der API, der AWS CLI oder AWS SDKs mit AWS IoT Device Defender oder anderen AWS-Services arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet

werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Identity and Access Management für AWS IoT Device Defender

AWS Identity and Access Management (IAM) ist ein AWS-Service, mit dem Administratoren den Zugriff auf AWS-Ressourcen sicher steuern können. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (Berechtigungen besitzt) ist, AWS IoT Device Defender-Ressourcen zu nutzen. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- Funktionsweise von AWS IoT Device Defender mit IAM
- Beispiele für identitätsbasierte Richtlinien für AWS IoT Device Defender
- Fehlerbehebung für AWS IoT Device Defender-Identität und -Zugriff

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Fehlerbehebung für AWS IoT Device Defender-Identität und -Zugriff).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe Funktionsweise von AWS IoT Device Defender mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe Beispiele für identitätsbasierte Richtlinien für AWS IoT Device Defender).

Authentifizierung mit Identitäten

Sie melden sich über eine Authentifizierung mit Ihren Anmeldeinformationen bei AWS an. Sie müssen als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle authentifiziert sein.

Sie können sich als Verbundidentität anmelden, indem Sie Anmeldeinformationen von einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Google/Facebook-Anmeldeinformationen verwenden. Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung.

Für den programmatischen Zugriff bietet AWS ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto-Root-Benutzer

Wenn Sie ein AWS-Konto neu erstellen, enthält es zunächst nur eine einzelne Anmeldeidentität, die als Root-Benutzer für das AWS-Konto bezeichnet wird und über kompletten Zugriff auf sämtliche AWS-Services und Ressourcen im Konto verfügt. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Verbundidentität

Fordern Sie als Best Practice menschliche Benutzer auf, den Verbund mit einem Identitätsanbieter zu verwenden, um mit temporären Anmeldeinformationen auf AWS-Services zuzugreifen.

Eine Verbundidentität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Webidentitätsanbieter oder Directory Service, der mit Anmeldeinformationen aus einer Identitätsquelle auf AWS-Services zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie unter Menschliche Benutzer auffordern, den Verbund mit einem Identitätsanbieter zu verwenden, um mit temporären Anmeldeinformationen auf AWS zuzugreifen im IAM-Benutzerhandbuch.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einem Benutzer zu einer IAM-Rolle (Konsole) wechseln oder einen AWS CLI oder AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Für die Zugriffssteuerung in AWS erstellen Sie Richtlinien und weisen diese den AWS-Identitäten oder -Ressourcen zu. Eine Richtlinie definiert Berechtigungen, wenn sie einer Identität oder Ressource zugeordnet wird. AWS wertet diese Richtlinien aus, sobald ein Prinzipal eine Anfrage stellt. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können.

IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können verwaltete AWS-Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, die die maximalen Berechtigungen festlegen können, die von gängigeren Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs) – SCPs legen die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in AWS Organizations fest. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations-Benutzerhandbuch.

- Ressourcen-Kontrollrichtlinien (RCPs) – RCPs definieren die maximale Anzahl an Berechtigungen, die Ressourcen in Ihren Konten zur Verfügung stehen. Weitere Informationen finden Sie unter Ressourcen-Kontrollrichtlinien im AWS Organizations-Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen dazu, wie AWS die Zulässigkeit einer Anforderung ermittelt, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter Logik für die Richtlinienauswertung im IAM-Benutzerhandbuch.

Funktionsweise von AWS IoT Device Defender mit IAM

Bevor Sie IAM zum Verwalten des Zugriffs auf AWS IoT Device Defender verwenden, erfahren Sie, welche IAM-Funktionen Sie mit AWS IoT Device Defender verwenden können.

IAM-Features, die Sie mit verwenden können AWS IoT Device Defender

IAM-Feature	AWS IoT Device Defender-Support
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Teilweise
Temporäre Anmeldeinformationen	Ja

IAM-Feature	AWS IoT Device Defender-Support
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen Überblick über das Zusammenwirken von AWS IoT Device Defender und anderen AWS-Services mit den meisten IAM-Funktionen finden Sie unter AWS-Services, die mit IAM funktionieren im IAM-Benutzerhandbuch.

Identitätsbasierte -Richtlinien für AWS IoT Device Defender

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für AWS IoT Device Defender

Beispiele für identitätsbasierte Richtlinien in AWS IoT Device Defender finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS IoT Device Defender.

Ressourcenbasierte Richtlinien in AWS IoT Device Defender

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und

Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder umfassen AWS-Services.

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Richtlinienaktionen für AWS IoT Device Defender

Unterstützt Richtlinienaktionen: Ja

Administratoren können mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der AWS IoT Device Defender-Aktionen finden Sie in der Referenz zur Serviceautorisierung.

Richtlinienaktionen in AWS IoT Device Defender verwenden das folgende Präfix vor der Aktion:

```
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      ":action1",
      ":action2"
         ]
```

Beispiele für identitätsbasierte Richtlinien in AWS IoT Device Defender finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS IoT Device Defender.

Richtlinienressourcen für AWS IoT Device Defender

Unterstützt Richtlinienressourcen: Ja

Administratoren können mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste von AWS IoT Device Defender-Ressourcentypen und deren ARNs finden Sie in der Referenz zur Serviceautorisierung. Informationen dazu, mit welchen Aktionen Sie den ARN der einzelnen Ressourcen angeben können, finden Sie unter .

Beispiele für identitätsbasierte Richtlinien in AWS IoT Device Defender finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS IoT Device Defender.

Richtlinien-Bedingungsschlüssel für AWS IoT Device Defender

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Eine Liste aller globalen AWS-Bedingungsschlüssel finden Sie unter Globale AWS-Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

Eine Liste von AWS IoT Device Defender-Bedingungsschlüsseln finden Sie in der Referenz zur Serviceautorisierung. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter .

Beispiele für identitätsbasierte Richtlinien in AWS IoT Device Defender finden Sie unter Beispiele für identitätsbasierte Richtlinien für AWS IoT Device Defender.

ACLs in AWS IoT Device Defender

Unterstützt ACLs: Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit AWS IoT Device Defender

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-Entitäten und AWS-Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit AWS IoT Device Defender

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen bieten kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie den Verbund verwenden oder die Rolle wechseln. AWS Es wird

empfohlen, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporary security credentials in IAM und AWS-Services that work with IAM im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipal-Berechtigungen für AWS IoT Device Defender

Unterstützt Forward Access Sessions (FAS): Ja

Forward access sessions (FAS) verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für AWS IoT Device Defender

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Rollen zum Delegieren von Berechtigungen an einen AWS-Service erstellen im IAM-Benutzerhandbuch.

> **Warning**
>
> Die Änderung der Berechtigungen für eine Servicerolle kann die Funktionalität von AWS IoT Device Defender beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn AWS IoT Device Defender eine Anleitung dazu gibt.

Serviceverknüpfte Rollen für AWS IoT Device Defender

Unterstützt serviceverknüpfte Rollen: Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS-Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der

Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS IoT Device Defender

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, Ressourcen in AWS IoT Device Defender zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu den von AWS IoT Device Defender definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS IoT Device Defender in der Referenz zur Serviceautorisierung.

Themen

- Best Practices für Richtlinien
- Verwenden der AWS IoT Device Defender-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Ressourcen in AWS IoT Device Defender unter Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit von AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine Anwendungsfälle gewähren. Sie sind in Ihrem verfügbar AWS-Konto. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete AWS-Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS-Service, wie beispielsweise CloudFormation, verwendet werden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Bedarf einer Multi-Faktor-Authentifizierung (MFA) – Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der AWS IoT Device Defender-Konsole

Für den Zugriff auf die Konsole von AWS IoT Device Defender benötigen Sie einen Mindestsatz von Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details über die Ressourcen in AWS IoT Device Defender in Ihrem AWS-Konto aufzulisten und anzuzeigen. Wenn

Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Damit Benutzer und Rollen weiterhin die AWS IoT Device Defender-Konsole verwenden können, fügen Sie den Entitäten auch die verwaltete AWS IoT Device Defender *ConsoleAccess*- oder *ReadOnly* AWS-Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

```
{
    "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": "*"
        }
    ]
}
```

Fehlerbehebung für AWS IoT Device Defender-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die beim Arbeiten mit AWS IoT Device Defender und IAM auftreten könnten.

Themen

- Ich bin nicht autorisiert, eine Aktion in auszuführen. AWS IoT Device Defender
- Ich bin nicht zur Ausführung von iam:PassRole autorisiert
- Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine AWS IoT Device Defender-Ressourcen gewähren

Ich bin nicht autorisiert, eine Aktion in auszuführen. AWS IoT Device Defender

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to
 perform: :GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht zur Ausführung von iam:PassRole autorisiert

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an AWS IoT Device Defender übergeben zu können.

Einige AWS-Services erlauben die Übergabe einer vorhandenen Rolle an diesen Dienst, sodass keine neue Servicerolle oder serviceverknüpfte Rolle erstellt werden muss. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS IoT Device Defender auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine AWS IoT Device Defender-Ressourcen gewähren

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob AWS IoT Device Defender diese Funktionen unterstützt, finden Sie unter Funktionsweise von AWS IoT Device Defender mit IAM.

- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konto im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie AWS-Konten-Drittanbieter Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter Gewähren des Zugriffs auf AWS-Konten von externen Benutzern im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Compliance-Validierung für AWS IoT Device Defender

Unser neues AWS-Anmeldeverfahren ist nicht für regulierte Workloads ausgelegt. Wenn Sie unser neues AWS-Anmeldeverfahren nutzen, AWS jedoch für regulierte Workloads einsetzen möchten, können Sie sich für AWS registrieren (fortgeschritten) oder erweiterte Funktionen für Ihre AWS-Umgebung aktivieren.

Informationen dazu, ob ein AWS-Service in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter AWS-Services im Umfang nach Compliance-Programm. Wählen Sie dort das Compliance-Programm aus, das Sie interessiert. Allgemeine Informationen finden Sie unter AWS-Compliance-Programme.

Sie können Auditberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen in AWS Artifact.

Ihre Compliance-Verantwortung bei Verwendung von AWS-Services hängt von der Vertraulichkeit der Daten, den Compliance-Zielen des Unternehmens und den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Compliance-Verantwortung bei der Verwendung von AWS-Services finden Sie in der AWS-Sicherheitsdokumentation.

Ausfallsicherheit in AWS IoT Device Defender

Die globale AWS-Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk

mit geringer Latenz, hohem Durchsatz und hoher Redundanz verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter AWSGlobale Infrastruktur.

Neben der globalen AWS-Infrastruktur stellt AWS IoT Device Defender verschiedene Funktionen bereit, um Ihren Anforderungen an Ausfallsicherheit und Datensicherung gerecht zu werden.

Dokumentverlauf für das AWS IoT Device Defender-Benutzerhandbuch

Die folgende Tabelle beschreibt die Dokumentationsversionen von AWS IoT Device Defender.

Änderung	Beschreibung	Datum
Änderungen an der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender	Die Erkennungsfunktion von AWS IoT Device Defender wird für Neukunden ab dem 31. August 2026 nicht mehr verfügbar sein. Bestandskunden können die Funktion weiterhin verwenden. Weitere Informationen finden Sie unter Änderungen an der Verfügbarkeit der Erkennungsfunktion von AWS IoT Device Defender.	30. Juni 2026
Allgemein verfügbar	Dies ist die erste veröffentlichte Version von AWS IoT Device Defender.	02. August 2023
AWS IoT Device Defender unterstützt jetzt die Überwachung der Dauer von Geräteunterbrechungen	AWS IoT Device Defender Rules Detect unterstützt jetzt eine neue Metrik für die Dauer von Unterbrechungen, um die Dauer von Geräteunterbrechungen jedes Geräts zu überwachen. Mit dieser zusätzlichen Metrik können Sie verfolgen, wie lange ein Gerät getrennt war, um zu erfahren, ob es erwartungsgemäß funktioniert. Sie	20. Juli 2023

	können Alarme auch auf vordefinierte Schwellenwerte einstellen und bei anhaltenden Problemen mit der Gerätekon nektivität benachrichtigt werden. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	
AWS IoT Device Defender Das -Prüfungsfeature identifiziert potenzielle Fehlkonfigurationen in IoT-Richtlinien	Identifizieren Sie Fehler, beheben Sie Probleme und ergreifen Sie die erforderlichen Korrekturmaßnahmen mithilfe des Prüfungsfeatures. Dieses neue Feature hilft auch bei der Identifizierung von IoT-Richtlinien mit permissiven Allow-Anweisungen, durch die Geräte unbeabsichtigt Zugriff auf bestimmte Ressourcen erhalten könnten. Außerdem wird geprüft, ob MQTT-Platzhalter in Deny-Anwe isungen verwendet werden, die möglicherweise von Geräten umgangen werden könnten, wenn Platzhalter durch bestimmte Zeichenfo lgen ersetzt werden. Weitere Informationen finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwickle rhandbuch.	6. Dezember 2022

AWS IoT Device Defender Unterstützung für benutzerdefinierte -ML-Detect-Metriken und -Dimensionen	AWS IoT Device Defender unterstützt jetzt eine neue Auditprüfung für widerrufene temporäre Zertifizierungsstellen (CA). Wenn eine CA eine temporäre CA widerruft, weil sie potenziell kompromittiert ist, sind alle von dieser temporären CA ausgestellten Zertifikate ebenfalls potenziell kompromittiert und ungültig. Diese neue Auditprüfung identifiziert aktive Gerätezertifikate, die von einer widerrufenen temporären Zertifizierungsstelle ausgestellt wurden, und hilft Kunden dabei, diese aktiven Gerätezertifikate zu überprüfen und zu ersetzen. Weitere Informationen finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	10. November 2022

AWS IoT Device Defender Unterstützung für benutzerdefinierte -ML-Detect-Metriken und -Dimensionen	ML Detect unterstützt jetzt die Überwachung benutzerdefinierter Metriken, damit Sie für Ihre Flotte einzigartige Betriebszustandsparameter auswerten können. Neben der manuellen Einstellung statischer Alarme mit Rules Detect können Sie jetzt Machine Learning verwenden, um die erwarteten Verhaltensweisen Ihrer Flotte anhand benutzerdefinierter Metriken automatisch zu erlernen. Darüber hinaus können Sie mit dem neuerlich unterstützten Dimensionsfilter für ML Detect Attribute definieren, um genauere Metriken in Ihrem ML-Sicherheitsprofil auszuwerten. Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch	14. September 2022

Unterstützung von AWS IoT Device Management- und AWS IoT Device Defender-Gerätemetriken über die ListMetricValues-API	Greifen Sie mit der ListMetricValues-API von verbundenen Geräten, die zu einem Sicherheitsprofil gehören, auf historische geräteseitige, cloudseitige und benutzerdefinierte Metriken zu. Zusätzlich zum Anzeigen der Daten in der AWS-IoT-Management konsole können Sie jetzt Ihre eigene Visualisierung flexible programmgesteuert überwachen und erstellen. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	5. April 2022
AWS IoT Device Defender unterstützt jetzt Detect-Alarmverifizierungstatus	Überprüfen Sie einen Alarm basierend auf der Untersuchung festgestellter Verhaltensanomalien. Sie können einen Alarm als wahr positiv, gutartig positiv, falsch positiv oder unbekannt verifizieren und eine entsprechende Beschreibung angeben. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	24. September 2021

AWS IoT Device Defender Audit-One-Click-Version	Audit One-Click erleichtert es AWS-IoT-Core-Kunden, ihre Sicherheitsbasis zu verbessern, indem sie mit einem einzigen Klick die Prüfung ihres Kontos und ihrer IoT-Geräte anhand bewährter Sicherheitsmethoden beginnen können. Audit One-Click ermöglicht es Kunden, eine AWS IoT Device Defender-Prüfung mit voreingestellten Konfigurationen zu aktivieren, einschließlich der Aktivierung aller verfügbaren Prüfungen und eines täglichen Audit-Zeitplans. Es enthält zudem kontextbezogene Erläuterungen zu den Vorteilen regelmäßiger Sicherheitsprüfungen. Audit One-Click ist nur über die AWS-IoT-Konsole verfügbar. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	22. September 2021

AWS IoT Device Defender CloudFormation-Unterstützung	AWS IoT Device Defender unterstützt jetzt CloudFormation, um Ressourcen von AWS IoT Device Defender – wie beispielsweise geplante Audits und Sicherheitsprofile – auf sichere, effiziente und wiederholbare Weise zu erstellen und zu konfigurieren. Weitere Informationen zu den von AWS IoT Device Defender unterstützten CloudFormation-Ressourcentypen finden Sie in der Ressourcentypreferenz für AWS IoT.	5. März 2021
AWS IoT Device Defender bietet jetzt Unterstützung für benutzerdefinierte Metriken	Verwenden Sie AWS IoT Device Defender, um betriebliche Zustandsmetriken zu überwachen, die für Ihre Flotte oder Ihren Anwendungsfall einzigartig sind. Die Warnungen können in der Device-Defender-Konsole angezeigt oder über AWS Simple Notification Service (SNS) freigegeben werden. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	15. Dezember 2020

AWS IoT Device Defender startet die Unterdrückung von Prüfungsergebnissen	Mit dem Feature der Unterdrückung von Prüfungsergebnissen können Sie auswählen, welche Prüfungsergebnisse Sie anzeigen möchten, und nicht konforme Ergebnisse für bestimmte Ressourcen deaktivieren. Darüber hinaus können Sie die Unterdrückung von Prüfungsergebnissen für einen bestimmten Zeitraum oder auf unbestimmte Zeit konfigurieren. Die zugehörige Dokumentation finden Sie unter Prüfung im AWS IoT Device Defender-Entwicklerhandbuch.	12. August 2020
AWS IoT Device Defender unterstützt jetzt Dimensionen für die themenbasierte Metrikü berwachung	Mit dem Dimensionen-Feature können Kunden die Metriken filtern, die Device Defender Detect nach MQTT-Themen auswertet. Dimensionen unterstützen die folgenden cloudseitigen Metriken: Anzahl der empfangenen Nachrichten, Nachrichten-Bytegröße, Anzahl der gesendeten Nachrichten, Quell-IP und Anzahl der Autorisierungsfehler. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	2. April 2020

AWS IoT Device Defender Allgemeine Verfügbarkeit von ML Detect	Das ML-Detect-Feature von AWS IoT Device Defender erkennt durch Lernen aus früheren Daten automatisch operative und Sicherheitsanomalien auf Geräteebene in Ihrer Flotte. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	24. März 2020

AWS IoT Device Defender Die Prüfungsfähigkeit von wurde um vier neue Prüfungen erweitert	Verwenden Sie AWS IoT Device Defender Audit, um nach Geräten in Ihrer Flotte zu suchen, die übermäßig permissive Berechtigungen und Zugriff auf Services haben, die seit 365 Tagen nicht mehr verwendet wurden, OpenSSL-Versionen auf Debian-basierten Betriebss ystemen verwenden, die als vorhersehbare kryptogra fische Schlüssel identifiziert wurden, die sie anfällig für Brute-Force-Angriffe machen, oder Infineon-RSA-Bibli otheksversionen verwenden , die die RSA-Schlüsselgener ierung nachweislich falsch handhaben, wodurch sie anfällig für Hacking sind. Die zugehörige Dokumentation finden Sie unter Prüfung im AWS IoT Device Defender-Entwicklerhandbuch.	25. November 2019
AWS IoT Device Defender unterstützt Abschwäch ungsaktionen für Prüfungse rgebnisse	AWS IoT Device Defender unterstützt die Möglichkeit für Kunden, Abschwächungsaktio nen auf Prüfungsergebnisse anzuwenden. Die zugehörig e Dokumentation finden Sie unter Prüfung im AWS IoT Device Defender-Entwickle rhandbuch.	6. August 2019

AWS IoT Device Defender unterstützt die Überwachung des Verhaltens nicht registrierter Geräte	Identifizieren Sie ungewöhnliches Verhalten für Geräte, die nicht in der AWS-IoT-Core-Registrierung eingetragen sind. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	15. Mai 2019
AWS IoT Device Defender bietet jetzt Erkennung von statischen Anomalien und Datenvisualisierung	Verwenden Sie die statistische Anomalieerkennung und erhalten Sie Warnungen, wenn ein Gerät nicht innerhalb des perzentilbasierten Schwellenwerts liegt. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	19. Februar 2019
AWS IoT Device Defender unterstützt jetzt die Überwachung der Dauer von Geräteunterbrechungen	AWS IoT Device Defender unterstützt jetzt zwei zusätzliche cloudseitige Metriken: die Anzahl der Verbindungsversuche und die Anzahl der Verbindungsabbrüche. Die zugehörige Dokumentation finden Sie unter Cloudseitige Metriken im AWS IoT Device Defender-Entwicklerhandbuch.	19. Dezember 2018